

Міністерство освіти і науки України  
Сумський державний університет  
Навчально-науковий інститут бізнес-технологій «УАБС»  
Кафедра економічної кібернетики

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

на тему «АВТОМАТИЗАЦІЯ ПЛАНУВАННЯ РОБІТ СЛУЖБИ  
ПРОГРАМНОГО СУПРОВОДУ ПІДПРИЄМСТВА»

Виконав студент 2 курсу, групи ЕК.м-71а  
(номер курсу) (шифр групи)

Спеціальності 051 «Економіка»  
(«Економічна кібернетика»)

Олексенко С. В.

(прізвище, ініціали студента)

Керівник професор, д.е.н., Олійник В.М.

(посада, науковий ступінь, прізвище, ініціали)

Суми – 2018 рік

## РЕФЕРАТ

### кваліфікаційної магістерської роботи на тему «АВТОМАТИЗАЦІЯ ПЛАНУВАННЯ РОБІТ СЛУЖБИ ПРОГРАМНОГО СУПРОВОДУ ПІДПРИЄМСТВА»

студента Олексенка Станіслава Вікторовича

Актуальність теми кваліфікаційної роботи пояснюється тим, що при нинішніх тенденціях розвитку технологій, для того щоб підприємство було конкурентоздатним та ефективним, необхідно досягти максимальної корисної продуктивності працівників. Саме через це необхідно звести до мінімуму час, який затрачається на формування та узгодження різного роду документів, серед яких план робіт та звіт про виконані роботи.

Метою даної роботи є створення прототипу web-додатку для автоматизації процесу формування плану робіт та звіту про виконану роботу.

Об'єктом дослідження у роботі є діяльність служби програмного супроводу дирекції з інформаційних технологій ПАТ «Сумиобленерго».

Предметом дослідження є процес формування плану робіт та звіту про виконану роботу.

У відповідності до завдань даної роботи було здійснено: дослідження суті поставленої задачі та предметної області; дослідження існуючих інформаційних системи для автоматизації процесу формування документів; розробку вимог до створюваної системи; розробку проекту та прототипу автоматизованої системи; визначення економічного ефекту від впровадження системи.

Результатом даної роботи є створений прототип автоматизованої системи, що містить в собі всі необхідні функціональні модулі та ергономічний інтерфейс. Даний прототип рекомендується використовувати на малих та середніх підприємствах.

Ключові слова: автоматизація, планування робіт, технологія AJAX, web-додаток, всесвітня мережа Інтернет, СУБД MySQL, phpMyAdmin, PHPWord, PHP, ефективність, бізнес-процес, word-документ.

Основний зміст кваліфікаційної магістерської роботи викладено на 48 сторінках, зокрема список використаних джерел із 71 найменувань, розміщений на 7 сторінках. Робота містить 3 таблиці, 12 рисунків, а також 13 додатків, розміщених на 40 сторінках.

Рік виконання кваліфікаційної роботи – 2018 рік

Рік захисту роботи – 2018 рік

Міністерство освіти і науки України  
Сумський державний університет  
Навчально-науковий інститут бізнес-технологій «УАБС»  
Кафедра економічної кібернетики

ЗАТВЕРДЖУЮ  
Завідувач кафедри  
д.е.н., доцент  
\_\_\_\_\_ О.В. Кузьменко  
“\_\_” \_\_\_\_\_ 2018 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ МАГІСТЕРСЬКУ РОБОТУ  
(спеціальність 051 «Економіка («Економічна кібернетика»))  
студенту 2 курсу, групи ЕК.м-71а

\_\_\_\_\_ Олексенка Станіслава Вікторовича

(прізвище, ім'я, по батькові студента)

1. Тема роботи Автоматизація планування робіт служби програмного супроводу підприємства

затверджена наказом по університету від «\_\_» \_\_\_\_\_ 2018 року № \_\_\_\_\_

2. Термін подання студентом закінченої роботи «\_\_» \_\_\_\_\_ 20\_\_ року

3. Мета кваліфікаційної роботи створення прототипу web-додатку для автоматизації процесу формування плану робіт та звіту про виконану роботу

4. Об'єкт дослідження діяльність служби програмного супроводу дирекції з інформаційних технологій ПАТ «Сумиобленерго»

5. Предмет дослідження процес формування плану робіт та звіту про виконану роботу

6. Кваліфікаційна робота виконується на матеріалах

7. Орієнтовний план кваліфікаційної роботи, терміни подання розділів керівникові та зміст завдань для виконання поставленої мети

Розділ 1 Аналіз стану автоматизації ПАТ«Сумиобленерго»

(назва – термін подання)

У розділі 1 проаналізувати діяльність ПАТ «Сумиобленерго», визначити рівень автоматизації підприємства та перспективні напрямки розвитку, сформулювати завдання та обмеження автоматизованої системи

(зміст конкретних завдань до розділу, які повинен виконати студент)

Розділ 2 Проектування web-додатку для автоматизації формування плану робіт

(назва – термін подання)

У розділі 2 сформувані вимоги та обмеження при реалізації системи, визначити етапи її створення, визначити методи управління інформаційними потоками

(зміст конкретних завдань до розділу, які має виконати студент)

Розділ 3 Реалізація прототипу автоматизованого рішення формування плану робіт на підприємстві

(назва – термін подання)

У розділі 3 розробити архітектуру системи, реалізувати серверну логіку web-додатку, оцінити ефективність створеної системи

(зміст конкретних завдань до розділу, які повинен виконати студент)

8. Консультації з роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1			
2			
3			

9. Дата видачі завдання: «\_\_\_» \_\_\_\_\_ 20\_\_ року

Керівник кваліфікаційної роботи

\_\_\_\_\_ ( підпис)

\_\_\_\_\_ (ініціали, прізвище)

Завдання до виконання одержав

\_\_\_\_\_ (підпис)

\_\_\_\_\_ (ініціали, прізвище)

## ЗМІСТ

ВСТУП .....	3
1 АНАЛІЗ СТАНУ АВТОМАТИЗАЦІЇ ПАТ «СУМІОБЛЕНЕРГО».....	5
1.1 Характеристика ПАТ «Сумиобленерго» .....	5
1.2 Рівень автоматизації підприємства та перспективні напрямки розвитку ...	8
1.3 Визначення задач та обмежень автоматизованої системи.....	16
2 ПРОЕКТУВАННЯ WEB-ДОДАТКУ ДЛЯ АВТОМАТИЗАЦІЇ ФОРМУВАННЯ ПЛАНУ РОБІТ .....	20
2.1 Обмеження та вимоги при реалізації web-додатку.....	20
2.2 Основні етапи розробки додатку .....	24
2.3 Управління інформаційними потоками .....	31
3 РЕАЛІЗАЦІЯ ПРОТОТИПУ АВТОМАТИЗОВАНОГО РІШЕННЯ ФОРМУВАННЯ ПЛАНУ РОБІТ НА ПІДПРИЄМСТВІ .....	35
3.1 Розробка архітектури автоматизованої системи.....	35
3.2 Реалізація серверної логіки web-додатку .....	41
3.3 Оцінка ефективності автоматизованої системи .....	45
ВИСНОВКИ.....	49
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	51
ДОДАТКИ.....	58

## ВСТУП

В процесі функціонування будь-якого підприємства не від'ємною складовою, яка є запорукою виконання всіх необхідних задач, що стоять перед робітниками підприємства, є план робіт працівників, обмежений певними часовими проміжками. Саме наявність чіткого плану дозволить максимально підвищити продуктивність, ефективність роботи та зайнятість працівників. Зі сторони керівництва, аналіз відповідності плану робіт та звіту про їх виконання, дозволить чітко контролювати ефективність роботи кожного з працівників та визначити їх потенційну продуктивність.

Процес зіставлення такого плану робіт, в відділі програмного супроводу ПАТ «Сумиобленерго», передбачає собою довготривале узгодження його пунктів. Адже кожен робітник сформувавши для себе план робіт на певний період, повинен узгодити його з керівником підрозділу, а потім вже він має представити план робіт працівників свого підрозділу на затвердження начальнику відділу. Очевидно, що весь цей процес забирає досить багато часу, який міг би бути використаний для виконання власне своїх професійних обов'язків працівниками підприємства. Тому виникає необхідність дослідження можливих варіантів автоматизації даного процесу.

Актуальність даної проблеми полягає в тому, що при нинішніх тенденціях розвитку технологій, для того щоб підприємство було конкурентоздатним та ефективним, необхідно досягти максимальної корисної продуктивності працівників. Саме через це необхідно звести до мінімуму час, який затрачається на формування та узгодження різного роду документів, в нашому випадку це плану робіт на місяць та звіту про виконання цих робіт. Вибір формату рішення цієї проблеми у вигляді web-додатку, робить його актуальним варіантом вирішення даної проблеми на малих та середніх підприємствах через його цінову доступність та простоту впровадження.

Об'єктом дослідження є служба програмного супроводу дирекції з інформаційних технологій ПАТ «Сумиобленерго».

Предметом дослідження є процес формування плану робіт та звіту про виконану роботу.

Метою дослідження є створення прототипу web-додатку для автоматизації процесу формування плану робіт та звіту про виконану роботу.

Завданням дослідження є:

- дослідження існуючої структури підприємства ПАТ«Сумиобленерго» в цілому та безпосередньо служби програмного супроводу дирекції з інформаційних технологій;

- визначення рівня автоматизації підприємства та перспективні напрямки розвитку

- вивчення процесу формування плану робіт та звіту про виконану роботу, з визначенням відповідних прав працівників на редагування, перегляд та створення пунктів, та документа в цілому;

- визначити та дослідити вимоги до додатку, що накладаються технічними обмеженнями серверу та робочих машин майбутніх користувачів, і відповідно до цього обрати серверну мову програмування, систему управління базою даних та їх необхідні версії;

- визначити та дослідити технології та методи, якими знадобиться користуватися для виконання поставлених завдань;

- безпосередня розробка прототипу додатку з урахуванням всіх вимог та обмежень покладених на нього.

Розроблений додаток було впроваджено у тестовому режимі в службі програмного супроводу дирекції з інформаційних технологій ПАТ «Сумиобленерго».



# 1 АНАЛІЗ СТАНУ АВТОМАТИЗАЦІЇ ПАТ «СУМІОБЛЕНЕРГО»

## 1.1 Характеристика ПАТ «Сумиобленерго»

У 1931 році було створено підприємство „Суменерго”, для подальшого розвитку електричних мереж міст Сумської області. За роки Великої Вітчизняної війни електроенергетичне господарство в багатьох містах області було повністю зруйноване. Відразу після визволення міста Суми 2 вересня 1943 року, було відновлено роботу „Суменерго”. Галузь почала стрімко розвиватись, постійно створювалися нові та модернізувалися старі електричні станції, так за для задоволення зростаючих електричних потреб в 1957 році в місті Суми була побудована теплова електрична станція потужністю 24 мВт.

В цей період розпочалося широке впровадження електричної енергії і в сільськогосподарське виробництво. Це вимагало прокладення нових високовольтних ліній електропередач для обслуговування яких в 1965 році було створено Сумське підприємство високовольтних мереж. Міські електричні мережі стало обслуговувати Сумське виробниче підприємство електричних мереж. Сільські електричні мережі спочатку підпорядковувались в обласній конторі „Сільенерго”, але в 1960 році було проведено її реорганізацію в обласне експлуатаційне енергетичне управління сільського господарство.

Розташування на території області двох окремих спеціалізованих підприємств, одне з яких обслуговувало високовольтні мережі, а друге – розподільчі сільські електромережі, породжувало паралелізм та дублювання в роботі. Міністерством Енергоносіїв і електрифікації УРСР було вирішено розукрупнити існуючі підприємства і створити на їх основі нові комплексні територіальні підприємства обсяги обслуговування яких приблизно збігалися.

В 1978 році в Сумській області було створено три самостійних підприємства – Сумське центральне підприємство електричних мереж (СЦПЕМ), Сумське південне підприємство електричних мереж (СППЕМ) і Шосткінське підприємство електричних мереж (ШПЕМ). Підприємства стали підпорядковуватись виробничому енергетичному управлінню „Харківенерго”.

В такому складі підприємства працювали до 1995 року, коли відповідно до Указу Президента України № 282 від 04.04.1995 р. створено Державну акціонерну компанію „Сумиобленерго”. Основна діяльність якої – прийом, передача, розподіл та постачання електричної енергії споживачам.

01.10.1998 року рішенням загальних зборів акціонерів створено ВАТ „Сумиобленерго”, яке стало правонаступником усіх прав та зобов’язань Державної акціонерної компанії „Сумиобленерго”.

22.04.2011 року згідно вимог Закону України «Про акціонерні товариства» рішенням загальних зборів акціонерів було змінено назву на Публічне Акціонерне Товариство "Сумиобленерго". Засновником товариства виступає Держава в особі Міністерства енергетики України. За структурою власності 25% акцій належать державі, а 75% - акціонерам.

На початок 2018 року ПАТ "Сумиобленерго" обслуговувало 520 835 абонентів, в тому числі 507 802 побутових абонента.

Мета діяльності підприємства є:

- задоволення потреб ринку електроенергетики області у продукції, роботах та послугах, розширення асортименту, підвищення якості обслуговування клієнтів, ефективне управління майном, одержання прибутку, забезпечення інтересів акціонерів і задоволення економічних та соціальних потреб працівників;
- передача і надійне постачання електричної енергії кінцевим споживачам на умовах укладених договорів та по тарифах, що регулюються відповідно до чинного законодавства;

- проведення єдиної науково-технічної політики та впровадження нових прогресивних методів та технологій надання послуг та обслуговування клієнтів;

Предметом діяльності підприємства є:

- передача та розподіл електричної енергії місцевим електричним мережам;
- експлуатація ліній електропередач і підстанцій, а також підтримка їх в належному морально-технічному стані;
- виробництво, передача та постачання теплової енергії.

Структура управління в ПАТ «Сумиобленерго» включає наступні органи:

- Загальні збори – вищий орган товариства;
- Наглядова рада – орган, який здійснює захист прав акціонерів, і в межах компетенції, визначеної законодавством, статутом та положенням про Наглядову раду, контролює та регулює діяльність Правління;
- Правління – колегіальний виконавчий орган товариства, який в межах компетенції, здійснює управління поточною діяльністю підприємства;
- ревізійна комісія – орган, який в межах компетенції, здійснює перевірку господарської діяльності підприємства.

Посадові особи органів товариства – фізичні особи – Голова та члени Наглядової ради, Правління, Ревізійної комісії.

Посадові особи органів товариства зобов'язані:

- ставитися з відповідальністю до виконання своїх обов'язків;
- приймати рішення в межах наданих їм повноважень;
- не використовувати своє становище у власних інтересах;
- не розголошувати конфіденційну інформацію про діяльність товариства, крім випадків, передбачених законодавством.

До складу підприємства ПАТ «Сумиобленерго» входить 16 районних філій РЕМ (район електричних мереж), Учбовий центр ПАТ

«Сумиобленерго» та філія «Сумське міжрайонне відділення «Енергозбут» Публічного акціонерного товариства «Сумиобленерго».

Саме в останній філії знаходиться відділ інформаційних технологій основними завданнями якого є:

- забезпечення функціонування діючих на підприємстві програмно-апаратних комплексів;
- розробка та впровадження нового програмного забезпечення;
- удосконалення автоматизації діяльності підрозділів ПАТ «Сумиобленерго»;
- обслуговування комп'ютерних мереж та корпоративної мережі підприємства;
- обробка даних зі всіх районних філій, її систематизація та внесення до загальної бази даних;
- підготовка заходів по забезпеченню інформаційної безпеки та контроль за їх виконанням.

Управлінням відділом здійснює начальник та його заступник, яким підпорядковуються керівники структурних підрозділів, котрі в свою чергу координують роботу рядових працівників. Одним зі підрозділів відділу є служба програмного супроводу саме для якої й має бути розроблений додаток з подальшим його впровадженням для всього відділу інформаційних технологій.

## 1.2 Рівень автоматизації підприємства та перспективні напрямки розвитку

В енергетичному секторі усього світу все більш очевидним стає напрямок автоматизації процесів по виконанню профільних робіт та суміжних з ними задач. Значний вплив при цьому мають інформаційні

технології, які впливають на якість надання послуг енергопостачання, обслуговування об'єктів електромереж та продовження їх життєвого циклу.

Для того, щоб на нинішньому етапі розвитку утримати існуючих та залучити нових клієнтів на конкурентному ринку енергетики, потрібно активно впроваджувати здобутки ІТ-технологій, які безпосередньо підвищують продуктивність й ефективність персоналу, та спрощують обслуговування і взаємодію зі споживачами.

Впровадження нових технологій в українських енергетичних компаніях проходить поступово. До цього часу витрати на ІТ все ще традиційно фінансуються за залишковим принципом, через те, що належать до незахищених статей інвестиційного і фінансового планування. Обмеження інвестицій в цю сферу зумовлене методами формування тарифу, які поки ще не відповідають ринковим правилам. Учасники ринку очікують впровадження рішення – RAB-тарифу, який підтримає реалізацію відповідних проектів у поточних умовах і в умовах підготовки до нового ринку електроенергії [4].

Компанії розуміють, що саме це дозволить підвищити якість роботи в цілому, дасть змогу досягти нових рівнів ефективності, закладе платформу для потенційного впровадження цифрових технологій. Однією з таких компаній є наш об'єкт дослідження – ПАТ «Сумиобленерго». Саме дане підприємство одне з перших українських енергетичних підприємств пройшло фазу трансформації і впровадити єдину інформаційну систему.

ПАТ "Сумиобленерго" слідкує за інноваційними технологіями, і регулярно удосконалює свою внутрішню та зовнішню діяльність. Така ІТ-стратегія є не забаганкою, а необхідністю. Адже в сучасному світі зростання потоків інформації відбувається з кожним днем все швидше, а сучасні системи (такі, як SAP ERP) допомагають проводити структурування даних та надають зручні інструменти для операцій з ними.

Впровадження SAP – це не інформаційно-технологічний проект, а принципова бізнес-стратегія товариства. Вибір саме цієї системи управління

є не випадковим. Компанія SAP є провідним виробником стандартизованого прикладного програмного забезпечення, призначених для автоматизації промислових підприємств. Система ERP дає можливість всім структурним одиницям ПАТ "Сумиобленерго" функціонувати як єдиний організм – запорукою чого є інтеграція всіх функцій, таких, як бухгалтерія та фінанси, персонал та адміністрація.

Впровадження системи SAP ERP на ПАТ "Сумиобленерго" стартувало ще у 2014 році з підготовки та занесення даних про енергетичні об'єкти філій та служб працівниками підприємства в модуль ТОРО (технічне обслуговування та ремонт обладнання). Внесені дані про технічні об'єкти і устаткування дозволяють вести плани та графіки ремонтів, реєструвати актуальний стан обладнання, вести облік їх дефектів та відключень, формувати й обробляти заявки, отримувати розгорнуту звітність про стан електричних мереж у різних ракурсах[7].

Масштаби ПАТ «Сумиобленерго» вражають, але проте навіть така складна структура не стала перепорою перед впровадження технологій на базі уніфікованої платформи. Активна фаза установки і запуску компонентів системи для автоматизації роботи підприємства розпочалася в січні 2017 року і зайняла 8 місяців до її повного впровадження.

Так як одним з найважливіших завдань підприємства є максимальне задоволення потреб клієнтів, головною задачею впровадженої системи стало підвищення якості енергопостачання. На разі підготовка і передача даних між структурними одиницями стала займати значно менше часу, адже всі служби та підрозділи функціонують в єдиній системі. За кожною одиницею обладнання досить легко відстежити історію експлуатації. Як наслідок, можна своєчасно попередити неполадку та усунути проблему.

На шляху досягнення європейських стандартів якості роботи, українським енергетичним компаніям необхідно вирішити проблему формування оперативної та точної звітності. Для ПАТ «Сумиобленерго» єдина система для бухгалтерського і податкового обліку, що дозволяє

формування фінансової звітності відповідно до вимог міжнародних стандартів у комплексі SAP ERP була швидко та успішно впроваджена, що дозволило скоротити формування звіту про неліквідні активи з 7 днів до 15 хвилин.

З огляду на все це стає зрозуміло, що дане підприємство чітко націлене на максимальну автоматизацію всіх процесів, що дозволить максимально корисно використовувати робочий час працівників. Дані заходи дозволять зменшити час, що затрачається на рутинну роботу і дозволить докласти більших зусиль до покращення якості надання послуг. Можливим позитивним ефектом, для підприємства, стане можливість скорочення персоналу, що дозволить підняти зарплатню іншим робітникам, що на фоні тотального відтоку робочої сили з України, дозволить зберегти в себе висококваліфікованих та перспективних робітників[12].

Саме тому стає зрозуміла актуальність розробки web-додатку для автоматизації процесу формування планів робіт та звітів про виконані роботи. Адже хоча це і не забирає дуже багато часу у рядових працівників, проте цей процес стає справжньою проблемою для керівників підрозділів та начальника відділу, адже постає необхідність узгодження кожного пункту роботи з відповідним працівником. Тому дане рішення пришвидшить виконання відповідного процесу і буде сприяти руху в напрямку автоматизації рутинних процесів на підприємстві.

На підприємстві ПАТ «Сумиобленерго» прийнято положення згідно з яким, кожен відділ підприємства до початку місяця повинен подати план робіт працівників відділу на місяць. Відповідний план має відповідати стандартизованому вигляду. В ньому повинно бути зазначені роботи які повинен виконувати кожен працівник в відповідному місяці. При цьому вказується період виконання роботи, відповідно, вона може відбуватися:

- в певний день місяця;
- до певного дня місяця;
- в де-який період протягом місяця;
- на протязі всього місяця.

При розробці плану передбачається, що кожна з робіт займе у працівника певну частину його робочого часу. Відповідне значення записується у спеціальну колонку «Показник» у відсотках, з розрахунку, що весь робочий час працівника протягом місяця приймається за 100%. Показники всіх робіт по певному працівнику за місяць в сумі повинно дорівнювати 100%. У випадку коли одна робота виконується декількома працівниками, відповідна задача дублюється для кожного з працівників.

Алгоритм формування плану робіт на місяць має наступний вигляд:

1. Працівник особисто формує для себе план робіт на місяць.
2. Працівник проводить редагування робіт з вказуванням необхідного періоду виконання та показника витрати часу, та передає на затвердження керівнику підрозділу.
3. Керівник підрозділу переглядає запропонований план та вирішує чи задовольняє його сформований план, якщо так то пункт 5, якщо ні – пункт 4.
4. Вносить рекомендації для працівника, що необхідно змінити, додати чи видалити, та передає на доопрацювання – пункт 2.
5. Вносить до єдиного плану робіт по всіх працівниках підрозділу, формує для себе план робіт та передає на затвердження начальнику відділу.
6. Начальник відділу переглядає поданій план робіт працівників підрозділу та вирішує чи задовольняє його сформований план, якщо так то пункт 8, якщо ні – пункт 7.
7. Вносить рекомендації, що до необхідних змін по кожному з працівників, та передає на доопрацювання керівнику підрозділу – пункт 4.
8. Затверджує сформований план та передає для ознайомлення кожним працівником з відповідними роботами для виконання.
9. Працівники підтверджують ознайомлення з роботами для виконання підписом в відповідній колонці документу.

Блок-схема відповідного процесу представлена на рисунку 1.1.



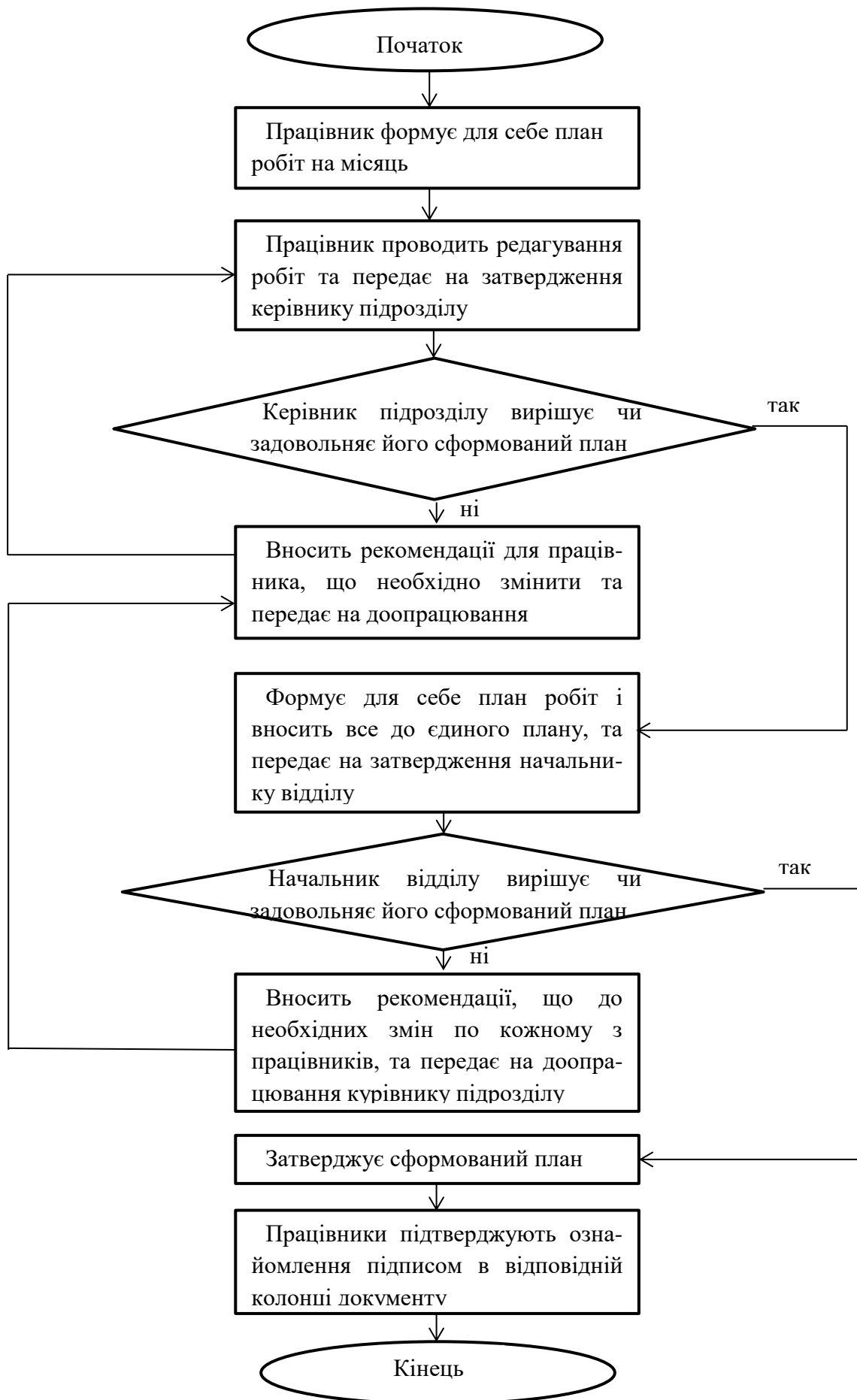


Рисунок 1.1 – Блок-схема формування плану робіт на місяць

Як бачимо процес формування плану робіт передбачає собою «блукання» відповідного документу, між різними працівниками, з внесенням до нього кожним з них змін. Як наслідок не складно здогадатись, зважаючи про те, що кожен з працівників чимось зайнятий, і можливо не відразу зможе ввести необхідні зміни, що даний процес може розтягуватися в часі на декілька днів. Проте формування плану робіт це лише квіточки, ягідки починаються коли настає час формувати звіт про виконану роботу за місяць.

Звіт про виконану роботу за місяць також обов'язково формується кожним відділом підприємства. Період формування звіту припадає на перший тиждень місяця, після звітнього. Звичайно ж він формується на основі плану робіт проте містись свої особливості.

До існуючих характеристик терміну виконання та показника зайнятості роботою, додається відмітка про виконання, яка може набувати значень: виконано, виконується і не виконано. Та й це ще не все, адже на протязі місяця працівник міг не виконати планову роботу через те, що виникла якась позапланова робота, яку необхідно врахувати. А це може тягнути за собою, також, зміну показника зайнятості роботами. Також значення реального терміну виконання робіт, можуть відрізнитися від планових, що також створює необхідність відобразити це в звіті.

Позапланові роботи заносяться до іншої таблички звіту, відмінної від тієї в якій занесені планові роботи. Табличка позапланових робіт не містить терміну виконання та показника, проте зазначається відмінка про виконання як і для планових робіт.

Побудуємо алгоритм формування звіту про виконані роботи:

1. Працівник на основі плану робіт та враховуючі реальні роботи що він виконав, формує перелік планових та окремо перелік позапланових робіт.
2. Вносить зміни до планових і позапланових робіт та відповідних їм характеристик, і передає звіт керівнику підрозділу.
3. Керівник підрозділу переглядає запропонований звіт та вирішує чи задовольняє він його, якщо так то пункт 5, якщо ні – пункт 4.

4. Вносить рекомендації для працівника, що необхідно змінити, додати чи видалити, чи то самі роботи чи відповідні їм характеристики, та передає на доопрацювання – пункт 2.

5. Формує для себе звіт та вносить до єдиного звіту про виконані роботи по всіх працівниках підрозділу, та передає на затвердження начальнику відділу.

6. Начальник відділу переглядає поданий звіт про виконання робіт працівниками підрозділу та вирішує чи задовольняє він його, якщо так то пункт 8, якщо ні – пункт 7.

7. Вносить рекомендації, що до необхідних змін по кожному з працівників, та передає на доопрацювання курівнику підрозділу – пункт 4.

8. Затверджує сформований завіт та передає для підтвердження кожним працівником відповідних планових та позапланових робіт.

9. Працівники підтверджують роботи та правильність їх характеристик підписом в відповідній колонці документу.

Блок-схема відповідного процесу звичайно дуже схожа з процесом формування плану робіт, проте через те, що з'являється більше характеристик робіт та позапланові роботи, цей процес має ще більше ітерацій до свого кінцевого результату (рис. Б.3).

З огляду на все перераховане приймалося рішення, стосовно того як можна автоматизувати цей процес. Найпростішим варіантом вирішення проблеми могло стати використання сервісу Google Docs.

Google Docs – це безкоштовний сервіс від компанії Google. Це web-орієнтоване програмне рішення, тобто таке, що функціонує без встановлення його на машину користувача. Дане рішення призначене для створення документів та таблиць на спеціальному сервері Google, чи можуть бути експортовані в файл. Головною особливістю даного сервісу є те, що редагування може відбуватися з будь-якого пристрою підключеного до мережі інтернет, при цьому право доступу підтверджується паролем.

В нашому випадку нам необхідний текстовий редактор, який дозволяє редагувати текстові документи та електронні таблиці. При редагуванні збереження змін відбувається відразу, але кожна зміна фіксується і зберігається, що дає можливість за необхідності відмінити внесені зміни. Для реалізації цих можливостей надається безкоштовно 15 Гб серверної пам'яті, а коли цього стане недостатньо, необхідно буде купити додаткове місце за 2,49 доларів США за 100 Гб на місяць користування[15].

Головною перешкодою для впровадження цього варіанту, стала саме його доступність для редагування кожним з працівників, що зайшов під своїм акаунтом, що не виключає банальну можливість злому доступу до даних та їх редагування. Вагомим мінусом, також можна вважати відсутність можливості налаштування різних прав доступу, для різних ролей користувачів.

### 1.3 Визначення задач та обмежень автоматизованої системи

З огляду на розглянуті алгоритми формування документів, стає очевидно, що рішення повинне бути в форматі web-додатку. Адже саме такого типу додаток, дозволить без зайвих проблем, зі встановленням, отримати певним користувачам доступ до створення, редагування чи видалення необхідної інформації, з миттєвим збереженням змін та їх доступністю для інших користувачів.

Так як була відкинута можливість використання сервісу Google Docs, виникла необхідність створення власного рішення, яке буде задовільняти всі покладені на нього завдання.

Для користування робочим комп'ютером працівник підприємства здійснює вхід до внутрішньої мережі підприємства, шляхом авторизації до серверної служби каталогів Active Directory. При вході в систему під власним обліковим записом кожен працівник може виконувати, відповідно до своєї

ролі на підприємстві, певний спектр задач та отримує доступ до чіткого переліку сервісів, якими може користуватись зі своїми правами. Саме даний тип авторизації необхідно застосувати для нашого додатку.

Користування системою авторизації Active Directory (AD) в додатку дозволить виділити працівників, які повинні мати доступ до нього, а вже з них виділити певні ролі доступу. Додаток має підтримувати такі ролі користувачів:

- Адміністратор – дана роль користувача надає йому глобальні права по редагуванню, додаванню та видаленню всіх робіт по всіх абсолютно робітниках. Цю роль матимуть лише два користувача – начальник відділу та його заступник.

- Керівник – дану роль будуть мати керівники підрозділів, їхні права на редагування, додавання та видалення обмежуватимуться роботами працівників лише їхнього підрозділу, при цьому доступу навіть для перегляду робіт інших підрозділів, вони не матимуть. Саме керівники підрозділів мають можливість формувати кінцевий вигляд документу та роздруковувати його.

- Працівник – дану роль мають всі рядові працівники підрозділів, їхні права дають можливість редагувати, додавати та видаляти лише свої роботи, при цьому переглядати роботи навіть працівників їхнього підрозділу права не мають.

Саме використання AD дозволить без зайвих проблем виділити ці ролі та попередити сторонній доступ до зміни даних. Ще одним плюсом є те, що в базі AD міститься інформація про наявні поштові скриньки працівників, це дозволить реалізувати сповіщення.

В процесі створення та редагування робіт певним користувачем може виникнути необхідність сповістити керівника підрозділу про внесені зміни і навпаки при зміні зі сторони керівника виникає необхідність сповістити конкретного робітника про проведені маніпуляції. Для цього може слугувати надсилання відповідному користувачу на email листа з повідомлення про

зміни, звичайно це можна зробити шляхом написання та відправлення його з власної пошти, проте все це буде забирати дорогоцінний час.

Тому очевидно, що однією з задач, які стоять перед додатком є реалізація можливості email-сповіщення. При цьому важливим є те, щоб дана операція здійснювалась не автоматично, при будь-якій зміні, додаванні чи видаленні, адже це спричинить масовість таких повідомлень на пошті, і буде сприяти їх ігноруванню з боку користувачів, а лише у випадку коли користувач, який ввід певну інформацію, буде вважати за необхідне сповістити про це. Тому в додатку після додавання або зміни інформації повинна з'являтися кнопка, яка дозволить надіслати стандартизоване повідомлення на пошту працівнику, якого стосуються ці зміни або керівнику підрозділу.

При роботі з додатком повинна бути реалізована можливість перегляду списку задач. Так як база даних буде містити назви робіт з обов'язковою їх прив'язкою до певного працівника та місяця, в якому реалізується дана робота, тому очевидно, що для відображення списку задач необхідно буде вказати два параметра:

- місяць певного року, відповідно з бази будуть повернуті роботи, які відбувалися на протязі заданого місяця;
- працівники роботи яких необхідно відобразити.

Стосовно першого параметра все зрозуміло, а що стосується другого тут дають про себе згадати ролі користувачів додатку. Якщо роль «працівник», то в поле буде передано прізвище працівника і воно буде не активне для змін, тобто він зможе переглянути роботи лише ті, що стосуються його. Якщо роль «керівник», то для відображення йому будуть доступні лише працівники свого підрозділу, він зможе вибрати відображення робіт одного, декількох чи всіх працівників підрозділу. Якщо ж роль «адміністратор», то можливості такі як і у «керівника», за винятком того, що йому доступні до відображення всі працівники відділу.

При відображенні робіт користувачу будуть доступні можливості виділення роботи та її редагування. При натисненні кнопки видалити відразу повинне відбуватися видалення і відображення відповідних змін без перезавантаження сторінки. Редагування обраної роботи повинно відбуватися на окремій сторінці, по завершення редагування дані повинні бути змінені в базі, а користувачу надані можливості здійснити email оповіщень та відображення зміненої та інші роботи відповідного працівника за місяць.

Також повинна бути реалізована можливість створення нової роботи, звичайно, з урахуванням можливостей кожної з ролей, та додавання позапланових робіт. Особливості відповідних процесів розглянуто в пункті 2.1 наступного розділу.

Найважливішою задачею, заради якої власне і створюватиметься даний додаток, це створення планів робіт на місяць (рис. Б.1) та звітів про виконані роботи за місяць (рис. Б.2). Кожен із зазначених документів в кінцевому вигляді формується керівником свого підрозділу, або заступником начальника відділу, який зазвичай і виконує відповідні задачі, за дорученням начальника відділу.

## 2 ПРОЕКТУВАННЯ WEB-ДОДАТКУ ДЛЯ АВТОМАТИЗАЦІЇ ФОРМУВАННЯ ПЛАНУ РОБІТ

### 2.1 Обмеження та вимоги при реалізації web-додатку

Обмеження, що можуть накладатися на певний web-продукт найчастіше стосуються технічної, програмної та функціональної складових, яким він має відповідати під час розробки та функціонування готової версії продукту. Відповідні обмеження формуються з огляду на якому сервері буде розміщуватись додаток, які засоби для роботи з ним буде використовувати користувач та, які функції має виконувати, з позиції замовника.

В першу чергу необхідно визначитися з технічними обмеженнями, яким має відповідати додаток. Відповідні обмеження накладає сервер, на якому буде розміщуватись додаток та серверна програма, яка реалізує сполучення між браузером користувача та сервером. На підприємстві ПАТ «Сумиобленерго» для внутрішньої мережі використовуються windows-сервери, з встановленою серверною програмою Apache HTTP-сервер. Було надано інформацію, що на відповідному сервері встановлені модулі, які підтримують:

- версію PHP 5.6;
- версію MySQL 5.5.

Інформація про решту модулів, що встановлені на сервері нас не цікавить.

Відповідно до цього, були встановлені обмеження: для використання серверної мови програмування – PHP версій 5.6, використання системи управління базами даних – MySQL версії 5.5.

Як відомо користування web-продуктами здійснюється за допомогою браузерів. При розробці web-додатку необхідно знати, якими браузерами та їх версіями користуються цільові користувачі, задля того, щоб кожен з них міг в необхідному вигляді переглядати контент та користуватися



функціоналом сайту. Для визначення відповідної інформації, було проведено усне опитування частини працівників відділу інформаційних технологій ПАТ «Сумиобленерго». Метою опитування було визначити:

- перелік браузерів, якими користуються більшість користувачів;
- чи оновлюють вони браузери, якщо ні, то якими їх версіями користуються;
- чи відмикають вони виконання скриптів JavaScript;

Останній пункт містив суто попереджувальний характер, адже очевидно, що використання скриптів JavaScript буде невід’ємною частиною додатку, і при їх відключенні функціонування рішення буде неможливим. Тому добрим знаком було те, що жоден з опитаних цього не робить. Що стосується перших двох пунктів, то результати їх дослідження, створюють вимоги до кросбраузерності додатку, відповідно до яких додаток має однаково відображатись та функціонувати в таких браузерах:

- Google Chrome версії 50.0+;
- Opera 14.0+;
- Firefox 50.0+;
- Safari 10.0+.

Отримані результати створюють обмеження на використання де-яких тегів мови розмітки HTML та атрибутів деяких тегів, які були у певних версіях відповідних браузерів доданими або зміненими. Також формуються обмеження на використання деяких властивостей та їх значень формальної мови стилів CSS, можливі також обмеження стосовно використання певних форматів рисунків та інше.

Відповідно до завдань, які покладені на додаток та побажань замовника не потрібно реалізувати:

- адаптивність рішення для різних пристроїв (планшетів, телефонів), адже робота з ним буде здійснюватися суто на робочих комп’ютерах працівників;

– багатомовність додатку, передбачена реалізація лише на українській мові.

Функціональні обмеження, яким має відповідати рішення, здебільшого, витікають з завдань, які воно має виконувати. Деякі з цих обмежень було розглянуто в попередньому розділі, в повній мірі або поверхнево. Саме для визначення всіх обмежень при реалізації кожного з завдань, з замовником було проведено чітке обговорення кожного з них, з визначенням можливих обмежень, які при цьому виникають. Розглянемо обмеження, які були додатково визначені.

В процесі додавання та редагування робіт мають заповнюватись такі поля:

- найменування роботи (заходу);
- відповідальний працівник;
- дата початку роботи;
- дата завершення роботи;
- показник виконання роботи у відсотках;
- статус роботи.

При цьому одразу виникає обмеження, щодо обов'язковості заповнення всіх полів, можливим виключення може бути лише поле роботи.

При заповненні поля «відповідальний працівник», виникає таке ж обмеження як і при відображенні робіт, тому необхідно передбачити різні варіанти вводу для різних ролей користувачів. Також важливими є перевірки при вводі дат початку та кінця, адже дати повинні бути задані в певному місяці року, і дата закінчення не повинна бути раніше дати початку. При вводі статусу роботи, має бути передбачено лише вибір із запропонованих варіантів: не виконано (за замовчуванням), виконується, виконано.

Головна складність виникає при перевірці заповнення показнику виконання робіт. Найважливішим обмеженням при цьому є те, що сума показників робіт за місяць, для певного працівника, не має перевищувати

100%, тому потрібно буде реалізувати таку перевірку при додаванні та редагуванні робіт.

Окремої уваги заслуговує додавання чи редагування позапланових робіт, для здійснення цих завдань повинні використовуватись ті самі форми, що і для звичайних робіт, а інформація про них повинна заноситися до однієї загальної таблиці в базі даних. Необхідно передбачити обмеження, яке дозволить реалізувати цю можливість.

Після того, як роботи по кожному працівнику занесені до бази та узгоджені з начальником відділу, настає етап формування кінцевого файлу у вигляді word-документу в форматі \*.docx. При формуванні відповідного документу потрібно дотримуватися чіткої стандартизації, адже він має стандартизовану шапку та назву, проте в них зазначаються унікальний місяць та рік для відповідного документу, тому необхідно реалізувати механізм їх автоматичного заповнення. В табличну частину з бази даних переносяться всі роботи по працівниках підрозділу з відповідними характеристиками, за винятком робіт, що помічені як позапланові. В кінці документу зазначається начальник відділу для якого створено даний план.

Стосовно формування звіту про виконані роботи за місяць, то його формування аналогічне плану робіт, винятком слугує формування окремої таблиці з зазначеними в ній позаплановими роботами. Відповідні роботи заносяться до таблиці з зазначенням лише двох характеристик: відповідального робітника та статусу роботи, решта характеристик до уваги не беруться.

На основі визначених завдань та окреслених обмежень необхідно сформувати web-додаток, при цьому всьому його дизайн повинен відповідати фірмовій кольоровій гамі, для ПАТ «Сумиобленерго» це відтінки синього кольору. Функціонал сайту повинен бути інтуїтивно зрозумілим для користувача, мати просту структуру та максимально швидко виконувати покладені на нього задачі.

## 2.2 Основні етапи розробки додатку

Процес розробки та тестування web-продукту передбачає виконання таких ключових етапів:

- розробка дизайну сайту;
- верстка сайту;
- створення бази даних;
- програмування на стороні клієнта;
- програмування на стороні сервера.

Розробка дизайну сайту передбачає створення в певному графічному редакторі зовнішнього вигляду web-сторінок. Кожна сторінка створюється з дотриманням необхідних розмірів елементів, місць їх розташування на сайті та кольорової гами. Під час створення дизайну необхідно створити всі рисунки, визначити стилі тексту, елементи розмітки та їх форми на кожній зі сторінок майбутнього сайту. У нашому випадку для створення дизайну було використано Adobe Photoshop, адже саме цей графічний редактор є лідером ринку в області засобів для редагування растрових зображень, має зручний інтерфейс та широкий функціонал.

Верстка сайту передбачає створення в точній відповідності дизайну html-розмітки сайту. Це включає розміщення на web-сторінці елементів мовою розмітки html, задання їх стильового оформлення та його зміна при здійсненні певних маніпуляцій користувачем (наведення курсору на елемент, натискання та інше) мовою стилів CSS та програмування анімації мовою Javascript. Для здійснення цього цих завдань використовувався редактор JetBrains PhpStorm[42].

PhpStorm це – інтелектуальний редактор для PHP, HTML і JavaScript з можливостями аналізу коду на льоту, запобігання помилок у сирцевому коді і автоматизованими засобами рефакторинга для PHP і JavaScript. Автодоповнення коду в PhpStorm підтримує специфікацію PHP

5.3/5.4/5.5/5.6/7.0/7.1 , включаючи співпрограми, генератори, простори імен, замикання, типажі і синтаксис коротких масивів. Присутній повноцінний SQL-редактор, який дає також можливість редагування отриманих результатів запитів.

Розробка бази даних не є обов'язковим етапом для створення всіх сайтів. Бази даних для сайтів (БД) використовуються для зберігання певної інформації і, спрощено, представляють собою деякий набір взаємозалежних таблиць. Розміри таблиць в БД можуть бути різними, їх кількість довільна. Саме в БД зберігається вся необхідна інформація для функціонування сайту, в нашому випадку це буде інформація про працівників та їх роботи.

Для створення та керування базою даних була обрана система управління реляційними базами даних MySQL. У реляційній базі даних дані зберігаються в окремих таблицях, завдяки чому досягається вигреш у швидкості й гнучкості. Таблиці зв'язуються між собою за допомогою відносин, завдяки чому забезпечується можливість поєднувати при виконанні запиту дані з декількох таблиць. SQL як частина системи MySQL можна охарактеризувати як мову структурованих запитів, що використовується для доступу до баз даних. Для управління MySQL будемо використовувати web-додаток для її адміністрування phpMyAdmin.

phpMyAdmin – web-додаток з відкритим кодом на мові PHP із графічним інтерфейсом для адміністрування бази даних MySQL. phpMyAdmin дозволяє через браузер здійснювати управління сервером MySQL, генерувати та виконувати запити SQL, створювати, переглядати та редагувати вміст таблиць баз даних. Додаток дозволяє керувати базою даних MySQL без вводу SQL команд з будь-якого комп'ютера під'єданого до інтернету без необхідності встановлення додаткового програмного забезпечення.

Етап програмування на стороні клієнта передбачає створення JS-скриптів, які будуть виконуватися браузером. Головною особливістю даних

скриптів є те, що відображення результатів їх виконання відбувається без перезавантаження.

Етап програмування на стороні сервера передбачає створення коду виконання якого буде інтерпретуватись web -сервером у html-код. Серверне програмування може здійснюватися будь-якою серверною мовою, у нашому випадку це буде реалізовано за допомогою мови PHP.

PHP це – скриптова мова програмування, яка створена для генерації HTML-сторінок на стороні web-сервера. PHP являється однією з найпоширеніших мов у сфері веб-розробок. Цю мову підтримують більшість хостинг-провайдерів. Код інтерпретується сервером у формат html -коду, який виконується в браузері клієнта. На відміну від JavaScript, користувачу не доступний PHP-коду, адже браузер отримує вже готовий html-код. Це надає перевагу з точки зору безпеки, але негативно відображається на інтерактивності сторінок. Але це вирішується можливістю використовувати PHP для генерації JavaScript-кодів, які виконуються вже на стороні клієнта.

Етапи кодування функціоналу сайту на стороні клієнта та сервера будуть також здійснюватися в PhpStorm. Ці два процеси є досить взаємозалежними, так в PHP можна генерувати JavaScript-коди, а в JS за допомогою технології AJAX з'являється можливість виконувати PHP скрипти без перезавантаження сторінки.

AJAX – це не самостійна технологія, а швидше концепція використання декількох суміжних технологій. AJAX-підхід до розробки, який призначений для користувачів інтерфейсів, реалізує комбінацію кількох основних методів та прийомів:

- використання DHTML для динамічної зміни змісту сторінки;
- використання XMLHttpRequest для звернення до сервера «на льоту», не перезавантажуючи всю сторінку повністю;

- альтернативний метод – динамічне підвантаження коду JavaScript в тег `<SCRIPT>` з використанням DOM, що здійснюється із використанням формату JSON;
- динамічне створення дочірніх фреймів.

Використання цих підходів надає можливість створювати більш зручніші web-інтерфейси користувача в тих випадках, коли необхідна активна взаємодія з користувачем. AJAX – асинхронний, тобто в процесі виконання відповідного запиту до серверу користувач може вільно переглядати далі контент сайту. Браузер не здійснює перезавантаження веб-сторінки і дані посилаються на сервер та отримуються результати без візуального підтвердження (крім випадків, коли розробник сам захоче продемонструвати відповідний процес взаємодії з сервером). Використання AJAX набуло популярності після того, як компанія Google почала активно використовувати його для створенні функціоналу сайтів, таких як Gmail, Google Maps і Google Suggest. Приклад вдалого їх функціонування підтвердив ефективність використання цього підходу.

Процес створення AJAX запиту на чистому JS передбачає написання багатьох стандартних команд так як і у випадках виконання деяких стандартизованих дій. Для пришвидшення процесу програмування було створено багато бібліотек для мови JS, які містять набори функцій, які дозволяють реалізувати необхідні дії викликавши певну функцію та задавши їй необхідні параметри. Однією з таких бібліотек є jQuery. Синтаксис jQuery розроблений, щоб зробити орієнтування у навігації зручнішим завдяки вибору елементів DOM, створенню анімації, обробки подій, і розробки AJAX-застосунків. Це сприяє створенню потужних і динамічних веб-сторінок. Для того, щоб використовувати можливості бібліотеки, достатньо у html-файлі вказати шлях до її копії на локальній машині або на одному з публічних серверів і за допомогою стандартизованого звернення, яким є знак долара, користуватися існуючими функціями.

В процесі функціонування додатку невід'ємним елементом його функціонування є взаємодія з базою даних засобами мови PHP. Так як досить частою буде робота з базою для додавання нових користувачів та робіт, здійснення редагування робіт, авторизації користувачів та постійного підвантаження окремих даних з бази, досить ефективним заходом буде використання open-source бібліотеки RedBeanPHP.

RedBeanPHP це – ORM бібліотека, яка дозволяє задавати зв'язок між об'єктами та СУБД, в якій кожен об'єкт запису називається біном. Біни можна розуміти як звичайні об'єкти, властивостями яких є записи в таблицях, завдяки чому керувати ними можна як звичайними масивами, що значно спрощує процес взаємодії з БД на відміну від прямих SQL-запитів[49].

Головним результатом дії додатку є документи плану робіт та звіту про виконані роботи. Відповідні документи повинні бути сформовані за допомогою методів мови PHP на основі інформації в базі даних та зберігатися у вигляді word-документу в форматі \*.docx. Для реалізації цієї можливості було використано PHPWord. PHPWord це – бібліотека, написана на чистому PHP, яка являє собою набір класів для зчитування та запису різних форматів файлів документів. Нинішня версія PHPWord підтримує Microsoft Office Open XML (OOXML або OpenXML), OASIS Open Document Format для додатків Office (OpenDocument або ODF) та Rich Text Format (RTF).

Процес розробки та тестування додатку не можливий без його запуску на сервері, адже вся обробка мовою PHP здійснюється саме на ньому, для цього використовувався Open Server.

Open Server Panel – це портативна серверна платформа і програмне середовище, створена безпосередньо для web-розробників з урахуванням їх рекомендацій і побажань. Програмний комплекс має багатий набір серверного програмного забезпечення, зручний, багатофункціональний продуманий інтерфейс, має потужні можливості з адміністрування та налаштування



компонентів. Платформа широко використовується з метою розробки, налагодження і тестування web-проектів, а так само для надання web-сервісів в локальних мережах[60].

Можливості керуючої програми:

- непомітна робота в треї Windows;
- швидкі старт і зупинка;
- автостарт сервера при запуску програми;
- монтування віртуального диска;
- підтримка управління через командний рядок;
- підтримка профілів налаштувань;
- зручний перегляд логів всіх компонентів;
- перемикання HTTP, MySQL і PHP модулів;
- доступ до доменів в один клік;
- швидкий доступ до шаблонів конфігурації.

Особливості комплексу:

- не вимагає установки (портативність);
- можливість роботи з USB накопичувача;
- одночасна робота з Denwer, Xampp і т.д .;
- робота на локальному / мережевому / зовнішньому IP адресу;
- підтримка SSL без всякої додаткової. настройки;
- створення домену шляхом створення звичайної папки;
- підтримка кирилических доменів;
- захист сервера від зовнішнього доступу;
- Punycod конвертер доменних імен;
- пакет з понад 40 портативних програм;
- створення локального піддомена без втрати видимості основного домену в мережі інтернет.

Досить важливою вимогою якій повинен відповідати додаток є його подальше підключення до серверної служби каталогів Active Directory. Тому

важливо визначити, що вона собою являє та яким особливостям має відповідати додаток. Для початку визначимося з поняттям домен, це основна адміністративна одиниця в мережевій інфраструктурі підприємства, до якої належать всі мережеві об'єкти, такі як користувачі, комп'ютери, загальні ресурси та тому подібне. Набір таких доменів називається лісом.

Служби Active Directory (активного каталогу) являють собою розподілену БД, яка містить всі об'єкти домену. У відповідному лісі доменів AD є єдиною точкою автентифікації та авторизації користувачів і програм в масштабах підприємства.

База даних AD зберігається на виділених серверах - контролерах домену. Служби AD є роллю серверних операційних систем Microsoft Windows Server, її служби мають широкі можливості масштабування. У лісі AD може бути створено понад 2-х мільярдів об'єктів, що дозволяє впроваджувати службу каталогів в компаніях з сотнями тисяч комп'ютерів і користувачів.

Ієрархічна структура доменів дозволяє гнучко масштабувати інфраструктуру на всі філії та регіональні підрозділи ПАТ «Сумиобленерго». Для кожної філії та дирекцій компанії створений окремий домен, зі своїми політиками, своїми користувачами і групами. Для кожного дочірнього домену делеговані адміністративні повноваження місцевим системним адміністраторам. При цьому все одно дочірні домени підкоряються батьківським.

Крім того, служби Active Directory дозволяють налаштувати довірчі відносини між доменними лісами. Кожна компанія має власний ліс доменів, кожен з яких має власні ресурси. Але іноді буває потрібно надати доступ до своїх корпоративних ресурсів співробітникам іншої компанії - робота з загальними документами і додатками в рамках спільного проекту. Для цього між лісами організацій можна налаштувати довірчі відносини, що дозволить співробітникам однієї організації авторизуватися в домені інший.

Для забезпечення відмовостійкості служб Active Directory розгортаються два або більше контролера домену в кожному домені. Між контролерами домену забезпечується автоматична реплікація всіх змін. У разі виходу з ладу одного з контролерів домену працездатність мережі не порушується, адже залишилися продовжують працювати.

### 2.3 Управління інформаційними потоками

Для того, щоб автоматизувати процес формування плану робіт та звіту про їх виконання, необхідно побудувати систему, яка дозволить керувати потоками інформації, які надходять від кожного з користувачів. Дані які вводить рядовий працівник повинні максимально швидко бути представленими для завірки керівнику відповідного підрозділу. Після перегляду наданої інформації перед керівником повинна з'явитися можливість її верифікувати, тобто завірити істинність переданої інформації, або ввести необхідні зміни, про які варто повідомити відповідного працівника. Лише після того як кожна із робіт була верифікована працівником та керівником підрозділу відповідні роботи можуть бути внесеними до плану чи звіту.

Начальник відділу повинен мати можливість в будь-який момент часу переглянути відповідні дані та внести необхідні на його погляд корективи, які після цього повинні знову верифікуватися працівником та керівником. Лише для начальника відділу повинна бути доступна можливість виставляти галочку про підтвердження верифікації, за будь-якого з користувачів, це необхідно лише для тих випадків коли користувач сам не зможе цього зробити через певні причини.

Для реалізації відповідного процесу необхідно передбачити механізм, який дозволить зберігати дані у необхідному форматі. Саме для цього при

розробці web-додатку необхідно створити базу даних у яку будуть зберігатися відповідні потоки інформації, а за запитом повертатимуться необхідному користувачу для їх перегляду чи зміни. База даних розміщується на сервері, тому процесі розробки web-додатку невід'ємною складовою є реалізація руху інформаційних потоків між користувачем та сервером. При цьому дані, що надходять від користувача, повинні зберігатися на сервері в чітко структурованій формі, щоб в будь який момент часу, по запиті, їх можна було отримати чи змінити. Для того, щоб здійснювати управління базою та даними в ній, необхідно користуватися системою управління базами даних (СУБД)[68].

Перш за все необхідно продумати, які дані нам знадобиться зберігати та як ми зможемо їх структурувати в БД, передбачити необхідний перелік таблиць та полів у кожній з них, а також які зв'язки будуть між таблицями. В нашому випадку нам необхідно зберігати інформацію про роботи та працівників. Варто зауважити, що використання Active Directory, передбачає доступ до внутрішньої бази даних, в якій буде міститися вся необхідна інформація про працівників. Тому в кінцевому результаті в нашій базі, необхідно буде зберігати лише інформацію про роботи з ідентифікатором певного працівника в базі AD, проте при розробці прототипу потрібно створити спрощений аналог бази AD, з тестовою інформацією про умовних працівників.

Зважаючи на це, потрібно створити дві основні таблиці: робітники та роботи, зв'язок між ними буде один до багатьох, відповідно один працівник може виконувати багато робіт. Також необхідно створити дві допоміжні таблиці: ролі та підрозділи, в яких буде зберігатися інформація про ролі, які можуть бути надані користувачам та назви підрозділів у яких можуть працювати працівники відповідно. Кожна з цих таблиць буде пов'язана з таблицею робітники, зв'язком один до багатьох.

Допоміжні таблиці матимуть по два поля, одне з них буде індексне, інше міститиме назву ролі в одній таблиці, а в іншій назву підрозділу.

Таблиці робітників та робіт матимуть поля, що представлені у таблицях 2.1 та 2.2.

Таблиця 2.1 – Поля таблиці робітники (worker) та їх характеристики

№	Назва	Тип даних	Індекс	Null	Додатково	Опис
1	ID	int(3)	ПК	ні	автозаповнення	індексне поле
2	ful_name	varchar(25)	–	ні	–	прізвище та ініціали працівника
3	email	varchar(30)	–	ні	–	електронна пошта
4	password	varchar(30)	–	ні	–	пароль
5	service_ID	int(2)	ІДД	ні	–	ідентифікатор підрозділу
6	role_ID	int(1)	ІДД	ні	–	ідентифікатор ролі

Таблиця 2.2 – Поля таблиці роботи (work) та їх характеристики.

№	Назва	Тип даних	Індекс	Null	Додатково	Опис
1	ID	int(7)	ПК	ні	автозаповнення	індексне поле
2	name_work	varchar(250)	–	ні	–	назва роботи
3	state_work	varchar(12)	–	ні	–	статус виконання роботи
4	persent	int(3)	–	ні	натуральне число	відсоток роботи
5	worker_ID	int(3)	ІДД	ні	–	ідентифікатор працівника
6	date_start	date	–	ні	–	дата початку виконання роботи
7	date_finish	date	–	ні	–	дата завершення виконання роботи
8	verif_ker	boolean	–	ні	за замовчуванням(0)	верифікація керівником
9	verif_user	boolean	–	ні	за замовчуванням(0)	верифікація працівником

Заповнення таблиці роботи в повній мірі буде реалізуватись через додаток, відповідно всі необхідні обмеження будуть перевірятися в процесі введення даних користувачем. Вивід даних для перегляду користувачем у web-додатку та для формування текстового документу, будуть здійснюватися засобами серверної мови PHP з відповідними запитам до БД. Що стосовно таблиці робітників, то необхідно реалізувати можливість додавання та

видалення робітників в процесі функціонування додатку, для запуску тестової версії (без підключення Active Directory). Необхідно виконати всі необхідні заходи, що дозволять максимально підвищити зручність для подальшої реалізації підключення AD.

В результаті дотримання всіх зазначених вимог до структури та наповнення БД отримано таку логічну модель бази даних (рис. 2.1).

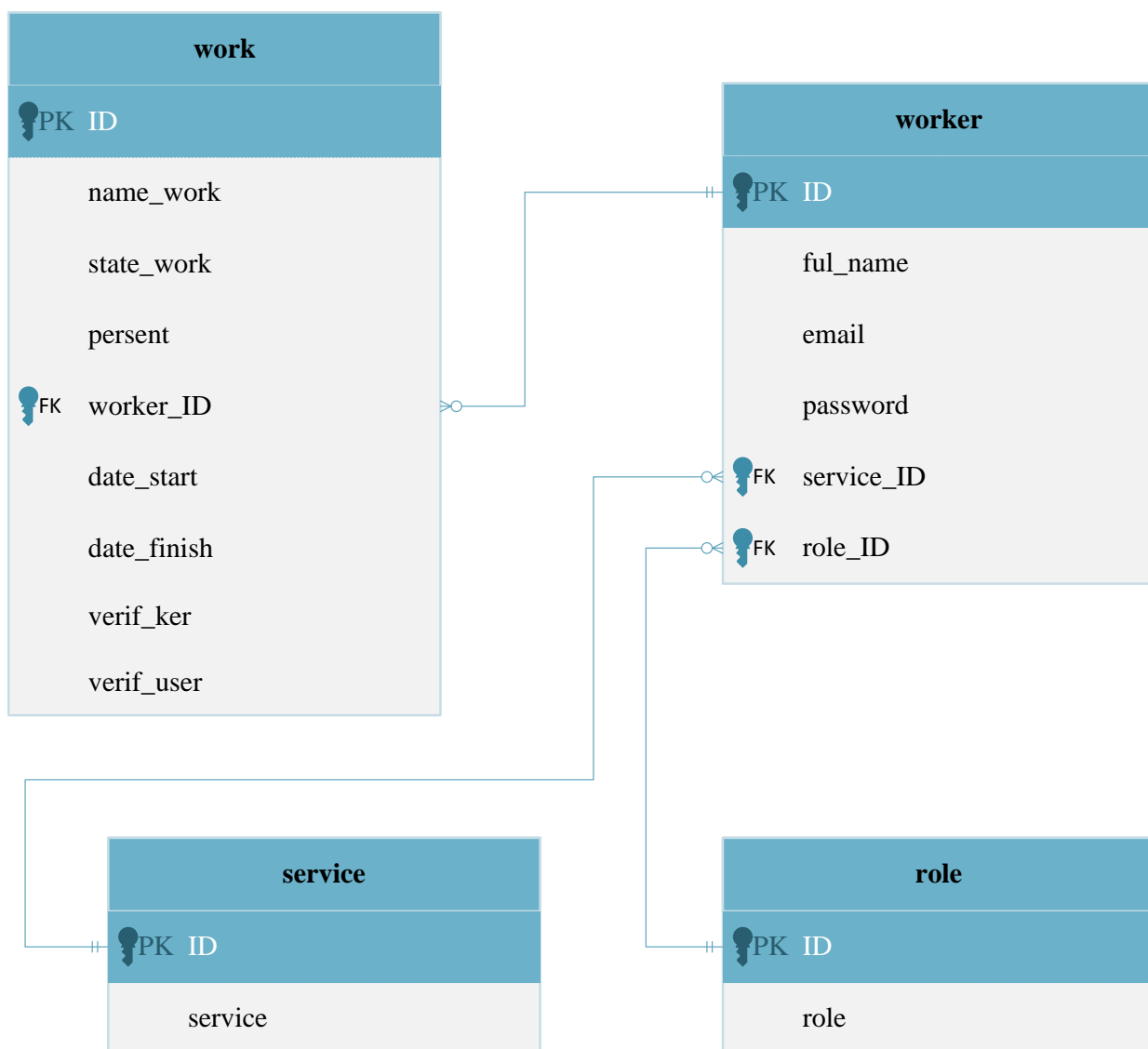


Рисунок 2.1–Логічна модель бази даних

Відповідно до побудованої логічної моделі бази даних було реалізовано її створення, за допомогою web-додатку для адміністрування phpMyAdmin.

## 3 РЕАЛІЗАЦІЯ ПРОТОТИПУ АВТОМАТИЗОВАНОГО РІШЕННЯ ФОРМУВАННЯ ПЛАНУ РОБІТ НА ПІДПРИЄМСТВІ

### 3.1 Розробка архітектури автоматизованої системи

Функціонування web-додатку передбачає дві складових обробки: клієнтської та серверної. Клієнтська обробка включає в собі представлення зовнішнього вигляду сторінок мовою розмітки HTML, формальною мовою стилів CSS та реалізація поведінки елементів без запиту до серверу, за допомогою мови програмування JavaScript (анімація, перевірка вводу даних та інше). Серверна обробка включає в себе реалізацію всієї логіки, що здійснюється на сервері, це може бути здійснення взаємодії з базою даних, з файлами на сервері, управління сесіями, куками та інше. Серверна обробка може здійснюватися багатьма мовами програмування, такими як Java, Perl, Python, Ruby та PHP, саме остання була обрана для реалізації додатку.

PHP інтерпретується web-сервером у html -код, який реалізується на стороні клієнта. На відміну від мови JavaScript, користувач не бачить коду PHP, бо браузер приймає вже готовий html-код.

При розробці дизайну сайту важливо визначити, яке враження має скласти додаток на користувача чи вразити його контрастністю та анімацією, чи бути мінімалістичним, але максимально функціональним. Завданням яке стоїть перед розробкою даного додатку є досягнення структурної простоти сайту з яскраво вираженими та зрозумілими елементами управління. Додаток має максимально швидко загрузатись та здійснювати свої головні функціональні завдання.

Розробка web-додатку в першу чергу передбачає формування моделі сайту з функціональними блоками (рис.3.1). Кожна сторінка буде відповідати стандартизованій структурі і включатиме такі блоки:

- хедер сайту;
- меню сайту;

- контент сайту;
- футер сайту.

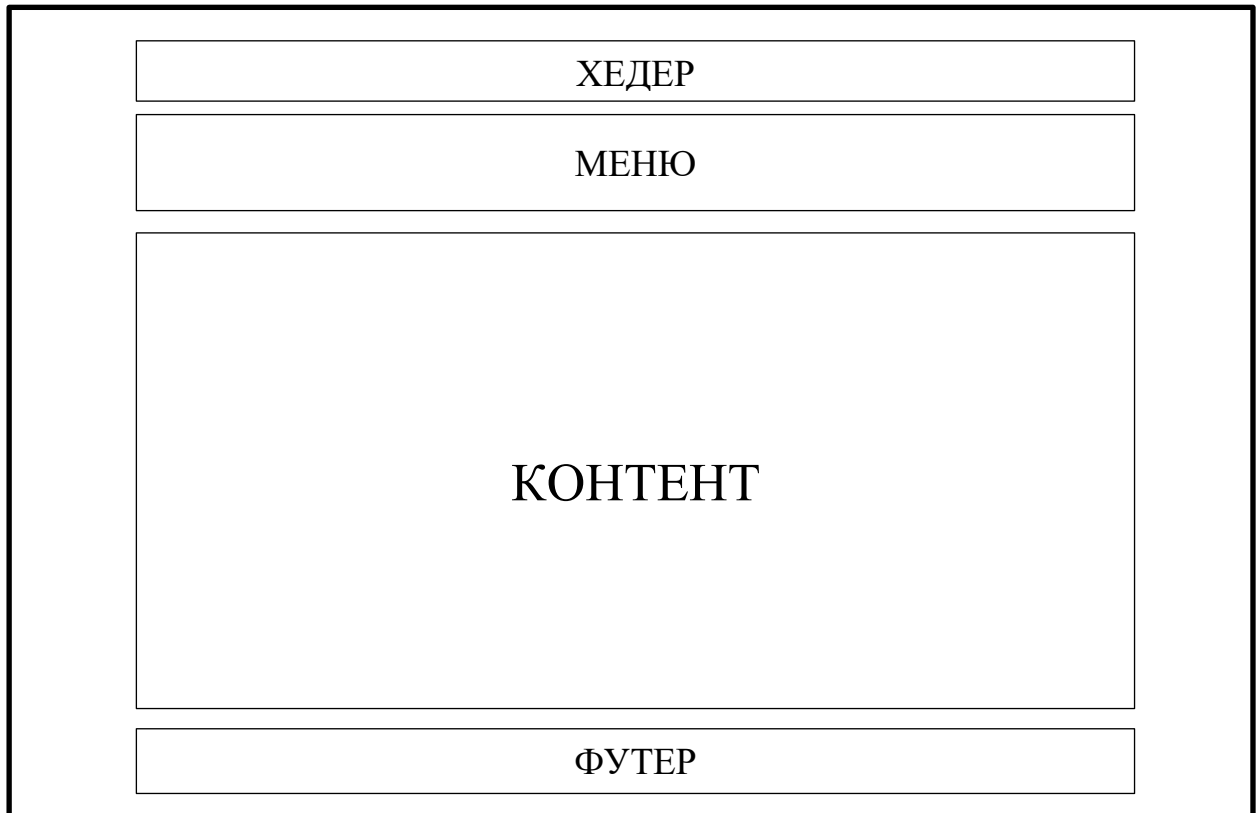


Рисунок 3.1–Функціональна модель сайту

В хедері сайту міститься назва підприємства, його логотип, логін авторизованого користувача та посилання на сторінку для виходу з акаунту, відповідне посилання з'являється лише після того як користувача було авторизовано (рис. 3.2).

Меню сайту містить посилання на основні сторінки сайту з відзначенням активної сторінки (рис. 3.3).

Футер сайту містить відомості про автора додатку (рис. 3.4).

Контент є єдиним змінним блоком сайту, в залежності від обраної сторінки наповнення цього блоку може бути різним, основне управління сайтом реалізовано саме в даному блоці. На відміну від інших блоків, його наповнення, залежно від сторінок, здебільшого реалізується засобами PHP, а



вже потім інтерпретуються в html-код. Проте важливо саме на цьому етапі створити CSS стилі для відповідних елементів.



Рисунок 3.2–Хедер сайту

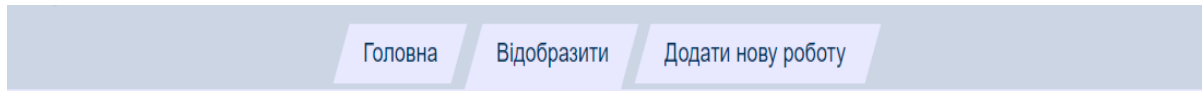


Рисунок 3.3–Меню сайту

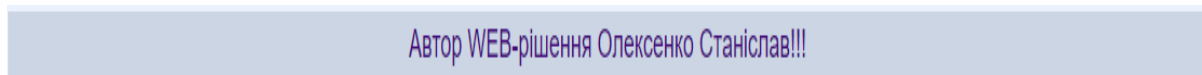


Рисунок 3.4–Футер сайту

Зважаючи на завдання які покладені на додаток очевидно, що він має містити такі сторінки:

- головна;
- сторінка вибору параметрів відображення та формування результуючих документів;
- сторінка відображення робіт;
- сторінка для додавання нових робіт;
- сторінка для зміни існуючих робіт;
- сторінка для додавання нових працівників;
- сторінка для видалення працівників.

Після того як web-сторінка завантажена з сервера перед користувачем з'являється можливість не лише переглядати її, а й передбаченим чином на неї впливати. Відповідний вплив може бути реалізований вводом даних у передбачені поля сторінки, натисканням на кнопки, наведенням курсору на певні елементи і тому подібне. В результаті кожної з цих дій сторінка має певним чином реагувати. Ця реакція може бути реалізована засобами CSS та

виконанням скриптів JavaScript на стороні клієнта, тобто без перезавантаження сторінки.

Важливим завданням при розробці додатку є максимально зменшити завантаженість серверу та використання трафіку, що позитивно відзначиться на швидкості роботи додатку. Особливої актуальності це набуває при заповненні полів форм, так на дані, введені до них, накладається багато обмежень і бажано більшість з них перевірити на клієнтській стороні до запиту на сервер.

Яскравим прикладом реалізації цього механізму є сторінки додавання (рис. 3.5) та зміни робіт. Так як поля на цих сторінках однакові, відповідно і обмеження будуть майже однаковими, тому доцільно розглянути узагальнену роботу скриптів JavaScript (JS). Назви полів на цих сторінках та обмеження до їх вводу представлені в таблиці 3.1. Варто зазначити, що завантажені сторінки мають не активну кнопку відправки форми на сервер, її активація відбувається лише після дотримання всіх обмежень.

The screenshot shows a web interface for 'Сумиобленерго' with a user profile 'Олексенко С.В. вийти'. The navigation menu includes 'Головна', 'Відобразити', and 'Додати нову роботу'. The main form for adding a job contains the following fields and messages:

Найменування роботи(заходу)	Назва роботи	
Прізвище відповідального працівника	Не вірний ввід	Оберіть варіант зі списку
Дата початку роботи	17.11.2018	Робота задається на певний місяць року
Дата завершення роботи	16.12.2018	Робота задається на певний місяць року
Показник,%	1	Заповніть поля працівника та дати!!!
Статус роботи	Не виконано	

Buttons: Додати роботу, Стерти

Footer: Активация Windows. Чтобы активировать Windows, перейдите в раздел "Параметры".

Рисунок 3.5—Сторінка додавання роботи з помилками заповнення

Після активації кнопки, додавання записів до бази даних виконується за допомогою AJAX запиту, за тим засобами JS поля очищаються та створюються дві кнопки, одна з них перенаправляє на сторінку з відображенням робіт відповідного працівника за відповідний місяць, а інша надає можливість надіслати повідомлення на пошту про зміни відповідному працівнику, яка реалізується також за допомогою асинхронного AJAX запиту (рис. 3.6). В процесі реалізації кожного запиту на сервер для користувача з'являються відповідні повідомлення про перебіг тієї чи іншої операції. Якщо користувач знову починає заповнювати поля створені кнопки зникають і сторінка стає знову готова для додавання вже нової роботи.

Таблиця 3.1–Поля сторінок додавання та зміни робіт, та їх обмеження на стороні клієнта

Назва поля	Тип поля	Обмеження
Назва роботи	textarea	Максимальна довжина назви роботи 250 символів, поле не є обов'язковим до заповнення.
Відповідальний працівник	input[type=text]	Під час заповнення поля запускається JS скрипт, який за допомогою AJAX запиту до бази даних, підтягує перелік працівників прізвище, яких починається зі введених символів і вже з цього переліку користувач обирає працівника. Якщо користувач ввів дані, які не відповідають жодному працівнику, формується напис помилки у підготовленому елементі сторінки.
Дата початку	input[type=date]	Після заповнення обох полів і подальшого їх редагування, здійснюється перевірка на те чи дата завершення не є меншою дати початку та чи зазначені дати належать одному місяцю та року, в випадку не виконання формується повідомлення про помилку.
Дата завершення	input[type=date]	
Показник	input [type=number]	Лише заповнення даного поля активує кнопку відправки форми. Перш за все перевіряється чи вірно заповнені решта полів, якщо так то за допомогою AJAX запиту до бази даних, визначається в яких межах може вводиться значення показника, з огляду на те, що в кожному місяці сума показників всіх робіт робітника не може перевищувати 100%.
Статус роботи	select	Дане поле обмежене вибором одного з трьох варіантів статусу робіт.

Сумиобленерго Зімак В.О. вийти

Головна | Відобразити | Додати нову роботу

Найменування роботи(заходу)

Прізвище відповідального працівника

Дата початку роботи

Дата завершення роботи

Показник,%

Статус роботи

Додати роботу | Стерти

Дані додані в базу!!!

Відобразити роботи Козак А.О. за 12-й місяць 2018 року | Відправити повідомлення для Козак А.О. про зміни

Рисунок 3.6–Сторінка додавання роботи після запису даних в базу

Також прикладом активного використання клієнтської обробки є сторінка для вибору робіт для відображення та формування плану та звіту робіт (рис. 3.7). На відповідній сторінці є два або три поля залежно від ролі користувача:

- прізвище відповідального працівника;
- вибір підрозділу (лише для адміна);
- місяць та рік виконання робіт.

Сумиобленерго Олександренко С.В. вийти

Головна | Відобразити | Додати нову роботу

Оберіть необхідні параметри відображення

Вкажіть прізвище відповідального працівника

Оберіть підрозділ

Оберіть рік та місяць

Сформувати план | Відобразити | Сформувати звіт

Автор WEB-рішення Олександренко Станіслав!!!

Рисунок 3.7–Сторінка вибору параметрів відображення та формування документів

В першому полі зазначаються працівники роботи яких потрібно відобразити, це може бути один, декілька чи всі працівники. При виборі кожного з працівників формується окремий блок, який в разі бажання можна стерти, саме на основі цих блоків формується відображення. Варто зазначити, що заповнення даного поля не впливає на формування документів.

Поле вибору підрозділу сформоване в формі селектора, відповідно його значення можна обрати лише з запропонованих. Поле вибору місяця та року методами JS заповнюється по замовчуванню нинішнім значенням місяця та року, проте за бажанням може бути обраний інший місяць.

В результаті натискання кнопок сформувати план чи сформувати звіт, здійснюється AJAX запит, який формує відповідний документ, в результаті вдалого виконання з'являється кнопка скачати, після натискання на яку реалізується скачування файлу та стирання відповідної кнопки, що запобігає багаторазовому скачуванню файла.

### 3.2 Реалізація серверної логіки web-додатку

Реалізація майже всіх процесів функціонування додатку здійснюється на сервері з винятком JS скриптів, проте у випадку виконання запитів AJAX, хоча вони і здійснюються мовою JS, основна логіка відбувається на сервері, в файлі php на який посилаються відповідні запити та від якого отримуються результати.

Одним із ключових завдань, яке реалізується на сервері є авторизація користувачів. Відповідна дія здійснюється на сторінці головна, перед неавторизованим користувачем відразу з'являється форма авторизації з полями логін, який представляє собою прізвище та ініціали користувача в базі та пароль(рис. 3.8).

Рисунок 3.8–Форма авторизації користувача

Після заповнення форми та відправки значень полів, реалізується запит до бази даних з перевіркою наявності відповідного логіну, а потім перевіряється шляхом дешифрування пароль відповідного користувача. Після проходження авторизації до супер-глобальної змінної `$_SESSION` заносяться: прізвище, код ролі та підрозділу користувача. Відповідні дані будуть перевірятися при намаганні зайти на будь-яку сторінку додатку, якщо вони відсутні, тобто користувач не авторизований, його буде пере направлено на сторінку авторизації. У випадку наявності цих даних визначається роль користувача і відповідно вигляд сторінки.

На головній сторінці лише для ролі адміна з’являються доступні посилання на сторінки додавання та видалення користувачів (рис. 3.9).

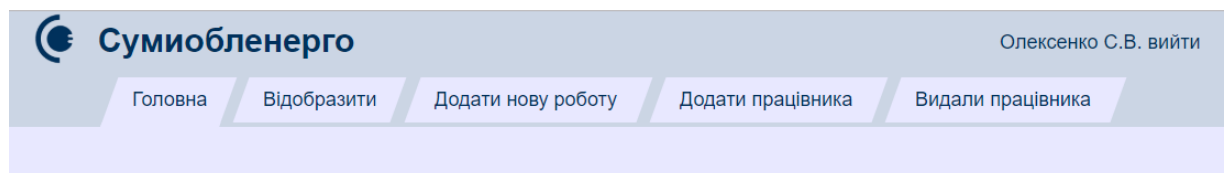


Рисунок 3.9–Вигляд головної сторінки для користувача з роллю «адмін»

На сторінці вибору параметрів відображення та формування документів для ролі «працівник» не буде доступним для редагування поле вибору працівника, яке по замовчуванню буде мати значення його прізвища, а кнопки формування плану та звіту для нього відобразатись не будуть. Після натискання кнопки «Відобразити», відповідно заданим параметрам будуть повернені з бази даних роботи та їх характеристики у вигляді таблиць (рис.3.10).

Сумиобленерго		Зімак В.О. вийти					
		Головна	Відобразити	Додати нову роботу			
	Найменування роботи(заходу)	Дата початку виконання	Дата завершення виконання	Відповідальний	Показник,%	Відмітка про виконання	Узгодження
	Супровід модулю PSM-FM системи SAP ERP.	2018-11-02	2018-11-22	Зімак В.О.	80	Виконується	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
	Переведення баз даних ПК «Фінансист» на окремий сервер для Управління та філій	2018-11-08	2018-11-23	Зімак В.О.	20	Виконано	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
	(000)Їсти кашу	2018-11-01	2018-11-09	Зімак В.О.	0	Виконано	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
	(000)Їсти м'ясо	2018-11-08	2018-11-29	Зімак В.О.	0	Не виконано	<input checked="" type="checkbox"/> <input type="checkbox"/>
	Впровадження SAP IS-U. Модуль BW. Відладка екст-ракторів та звітів.	2018-11-01	2018-11-30	Зин А.Р.	30	Виконано	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
	Супровід модулю PSM-FM SAP ERP.	2018-11-09	2018-11-09	Олексенко С.В.	30	Виконано	<input type="checkbox"/> <input checked="" type="checkbox"/>
	Дослідження	2018-11-08	2018-11-30	Олексенко С.В.	30	Виконано	<input type="checkbox"/> <input checked="" type="checkbox"/>
	Супровід ПЗ «Фінансист», «Зарплата», «Канцелярія», «Ліміти»	2018-11-02	2018-11-22	Олексенко С.В.	10	Не виконано	<input type="checkbox"/> <input checked="" type="checkbox"/>

Рисунок 3.10–Сторінка «Відобразити»

Відображення робіт на сторінці здійснюється відносно обраних параметрів, тобто для обраних працівників визначеного відділу за зазначений місяць. Особливістю відображень залежно від ролей наступні для ролі «працівник» будуть відображені лише його роботи, для ролі «керівник» можуть бути відображені лише роботи працівників його підрозділу, для ролі «адмін» можуть бути відображені роботи всіх працівників, але з огляду на обраний підрозділ при виборі параметрів відображення.

На сторінці відображення у останніх двох колонках знаходяться елементи управління «input» з типом «radio», які відображають значення стану верифікації роботи у форматі так або ні (див. рис. 3.10). Елемент у першій колонці може змінити лише користувач, який є працівником, що виконує цю роботу, тобто верифікувати її, а у другій колонці лише керівник підрозділу. Дані про відповідний стан кожної роботи міститься у базі даних. Важливо зазначити, що без верифікації всіх робіт за певний місяць план та

звіт сформовано не буде. Право зміни цих елементів також є у адміна, на екстрений випадок.

Проте це не єдині елементи управління на цій сторінці, у першій колонці реалізовані дві кнопки у вигляді рисунків: хрестика та олівчика (див. рис. 3.10). При наведенні на них з'являється підказка, яка сповіщає користувача, що хрестик – призначений для видалення робіт з бази даних, а олівчик – перенаправляє користувача на сторінку зміни обраної роботи. У випадку натискання на хрестик реалізується AJAX запит на сервер і робота видаляється з бази даних, про це користувача інформує спливаюче віконце, та зникнення роботи з відображення. У випадку натискання на олівчик, інформація про обраний елемент для зміни, передається за допомогою метода POST на сторінку зміни роботи. Варто зазначити, що після збереження будь-яких змін до бази, значення показників верифікації ставиться в позицію «ні», тобто робота вимагає повторної верифікації.

Можливість відправки повідомлення про зміни, доступні на сторінках змінити та додати роботи. Після того як здійснилася зміна або додавання роботи з'являється кнопка, яка дозволяє відправити повідомлення за допомогою AJAX запиту на пошту. У випадку коли користувач авторизований як працівник, йому пропонується відправити повідомлення керівнику свого підрозділу, якщо користувач керівник або адмін то повідомлення може бути відправлене працівнику, якого стосуються відповідні зміни.

Формування результуючих документів також буде відбуватися за допомогою РНР. Перш за все перевіряється роль користувача, якщо роль «керівник», то здійснюється запит до бази даних і повертаються значення: прізвища керівника підрозділу, назви та перелік робіт працівників підрозділу, якщо роль «адмін», то повертаються відповідні значення для обраного підрозділу. Після цього перевіряється стан верифікації кожної з робіт, якщо якісь з робіт не були підтвержені відповідними працівниками, то з'являється перелік таких працівників та можливість їх проінформувати за допомогою



відправки повідомлення на пошту, а якщо справа лише за керівником підрозділу, то здійснюється відображення робіт, які необхідно верифікувати. При цьому процес формування документу переривається.

У випадку коли перший етап успішно пройдено перевіряється наявність мітки позапланової роботи –«(000)». Для формування плану робіт роботи з такими мітками відкидаються, а з решти формується наповнення документа (див. рис. Б.1) та зберігається на сервер у відповідну папку. Для звіту роботи без міток заносяться до таблиці планових робіт, а ті що з міткою до таблиці позапланових відповідний документ також зберігається на сервер (див. рис. Б.2). Сформовані плани та звіти на сервері зберігаються до того моменту як буде сформовано новий документ відповідно, це реалізовано для того щоб, кожного разу за запитом формувалася новий документ з актуальною інформацією.

### 3.3 Оцінка ефективності автоматизованої системи

Звичайно реалізована автоматизована система (АС) створення плану робіт та звіту про виконані роботи дозволить спростити виконання рутинної роботи по формуванню та наповненню цих документів, що дозволить вивільнити певну частину робочого часу працівників та використати цей час на виконання першочергових завдань. Поряд з тим було витрачено час на реалізацію відповідного проекту підрахунок вартості якого, дозволить визначити вартість системи, а також процес впровадження системи вимагає додаткових затрат від підприємства. Саме розрахунок відповідності вартості суми витрат та суми економії дозволить визначити, чи буде мати позитивний економічний ефект впроваджена система для підприємства.

Для розрахунку показника ефективності в першу чергу необхідно визначити капітальні витрати за наступною формулою:

$$C = K_{\text{соб.}} + K_{\text{впров.}} \quad (3.1)$$

де  $K_{\text{соб.}}$  – собівартість проекту;

$K_{\text{впров.}}$  – вартість впровадження проекту.

Показник економічного ефекту будемо визначати на прикладі впровадження АС на ПАТ «Сумиобленерго». Для початку необхідно визначити вартість години роботи працівника служби програмного супроводу дирекції інформаційних технологій, відповідно до тарифікації, вартість години робочого часу рядового працівника без жодного роду надбавок становить – 50 грн/год. Відповідний проект було реалізовано на протязі переддипломної практики протяжність якої була 4 тижні, передбачався восьмигодинний робочій графік п'ять днів на тиждень. Відповідно знаючи ці показники можемо розрахувати собівартість відповідного проекту:  $K_{\text{соб.}} = 8\text{год} * 20\text{днів} * 50\text{грн/год} = 8\ 000\ \text{грн}$ .

Після того як проект створено, необхідно розглянути механізм впровадження його у виробничий процес. Для впровадження нам не знадобиться здійснювати витрати на купівлю певних технічних засобів, адже проект від початку був «заточений» під впровадження його на існуючу технічну базу, а знову таки знадобиться витратити лише певний об'єм робочого часу. При впровадженні повинні бути здійснені наступні дії:

- а) завантаження розробленої бази даних на сервер (1 год);
- б) завантаження на сервер всього проекту та налаштування відповідних параметрів роботи як серверу так і додатку разом з базою даних(3 год);
- в) тестування роботи web-додатку та додавання до баз всіх необхідних користувачів (4 год);
- г) ознайомлення користувачів з інструкцією до користування додатком та перше знайомство з його функціоналом, відповідно 5 підрозділів по 30 працівників, на ознайомлення кожному з них потрібно 30 хв, тобто весь цей етап займе:  $5 * 30 * 30\text{хв} = 75\ \text{год}$  ;

д) після проходження тестового періоду, необхідно буде здійснити вирішення виникнувши проблем та впровадження авторизації за допомогою служби каталогів Active Directory (8 год).

Просумувавши витрати часу на кожному етапі та знаючи вартість години робочого часу можемо визначити вартість впровадження автоматизованої системи:  $K_{\text{впрров.}} = (1 + 3 + 4 + 75 + 8)\text{год} * 50\text{грн/год} = 4\ 550\ \text{грн.}$

Відповідно до формули (3.1) визначимо загальні капітальні витрати проекту:  $C = 8\ 000\ \text{грн} + 4\ 550\ \text{грн} = 12\ 550\ \text{грн.}$

На наступному етапі необхідно визначити суму річної економії за формулою:

$$S = B_{\text{б}} - B_{\text{п}}, \quad (3.2)$$

де  $B_{\text{б}}$  – витрати за базовий рік;

$B_{\text{п}}$  – прогнозовані витрати за рік.

Для того, щоб визначити витрати за базовий рік, необхідно визначити загальний час, який витрачається на формування відповідних документів за рік та помножити на вартість години робочого часу. План робіт та звіт про виконані роботи формується кожного місяця у кожному підрозділі, процес формування кожного з цих документів за звичайних умов складає в середньому 3 дні, зважаючи на те, що на протязі цього часу хтось обов'язково вносить до нього певні зміни, то можна вважати, що в сукупності всіма працівниками підрозділу для формування кожного з цих документів протягом одного місяцю було витрачено:  $3 * 8 = 24\ \text{год}$ , а на два документи відповідно – 48 год .

Так як відповідні документи формуються кожним з 5 підрозділів, по 12 разів на рік витрати на їх формування в базовому році будуть наступними:  $B_{\text{б}} = 5 * 12 * 48\ \text{год} * 50\ \text{грн/год} = 144\ 000\ \text{грн.}$

У випадку впровадження автоматизованої системи передбачається зменшення часу формування кожного документу, що найменше на третину. Відповідно в місяць кожним підрозділом для формування документів буде витрачено 32 год, тоді витрати за рік на їх формування після впровадження АС:  $V_{\Pi} = 5 * 12 * 32 \text{ год} * 50 \text{ грн/год} = 108\,000 \text{ грн}$ .

Отже відповідно до формули (3.2) сума річної економії буде дорівнювати:  $S = 144\,000 \text{ грн} - 108\,000 \text{ грн} = 36\,000 \text{ грн}$ .

Тепер перейдемо до визначення стандартних показників ефективності впровадження інформаційних систем: річного економічного ефекту (3.3), показника економічної ефективності капітальних витрат (3.4) та терміну окупності впровадженої автоматизованої системи (3.5).

$$E_y = S - C * r_n, \quad (3.3)$$

$$R_{ce} = \frac{S}{C}, \quad (3.4)$$

$$P_p = \frac{C}{S} = \frac{1}{R_{ce}}. \quad (3.5)$$

де  $E_y$  – річний економічний ефект;

$r_n$  – нормативний коефіцієнт окупності капітальних вкладень, узятий для конкретної галузі (для ІТ-сфери даний коефіцієнт знаходиться в межах від 0,3 до 0,35);

$R_{ce}$  – показник економічної ефективності капітальних витрат;

$P_p$  – терміну окупності.

Відповідно до формули (3.3) річний економічний ефект:  $E_y = 36\,000 - 12\,550 * 0,35 = 31\,607,5$  (грн); згідно формули (3.4) показник економічної ефективності капітальних витрат:  $R_{ce} = \frac{36\,000}{12\,550} \approx 2,87$ ; відповідно до формули (3.5) термін окупності:  $P_p = \frac{12\,550}{36\,000} \approx 0,35$  (роки). Тобто якщо вірити нашим розрахункам, витрати на впровадження розробленої автоматизованої системи окупляться вже після перших чотирьох місяців її експлуатації.

## ВИСНОВКИ

На нинішньому етапі розвитку технологій, все більшої необхідності набуває процес автоматизації робіт на підприємстві. Перед існуючими компаніями вже давно не існує вибору впроваджувати АС чи ні, адже це рівносильно запитанню: «Чи планує підприємство надалі здійснювати свою діяльність та бути конкурентоздатним?» Саме ті з них хто вчасно зрозумів цю просту істину, на разі перебувають на вершині успішності, проте досягти цього значно простіше чим зберегти. З кожним днем з'являються все новіші технології, для впровадження кращих з них має бути відкрите підприємство.

В роботі було досліджено діяльність компанії ПАТ «Сумиобленерго», яка є одним із лідерів серед енергетичних підприємств України. І це не випадково, адже було визначено, що товариство одне з перших на ринку енергетики впровадило систему SAP ERP, яка дала можливість автоматизувати такі ключові функції, як бухгалтерія та фінанси, персонал та адміністрація. Повністю впровадивши відповідну систему лише в кінці минулого року, вже було досягнуто колосальних результатів, і було обрано напрямом на подальшу автоматизацію рутинних процесів. Одним із них став процес формування плану робіт та звіту про виконані роботи.

В роботі було детально досліджено відповідний процес. З огляду на складності, які при цьому виникають було сформовано вимоги, яким мала б відповідати АС. Було розглянуто існуючі варіанти рішень, які могли б допомогти у вирішенні поставлених задач. Проте, по тих чи інших причинах ці варіанти були відкинуті і прийнято рішення створити власну систему для автоматизації відповідного процесу у вигляді web-додатку.

Було визначено чіткий перелік задач, які повинна виконувати система та обмеження, які вона має перевіряти. Після чого були сформовані обмеження та вимоги, яких необхідно було дотримуватися в процесі реалізації архітектури та функціоналу web-додатку.

Наступним завданням, яке необхідно нам було вирішити, стало формування етапів розробки відповідної системи та вибір технологічних засобів для реалізації кожного з них у відповідності до поставлених завдань. Також було передбачено механізми управління інформацією та її збереження.

Після проведення всіх підготовчих етапів, було розроблено архітектуру web-додатку та логіку його функціонування на стороні клієнта та сервера. В результаті було створено прототип системи, яка повністю відповідає поставленим до неї вимогам та виконує свою головну задачу – формує вірно структуровані та наповнені актуальною інформацією документи плану робіт та звіту про виконані роботи в форматі \*.docx.

В процесі розрахунку ефективності створеної автоматизованої системи було визначено, що період окупності рішення буде меншим за чотири місяці, а річний економічний ефект від її впровадження для підприємства буде перевищувати 30 тисяч гривень.

Звичайно прототип системи, після її тестування, ще вимагатиме доопрацювання, серед очевидного, це підключення Active Directory та реалізація механізму, який дозволить створити архів відповідних документів, що дасть можливість за необхідності не генерувати їх, а скачати відповідний файл, якщо він вже був створений раніше.

Відповідне рішення також може бути впроваджене і на інших підприємствах, де існує схожа проблема, з мінімальними змінами у структурі результуючих документів, що робить роботу актуальною не лише для ПАТ «Сумиобленерго», а й для інших компаній особливо малих та середніх через не високу вартість та простоту впровадження відповідної системи.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дерев'янку А.С. Технології та засоби консолідації інформації: навчальний посібник/ А.С. Дерев'янку, М.Н. Солощук. – Харків: НТУ "ХПІ", 2008. – 432 с.
2. Компас у світі сервіс-орієнтованої архітектури (SOA): цінність для бізнесу/ Біберштейна Н., Боуз С., Джонс К., Фіаммант М., Ша Р. – М.: КУДИЦ-ПРЕСС, 2008. – 256 с.
3. Статут Публічного акціонерного товариства «Сумиобленерго» [Електронний ресурс]: Постанова правління ПАТ «Сумиобленерго» від 28.09.2011 №3 – Режим доступу: [https://www.soe.com.ua/images/docs/akcioneram/ustanovchi\\_documentu/status\\_PAT\\_Symuoblenergo.pdf](https://www.soe.com.ua/images/docs/akcioneram/ustanovchi_documentu/status_PAT_Symuoblenergo.pdf).
4. Енергетика по-європейськи: які інновації доступні українським енергокомпаніям [Електронний ресурс]: /Світлана Ларченко // Mind. – Режим доступу: <https://mind.ua/openmind/20188145-energetika-po-evropejski-yaki-innovaciyi-dostupni-ukrayinskim-energo kompaniyam>.
5. Положення про Філію «Сумське міжрайонне відділення «Енергозбут» [Електронний ресурс]: Постанова правління ПАТ «Сумиобленерго» від 12.10.2011 №19 – Режим доступу: [https://www.soe.com.ua/images/docs/akcioneram/ustanovchi\\_documentu/status\\_PAT\\_Symuoblenergo.pdf](https://www.soe.com.ua/images/docs/akcioneram/ustanovchi_documentu/status_PAT_Symuoblenergo.pdf).
6. Журнал Енергетик № 3 2017 року [Електронний ресурс]. – Режим доступу: <https://www.soe.com.ua/images/energetik/03-2017.pdf>.
7. Software as a Service: Strategic Backgrounder: [Електронний ресурс]. – Режим доступу: <http://www.siiia.net/estore/ssb-01.pdf>.
8. UP2010 Virtual Cloud Conference – Microsoft's PaaS Solution: [Електронний ресурс]. – Режим доступу: <http://channel9.msdn.com/posts/UP2010-Virtual-Cloud-ConferenceMicrosofts-PaaS-Solution>.

9. ORACLE Infrastructure as a service (IaaS) with capacity on demand: [Електронний ресурс]. – Режим доступу: <http://www.oracle.com/us/products/engineered-systems/iaas/iaasexec-brief1908773.pdf>.
10. Skinner, C. Digital Bank: Strategies to Launch or Become a Digital Bank [Text] / C. Skinner. – Marshall Cavendish Business, 2014. – 315 p.
11. Архитектура клієнт-сервер / [Електронний ресурс] – Режим доступу: <http://portal.tpu.ru:7777/SHARED/f/FAS/study/avis/lectures/cli-se.pdf>
12. Автоматизовані інформаційні технології в економіці / Під ред. Г.А. Титоренко / М.: Комп'ютер, ЮНІТИ. – 2008. – 348 с.
13. Інформаційні системи і технології в економіці. Посібник / За ред. д.е.н. В. С. Пономаренка. – К.: Видавничий центр «Академія», 2002. – 542 с.
14. Береза А. М. Основи створення інформаційних систем: Навч. посібник. – К.: КНЕУ, 2001. – 214 с.
15. Сазонець О. М. Інформаційні системи і технології в управлінні зовнішньоекономічною діяльністю [Текст] : навч. посіб. / О. М. Сазонець. – К.: Центр учбової літератури, 2014. – 256 с.
16. Сучасні інформаційні системи і технології: конспект лекцій / В. Г. Іванов, С. М. Іванов, В. В. Карасюк та ін.; за заг. ред. В. Г. Іванова, В. В. Карасюка. – Х.: Нац. юрид. ун-т ім. Ярослава Мудрого, 2014. – 347 с.
17. Інформаційні системи і технології на підприємствах: Конспект лекцій (для студентів і слухачів ФПО та ЗН спеціальності "Економіка підприємства") – Укл. В.М. Охріменко, Т.Б. Воронкова. – Харків: ХНАМГ, 2006. – 185 с.
18. Азарова А.О. Інформатика та комп'ютерна техніка [Текст]: навч. посіб. / А.О. Азарова, А.В. Поплавський; Вінниц. нац. техн. унт. – Вінниця: ВНТУ, 2012. – Ч. 1. – 2012. – 360 с.
19. Горлач А.С. Інформаційноаналітичне забезпечення управління діяльністю підприємства / А.С. Горлач // Науковий вісник Академії муніципального управління. Серія: Економіка. – 2013. – Вип. 1. – С. 179–184.



20. Томашевський О.М. Інформаційні технології та моделювання бізнеспроцесів: навч. посіб. для студ. ВНЗ / О.М. Томашевський, Г.Г. Цегелик, М.Б. Вітер, В.І. Дубук. – К.: Центр уч. лри, 2012. – 295 с.
21. Almalech M. E. Advertisements: Signs of Femininity and their Corresponding Color Meanings / M. E. Almalech. – Sofia : Prof. Marin Drinov Academic Publishing House, 2011. – 210 p.
22. Kress G. Colour as a semiotic mode: Notes for a grammar of colour / G. Kress , Th. van Leeuwen // Visual communication. – Vol. 1 (3). – SAGE Publications, 2012. – P. 343–368.
23. Інформаційні системи в економіці [Текст]: навч. посіб. / В.С. Пономаренко та ін.; Харк. нац. екон. унт. – Х.: Вид. ХНЕУ, 2011. – 175 с.
24. Padda Harkirat. QUIM: A Model for Usability/Quality in use Measurement. LAP Lambert Academic Publishing, 2009, 120 p.
25. Sanjay Kumar Dubey et. al. Analytical Roadmap to Usability Definitions and Decompositions/ International Journal of Engineering Science and Technology, Vol. 2(9), 2010, P. 473–479.
26. Chaffey D. E-business and E-commerce Management / D. Chaffey// Strategy, Implementation and Practice. – Prentice Hall, 2009. – 800 p.
27. Онищенко І. І. Аналіз ризиків в процесі управління ІТ-проектами [Текст] / І. І. Онищенко // Вісник Національного технічного університету «ХПІ» : зб. наук. пр. Серія: Стратегічне управління, управління портфелями, програмами та проектами. – Х.: НТУ «ХПІ». – 2014. – № 3. – С. 15–17.
28. Муравецький С. А. Планування процесів забезпечення якості у великих та географічно розподілених гібридних ІТ-проектах [Текст] / С. А. Муравецький, С. О. Крамський // Вісник НТУ «ХПІ». Серія: Стратегічне управління, управління портфелями, програмами та проектами. – Харків : НТУ «ХПІ», 2016. – № 1. – С. 26–28.
29. Laravel [Електронний ресурс] / Режим доступу: <https://laracasts.com/skills/laravel>. – Заголовок з екрану.

30. How To Install MySQL on Ubuntu 16.04 [Электронный ресурс] / Режим доступа: <https://www.digitalocean.com/community/tutorials/how-to-installmysql-on-ubuntu-16-04>. – Заголовок з екрану.
31. MySQL [Электронный ресурс] / Режим доступа: <http://znaimo.com.ua/MySQL>. – Заголовок з екрану.
32. Bringing MySQL to the web [Электронный ресурс] / Режим доступа: <https://www.phpmyadmin.net/>. – Заголовок з екрану.
33. The Apache Software Foundation [Электронный ресурс] / Режим доступа: <https://blogs.apache.org/foundation/entry/apache-is-open>. – Заголовок з екрану.
34. How To Move an Apache Web Root to a New Location on Ubuntu 16.04 [Электронный ресурс] / Режим доступа: <https://www.digitalocean.com/community/tutorials/how-to-move-an-apache-web-root-to-a-new-location-on-ubuntu-16-04>. – Заголовок з екрану.
35. How To Install the Apache Web Server on Ubuntu 16.04 [Электронный ресурс] / Режим доступа: <https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-webserver-on-ubuntu-16-04>. – Заголовок з екрану.
36. Mapping URLs to Filesystem Locations [Электронный ресурс] / Режим доступа: <https://docs.phpmyadmin.net/en/latest/index.html>. – Заголовок з екрану.
37. PHP – Taking the world by storm [Электронный ресурс] / Режим доступа: <http://www.phpbbhq.com/developmentofphp.php>. – Заголовок з екрану.
38. PhpStorm Early Access Program [Электронный ресурс] / Режим доступа: <https://www.jetbrains.com/phpstorm/ear/>. – Заголовок з екрану.
39. Docker Support in PhpStorm [Электронный ресурс] / Режим доступа: <https://confluence.jetbrains.com/display/PhpStorm/Docker+Support+in+PhpStorm>. – Заголовок з екрану.

40. How to Install and Configure PHP 7.0 [Электронный ресурс] / Режим доступа: <https://www.vultr.com/docs/how-to-install-and-configure-php-70-or-php71-on-ubuntu-16-04>. – Заголовок з екрану.
41. JavaScript - Document Object Model or DOM [Электронный ресурс]/ Режим доступа:[https://www.tutorialspoint.com/javascript/javascript\\_html\\_dom.htm](https://www.tutorialspoint.com/javascript/javascript_html_dom.htm). – Заголовок з екрану.
42. Documentation JQuery [Электронный ресурс] / Режим доступа: <https://learn.jquery.com/>. – Заголовок з екрану.
43. The Progressive JavaScript Framework [Электронный ресурс] / Режим доступа: <https://vuejs.org/>. – Заголовок з екрану.
44. Установить Linux, Apache, MySQL, PHP [Электронный ресурс] / Режим доступа:<https://www.digitalocean.com/community/tutorials/linuxapachemysql-php-lamp-ubuntu-16-04-ru>– Заголовок з екрану.
45. LAMP [Электронный ресурс] / Режим доступа: <https://uk.wikipedia.org/wiki/LAMP>– Заголовок з екрану.
46. Official Ubuntu Documentation [Электронный ресурс] / Режим доступа: <https://help.ubuntu.com/>– Заголовок з екрану.
47. RussianDocumentation Ubuntu [Электронный ресурс] / Режим доступа: <https://help.ubuntu.com/community/RussianDocumentation/>– Заголовок з екрану.
48. Настройка веб сервера Apache через Htaccess [Электронный ресурс] / Режим доступа: <http://htaccess.net.ru/>– Заголовок з екрану.
49. Основы .htaccess на примерах [Электронный ресурс] / Режим доступа: <https://habrahabr.ru/post/31054/>– Заголовок з екрану.
50. Bootstrap[Электронныйресурс]/Режимдоступу: <https://getbootstrap.com/>– Заголовок з екрану.
51. HTML [Электронный ресурс] / Режим доступа: <http://devdocs.io/html/>– Заголовок з екрану.
52. Learn HTML [Электронный ресурс] / Режим доступа: <https://www.codecademy.com/learn/learn-html>– Заголовок з екрану.

53. CSS Reference [Електронний ресурс] / Режим доступу: <https://www.w3schools.com/cssref/default.asp> – Заголовок з екрану.
54. Система управління сайтами DIAFAN.CMS: офіційний сайт [Електронний ресурс] / DIAFAN.CMS – Режим доступу до ресурсу: <https://www.diafan.ru>
55. Трёхуровневая архитектура [Електронний ресурс] // Википедия. Свободная энциклопедия – Режим доступу до ресурсу: [https://ru.wikipedia.org/wiki/Трёхуровневая\\_архитектура](https://ru.wikipedia.org/wiki/Трёхуровневая_архитектура).
56. .NET Framework (.NET) [Електронний ресурс] // Techopedia – Режим доступу до ресурсу: <https://www.techopedia.com/definition/3734/netframework-net>.
57. 5 Benefits of a 3-Tier Architecture [Електронний ресурс] // Izenda – Режим доступу до ресурсу: <https://www.izenda.com/blog/5-benefits-3-tierarchitecture>.
58. CREATE PROCEDURE (Transact-SQL) [Електронний ресурс] // Microsoft – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/sql/tsql/statements/create-procedure-transact-sql>.
59. Freeman A. Pro ASP.NET MVC 3 Framework / A. Freeman, S. Sanderson. – New York: Apress, 2011. – 852 p.
60. FURPS – Wikipedia [Електронний ресурс] / Режим доступу до ресурсу: <http://en.wikipedia.org/wiki/FURPS>.
61. Global B2C E-commerce Report 2016 [Електронний ресурс] // The Ecommerce Foundation. – 2017. – Режим доступу до ресурсу: [https://www.ecommercewiki.org/wikis/www.ecommercewiki.org/images/5/56/Global\\_B2C\\_Ecommerce\\_Report\\_2016.pdf](https://www.ecommercewiki.org/wikis/www.ecommercewiki.org/images/5/56/Global_B2C_Ecommerce_Report_2016.pdf).
62. Image.CMS: офіційний сайт [Електронний ресурс] / Режим доступу до ресурсу: <https://www.imagecms.net>.
63. Introduction to IDEF0/3 for Business Process Modelling [Електронний ресурс]. – 2013. – Режим доступу до ресурсу:

<http://businessprocessagility.com/wp-content/uploads/2015/06/IDEF03-guidebook.pdf>.

64. PrestaShop: офіційний сайт [Електронний ресурс]/Режим доступу до ресурсу: <https://www.prestashop.com/en>.
65. Retail e-commerce sales worldwide from 2014 to 2021 [Електронний ресурс] // Statista – Режим доступу до ресурсу: <https://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/>.
66. Rosziati Ibrahim. A Formal Model for Data Flow Diagram Rules / Rosziati Ibrahim, Siow Yen Yen. // ARPN Journal of Systems and Software. – 2011. – №2. – С. 60–69.
67. Shop-Script: офіційний сайт [Електронний ресурс] / Режим доступу до ресурсу: <http://www.shop-script.ru>. 91
68. SQL CREATE TABLE Statement [Електронний ресурс] // w3schools.com – Режим доступу до ресурсу: [https://www.w3schools.com/sql/sql\\_create\\_table.asp](https://www.w3schools.com/sql/sql_create_table.asp).
69. SQL PRIMARY KEY Constraint [Електронний ресурс] // w3schools.com – Режим доступу до ресурсу: [https://www.w3schools.com/sql/sql\\_primarykey.asp](https://www.w3schools.com/sql/sql_primarykey.asp).
70. Three-Tier Architecture [Електронний ресурс] // Techopedia – Режим доступу до ресурсу: <https://www.techopedia.com/definition/24649/three-tierarchitecture>.
71. Yaser Ahangari Nanekaran. An Introduction To Electronic Commerce / Yaser Ahangari Nanekaran. // International journal of Scientific & Technology research. – 2013. – №2. – С. 190–192.

# ДОДАТКИ

Додаток А  
(обов'язковий)

SUMMARY

Oleksenko S.V. Automation of work scheduling services for enterprise software support. - Masters-level Qualification Thesis. Sumy State University, Sumy, 2018.

In the master's thesis the essence of the process of works scheduling is investigated, the analysis of existing information systems for the automation of the process of forming the corresponding documents is carried out. The basic requirements for the system being created are described. The description of the main business processes at the enterprise is implemented. The technology of creating an automated system was proved. Economic efficiency from the implementation of the developed system at the enterprise is calculated.

Keywords: automation, work scheduling, AJAX technology, web-application, World Wide Web, MySQL database, phpMyAdmin, PHPWord, PHP, efficiency, business process, word-document.

АНОТАЦІЯ

Олексенко С.В. Автоматизація планування робіт служби програмного супроводу підприємства. – Кваліфікаційна магістерська робота. Сумський державний університет, Суми, 2018 р.

У роботі досліджено сутність процесу планування робіт, проведений аналіз існуючих інформаційних системи для автоматизації процесу формування відповідних документ. Наведені основні вимоги до створюваної системи. Здійснений опис основних бізнес-процесів на підприємстві. Була обґрунтована технологія створення автоматизованої системи. Розрахована

економічна ефективність від впровадження розробленої системи на підприємстві.

Ключові слова: автоматизація, планування робіт, технологія AJAX, web-додаток, всевітня мережа Інтернет, СУБД MySQL, phpMyAdmin, PHPWord, PHP, ефективність, бізнес-процес, word-документ.



Додаток Б  
(обов'язковий)

"ПОГОДЖЕНО"

Заступник директора з ІТ  
ПАТ «Сумнобленерго»  
\_\_\_\_\_ О.П. Колесніченко  
« \_\_\_ » листопада 2018 р.

"ЗАТВЕРДЖУЮ"

Директор з ІТ  
ПАТ «Сумнобленерго»  
\_\_\_\_\_ В.П. Цуканов  
« \_\_\_ » листопада 2018 р.

**План робіт**  
Служби програмного супроводу  
на листопад 2018 р.

№ п/п	Найменування роботи(заходу)	Термін виконання	Відповідальний	Показник, %	Підпис відп.особи
1.	Супровід модулю PSM-FM системи SAP ERP.	02.11-22.11	Зімак В.О.	80	
2.	Переведення баз даних ПК «Фінансист» на окремий сервер для Управління та філій	08.11-23.11	Зімак В.О.	20	
3.	Впровадження SAP IS-U. Модуль BW. Відладка екстракторів та звітів.	протягом місяця	Зин А.Р.	30	
4.	Супровід модулю PSM-FM SAP ERP.	09.11	Олексенко С.В.	30	
5.	Дослідження	08.11-30.11	Олексенко С.В.	30	
6.	Супровід ПЗ «Фінансист», «Зарплата», «Канцелярія», «Ліміти»	02.11-22.11	Олексенко С.В.	10	
7.	SAP ERP. Реалізація заявки GLPI № 11031 «Формування файлу формату lcodev»	протягом місяця	Олексенко С.В.	10	
8.	Переведення баз даних ПК «Фінансист» на окремий сервер для Управління та філій	до 16.11	Олексенко С.В.	10	
9.	Робити диплом	03.11-23.11	Олексенко С.В.	10	

Начальник СПС

Активаци:  
Чтобы актив  
раздел "Пара

Зімак В.О.

Рисунок Б.1 – План робіт за листопад 2018 р. сформований створеною системою на основі тестових даних

"ПОГОДЖЕНО"

Заступник директора з ІТ  
ПАТ «Сумнобленерго»  
\_\_\_\_\_ О.П. Колесніченко  
« \_\_\_ » листопада 2018 р.

"ЗАТВЕРДЖУЮ"

Директор з ІТ  
ПАТ «Сумнобленерго»  
\_\_\_\_\_ В.П. Цуканов  
« \_\_\_ » листопада 2018 р.

**Звіт до плану робіт  
Служби програмного супроводу  
дирекції з інформаційних технологій  
ПАТ "Сумнобленерго" за листопад 2018 р.**

№ п/п	Найменування роботи(заходу)	Термін виконання	Відповідальний	Показник, %	Відмітка про виконання	Підпис від особи
1.	Супровід модулю PSM-FM системи SAP ERP.	02.11-22.11	Зімак В.О.	80	Виконується	
2.	Переведення баз даних ПК «Фінансист» на окремий сервер для Управління та філій	08.11-23.11	Зімак В.О.	20	Виконано	
3.	Впровадження SAP IS-U. Модуль BW. Відладка ехст-факторів та звітів.	протягом місяця	Зиц А.Р.	30	Виконано	
4.	Супровід модулю PSM-FM SAP ERP.	09.11	Олексенко С.В.	30	Виконано	
5.	Дослідження	08.11-30.11	Олексенко С.В.	30	Виконано	
6.	Супровід ПЗ «Фінансист», «Зарплата», «Канцелярія», «Ліміти»	02.11-22.11	Олексенко С.В.	10	Не виконано	
7.	SAP ERP. Реалізація заявки GLPI № 11031 «Формування файлу формату lcode»	протягом місяця	Олексенко С.В.	10	Виконано	
8.	Переведення баз даних ПК «Фінансист» на окремий сервер для Управління та філій	до 16.11	Олексенко С.В.	10	Виконується	
9.	Робити диплом	03.11-23.11	Олексенко С.В.	10	Виконується	

**Позапланові роботи**

№ п/п	Найменування роботи(заходу)	Відповідальний	Відмітка про виконання	Підпис від особи
1.	Їсти кашу	Зімак В.О.	Виконано	
2.	Їсти м'ясо	Зімак В.О.	Не виконано	
3.	Позапланова	Олексенко С.В.	Не виконано	
4.	Додробка	Олексенко С.В.	Не виконано	
5.	в'язати	Олексенко С.В.	Не виконано	

Начальник СПС

Зімак В.О.

Рисунок Б.2 – Звіт про виконані роботи за листопад 2018 року сформований розробленою системою на основі тестових даних

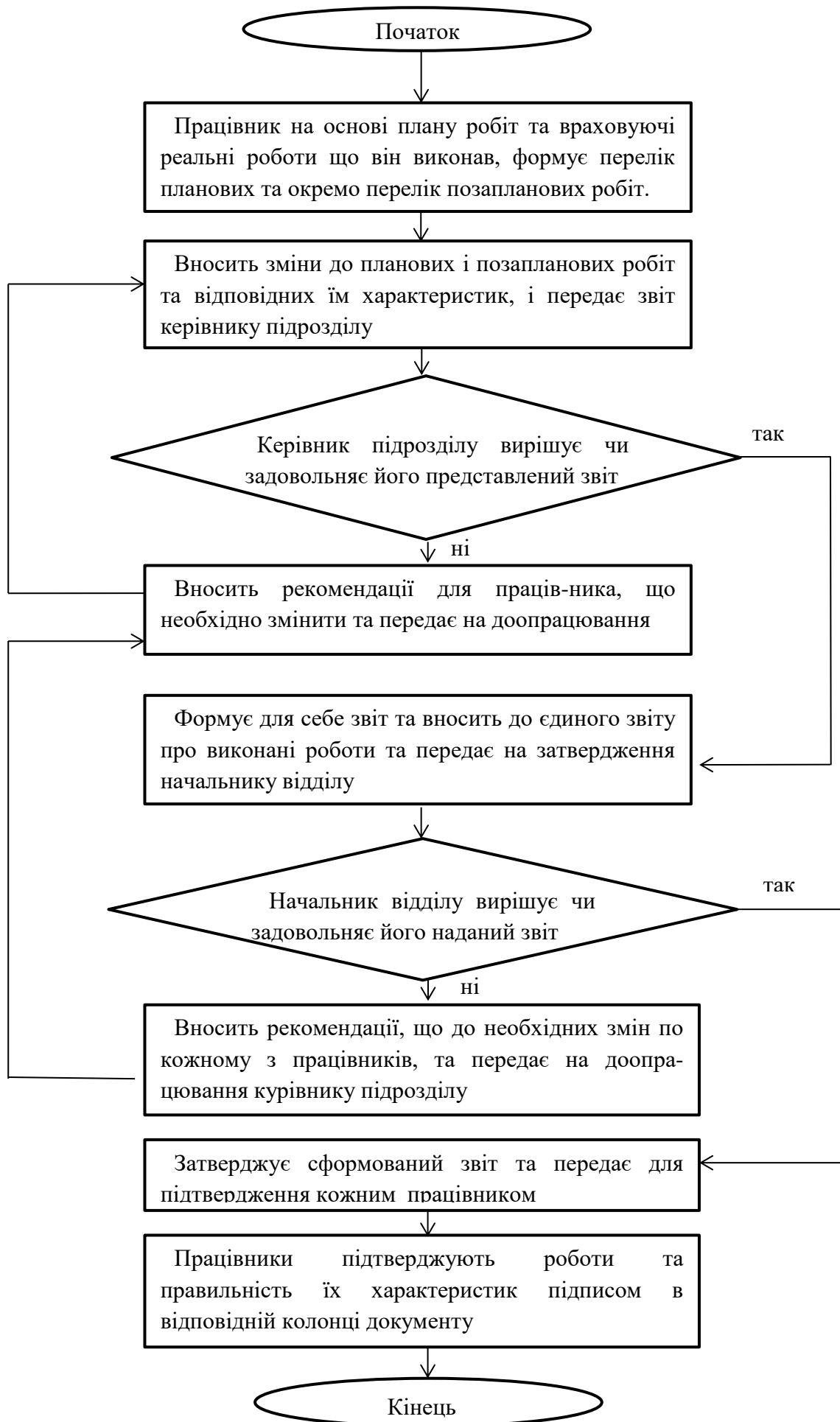


Рисунок Б.3 – Блок-схема формування звіту про виконані роботи за місяць

## Додаток В

### Файл practic.php

```

<?php session_start();
    require "php/bd_con.php";
    $inf='Для роботи з додатком необхідно авторизуватись';
    ?>

<html>
<head>
    <title>Відобразити</title>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
    <link rel="stylesheet" href="css/practic.css">

</head>
<body>
<div ID="content">
    <div id="headerMain">
        <header>
            <div class="log">
                
                <B class="rev">Сумиобленерго</B>
                <?if(isset($_SESSION['ful_name'])):?>
                <a class="toggle" href="php/logaut.php" ><?echo $_SESSION['ful_name'];?> вийти
                </a>
                <?endif;?>
            </div>
            <center>
                <ul class="manuul">
                    <li class='selected manuli'>
                        <a class="manua" href="practic.php" >Головна</a></li>
                    <li class="manuli"><a class="manua" href="vibor.php" >Відобразити</a></li>
                    <li class="manuli">
                        <a class="manua" href="add_work.php" >Додати нову роботу
                        <b class="caret"></b></a></li>
                    <?if($_SESSION['role_ID']==1):?>
                    <li class="manuli">
                        <a class="manua" href="php/add_worker.php" >Додати працівника
                        <b class="caret"></b></a></li>
                    <li class="manuli">
                        <a class="manua" href="php/drop_worker.php" >Видали працівника
                        <b class="caret"></b></a></li>
                    <?endif;?>
                </ul>
            </center>
        </header>
    </div>
    <div class="kol">
        <? if(isset($_SESSION['ful_name'])):?>
            <?
            echo '<h1 >Шановний '.$_SESSION['ful_name'].'!</H1>?>
            <div class='tex'><p >
            За допомогою даного рішення ти зможеш переглянути задачі</P>
            <p >
            для обраних працівників за певний місяць; додати нові задачі; </P>
            <p >
            змінити існуючі, сформувати план та звіт робіт за місяць!!! </P></div>
        <? else: ?>
            <?
            if(isset($_POST['exit']))
            {
                $errors=array();

```

```

if($_POST['login']==")
{
    $errors[]="Введіть логін користувача!";
}
if($_POST['password']==") {
    $errors[]="Введіть пароль!";
}
if(empty($errors)) {
    $qw="SELECT * FROM worker WHERE ful_name
    =".$_POST['login']."";

    $user=R::getAll($qw);

    if($user)
    {
        $user=$user[0];
        $pass=$_POST['password'];

        if(password_verify($pass,$user['password']))
        {

            $_SESSION['ful_name']=$user['ful_name'];
            $_SESSION['service_ID']=$user['service_ID'];
            $_SESSION['role_ID']=$user['role_ID'];
            $errors[]="Користувача ".$user['ful_name']."' успішно
авторизовано';
            //echo "<div class='error'>Користувача успішно
авторизовано</div>";

        }
        else{
            $errors[]="Пароль не вірний!";
        }
    }
    else{
        $errors[]="Користувача з таким логіном не знайдено!";
    }
    $inf=array_shift($errors);
}
else{
    $inf=array_shift($errors);
}
}?)>
<div class='error'><?echo $inf;?></div>
<form method="post" action="practic.php" class="login"><p>
<label for="login">Логін:</label>
<input class="log" type="text" name="login" id="login"
value="<?=$_POST['login']?>" </p>
<p ID='ko'>
<label for="password">Пароль:</label>
<input class="log" type="password" name="password" id="password"
value=""></p>
<p class="login-submit">
<button type="submit" class="login-button"
name="exit">Ввійти</button>
</p>
</form>
<?endif;?>
</div>
<?require_once 'footer.html'?></div>
</body>
</html>

```

## Додаток Г

### Файл practic.css

```

::-webkit-scrollbar {
    width: 10px;
    height: 3px;
}
::-webkit-scrollbar-button {
    background: #00315C;
}
::-webkit-scrollbar-track-piece {
    background: #EBF1FD
}
::-webkit-scrollbar-thumb {
    background: #CBD5E4;
}
html{

        body{
            margin-top: 0px;
            margin-right: 0px;
            margin-bottom: 0px;
            margin-left: 0px;
            background-
color:EBF1FD;
        }
        .login-submit, .login-submit:before,
        .login-submit:after {
            background-
color:E8E8FF;
        }
        #content{
            width: 1110px;
            margin: 0 auto;
            position: relative;
            font-family:"Helvetica
Neue",Helvetica,Arial,sans-serif;
        }
        #headerMain {
            width:1110px;
            height: 105px;
            margin:0;
            z-index:0;
        }
        header{
            z-index:999;
            width:1110px;
            position: fixed;
            font-size:18px;

            background: #CBD5E4;
            height:105px;
        }
        footer{
            width:1110px;
            position:relative;
            height:30px;
            font-size:18px;
            background: #CBD5E4;
            vertical-align:bottom;
        }
    }

img.logo{
    margin-left:10px;
    height: 50px;
}
img.x{
    vertical-align:bottom;
    margin:0;
    padding:0;
    height: 18px;
}
.log{
    position: relative;
}
.rev{
    vertical-align:top;
    margin-top:10px;
    margin-left:-5px;
}
b{
    cursor:default;
    color: #00315C;
    font-size:30px;
    display: inline-block;
}
.manuul{
    padding:0;
    margin-bottom:0;
    margin-top:5px;
}
.manuli{
    display: inline-block;
    background: #E8E8FF;
    margin-right: 7px;
    transform: skewX(-20deg);
}
.manua {
    color: #00315C;
    display: block;
    padding: 10px 20px;
    text-decoration: none;
    transform: skewX(20deg);
}
.toggle{
    color: #00315C;
    text-decoration: none;
    float: right;
    margin-right:25px;
    margin-top:18px;
}
H3{
    font-size:25px;
    cursor:default;
    color: #00315C;
    text-align:center;
    margin:0;
    padding-top:40px;
    margin-bottom:50px; }
.togg{
    text-decoration: none;

```

```

        height:18px;
    }
    li.selected {
        border-bottom: 5px
#E8E8FF solid;
    }
    .tex{
        text-align:center;
        font-size:18px;
        color:001B38;
    }
    .error{
        padding-top:20px;
        font-size:35px;
        text-align:center;
        color:#43097A;
        margin-bottom:0;
    }
    .kol{
        font-family:"Helvetica
Neue",Helvetica,Arial,sans-serif;
        position:relative;
        min-height:490;
        background-
color:E8E8FF;
    }
    footer p{
        margin:0;
        padding-top:5px;
        text-align:center;
        color:#43097A;
    }
    H1{
        cursor:default;
        margin:0;
        padding-bottom:50px;
        padding-top:50px;
        text-align:center;
        color:#022C56;
    }
    p.info{
        margin:0 100px 0 100px;
        font-size:18px;
        color:001B38;
    }
}
.login {
    position: relative;
    margin: 70px auto;
    margin-bottom:0;
    margin-top:0;
    padding-top:40;
    width: 400px;
    padding-right: 32px;
    font-weight: 300;
    color: #a8a7a8;
    text-shadow: 1px 1px 0 rgba(0, 0, 0, 0.8);
}
.login p {
    margin: 0 0 10px;
}

        .login input, button, label {
            font-family: 'Helvetica Neue', Helvetica,
            Arial, sans-serif;
            font-size: 15px;
            font-weight: 300;
            -webkit-box-sizing: border-box;
            -moz-box-sizing: border-box;
            box-sizing: border-box;
        }
        .niz{
            margin-bottom:0;
        }
        .login input[type=text],
input[type=password] {
            padding: 0 10px;
            width: 300px;
            height: 40px;
            color: #bbb;
            text-shadow: 1px 1px 1px black;
            background: rgba(0, 0, 0, 0.16);
            border: 0;
            border-radius: 5px;
            -webkit-box-shadow: inset 0 1px 4px
            rgba(0, 0, 0, 0.3), 0 1px rgba(255, 255, 255, 0.06);
            box-shadow: inset 0 1px 4px rgba(0, 0, 0,
            0.3), 0 1px rgba(255, 255, 255, 0.06);
        }
        .login input[type=text]:focus,
input[type=password]:focus {
            color: white;
            background: rgba(0, 0, 0, 0.1);
            outline: 0;
        }
        .login label {
            float: left;
            width: 100px;
            line-height: 40px;
            padding-right: 10px;
            font-weight: 100;
            text-align: right;
            letter-spacing: 1px;
        }
        .forgot-password {
            padding-left: 100px;
            font-size: 13px;
            font-weight: 100;
            letter-spacing: 1px;
        }
        .login-submit {
            margin-bottom:0;
            position: absolute;
            top: 52px;
            right: 0;
            width: 48px;
            height: 48px;
            padding: 8px;
            border-radius: 32px;
            -webkit-box-shadow: 0 0 4px rgba(0, 0, 0,
            0.35);
            box-shadow: 0 0 4px rgba(0, 0, 0, 0.35);
        }

```

```

    }
    .login-submit:before, .login-submit:after {
        content: "";
        z-index: 1;
        position: absolute;
    }
    .login-submit:before {
        top: 28px;
        left: -4px;
        width: 4px;
        height: 10px;
        -webkit-box-shadow: inset 0 1px rgba(255,
255, 255, 0.06);
        box-shadow: inset 0 1px rgba(255, 255,
255, 0.06);
    }
    .login-submit:after {
        top: -4px;
        bottom: -4px;
        right: -4px;
        width: 36px;
    }

    .login-button {
        position: relative;
        z-index: 2;
        width: 48px;
        height: 48px;
        padding: 0 0 48px;
        /* Fix wrong positioning in Firefox 9 &
older (bug 450418) */
        text-indent: 120%;
        white-space: nowrap;
        overflow: hidden;
        background: none;
        border: 0;
        border-radius: 24px;
        cursor: pointer;
        -webkit-box-shadow: inset 0 1px 1px
rgba(0, 0, 0, 0.2), 0 1px rgba(255, 255, 255, 0.1);
        box-shadow: inset 0 1px 1px rgba(0, 0, 0,
0.2), 0 1px rgba(255, 255, 255, 0.1);
        /* Must use another pseudo element for the
gradient background because Webkit */
        /* clips the background incorrectly inside
elements with a border-radius. */
    }

    .login-button:before {
        content: "";
        position: absolute;
        top: 5px;
        bottom: 5px;
        left: 5px;
        right: 5px;
        background: #999;
        border-radius: 24px;
        background-image: -webkit-linear-
gradient(top, #00a2d3, #0d7796);
        background-image: -moz-linear-
gradient(top, #00a2d3, #0d7796);
        background-image: -o-linear-gradient(top,
#00a2d3, #0d7796);
        background-image: linear-gradient(to
bottom, #00a2d3, #0d7796);
        -webkit-box-shadow: inset 0 0 1px
#00a2d3, 0 0 5px rgba(0, 0, 0, 0.16);
        box-shadow: inset 0 0 1px #00a2d3, 0 0 0
5px rgba(0, 0, 0, 0.16);
    }
    .login-button:active:before {
        background: #999;
        background-image: -webkit-linear-
gradient(top, #0591ba, #00a2d3);
        background-image: -moz-linear-
gradient(top, #0591ba, #00a2d3);
        background-image: -o-linear-gradient(top,
#0591ba, #00a2d3);
        background-image: linear-gradient(to
bottom, #0591ba, #00a2d3);
    }
    .login-button:after {
        content: "";

        position: absolute;
        top: 15px;
        left: 12px;
        width: 25px;
        height: 19px;
        background: url("../img/arrow.png");
    }
}

```



## Додаток Д

### Файл vibor.php

```

<?php session_start();
if(!isset($_SESSION['ful_name']))
{
    header('Location:practic.php');
}
?>
<html>
<head>
    <title>Вибір</title>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
    <link rel="stylesheet" href="css/practic.css">
    <link rel="stylesheet" href="css/vibor.css">
    <script src="https://code.jquery.com/jquery-1.10.2.js"></script>
</head>
<body>
<div ID="content">
    <div id="headerMain">
        <header>
            <div class="log">
                
                <B class="rev">Сумиобленерго</B>
                <a class="toggle" href="php/logaut.php" ><?echo $_SESSION['ful_name'];?> вийти
                </a>
            </div>
            <center>
                <ul class="manuul">
                    <li class='manuli'>
                        <a class="manua" href="practic.php" >Головна</a></li>
                    <li class="selected manuli"><a class="manua" href="vibor.php"
>Відобразити</a></li>
                    <li class="manuli">
                        <a class="manua" href="add_work.php" >Додати нову роботу
                        <b class="caret"></b></a>
                    </li></ul>
                </center>
            </header>
        </div>
        <div class="kol">
            <? if($_SESSION['role_ID']==1||$_SESSION['role_ID']==3 ):?>
            <form METHOD="POST" action="php/vidobr.php" class="vibor">
            <H3>Оберіть необхідні параметри відображення</H3>
            <label class='fio' for="city">Вкажіть прізвище відповідального
            працівника </label>
            <ul class='blok'>
                <li class='city'><input id="city" autocomplete="off" name="city"
onkeyup='PressKey(event)' placeholder='Відобразити всix' />
                <select id="info" size=5
                    onchange="getObj('city').value=this.options[this.selectedIndex].value;"
                    onkeyup="getObj('city').value=this.options[this.selectedIndex].value;"
                    onclick="tuoclick();
getObj('city').value=this.options[this.selectedIndex].value;
var ki=this.options[this.selectedIndex].value;
$('.blok').append(vibor(ki));">
            </select></li>

```

```

</ul>
<? if($_SESSION['role_ID']==1):?>
<p ><label class='fio'>Оберіть підрозділ </label>
      <select id='sel' name="service">
        <option value="1" selected >СПС</option>
        <option value="2">СПП</option>
      </select></p>
<?endif;?>
<p ><label class='fio' >Оберіть рік та місяць</label>
<input id="month" name='month' type="month" value="" min='2018-10' max=
'2025-12'/>
</p>
<p class="bott">
<input id='addwork' value='Сформувати план' type='button'>
<Button id='submit' type='submit' autofocus>Відобразити</Button>
<input id='addzvrit' value='Сформувати звіт' type='button'>
</p>
</form>
      <? else: ?>
      <form METHOD="POST" action="php/vidobr.php" class="vibor">
<H3>Оберіть необхідні параметри відображення</H3>
      <label class='fio' for="city">Вкажіть прізвище відповідального
працівника</label>
      <ul class='blok'>
        <li class='city'><input id="city" name="city" readonly
value='<?=$_SESSION['ful_name']?' /> />
      </ul>
      <p ><label class='fio' >Оберіть місяць та рік виконання робіт</label>
<input id="month" name='month' type="month" value="" min='2018-10' max=
'2025-12'/>
</p>
<p class="bott">
<Button id='submit' type='submit' autofocus>Відобразити</Button>
</p>
</form>
<?endif;?>
</div>
<?require_once 'footer.html'?>
</div>
</body>
<script type="text/javascript" src="javascript/vibor.js">
</script>
</html>

```

## Додаток Е

## Файл vibor.js

```

var ot="", timer=0, x=-1,y=0;
    var date = new Date();
var year=date.getFullYear();
    var month=date.getMonth()+1;
    var yearmo=year+'-'+month;
    document.getElementById('month').value=yearmo;
    function del(lo)
    {
    $(getObj(lo)).remove();
    }
    function vibor(vibor)
    {
    i=0;
    while(document.getElementById(i)){
    i = (new Date()).getTime();
    }
    vibor="<li class='livib' id='"+i+"'><input name='worker[]' class='vibor' value='"+vibor+"
readonly ><a class='togg' href='#' onclick='del("+i+" )><img src='img/x.png' class='x'></a></li>";
    return vibor;
    }
    function tuoclick()
    {
    getObj('city').focus();
    getObj('info').style.display = 'none';
    }
function PressKey2(e,po){ // вызывается при нажатии клавиши в select
    e=e||window.event;
    t=(window.event) ? window.event.srcElement : e.currentTarget; // объект для которого вызывно
    if(e.keyCode==13){ // Enter

        getObj('city').focus();
        getObj('info').style.display = 'none';

        return;}
    }
// Определение координаты элемента
function pageX(elem) {
    return elem.offsetParent ?
        elem.offsetLeft + pageX( elem.offsetParent ) :
        elem.offsetLeft;
}
function pageY(elem) {
    return elem.offsetParent ?
        elem.offsetTop + pageY( elem.offsetParent ) :
        elem.offsetTop;
}
function PressKey(e){
    e=e||window.event;
    t=(window.event) ? window.event.srcElement : e.currentTarget; // объект для которого вызвано
    g=getObj('info');
    /*if(x===-1&&y==0){// при первом обращении просчитываю координаты
        x=pageX(t); y=pageY(t);
        g.style.top = y + t.clientHeight+1 + "px";
        g.style.left = x + "px";
    }*/
    if(e.keyCode==40){g.focus();
        g.selectedIndex=0;return;}
    if(ot==t.value)return; // если ничего не изменилось не "замучить" сервер
    ot=t.value;

```

```

    if(ot.length<1){
        getObj('info').style.display='none'; // спрячем select
        return;}
    timer=window.setTimeout('Load()',50); // загружаю через 1 секунду после последнего нажатия
клавиши
}
function Load(){
    timer=0;
    o=getObj('info');
    o.options.length=0;
    ajaxLoad('info', './php/new.php?city_name='+ot,"<option value='Відобразити всіх' >Відобразити
всіх</option>\n","");
    o.style.display='block';
}
function getObj(objID)
{if (document.getElementById) {return document.getElementById(objID);}
else if (document.all) {return document.all[objID];}
else if (document.layers) {return document.layers[objID];}
}
function ajaxLoad(obj,url,defMessage,callback){
    var ajaxObj;
    if (defMessage) document.getElementById(obj).innerHTML=defMessage;
    if(window.XMLHttpRequest){
        ajaxObj = new XMLHttpRequest();
    } else if(window.ActiveXObject){
        ajaxObj = new ActiveXObject("Microsoft.XMLHTTP");
    } else {
        return;
    }
    ajaxObj.open ('GET', url);
    if (ajaxObj.setRequestHeader)
        ajaxObj.setRequestHeader("Content-Type", "application/x-www-form-urlencoded; charset=utf-8");

    ajaxObj.onreadystatechange = ajaxCallBack(obj,ajaxObj,(callback?callback:null));
    ajaxObj.send();
    return false;
}

function updateObj(obj, data, bold, blink){
    if(bold)data=data.bold();
    if(blink)data=data.blink();
    if(data.length>1) document.getElementById(obj).innerHTML = data; // упрощенный
вариант, работает не во всех браузерах
}
function ajaxCallBack(obj, ajaxObj, callback){
    return function(){
        if(ajaxObj.readyState == 4){
            if(callback) if(!callback(obj,ajaxObj))return;
            if (ajaxObj.status==200)
                updateObj(obj, ajaxObj.responseText);
            else updateObj(obj, ajaxObj.status+' '+ajaxObj.statusText,1,1);
        }
    }
}

$('#addzvit').bind("click", function()
{
    var month = $('#month').val();
    var service=$('#sel').val();
    $('#addzvit').remove();
    $.ajax({
        url:'php/zvit.php',
        type:'POST',

```

```

data: "month="+month+"&service="+service,
beforeSend: function(html) {
    $('#bott').append("<input id='addzvit' value='Формую звіт...'
type='button'>");

    },
success: function(result){
    if(result)
    {
        $('#addzvit').remove();
        $('#bott').append("<a href='php/down.php?'>"+result+"</a>");
    }
    else{
        $('#Danzvit').val('Помилка');
    }
    }
});

$('#addwork').bind("click", function()
{
    var month = $('#month').val();
    var service=$('#sel').val();
    $('#addwork').remove();
    $.ajax({
        url:'php/plan.php',
        type:'POST',
        data: "month="+month+"&service="+service,
        beforeSend: function(html) {
            $('#bott').append("<input id='addwork' value='Формую план...'
type='button'>");

            },
success: function(result){
            if(result)
            {
                $('#addwork').remove();
                $('#bott').append("<a href='php/davp.php?'>"+result+"</a>");
            }
            else{
                $('#Danplan').val('Помилка');
            }
            }
        });

        $('#Danplan').remove();
    });

    $('#bott').on("click", "a#Danplan",function()
    {
        $('#Danplan').remove();
    });

    $('#bott').on("click", "a#Danzvit",function()
    {
        $('#Danzvit').remove();
    });
});

```

## Додаток Ж

## Файл vidobr.php

```

<?php session_start();
if(!isset($_SESSION['ful_name'])) {
    header('Location:../practic.php'); }
?>
<html>
<head>
    <title>Главная</title>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
    <link rel="stylesheet" href=" ../css/practic.css">
    <script src="https://code.jquery.com/jquery-1.10.2.js"></script>
    <link rel="stylesheet" href=" ../css/vidobr.css">
</head>
<body>
<div ID="content">
    <div id="headerMain">
        <header>
            <div class="log">
                
                <B class="rev">Сумиобленерго</B>
                <a class="toggle" href="logaut.php" ><?echo $_SESSION['ful_name'];?> вийти
</a>
            </div>
            <center>
            <ul class="manuul">
                <li class='manuli'>
                    <a class="manua" href=" ../practic.php" >Головна</a></li>
                <li class="selected manuli"><a class="manua" href=" ../vibor.php"
>Відобразити</a></li>
                <li class="manuli">
                    <a class="manua" href=" ../add_work.php" >Додати нову роботу
                    <b class="caret"></b></a>
                </li>
            </ul>
            </center>
        </header>
    </div>
    <div class="kol">
        <h3 align=center class='vidobr'><Button id='addwork'><a href=' ../vibor.php'>Змінити
параметри</a></Button>Каталог робіт за місяць
        <Button id='addworker'><a href=' ../add_work.php'>Додати роботу</a></Button></h3>
</div>
</div>
<?php
if ($_POST['month']) {
    $month = $_POST['month'];
    list ($year,$month) = explode("-", $month);
    $link = mysqli_connect("localhost", "root", "", "practic")
    or die("Ошибка " . mysqli_error($link));
    mysqli_query ($link,"set names 'utf8'");
    $ful_name=$_SESSION['ful_name'];
    $service=$_SESSION['service_ID'];
    $role=$_SESSION['role_ID'];
    function hello($ful_name, $basa, $Y, $M, $rol=", $serv=")
    {
        if($rol!=1&&$rol!=3) {
            $where_serv="";
        }
        else if($rol==3) {
            $where_serv=" AND worker.service_ID= ".$serv." ";}
    }
}

```

```

else{
    $where_serv="          AND          worker.service_ID=
".$_POST['service']."" ";}
if(!$ful_name == ") {
    $total = count($ful_name)-1;
    $k=0;
    $where_worker=array();
    foreach ($ful_name as $i) {
        if($k==0) {
            if($k==$total)
            {
                $ind="AND worker.ful_name= ".$i." ";
            }
            else
            {
                $ind="AND (worker.ful_name= ".$i." ";
            }
        }
        else
        {
            if($k==$total)
            {
                $ind="OR worker.ful_name= ".$i."";
            }
            else
            {
                $ind="OR worker.ful_name= ".$i."";
            }
        }
        $where_worker[$k]=$ind;
        $k++;
    }
    $where_worker = implode(" ", $where_worker);
}
$query="SELECT work.ID, work.name_work, worker.ful_name,
work.persent, work.state_work, work.date_start, work.date_finish
FROM work JOIN worker ON `work`.`worker_ID` = `worker`.`ID`
WHERE YEAR(work.date_start)=$Y AND MONTH(work.date_start)=$M
AND MONTH(work.date_finish)=$M $where_worker $where_serv
ORDER BY worker.ful_name";

$result = mysqli_query($basa,$query);
print "<form METHOD='POST' action='izm.php' class='fom'>";
print "<input type='hidden' ID='work' name='ID' value='>";
print "<table class='rez' align=center>";
print "<tr class='head'><th class='red'></th><th>Найменування
роботи(заходу)</th><th class='dat'>Дата початку виконання</th><th class='dat'>Дата завершення
виконання</th>
<th class='pers'>Показник,%</th><th class='dat'>Відмітка про виконання</th></tr>";
while($row = mysqli_fetch_array($result)){
    print "<tr class='cont' ID=".$row["ID"].">\n";
    print "<td class='bott'><input ID='drop_ ".$row['ID'].
type='button' class='drop' title='Видалити'>
<input type='image' src='./img/izm.jpg'
ID='izm_ ".$row['ID']."' class='izm' title='Змінити'></td>
<td>".$row["name_work"]."</td>\n<td class='dat'>
".$row["date_start"]. "</td>\n<td class='dat'> ".$row["date_finish"]. "</td>\n
<td class='dat'>".$row["ful_name"]."</td>\n<td class='pers'>
".$row["persent"]. "</td>\n<td class='dat'> ".$row["state_work"]. "</td>\n";
    print "</tr>\n";
}

```

```

    }
    echo "</table>";
    print "</form>";
    mysqli_close($basa);
}
if (isset($_POST['worker']))
{
    $worker=array($_POST['worker']);
    $worker=$worker[0];
    $vse='Відобразити всіх';
    if (in_array($vse,$worker))
    {
        $worker="; //необхідно відобразити всіх
робітників
        hello($worker, $link, $year, $month, $role, $service);
    }
    Else
    {
        $worker = array_unique($worker);//отримано масив
унікальних значень
        hello($worker, $link, $year, $month, $role, $service);
    }
}
else
{
    if($role==2)
    {
        $worker[]=$ful_name;
        hello($worker, $link, $year, $month);
    }
    else
    {
        $worker=""; //необхідно відобразити всіх робітників
hello($worker, $link, $year, $month, $role, $service);
    }
}
}
else
{
    echo "<p>Введено не всі дані <a href='../vibor.php'>Обрати
параметри</a></P>";//значення місяця не отримано
}
?>
</div>
<footer>
<p >
Автор WEB-рішення Олексенко Станіслав!!!</p>
</footer>
</div>
</body>
</html>

```



## Додаток И

## Файл plan.php

```

<?php session_start();
if(!isset($_SESSION['ful_name']))
{die;
}
if($_SESSION['role_ID']!=1 && $_SESSION['role_ID']!=3)
{
    die;
}

    if ($_POST['month'])
    {
        $sq="";
        if($_SESSION['role_ID']==3)
            {
                $role=$_SESSION['role_ID'];
                $serv=$_SESSION['service_ID'];
                $sq=" AND worker.service_ID='$serv' ";
                $ker_query="SELECT ful_name FROM worker WHERE
role_ID='$role' AND service_ID='$serv'";
                $serv_query="SELECT service FROM service WHERE ID='$serv'";
            }
        else{
            $role=3;
            $serv=$_POST['service'];
            $sq=" AND worker.service_ID='$serv' ";
            $ker_query="SELECT ful_name FROM worker WHERE
role_ID='$role' AND service_ID='$serv'";
            $serv_query="SELECT service FROM service WHERE
ID='$serv'";
        }
        $month = $_POST['month'];
        list ($year,$mon) = explode("-", $month);
        $mo=$mon-1;
        $link = mysqli_connect("localhost", "root", "", "practic")
or die("Ошибка " . mysqli_error($link));
        mysqli_query ($link,"set names 'utf8'");
        $ker_mas=mysqli_query($link,$ker_query);
        $row_ker = mysqli_fetch_array($ker_mas);
        $ker=$row_ker['ful_name'];
        $serv_mas=mysqli_query($link,$serv_query);
        $row_serv = mysqli_fetch_array($serv_mas);
        $service=$row_serv['service'];
        //var_dump($service);
        $query="SELECT work.ID, work.name_work, worker.ful_name,
work.persent, work.state_work, work.date_start, work.date_finish
FROM work JOIN worker ON `work`.`worker_ID` = `worker`.`ID`
WHERE YEAR(work.date_start)=$year AND
YEAR(work.date_finish)=$year AND MONTH(work.date_start)=$mon
AND MONTH(work.date_finish)=$mon $sq ORDER BY
worker.ful_name";

        $result = mysqli_query($link,$query);
        $months=array('січня','лютого','березня', 'квітня', 'травня', 'червня', 'липня',
'серпня', 'вересня', 'жовтня', 'листопада', 'грудня' );
        $monthsn=array('січень','лютий','березень', 'квітень', 'травень', 'червень', 'липень', 'серпень', 'вересень',
'жовтень', 'листопад', 'грудень' );
        require 'vendor/autoload.php';
        $phpWord=new \PhpOffice\PhpWord\PhpWord();
        $section = $phpWord->addSection(array('marginTop'=>500,'marginRight'=>900,'marginLeft'=>900,));
        $fontStyleName = 'Head';

```

```

$phpWord->addFontStyle(
    $fontStyleName,
    array('name' => 'Times New Roman', 'size' => 12, 'bold' => true)
);
$section->addText(
    '        "ПОГОДЖЕНО"
"ЗАТВЕРДЖУЮ"',
    $fontStyleName,
    array(
        'space' => array('before' => 0, 'after' => 0),
        'spacing'=>250
    )
);

$section->addText(
    '        Заступник директора з ІТ
        '        ПАТ «Сумиобленерго»
        '        _____ О.П. Колесніченко
        _____ В.П.Цуканов
        '        « ____ »'. $months[$mo]. '$year.' p.
'. $months[$mo]. '$year.' p.',
    $fontStyleName,
    array('spacing'=>15)
);
$section->addText(
    'План робіт',
    array('name' => 'Times New Roman', 'size' => 16, 'bold' => true),
    array(
        'space' => array('before' => 0, 'after' => 0),
        'spacing'=>15,
        'align'=>'center'
    )
);
$section->addText(
    'Служби програмного супроводу',
    $fontStyleName,
    array(
        'space' => array('before' => 0, 'after' => 0),
        'spacing'=>15,
        'align'=>'center'
    )
);

$section->addText(
    'на '. $months[$mo]. '$year.' p.',
    $fontStyleName,
    array(
        'space' => array('before' => 0, 'after' => 200),
        'spacing'=>15,
        'align'=>'center'
    )
);
$fontStyleTabl = 'tabl';
$phpWord->addFontStyle(
    $fontStyleTabl,
    array('name' => 'Times New Roman', 'size' => 10)
);

$styleTable = array('borderSize' => 10, 'borderColor' => '000000');

```

```

$cellTO=array('space' => array('before' => 20, 'after' => 25),
              'spacing'=>15);

$cellHCentered = array('align' => 'center','space' => array('before' => 20, 'after' => 25),
                       'spacing'=>15);
$haed=array('name' => 'Times New Roman', 'size' => 10, 'bold' => true);
$cellVCentered = array('valign' => 'center');

$phpWord->addTableStyle('Colspan Rowspan', $styleTable);
$stable = $section->addTable('Colspan Rowspan');

        $stable->addRow(null, array('tblHeader' => true));
$stable->addCell(400, $cellVCentered)->addText('№ п/п', $haed, $cellHCentered);
$stable->addCell(5700, $cellVCentered)->addText('Найменування роботи(заходу)', $haed,
$cellHCentered);
$stable->addCell(1250, $cellVCentered)->addText('Термін виконання', $haed, $cellHCentered);
        $stable->addCell(1550, $cellVCentered)->addText('Відповідальний', $haed, $cellHCentered);
$stable->addCell(700, $cellVCentered)->addText('Пока-зник, %', $haed, $cellHCentered);
$stable->addCell(1000, $cellVCentered)->addText('Підпис відп.особи', $haed, $cellHCentered);
        $i=1;
        $cont=array('name' => 'Times New Roman', 'size' => 10);

        while($row = mysqli_fetch_array($result)){
                if(stristr($row['name_work'], '(000)') === FALSE)
                {
                        list ($year,$month1,$day1) = explode("-", $row["date_start"]);
                        list ($year,$month2,$day2) = explode("-", $row["date_finish"]);
                        $legh= date("t", mktime(0, 0, 0, $mon+1, 0, $year));
                        if($day1==$day2){
                                $stable->addRow();
                                $stable->addCell(400, $cellVCentered)->addText($i.'.', $cont,
$cellHCentered);
                                $stable->addCell(5700, $cellVCentered)->addText($row['name_work'],
$cont, $cellTO);
                                $stable->addCell(1250, $cellVCentered)->addText($day2.'.'. $month2,
$cont, $cellHCentered);
                                $stable->addCell(1550, $cellVCentered)->addText($row["ful_name"],
$cont, $cellHCentered);
                                $stable->addCell(700, $cellVCentered)->addText($row["persent"],
$cont, $cellHCentered);
                                $stable->addCell(1000, $cellVCentered)->addText(' ', $cont,
$cellHCentered);
                        }
                        else if($day1==1&&$day2==$legh){
                                $stable->addRow();
                                $stable->addCell(400, $cellVCentered)->addText($i.'.', $cont,
$cellHCentered);
                                $stable->addCell(5700, $cellVCentered)->addText($row['name_work'],
$cont, $cellTO);
                                $stable->addCell(1250, $cellVCentered)->addText('протягом місяця',
$cont, $cellHCentered);
                                $stable->addCell(1550, $cellVCentered)->addText($row["ful_name"],
$cont, $cellHCentered);
                                $stable->addCell(700, $cellVCentered)->addText($row["persent"],
$cont, $cellHCentered);
                                $stable->addCell(1000, $cellVCentered)->addText(' ', $cont,
$cellHCentered);
                        }
                }
                else if($day1==1){
                        $stable->addRow();

```

```

$cellHCentered);
$cont, $cellTO);
$.day2.'.'.month2, $cont, $cellHCentered);
$cont, $cellHCentered);
$cont, $cellHCentered);
$cellHCentered);

$cont, $cellTO);
$.day2.'.'.month2, $cont, $cellHCentered);
$cont, $cellHCentered);
$cont, $cellHCentered);
$cellHCentered);

}
    $i++;
}

mysqli_close($link);

    $section->addText(
        '    Начальник '.$service.'    ',
        $fontStyleName,
        array(
            'space' => array('before' => 600, 'after' => 0)
        )
    );
$objWriter = \PhpOffice\PhpWord\IOFactory::createWriter($phpWord, 'Word2007');
$objWriter->save('word/plan.docx');

echo ('Скачати план');
}
?>

```

## Додаток К

### Файл zvit.php

```

<?php session_start();
if(!isset($_SESSION['ful_name']))
{
    die;
}
if($_SESSION['role_ID']!=1 && $_SESSION['role_ID']!=3)
{
    die;
}

if ($_POST['month'])
{
    $sq="";
    if($_SESSION['role_ID']==3)
    {
        $role=$_SESSION['role_ID'];
        $serv=$_SESSION['service_ID'];
        $sq=" AND worker.service_ID='$serv' ";
        $ker_query="SELECT ful_name FROM worker WHERE
role_ID='$role' AND service_ID='$serv'";
        $serv_query="SELECT service FROM service WHERE
ID='$serv'";
    }
    else{
        $role=3;
        $serv=$_POST['service'];
        $sq=" AND worker.service_ID='$serv' ";
        $ker_query="SELECT ful_name FROM worker WHERE
role_ID='$role' AND service_ID='$serv'";
        $serv_query="SELECT service FROM service WHERE
ID='$serv'";
    }
    $month = $_POST['month'];
    list ($year,$mon) = explode("-", $month);

    $mo=$mon-1;
    $link = mysqli_connect("localhost", "root", "", "practic")
or die("Ошибка " . mysqli_error($link));
    mysqli_query ($link,"set names 'utf8'");
    $ker_mas=mysqli_query($link,$ker_query);
    $row_ker = mysqli_fetch_array($ker_mas);
    $ker=$row_ker['ful_name'];
    $serv_mas=mysqli_query($link,$serv_query);
    $row_serv = mysqli_fetch_array($serv_mas);

    $service=$row_serv['service'];
    $query="SELECT work.ID, work.name_work, worker.ful_name,
work.persent, work.state_work, work.date_start, work.date_finish
FROM work JOIN worker ON `work`.`worker_ID` = `worker`.`ID`
WHERE YEAR(work.date_start)=$year AND
YEAR(work.date_finish)=$year AND MONTH(work.date_start)=$mon
AND MONTH(work.date_finish)=$mon $sq ORDER BY
worker.ful_name";

    $result = mysqli_query($link,$query);
    $months=array('січня','лютого','березня', 'квітня', 'травня', 'червня', 'липня',
'серпня', 'вересня', 'жовтня', 'листопада', 'грудня' );

```

```

$months=array('січень','лютий','березень', 'квітень', 'травень', 'червень',
'липень', 'серпень', 'вересень', 'жовтень', 'листопад', 'грудень' );
require 'vendor/autoload.php';
$phpWord=new \PhpOffice\PhpWord\PhpWord();
$section = $phpWord-
>addSection(array('marginTop'=>500,'marginRight'=>900,'marginLeft'=>900,));
$fontStyleName = 'Head';
$phpWord->addFontStyle(
    $fontStyleName,
    array('name' => 'Times New Roman', 'size' => 12, 'bold' => true)
);

$section->addText(
    '                "ПОГОДЖЕНО"
"ЗАТВЕРДЖУЮ"',
    $fontStyleName,
    array(
        'space' => array('before' => 0, 'after' => 0),
        'spacing'=>250
    )
);

$section->addText(
    '                Заступник директора з ІТ
Директор з ІТ                '
    '                ПАТ «Сумиобленерго»
ПАТ «Сумиобленерго»                '
    '                _____ О.П. Колесніченко
_____ В.П.Цуканов                '
    '                « ____ » '.$months[$mo].'. $year.' p.
« ____ » '.$months[$mo].'. $year.' p.',
    $fontStyleName,
    array(
        'spacing'=>15
    )
);

$section->addText(
    'Звіт до плану робіт',
    $fontStyleName,
    array(
        'space' => array('before' => 0, 'after' => 0),
        'spacing'=>15,
        'align'=>'center'
    )
);

$section->addText(
    'Служби програмного супроводу',
    $fontStyleName,
    array(
        'space' => array('before' => 0, 'after' => 0),
        'spacing'=>15,
        'align'=>'center'
    )
);

$section->addText(
    'дирекції з інформаційних технологій',
    $fontStyleName,
    array(
        'space' => array('before' => 0, 'after' => 0),
        'spacing'=>15,
        'align'=>'center'
    )
);

```

```

    )
);
$section->addText(
    'ПАТ "Сумиобленерго" за '.$monthsn[$mo].'.'.$year.' р.',
    $fontStyleName,
    array(
        'space' => array('before' => 0, 'after' => 200),
        'spacing'=>15,
        'align'=>'center'
    )
);
$fontStyleTabl = 'tabl';
$phpWord->addFontStyle(
    $fontStyleTabl,
    array('name' => 'Times New Roman', 'size' => 10)
);
    $styleTable = array('borderSize' => 10, 'borderColor' => '000000');
    $cellTO=array('space' => array('before' => 20, 'after' => 25),
'spacing'=>15);
    $cellHCentered = array('align' => 'center','space' => array('before' =>
0, 'after' => 0),
'spacing'=>15);
    $haed=array('name' => 'Times New Roman', 'size' => 10, 'bold' =>
true);
        $cellVCentered = array('valign' => 'center');
        $phpWord->addTableStyle('Colspan Rowspan', $styleTable);
        $table = $section->addTable('Colspan Rowspan');
    $table->addRow();
    $table->addCell(400, $cellVCentered)->addText('№ п/п', $haed,
$cellHCentered);
        $table->addCell(4500, $cellVCentered)->addText('Найменування
роботи(заходу)', $haed, $cellHCentered);
        $table->addCell(1050, $cellVCentered)->addText('Термін виконання', $haed,
$cellHCentered);
        $table->addCell(1550, $cellVCentered)->addText('Відповідальний', $haed,
$cellHCentered);
        $table->addCell(700, $cellVCentered)->addText('Пока-зник, %', $haed,
$cellHCentered);
        $table->addCell(1400, $cellVCentered)->addText('Відмітка про виконання',
$haed, $cellHCentered);
        $table->addCell(1000, $cellVCentered)->addText('Підпис відп.особи', $haed,
$cellHCentered);
    $i=1;
    $cont=array('name' => 'Times New Roman', 'size' => 10);
    while($row = mysqli_fetch_array($result)){
        if(stristr($row['name_work'], '(000)') === FALSE)
        {
            list ($year,$month1,$day1) = explode("-", $row["date_start"]);
            list ($year,$month2,$day2) = explode("-", $row["date_finish"]);
            $legh= date("t", mktime(0, 0, 0, $mon+1, 0, $year));
            if($day1===$day2){
                $table->addRow();
                $table->addCell(400, $cellVCentered)->addText($i.'.',
$cont,$cellHCentered);
                $table->addCell(4500, $cellVCentered)->addText($row['name_work'],
$cont,$cellTO);
                $table->addCell(1050, $cellVCentered)->addText($day2.'.'.$month2,
$cont, $cellHCentered);
                $table->addCell(1550, $cellVCentered)->addText($row["ful_name"],
$cont, $cellHCentered);
            }
        }
    }

```

```

$cont, $cellHCentered);
$cont, $cellHCentered);
$cellHCentered);
}
else if($day1==1&&$day2==$legh){
    $table->addRow();
    $table->addCell(400, $cellVCentered)->addText($i.', $cont,
$cellHCentered);
    $table->addCell(4500, $cellVCentered)->addText($row['name_work'],
$cont,$cellTO);
    $table->addCell(1050, $cellVCentered)->addText('протягом місяця',
$cont, $cellHCentered);
    $table->addCell(1550, $cellVCentered)->addText($row["ful_name"],
$cont, $cellHCentered);
    $table->addCell(700, $cellVCentered)->addText($row["persent"],
$cont, $cellHCentered);
    $table->addCell(1400, $cellVCentered)->addText($row["state_work"],
$cont, $cellHCentered);
    $table->addCell(1000, $cellVCentered)->addText(' ', $cont,
$cellHCentered);
}
else if($day1==1){
    $table->addRow();
    $table->addCell(400, $cellVCentered)->addText($i.', $cont,
$cellHCentered);
    $table->addCell(4500, $cellVCentered)->addText($row['name_work'],
$cont,$cellTO);
    $table->addCell(1050, $cellVCentered)->addText('до
'. $day2.'.'. $month2, $cont, $cellHCentered);
    $table->addCell(1550, $cellVCentered)->addText($row["ful_name"],
$cont, $cellHCentered);
    $table->addCell(700, $cellVCentered)->addText($row["persent"],
$cont, $cellHCentered);
    $table->addCell(1400, $cellVCentered)->addText($row["state_work"],
$cont, $cellHCentered);
    $table->addCell(1000, $cellVCentered)->addText(' ', $cont,
$cellHCentered);
}
}
else{
    $table->addRow();
    $table->addCell(400, $cellVCentered)->addText($i.', $cont,
$cellHCentered);
    $table->addCell(4500, $cellVCentered)->addText($row['name_work'],
$cont,$cellTO);
    $table->addCell(1050, $cellVCentered)->addText($day1.'.'. $month1.'-
'. $day2.'.'. $month2, $cont, $cellHCentered);
    $table->addCell(1550, $cellVCentered)->addText($row["ful_name"],
$cont, $cellHCentered);
    $table->addCell(700, $cellVCentered)->addText($row["persent"],
$cont, $cellHCentered);
    $table->addCell(1400, $cellVCentered)->addText($row["state_work"],
$cont, $cellHCentered);
    $table->addCell(1000, $cellVCentered)->addText(' ', $cont,
$cellHCentered);
}
}
/*<td>". $row["name_work"]. "</td>\n<td class='dat'"
$.row["date_start"]. "</td>\n<td class='dat'" $.row["date_finish"]. "</td>\n
<td class='dat'" $.row["ful_name"]. "</td>\n<td class='pers'"
$.row["persent"]. "</td>\n<td class='dat'" $.row["state_work"]. "</td>\n";
print "</tr>\n"; */

```



```

        $i++;
    }
}
$section->addText(
    'Позапланові роботи',
    $fontStyleName,
    array(
        'space' => array('before' => 200, 'after' => 200),
        'spacing'=>15,
        'align'=>'center' ) );
$phpWord->addTableStyle('Colspan Rowspan', $styleTable);
    $table2 = $section->addTable('Colspan Rowspan');
    $table2->addRow();
    $table2->addCell(400, $cellVCentered)->addText('№ п/п', $haed,
$cellHCentered);
    $table2->addCell(5700, $cellVCentered)->addText('Найменування
роботи(заходу)', $haed, $cellHCentered);
    $table2->addCell(1800, $cellVCentered)->addText('Відповідальний', $haed,
$cellHCentered);
    $table2->addCell(1500, $cellVCentered)->addText('Відмітка про виконання',
$haed, $cellHCentered);
    $table2->addCell(1200, $cellVCentered)->addText('Підпис відп.особи',
$haed, $cellHCentered);

    $t=1;
    $cont=array('name' => 'Times New Roman', 'size' => 10);
    $result = mysqli_query($link,$query);
    while($row = mysqli_fetch_array($result)){
        if(stristr($row['name_work'], '(000)') !== FALSE)
        {
            $work = trim($row['name_work'], '(000)');

            $table2->addRow();
            $table2->addCell(400, $cellVCentered)->addText($t.',
$cont,$cellHCentered);
            $table2->addCell(5700, $cellVCentered)->addText($work,
$cont,$cellTO);
            $table2->addCell(1800, $cellVCentered)->addText($row["ful_name"],
$cont, $cellHCentered);
            $table2->addCell(1500, $cellVCentered)-
>addText($row["state_work"], $cont, $cellHCentered);
            $table2->addCell(1200, $cellVCentered)->addText(' ', $cont,
$cellHCentered);

            $t++;
        }
    }
    mysqli_close($link);

    $section->addText(
        'Начальник '.$service.' $ker,
$fontStyleName,
    array(
        'space' => array('before' =>600, 'after' => 0)
    )
);
$objWriter = \PhpOffice\PhpWord\IOFactory::createWriter($phpWord,
'Word2007');
$objWriter->save('word/zwit.docx');
echo ('Скачати звіт');
}
?>

```

## Додаток Л

## Файл add\_work.php

```

<?php session_start();
    require "bd_con.php";
    if(!isset($_SESSION['ful_name']))
    {
        header('Location:../practic.php');
    }
if($_SESSION['role_ID']!=1)
{
    header('Location:../practic.php');
}

if(isset($_POST['add']))
{
    $errors=array();
if(trim($_POST['worker'])=="")
    {
        $errors[]='Введіть логін користувача!';
    }
if($_POST['password']=="")
    {
        $errors[]="Введіть пароль!";
    }
if(trim($_POST['email'])=="")
    {
        $errors[]="Укажіть пошту!";
    }
if(empty($errors))
    {
        $qw="SELECT * FROM worker WHERE ful_name
        =".$_POST['worker']."";

        $user=R::getAll($qw);
if(!$user)
    {

        $ful_name=$_POST['worker'];
        $email=$_POST['email'];
        $password=
password_hash($_POST['password'],PASSWORD_DEFAULT);
        $service_ID=(int)$_POST['service'];
        $role_ID=$_POST['role'];

        $query_insert="INSERT INTO worker (ID, ful_name, email,
password, service_ID, role_ID)
VALUES (NULL, '$ful_name','$email', '$password',
'$service_ID', '$role_ID)";

        $result= R::exec( $query_insert );
if($result==0)
    {
        $errors[]='Виникла помилка!';
    }
else
    {
        $errors[]='Користувач '.trim($_POST['worker']).'
успішно зареєстрований!';

        unset($_POST);
    }
}
}

```

```

        }
    }
    else
    {
        $errors[]='Користувач з таким логіном уже існує!';
    }
    $er=array_shift($errors);
}
else
{
    $er=array_shift($errors);
}
}
}
?>
<html>
<head>
<title>Відобразити</title>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
<link rel="stylesheet" href=" ../css/add_worker.css">

</head>
<body>
<div ID="content">
    <div id="headerMain">
        <header>
            <div class="log">
                
                <B class="rev">Сумиобленерго</B>

                <a class="toggle" href="logout.php" ><?echo $_SESSION['ful_name'];?> вийти
                </a>

            </div>
            <center>
                <ul class="manuul">
                    <li class='manuli'>
                        <a class="manua" href=" ../practic.php" >Головна</a></li>
                    <li class="manuli"><a class="manua" href=" ../vibor.php"
                        >Відобразити</a></li>

                    <li class="manuli">
                        <a class="manua" href=" ../add_work.php" >Додати нову роботу
                        <b class="caret"></b></a>
                    </li>

                    <li class="selected manuli">
                        <a class="manua" href="add_worker.php" >Додати працівника
                        <b class="caret"></b></a>
                    </li>
                    <li class="manuli">
                        <a class="manua" href="drop_worker.php" >Видали працівника
                        <b class="caret"></b></a>
                    </li>

                </ul>
            </center>

        </header>

```

```

</div>
<div class="kol">
<div class='error'><?echo $er;?></div>
<form METHOD="POST" action="add_worker.php" >

    <table align='center'>
        <tr>
            <td class='labl'><label class='fio' >Прізвище та ініціали
працівника(логін)</label></td>
            <td class='inputi'><input type='text' ID='worker' name="worker"
maxlength='25' value='<?=@$_POST['worker'];?>' /> </td>
        </tr>
        <tr>
            <td class='labl'><label class='fio'>Електронна пошта працівника</label>
</td>
            <td class='inputi'><input type="email" name="email" maxlength='30'
value='<?echo @$_POST['email'];?>'></td>
        </tr>
        <tr>
            <td class='labl'><label class='fio'>Пароль</label> </td>
            <td class='inputi'><input type="password" name="password" maxlength='16'
></td>
        </tr>
        <tr>
            <td class='labl'><label class='fio'>Підрозділ </label> </td>
            <td class='inputi'><select name="service">
                <option value="1" selected >СПС</option>
                <option value="2">СРП</option>
            </select>
        </td>
        </tr>
        <tr>
            <td class='labl'><label class='fio'>Роль доступу до додатку</label> </td>
            <td class='inputi'><select name="role">
                <option value="1" >Адміністратор</option>
                <option value="2" selected>Працівник</option>
                <option value="3">Керівник</option>
            </select>
        </td>
        </tr>
    </table>
    <p style='text-align:center;'> <input type="submit" name="add" value="Додати працівника">
    <input type="reset" value="Стерти"></p>
</form>
</div>
<footer>
<p >
Автор WEB-рішення Олексенко Станіслав!!!</p>
</footer>
</div>
</body>
</html>

```

## Додаток М

## Файл add\_work.js

```

var date = new Date();
var year=date.getFullYear();
var month=date.getMonth();
var day=1;

fin = new Date(year, month+1, day);
st=new Date(year, month, day+1);

document.getElementById('start').valueAsDate=st;
document.getElementById('finish').valueAsDate=fin;
$('.lol').bind("click", function()
{
    var work = $('#work').val();
    var worker = $('#worker').val();
    var start = $('#start').val();
    var finish = $('#finish').val();
    var num = $('#num').val();
    var state = $('#state').val();
    var ker=$('#worker').val();
    if($('#ker').val()!=""){
        ker=$('#ker').val();
    }
    $('#reset').click();
    $.ajax({
        url: 'php/add.php',
        type: 'POST',
        data:
"work="+work+"&worker="+worker+"&start="+start+"&finish="+finish+"&num="+num+"&state="+state+"",
        beforeSend: function(html) {
            $('#mes').append("<b class='add'>Запит обробляється</b>");
        },
        success: function(result){
            if(result)
            {
                start=new Date(start);
                mo=start.getMonth()+1;
                year =start.getFullYear();
                month=year+'-'+mo;
                $('#add').remove();
                $('#mes').append("<b class='add'>"+result+"</b>");
                //$('#butt').append("<input class='add' type='button'
value='Відобразити роботи "+worker+" за "+mo+" місяць "+year+" р.>");
                $('#butt').append("<input class='emal' type='button'
value='Відправити повідомлення для "+ker+" про зміни>");
                $('#kol').append("<form METHOD='POST'
action='php/vidobr.php' class='add'><input type='hidden' id='wokr' name='worker[]' value='"+worker+
"> <input type='hidden' name='month' value='"+month+
"><input type='hidden' name='work' ID='wok' value='"+work+
"><input class='add' id='vido' type='submit'
value='Відобразити роботи "+worker+" за "+mo+"-й місяць "+year+
" року'></form>");
            }
            else{
                $('#mes').append("<b class='add'>"+result+"</b>");
                $('#mes').append("<b class='add'>Виникла помилка!!!</b>");
            }
        }
    }
}

```

```

});
});
$('.butt').on("click", "input.emal",function()
{
    $('.emal').remove();
    var work = $('#wok').val();
    var worker = $('#wokr').val();
    if($('#ker').val()!=""){
        worker=$('#ker').val();
    }
    $.ajax({
        url:'php/emal.php',
        type:'POST',
        data: "work="+work+"&worker="+worker,
        beforeSend: function(html) {
            $('.butt').append("<b class='emal'>Запит обробляється</b>");
        },
        success: function(result){
            if(result)
            {
                $('.emal').remove();
                $('.butt').append("<b class='emal'>"+result+"</b>");
            }
            else{
                $('.emal').remove();
                $('.butt').append("<b class='emal'>Сталася помилка!!!</b>");
            }
        }
    });
});

var ot="", timer=0;
function tuoclick()
{
    getObj('worker').focus();
    getObj('info').style.display = 'none';
}
function PressKey2(e){ // вызывается при нажатии клавиши в select
    e=e||window.event;
    t=(window.event) ? window.event.srcElement : e.currentTarget; // объект для которого вызывно
    if(e.keyCode==13){ // Enter
        getObj('worker').focus();
        getObj('info').style.display = 'none';
    }
}
function PressKey(e){
    e=e||window.event;
    t=(window.event) ? window.event.srcElement : e.currentTarget; // объект для которого вызвано
    g=getObj('info');
    if(e.keyCode==40){g.focus();
        g.selectedIndex=0;return;}
    if(ot==t.value)return; // если ничего не изменилось не "замучить" сервер
    ot=t.value;

    if(ot.length<1){
        getObj('info').style.display='none'; // спрячем select
        return;}
    timer=window.setTimeout('Load()',50); // загружаю через 1 секунду после последнего нажатия
    клавиши
}

```

```

function Load(){
    timer=0;
    o=getObj('info');
    o.options.length=0;
    ajaxLoad('info', 'php/worker.php?city_name='+ot,"<option disabled>Не вірний ввід</option>\n",");
    o.style.display='block';
}
function getObj(objID)
{if (document.getElementById) {return document.getElementById(objID);}
else if (document.all) {return document.all[objID];}
else if (document.layers) {return document.layers[objID];}
}
function ajaxLoad(obj,url,defMessage,callback){
    var ajaxObj;
    if (defMessage) document.getElementById(obj).innerHTML=defMessage;
    if(window.XMLHttpRequest){
        ajaxObj = new XMLHttpRequest();
    } else if(window.ActiveXObject){
        ajaxObj = new ActiveXObject("Microsoft.XMLHTTP");
    } else {
        return;
    }
    ajaxObj.open ('GET', url);
    if (ajaxObj.setRequestHeader)
        ajaxObj.setRequestHeader("Content-Type", "application/x-www-form-urlencoded; charset=utf-8");

    ajaxObj.onreadystatechange = ajaxCallBack(obj,ajaxObj,(callback?callback:null));
    ajaxObj.send();
    return false;
}
function updateObj(obj, data, bold, blink){
    if(bold)data=data.bold();
    if(blink)data=data.blink();
    if(data.length>1) document.getElementById(obj).innerHTML = data; // упрощений
    вариант, работает не во всех браузерах
}
function ajaxCallBack(obj, ajaxObj, callback){
    return function(){
        if(ajaxObj.readyState == 4){
            if(callback) if(!callback(obj,ajaxObj))return;
            if (ajaxObj.status==200)
                updateObj(obj, ajaxObj.responseText);
            else updateObj(obj, ajaxObj.status+' '+ajaxObj.statusText,1,1);
        }
    }
}
worker.onblur = function (){
    sw=document.getElementsByName("lol")[0];
    sw.disabled=true;
    nusp.innerHTML = "Введіть показник знову!!!";
    nusp.style = "color:red;";
    var s=getObj('worker');
    var el=getObj('worker_span');
    s=s.value;
    if(s=="")
    {
        el.innerHTML = "Оберіть варіант зі списку";
        el.style = "color:red;";
        (function(){return this;})();woker = undefined;
    }
    else if(getObj('info').style.display == 'block')
    {
        el.innerHTML = "Оберіть варіант зі списку";
    }
}

```

```

        el.style = "color:red;";
        (function(){return this;})();woker = undefined;
    }
else if(s=='Не вірний ввід')
{
    el.innerHTML = "Оберіть варіант зі списку";
    el.style = "color:red;";
    (function(){return this;})();woker = undefined;
}
else{
el.innerHTML = "*";
el.style = "color:green;";
(function(){return this;})();woker = s; });
start.onblur = function ifdata()
{
    sw=document.getElementsByName("lol")[0];
    sw.disabled=true;
    nusp.innerHTML = "Введіть показник знову!!!";
    nusp.style = "color:red;";
    var start=getObj('start').value;
    var st=getObj('start_span');
    var finish=getObj('finish').value;
    var fin=getObj('finish_span');
    if(start && finish)
    {
        start=new Date(start);
        finish=new Date(finish);
        if(start.getMonth()==finish.getMonth()&&
start.getFullYear()==finish.getFullYear())
        {
            if(start>finish)
            {
                st.innerHTML = "Дата початку пізніше дати кінця";
                st.style = "color:red;";
                fin.innerHTML = "Дата початку пізніше дати кінця";
                fin.style = "color:red;";
                (function(){return this;})();yearmo =undefined;
            }
            else{
                st.innerHTML = "*";
                st.style = "color:green;";
                fin.innerHTML = "*";
                fin.style = "color:green;";
                mo=start.getMonth()+1;
                (function(){return this;})();yearmo =start.getFullYear()+'-'+mo; }
            }
        }
        else
        {
            st.innerHTML = "Робота задається на певний місяць року";
            st.style = "color:red;";
            fin.innerHTML = "Робота задається на певний місяць року";
            fin.style = "color:red;";
            (function(){return this;})();yearmo =undefined;
        }
    }
}
else
{
    return;
}
};
finish.onblur = function ifdata()
{

```



```

sw=document.getElementsByName("lol")[0];
sw.disabled=true;
var nusp=getObj('num_span');
nusp.innerHTML = "Введіть показник знову!!!";
nusp.style = "color:red;";
var start=getObj('start').value;
var st=getObj('start_span');
var finish=getObj('finish').value;
var fin=getObj('finish_span');
if(start && finish)
{
    start=new Date(start);
    finish=new Date(finish);
if(start.getMonth()==finish.getMonth()&&
start.getFullYear()==finish.getFullYear())
{
    if(start>finish)
    {
        st.innerHTML = "Дата початку пізніше дати кінця";
        st.style = "color:red;";
        fin.innerHTML = "Дата початку пізніше дати кінця";
        fin.style = "color:red;";
        (function(){return this;})();yearmo =undefined;
    }
    else
    {
        st.innerHTML = "*";
        st.style = "color:green;";
        fin.innerHTML = "*";
        fin.style = "color:green;";

        mo=start.getMonth()+1;
        (function(){return this;})();yearmo =start.getFullYear()+'-'+mo;
    }
}
else
{
    st.innerHTML = "Робота задається на певний місяць року";
    st.style = "color:red;";
    fin.innerHTML = "Робота задається на певний місяць року";
    fin.style = "color:red;";
    (function(){return this;})();yearmo = undefined;
}
}
else
{
    return;
}
};
num.onfocus= function (){
sw=document.getElementsByName("lol")[0];
sw.disabled=true;
var nu=getObj('num').value;
var nusp=getObj('num_span');
if(typeof(yearmo) !== "undefined"&&typeof(woker) !== "undefined")
{
    $.ajax({
        type: "POST",
        url: "php/pepsent.php",
        data: "month="+yearmo+"&woker="+woker+"",
        //dataType:"json",
        success: function(result){

```

```

        var sum=result;

        sum=100-sum;
        nusp.innerHTML = "Максимальне значення показника для даного
працівника в обраному місяці: "+sum+"%!!!";
        nusp.style = "color:green;";
        (function(){return this;})();sum =sum;
    }
});
}
nusp.innerHTML = "Заповніть поля працівника та дати!!!";
nusp.style = "color:red;";
(function(){return this;})();sum =-1;
};
num.onblur= function (){
sw=document.getElementsByName("lol")[0];
sw.disabled=true;
var nu=getObj('num').value;
var nusp=getObj('num_span');
if(sum==1)
{
    nusp.innerHTML = "Введіть знову всі значення";
    nusp.style = "color:red;";
    return;
}
if(nu>sum||nu=="|| nu<0)
{
    nusp.innerHTML = "Максимальне значення показника для даного
працівника в обраному місяці: "+sum+"%!!!";
    nusp.style = "color:red;";
    return;
}
else
{
    sw.disabled=false;
    nusp.innerHTML = "*";
    nusp.style = "color:green;";
}
}
};
var nusp=getObj('num_span');
nusp.innerHTML = "Заповніть поля працівника та дати!!!";
nusp.style = "color:red;";
function clin()
{
    var clas = document.getElementsByClassName('sp');
    for(var i=0; i<clas.length; i++)
    {
        clas[i].innerHTML="";
    }
    sw=document.getElementsByName("lol")[0];
    sw.disabled=true;
    yearmo = undefined;
    woker = undefined;
    sum = -1;
}
$('form').focusin(function(){
$('.add').remove();
$('.emal').remove();
});
});

```

## Додаток Н

## Файл izm\_work.php

```

<?session_start();
if(!isset($_SESSION['ful_name']))
{
    header('Location:practic.php');
}??>
<?php if (isset($_POST))
    {
        $ID=$_POST['ID'];
        $link = mysqli_connect("localhost", "root", "", "practic")
        or die("Ошибка " . mysqli_error($link));
        mysqli_query ($link,"set names 'utf8'");
        $query="SELECT      work.name_work,   worker.ful_name,   work.persent,
work.state_work, work.date_start, work.date_finish
        FROM work JOIN worker ON `work`.`worker_ID` = `worker`.`ID`
        WHERE work.ID=".$ID;
        $result = mysqli_query($link,$query);
        $row = mysqli_fetch_array($result);
        $name_work=$row['name_work'];
        $worker=$row['ful_name'];
        $persent=$row['persent'];
        $state_work=$row['state_work'];
        $date_finish=$row['date_finish'];
        $date_start=$row['date_start'];
    }
    ?>

<html>
<head>
    <title>Змінити</title>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
    <link rel="stylesheet" href=" ../css/practic.css">
    <link rel="stylesheet" href=" ../css/add_work.css">
    <script src="https://code.jquery.com/jquery-1.10.2.js"></script>
</head>
<body>
<div ID="content">
    <div id="headerMain">
        <header>
            <div class="log">
                
                <B class="rev">Сумиобленерго</B>
                <a class="toggle" href="logaut.php" ><?echo $_SESSION['ful_name'];?> вийти
                </a>
            </div>
            <center>
                <ul class="manuul">
                    <li class='manuli'>
                        <a class="manua" href=" ../practic.php" >Головна</a></li>
                        <li class="manuli"><a class="manua" href=" ../vibor.php"
>Відобразити</a></li>
                        <li class="selected manuli">
                            <a class="manua" href=" ../add_work.php" >Змінити роботу
                            <b class="caret"></b></a>
                        </li></ul>
                    </center>
                </header>
            </div>

```

```

<div class="kol">
  <form METHOD="POST" action="" >
    <table align='center'>
      <tr>
        <td class='labl'><label class='fio' >Найменування
роботи(заходу)</label></td>
        <td class='inputi' colspan=2><textarea class='forma' type='text'
ID='work' name='work' maxlength=250
        ></textarea></td>
      </tr>
      <tr>
        <td class='labl'><label class='fio' for="worker">Прізвище
відповідального працівника</label> </td>
        <? if($_SESSION['role_ID']==1||$_SESSION['role_ID']==3
):?>
        <td class='inputi'><input type='text' ID='worker'
name="worker" onkeyup='PressKey(event)' />
        <select id="info" size=5
        onchange="getObj('worker').value=this.options[this.selectedIndex].value; "
        onkeyup="PressKey2(event); getObj('worker').value=this.options[this.selectedIndex].value;"
        onclick="tuoclick();
getObj('worker').value=this.options[this.selectedIndex].value; ">
        </select></td>
        <? else:??>
        <td class='inputi'><input type='text' ID='worker'
name="worker" readonly/>
        </td>
        <?endif:??>
        <td><span Class="sp" ID='worker_span' style="color:red;
font-size:14px"></span>
        </td>
      </tr>
      <tr>
        <td class='labl'><label class='fio'>Дата початку
роботи</label> </td>
        <td class='inputi'><input class='forma' id="start" type='date'
name='start' ></td>
        <td><span Class="sp" id="start_span" style="color:red; font-
size:14px"></span>
        </td>
      </tr>
      <tr>
        <td class='labl'><label class='fio' >Дата завершення роботи</label> </td>
        <td class='inputi'><input class='forma' id="finish" type='date' name='finish'
        </td>
        <td><span Class="sp" id="finish_span" style="color:red; font-
size:14px"></span>
        </td>
      </tr>
      <tr>
        <td class='labl'><label class='fio'>Показник,%</label> </td>
        <td class='inputi'><input type="number" size="2" ID="num" name="num"
min="1" max="100" ></td>
        <td><span Class="sp" ID="num_span" style="color:red; font-size:14px">
        </span>
        </td>
      </tr>
      <tr>
        <td class='labl'><label class='fio'>Статус роботи</label> </td>
        <td class='inputi'><select ID="state" name="state" >
        <option value="Не виконано" selected >Не виконано</option>

```

```

        <option value="Виконується">Виконується</option>
        <option value="Виконано">Виконано</option>
    </select>
    </td>
    <td>
    </tr>
</table>
<input id="hidd" type='hidden' value='<?echo $ID;?>'>
<p style='text-align:center;'> <input name="lol" type="button" class='lol' disabled=true
value="Змінити">
</form>
<p id='mes' style='text-align:center;'></p>
<p class='butt' ></p>
<?if($_SESSION['role_ID']==2)
{
    $role=3;
    $serv=$_SESSION['service_ID'];
    $ker_query="SELECT ful_name FROM worker WHERE role_ID='$role' AND
service_ID='$serv'";
    $link = mysqli_connect("localhost", "root", "", "practic")
or die("Ошибка " . mysqli_error($link));
    mysqli_query ($link,"set names 'utf8'");
    $ker_mas=mysqli_query($link,$ker_query);
    $row_ker = mysqli_fetch_array($ker_mas);
    $ker=$row_ker['ful_name'];
    echo "<input type='hidden' id='ker' value='".$ker."'>";
}
Else{
    echo "<input type='hidden' id='ker' value='/'>";
}??>
</div>
</body>
<script type="text/javascript" src="../javascript/izm.js">
</script>
<script type="text/javascript" >
var wk='<?echo $name_work;?>';
var wrkr='<?echo $worker;?>';
var date_st='<?echo $date_start;?>';
var date_fin='<?echo $date_finish;?>';
var pers='<?echo $persent;?>';
var stat='<?echo $state_work;?>';
$('#work').val(wk);
$('#worker').val(wrkr);
$('#start').val(date_st);
$('#finish').val(date_fin);
$('#num').val(pers);
$('#state').val(stat);
if($('#worker').is('[readonly]'))
{
    var s=getObj('worker');
    var el=getObj('worker_span');
    s=s.value;
    el.innerHTML = "*";
    el.style = "color:green;";

    (function(){return this;})();.woker = s;

}
</script>
</html>

```

## Додаток О

## Файл add\_worker.php

```

<?php session_start();
    require "bd_con.php";
    if(!isset($_SESSION['ful_name']))
    {
        header('Location:../practic.php');
    }
if($_SESSION['role_ID']!=1)
{
    header('Location:../practic.php');
}

    if(isset($_POST['add']))
    {
        $errors=array();
        if(trim($_POST['worker'])=="")
        {
            $errors[]='Введіть логін користувача!';
        }
        if($_POST['password']=="")
        {
            $errors[]="Введіть пароль!";
        }
        if(trim($_POST['email'])=="")
        {
            $errors[]="Укажіть пошту!";
        }
        if(empty($errors))
        {
            $qw="SELECT * FROM worker WHERE ful_name
            =".$_POST['worker']. """;

            $user=R::getAll($qw);
            if(!$user)
            {

                $ful_name=$_POST['worker'];
                $email=$_POST['email'];
                $password=
password_hash($_POST['password'],PASSWORD_DEFAULT);
                $service_ID=(int)$_POST['service'];
                $role_ID=$_POST['role'];

                $query_insert="INSERT INTO worker (ID, ful_name, email,
password, service_ID, role_ID)
VALUES (NULL, '$ful_name','$email', '$password',
'$service_ID', '$role_ID)";

                $result= R::exec( $query_insert );
                if($result==0)
                {
                    $errors[]='Виникла помилка!';
                }
                else
                {
                    $errors[]='Користувач '.trim($_POST['worker']).'
                    успішно зареєстрований!';

                    unset($_POST);
                }
            }
        }
    }

```

```

        }
        else
        {
            $errors[]="Користувач з таким логіном уже існує!";
        }
        $er=array_shift($errors);
    }
    else
    {
        $er=array_shift($errors);
    }
}
?>
<html>
<head>
    <title>Відобразити</title>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
    <link rel="stylesheet" href="../css/add_worker.css">

</head>
<body>
<div ID="content">
    <div id="headerMain">
        <header>
            <div class="log">
                
                <B class="rev">Сумиобленерго</B>

                <a class="toggle" href="logout.php" ><?echo $_SESSION['ful_name'];?> вийти
                </a>

            </div>
            <center>
            <ul class="manuul">
                <li class="manuli">
                    <a class="manua" href="../practic.php" >Головна</a></li>
                    <li class="manuli"><a class="manua" href="../vibor.php"
>Відобразити</a></li>
                    <li class="manuli">
                    <a class="manua" href="../add_work.php" >Додати нову роботу
                    <b class="caret"></b></a>
                    </li>
                    <li class="selected manuli">
                    <a class="manua" href="add_worker.php" >Додати працівника
                    <b class="caret"></b></a>
                    </li>
                    <li class="manuli">
                    <a class="manua" href="drop_worker.php" >Видали працівника
                    <b class="caret"></b></a>
                    </li>

                </ul>
            </center>

        </header>
    </div>
    <div class="kol">

```

```

<div class='error'><?echo $er;?></div>
<form METHOD="POST" action="add_worker.php" >

    <table align='center'>
        <tr>
            <td class='labl'><label class='fio' >Прізвище та ініціали
                працівника(логін)</label></td>
            <td class='inputi'><input type='text' ID='worker' name="worker"
                maxlength='25' value='<?=@$_POST['worker'];?>' />
            </td>
        </tr>
        <tr>
            <td class='labl'><label class='fio'>Електронна пошта працівника</label>
            <td class='inputi'><input type="email" name="email" maxlength='30'
                value='<?echo @$_POST['email'];?>'></td>
        </tr>
        <tr>
            <td class='labl'><label class='fio'>Пароль</label> </td>
            <td class='inputi'><input type="password" name="password" maxlength='16'
            ></td>
        </tr>
        <tr>
            <td class='labl'><label class='fio'>Підрозділ </label> </td>
            <td class='inputi'><select name="service">
                <option value="1" selected >СПС</option>
                <option value="2">СРП</option>
            </select>
            </td>
        </tr>
        <tr>
            <td class='labl'><label class='fio'>Роль доступу до додатку</label> </td>
            <td class='inputi'><select name="role">
                <option value="1" >Адміністратор</option>
                <option value="2" selected>Працівник</option>
                <option value="3">Керівник</option>
            </select>
            </td>
        </tr>
    </table>
    <p style='text-align:center;'> <input type="submit" name="add" value="Додати працівника">
    <input type="reset" value="Стерти"></p>
    </form>
</div>

</div>
</body>
</html>

```