

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій

«До захисту допущено»

В.о. завідувача кафедри

_____ Світлана ВАЩЕНКО

_____ 2024р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня бакалавр

зі спеціальності 122 «Комп'ютерні науки»,

освітньо-професійної програми «Інформаційні технології проектування»

на тему: «Вебдодаток ведення документації викладачем.

Модуль планування позааудиторної роботи викладача»

Здобувача групи ІТ-02 Галкіна Дмитрія Олексійовича
(шифр групи) (прізвище, ім'я, по батькові)

Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело.

(підпис)

Дмитрій ГАЛКІН
(Ім'я та ПРІЗВИЩЕ здобувача)

Керівник доцент кафедри ІТ, к. т. н., доцент, Світлана ВАЩЕНКО _____
(посада, науковий ступінь, вчене звання, ім'я та ПРІЗВИЩЕ) (підпис)

Суми-2024

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

В.о. зав. кафедри ІТ

_____ Світлана ВАЩЕНКО
«__» _____ 2024 р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ

Галкін Дмитрій Олексійович

- 1 Тема роботи** Вебдодаток ведення документації викладачем.
Модуль планування позааудиторної роботи викладача
керівник роботи Ващенко Світлана Михайлівна, к.т.н., доцент,
затверджені наказом по університету від « 13 » травня 2024 р. № 0516-VI
- 2 Строк подання студентом роботи** «26» травня 2024 р.
- 3 Вхідні дані до роботи** технічне завдання на розробку вебдодатку планування
позааудиторного навантаження науково-педагогічних працівників
- 4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)** вступ, аналіз предметної області, проектування вебдодатку,
програмна реалізація вебдодатку, розробка технічного завдання, планування
робіт.
- 5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)** контекстна діаграма у нотації IDEF0 з точки зору викладача, перший
рівень декомпозиції IDEF0-діаграми з точки зору викладача, діаграма варіантів
використання в нотації uml, діаграма варіантів використання в нотації uml,
логічна модель бази даних, фізична модель бази даних, засоби реалізації
вебдодатку, архітектура вебдодатку.

6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 20.09.2023

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Дослідження предметної області	20.09.2023-03.10.2023	
2	Планування проекту	04.10.2023-20.10.2023	
3	Розробка моделі вебдодатку	23.10.2023-15.12.2023	
4	Розробка вебдодатку	20.11.2023-16.01.2024	
5	Тестування	17.01.2024-11.03.2024	
6	Реліз вебдодатку	12.03.2024-16.03.2024	
	Загальна тривалість робіт	203 днів	

Студент

(підпис)

Дмитрій ГАЛКІН

Керівник роботи

(підпис)

к.т.н., доц. Світлана ВАЩЕНКО

АНОТАЦІЯ

Тема кваліфікаційної роботи бакалавра «Вебдодаток ведення документації викладачем. Модуль планування позааудиторної роботи викладача».

Пояснювальна записка складається зі вступу, 3 розділів, висновків, списку використаних джерел із 21 найменувань, додатків. Загальний обсяг роботи – 103 сторінки, у тому числі 58 сторінок додатків.

Актуальність роботи полягає в необхідності створення ефективного інструменту для автоматизації робочого процесу в сучасному університеті, так як викладач заповнює дуже багато документації вручну, що є доволі незручною та складною процедурою.

Метою даної роботи є розробка вебдодатка, спрямованого на автоматизацію процесу планування позааудиторної роботи викладача.

У першому розділі роботи було проведено аналіз процесів планування викладачами позааудиторної роботи. Було здійснено огляд останніх досліджень і публікацій, аналіз існуючих продуктів-аналогів і виконана постановка задачі.

У другому розділі роботи було виконано структурно-функціональне моделювання вебдодатку ведення документації викладачем, модуля планування позааудиторної роботи викладача. Було здійснено моделювання варіантів використання та проектування моделі бази даних.

В третьому розділі роботи наведено архітектуру створюваного вебдодатку, програмну реалізацію та приклади використання вебдодатку.

Результатом проведеної роботи став вебдодаток ведення документації викладачем з розробкою модуля планування позааудиторної роботи викладача.

Практичне значення роботи полягає у використанні вебдодатку для ведення документації викладачем з можливістю автоматичної генерації документації.

Ключові слова: вебдодаток, навантаження НПП, позааудиторна робота, CSS, DOCTEMPLATER, JAVASCRIPT, HTML, MYSQL, PHP.

ЗМІСТ

ВСТУП.....	5
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	7
1.1 Дослідження актуальності проблеми	7
1.2 Аналіз існуючих продуктів-аналогів	11
1.3 Постановка задачі	15
2 ПРОЕКТУВАННЯ ТА МОДЕЛЮВАННЯ ВЕБДОДАТКУ ВЕДЕННЯ ДОКУМЕНТАЦІЇ ВИКЛАДАЧЕМ. МОДУЛЬ ПЛАНУВАННЯ ПОЗААУДИТОРНОЇ РОБОТИ ВИКЛАДАЧА	18
2.1 Структурно-функціональне моделювання.....	18
2.2 Проектування інформаційної системи	20
2.3 Проектування моделі бази даних	22
3 РОЗРОБКА ВЕБДОДАТКУ ВЕДЕННЯ ДОКУМЕНТАЦІЇ ВИКЛАДАЧЕМ. МОДУЛЬ ПЛАНУВАННЯ ПОЗААУДИТОРНОЇ РОБОТИ ВИКЛАДАЧА.....	24
3.1 Архітектура вебдодатку	24
3.2 Програмна реалізація	25
3.3 Використання розроблюваного вебдодатку.....	28
3.4 Тестування вебдодатку	38
ВИСНОВКИ	42
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	43
ДОДАТОК А	46
ДОДАТОК Б.....	59
ДОДАТОК В.....	74

ВСТУП

Зважаючи на сучасні тенденції у сфері освіти та науки, важливим аспектом стає ефективне управління робочим навантаженням науково–педагогічних працівників (НПП). У сучасному вищому навчальному закладі невід’ємною частиною є впровадження інформаційних технологій для оптимізації процесів та підвищення продуктивності праці.

Важливим аспектом роботи є автоматизація процесів обліку навчального навантаження НПП. Цифровізація обліку буде важливим та необхідним кроком для підвищення ефективності роботи як окремих викладачів, так і кафедри, або університету в цілому. Оскільки ручна обробка великих об’ємів інформації є часоємним процесом.

Робота викладача закладу вищої освіти пов’язана з заповненням великої кількості різного роду документації, обсяг якої щороку збільшується. Основна проблема, з якою стикається викладач – це формат отримуваних даних, роботу з яким досить важко автоматизувати, що змушує перераховувати усі дані самостійно. Це призводить до збільшення витраченого часу та ймовірності помилки у розрахунках через людський фактор.

Тому виникає необхідність створення ефективного інструменту для оптимізації робочого процесу в сучасному університеті.

Об’єкт дослідження – процес планування викладачами позааудиторної роботи.

Предмет дослідження – програмні засоби, що автоматизують процес планування навантаження.

Мета даної роботи полягає в розробці та впровадженні вебдодатка, спрямованого на підтримку процесу планування позааудиторної роботи НПП.

Для досягнення мети проекту необхідно виконати наступні задачі:

– визначити актуальність роботи з розробки вебдодатку обліку позанавчального навантаження НПП;

- дослідити предметну область та провести аналіз існуючих аналогічних сервісів обліку;
- розробити структуру та необхідний функціонал програмного продукту, врахувавши потреби викладачів та специфіку сфери застосування;
- реалізувати структуру та необхідний функціонал програмного продукту;
- виконати тестування вебдодатку.

Вебдодаток ведення документації та планування позааудиторної роботи викладача має на меті забезпечити зручний та ефективний інтерфейс для планування навантаження викладачів, а також опцію автоматичної генерації документації. Впровадження даного проекту може значно полегшити роботу викладачів навчального закладу, дозволяючи більш ефективно розподіляти ресурси та сприяти створенню сприятливого середовища для науково–педагогічної діяльності.

Таким чином, обрана тема розробки вебдодатка планування та обліку позааудиторного навантаження НПП є актуальною та обґрунтованою, оскільки сприятиме покращенню управління навчальним процесом і підвищенню продуктивності праці в університетському середовищі.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Дослідження актуальності проблеми

Сучасні системи управління та обліку в освітніх установах визначаються неабиякими трансформаціями, породженими цифровими інноваціями, які змінюють весь процес роботи та навчання в цілому [1]. Відзначається, що комунікація та обмін інформацією стають ключовими компонентами сучасного життя, як особистого, так і професійного.

Спостереження за цифровими перетвореннями показує, що інформаційні технології вже змінили підходи до управління бізнесом, спричинили значні зміни в освітній системі та змінили різні аспекти нашого життя в цілому. Зокрема, використання інноваційно-технічних розробок в бізнесі дозволяє підвищити ефективність управління, забезпечуючи постійний обмін інформацією між учасниками та автоматизуючи більшість процесів.

Також у сучасному світі, де обсяги інформації щороку зростають, важливо мати системи, які можуть ефективно зберігати, обробляти та візуалізувати дані. Аналіз використання інформаційних систем в різних галузях підтверджує, що вони дозволяють забезпечити доступ до великого обсягу інформації, а також виконувати аналітичні завдання за мінімальний час.

На теперішній час в університеті більшість роботи, пов'язаної з формуванням позааудиторного навантаження викладача, досі виконується самими працівниками та з мінімальною кількістю автоматизації процесів, а саме за допомогою засобів Excel. При цьому викладачеві постійно потрібно мати поруч інформацію про навантаження в розрізі видів позааудиторної роботи. Тому для можливості автоматизації цього процесу було проведено дане дослідження.

Дослідимо процес панування позааудиторної роботи НПП СумДУ.

Робочий час НПП визначається Кодексом законів про працю України (ст. 51, 52) [2], Законом України «Про вищу освіту» (ст. 56) [3] та Кодексом цивільного захисту України (ст. 128, 129) [4]. Робочий час НПП становить 36 годин на тиждень (скорочена тривалість робочого часу) на одну повну ставку НПП. Згідно наказу СумДУ «Про завершення навчальної роботи у весняному семестрі 2022/23 н. р. та планування її на 2023/24 н. р.» (№0428-VI від 28.04.2023 року) річний обсяг робочого часу науково-педагогічного працівника становить 1548 годин.

Навчальне навантаження складає 590 годин на рік, все решта позааудиторна робота.

Обсяги методичної, наукової та організаційної роботи не включаються до навчального навантаження НПП, а є окремими видами робіт, які відносяться до позааудиторної роботи.

Норми часу для планування й обліку методичної, наукової та організаційної роботи НПП, визначені в «Положенні про планування та облік роботи науково-педагогічних працівників Сумського державного університету», яке введено в дію наказом ректора СумДУ (№ 0375-I від 12.04.2021р.) [5], і яким визначено порядок нормативного обрахування навчальної, методичної, наукової та організаційної роботи НПП. Вони визначаються відповідною кафедрою з урахуванням специфіки його діяльності, галузі знань, спеціальності, особливостей видів робіт при їх персоніфікації, особливостей закріплення контингенту здобувачів вищої освіти тощо; можуть бути диференційованими та затверджуються розпорядженнями їх керівників. Види та норми часу для планування й обліку навчальної роботи розробляються під керівництвом першого проректора та затверджуються відповідним наказом ректора, що є нормативним документом.

До методичної роботи НПП відноситься робота в якості керівника (члена) у складі експертних рад МОН, навчально-методичних комісій; робота зі створення нормативної документації по підготовці фахівців за відповідною освітньою програмою; методичне забезпечення супроводження навчального

процесу; перевірка робіт здобувачів освіти; підготовка до видання, прийняття до друку методичних матеріалів; підвищення кваліфікації; організація, супроводження самостійної роботи здобувачів та інші види методичної роботи [5].

До наукової роботи НПП відносять виконання планових наукових досліджень і розробок, публікаційну діяльність та отримання охоронних документів, наукову роботу зі здобувачами, роботу керівника (члена) у складі редколегій іноземних наукових видань або фахових видань України, атестацію наукових кадрів та інші види наукової і науково-технічної діяльності.

Також до позааудиторної роботи відноситься організаційна робота НПП, яка включає в себе організаційну роботу на посадах керівника (заступника керівника) кафедри або іншого відповідального за підготовку здобувачів вищої освіти підрозділу / лабораторії / навчально-наукового (інноваційного) структурного підрозділу / вченого секретаря ради університету / інституту / факультету, здійснення профорієнтаційної роботи для навчання або стажування в СумДУ, участь у якості члена в засіданнях вченої ради університету / інституту / факультету, участь у засіданнях/семінарах кафедри, виконання обов'язків куратора академічної групи здобувачів тощо [5].

Всі норми часу обговорюються на засіданні кафедри і зведені в таблицю, приклад якої наведений на рисунку 1.1.

Планування та облік роботи науково-педагогічних працівників (НПП) на 2023-2024 н.р.

секція ГПП кафедри КН Сумського державного університету, факультет ЕЛПТ

Згідно наказу «Про завершення навчальної роботи у весняному семестрі 2022/23 н. р. та планування її на 2023/24 н. р.» (№0428-VI від 28.04.2023 року) річний обсяг робочого часу науково-педагогічного працівника становить 1548 годин.

		Norma факультету, год	Norma по кафедрі, год
I. НАВЧАЛЬНЕ НАВАНТАЖЕННЯ НПП			
6	навчальне навантаження викладача на 1 ставку	590	590
II. МЕТОДИЧНА РОБОТА НПП			
а) робота в якості керівника (члена) у складі:			
8	експертних рад МОН, навчально-методичних комісій (підкомісій) МОН або галузевих експертних рад Національного агентства із забезпечення якості вищої освіти, їх експертних комісій;	100	100
9	робочої проєктної групи освітньої програми, групи забезпечення спеціальності, секретар ради роботодавців;	50	50
10	ради із забезпечення якості освітніх послуг та якості вищої освіти структурного підрозділу;	50	50
б) робота зі створення нормативної документації по підготовці фахівців за відповідною освітньою програмою			
11	оновлення освітньої програми, навчальних планів, вивірення семестрових навчальних планів	50	50
12	зав. кафедри / гаранту освітньої програми для перегляду робочих програм	1 година на робочу програму	1 година на робочу програму

Рисунок 1.1 – Демонстраційний приклад планування та обліку роботи НПП

Кожен викладач заповнює свою документацію вручну, що є доволі незручною та складною процедурою. При цьому найчастіше використовується табличний процесор MS Excel, оскільки дозволяє використовуючи інструментарій формул автоматизувати розрахункові процедури. Проте роботи по формуванню документу по шаблону та внесенню текстової інформації виконується викладачем. В підсумку, створюється файл, зразок якого показано на рисунку 1.2.

II. МЕТОДИЧНА РОБОТА					
№ з/п	Вид роботи (за необхідності – кількість)	Норма часу, год	Всього, год		Пояснення про виконання/невиконання
			план	факт	
1	керівник робочої проектної групи освітньої програми, член групи забезпечення спеціальності	50	50		
2	член ради із забезпечення якості освітніх послуг та якості вищої освіти факультету ЕлПТ	50	50		
3	оновлення освітньої програми, навчальних планів, вивчення семестрових навчальних планів	50	50		
4	перегляд робочих програм як гарант освітньої програми	1 год / програму	30		
5	оновлення робочих програм навчальних дисциплін	20	60		
6	підготовка до проведення лекційних занять	0,5 год / 1 год лекції	60		
7	підготовка до проведення лабораторних занять	0,5 год / 1 заняття	100		
8	підготовка до проведення курсового та дипломного проектування	1 год/ 1 год роботи	60		
9	підготовка до проведення атестаційних заходів	3 год /захід	30		
10	перевірка звітів з лабораторних робіт	0,5 год/робота	200		
11	перевірка курсових робіт та ОДЗ	2 год/робота	100		
12	звіти з практик керівникам від кафедри	1 год/ робота	20		
13	проведення консультацій з навчальних дисциплін	50 год / рік	50		
Всього по р. II					860

Рисунок 1.2 – Приклад формування звіту по плануванню позааудиторної роботи

Як видно з аналізу, ця робота є рутинною та трудозатратною, а тому потребує автоматизації.

1.2 Аналіз існуючих продуктів-аналогів

В межах кафедри було проведено дослідження та виявлено, що програмного засобу, що б комплексно вирішував задачу автоматизації процесу планування та обліку позааудиторного навантаження науково-педагогічних працівників кафедри не має.

На сьогодні вже існують різні системи обліку заданої предметної області, але вибір цих інформаційних систем зовсім невеликий, а їх можливості та направленість не зовсім відповідають поставленим вимогам.

Після проведення детального аналізу було виявлено дві групи подібних програм: перша – системи обліку часу, друга – системи освітньої підтримки.

До першої групи відносяться такі додатки, як TimeCamp, Toggl, Yaware, а також система Hours, що була розроблена студентом КПІ ім. І. Сікорського.

TimeCamp – це надійний додаток, створений для будь-яких сфер діяльності та який надає великий перелік можливостей для кожного користувача програмою [6]. Особливостями програми TimeCamp є його можливість автоматизувати процеси, які пов’язані з керуванням проектами та їх задачами, а також TimeCamp є інструментом для точного та автоматичного підрахування часу витраченого на виконання задач. Інтерфейс цієї програми дуже зручний, він продемонстрований нижче на рисунку 1.3.

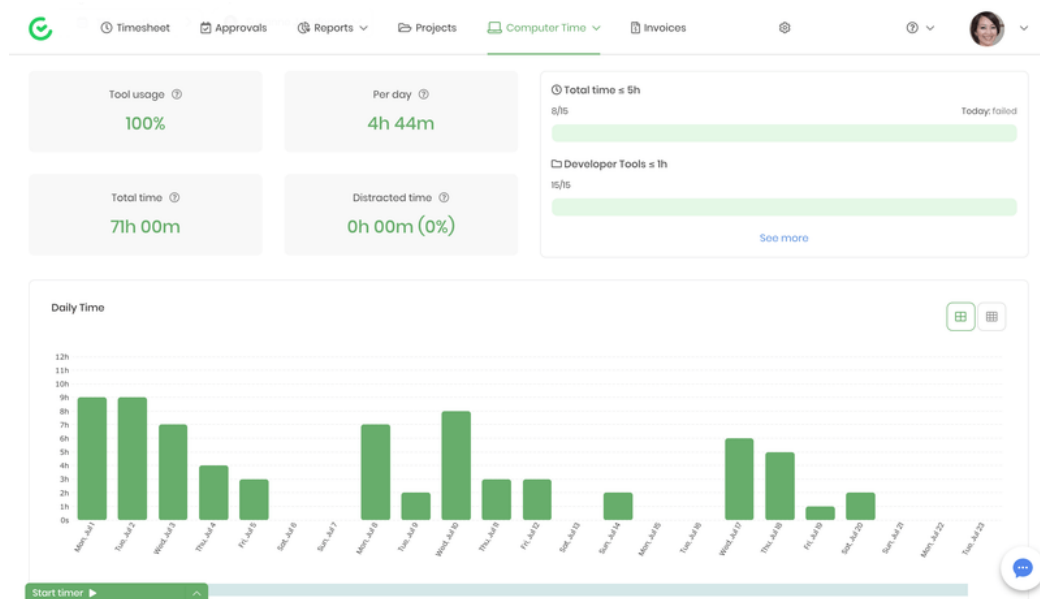


Рисунок 1.3 – Демонстраційний приклад додатку TimeCamp

Також існує ще одна система обліку робочого часу – **Yaware** [7]. Ця система обліку є автоматичною для відслідковування та оцінювання ефективності працівників. Вона також має дуже зручний інтерфейс, який наведено на рисунку 1.4. Це програмне забезпечення відрізняється тим, що встановлюється всередині компанії при зберіганні даних тільки на корпоративних серверах. Переваги цього програмного додатку полягають в

дотриманні безпеки даних та повна ізоляція їх на сервері з можливістю доступу тільки для співробітників. Ця система дозволяє оптимізувати роботу та допомагає ефективно керувати компанією. При цьому затрати часу є мінімальними. Ще однією перевагою програми є те, що вона надає можливість проводити звітність за різними параметрами.

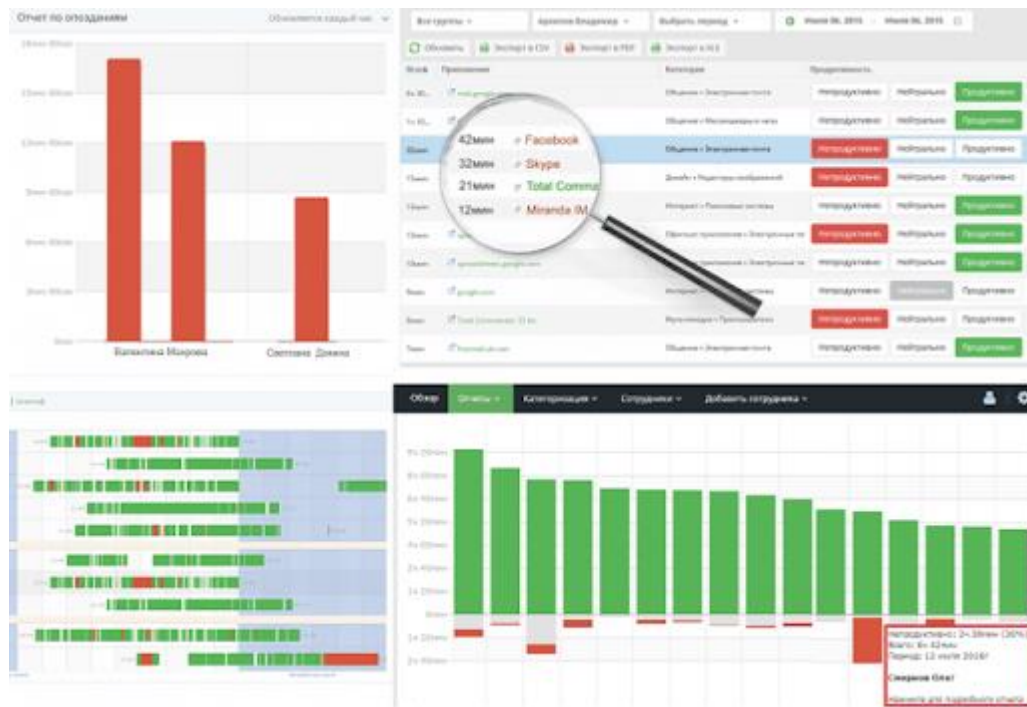


Рисунок 1.4 – Демонстраційний приклад додатку Yaware

Для відстеження робочого часу існує система **Toggl**, яка надає можливість аналізу дій кожного робочого дня та самостійного редагування робочого часу при виникненні невідповідностей [8]. Також, функціонал програми дозволяє фіксувати перегляд та відвідування сайтів на яких було витрачено більше 10 секунд. Це надає можливість аналізувати сайти з найбільшою та найменшою кількістю відвідувань. Інтерфейс додатку наведено на рисунку 1.5. Особливістю системи є використання хмарного сервісу, що дає можливість швидко синхронізувати в режимі реального часу записи годин та спільно працювати над документами та захищати їх [8]. Цей додаток відрізняється швидким розгортанням інфраструктури.

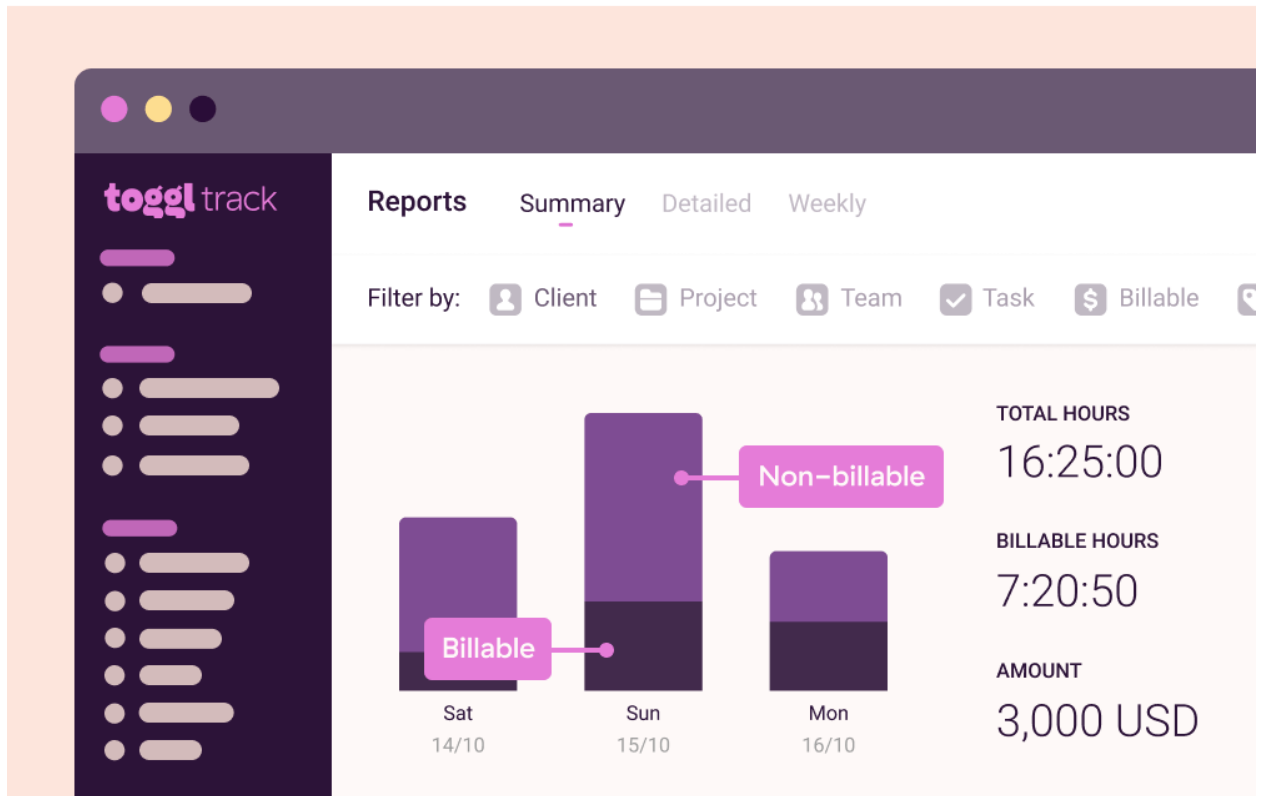


Рисунок 1.5 – Демонстраційний приклад додатку Toggl

Для обліку та контролю робочого часу працівників існує додаток **Hours**, наведений на рисунку 1.5 [9]. Він має простий зручний інтерфейс та надає можливість встановлювати розумні нагадування, робити звіти, дозволяє обирати різні версії: Professional, Team або Personal, змінювати потрібні функції.



Рисунок 1.5 – Демонстраційний приклад додатку Hours

Після розгляду і аналізу існуючих програмних рішень можна сказати, що вони не дозволяють вирішити всі поставлені задачі. Бачимо їх направленість на певне вирішення однієї задачі або декілька задач, але аж ніяк не на цілий комплекс рішень, які б дозволяли автоматизувати процес планування позааудиторного навантаження НПП. Також недоліком вищевказаних програм є те, що деякі з них є платними.

Тому розробка вебдодатку, який автоматизує планування та облік позааудиторного навантаження НПП, стає актуальною в контексті цифрового середовища, яке швидко змінюється.

1.3 Постановка задачі

Мета цієї дипломної роботи полягає в розробці та впровадженні вебдодатку, спрямованого на автоматизацію процесу планування позааудиторного навантаження НПП у Сумському державному університеті.

У результаті роботи розроблено та імплементовано вебдодаток, спрямований на вирішення виявлених проблем та задоволення конкретних потреб управлінців та науково–педагогічного персоналу. Відповідно до цього, можна очікувати полегшення планування та обліку позааудиторного навантаження, спрощення процедури формування звітності, а також підвищення загальної ефективності управління ресурсами вищого навчального закладу.

Проаналізувавши потреби до персоналу закладу освіти було визначено наступні функціональні вимоги. Користувальницька частина додатку повинна забезпечувати виконання наступних функцій:

- редагування особистої інформації;
- перегляд особистої інформації;
- перегляд архівних відомостей навантаження НПП за попередні навчальні роки;

- редагування відомостей навантаження НПП;
- створення нових відомостей навантаження НПП;
- генерація документів за відомостями;

Адміністраторська панель додатку повинна забезпечувати виконання наступних функцій:

- зміна норм;
- редагування, додавання і видалення типу та виду навантаження, роду та виду діяльності;
- пошук, редагування та видалення інформації про НПП;
- додавання нових працівників.

Структура вебдодатку, види забезпечення та функціонування системи докладно описані в технічному завданні на розробку проекту (додаток А).

Щоб створити цей застосунок були використані наступні технології: мова розмітки HTML для розробки каркасу вебсторінок, таблиці стилів CSS з використанням технології flexbox для додавання візуальних ефектів і забезпечення адаптивності, а також мова JavaScript для забезпечення динамічності та асинхронної взаємодії з сервером. Для реалізації бази даних програмного продукту була обрана СУБД MySQL.

HTML, CSS, JavaScript – це три базові технології, на яких працюють вебзастосунки: HTML – каркас, CSS – візуальне оформлення, а JavaScript – логіка, інтерактивність та взаємодія з користувачем [10].

Зокрема, CSS було використано для налаштування шрифтів і кольорів, відступів та інших візуальних ефектів відображення блоків сторінок вебдодатку [11].

CSS надає значний перелік інструментів і є більш точний та опрацьований. Так як візуальне оформлення вебдодатку повинно за стилістикою відповідати сайту кафедри інформаційних технологій [12], то при його розробці було прийнято рішення використовувати CSS для більш ретельного налаштування дизайну елементів сторінок.

Крім того CSS підтримується більшою частиною сучасних веббраузерів, що надає більші можливості та перспективи подальшого використання застосунку [11].

JavaScript є високорівневою мовою програмування, яка підтримує різні підходи: імперативний, функціональний та подієво-орієнтований. Вона має динамічну типізацію і використовується для написання "скриптів" або "сценаріїв" – послідовностей операцій, які, на відміну від компільованих мов, інтерпретуються безпосередньо, що усуває потребу в додаткових інструментах для перетворення коду [10].

Завдяки JavaScript, частина обробки веб-сторінок може відбуватися на стороні клієнта, без звернення до сервера. Це економить час, знижує трафік та зменшує навантаження на сервер. JavaScript значно покращує взаємодію з користувачем: заповнення форм, вибір опцій, активація кнопок, перевірка введених даних і реакція на події, такі як наведення курсора чи кліки миші.

JavaScript відзначається динамічністю, гнучкістю у роботі з функціями та універсальністю. Він підтримується всіма сучасними браузерами, легко інтегрується з HTML і дозволяє ефективно організувати комунікацію з сервером.

SQL - дає змогу отримувати точні дані зі значних і складних баз даних, направлений на великий обсяг оброблюваної інформації. У ньому, в першу чергу, увага приділяється тому, які дані викликати, видаляти чи вставляти, а не яким чином відбувається даний процес [13]. У реляційній базі даних дані зберігаються не відразу повністю, а, насамперед, у вигляді окремих таблиць, через які досягається активізація швидкості й гнучкості. Під час запити існує можливість поєднувати дані з деякого числа таблиць. Ще велика кількість доведених до готовності методів роботи з нею під час розробки вебдодатку обумовлює прийняття рішення щодо застосування власне MySQL.

2 ПРОЕКТУВАННЯ ТА МОДЕЛЮВАННЯ МОДУЛЯ ПЛАНУВАННЯ ПОЗААУДИТОРНОЇ РОБОТИ ВИКЛАДАЧА

2.1 Структурно-функціональне моделювання

Для того щоб успішно виконати проект, об'єкт проектування повинен бути, перш за все, правильно й адекватно описаний, тобто необхідно побудувати повноцінні та функціональні інформаційні моделі об'єкта проектування [14].

IDEF0 представляє собою підхід до моделювання. Як правило, моделювання засобами IDEF0 є першим етапом вивчення будь-якої системи [15].

Моделювання починається з побудови контекстної діаграми, у нашому випадку, в нотації IDEF0.

Початок моделювання процесів у нотації IDEF0 відзначається створенням, так би мовити, контекстної діаграми, що описує діяльність організації чи процесу загалом. Контекстна діаграма демонструє найбільш важливі входи та виходи, механізми, що є потрібними для роботи, а ще керуючі дії [15].

Моделювання процесів у нотації IDEF0 починається зі контекстної діаграми, яка має описувати діяльність організації або процесу в цілому. На контекстній діаграмі відображаються найважливіші входи і виходи, механізми, необхідні для роботи, а також керуючі дії [15].

Контекстну діаграму IDEF0 вебдодатку ведення документації викладачем, модуль планування позааудиторної роботи викладача представлено на рисунку 2.1.



Рисунок 2.1 - Контекстна діаграма у нотації IDEF0 з точки зору викладача

Із складеної контекстної діаграми видно, що для проведення планування позааудиторної роботи використовується інформація про навчальне навантаження та персональна інформація про викладача, а на виході отримуємо створений план викладача з різної позааудиторної роботи та звіт про навантаження. Користувач, вебдодаток, прикладне програмне та апаратне забезпечення виступають механізмами. А методами управління процесами є інструкція по використанню вебдодатка та погодинні норми, які затверджені в «Положенні про планування та облік роботи науково-педагогічних працівників Сумського державного університету» [5].

Далі після створення контекстної діаграми виконується її декомпозиція. Нижче наведено діаграму декомпозиції першого рівня, яка складається з наступних підпроцесів: створення нового звіту, додавання нових видів навантаження, заповнення звіту, формування звіту. Таким чином, аналізувати всі елементи процесів стане легше. Декомпозиція IDEF0 з точки зору викладача представлена на рисунку 2.2.

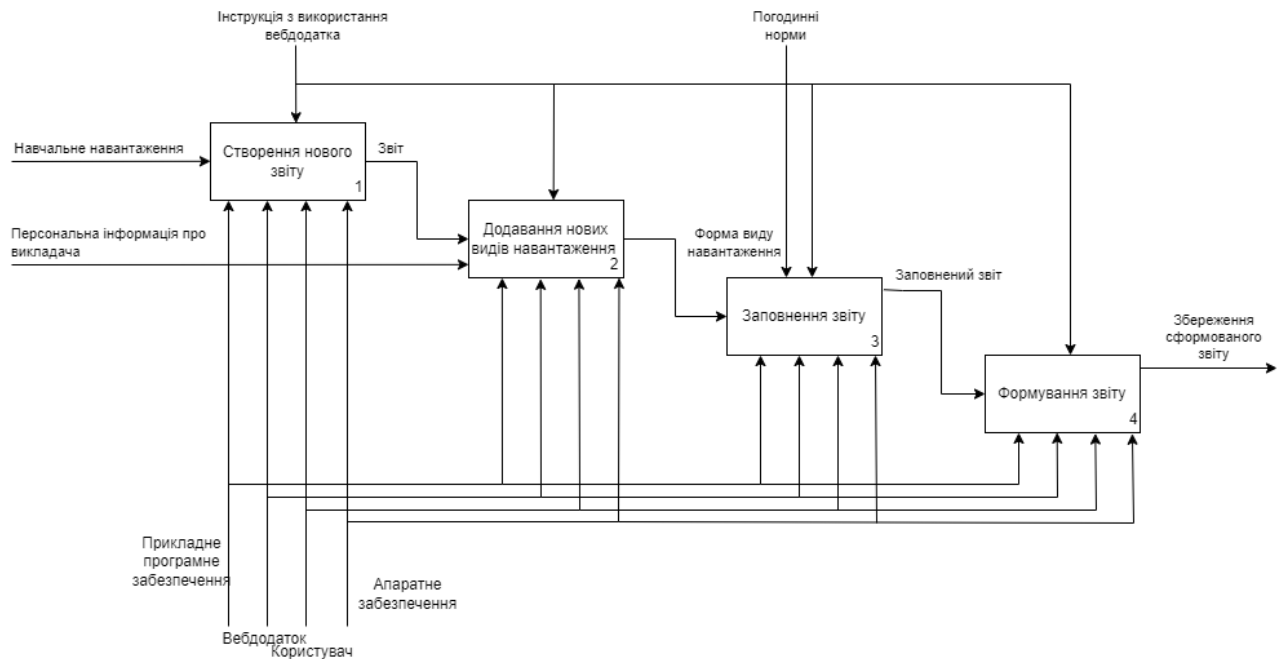


Рисунок 2.2 - Перший рівень декомпозиції IDEF0-діаграми з точки зору викладача

2.2 Проектування інформаційної системи

Для демонстрації базових сутностей інформаційної системи було обрано уніфіковану мову UML. UML створено на основі інженерних розробок, які тепер застосовуються для моделювання складних масштабних систем [16].

Опис взаємозв'язків та залежності між групою варіантів використання й акторами, що беруть участь у процесі, належить діаграмам варіантів використання.

Треба акцентувати, що дані діаграми не призначені для демонстрації компонування, адже вони не здатні описувати внутрішню будову системи. Такі діаграми потрібні для спрощення процесу обміну інформацією між майбутніми користувачами системи та замовником. Діаграми варіантів використання є дуже корисними для визначення списку можливостей, що мусять бути у системи. За цими діаграмами можна визначити, що саме системі слід робити, але не те, як,

власне, вона отримує потрібні результати, бо для крайнього пункту ці діаграми є не придатними [17].

Діаграма варіантів використанні демонструє можливі сценарії користування додатком для акторів (викладача, БД та адміністратора [17].

Діаграму ВВ модуля планування позааудиторної роботи викладача представлено на рисунку 2.3.

У викладача є можливість взаємодії з такими варіантами використання: створення нових звітів про навантаження, перегляд всіх звітів про навантаження, формування звітів про навантаження та їх завантаження.

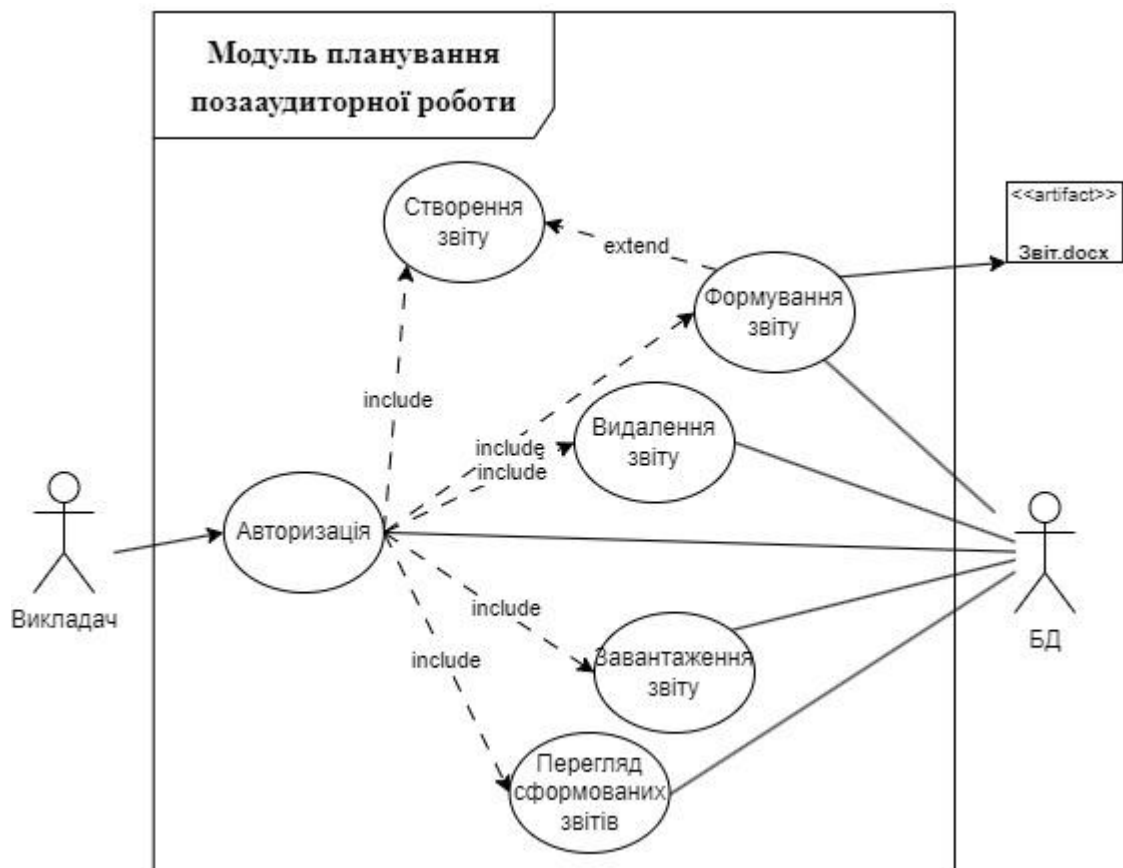


Рисунок 2.3 – Діаграма варіантів використання в нотації UML викладача

На рисунку 2.4 представлено діаграму варіантів використання адміністратора, що описує наступні дії, можливі до виконання адміністратором, після проходження авторизації, а саме: додавання типів і видів навантаження,

видів занять, зміна норм навантаження, додавання та видалення викладачів, редагування персональної інформації.

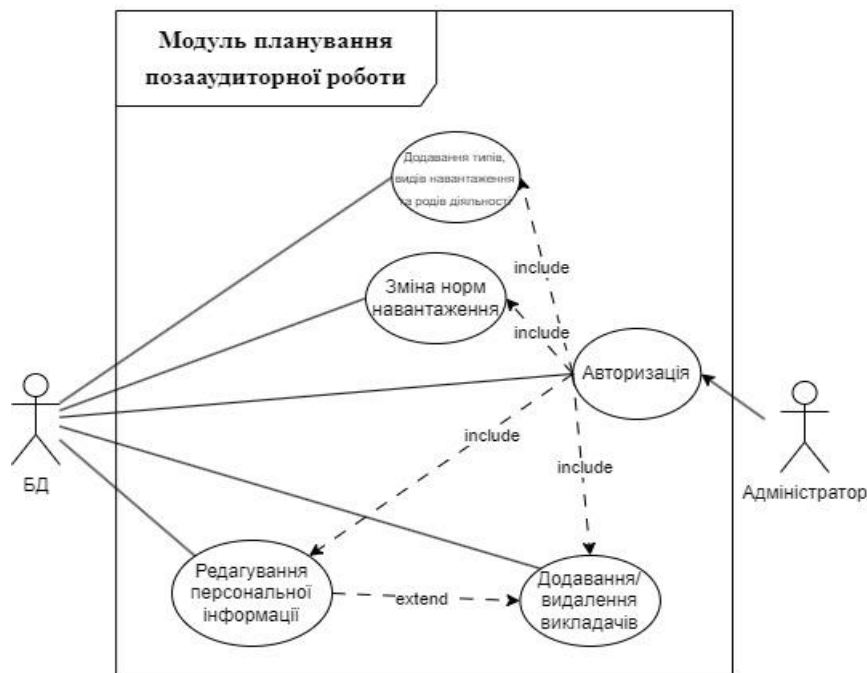


Рисунок 2.4 – Діаграма варіантів використання в нотації UML адміністратора

2.3 Проектування моделі бази даних

Проектування бази даних є дуже важливим етапом розробки проєкту.

База даних (БД) – це будь-яка пов’язана між собою за певними ознаками інформація, що зберігається і організовується певним чином, як правило у вигляді таблиць [18].

В результаті здійснення проектування отримуємо логічну модель бази даних вебдодатку ведення документації викладачем, яка наведена на рисунку 2.5 (таблиці, що використовуються в роботі модуля планування позааудиторної роботи виділено синім кольором). З неї ми можемо бачити кілька основних частин діаграми, кожна з яких має своє призначення.

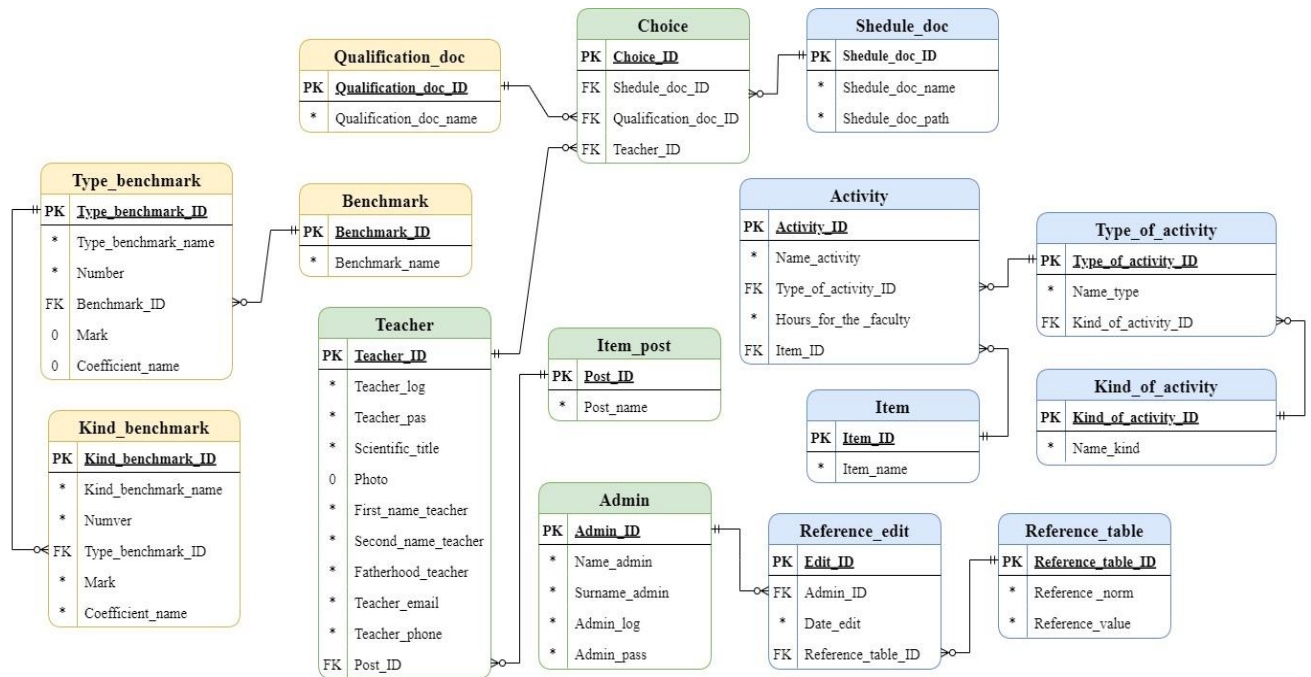


Рисунок 2.5 – Логічна модель бази даних

Розглянемо основні таблиці: таблиця *Teacher*, яка містить інформацію про викладачів, їх персональні дані, посаду, логін і пароль для авторизації та інше. таблиці *Kind_of_activity*, *Type_of_activity*, *Activity* і *Item* мають інформацію про тип і вид навантаження, рід діяльності та вид занять.

Таблиця *Shedule_doc* слугує для збереження згенерованих звітів користувачів. Таблиця *Admin* містить логіни і паролі для авторизації адміністраторів, таблиця *Reference_table* містить погодинні норми навантаження, а в таблиці *Reference_edit* фіксує зміни норм навантаження.

3 РОЗРОБКА МОДУЛЯ ПЛАНУВАННЯ ПОЗААУДИТОРНОЇ РОБОТИ ВИКЛАДАЧА

3.1 Архітектура вебдодатку

Для розробки даного вебдодатку було обрано стандартну клієнт-серверну архітектуру.

Клієнт – це особа або організація, яка використовується як послуга. У контексті ІТ клієнт – це комп'ютер/пристрій, також званий хостом, який фактично використовує службу або приймає інформацію. Клієнтські пристрої включають ноутбуки, робочі станції, пристрої ІоТ та подібні пристрої, які підтримують мережу [19].

Сервер у світі ІТ – це віддалений комп'ютер, який забезпечує доступ до даних і послуг. Сервери, як правило, є фізичними пристроями, такими як стійкові сервери, хоча розвиток хмарних обчислень призвів до рівняння віртуальних серверів. Сервер обробляє такі процеси, як електронна пошта, розміщення програм, підключення до Інтернету, друк тощо [19].

Клієнт-серверна архітектура є одним з найпопулярніших рішень, яке добре підходить для роботи у вебзастосунках.

Архітектура клієнт-сервер, що ще має назву модель клієнт-сервер, — це мережева програма, що робить розподіл завдань та навантаження між клієнтами й серверами, що перебувають в одній системі або ж об'єднані комп'ютерною мережею [19].

В розроблюваному вебдодатку ведення документації викладачем користувач взаємодіє з вебдодатком за допомогою веббраузера, в якому відкритий вебінтерфейс додатка. Далі запит іде на обробку до сервера, звідки вже потрапляє на обробку до БД. Також на сервері знаходиться модуль генерації звітів. Архітектура вебдодатку представлена на рисунку 3.1.

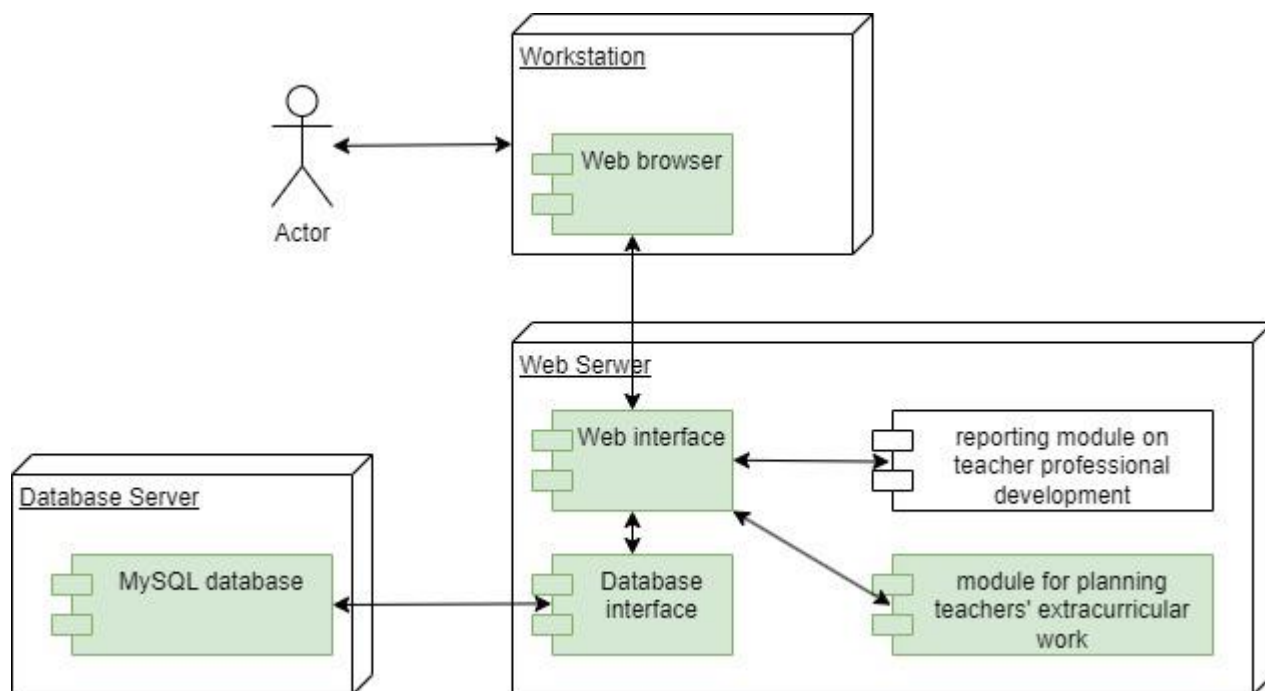


Рисунок 3.1 – Архітектура вебдодатку

3.2 Програмна реалізація

Програмна реалізація вебдодатку включає в себе фізичне розгортання бази даних, верстку сторінок вебсайту та розробка логіки його роботи.

Як зазначалося вище для реалізації даного вебдодатку було обрано наступні технології: як мову гіпертекстової розмітки, каскадні таблиці стилів CSS з технологією flexbox та мову програмування JavaScript. Для реалізації бази даних розроблюваного програмного продукту було обрано СУБД MySQL.

Оглянемо модель БД, яку представлено на рисунку 3.2.

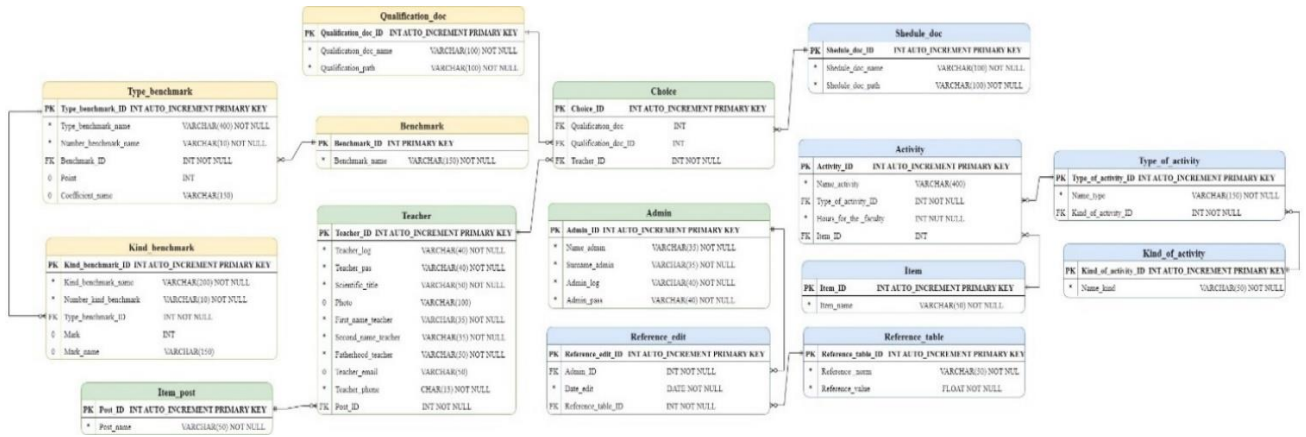


Рисунок 3.2 – Фізична модель БД

Дана модель бази даних представляє собою класи, в яких атрибути мають свої власні типи даних, відповідно до БД MySQL. На рисунку 3.3 наведено вигляд БД після її розгортання на сервері. База даних має 15 таблиць відповідних сутностей.

Kind benchmark	Activity	☆	🔍	🔗	🔧	🔒	🔑	3	InnoDB	utf8mb4_unicode_ci	48.0 КиБ	-
Kind of activity	Admin	☆	🔍	🔗	🔧	🔒	🔑	2	InnoDB	utf8mb4_unicode_ci	16.0 КиБ	-
Qualification doc	Benchmark	☆	🔍	🔗	🔧	🔒	🔑	5	InnoDB	utf8mb4_unicode_ci	16.0 КиБ	-
Reference edit	Choice	☆	🔍	🔗	🔧	🔒	🔑	0	InnoDB	utf8mb4_unicode_ci	64.0 КиБ	-
Reference table	Item	☆	🔍	🔗	🔧	🔒	🔑	4	InnoDB	utf8mb4_unicode_ci	16.0 КиБ	-
Schedule	Item post	☆	🔍	🔗	🔧	🔒	🔑	3	InnoDB	utf8mb4_unicode_ci	16.0 КиБ	-
Teacher	Kind benchmark	☆	🔍	🔗	🔧	🔒	🔑	7	InnoDB	utf8mb4_unicode_ci	32.0 КиБ	-
Type benchmark	Kind of activity	☆	🔍	🔗	🔧	🔒	🔑	4	InnoDB	utf8mb4_unicode_ci	16.0 КиБ	-
Type of activity	Qualification doc	☆	🔍	🔗	🔧	🔒	🔑	0	InnoDB	utf8mb4_unicode_ci	16.0 КиБ	-
diploma2	Reference edit	☆	🔍	🔗	🔧	🔒	🔑	0	InnoDB	utf8mb4_unicode_ci	48.0 КиБ	-
Новая	Reference table	☆	🔍	🔗	🔧	🔒	🔑	0	InnoDB	utf8mb4_unicode_ci	16.0 КиБ	-
Activity	Schedule	☆	🔍	🔗	🔧	🔒	🔑	0	InnoDB	utf8mb4_unicode_ci	32.0 КиБ	-
Admin	Teacher	☆	🔍	🔗	🔧	🔒	🔑	1	InnoDB	utf8mb4_unicode_ci	32.0 КиБ	-
Benchmark	Type benchmark	☆	🔍	🔗	🔧	🔒	🔑	19	InnoDB	utf8mb4_unicode_ci	32.0 КиБ	-
Choice	Type of activity	☆	🔍	🔗	🔧	🔒	🔑	2	InnoDB	utf8mb4_unicode_ci	32.0 КиБ	-
Item												
Item post												
Kind benchmark												
Kind of activity												
		15 таблиц	Всего	50 InnoDB utf8mb4_unicode_ci		432.0 КиБ	0 Байт					

Рисунок 3.3 – Список таблиц БД на сервері

При розробці сторінок було розроблено відповідні шаблони верхньої і нижньої частини сторінки, приклади яких наведено нижче на рисунках 3.4 - 3.5.

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8" />
5 <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6 <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7 <link
8   rel="stylesheet"
9   href="https://fonts.googleapis.com/css2?family=Manrope:wght@200;300;400;500;600;700;800&display=swap"
10 />
11 <link rel="icon" href="../img/logo.png" type="image/x-icon" />
12 <link type="text/css" href="style/vid_KP.css" rel="stylesheet" />
13 <title>Адміністративна панель</title>
14 </head>
15
16 <body>
17 <div class="all">
18 <!-- Хедер -->
19 <header class="header">
20 <div class="img"></div>
21 <p>Кафедра інформаційних технологій</p>
22 </header>

```

Рисунок 3.4 – Вид шапки сайту

```

139 <!-- Футер -->
140 <footer class="footer">
141 <div class="location">
142 <p class="imgloc">
143 
144 </p>
145 <p class="adres">
146 вул. Харківська, 116, Суми, Україна, 40007 Головний корпус, 13-й
147 поверх, ауд. Г1303
148 </p>
149 </div>
150
151 <div class="phone">
152 <div class="phone1">
153 <p class="img">
154 
155 </p>
156 <p class="pr">приймальня +38 (0542) 68-78-54</p>
157 </div>
158 <div class="phone2">
159 <p class="img">
160 
161 </p>
162 <p class="pr">завідувач кафедри +38 (0542) 68-78-63</p>
163 </div>
164 </div>
165
166 <div class="email">
167 <div class="insta">
168 <a href="https://www.instagram.com/itp_sumdu?igshid=fopu9dqxdigb">
169 
174 <span>Instagram</span>
175 </a>
176 </div>
177 <div class="em">
178 <p class="img2"></p>
179 <p class="adres">info@itp.sumdu.edu.ua</p>
180 </div>
181 </div>
182 </footer>
183 </div>
184 </body>
185 </html>
186

```

Рисунок 3.5 – Вид підвалу сайту

Програмний код розробленої частини вебдодатку ведення документації викладачем наведено в додатку В.

3.3 Використання розроблюваного вебдодатку

Додаток виконаний у корпоративних стилях кафедри ІТП, відповідно до ТЗ. Усі сторінки мають зручну навігацію, виконану за допомогою меню на боковій панелі.

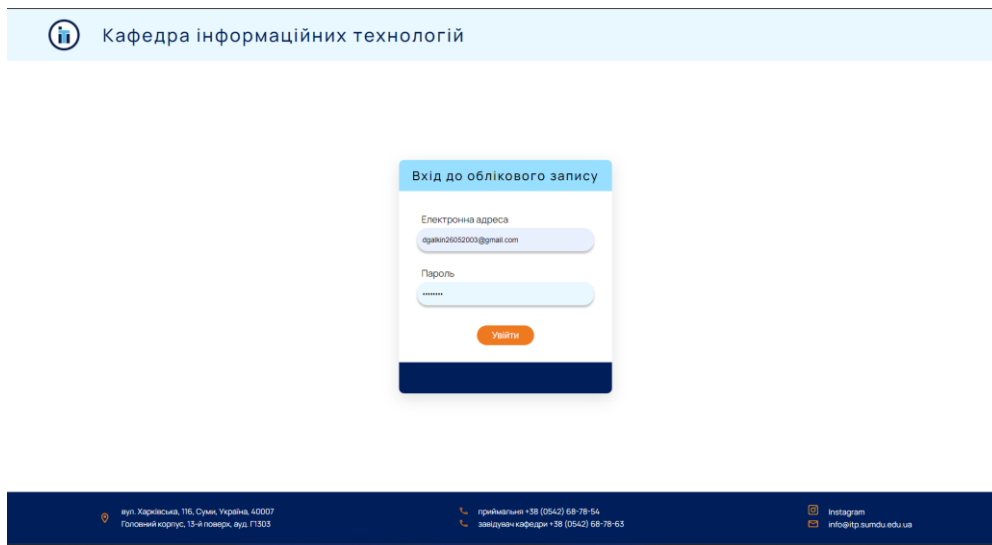


Рисунок 3.6 – Вікно авторизації сайту

Першою сторінкою нашого додатка є сторінка авторизації. В залежності, від введених користувачем логіна та пароля користувач потрапляє до адмінпанелі, або в частину користувача.

Розглянемо користувальницьку частину додатку. Після проходження авторизації викладач потрапляє на сторінку вибору, за допомогою якої може обрати який саме звіт потрібно сформувати.

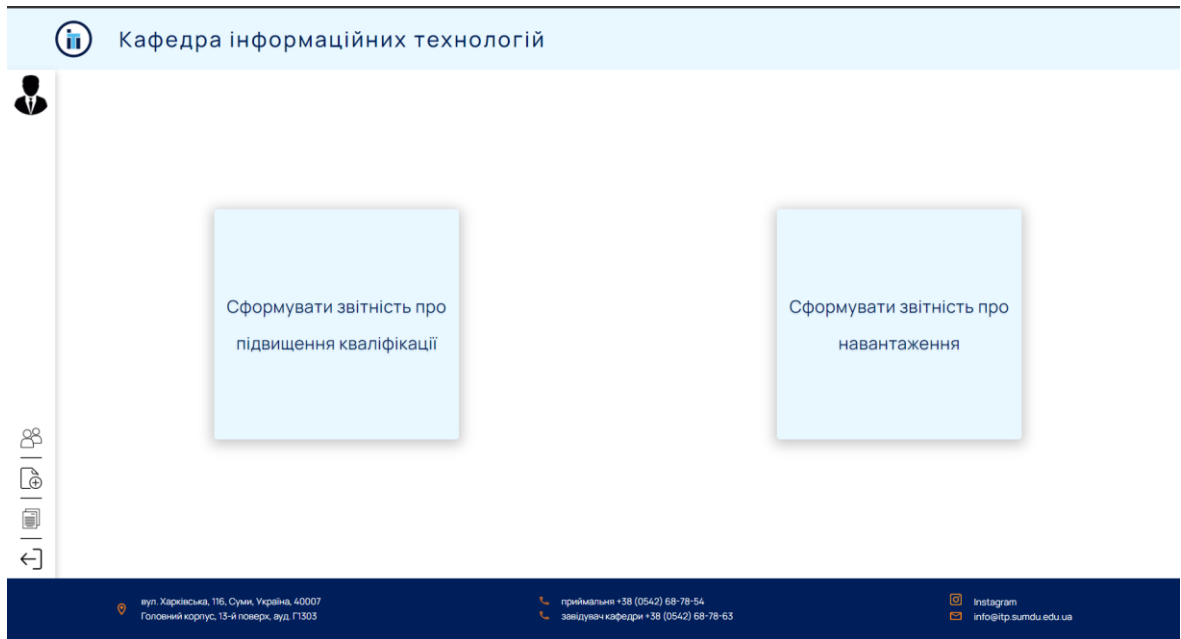


Рисунок 3.7 – Сторінка вибору

Після обрання відповідної кнопки на сторінці вибору користувач потрапляє на сторінку формування звіту за навантаженням. Після чого він має обрати відповідні дати обліку документу та частку ставки.

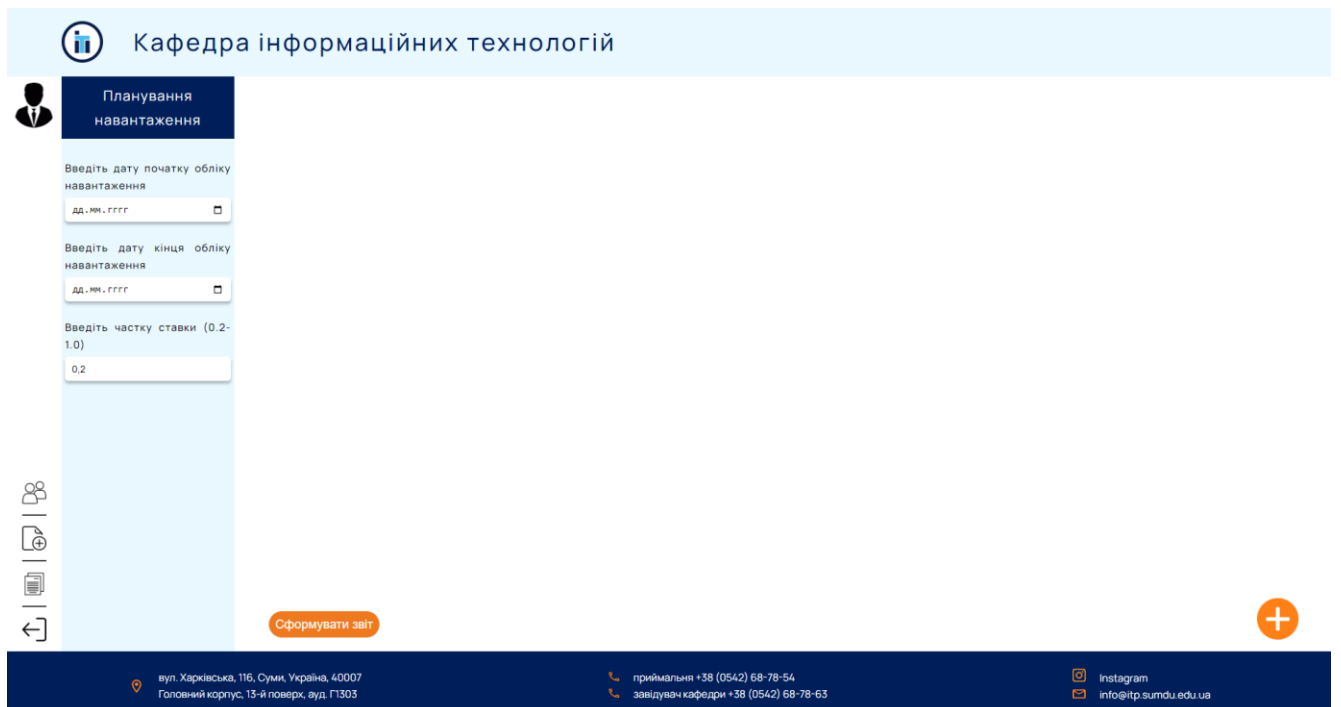


Рисунок 3.8 – Сторінка формування звіту

Рисунок 3.9 – Заповнення основних відомостей про звіт

Далі за допомогою додавання користувач бачить модальне вікно з можливістю вибору необхідного виду навантаження.

Рисунок 3.10 – Модальне вікно вибору виду навантаження

Після підтвердження дії в модальному вікні користувач отримує можливість обрати тип навантаження та рід діяльності з випадаючих списків, а також за потреби ввести вид занять та кількість годин. За потреби можна додати будь яку кількість нових форм, які потім сформуєть таблицю згенерованого звіту.

Кафедра інформаційних технологій

Планування навантаження

2. МЕТОДИЧНА РОБОТА НПП

Введіть дату початку обліку навантаження: 02.05.2024

Введіть дату кінця обліку навантаження: 30.05.2024

Введіть частку ставки (0.2-1.0): 0.2

Тип навантаження: Робота в якості керівника (члена) у складі

Рід діяльності: експертних рад МОН, навчально-методичних комісій (підкомісії) МОН або галузевих експертних рад Національного агентства із забезпечення якості вищої освіти, їх експертні комісії, експертних рад МОН, навчально-методичних комісій (підкомісії) МОН або галузевих експертних рад Національного агентства із забезпечення якості вищої освіти, їх експертних комісій, робочої проєктної групи освітньої програми, групи забезпечення спеціальності, секретар ради роботодавців, лекційні

Вид занять: лекційні

Кількість годин по кафедрі: 20

Сформувати звіт

ул. Харківська, 116, Суми, Україна, 40007
Головний корпус, 13-й поверх, ауд. П1303

приміщення +38 (0542) 68-78-54
завідувач кафедри +38 (0542) 68-78-63

Instagram
info@itp.sumdu.edu.ua

Рисунок 3.11 – Заповнення форми

Після заповнення всіх необхідних норм, користувач має можливість сформувати звіт, після чого він отримує .docx файл зформованого документу

Кафедра інформаційних технологій

Планування навантаження

2. МЕТОДИЧНА РОБОТА НПП

Введіть дату початку обліку навантаження: 10.05.2024

Введіть дату кінця обліку навантаження: 17.05.2024

Введіть частку ставки (0.2-1.0): 0.2

Тип навантаження: Робота в якості керівника (члена) у складі

Рід діяльності: експертних рад МОН, навчально-методичних комісій (підкомісії) МОН або галузевих експертних рад Національного агентства із забезпечення якості вищої освіти, їх експертних комісій, експертних рад МОН, навчально-методичних комісій (підкомісії) МОН або галузевих експертних рад Національного агентства із забезпечення якості вищої освіти, їх експертних комісій, робочої проєктної групи освітньої програми, групи забезпечення спеціальності, секретар ради роботодавців

Вид занять: Введіть вид занять

Норма годин по кафедрі: 10

Сформувати звіт

Рисунок 3.12 – Заповнений звіт

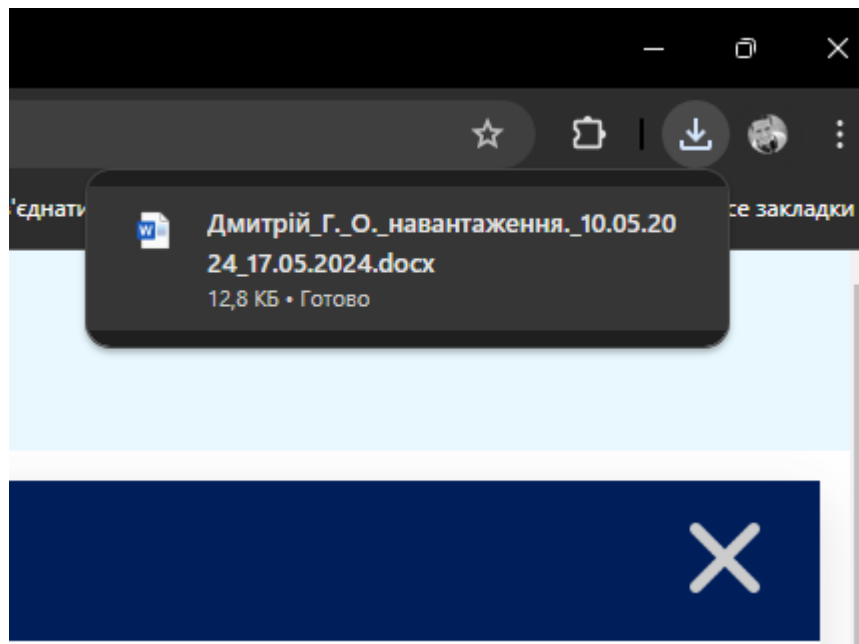


Рисунок 3.13 – Завантаження звіту

Файл сформованого звіту за навантаженням складається із заголовку із вказанням дат, посади та П.І.Б викладача, та ставки. Основна частина файлу має таблицю з інформацією про навантаження. Файл має наступний вигляд, представлений на рисунку 3.14.

Планування та облік роботи на 10.05.2024-17.05.2024н.р.
секція ІТІ кафебри КН Сумського державного університету, факультет

ЕЛІТ

Професор, кафедра інформаційних технологій

Дмитрій Галкін Олексійович

Згідно наказу «Про завершення навчальної роботи у весняному семестрі
2022/23 н. р. та планування її на 2023/24 н. р.» (№0428-VI від 28.04.2023 року)

Ставка викладача: 0.2

<u>2. МЕТОДИЧНА РОБОТА НПП</u>		
<u>Робота в якості керівника (члена) у складі:</u>		
<u>експертних рад МОН,</u> <u>навчально-методичних</u> <u>комісій (підкомісій)</u> <u>МОН або галузевих</u> <u>експертних рад</u> <u>Національного</u> <u>агентства із</u> <u>забезпечення якості</u> <u>вищої освіти, їх</u> <u>експертних комісій</u>	<u>лекційні</u>	20
<u>2. МЕТОДИЧНА РОБОТА НПП</u>		
<u>Робота в якості керівника (члена) у складі:</u>		
<u>робочої проєктної</u> <u>групи освітньої</u> <u>програми, групи</u> <u>забезпечення</u> <u>спеціальності,</u>		10

Рисунок 3.14 – Приклад сформованого звіту

Також користувач має можливість перегляду раніше сформованих звітів, за допомогою вікна вибору необхідно обрати тип звітів, доступних до перегляду, після чого користувач потрапляє на наступну сторінку. На даній сторінці можна переглянути весь перелік раніше сформованих звітів, видалити непотрібні, або завантажити їх повторно на ПК.

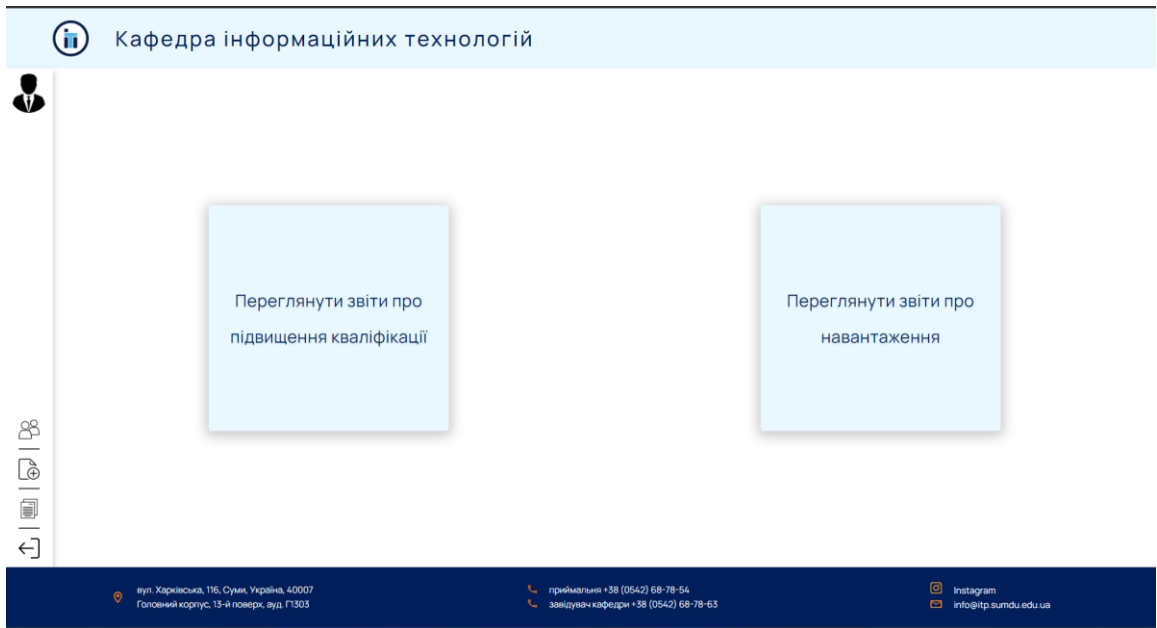


Рисунок 3.15 – Сторінки вибору



Рисунок 3.16 – Сторінка зі звітами

Далі розглянемо адміністративну панель додатку. Після авторизації адміністратор потрапляє на сторінку пошуку викладачів, де бачить увесь перелік викладачів в алфавітному порядку, що є на даний момент в БД. Також адміністратор має можливість пошуку викладача за ініціалами, а також функції додавання нових викладачів, редагування їх персональної інформації та видалення в разі потреби. Також адміністратор має в боковій панелі ще дві окремі кнопки меню, які відповідають за два різні розділи вебдодатку. Перша кнопка відповідає за зміну параметрів, що відповідають за формування звітів про підвищення кваліфікації, а друга за зміну параметрів навантаження НПП (рис. 3.17).

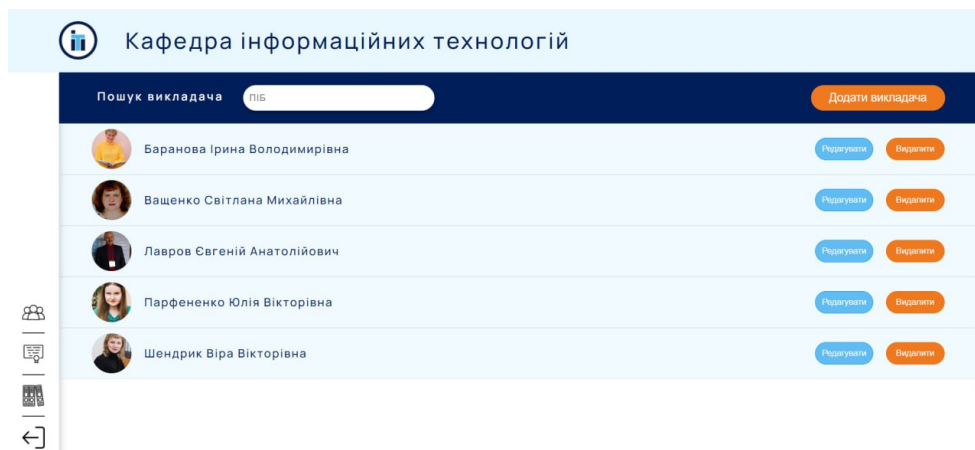


Рисунок 3.17 – Сторінка пошуку викладачів

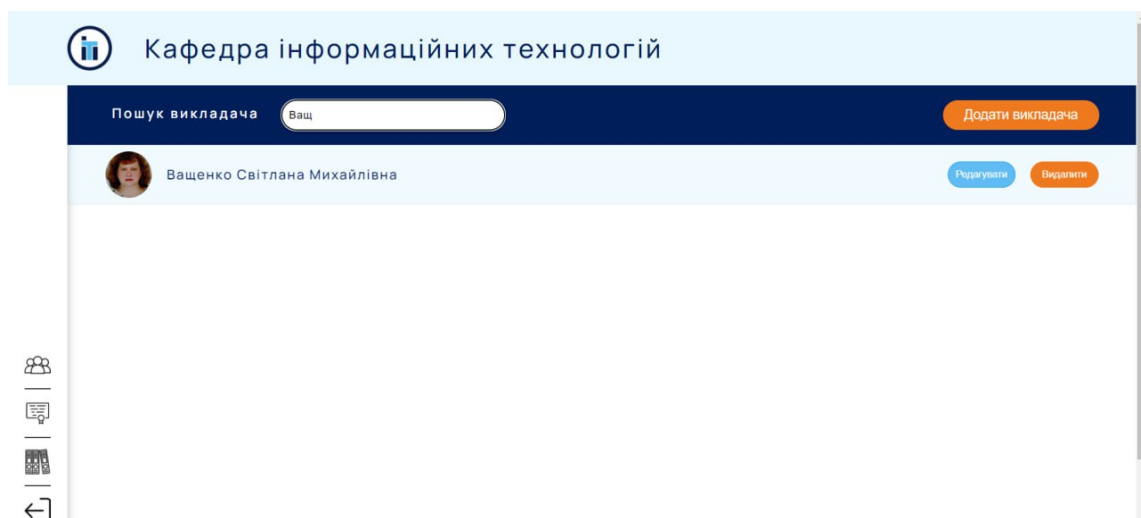


Рисунок 3.18 – Пошук викладачів

Оглянемо другий розділ більш детально. Розділ має чотири сторінки, кожна з яких відповідає за редагування конкретних параметрів. Перший за зміну, додавання та видалення видів навантаження, другий за типи навантаження, третій за рід діяльності та четвертий за редагування погодинних норм, які змінюються щороку (рисунки 3.19-3.22).

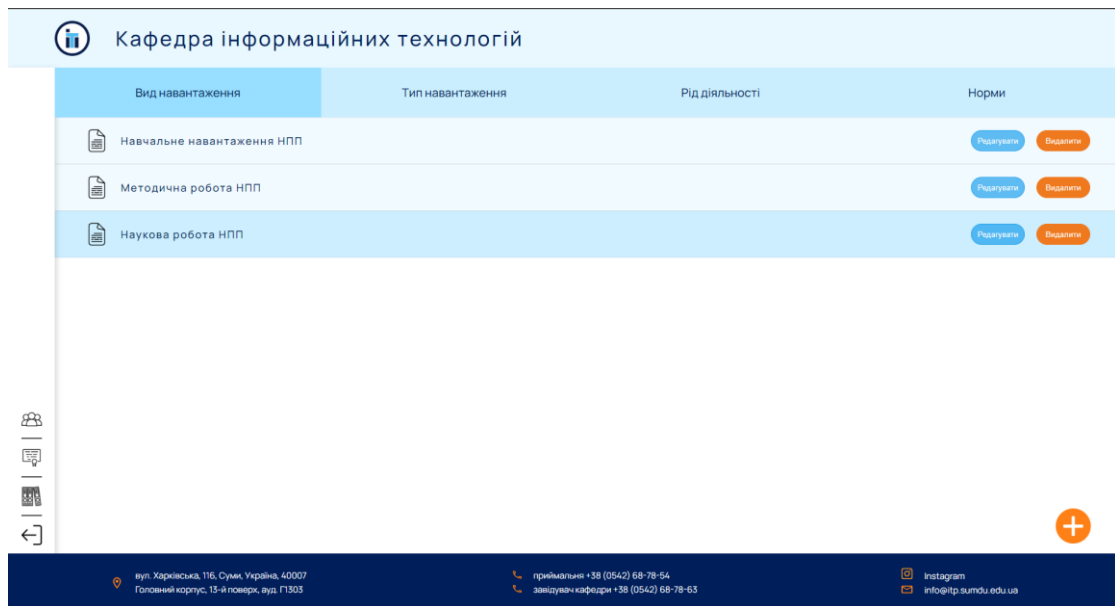


Рисунок 3.19 – Сторінка з переліком видів навантаження

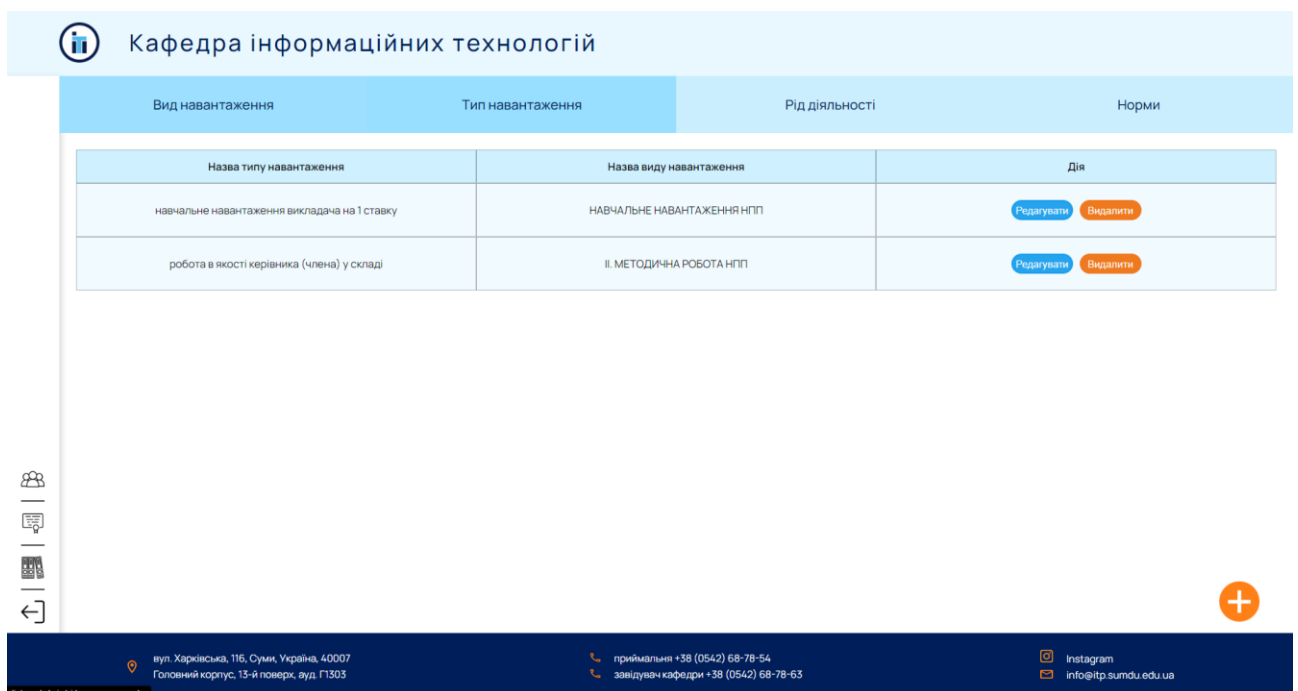


Рисунок 3.20 – Сторінка з переліком типів навантаження



Рисунок 3.21 – Сторінка з переліком родів діяльності

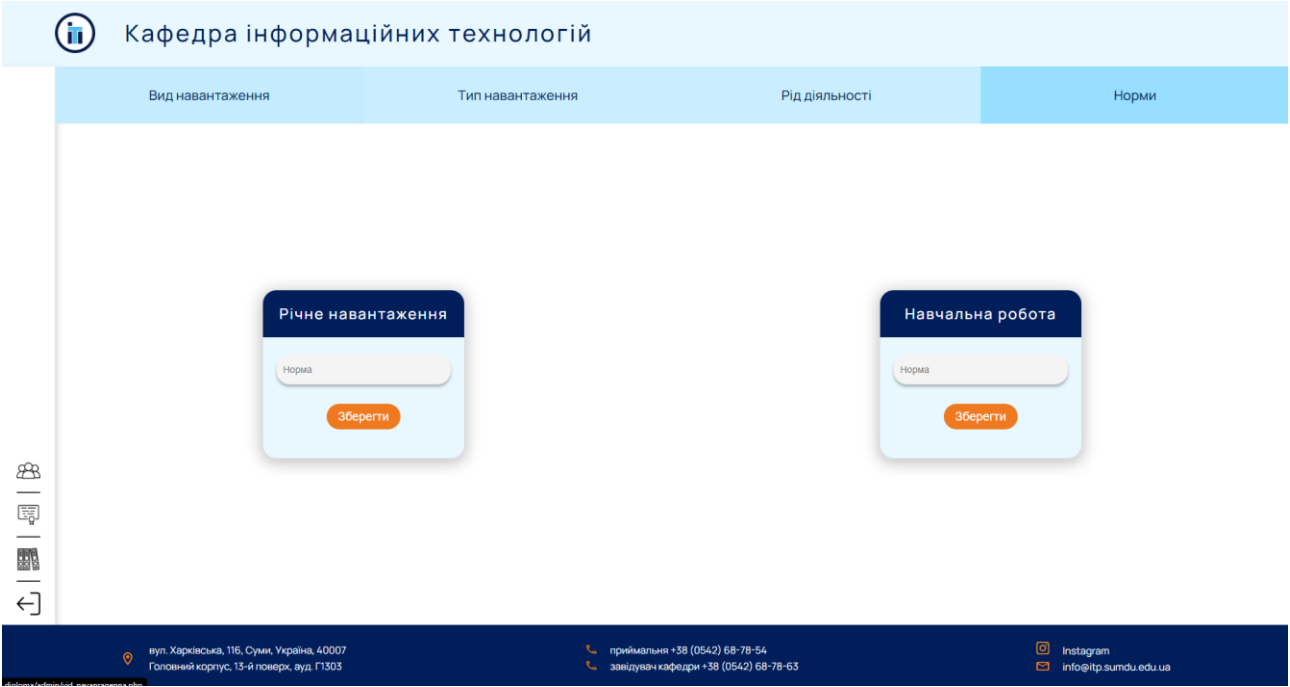


Рисунок 3.22 – Редагування норм навантаження

3.4 Тестування вебдодатку

Тестування програмного забезпечення - це процес виконання програми з метою виявлення програмних помилок. Тестування програмного забезпечення є важливою частиною розробки програмного забезпечення. Воно необхідне для виявлення помилок в програмному забезпеченні, перевірки відповідності програмного забезпечення вимогам [20].

Принципи тестування – це основні концепції та підходи, які лежать в основі процесу тестування програмного забезпечення. Одним із фундаментальних принципів тестування є раннє тестування. Цей принцип полягає в тому, що тестування має починатися якомога раніше в процесі розробки ПЗ. Замість того щоб відкладати його до кінця розробки або бета-тестування, воно інтегрується в кожен етап життєвого циклу розробки. Це дає змогу виявити та виправити проблеми на ранніх етапах, коли їх легше та дешевше виправляти [21].

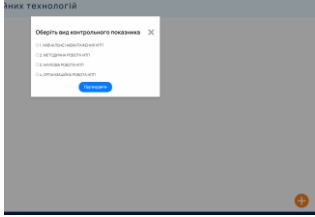
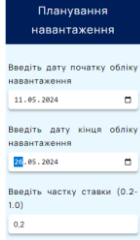

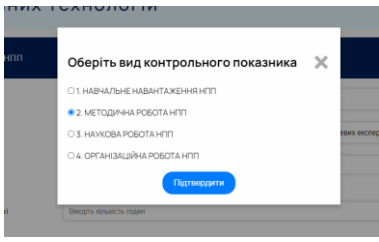
В ході розробки застосунку проводилось модульне тестування для переконання в працездатності. На кожному етапі розробки вебдодатку проводилось тестування його сторінок і модулів на стабільну роботу та відповідності до ТЗ.

Далі наведено тестування окремих важливих частин вебдодатку методом білого ящика (табл. 3.1).

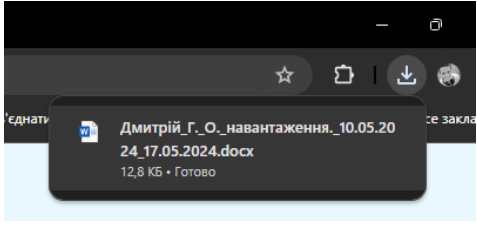
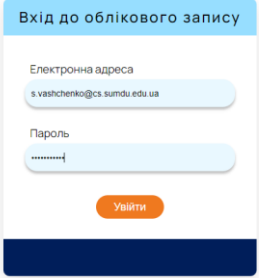

Таблиця 3.1 – Тестування методом білого ящика

№ тесту	Умова, яка тестується	Очікуваний результат	Фактичний результат	0/1
1	Клікабельність кнопки нижньої частини сайту	При натисканні на відповідну кнопку повинен здійснюватись перехід на Інстаграм сторінку кафедри ІТ.	<p>Посилання активне, при натисканні на логотипа Інстаграму відбувається перехід на відповідну сторінку.</p> 	1
2	Клікабельність бокового меню	При натисканні на кнопку відбувається перехід на відповідну сторінку сайту.	 <p>Всі кнопки бокового меню на всіх сторінках відкликаються на натискання</p>	1
3	Тестування пошуку	При введенні певних ініціалів викладача повинен відбуватись його пошук.	 <p>При введенні певних ініціалів викладача відбувається його пошук.</p>	1

Продовження табл. 3.1.

№ тесту	Умова, яка тестується	Очікуваний результат	Фактичний результат	0/1
4	Клікабельність кнопки додавання форми	Після натискання має відкриватись модальне вікно.	 <p>Після натискання відбувається відкриття модального вікна з відповідними до БД типами навантаження.</p>	1
5	Встановлення частки ставки	Повинні бути обмеження на введення від 0.1 до 1.0	 <p>Поле не дозволяє ввести інші значення, крім 0.1 - 1.0</p>	1
6	Функціонування випадаючих списків форми	При обранні відповідних видів навантаження повинні бути доступні відповідні типи, аналогічно з родом діяльності.		1
7	Робота модального вікна	Модальне вікно повинно надавати можливість вибору видів навантаження, у відповідності до БД		1

Продовження табл. 3.1.

№ тесту	Умова, яка тестується	Очікуваний результат	Фактичний результат	0/1
8	Робота кнопки генерації звіту паролю у вікні авторизації	При натисканні на кнопку повинен автоматично завантажуватись готовий документ звіту.		1
9	Робота кнопки авторизації	При умові введення правильного логіну та паролю кнопка повинна переводити на сторінку видору обліку.		1
10	Коректність роботи меню адмінпанелі	Перехід між усіма сторінками додатку	 <p>На всіх сторінках меню залишається клікабельним та дозволяє переходити між вікнами додатку</p>	1

ВИСНОВКИ

У результаті виконання дипломної роботи було розроблено вебдодаток ведення документації та планування позааудиторної роботи викладача.

Під час виконання роботи було детально досліджено процес планування викладачами позааудиторної роботи та виконано аналіз потреб НПП, визначено актуальність розробки вебдодатку, спрямованого на планування та облік позааудиторного навантаження НПП. Була сформульована постановка задачі та технічне завдання на виконання проекту.

Виконано моделювання додаку з використанням діаграм IDEF-0 та UML.

В ході розробки проекту було проведено моделювання застосунку, зокрема діаграми варіантів використання та модель БД.

Також було проведено тестування головних важливих частин додатку, що дало можливість перевірити коректність та стабільність його роботи, стікість до помикowego введення.

Розроблений програмний продукт спрямований на підтримку процесу планування позааудиторної роботи викладача та автоматизацію цього процесу з можливістю автоматичної генерації документації. Він має зручний у використанні інтерфейс та якісне візуальне оформлення, що за стилістикою відповідає сайту кафедри інформаційних технологій.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Автоматизація та комп'ютерно-інтегровані технології у виробництві та освіті: стан, досягнення, перспективи розвитку : Матеріали Всеукр. науково-практ. Інтернет-конф., м. Черкаси, 13–19 берез. 2023 р. Черкаси, 2023. С. 202. URL: https://conference.ikto.net/pub/akit_2023_13-19march.pdf (дата звернення: 30.04.2024).
2. Кодекс законів про працю України : Кодекс України від 10.12.1971 р. № 322-VIII : станом на 24 груд. 2023 р. URL: <https://zakon.rada.gov.ua/laws/show/322-08#Text> (дата звернення: 06.05.2024).
3. Кодекс цивільного захисту України : Кодекс України від 02.10.2012 р. № 5403-VI : станом на 19 квіт. 2024 р. URL: <https://zakon.rada.gov.ua/laws/show/5403-17#Text> (дата звернення: 06.05.2024).
4. Про вищу освіту : Закон України від 01.07.2014 р. № 1556-VII : станом на 24 берез. 2024 р. URL: <https://zakon.rada.gov.ua/laws/show/1556-18#Text> (дата звернення: 06.05.2024).
5. Положенні про планування та облік роботи науково-педагогічних працівників Сумського державного університету (наказ № 0375-I від 12.04.2021р.). URL: <https://normative.sumdu.edu.ua/?task=getfile&tmpl=component&id=1e824148-2e9c-eb11-9d2b-d4856459ca35&kind=1&version=1715070607911> (дата звернення: 06.05.2024).
6. Time Tracking Software - Free Automated Time Tracker TimeCamp. Time Tracking Software - Free Automated Time Tracker TimeCamp. URL: <https://www.timesamp.com/> (date of access: 04.05.2024).
7. Система обліку робочого часу в Україні - Yaware.TimeTracker. URL: <https://yaware.com.ua/> (дата звернення: 04.05.2024).

8. Toggl. Toggl Track: Time Tracking Software for Any Workflow. Toggl Track: Time Tracking Software for Any Workflow. URL: <https://toggl.com/> (date of access: 04.05.2024).
9. LLC H Hours Time Tracking. App Store. URL: <https://apps.apple.com/ua/app/hours-time-tracking/id895933956> (date of access: 04.05.2024).
10. Програмування web-сторінок засобами PHP та Javascript. Репозитарій ОНУ імені І.І.Мечникова. URL: <https://dspace.onu.edu.ua/items/dad10327-ebce-491f-b3cf-5fd9d313a7f3> (дата звернення: 07.05.2024).
11. eVNUIR: Верстка веб-сторінок з допомогою HTML та CSS : методичні рекомендації. eVNUIR: Домівка. URL: <https://evnuir.vnu.edu.ua/handle/123456789/20751> (дата звернення: 07.05.2024).
12. Сайт кафедри Інформаційних технологій СумДУ. URL: <https://itp.elit.sumdu.edu.ua/> (дата звернення: 06.05.2024).
13. SQL база даних Для чого призначена база даних?. ukraine.com.ua. URL: <https://www.ukraine.com.ua/blog/programming/sql-baza-dannih-dlya-chego-prednaznachena-baza-dannih.html> (дата звернення: 06.05.2024).
14. Мінухін С. CASE-технології: Конспект лекцій / С. В. Мінухін. – Харків: ХНЕУ, 2005. – 156 с.
15. Практичні аспекти реінжинірингу бізнес-процесів / О.Б. Данченко. – Київ: Університет економіки та права «КРОК», 2013. – 239с.
16. Олешко Т., Бахорчук В. Моделювання бізнес-процесів за допомогою діаграми випадків використання. Вчені записки ТНУ імені В. І. Вернадського. 2020. Т. 31, № 3. С. 178–184. URL: <https://doi.org/10.32838/2523-4803/70-3-64> (дата звернення: 23.05.2024).
17. Основи UML. Елементи UML. URL: <https://docs.kde.org/trunk5/uk/umbrello/umbrello/uml-elements.html#use-case-diagram> (дата звернення: 23.05.2024).

18. Харів Н. О. Бази даних та інформаційні системи: Навч. посіб. / Н. О. Харів. - Рівне, 2018. –127 с.
19. Terra J. What is client-server architecture? Everything you should know | simplilearn. Simplilearn.com. URL: <https://www.simplilearn.com/what-is-client-server-architecture-article> (date of access: 23.05.2024).
20. К. А. Паламарчук. Тестування. Моделі розробки програмного забезпечення. Конференції ВНТУ електронні наукові видання. URL: <https://conferences.vntu.edu.ua/index.php/mn/mn2020/paper/viewFile/8611/7186> (дата звернення: 25.05.2024).
21. Принципи тестування: їх концепції та підходи. FoxmindEd. URL: <https://foxminded.ua/pryntsyru-testuvannia/> (дата звернення: 25.05.2024).

ДОДАТОК А

ТЕХНІЧНЕ ЗАВДАННЯ
на розробку інформаційної системи
«Вебдодаток планування та обліку позааудиторного навантаження
науково-педагогічних працівників»

ПОГОДЖЕНО:

Доц. кафедри інформаційних технологій

_____ Ващенко С. М.

Студент групи ІТ-02

_____ Галкін Д.О.

Суми 2024

1 Призначення й мета створення програмного продукту

1.1 Призначення розробки

Додаток призначений для допомоги НПП в обліку навантаження другої половини дня та автоматичному створенні відповідної документації. Автоматизації процесу ведення обліку та оптимізації робочого часу викладачів.

1.2 Мета створення додатку

Головною метою створення вебдодатку для обліку навантаження НПП другої половини дня є забезпечення працівників закладів вищої освіти можливістю спрощення процесу ведення обліку навантаження та автоматизації формування відповідної документації.

1.3 Цільова аудиторія

Цільова аудиторія вебдодатку планування та обліку навантаження НПП:

Адміністрація навчальних закладів: директори та керівництво мають інтерес у використанні додатку для ефективного планування та контролю робочого навантаження працівників, а також здійснення моніторингу робочого часу НПП.

Науково-педагогічні працівники: викладачі мають змогу відстежувати і планувати власне робоче навантаження, а також автоматично генерувати звітну документацію.

2 Вимоги до проекту

2.1 Вимоги до проекту в цілому

2.1.1 Вимоги до структури й функціонування

Розроблюваний програмний продукт повинен бути реалізований за допомогою вебінструментів у вигляді вебдодатку, (далі – продукт, програмний продукт), який складатиметься з двох частин: клієнтської та адміністраторської. Клієнтська сторона продукту містить кілька вебсторінок, з набором нижче описаних можливостей.

Адміністраторська частина додатку містить більший функціонал, в порівнянні з клієнтською та містить вебсторінки, призначені для керування додатком.

Кінцевий продукт даного проекту має бути представлений вебдодатком, який має зручний у використанні інтерфейс та якісне візуальне оформлення, що за стилістикою відповідає сайту кафедри інформаційних технологій <https://itp.elit.sumdu.edu.ua/>.

2.1.2 Вимоги до персоналу

Персонал не повинен мати особливих технічних навичок для роботи з додатком. Вимогами є наявність навичок користування персональним комп'ютером та веб браузером.

2.1.3 Вимоги до збереження інформації

Уся інформація повинна зберігатися у базі даних, реалізованій засобами системи управління базами даних MySQL.

2.1.4 Вимоги до розмежування доступу

Розроблюваний додаток має бути загальнодоступним у мережі Інтернет. Права доступу до інформації розмежовані за групами користувачів: адміністратор та співробітник (НПП). Адміністратор має доступ до налаштувань додатку: погодинних норм, внесення нових користувачів та адміністраторів до системи, редагування або видалення інформації про них. Додавати до системи нові види та типи навантажень, роди діяльності та види занять. Доступ до адміністративної панелі надається за спеціальним логіном та паролем.

Співробітник має можливість створювати, редагувати чи видаляти відомості про планування та облік навантаження (за роками), генерувати відповідну документацію за заповненими відомостями. Доступ до персональної сторінки користувача надається за спеціальним логіном та паролем.

2.2 Структура вебдодатку

2.2.1 Загальна інформація про структуру вебдодатку

До структури вебдодатку входять усі його вебсторінки. Зокрема сторінка авторизації, особистий кабінет користувача та адміністративна панель. В залежності від логіна та пароля визначається статус користувача.

Перелік сторінок вебдодатку наступний:

- сторінка «Авторизація» дозволяє увійти до персонального кабінету користувача, або до адміністративної панелі. Сторінка містить вікно для введення логіна та пароля;
- сторінка «Перелік відомостей навантаження» дозволяє користувачу бачити власні відомості навантаження другої половини дня за роками та відсотком ставки;
- сторінка «Перегляд персональних даних» дозволяє користувачу переглядати власну персональну інформацію;
- сторінка «Редагування відомості» дозволяє користувачу редагувати будь яку відомість, додавати, чи видаляти з відомостей види та типи навантажень, роди діяльності або види занять;
- сторінка «Перелік працівників» дозволяє адміністратору бачити наявних в системі працівників, додавати нових, або видаляти існуючих;
- сторінка «Редагування персональних даних» дозволяє адміністратору редагувати персональну інформацію будь-якого працівника;
- сторінка «Додавання викладача» дозволяє адміністратору внести інформацію про нового співробітника;
- сторінка «Норми» дозволяє адміністратору бачити нагальні погодинні норми робочого та річного навантаження та редагувати їх за потреби;
- сторінка «Вид навантаження» дозволяє адміністратору бачити перелік видів навантаження, додавати, видаляти, або редагувати їх за потреби;
- сторінка «Тип навантаження» дозволяє адміністратору бачити перелік типів навантаження, додавати, видаляти, або редагувати їх за потреби;
- сторінка «Рід діяльності» дозволяє адміністратору бачити перелік родів діяльності, додавати, видаляти, або редагувати їх за потреби;
- сторінка «Вид занять» дозволяє адміністратору бачити перелік видів занять, додавати, видаляти, або редагувати їх за потреби;
- сторінка «Вид навантаження» дозволяє адміністратору бачити перелік видів навантаження, додавати, видаляти, або редагувати їх за потреби;

– сторінка «Планування навантаження» дозволяє адміністратору додавати тип або вид навантаження, рід діяльності, чи вид занять за потреби;

2.2.2 Навігаційне меню

Для зручної навігації повинно бути створене меню як для користувача, так і для адміністратора окремо. Це забезпечить швидке переміщення по всім доступним сторінкам вебдодатку. Меню має бути закріплене і розташовуватися зверху (в «шапці») на кожній сторінці.

2.2.3 Управління контентом

Управління контентом вебдодатку обліку навантаження НПП має здійснюватися за допомогою адміністративної панелі. Усе інформаційне наповнення вебдодатку має міститися у базі даних.

2.2.4 Дизайн вебдодатку

Дизайн вебдодатку має бути виконаний у сучасному стилі, мати зрозумілий та легкий для сприйняття інтерфейс. Корпоративними кольорами вебдодатку є корпоративні кольори сайтів СумДУ та ІПП - відтінки синього та блакитного. Тому під час розробки вебдодатку треба використовувати саме їх.

Види і розміри шрифтів також повинні бути схожими на корпоративні шрифти сайтів СумДУ та ІПП і бути комфортними для перегляду. Інформаційні

блоки, графічні матеріали та інші елементи web-сторінок повинні мати зручне і логічне розташування. Шаблон майбутнього програмного продукту зображено на рисунках А.1 та А.2.

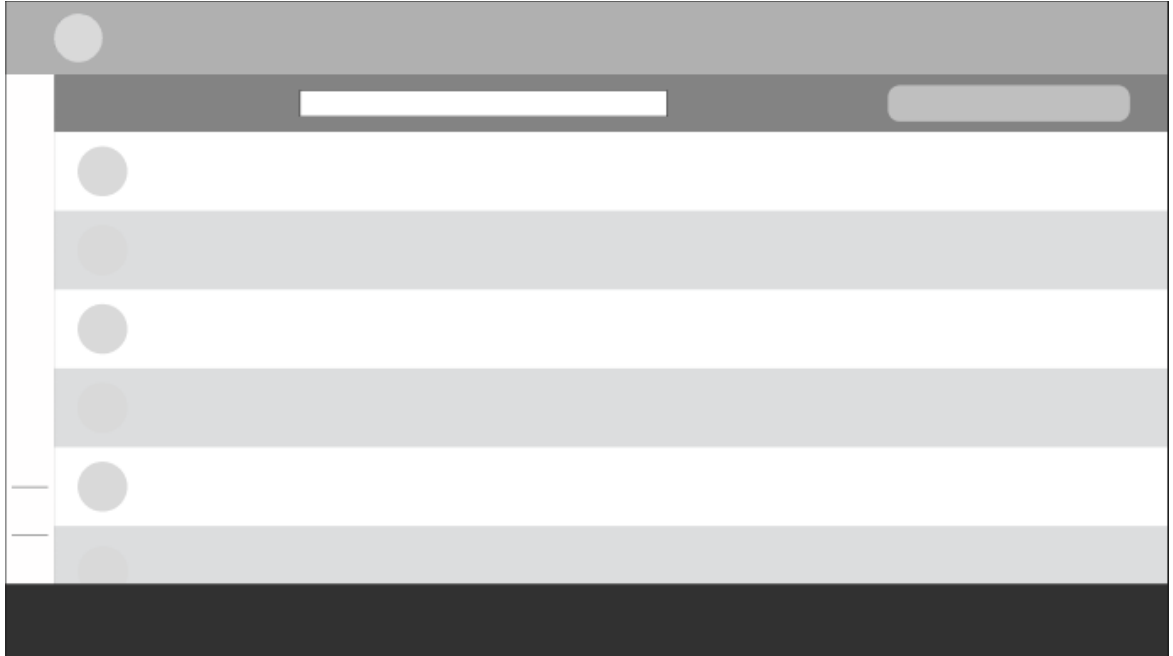


Рисунок А.1 – Схема головної сторінки додатку з точки зору викладача

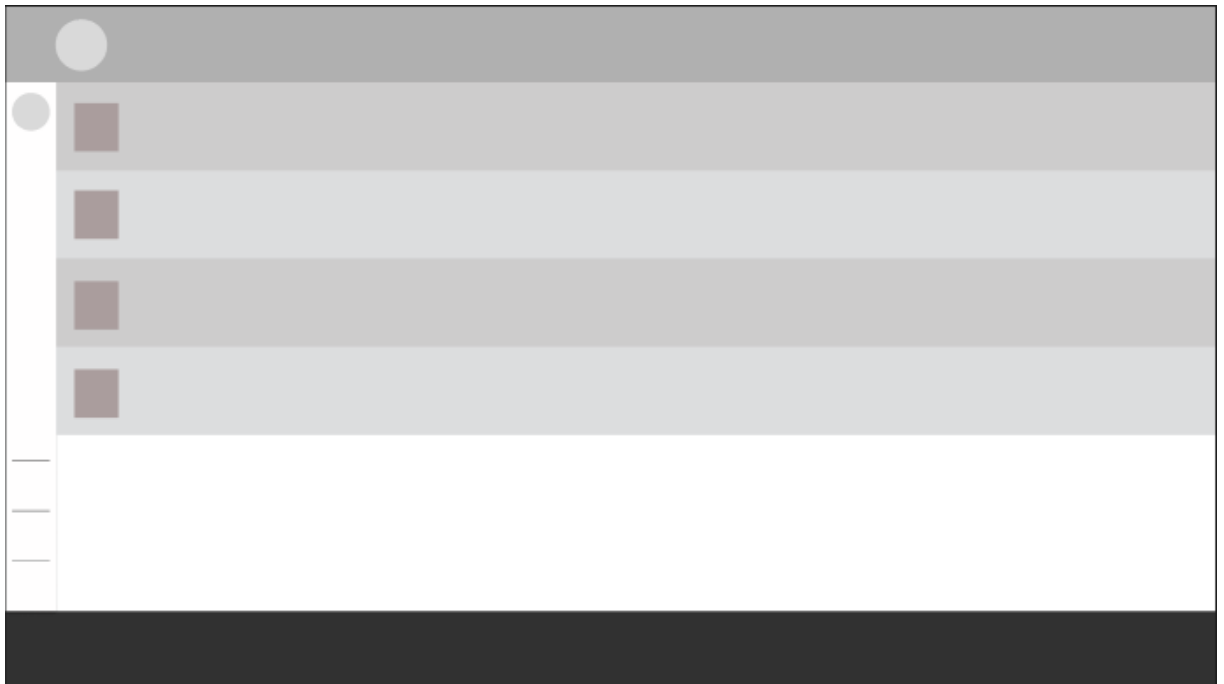


Рисунок А.2 – Схема головної сторінки додатку з точки зору адміністратора

2.2.5 Система навігації (карта вебдодатку)

Карта вебдодатку зображена на рисунку А.3.

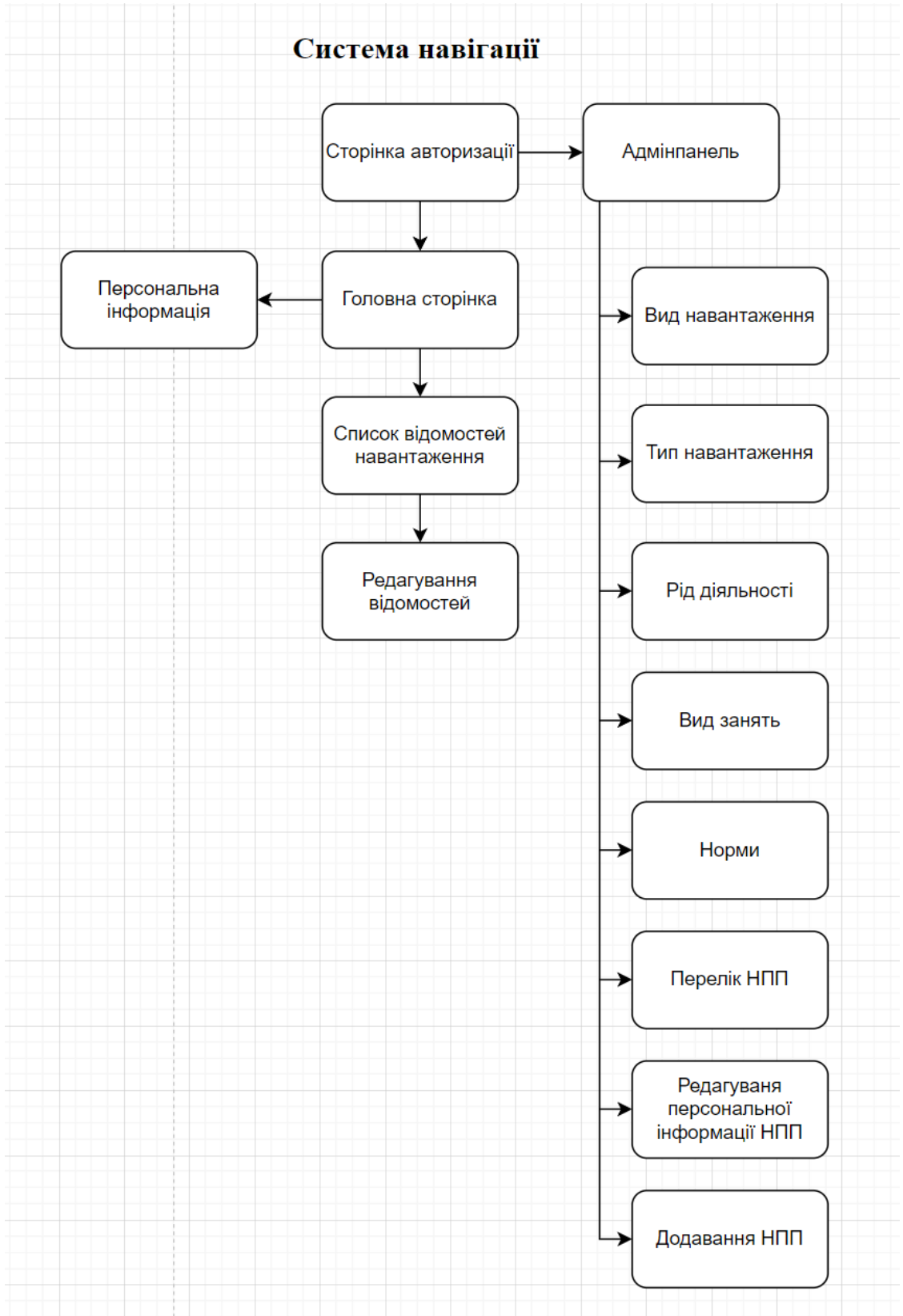


Рисунок А.3 – Система навігації

2.3 Вимоги до видів забезпечення

2.3.1 Вимоги до лінгвістичного забезпечення

Весь текст у вебдодатку має бути виконаний українською.

2.3.2 Вимоги до технології розробки

Продукт буде реалізовано з використанням ітеративної моделі управління розробкою та з урахуванням принципів і технологій уніфікованого процесу розроблення програмного забезпечення. Вебдодаток повинен бути реалізований засобами HTML, CSS та мови JavaScript. Також буде використана мова програмування SQL, як засіб для створення бази даних.

В якості середовища розробки було обрано додаток Visual Studio Code та Open Server Panel для розробки бази даних.

2.3.3 Вимоги до програмного забезпечення

Для коректної роботи всіх можливостей онлайн-системи програмне забезпечення клієнтської частини повинне задовольняти наступним вимогам для веб-браузера: Mozilla Firefox – починаючи з версії 4.0, Google Chrome – 69 починаючи з версії 9.0, Opera – починаючи з версії 12.0, Safari – починаючи з версії 8.0, Interten Explorer – починаючи з версії 11.0.

2.3.4 Вимоги до технічних характеристик ПК

Вимоги до технічних характеристик ПК користувача додатком мають бути наступними: ОЗУ від 4гб, процесор intel core i3 3600 і вище, або їх аналоги AMD.

2.4 Вимоги до функціонування системи

2.4.1 Потреби користувача

Викладач: повинен увійти в аккаунт, щоб мати змогу виконувати наступні дії: переглянути власну інформацію, сформувати новий звіт по навантаженню, відредагувати вже існуючий, або завантижити готовий звіт.

Адміністратор: має право додавати та видаляти працівників, редагувати їх персональні дані. Може додавати, видаляти та редагувати тип та вид навантаження, рід діяльності, вид занять.

Потреби користувача, визначені на основі рішення замовника, представлені у таблиці А.1.

Таблиця А.1 – Потреби користувача

I D	Потреби користувача	Джерело
U N-01	Перегляд особистої інформації	Викладач
U N-02	Створення нових звітів про навантаження	Викладач
U N-03	Перегляд всіх звітів про навантаження	Викладач

Продовження таблиці А.1 – Потреби користувача

U N-04	Редагування звітів про навантаження	Викладач
U N-05	Додавання нових викладачів	Адміністратор тор
U N-06	Додавання нових адміністраторів	Адміністратор тор
U N-07	Редагування персональних даних викладача	Адміністратор тор
U N-08	Додавання, редагування, видалення, перегляд типів навантаження	Адміністратор тор
U N-09	Додавання, редагування, видалення, перегляд видів навантаження	Адміністратор тор
U N-10	Додавання, редагування, видалення, перегляд родів діяльності	Адміністратор тор
U N-11	Додавання, редагування, видалення, перегляд видів занять	Адміністратор тор
U N-12	Редагування норм навантаження	Адміністратор тор

2.4.2 Функціональні вимоги

Проаналізувавши потреби до персоналу закладу освіти було визначено наступні вимоги. Користувальницька частина додатку повинна забезпечувати виконання наступних функцій:

- Перегляд особистої інформації;
- Перегляд архівних відомостей навантаження НПП за попередні навчальні роки;

- Редагування відомостей навантаження НПП;
- Створення нових відомостей навантаження НПП;
- Генерація документів за відомостями;

Адміністраторська панель додатку повинна забезпечувати виконання наступних функцій:

- Зміна норм;
- Редагування, додавання і видалення типу та виду навантаження, роду та виду діяльності;
- Пошук, редагування та видалення інформації про НПП;
- Додавання нових працівників

3 Склад і зміст робіт зі створення вебдодатку

Детальний опис етапів створення вебдодатку наведено в таблиці А.2.

Таблиця А.2 – Етапи створення вебдодатку

№	Склад і зміст робіт	Строк розробки
1	Дослідження предметної області	5 днів
2	Планування проекту	7 днів
3	Розробка моделі web-додатку	4 дні
4	Розробка дизайну web-додатку	8 днів
5	Розробка моделі БД	8 днів
6	Верстка web-додатку	10 днів
7	Розробка web-додатку	15 днів
8	Розробка БД	7 днів
9	Створення web-додатку	15 днів

Продовження таблиці А.2 – Етапи створення вебдодатку

10	Наповнення web-додатку	5 днів
11	Beta - тестування	15 днів
12	Alpha - тестування	15 днів
13	Розміщення на хостингу	1 день
14	Перевірка працездатності	2 дні
15	Написання супровідної документації	2 дні
16	Реліз web-додатку	1 день
	Загальна тривалість робіт	135 днів

4 Вимоги до складу й змісту робіт із введення вебдодатку в експлуатацію

Під час розробки продукту необхідно здійснювати контроль за календарним планом проекту, тестування, перевірити продукт до готовності введення в експлуатацію, а також відповідність розроблюваного додатку до ТЗ. Вебдодаток має бути затверджено та розміщено на вебхостингу.

5 Порядок контролю та приймання

Контроль відповідності процесу розробки графіку календарного планування здійснюється розробником на основі розробленої діаграми Ганта та мережевого графіку. Тестування проводиться розробником згідно вимог ТЗ.

Перевірку придатності додатку та відповідності його до ТЗ здійснюється замовником на основі наданої пояснювальної записки до дипломної роботи та наданих файлів проекту.

ДОДАТОК Б

Планування робіт

Ідентифікація ідеї проекту. Розробка вебдодатку для планування та обліку позааудиторного навантаження науково-педагогічних працівників у вищих навчальних закладах має позитивно вплинути на продуктивність роботи викладача звільнивши цього від рутинних операцій.

Додаток дозволяє вести документаційний облік та відстежувати навантаження викладачів. Також він автоматизує процес створення звітності, що буде значно полегшувати ведення обліку.

Результатом впровадження цього вебдодатка буде полегшення процесів планування, а також створення умов для максимальної продуктивності науково-педагогічного персоналу.

Деталізація мети методом SMART. Мета проекту є створення централізованого вебдодатка, спрямованого на поліпшення планування позааудиторної роботи НПП вищих навчальних закладів. Основні цілі проекту включають в себе підвищення ефективності планування і формування звітної документації, забезпечення зручного планування робочого графіку та обліку їх діяльності.

Результати деталізації методом SMART розміщені у таблиці Б.1.

Таблиця Б.1 - Деталізація мети проекту методом SMART

Specific	Забезпечити потреби НПП та керівництва навчальних закладів необхідним ПЗ для проведення планування та обліку позанавчального навантаження НПП за рахунок створення відповідного вебдодатку
Measurable	Зниження часу, необхідного на планування та облік навантаження НПП на 20% до кінця першого року використання системи; Підвищення ефективності управління персоналом кафедр на 15% до кінця першого року використання системи
Achievable	Наявна необхідна кількість фінансових, технічно-інформаційних та людських ресурсів
Relevant	Відповідність продукту до ТЗ та задоволення потреби замовника
Time-framed	Проект має бути реалізований до 30 червня 2024 року

Планування змісту структури робіт ІТ-проекту. Ієрархічна структура робіт (Work Breakdown Structure) є інструментом, який дозволяє розділити проект на частини. WBS встановлює ієрархічно структурований поділ праці з реалізації проекту для всіх залучених до нього працівників.

Під час планування розробки інформаційної системи була створена WBS. Ієрархічна структура побудована з метою логічного розподілу усіх робіт з виконання проекту і подана у графічному вигляді. На структурі відображено сукупність декількох рівнів, кожний з яких формується в результаті розподілу роботи попереднього рівня на її складові. Елементом найнижчого рівня є група робіт, або так званий робочий пакет (work package).

WBS-структура для даного проекту представлена на рис. Б.1

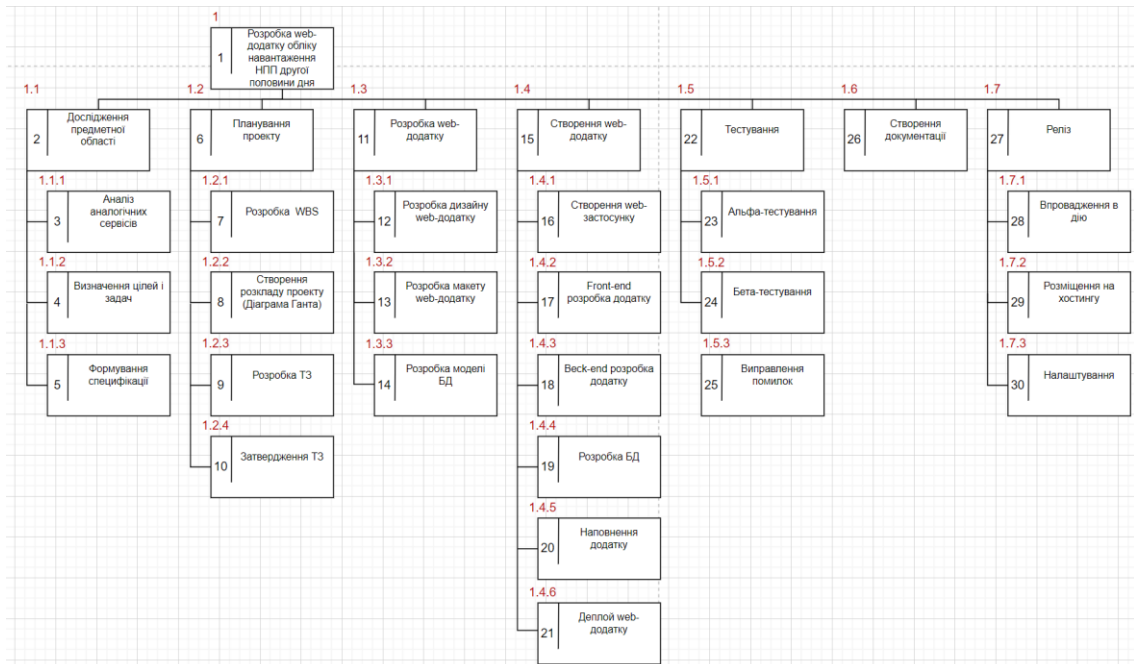


Рисунок Б.1 – WBS-структура вебдодатку планування та обліку позааудиторного навантаження науково-педагогічних працівників

Після розробки WBS-структури проекту наступним етапом є розроблення OBS (Organization structure). Організаційна структура проекту (OBS) є ієрархічною структурою управління проектом і показує відносини між учасниками проекту.

Організаційна структура проекту створюється на рівні підприємства і має елементи, які призначаються на рівні пакетів робіт структури декомпозиції робіт (WBS), що дозволяє контролювати доступ користувачів до інформації відповідного рівня.

OBS-структура для даного проекту представлена на рис. Б.2.

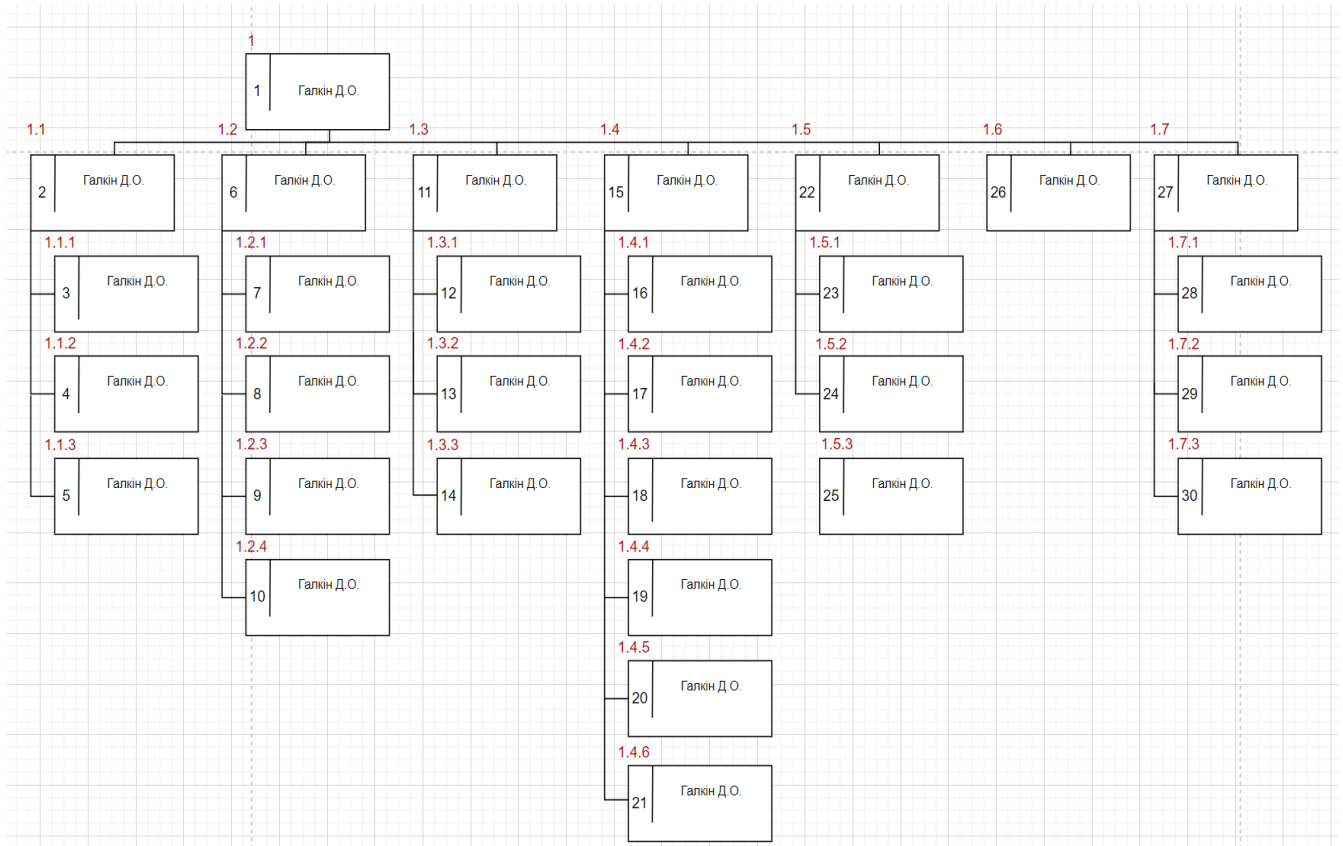


Рисунок Б.2 – Організаційна структура виконавця проекту

Розробка RAM власного проекту. Розробка RAM проекту має величезну актуальність в управлінні проектами та бізнес-процесами. Дана система відображає ролі та відповідальності учасників проекту, що є важливим інструментом для досягнення успішної та ефективної роботи команди розробників.

Таблиця Б.2 – Матриця відповідальності

№	Назва задачі	Відповідальний: Галкін Д. О.
Розробка вебдодатку обліку навантаження НПП другої половини дня		
1	Дослідження предметної області	+
1.1	Аналіз аналогічних сервісів	+

Продовження таблиці Б.2

1.2	Визначення цілей і задач	+
1.3	Формування специфікації	+
2	Планування проекту	+
2.1	Розробка WBS	+
2.2	Створення розкладу проекту (Діаграма Ганта)	+
2.3	Розробка ТЗ	+
2.4	Затвердження ТЗ	+
3	Розробка вебдодатку	+
3.1	Розробка дизайну вебдодатку	+
3.2	Розробка макету вебдодатку	+
3.3	Розробка моделі БД	+
4	Створення вебдодатку	+
4.1	Створення вебзастосунку	+
4.2	Front-end розробка додатку	+
4.3	Back-end розробка додатку	+
4.4	Розробка БД	+
4.5	Наповнення додатку	+

Продовження таблиці Б.2

4.6	Деплой вебдодатку	+
5	Тестування	+
5.1	Alpha-тестування	+
5.2	Beta-тестування	+
5.3	Виправлення помилок	+
6	Створення супровідної документації	+
6.1	Створення супровідної документації	+
7	Реліз	+
7.1	Впровадження в дію	+
7.2	Розміщення на хостингу	+
7.3	Налаштування	+

Побудова календарного графіку виконання ІТ – проекту. Діаграма Ганта (Gantt Chart) – один з найпопулярніших інструментів для наочної ілюстрації календарного плану в проектному менеджменті.

Діаграма Ганта допомагає відслідковувати відсоток робіт, виконаних по кожному завданню. Керівникам проектів дуже важливо правильно розподілити

завдання і бути впевненими в тому, що проект буде завершений вчасно. Основна увага діаграм Ганта зосереджена на процентному завершенні кожного завдання. Під час планування проекту за допомогою Діаграми Ганта представлення робіт проекту відображається у вигляді гістограми, яка вказує тривалість кожного процесу, що був визначений на етапі формування WBS, календарне планування процесів та загальний обсяг робіт.

Графік виконання дипломного проекту представлено у вигляді Діаграми Ганта на рис. Б.3.

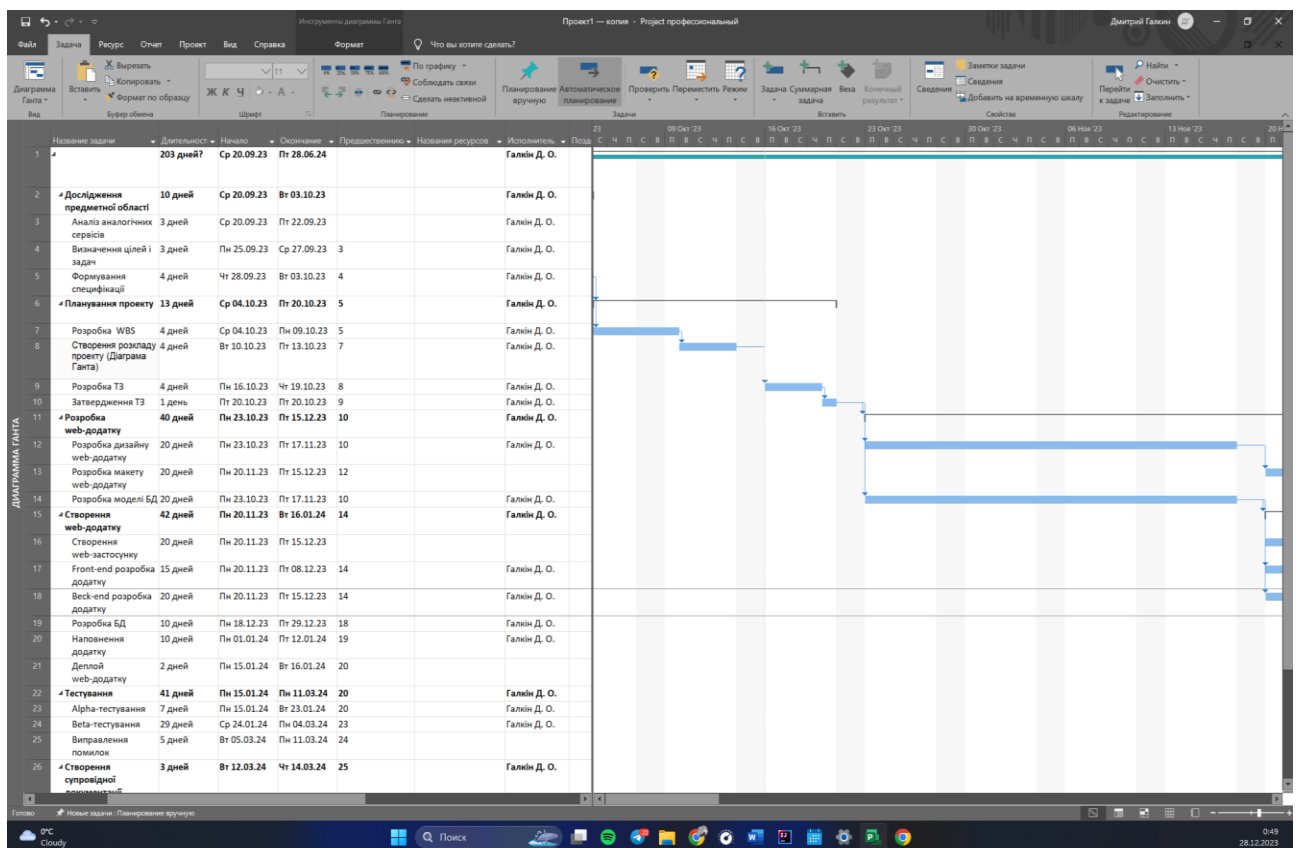


Рисунок Б.3 – Розробка плану робіт проекту за допомогою Діаграми Ганта

Наступним важливим кроком можна вважати побудову та використання мережевого планування, що вже може служити достатньою основою для подальших економічних та математичних розрахунків, аналітики прийнятих рішень та побудови майбутніх стратегічних планів.

Мережевий графік є інструментом якості. За допомогою нього проводиться планування, а також управління роботами. При управлінні якістю на практиці дуже часто користуються саме мережевим графіком.

Найчастіше мережевий графік використовують для проектування різних робіт, що складають набір різного роду дій. Застосування мережевого графіка дозволяє визначити терміни, коли проект завершиться, також допомагає виявити можливість скорочення термінів проведення робіт. Це означає, що в мережевому графіку роботи за часом взаємопов'язані, і саме це надає хорошу можливість контролювати їх хід. Мережевий графік виконання дипломного проекту представлений на рисунку Б.4.

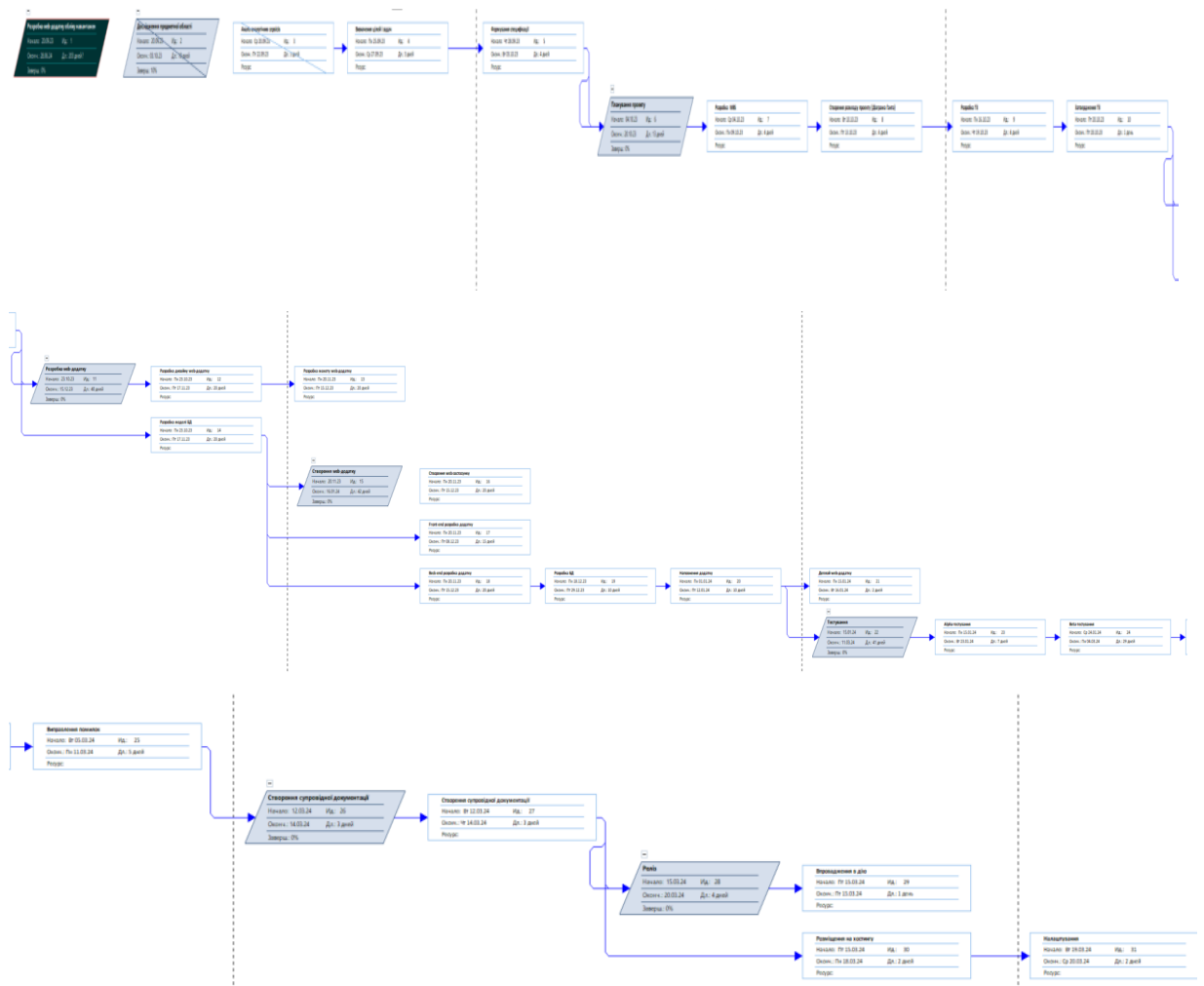


Рисунок Б.4 – Мережевий графік виконання дипломного проекту

Планування ризиків проекту, ідентифікація ризиків проекту. При розробці будь-якого проекту необхідно передбачити всі можливі ризики, які можуть трапитись та провести необхідні заходи для їх уникнення чи зменшення збитків.

Для початку сформуємо перелік основних ризиків проекту, перелік ризиків наведено в таблиці Б.3.

Таблиця Б.3 – Ідентифікація ризиків проекту

№ ризику	Назва (опис) ризику
1	Відсутність чіткого ТЗ
2	Проблеми з розміщенням на хостингу, неякісний деплой
3	Постійні зміни ТЗ
4	Технічно неякісна реалізація веб додатку
5	Різкі зміни в робочих процесах
6	Перевищення строку здачі проекту
7	Неякісний інтерфейс та графічне наповнення
8	Проблеми комунікації між науковим керівником і студентом
9	Вимкнення світла, блекаути
10	Хвороба членів команди розробників
11	Неякісно розроблена БД

Оцінка ризиків проекту. Оцінка ризиків проекту є критичним етапом у його життєвому циклі. Цей процес передбачає ідентифікацію, аналіз та управління можливими негативними подіями, які можуть вплинути на досягнення мети проекту. Актуальність оцінки ризиків полягає у здатності заздалегідь визначити ймовірні проблеми та вжити заходів для їх попередження чи ефективного управління в разі виникнення.

Оцінка ризиків проекту представлена в таблиці Б.4.

Таблиця Б.4 – Ідентифікація ризиків проекту

№ ризику	Назва (опис) ризику	Ймовірність (0,1-0,9)	Вплив (0,05-0,8)	Ранг
1	Відсутність чіткого ТЗ	0,3	0,8	0,24
2	Проблеми з розміщенням на хостингу, неякісний деплой	0,2	0,8	0,16
3	Постійні зміни ТЗ	0,1	0,5	0,05
4	Технічно неякісна реалізація веб додатку	0,2	0,7	0,14
5	Різкі зміни в робочих процесах	0,2	0,7	0,14
6	Перевищення строку здачі проекту	0,1	0,8	0,08
7	Неякісний інтерфейс та графічне наповнення	0,3	0,2	0,06
8	Проблеми комунікації між науковим керівником і студентом	0,1	0,4	0,04
9	Вимкнення світла, блекаути	0,5	0,6	0,3
10	Хвороба членів команди розробників	0,4	0,4	0,16
11	Неякісно розроблена БД	0,3	0,4	0,12

Побудова матриці ймовірності. Побудова матриці ймовірності - це важливий етап у процесі оцінки ризиків проекту, який дозволяє систематично оцінити ймовірність виникнення ризикованих подій та їх вплив на успішність проекту. Цей інструмент використовується для кількісної оцінки ризиків і визначення їхнього потенційного впливу.

При побудові матриці ймовірності, перш за все, ідентифікуються конкретні ризиковані події, які можуть впливати на проект. Далі визначається ймовірність кожної з цих подій та їх можливий вплив на проектні цілі.

Таблиця Б.5 – Матриця ймовірності та впливу згідно проекту

Ймовірність ризику (Й)	Вплив загрози (ризик)				
	Дуже малий	Малий	Середній	Великий	Дуже великий
	0,05	0,1	0,2	0,4	0,8
0,9	0,045	0,09	0,18 R2 R10	0,36 R9	0,72
0,7	0,035	0,07	0,14 R4 R5	0,28	0,56
0,5	0,025	0,05 R3	0,1	0,2	0,4
0,3	0,015	0,03	0,06 R7	0,12 R11	0,24 R1
0,1	0,005	0,01	0,02	0,04 R8	0,08 R6

Таблиця Б.6 – Шкала оцінювання за рівнем ризику

№	Назва	Межі	Ризики, які входять (номера)
1	Прийнятні	$0,005 \leq R \leq 0,05$	3, 8
2	Виправдані	$0,05 < R \leq 0,14$	4, 5, 6, 7, 11
3	Недопустимі	$0,14 < R \leq 0,72$	1, 2, 9, 10

Проведення аналізу недопустимих ризиків. Для усіх недопустимих ризиків необхідно застосувати відповідні стратегії реагування на загрози (ухилення, зменшення або передача) і зазначити якими заходами це можна досягти.

Результати представлені в таблиці Б.7.

Таблиця Б.7 – Результати застосування стратегій реагування

№	Назва ризику	Стратегія реагування на загрозу	Відповідні заходи	Результат (уникнення, числове значення зменшення ймовірності чи/або впливу тощо)
1	Відсутність чіткого ТЗ	Зменшення	Формування чітких вимог до проекту та ТЗ, проведення аудиту вимог, ведення відповідної документації. Підбір більш досвідченої команди розробників	0,1/0,5/0,05 Ризик великий, прийнятний
2	Проблеми з розміщенням на хостингу, неякісний деплой	Зменшення	Формування чіткої процедури процесу розміщення додатку на хостингу, вивчення документації на дану тематику.	0,1/0,5/0,05 Ризик великий, прийнятний
9	Вимкнення світла, блекаут	Зменшення	Мати резервне джерело живлення та інтернету для забезпечення працездатності за будь-яких умов.	0,5/0,1/0,05 Ризик великий, прийнятний
10	Хвороба членів команди розробників	Зменшення	Планування проекту з урахуванням можливих ризиків	0,4/0,1/0,04 Ризик великий, прийнятний

Таблиця Б.8 – Шкала оцінювання за рівнем ризику

№	Назва	Межі	Ризики, які входять (номера)
1	Прийнятні	$0,005 \leq R \leq 0,05$	1, 2, 3, 8, 9, 10
2	Виправдані	$0,05 < R \leq 0,14$	4, 5, 6, 7, 11
3	Недопустимі	$0,14 < R \leq 0,72$	-

Розробка заходів реагування на виявлені ризики проекту. На основі попередньо представлених даних створимо заходи реагування на виявлені ризики проекту. Результати наведені в таблиці Б.10.

Таблиця Б.10 – Ризики проекту та стратегії реагування

ID Ризику	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А (заходи запобігання виникненню ризику)	Тип стратегії реагування	План Б (заходи усунення наслідків ризику)
1	Прийнятий	Відсутність чіткого ТЗ	0,1	0,5	0,05	Розробка докладного технічного завдання, уточнення вимог	Зменшення	Проведення аудиту вимог, перегляд проекту
2	Прийнятий	Проблеми з розміщенням на хостингу, неякісний деплой	0,1	0,5	0,05	Формування чіткої процедури процесу розміщення додатку на хостингу, вивчення документації на дану тематику.	Ухилення	Розробка докладного плану розміщення на хостингу та розгляд альтернатив.
3	Прийнятий	Постійні зміни ТЗ	0,1	0,5	0,05	Фіксовані умови щодо змін в ТЗ, систематичне оновлення документації, покращення комунікації з науковим керівником	Зменшення	Визначення процедур зміни ТЗ, зменшення ризику
4	Виправлений	Технічно неякісна реалізація вебдодатку	0,2	0,7	0,14	Перевірка якості розробки та тестування, вивчення документації, тощо	Зменшення	Виправлення дефектів, переробка коду, збільшення тестування

Продовження таблиці Б.10

5	Виправданний	Різкі зміни в робочих процесах	0,2	0,7	0,14	Ретельне планування проекту, аудит поетапного виконання процесів	Зменшення	Контроль ходу робіт проекту
6	Виправданний	Перевищення строку задачі проекту	0,1	0,8	0,08	Ретельне планування проекту, відстеження та контроль робочого процесу	Зменшення	Додаткові ресурси, перегляд термінів проекту
7	Виправданний	Неякісний інтерфейс та графічне наповнення	0,3	0,2	0,06	Підвищення кваліфікації розробника, перевірка референсів	Зменшення	Співпраця з консультантами
8	Прийнятний	Проблеми комунікації між науковим керівником і студентом	0,1	0,4	0,04	Забезпечення чіткої комунікації, планування регулярних зустрічей	Зменшення	Врегулювання конфліктів, зміцнення комунікації
9	Прийнятний	Вимкнення світла, блекауту	0,5	0,1	0,05	Завершення резервних джерел живлення, резервне зберігання даних	Зменшення	Відновлення системи, відновлення даних
10	Прийнятний	Хвороба розробника	0,5	0,1	0,05	Планування проекту з урахуванням можливих ризиків	Ухилення	Ретельне планування робочих графіків
11	Виправданний	Неякісно розроблена БД	0,3	0,4	0,12	Підвищення кваліфікації розробника, перевірка якості та тестування	Зменшення	Контроль за якістю розробленої БД керівником проекту

ДОДАТОК В

Лістинг коду

```

creating_report_npp.php
<?php
session_start();
require_once "../configuration/connect.php";

$Kind_of_activity = mysqli_query($connect, "SELECT * FROM `Kind_of_activity`");
$Kind_of_activity = mysqli_fetch_all($Kind_of_activity);

$Type_of_activity = mysqli_query($connect, "SELECT * FROM `Type_of_activity`");
$Type_of_activity = json_encode(mysqli_fetch_all($Type_of_activity));

$Activity = mysqli_query($connect, "SELECT * FROM `Activity`");
$Activity = json_encode(mysqli_fetch_all($Activity));

$Item = mysqli_query($connect, "SELECT * FROM `Item`");
$Item = json_encode(mysqli_fetch_all($Item));
?>

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link
      rel="stylesheet"
      href="https://fonts.googleapis.com/css2?family=Manrope:wght@200;300;400;500;600;700;800&di
      splay=swap"
    />
    <link rel="icon" href="../img/logo.png" type="image/x-icon" />
    <link
      type="text/css"
      href="style/creating_report_npp.css"
      rel="stylesheet"
    />
    <title>Ведення документації</title>
  </head>

  <script
    src="https://cdnjs.cloudflare.com/ajax/libs/docxtemplater/3.45.0/docxtemplater.js"></script>
    <script src="https://unpkg.com/pizzip@3.1.6/dist/pizzip.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/FileSaver.js/1.3.8/FileSaver.js"></script>
    <script src="https://unpkg.com/pizzip@3.1.6/dist/pizzip-utils.js"></script>

```

```

<body>
<div class="all">
  <!-- Хедер -->
  <header class="header">
    <div class="img"></div>
    <p>Кафедра інформаційних технологій</p>
  </header>

  <!-- Основна частина сайту -->
  <div class="main">
    <!-- Меню сайту -->
    <div class="menu">
      <div class="photoperson">
        <?php
        if($_SESSION['user']['Photo'] != NULL)
        {
          echo '';
        }
        else
        {
          echo '';
        }
        ?>
      </div>
      <div class="menunumber">
        <ul>
          <li>
            <a href="person_information.php">
              
            </a>
          </li>
          <li>
            <a href="choice.php">
              
            </a>
          </li>
          <li>
            <a href="choiceview.php">
              
            </a>
          </li>
          <li>
            <a href="../configuration/exit.php">
              
            </a>
          </li>
        </ul>
      </div>
    </div>
    <div class="adapt">
      <div class="name">

```

```

<div class="namereport">
  <p>Планування навантаження</p>
</div>
<div class="date">
  <form method="post" id="dateForm">
    <div class="form-group">
      <label for="date">
        Введіть дату початку обліку навантаження
      </label>
      <input
        type="date"
        id="date"
        placeholder="Виберіть дату початку"
        name="date_start"
        required
      />
    </div>
    <div class="form-group">
      <label for="date">
        Введіть дату кінця обліку навантаження
      </label>
      <input
        type="date"
        id="date"
        placeholder="Виберіть дату кінця"
        name="date_end"
        required
      />
    </div>
    <div class="form-group">
      <label for="date">
        Введіть частку ставки (0.2-1.0)
      </label>
      <input type="number" id="valueNumber" name="valueNumber" min="0.2" max="1.0"
step="0.1" value="0.2">
    </div>
  </form>
</div>
<div class="ma">

  <!-- <div class="formreport">
    <div class="form">
      <form method="post">
        <div class="name">
          <input type="text" name="readonlyInput" value="НАВЧАЛЬНЕ НАВАНТАЖЕННЯ
НПП"
          readonly style="border:none; background:transparent; font-size:inherit;">
          <button class="del">
            
          </button>
        </div>
      </form>
    </div>
  </div>

```

```

<div class = "d">
  <div class="form-group">
    <label> Тип навантаження </label>
    Placeholder for selectElement
    <div id="selectContainer">
      <select name="typebenchmark" required>
        <option value="Kzkzk">Kzkzk</option>
        <option value="Fff">Fff</option>
      </select>
    </div>
  </div>
  <div class="form-group">
    <label> Рід діяльності </label>
    <select name="subtype" id="">
      <option value="Kzkzk">Kzkzk</option>
      <option value="Fff">Fff</option>
    </select>
  </div>
  <div class="form-group">
    <label> Вид занять </label>
    <input
      type="text"
      id="thematicfocus"
      placeholder="Введіть вид занять"
      name="thematicfocus"
      required
    />
  </div>
  <div class="form-group" id="form-group1">
    <label>
      Кількість годин по кафедрі
    </label>
    <input
      type="number"
      id="points"
      placeholder=""
      name="points"
      required
    />
  </div>
</div>
</form>
</div>
</div -->

<!-- Контейнер для створених блоків -->
<div id="container"></div>
<div class="add">
  <div class="done">
    <button type="button" id="submitAll">Сформувати звіт</button>
  </div>
  <div class="add1">

```

```

        <a href="" id="openModal"
          ></a>
      </div>
    </div>
  </div>
</div>
</div>
</div>
</div>
<!-- Футер -->
<footer class="footer">
  <div class="location">
    <p class="imgloc">
      
    </p>
    <p class="adres">
      вул. Харківська, 116, Суми, Україна, 40007 Головний корпус, 13-й
      поверх, ауд. Г1303
    </p>
  </div>

  <div class="phone">
    <div class="phone1">
      <p class="img">
        
      </p>
      <p class="pr">приймальня +38 (0542) 68-78-54</p>
    </div>
    <div class="phone2">
      <p class="img">
        
      </p>
      <p class="pr">завідувач кафедри +38 (0542) 68-78-63</p>
    </div>
  </div>

  <div class="email">
    <div class="insta">
      <a href="https://www.instagram.com/itp_sumdu?igshid=fopu9dqxdigb">
        
        <span>Instagram</span>
      </a>
    </div>
    <div class="em">
      <p class="img2"></p>
      <p class="adres">info@itp.sumdu.edu.ua</p>
    </div>
  </div>

```

```

</footer>
</div>

<!-- Модальне вікно -->
<div id="modal" class="modal">
  <div class="modal-content">
    <div class="c1">
      <h2>Оберіть вид контрольного показника</h2>
      <span class="close">&times;</span>
    </div>
    <form id="radioForm">

      <?php
      foreach ($Kind_of_activity as $Kind_of_activity_key) {
        echo
        '<div class="form-group">
          <label>
            <input
              type="radio"
              name="option"
              value="' . $Kind_of_activity_key[0] . ' . ' . $Kind_of_activity_key[1] . ' . '
              required
            />
            ' . $Kind_of_activity_key[0] . ' . ' . $Kind_of_activity_key[1] . '
          </label>
        </div>';
      }
      ?>

      <div class="donnn"><button type="submit">Підтвердити</button></div>
    </form>
  </div>
</div>

<script src="js/modal_creative_report.js"></script>
<script>
  document.addEventListener("DOMContentLoaded", function () {
  // Get modal window and form
  const modal = document.getElementById("modal");
  const radioForm = document.getElementById("radioForm");
  const container = document.getElementById("container");

  // Close modal window
  document.querySelector(".close").onclick = function () {
    modal.style.display = "none";
  };

  // Close modal window when clicking outside the window
  window.onclick = function (event) {
    if (event.target == modal) {
      modal.style.display = "none";
    }
  }
}

```



```

};

// Handle form submission
radioForm.onsubmit = function (event) {
  event.preventDefault();
  const selectedOption = document.querySelector(
    'input[name="option"]:checked'
  ).value;

  // Fetch data from the database
  var Type_of_activity = <?php echo $Type_of_activity; ?>;
  var Activity = <?php echo $Activity; ?>;
  var Item = <?php echo $Item; ?>;
  var parts = selectedOption.split(' ');
  var Kind_of_activity_ID = parts[0];
  const matchedValues = [];
  const activityValues = [];

  for (const item of Type_of_activity) {

    const lastValue = item[2];
    const idValue = item[0];

    if (lastValue === Kind_of_activity_ID) {
      matchedValues.push(item[1]);
      activityValues.push(item[0]);
    }
  }

  console.log(matchedValues);
  console.log(activityValues);

  // Create <select> element
  const selectElement = document.createElement('select');
  selectElement.name = 'type';
  selectElement.required = true;

  // Додавання опції за замовчанням
  const defaultOption = document.createElement('option');
  defaultOption.value = "";
  defaultOption.textContent = 'Оберіть тип навантаження';
  defaultOption.disabled = true;
  defaultOption.selected = true;
  selectElement.appendChild(defaultOption);

  // Видалення опції за замовчанням після вибору іншої опції
  selectElement.addEventListener('change', function() {
    const defaultOption = selectElement.querySelector('option[value=""]');
    if (defaultOption) {
      selectElement.removeChild(defaultOption);
    }
  });
});

```

```

// Додавання опцій в елемент <select>
matchedValues.forEach(value => {
  const option = document.createElement('option');
  option.value = value;
  option.textContent = value;
  selectElement.appendChild(option);
});

// Create <select> element для підтипу
const selectElement2 = document.createElement('select');
selectElement2.name = 'subtype';
selectElement2.required = true;

// Додавання опції за замовчанням для селекта підтипу
const defaultOption3 = document.createElement('option');
defaultOption3.value = "";
defaultOption3.textContent = 'Оберіть спочатку тип навантаження';
defaultOption3.disabled = true;
defaultOption3.selected = true;
selectElement2.appendChild(defaultOption3);

// Видалення опції за замовчанням після вибору іншої опції
selectElement.addEventListener('change', function() {
  const defaultOption3 = selectElement.querySelector('option[value=""]');
  if (defaultOption3) {
    selectElement.removeChild(defaultOption3);
  }
});

// Додавання обробника події зміни (change) для елемента <select> для типу
selectElement.addEventListener('change', function(event) {
  var nameActivity = [];
  selectElement2.innerHTML = "";

  const selectedValue = event.target.value;
  console.log('Вибране значення:', selectedValue);

  var typeValues = "";
  for (const item of Type_of_activity){
    if (selectedValue === item[1]) {
      typeValues = item[0];
    }
  }

  for (const item of Activity){
    const id = item[1];
    if (id === typeValues){
      nameActivity.push(item[3]);
    }
  }
}

```

```

// Додавання опцій в елемент <select> для підтипу
nameActivity.forEach(value => {
  const option = document.createElement('option');
  option.value = value;
  option.textContent = value;
  selectElement2.appendChild(option);
});

console.log(nameActivity);
});

// Create a new block
const newBlock = document.createElement("div");
newBlock.className = "formreport";

newBlock.innerHTML = `
<div class="form">
  <form method="post">
    <div class="name">
      <input type="text" name="readonlyInput" value="{selectedOption}"
        readonly style="border:none; background:transparent; font-size:inherit;">
      <button class="del">
        
      </button>
    </div>
    <div class="d">
      <div class="form-group">
        <label>Тип навантаження</label>
        <!-- Placeholder for selectElement -->
        <div id="selectContainer"></div>
      </div>
      <div class="form-group">
        <label>Рід діяльності</label>
        <!-- Placeholder for selectElement -->
        <div id="selectContainer2"></div>
      </div>
      <div class="form-group">
        <label>Вид занять</label>
        <input
          type="text"
          id="thematicfocus"
          placeholder="Введіть вид занять"
          name="thematicfocus"
          required
        />
      </div>
      <div class="form-group" id="form-group1">
        <label>Кількість годин по кафедрі</label>
        <input
          type="number"
          id="points"
          placeholder="Введіть кількість годин"

```

```

        name="points"
        required
    />
</div>
</div>
</form>
</div>
`;

// Append the selectElement to the placeholder div (для типу)
newBlock.querySelector('#selectContainer').appendChild(selectElement);
// Append the selectElement2 to the placeholder div (для підтипу)
newBlock.querySelector('#selectContainer2').appendChild(selectElement2);

// Add the new block to the container
container.appendChild(newBlock);

// Close the modal window
modal.style.display = "none";
};

// Remove block on delete button click
container.addEventListener("click", function (event) {
    if (event.target.closest(".del")) {
        event.target.closest(".formreport").remove();
    }
});
});
</script>

<script>
    // Збір даних з усіх форм
    document.getElementById('submitAll').onclick = function () {
        const formReports = document.querySelectorAll('.formreport');
        const collectedData = [];

        formReports.forEach(formReport => {
            const form = formReport.querySelector('form');
            const formData = new FormData(form);

            const data = {};
            formData.forEach((value, key) => {
                data[key] = value;
            });

            collectedData.push(data);
        });

        // const dateForm = document.getElementById('dateForm');

```

```

// const formData = new FormData(dateForm);

// const data = {};
// formData.forEach((value, key) => {
//   data[key] = value;
// });

// Отримати значення форми
const dateForm = document.getElementById('dateForm');
const formData = new FormData(dateForm);
const data = {};
const startDate = new Date(formData.get('date_start'));
const endDate = new Date(formData.get('date_end'));
const ставка = formData.get('valueNumber');
// Перевірка, чи всі необхідні поля містять значення
if (formData.has('date_start') && formData.has('date_end')) {
  // Отримати значення дат початку та кінця з FormData
  //console.log(startDate)
  //console.log(endDate)
  // Перевірка, чи дата початку є раніше за дату кінця
  if (startDate < endDate) {
    // Якщо усі перевірки пройдено успішно, додати дані до об'єкта data

    formData.forEach((value, key) => {
      data[key] = value;
    });
    console.log(data);
    console.log('Зібрані дані з усіх форм:', collectedData);
    console.log('Зібрані дані з форми дати та частки ставки:', data);
    let secondname2 = <?php echo json_encode($_SESSION['user']['Second_name_teacher']); ?>;
    let name2 = <?php echo json_encode($_SESSION['user']['First_name_teacher']); ?>;
    let fatherhood2 = <?php echo json_encode($_SESSION['user']['Fatherhood_teacher']); ?>;

    // Отримання дня, місяця та року
    let day = startDate.getDate();
    let month = startDate.getMonth() + 1; // Місяць від 0 до 11, тому додаємо 1
    let year = startDate.getFullYear();

    // Додаємо "0" перед одноцифровими числами
    let dayString = day < 10 ? '0' + day : day.toString();
    let monthString = month < 10 ? '0' + month : month.toString();
    let yearString = year.toString();

    // Використовуємо dayString, monthString, yearString для форматування дати
    let formattedDate = `${dayString}.${monthString}.${yearString}`;

    // Отримання дня, місяця та року
    let day2 = endDate.getDate();
    let month2 = endDate.getMonth() + 1; // Місяць від 0 до 11, тому додаємо 1
    let year2 = endDate.getFullYear();

```

```

// Додаємо "0" перед одноцифровими числами
let dayString2 = day2 < 10 ? '0' + day2 : day2.toString();
let monthString2 = month2 < 10 ? '0' + month2 : month2.toString();
let yearString2 = year2.toString();

// Використовуємо dayString, monthString, yearString для форматування дати
let formattedDate2 = `${dayString2}.${monthString2}.${yearString2}`;

<?php
$post_Id = $_SESSION['user']['Post_ID'];
$item_post = mysqli_query($connect, "SELECT `Post_name` FROM `Item_post` WHERE
`Post_ID` = '$post_Id'");
$item_post = mysqli_fetch_assoc($item_post);
?>

let post2 = <?php echo json_encode($item_post); ?>;
function loadFile(url, callback) {
  PizZipUtils.getBinaryContent(url, callback);
}

loadFile(
  "Template_npp.docx", // You need to replace this with the actual path to your template
  function (error, content)
  {
    if (error) {throw error;}
    const zip = new PizZip(content);
    const doc = new window.docxtemplater(zip,
    {
      paragraphLoop: true,
      linebreaks: true,
    });

    const transformedData = collectedData.map(item => ({
      points: item.points,
      readonlyInput: item.readonlyInput,
      subtype: item.subtype,
      thematicfocus: item.thematicfocus,
      type: item.type
    }));

    // Render the document
    doc.render({
      name: `${name2}`,
      surname: `${secondname2}`,
      fatherhood: `${fatherhood2}`,
      post: `${post2.Post_name}`,
      date_start: `${formattedDate}`,
      date_end: `${formattedDate2}`,

```

```

        transformedData: transformedData,
        stavka: `${stavka}`

    });

    const blob = doc.getZip().generate({
        type: "blob",
        mimeType: "application/vnd.openxmlformats-officedocument.wordprocessingml.document",
        compression: "DEFLATE",
    });
    console.log(blob);
    // Збереження файлу в браузері
    let fileName = secondname2 + "_" + name2[0] + "_" + fatherhood2[0] +
    "_навантаження." + formattedDate + "_" + formattedDate2 + ".docx";
    saveAs(blob, fileName);

    content = doc.getZip().generate({ type: "blob" });

    // Відправляємо Blob на сервер
    let formData = new FormData();
    formData.append('file', content, fileName); // 'report.docx' - ім'я файлу, яке ви хочете
    використовувати на сервері

    fetch('save.file.npp.php', {
        method: 'POST',
        body: formData
    })
    .then(response => {
        if (response.ok) {
            return response.text();
        }
        throw new Error('Network response was not ok.');
```

```
</script>
```

```
</body>
```

```
</html>
```

```
creating_report_npp.css
```

```
/* Reset стилів для браузера */
```

```
* {  
  -webkit-box-sizing: border-box;  
  -moz-box-sizing: border-box;  
  -o-box-sizing: border-box;  
  -ms-box-sizing: border-box;  
  box-sizing: border-box;  
  margin: 0;  
  padding: 0;  
}
```

```
/* Загальні стилі */
```

```
body {  
  font-family: "Манропе";  
}
```

```
/* Контейнер для всього */
```

```
.all {  
  display: flex;  
  flex-direction: column;  
  min-height: 100vh; /* Мінімальна висота сторінки */  
}
```

```
/* Стили для хедера */
```

```
.header {
```



```
display: flex;
flex-direction: row;
background-color: #e9f8ff;
height: 100px;
position: fixed;
width: 100vw;
}

.header .img {
display: flex;
align-items: center;
padding-left: 80px;
}

.header .img img {
height: 60px;
width: 60px;
}

.header p {
display: flex;
align-items: center;
padding-left: 44px;
font-weight: 490;
font-size: 35px;
line-height: 27px;
color: #001e5a;
letter-spacing: 3px; /* більший інтервал між буквами */
}

.main {
display: flex;
flex-direction: row;
flex: 1;
margin-top: 100px;
}

/* Стили для меню сайту */
.menu {
display: flex;
flex-direction: column;
justify-content: space-between; /* !!!! якщо потрібно без фото тоді END */
align-items: center;
width: 80px;
box-shadow: 2px 4.5px 10px rgba(0, 0, 0, 0.2);
background-color: #ffffff;
color: black;
}

.photoperson {
padding-top: 9px;
}
```

```
.photoperson img {
  width: 65px;
  height: 65px;
  border-radius: 50%;
  object-fit: cover; /* Масштаб фото, щоб воно повністю заповнило круг */
}

.menu ul {
  list-style-type: none;
  padding: 0; /* Видаляємо відступи по умовчанняю */
}

.menu li {
  padding-bottom: 9px;
  margin-bottom: 15px; /* Відступ між пунктами */
  border-bottom: 2px solid #3b3737; /* Додаємо лінію між пунктами */
}

.menu ul li:last-child {
  margin-bottom: 0px; /* Відступ між пунктами */
  border-bottom: none; /* Видаляємо лінію після останнього пункту */
}

.menu li a {
  color: black;
}

.menu li a img {
  width: 35px;
  height: 35px;
}

/* Стили для форми сайту дати*/
.adapt {
  display: flex;
  flex-direction: row;
  flex: 1;
}

.name {
  background-color: #e9f8ff;
  width: 250px;
}

.namereport {
  display: flex;
  align-items: center;
  text-align: center;
  background-color: #001e5a;
  color: #ffffff;
  height: 90px;
}
```

```
}

.namereport p {
  font-weight: 400;
  font-size: 20px;
  line-height: 35px;
  letter-spacing: 0.07em;
}

.date form {
  margin-top: 30px;
}

.date .form-group {
  display: flex;
  flex-direction: column;
  margin-left: 5px;
  margin-right: 5px;
  margin-bottom: 25px;
}

.date .form-group label {
  font-weight: 500;
  font-size: 15px;
  line-height: 25px;
  letter-spacing: 0.07em;
  margin-bottom: 5px;
  text-align: justify;
}

.date .form-group input {
  width: 240px;
  height: 35px;
  font-size: 12px;
  box-shadow: 0px 4px 4px rgba(0, 0, 0, 0.25);
  border-radius: 5px;
  padding: 10px;
  font-size: 14px;
  background-color: #ffffff;
  border: none;
  outline: none;
}

.date .form-group button {
  width: 100px;
  height: 28px;
  font-size: 16px;
  margin-bottom: -30px;
  margin-top: 9px;
}

/* Стили для форми сайту звіту*/
```

```
.ma {
  display: flex;
  flex-direction: column;
  flex-grow: 1;
}

.formreport {
  display: flex;
  flex-direction: column;
  flex-grow: 1;
  min-height: 400px;
  height: auto;
  margin: 15px;
  box-shadow: 0px 10px 30px rgba(48, 46, 46, 0.142);
}

.formreport .name {
  display: flex;
  align-items: center;
  justify-content: space-between;
  height: 80px;
  width: 100%; /* Займае всю ширину */
  background-color: #001e5a;
  padding-left: 20px;
}

.del {
  background-color: #001e5a;
  border: none;
  margin-right: 30px;
  cursor: pointer;
}

.del img {
  width: 35px;
  height: 35px;
  object-fit: cover;
}

.form .d {
  margin: 20px;
}

.form .form-group {
  display: flex;
  flex-direction: row;
  margin-left: 5px;
  margin-right: 5px;
  margin-bottom: 25px;
}

.form label {
```

```
display: block;
display: flex;
font-weight: 500;
font-size: 15px;
line-height: 25px;
letter-spacing: 0.07em;
margin-bottom: 5px;
text-align: justify;
width: 400px;
height: 35px;
align-items: center;
margin-right: 15px;
}
```

```
#selectContainer {
  width: 100%;
}
```

```
#selectContainer2 {
  width: 100%;
}
```

```
#selectContainer option {
  width: 100%;
  height: 35px;
  font-size: 12px;
  box-shadow: 0px 0px 6px rgba(0, 0, 0, 0.25);
  border-radius: 5px;
  padding: 10px;
  font-size: 14px;
  background-color: #ffffff;
  border: none;
  outline: none;
}
```

```
.form select {
  width: 100%;
  height: 40px;
  font-size: 12px;
  box-shadow: 0px 0px 6px rgba(0, 0, 0, 0.25);
  border-radius: 5px;
  padding: 10px;
  font-size: 14px;
  background-color: #ffffff;
  border: none;
  outline: none;
}
```

```
.form .d input {
  width: 100%;
  height: 35px;
  font-size: 12px;
```

```
box-shadow: 0px 0px 6px rgba(0, 0, 0, 0.25);
border-radius: 5px;
padding: 10px;
font-size: 14px;
background-color: #ffffff;
border: none;
outline: none;
}
```

```
.formreport .name input {
  color: #ffffff;
  font-weight: 500;
  font-size: 20px;
  line-height: 35px;
  letter-spacing: 0.07em;
  width: 100%;
  outline: none;
  margin-right: 5px;
}
```

```
#form-group1 {
  margin-top: 50px;
}
```

```
/* Стили для кнопки */
.add {
  display: flex;
  flex: 1;
  align-items: end;
  justify-content: space-between;
}
```

```
.add1 img {
  width: 60px;
  height: 60px;
  object-fit: cover;
  background-color: #ff8017;
  padding: 13px;
  border-radius: 100%;
  transition: 0.3s;
  margin-bottom: 12px;
  margin-top: 12px;
  margin-right: 50px;
  cursor: pointer;
}
```

```
.add1 img:hover {
  background-color: #ff7f17a3;
}
```

```
.done button {
  width: 160px;
```

```
height: 38px;
border: none;
font-size: 18px;
cursor: pointer;
background-color: #ef791f;
border-radius: 20px;
margin-bottom: 20px;
margin-top: 9px;
color: white;
margin-left: 50px;
}

.done button:hover {
border: 2px solid #ef791f; /* Товщина, стиль і колір бордєру */
background-color: #ffffff;
color: #ef791f;
}

/* Стили для футєра */
.footer {
display: flex;
justify-content: space-around;
align-items: center;
background-color: #001e5a;
color: #ffffff;
height: 100px;
font-weight: 400;
font-size: 15px;
line-height: 24px;
margin-top: auto; /* Футєр завжди внизу */
}

.footer .adres,
.pr {
padding-left: 20px;
}

.footer span {
padding-left: 17px;
}

.footer .location {
display: flex;
flex-direction: row;
width: 340px;
}

.footer .imgloc {
display: flex;
align-items: center;
}
```

```
.footer .imgloc img {
  width: 23px;
  height: 23px;
}

.footer .location .adres {
  display: flex;
  align-items: center;
}

.footer .phone {
  display: flex;
  flex-direction: column;
}

.footer .phone1,
.phone2 {
  display: flex;
  flex-direction: row;
}

.phone img {
  width: 15px;
  height: 13px;
}

.email {
  display: flex;
  flex-direction: column;
}

.email .insta {
  display: flex;
  flex-direction: row;
}

.email .em {
  display: flex;
  flex-direction: row;
}

.email .img2 img {
  width: 19px;
  height: 13px;
}

.email .img1 {
  width: 19px;
  height: 19px;
}

a {
```



```

text-decoration: none; /* Прибирає підкреслення */
color: #ffffff;
}

/* Підкреслення для посилань при наведенні курсора */
a span:hover {
text-decoration: underline;
}

/* Для модального вікна */
.modal {
display: none; /* Приховане за замовчуванням */
position: fixed;
z-index: 1;
left: 0;
top: 0;
width: 100%;
height: 100%;
background-color: rgba(0, 0, 0, 0.4);
}

.modal-content {
background-color: #fefefe;
margin: 5% auto;
padding: 20px;
width: 80%;
max-width: 550px;
box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.2);
animation: showModal 0.3s;
}

@keyframes showModal {
from {
transform: translateY(-50px);
opacity: 0;
}
to {
transform: translateY(0);
opacity: 1;
}
}

.cl {
display: flex;
flex-direction: row;
align-items: center;
justify-content: space-between;
}

.modal .close {
color: #aaa;
font-size: 68px;
}

```

```
font-weight: bold;
}

.modal .close:hover,
.modal .close:focus {
  color: black;
  text-decoration: none;
  cursor: pointer;
}

.modal .form-group {
  margin-bottom: 20px;
}

.donnn {
  display: flex;
  justify-content: center;
}

.modal button {
  padding: 10px 20px;
  background-color: #007bff;
  color: #fff;
  border: 2px solid #007bff;
  cursor: pointer;
  border-radius: 20px;
  font-size: 17px;
}

.modal button:hover {
  background-color: #ffffff;
  border: 2px solid #007bff;
  color: #007bff;
}

/* Медіа-запит для адаптивності */
@media (max-width: 768px) {
  .header {
    height: 70px;
  }

  .header p {
    font-weight: 600;
    padding-left: 20px;
    font-size: 18px;
    letter-spacing: 1px; /* більший інтервал між буквами */
  }

  .header .img {
    padding-left: 30px;
  }
}
```

```
.header .img img {
  height: 40px;
  width: 40px;
}

.footer {
  height: 162px;
  font-weight: 600;
  font-size: 10px;
  line-height: 18px;
  margin-top: auto; /* Футер завжди внизу */
}

.footer .adres,
.pr {
  padding-left: 10px;
}

.footer span {
  padding-left: 7px;
}

.footer .location {
  width: 150px;
}

.footer .imgloc img {
  width: 18px;
  height: 18px;
}

.footer .phone {
  margin: 20px;
}

.phone img {
  width: 11px;
  height: 9px;
}

.email .img2 img {
  width: 15px;
  height: 9px;
}

.email .img1 {
  width: 15px;
  height: 15px;
}

.ma {
  display: flex;
```

```
flex-direction: column;
flex: 1;
}

.main {
display: flex;
flex-direction: column;
flex: 1;
margin-top: 70px;
}

/* Стили для меню сайту */
.menu {
display: flex;
flex-direction: row;
justify-content: space-between; /* !!!! якщо потрібно без фото тоді END */
align-items: center;
width: 100%;
height: 80px;
background-color: #e9f8ff;
padding-left: 20px;
padding-right: 20px;
}

.photoperson {
padding-top: 0px;
}

.menunumber ul {
display: flex;
flex-direction: row;
padding-top: 22px;
height: 80px;
}

.menu li {
height: 50px;
padding-left: 9px;
margin-left: 15px; /* Відступ між пунктами */
border-left: none; /* Додаємо лінію між пунктами */
border-bottom: none; /* Додаємо лінію між пунктами */
}

.adapt {
display: flex;
flex-direction: column;
align-items: center;
flex: 1;
margin-top: 20px;
}

.adapt .name {
```

```
    box-shadow: 0px 10px 30px rgba(48, 46, 46, 0.142);  
}
```

```
.formreport {  
    min-height: 490px;  
    height: auto;  
    margin: 15px;  
}
```

```
.formreport .name {  
    min-height: 100px;  
    height: auto;  
}
```

```
.formreport .name p {  
    font-size: 15px;  
}
```

```
.form .d {  
    margin: 20px;  
}
```

```
.form .form-group {  
    margin-left: 5px;  
    margin-right: 5px;  
    margin-bottom: 25px;  
}
```

```
.form label {  
    font-weight: 500;  
    font-size: 12px;  
    line-height: 15px;  
    letter-spacing: 0.07em;  
    margin-bottom: 5px;  
    text-align: justify;  
    width: 190px;  
    height: 45px;  
    align-items: center;  
    margin-right: 15px;  
}
```

```
.form select {  
    width: 100%;  
    height: 45px;  
    font-size: 10px;  
}
```

```
.form .d input {  
    width: 100%;  
    height: 45px;  
    padding: 10px;  
    font-size: 10px;
```

```
}

textarea {
  font-size: 10px;
}

#form-group1 {
  margin-top: 50px;
}

/* Стили для кнопки */
.add {
  width: 100%;
  display: flex;
  flex: 1;
  align-items: end;
  justify-content: space-between;
}

.add1 img {
  width: 60px;
  height: 60px;
  object-fit: cover;
  background-color: #ff8017;
  padding: 13px;
  border-radius: 100%;
  transition: 0.3s;
  margin-bottom: 12px;
  margin-top: 12px;
  margin-right: 50px;
  cursor: pointer;
}

.add1 img:hover {
  background-color: #ff7f17a3;
}

.done button {
  width: 160px;
  height: 38px;
  border: none;
  font-size: 18px;
  cursor: pointer;
  background-color: #ef791f;
  border-radius: 20px;
  margin-bottom: 20px;
  margin-top: 9px;
  color: white;
  margin-left: 10px;
  margin-right: 50px;
}
```

```
.done button:hover {  
  border: 2px solid #ef791f; /* Товщина, стиль і колір бордеру */  
  background-color: #ffffff;  
  color: #ef791f;  
}  
}
```