

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«До захисту допущено»

В.о. завідувача кафедри

_____ Ігор Шелехов

_____ 18 грудня _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня магістр

зі спеціальності 122 - Комп'ютерних наук,

освітньо-професійної програми «Інформатика»

на тему: «Інформаційна технологія управління будівельними процесами»

здобувача групи ІН.м -24 Жукова Сергія Миколайовича

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

(підпис)

Сергій Жуков

Керівник,
доцент, кандидат технічних наук

Зоя Маслова

(підпис)

Суми – 2023

Сумський державний університет

Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«Затверджую»

В.о. зав. кафедри

_____ Ігор Шелехов

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня магістра

зі спеціальності 122 - Комп'ютерних наук, освітньо-професійної програми «Інформатика»
здобувача групи ІН.м-24 Жукова Сергія Миколайовича

1. Тема роботи: «Інформаційна технологія управління будівельними процесами»
затверджую наказом по СумДУ від «06» грудня 2023 р. № 1412-VI
2. Термін здачі здобувачем кваліфікаційної роботи «18» _____ грудня _____ 2023 р.
3. Вхідні дані до кваліфікаційної роботи результати роботи інформаційної технології
4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)
1) Аналіз проблеми предметної області, постановка й формування завдань дослідження.
2) Проектування інформаційної технології управління будівельними процесами.
3) Розробка інформаційної технології. 4) Аналіз результатів.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
актуальність проблеми, мета проекту, постановка задачі, функціональні вимоги до
проекту, контекстна діаграма IDEF0, декомпозиція IDEF0, схема інформаційної
технології, діаграма варіантів використання, фізична модель даних, засоби реалізації,
практична реалізація, висновки
6. . Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання _____.

Завдання прийняв до виконання _____
(підпис)

Керівник _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	<i>Аналіз проблеми предметної області, постановка й формування завдань дослідження</i>	до 18.11.2023	
2	<i>Огляд технологій, що використовуються для реалізації інформаційної технології</i>	до 22.11.2023	
3	<i>Реалізація інформаційної технології</i>	до 02.12.2023	

4	<i>Завершення</i>	до 13.12.2023	
5	<i>Оформлення документації</i>	до 18.12.2023	

Здобувач вищої освіти

(підпис)

Керівник

(підпис)

АНОТАЦІЯ

Записка: 71 стр., 39 рисунків, 4 додатки, 22 використаних джерела.

Обґрунтування актуальності теми роботи - Тема "Інформаційна технологія управління будівельними процесами" надзвичайно актуальна в сучасному будівельному секторі, оскільки впровадження інноваційних інформаційних технологій може значно покращити ефективність та якість будівництва. Це дозволяє автоматизувати процеси управління проектами, зменшувати час виконання робіт, а також сприяти оптимізації ресурсів і забезпеченню високого рівня координації між учасниками будівельного процесу. Враховуючи постійний розвиток будівельних технологій та вимоги до сталого будівництва, інформаційні технології в цьому контексті визначають перспективи подальшого прогресу та ефективного управління будівельними проектами.

Мета роботи – розробка інформаційної системи управління будівельними процесами з використанням технології мережевого планування.

Методи дослідження – алгоритм технології мережевого графіку та організація аналізу будівельних процесів.

Результати – визначено функціональні та нефункціональні вимоги до інформаційної технології управління будівельними процесами. Виходячи з вимог визначені програмні засоби реалізації та функціонал програмного забезпечення. Виконана розробка програмного забезпечення. Результати роботи програмного забезпечення продемонстровані на тестовому прикладі.

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ, JAVASCRIPT, PHP, MYSQL,
МЕРЕЖЕВЕ ПЛАНУВАННЯ.

ЗМІСТ

ВСТУП	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	9
1.1 Огляд останніх досліджень та публікацій	9
1.2 Аналіз програмних продуктів – аналогів	11
1.3 Постановка задачі	14
1.4 Аналіз технологій та засобів реалізації web-системи	16
2 МОДЕЛЮВАННЯ WEB-ДОДАТКУ	25
2.1 Структурно-функціональне моделювання	25
2.2 Моделювання варіантів використання	27
2.3 Проектування бази даних	28
3 РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ	33
3.1 Реалізація методу мережевого планування	33
3.2 Реалізація бази даних та інтеграція в інформаційну технологію	35
3.3 Реалізація інформаційної технології управління будівельними процесами	39
3.4 Приклад роботи інформаційної технології	43
ВИСНОВКИ	50
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	51
ДОДАТОК А	54
ДОДАТОК Б	60
ДОДАТОК В	65
ДОДАТОК Г	70

ВСТУП

На сьогоднішній день розвиток технологій, безпеки та швидкості інтернету значно розширив потенційні можливості веб-систем. Сьогодні ми маємо веб-системи бухгалтерського обліку бізнесу, веб-системи CRM, веб-версію Microsoft Office та багато іншого. Веб-додатки мають низку значних переваг над нативним, клієнтським програмним забезпеченням. Ось лише деякі з переваг веб-додатків для бізнесу.

Веб-інформаційна система базується на різних типах інтернет-протоколів і відіграє важливу роль у збиранні, зберіганні, передачі та поширенні інформації. Коли мова йде про соціальну поведінку та організацію, веб-технології мають вплив. Вони використовуються бізнесом для розвитку свого бізнесу та управління своїми операціями по всьому світу.

Важливість веб-систем управління будівельними процесами полягає в їх можливостях вирішення великої кількості завдань і проблем, які виникають у будівельній галузі. Ці системи стають важливими інструментами для оптимізації та вдосконалення управління будівельними проектами.

По-перше, мережеві системи дозволяють ефективно обмінюватися інформацією між усіма учасниками будівельного процесу, незалежно від їхнього розташування. Це сприяє командній роботі та покращує спілкування між архітекторами, інженерами, клієнтами та іншими учасниками проекту.

По-друге, використання мережевих систем дозволяє ефективно розраховувати ресурси, витрати та терміни. Це економить час і ресурси і дозволяє вчасно реагувати на зміни в процесі будівництва.

По-третє, ці системи дають можливість використовувати аналітичні інструменти для оцінки та підвищення ефективності стратегій управління проектами. Аналіз даних може допомогти виявити слабкі місця та прогалини в зусиллях і внести корективи на ранніх етапах реалізації проекту.

Актуальність. Використання веб-систем управління будівництвом обумовлено їх здатністю покращувати зв'язок, оптимізувати використання

ресурсів і надавати аналітичні інструменти для ефективного управління будівельними проектами. Оскільки технології та вимоги ринку швидко змінюються, використання таких систем стає ключовою умовою успішної реалізації будівельних проектів і конкурентної переваги.

Об'єкт дослідження. Процес управління будівельними процесами, що включає в себе планування, виконання та контроль етапів будівництва, а також взаємодію між учасниками будівельного процесу.

Предмет дослідження. Створення web-орієнтованої інформаційної технології, яка сприятиме оптимізації управління будівельними процесами, полегшить обмін даними між учасниками будівельного процесу та забезпечить оперативний моніторинг ходу будівництва.

Гіпотеза. Розробка та впровадження web-орієнтованої інформаційної технології сприятиме значному поліпшенню управління будівельними процесами, зменшить час виконання робіт та підвищить загальну якість будівництва.

Метою дослідження є створення web-орієнтованої інформаційної технології управління будівельними процесами, для досягнення якої необхідно виконати наступні задачі:

- визначити функціональні та нефункціональні вимоги;
- провести аналіз програмних продуктів-аналогів;
- визначити технології для реалізації;
- розробити архітектуру web-орієнтованої системи;
- виконати розробку бази даних;
- виконати етап проектування web-орієнтованої системи;
- виконати практичну реалізацію web-системи.

Новизна. Розробка web-орієнтованої інформаційної технології управління будівельними процесами актуальна через необхідність забезпечення ефективної комунікації та співпраці між всіма учасниками будівельного проекту, що розташовані на власних посадах. Дана система надасть можливості моніторингу процесу виконання замовлення, витрат та

покращити стратегії управління, забезпечуючи більш точне та ефективне виконання будівельних завдань. Враховуючи швидкі технологічні зміни та конкурентну природу галузі будівництва, впровадження web-орієнтованих систем стає ключовим фактором успіху в цій області.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд останніх досліджень та публікацій

З останніми технологічними досягненнями веб-сервіс стає все більш популярною технологією в сучасному світі. Це програмний сервіс, який використовується для зв'язку між двома електронними пристроями через мережу. Веб-сервіс популярний у веб-додатках завдяки своїй природній властивості бути самодостатнім і самоописуваним. У цій статті ми обговоримо переваги веб-сервісів і те, як вони допомагають у роботі програми.

Веб-сервіс - це організований код, який можна викликати і активувати віддалено за допомогою HTTP-запитів. Веб-сервіси дозволяють розробнику публікувати функціональність свого коду для решти світу. Після того, як його буде видно в мережі, існує ймовірність того, що інші програмісти використають цю функціональність у своїх додатках [1].

Ще однією перевагою веб-сервісу є те, що інтероперабельність має найвищий пріоритет. Мета інтероперабельності веб-сервісів полягає в тому, щоб дозволити програмним додаткам спілкуватися один з одним і обмінюватися даними або послугами між собою. Наприклад, додатки .NET можуть взаємодіяти з веб-сервісами Java і навпаки. Веб-сервіси використовуються для забезпечення незалежності технології та платформ додатків.

Таким чином, веб-сервіси пропонують численні переваги, такі як розкриття існуючих функцій в мережі, інтероперабельність і зниження витрат. Всі переваги веб-сервісів детально розглянуті вище. У GeoDataSource доступні такі веб-сервіси, як веб-сервіс пошуку місцезнаходження та веб-сервіс сусідніх країн. Веб-сервіс пошуку місцезнаходження дозволяє користувачеві шукати інформацію про міста, надаючи координати місцезнаходження у вигляді широти і довготи, в той час як веб-сервіс сусідніх країн дозволяє

користувачеві перевіряти наявність сусідніх країн на основі введення коду країни у форматі ISO3166-2.

Популярність веб-систем управління будівельними процесами зумовлена кількома ключовими факторами, які відповідають зростаючим вимогам і тенденціям сучасної будівельної галузі.

Простий і доступний віддалений доступ до інформації став ключовим фактором успішного управління будівельними проектами. Веб-системи полегшують комунікацію та співпрацю між керівниками проектів, підрядниками та іншими учасниками проекту, оскільки доступ до даних та обмін ними можливий з будь-якого місця через Інтернет.

Висока мобільність і гнучкість веб-систем ідеально підходять для сучасного динамічного будівельного середовища. Вони розширюють можливості учасників проекту та полегшують їхню роботу, дозволяючи аналізувати результати виконаних процесів з будь-якого місця [2].

Підвищення ефективності та точності управління ресурсами та фінансами в процесі будівництва є ключовим. Веб-системи надають інтегровані інструменти для розрахунку витрат, планування робіт і взаємодії з підрядниками, а також допомагають оптимізувати бюджети і терміни реалізації проектів.

Реалізована система надасть інструменти для виконання операцій з моніторингом будівельного процесу а також виконувати облік працівників з розподілом ролей. Для більш кращого відображення надано графічне та візуальне оформлення що в поєднанні з інтуїтивно зрозумілим інтерфейсом допоможе користувачу знаходити всю необхідну для себе інформацію. Загалом система надає наступні можливості:

- додавання нового виконання проекту;
- встановлення кінцевої дати виконання проекту;
- додавання працівників та розподіл ролей між ними;
- визначення всіх підпроцесів та створення діаграми для візуального відображення результату;

- встановлення бюджету проекту на кожному підпроцесі;
- можливість зміни ролей в залежності від задачі.

1.2 Аналіз програмних продуктів – аналогів

Для більш детального розуміння сучасних трендів та особливостей web-орієнтованих систем було обрано дві web-системи, які схожі за тематикою та призначенням:

- CRMSTROY;
- OpenBoxCRM;

CRMSTROY – сучасна CRM система яка надає можливість виконувати автоматизацію бізнес-процесів, управлінням проектами та базою даних клієнтів. Дана система розроблена на основі самих передових технологій, що дозволяє їй бути лідером з різноманітності функціоналу, якості продуктивності, стабільності роботи, дружелюбності інтерфейсу та багатьох інших параметрів. [3] За допомогою даної CRM системи користувач може досягати максимальних результатів у функціонуванні всіх елементів бізнесу, значно збільшуючи прибуток та зменшуючи можливі витрати. Переваги сервісу CRMSTROY наступні:

- дозволяє співробітникам підрахувати попередню вартість будівництва для будь-якого виду будівельних робіт. Завдяки гнучкій системі налаштування можна швидко та легко розрахувати навіть найскладніші проекти. Для зручності є низка готових шаблонів;
- відображення фінансового стану компанії. Існують функції показу всіх вхідних та вихідних фінансових операцій, графіка за рік, грошових витрат за останні кілька днів, сальдо за кожен місяць, а також багато інші;

- можливість додавання клієнтів різними способами: вручну, через форми захоплення на сайті або за допомогою JivoSite. У картці клієнта міститься вся докладна інформація про клієнта;
- опції перегляду вартості, вибору клієнта з бази, розрахованого кошторису, дати початку та закінчення будівництва.

На рисунку 1.1 представлено вигляд інтерфейсу системи CRMSTROY.

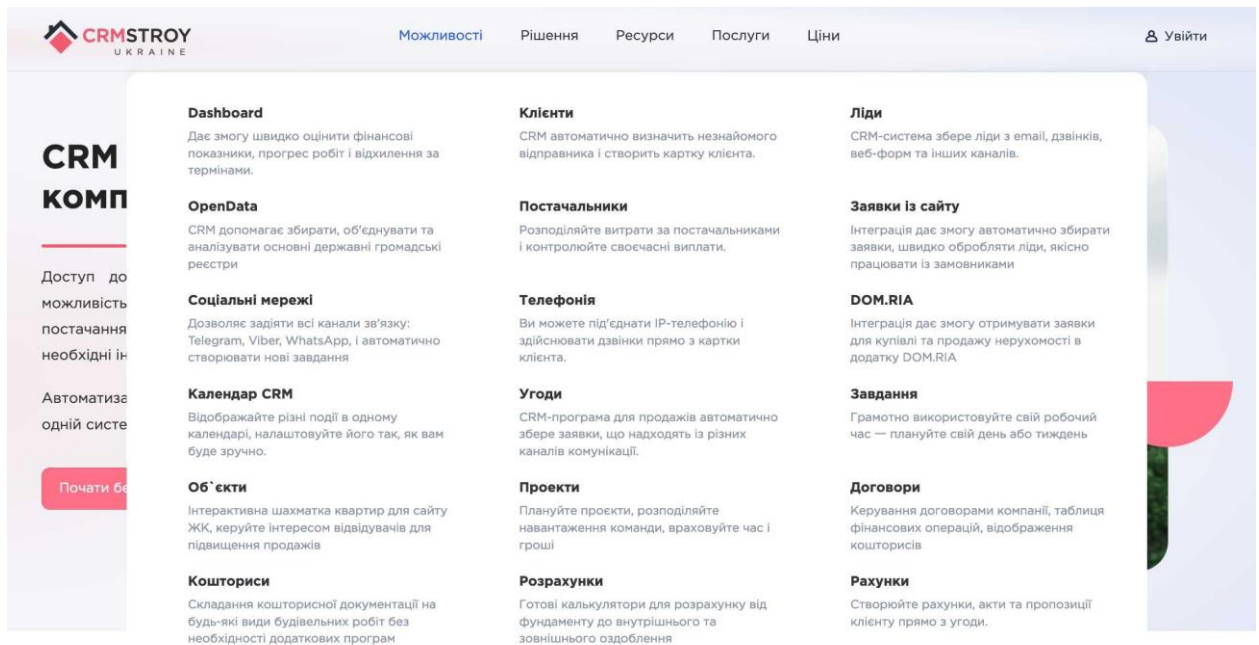


Рисунок 1.1 – Вигляд інтерфейсу системи CRMSTROY

OpenVoxCRM – типова CRM система, яка містить всі необхідні інструменти для управління будівельними процесами. Дана система має інструменти для планування проектів, ведення обліку ресурсів, витрат та строків виконання робіт, а також для співпраці з підрядниками та іншими учасниками. Також дана система забезпечує ефективний обмін інформацією між різними учасниками проекту, такими як архітектори, інженери, замовники та інші сторони. Інтуїтивно зрозумілий інтерфейс та адаптивність під всі актуальні пристрої надають комфортний процес керування процесом [4].

Також система може легко інтегруватися з іншими інструментами, такими як облік робочого часу, фінансовий облік, або іншими спеціалізованими програмами, які використовуються в будівельній галузі.

Головним компонентом даної системи є надання належної підтримки користувачів а також можливість для навчання задля ефективного використання всіх можливостей даної системи

На рисунку 1.2 представлено інтерфейс web-системи OpenBoxCRM

Категория	Товар	Серийный номер	Количество	Зарезервировано	Количество склад Днепро	Количество склад Киев	Количество склад Чернигов	Закупочная цена	Стоимость продажи
	LCD телевизор BRAVIS LED-32D2000		#1.000	#0.000			1.000	0.000	4388.50
	LCD телевизор BRAVIS LED-28D1070		#2.000	#0.000	2.000			0.000	7812.00
	LCD телевизор BRAVIS LED-40D2000		#3.000	#0.000		3.000		3.400	22696.50
LCD телевизоры	LCD телевизор Elenberg 32AH4030		#2.000	#0.000			2.000	5.000	8727.00
LCD телевизоры	LCD телевизор BRAVIS LED-1615		#3.000	#0.000	3.000			20.000	7227.00
LCD телевизоры	LCD телевизор BRAVIS LED-24D2000		#4.000	#0.000			4.000	10.000	13410.00
LCD телевизоры	LCD телевизор Elenberg 40AH4010		#1.000	#0.000			1.000	0.000	7169.50
LCD телевизоры	LCD телевизор Elenberg 22AH4010		#3.000	#0.000			3.000	5.000	8997.00

Рисунок 1.2 - інтерфейс web-системи OpenBoxCRM

Для більш детального аналізу аналогів було створено таблицю 1.1 для порівняння web-сервісів

Таблиця 1.1 - Порівняння web-сервісів для управління будівельними процесами

Критерій	CRMSTROY	OpenBoxCRM
Функціональність будівельного менеджменту	Доступність інструментів для планування проектів, обліку ресурсів, витрат та строків виконання робіт.	Забезпечення інструментів для ефективного планування та керування будівельним процесом.
Модуль комунікації та співпраці	Система дозволяє ефективний обмін інформацією між різними учасниками проекту.	Забезпечення можливості ефективної комунікації та співпраці між учасниками будівельного процесу.

Продовження табл. 1.1

Мобільність та доступність	Можливість ведення управлінської діяльності з різних пристроїв та з будь-якого місця.	Забезпечення мобільності та доступності для ефективного управління будівельними процесами.
Аналітичні засоби	Інструменти для аналізу даних та виявлення можливостей для оптимізації процесів та зменшення витрат.	Надання аналітичних засобів для ефективного управління та виявлення слабких місць.
Інтеграція з іншими системами	Легка інтеграція з іншими інструментами, такими як облік робочого часу, фінансовий облік.	Забезпечення можливості інтеграції з іншими спеціалізованими програмами для будівельної галузі.
Підтримка клієнтів та навчання	Наявність підтримки клієнтів та можливостей для навчання користувачів.	Забезпечення належної підтримки та навчання для ефективного використання системи.

Враховуючи всі переваги сервісів які описані вище, розроблена web-система управління будівельними процесами має надавати можливість користувачу швидко отримувати інформацію та інтуїтивно зрозумілий інтерфейс. Також важливим аспектом сервісу є захист особистих даних користувача.

1.3 Постановка задачі

За допомогою аналізу аналогів було досліджено головні переваги даних систем, які можна використати задля власної реалізації. Web-орієнтована система управління будівельними процесами повинна бути реалізована як web-додаток, який буде доступний для всіх актуальних систем та пристроїв в мережі Інтернет.

Web-система повинна надавати наступні можливості:

- авторизація за особистими даними;

- перегляд всіх актуальних будівельних проектів;
- наявність графічного відображення процесу роботи;
- додавання власного проекту та створення підпроцесів;
- додавання працівників та призначення ролей;
- надання проектам дедлайни за допомогою методу мережевого планування;

Даний метод базується на моделі, що описує проект як мережу робіт, і включає такі відомі методи, як метод критичного шляху (CPM) і метод оцінки та аналізу програм (PERT). CPM визначає критичний шлях, який включає так звані критичні роботи (роботи, що заслуговують на максимальну увагу, оскільки будь-яка затримка призводить до відтермінування дати завершення проекту), в той час як PERT оцінює розподіл ймовірностей дати завершення проекту. Це метод вимагає визначення кожної з робіт, що входять до складу будівельного проекту, послідовності їх виконання, а також робіт, які можуть виконуватися одночасно з іншими роботами [5].

Web-технологія повинна містити адміністративну частину, яка надає доступ за допомогою унікального логіну та паролю користувача та реалізована за допомогою клієнт-серверної архітектури. Також авторизований користувач отримує можливість виконувати операції щодо власного проекту, працівників та встановлювати власні дати щодо виконання.

Для адміністратора необхідно реалізувати інтуїтивно зрозумілий інтерфейс, який надаватиме швидке завантаження сторінок з даними.

Для успішної реалізації web-системи необхідно вирішити наступні завдання:

- провести детальний огляд сервісів-аналогів які надають аналогічні послуги та визначити головні переваги кожного;
- визначити функціональні можливості, такі як додавання підпроцесів, користувачів та проектів;
- реалізувати клієнт-серверну архітектуру;
- визначити технології для реалізації web-системи;

- реалізувати структуру, функціонал та дизайн web-системи;
- створити фізичну модель бази даних, визначити сутності та зв'язки, а також обрати систему управління базою даних та інтегрувати створену модель в web-систему.

1.4 Аналіз технологій та засобів реалізації web-системи

Для реалізації даної web-системи було обрано для аналізу 3 види архітектур:

- cloud [6];
- object-oriented [7];
- client-server [8].

В таблиці 1.2 представлено порівняльну характеристику архітектур.

Аспект	Хмарна архітектура	Об'єктно-орієнтована архітектура	Клієнт-серверна архітектура
Основний принцип	Система розташована в хмарі, а доступ до ресурсів та послуг надається через мережу.	Орієнтована на об'єкти структура, де дані та функціональність згруповані в об'єкти.	Розділення обов'язків між клієнтами (кінцевими користувачами) та серверами, які надають послуги або ресурси.
Масштабованість	Забезпечує легку масштабованість, оскільки ресурси можна додавати або зменшувати в хмарному середовищі.	Масштабованість залежить від об'єктів та їх взаємодії, що може потребувати перегляду архітектури при	Масштабованість реалізується через збільшення кількості серверів чи покращення їх продуктивності.

Аспект	Хмарна архітектура	Об'єктно-орієнтована архітектура	Клієнт-серверна архітектура
		зміні обсягів даних або функціональності.	
Розташування даних	Дані можуть бути розподілені в різних хмарних вузлах, а доступ до них може здійснюватися з будь-якого місця.	Дані можуть бути збережені в об'єктах, і доступ до них здійснюється через виклики методів цих об'єктів.	Дані можуть знаходитися на сервері, а клієнти отримують доступ до них через мережу.
Залежність від мережі	Велика залежність від якості мережі для доступу до хмарних ресурсів та послуг.	Об'єкти можуть взаємодіяти локально, але при зовнішньому виклику методу потрібен доступ до мережі.	Залежність від мережі присутня, особливо при обміні даними між клієнтами та серверами.
Вартість	Хмарні послуги можуть бути витратними, особливо при великому обсязі даних та інтенсивному використанні ресурсів.	Зазвичай менш витратна, оскільки базується на розподіленому об'єктному підході.	Вартість може бути помірною, але може зростати при збільшенні обсягів використання серверних ресурсів.

Після детального аналізу кожної з архітектур було вирішено використовувати клієнт-серверну архітектуру по наступним перевагам:

Централізація

Основна перевага мережі клієнт-сервер полягає в тому, що вона забезпечує централізоване управління - всю інформацію можна знайти в одному місці. Оскільки адміністратор мережі має повний контроль над управлінням та адмініструванням, це надзвичайно вигідно. Будь-яка проблема, що виникає у всій мережі, може бути вирішена в одному місці. Також стає простіше оновлювати дані [9].

Адаптивність

Клієнти можуть легко розширити кількість клієнтів або серверів. Оскільки сервер централізований, немає жодних проблем з авторизацією до мережеских ресурсів, що збільшуються в розмірі. Як результат, для налаштування потрібна лише невелика кількість людей.

Захист

Завдяки централізованій структурі мережі клієнт-сервер, дані належним чином захищені. Тільки якщо користувач авторизований, він може отримати доступ до даних за допомогою логіна та пароля і двофакторної автентифікації. Крім того, якщо дані втрачено, записи можна швидко відновити за допомогою однієї резервної копії.

Робота з файлами

Працювати з файлами просто, оскільки всі вони зберігаються на одному сервері. Клієнт-серверна мережа може просто відстежувати та отримувати доступ до необхідних файлових записів.

Для практичної реалізації web-орієнтованої системи було обрано наступні технології для аналізу:

- MERN;
- PHP + MySQL;
- C++.

MERN

Стек MERN - це поєднання перевірених і найсучасніших методів для створення масштабованих основних програмних додатків для інтерфейсу, бекенду та систем обробки даних. Платформи на основі JavaScript - єдині, що дозволяють створювати динамічні веб-сторінки та додатки. Як результат, компанії прагнуть використовувати її в якості платформи [10].

MERN Stack є спеціалізацією кількох компаній-розробників програмного забезпечення. Він включає в себе передове програмне забезпечення, таке як MongoDB, Express, React JS та Node.js.

Стек MERN має багато переваг. Нижче наведені деякі з найважливіших переваг, які допоможуть вам зрозуміти їх. Подивіться на переваги створення веб-додатків з використанням стека MERN.

MVC

Однією з головних переваг стеку MERN є архітектура Model-View-Controller, яку він пропонує, і те, наскільки просто розробникам створювати онлайн-програми. Одним з найважливіших аспектів MERN-додатків є те, що вони розроблені для розповсюдження з використанням архітектури мікросервісів, в якій кожен мікросервіс ідеально незалежний від інших і залежить лише від декількох вибраних інших сервісів. MERN дозволяє програмістам створювати неймовірно надійні програми.

Повний стек

Завдяки інструментам MERN Stack програмісти можуть писати програми швидше і з меншими зусиллями. Розробники можуть ефективніше концентрувати свій час на вирішенні проблем, створенні нових функцій та вдосконаленні продукту, усуваючи складність викликів API.

Стек MERN, мікросервісна архітектура, дозволяє описувати сервіси окремо. Кожен сервіс у стеку MERN складається з однієї задачі. Стек MERN є чудовою парадигмою архітектурного дизайну для програм будь-якого розміру та складності. Багато програм можуть використовувати стек MERN.

Легкість у вивченні

Стек MERN складається з фреймворку, системи виконання та мови програмування. MERN - це мова програмування, яка поєднує онлайн та офлайн світи. Таким чином, веб-розробники можуть створювати найпотужніші внутрішньо-браузерні додатки без вікна браузера. Для запуску MERN-додатків тепер можна використовувати ефективний і швидкий виконавчий вузол Node на основі JavaScript. Розробники стеку MERN повинні добре знати JavaScript та JSON, щоб використовувати переваги стеку MERN під час розробки онлайн-додатків.

Обслуговування коду

Використовуючи стек MERN, розробники можуть створювати додатки, якими простіше керувати, ніж звичайними веб-додатками. Це архітектура MERN з декількома рівнями. Це масштабоване рішення, яке дозволяє як збільшувати, так і зменшувати масштаби без шкоди для цілісності коду. Це значно економить час і гроші.

Сильна спільнота

Стек MERN стає таким відомим з часом завдяки своїй функціональності, яка об'єднує фронтенд, бек-енд та базу даних. Завдяки його популярності, зростаюча спільнота робить крок вперед, щоб підтримати розробників у їхньому пошуку ідеальної відповіді.

Гнучкість

Використання архітектури MERN має багато переваг, одна з яких полягає в тому, що вона дозволяє використовувати компоненти технологічного стеку з відкритим вихідним кодом, запобігаючи проблемам з ліцензіями.

Додатки, що потребують надійного розділення даних і бізнес-логіки, особливо добре підходять для MERN. Платформа MERN спрощує код для інтерфейсу, бек-енду та бази даних окремо, що є значною перевагою.

PHP + MySQL

PHP і MySQL часто використовуються разом для створення динамічних веб-систем, керованих базами даних. Наприклад, веб-сайт, який дозволяє користувачам створювати облікові записи, залишати коментарі або переглядати персоналізований контент, зазвичай використовує базу даних для зберігання та отримання необхідних даних. PHP використовується для взаємодії з базою даних і створення веб-сторінок, які відображаються користувачеві на основі даних, що зберігаються в базі даних [11].

Спільне використання PHP та MySQL має кілька переваг:

Відкритий вихідний код

І PHP, і MySQL - це програмне забезпечення з відкритим вихідним кодом, що означає, що їх можна вільно використовувати і поширювати. Це робить їх економічно ефективним рішенням для розробки веб-додатків. Широке використання: PHP - одна з найпопулярніших мов програмування для створення динамічних веб-сторінок, а MySQL - найпопулярніша СУБД.

Надійний захист

PHP та MySQL мають вбудовані функції безпеки, які допомагають захиститися від поширених веб-вразливостей, таких як атаки на введення SQL-коду. Масштабованість: І PHP, і MySQL добре масштабуються, що робить їх придатними для використання в додатках, які отримують великий обсяг трафіку.

Сумісність

PHP сумісний з широким спектром операційних систем, веб-серверів і баз даних, що полегшує його розгортання і запуск. MySQL також сумісна з різними платформами, що робить її гнучким вибором для зберігання даних.

Продуктивність

PHP та MySQL розроблені для швидкої та ефективної роботи, що робить їх придатними для використання у високопродуктивних додатках.

C++

Однією з причин, чому C++ ігнорують у веб-розробці, є те, що її традиційно пов'язують із системним програмуванням або розробкою ігор. Проте, коли ми занурюємося в деталі, стає зрозуміло, що C++ має такі атрибути, як швидкість та ефективність, які є золотими і для веб-розробки. Якщо вам цікаво, як саме C++ вписується в ширший ландшафт мов програмування, обов'язково прочитайте статтю про те, хто такий C++ розробник [12].

Швидкість та ефективність

C++ - це скомпільована мова, яка транслюється безпосередньо в машинний код, оминаючи потребу в інтерпретаторі. Це призводить до блискавичної швидкості виконання. Коли ваш сайт працює швидше, користувачі щасливіші, а алгоритми ранжування Google це помічають.

Гнучкість і контроль

C++ надає розробникам більший рівень контролю над системними ресурсами порівняно з такими мовами, як Python або JavaScript. Цей контроль може призвести до більш ефективного розподілу ресурсів, що може мати вирішальне значення для веб-додатків, які вимагають високої продуктивності.

Для більш детального аналізу технологій було створено порівняльну таблицю 1.3.

Таблиця 1.3 – Порівняльна таблиця технологій

Аспект	MERN	PHP + MySQL	C++
Мова програмування	JavaScript (Node.js, React.js)	PHP, JavaScript (в частині клієнтської сторони)	C++
База даних	MongoDB (NoSQL)	MySQL (SQL)	Зазвичай використовується окремо від бази даних, але може бути використаний для розробки back-end частини.

Продовження табл. 1.3

Серверна технологія	Node.js та Express.js	Apache, Nginx (для PHP), PHP (back-end)	Може використовуватися в різних типах застосунків, включаючи web, але потребує власної реалізації серверної частини.
Клієнтська технологія	React.js (JavaScript)	HTML, CSS, JavaScript (може використовуватися разом з фреймворками або бібліотеками)	Може використовуватися для створення клієнтської частини, але не є основною технологією для цього.
Спрощення розробки	Має велику екосистему та готові бібліотеки, що спрощує розробку.	PHP є простою мовою для вивчення, але може бути менш ефективною для великих проєктів.	Мова програмування C++ вважається потужною, але може бути складнішою для вивчення та розробки.
Масштабованість	Має добре розвинуту систему масштабування, зокрема завдяки Node.js та MongoDB.	Може масштабуватися за допомогою горизонтального та вертикального масштабування, але може виникнути складнощі з великим обсягом даних.	Масштабування C++ зазвичай визначається власноруч, і воно може бути більш складним.

Для реалізації поставленої задачі було обрано технологію PHP + MySQL.

PHP та MySQL є найпопулярнішими мовами програмування баз даних та скриптів з відкритим вихідним кодом у сучасному тренді. PHP - це серверна

скриптова мова, що дозволяє створювати значні динамічні сторінки з індивідуальними функціями. Однією з головних переваг використання PHP та MySQL є те, що вони забезпечують інтерактивний, зручний для користувача веб-сайт, а також дозволяють відвідувачам відкрито взаємодіяти, створюючи по-справжньому динамічний та гнучкий контент. PHP дуже легко вивчити порівняно з іншими мовами програмування, і його можна легко налаштувати відповідно до конкретних бізнес-потреб клієнтів [13].

Також для створення web-системи та перегляду готового результату необхідно використати наступні продукти:

- Visual Studio Code 2023 – OpenSource IDE для написання коду з використанням плагінів та розширень;
- OpenServer 2023 – Локальний сервер для перевірки кінцевого результату;
- Firefox, Google Chrome для перевірки адаптивності інтерфейсу.

2 МОДЕЛЮВАННЯ WEB-ДОДАТКУ

2.1 Структурно-функціональне моделювання

Існує велика різноманітність технік діаграм, таких як блок-схеми та діаграми потоків даних, але жодна з них не є строго визначеною і не є запатентованою.

IDEF0 був розроблений на основі Structured Analysis and Design Technique (SADT), добре відомої графічної мови. Ефективні моделі IDEF допомагають спростити розробку пристроїв і сприяють ефективній співпраці між аналітиками та замовниками [14].

IDEF0 корисна для визначення обсягу аналізу, особливо для функціонального аналізу. Як механізм співпраці, IDEF0 використовує спрощені інструменти візуалізації для полегшення залучення галузевих експертів і спільного прийняття рішень.

Як аналітичний інструмент, IDEF0 дозволяє моделювальнику визначати функції. Функції включають в себе виконання необхідних дій, перевірку правильності поточної системи та перевірку помилок існуючої системи.

На діаграмах IDEF0 "блок і стрілка" функція відображається у вигляді блоку, а інтерфейси - у вигляді стрілок, що входять або виходять з блоку до функції або з неї.

На рисунку 2.1 представлено контекстну діаграму процесу створення будівельного проекту.

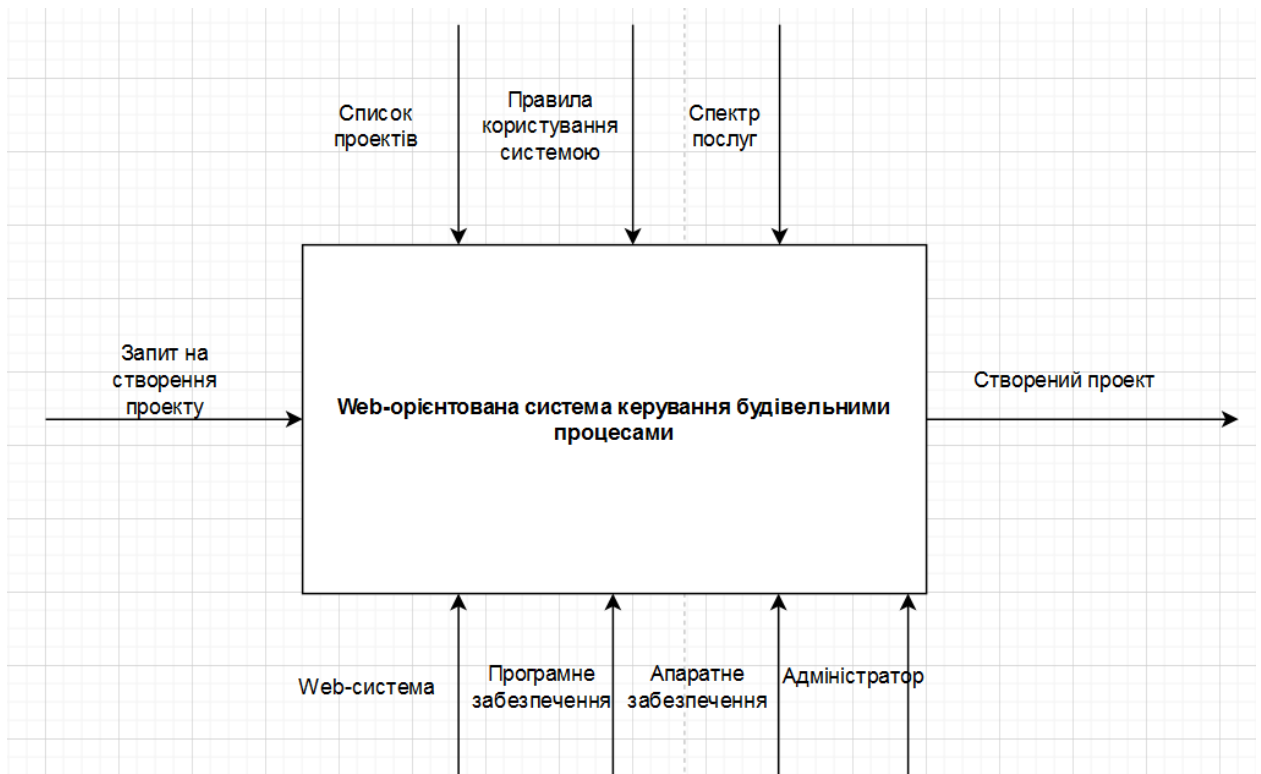


Рисунок 2.1 - Контекстна діаграма процесу створення будівельного проекту

Функціональна декомпозиція - це метод, який використовується для розробки детальної структури компонентів або модулів програмного забезпечення. Функціональна декомпозиція визначає функції, діяльність, процеси або дії, які повинен виконувати компонент або модуль програмного забезпечення [15].

На рисунку 2.2 представлено функціональну декомпозицію процесу створення будівельного проекту.

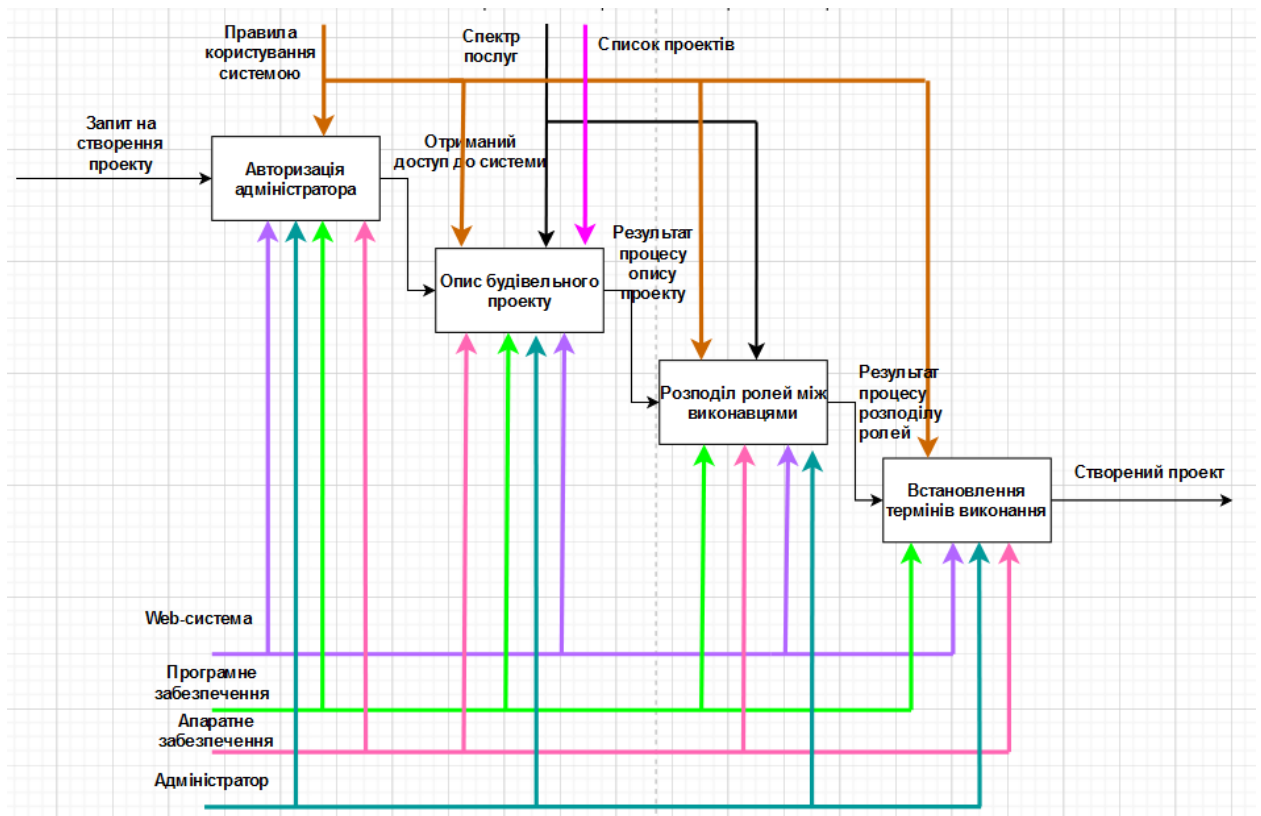


Рисунок 2.2 - Функціональна декомпозиція процесу створення будівельного проекту

2.2 Моделювання варіантів використання

Діаграма варіантів використання UML - це поведінкова діаграма, яка відображає спосіб, у який користувачі можуть взаємодіяти з системою. Щоб створити діаграму варіантів використання, необхідно створити основні типи користувачів, які також називаються акторами. Наступним етапом необхідно включити варіанти використання, розмістивши їх у вигляді кола або еліпсів. Діаграма варіантів використання буде завершена, коли буде з'єднано акторів з відповідними варіантами використання, а також виділено розширені зв'язки [16].

Важливо зазначити, що діаграма варіантів використання не фокусується на порядку дій, тому вона не відображає їх послідовність. Більше того, діаграма варіантів використання не заглиблюється в деталі, оскільки вона має

на меті дати вам загальне уявлення про систему. Насправді, вона працює як план, який може нагадувати вам про цілі, які потрібно тримати в голові.

На рисунку 2.3 представлено діаграму варіантів використання web-орієнтованої системи управління будівельними процесами.

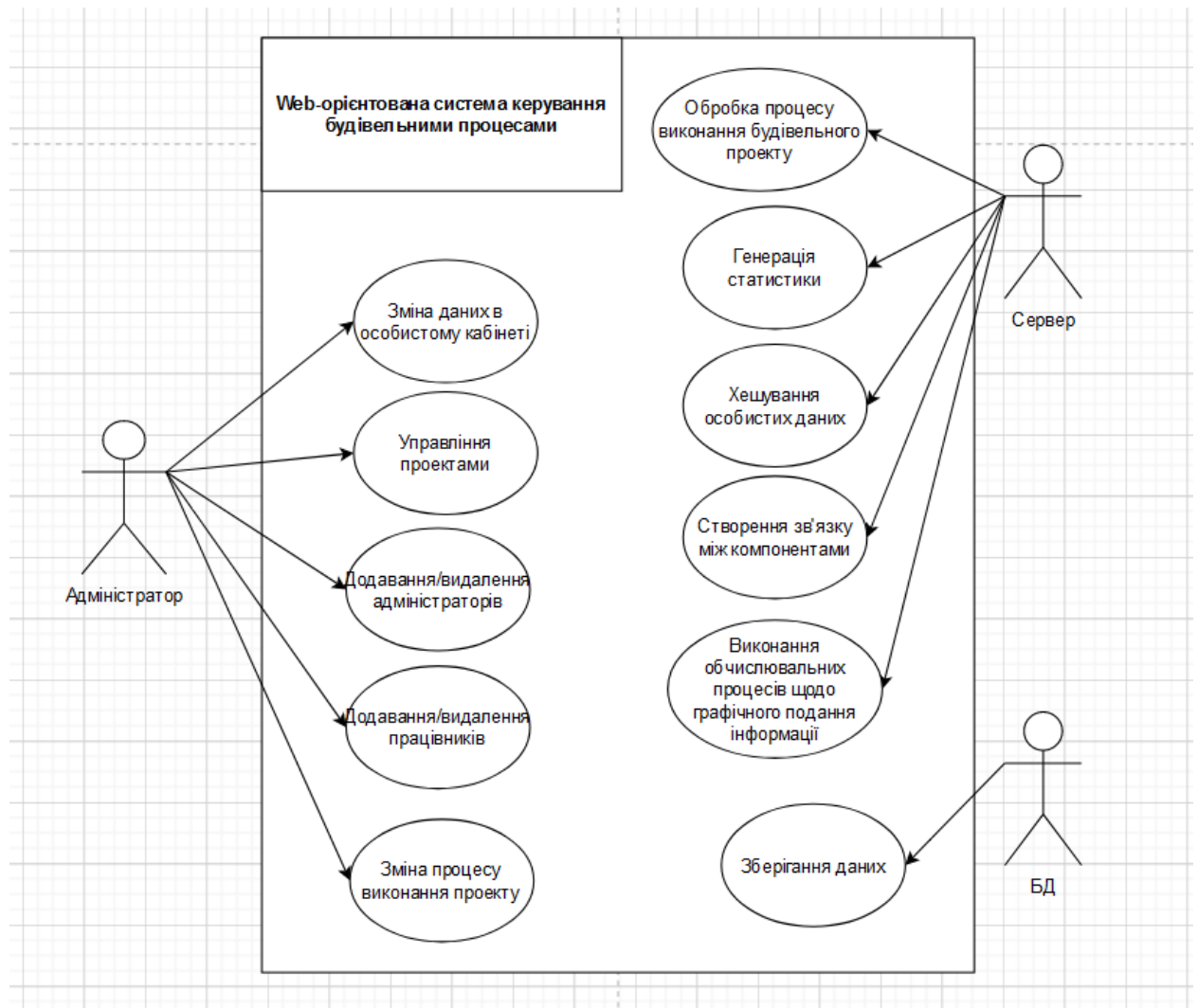


Рисунок 2.3 - Діаграма варіантів використання web-орієнтованої системи управління будівельними процесами

2.3 Проектування бази даних

MySQL - це СУБД з відкритим вихідним кодом або система управління реляційними базами даних. Дані створюються, змінюються та витягуються з реляційної бази даних за допомогою SQL або мови структурованих запитів. А

дані, включаючи типи даних, пов'язані один з одним, організуються в таблиці даних реляційною базою даних [17].

Хоча MySQL здебільшого використовується з іншими програмами, вона також має окремі клієнти. Більше того, багато популярних веб-сайтів, а також веб-додатків, керованих базами даних, використовують MySQL.

Для більш детального представлення переваг MySQL було створено таблицю 2.1.

Таблиця 2.1 – Переваги у використанні MySQL

Критерій	Детальне пояснення
Безпека даних	MySQL використовується популярними веб-додатками, такими як Drupal, Joomla, WordPress та популярними веб-сайтами, такими як Facebook і Twitter. Крім того, вона має функції безпеки для запобігання несанкціонованому доступу до конфіденційних даних і підходить для бізнесу, особливо для тих, хто потребує частих грошових переказів.
Масштабованість	MySQL пропонує масштабованість на вимогу, що може бути корисним у зв'язку зі зростанням обсягів даних і технологій. Таким чином, користувач може збільшувати або зменшувати свої вимоги в міру необхідності. Це полегшує управління додатками незалежно від обсягу даних.

Продовження табл. 2.1

Висока продуктивність	Структура механізму зберігання даних, що використовується в MySQL, дозволяє системним адміністраторам створювати високопродуктивні, але бездоганні сервери баз даних. За допомогою цього користувач може обробляти велику кількість запитів, які може отримати бізнес, і при цьому забезпечувати оптимальну швидкість.
Підтримка транзакцій	MySQL надає комплексну підтримку транзакцій з декількома функціями підтримки транзакцій. Вона включає блокування на рівні рядків, транзакції бази даних з ACID або атомарністю, узгодженість, ізоляцію та довговічність, підтримку багатOVERсійних транзакцій.
Контроль робочого процесу	MySQL має крос-платформні можливості, а час, необхідний для її завантаження та встановлення, відносно невеликий. Тому її можна використовувати одразу після завершення інсталяції, незалежно від того, чи це Windows, Linux, macOS чи інша системна платформа.

Продовження табл. 2.1

Зниження загальних витрат	MySQL пропонує надійність і легко керуваність завдяки багатьом функціям. Як результат, заощаджуються час і гроші, що витрачаються на пошук і усунення несправностей, усунення простоїв або проблем з продуктивністю, що знижує загальні витрати.
Гнучкість	MySQL полегшує налагодження, оновлення та обслуговування, а також покращує досвід кінцевого користувача. Крім того, він дозволяє вам налаштовувати конфігурації на свій розсуд, що робить його гнучкою системою управління базами даних з відкритим вихідним кодом.

На рисунку 2.4 представлено фізичну модель бази даних web-орієнтованої системи управління будівельними процесами, яка була реалізована за допомогою web-додатку phpMyAdmin.

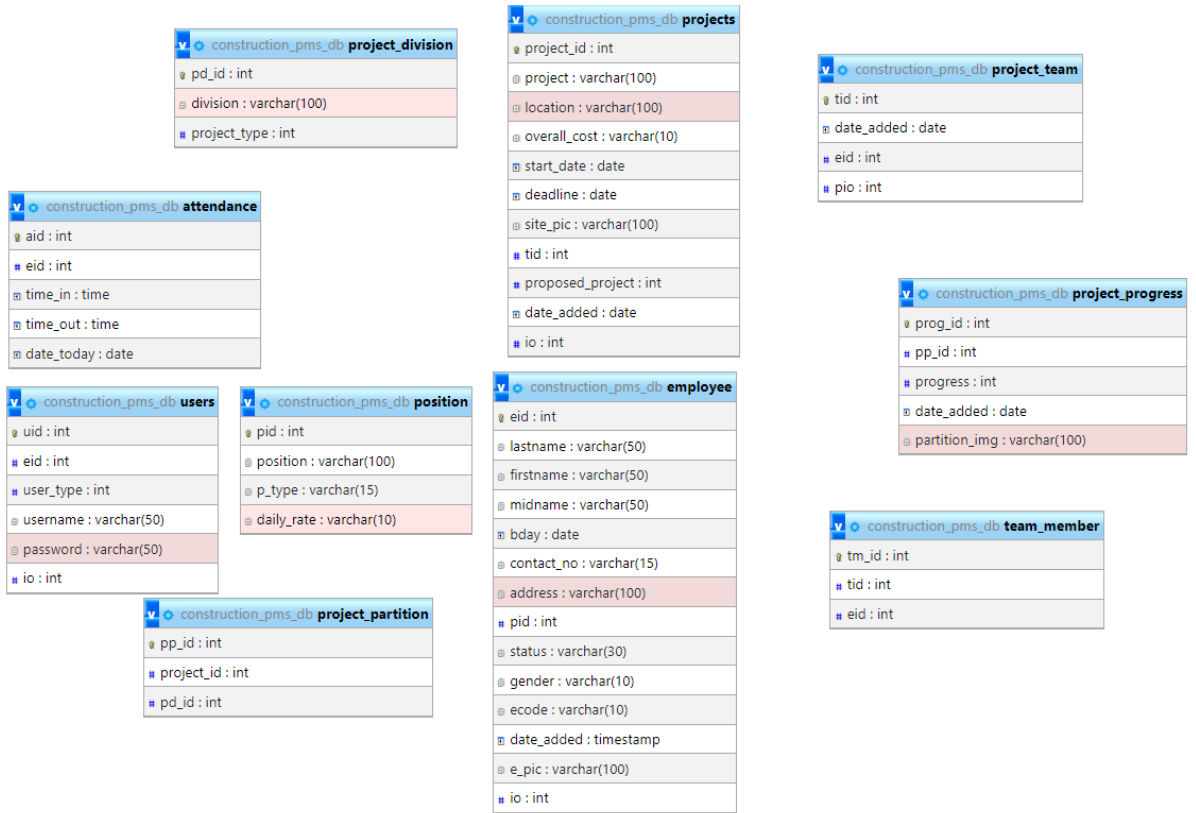


Рисунок 2.4 - Фізична модель бази даних web-орієнтованої системи управління будівельними процесами

3 РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ

3.1 Реалізація методу мережевого планування

Графік виконання проекту складається з поетапним перерахуванням у послідовному порядку робіт, пов'язаних з реалізацією проекту. Кроки мають бути чітко визначені разом із зазначенням часу, необхідного для виконання кожного кроку.

Мережеве планування - це категоризація дій, пов'язаних із реалізацією проекту, у послідовному порядку з подальшим схематичним представленням дій, необхідних для всього проекту.

Мережеве планування інформаційної технології складається з наступних етапів:

- 1) Необхідно визначити категорії заходів, що виконуються від початку до завершення будівельного проекту. Дії повинні бути згруповані за категоріями, які відрізняються одна від одної.
- 2) Розташувати список заходів, як зазначено в пункті , у порядку їх виконання. Може існувати діяльність, яку можна розпочати тільки після завершення будь-якої іншої діяльності, тоді як може існувати й інша незалежна діяльність, яку можна розпочати одночасно.
- 3) Використовуючи деталі А і В, наведені вище, необхідно візуально відобразити схему мережі дій, щоб можна було візуалізувати оперативне планування виконання всього будівельного проекту.

У мережевому аналізі подія вважається реалізованою, коли всі дії, що ведуть до події, завершено, і, оскільки дії виконують від однієї події до іншої, попередня подія називається "хвостовою подією", а подальша подія - "головна подія".

Дія починається з події, тобто "хвостової події", а дія після завершення певної роботи переходить до іншої події, головної події. Отже, жодна дія не може розпочатися, доки не буде реалізована хвостова подія, за винятком

найпершої дії, яка розпочинається з події номер один і яка, звісно, не має жодної хвостової події

Коли в результаті однієї події виникає більше однієї дії, така подія називається "вибуховою подією". Коли кілька дій завершуються в одній події, така подія називається "подією злиття".

Для більш чіткого розуміння процесу виконання, на рисунку 3.1 представлено візуальне відображення вибухової події та події злиття.

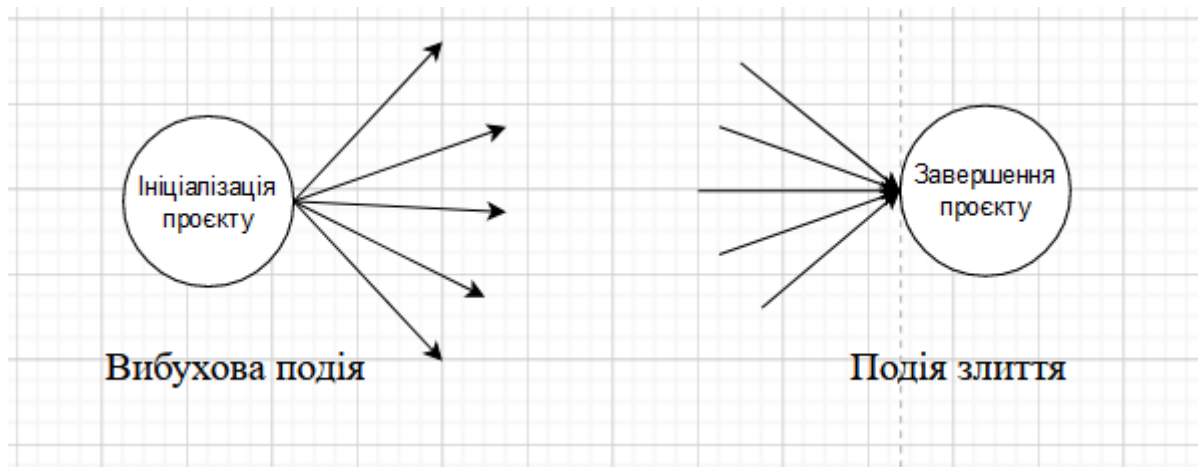


Рисунок 3.1 – Візуальне відображення вибухової події та події злиття

Виникає ситуація, коли завершення однієї задачі залежить від завершення якоїсь іншої задачі, а не тільки тієї, що їй передує. У таких випадках пов'язані події об'єднуються пунктирною стрілкою, що позначає нереальну дію.

На рисунку 3.2 показано фіктивну діяльність (4)-(2), яка насправді не є діяльністю сама по собі. Показано, як установити логіку на діаграмі, коли діяльність (2)-(5) залежить від завершення дій (1)-(2), а також (3)-(4). Однак дія (4)-(6) не залежить від дії (1)-(2), але, звісно, може бути розпочатою лише тоді, коли дія (3)-(4) завершена й подія (4) була реалізована.

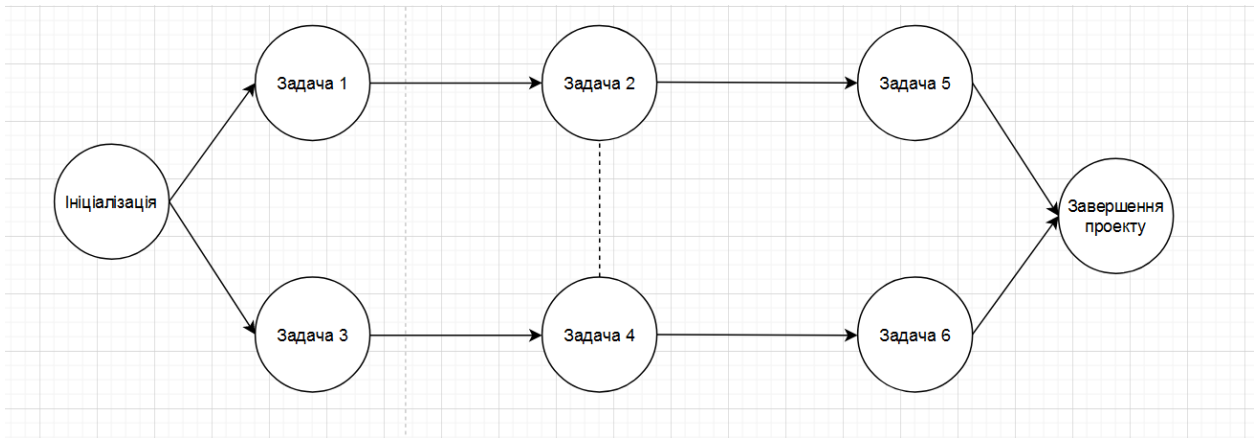


Рисунок 3.2 – Фіктивна діяльність (4)-(2)

Таким чином, мережу в таких випадках можна показати, розбиваючи кожну з цих дій на початок, прогрес і (передаючи завершену частину другому процесу, який потім може розпочатися) продовжувати хід і (передаючи завершену частину другому процесу).) і так далі, поки не буде виконаний кожний процес до кінця.

На рисунку 3.3 представлено діаграму послідовності виконання процесів.

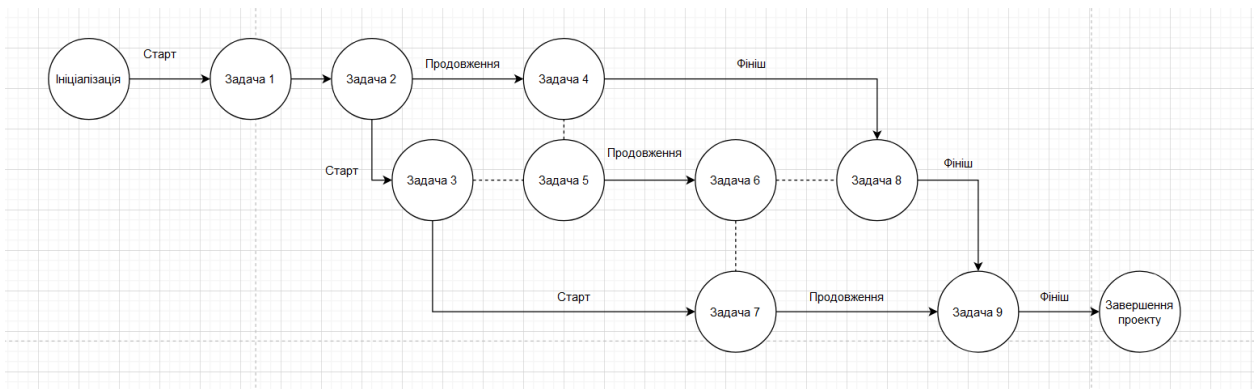


Рисунок 3.3 – Діаграма послідовності виконання процесів

3.2 Реалізація бази даних та інтеграція в інформаційну технологію

Для реалізації бази даних було використано OpenServer для локального серверу та MySQL. Для початку необхідно авторизуватися в системі phpMyAdmin. Після цього необхідно створити базу даних з назвою «construction_pms_db».

Тепер необхідно створити таблиці та надати атрибути з унікальним id. На рисунках 3.4 – 3.13 представлено результати створення таблиць з атрибутами.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/> 1	aid 🔑	int			Нет	Нет		AUTO_INCREMENT	Ещё ▾
<input type="checkbox"/> 2	eid	int			Нет	Нет			Ещё ▾
<input type="checkbox"/> 3	time_in	time			Нет	Нет			Ещё ▾
<input type="checkbox"/> 4	time_out	time			Нет	Нет			Ещё ▾
<input type="checkbox"/> 5	date_today	date			Нет	Нет			Ещё ▾

Рисунок 3.4 – Видяд таблиці «attendance»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/> 1	eid 🔑	int			Нет	Нет		AUTO_INCREMENT	Ещё ▾
<input type="checkbox"/> 2	lastname	varchar(50)	latin1_swedish_ci		Нет	Нет			Ещё ▾
<input type="checkbox"/> 3	firstname	varchar(50)	latin1_swedish_ci		Нет	Нет			Ещё ▾
<input type="checkbox"/> 4	midname	varchar(50)	latin1_swedish_ci		Нет	Нет			Ещё ▾
<input type="checkbox"/> 5	bday	date			Нет	Нет			Ещё ▾
<input type="checkbox"/> 6	contact_no	varchar(15)	latin1_swedish_ci		Нет	Нет			Ещё ▾
<input type="checkbox"/> 7	address	varchar(100)	latin1_swedish_ci		Нет	Нет			Ещё ▾
<input type="checkbox"/> 8	pid	int			Нет	Нет			Ещё ▾
<input type="checkbox"/> 9	status	varchar(30)	latin1_swedish_ci		Нет	Нет			Ещё ▾
<input type="checkbox"/> 10	gender	varchar(10)	latin1_swedish_ci		Нет	Нет			Ещё ▾
<input type="checkbox"/> 11	ecode	varchar(10)	latin1_swedish_ci		Нет	Нет			Ещё ▾
<input type="checkbox"/> 12	date_added	timestamp			Нет	CURRENT_TIMESTAMP		DEFAULT_GENERATED	Ещё ▾

Рисунок 3.5 – Видяд таблиці «employee»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/> 1	pid 🔑	int			Нет	Нет		AUTO_INCREMENT	Ещё ▾
<input type="checkbox"/> 2	position	varchar(100)	latin1_swedish_ci		Нет	Нет			Ещё ▾
<input type="checkbox"/> 3	p_type	varchar(15)	latin1_swedish_ci		Нет	Нет			Ещё ▾
<input type="checkbox"/> 4	daily_rate	varchar(10)	latin1_swedish_ci		Нет	Нет			Ещё ▾

Рисунок 3.6 – Видяд таблиці «position»

Структура таблицы Связи

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 project_id	int			Нет	Нет		AUTO_INCREMENT	Ещё
<input type="checkbox"/>	2 project	varchar(100)	latin1_swedish_ci		Нет	Нет			Ещё
<input type="checkbox"/>	3 location	varchar(100)	latin1_swedish_ci		Нет	Нет			Ещё
<input type="checkbox"/>	4 overall_cost	varchar(10)	latin1_swedish_ci		Нет	Нет			Ещё
<input type="checkbox"/>	5 start_date	date			Нет	Нет			Ещё
<input type="checkbox"/>	6 deadline	date			Нет	Нет			Ещё
<input type="checkbox"/>	7 site_pic	varchar(100)	latin1_swedish_ci		Нет	Нет			Ещё
<input type="checkbox"/>	8 tid	int			Нет	Нет			Ещё
<input type="checkbox"/>	9 proposed_project	int			Нет	Нет			Ещё
<input type="checkbox"/>	10 date_added	date			Нет	Нет			Ещё
<input type="checkbox"/>	11 io	int			Нет	Нет			Ещё

Рисунок 3.7 – Видяд таблиці «projects»

Структура таблицы Связи

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 pd_id	int			Нет	Нет		AUTO_INCREMENT	Ещё
<input type="checkbox"/>	2 division	varchar(100)	latin1_swedish_ci		Нет	Нет			Ещё
<input type="checkbox"/>	3 project_type	int			Нет	Нет			Ещё

Рисунок 3.8 – Видяд таблиці «project_division»

Структура таблицы Связи

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 pp_id	int			Нет	Нет		AUTO_INCREMENT	Ещё
<input type="checkbox"/>	2 project_id	int			Нет	Нет			Ещё
<input type="checkbox"/>	3 pd_id	int			Нет	Нет			Ещё

Рисунок 3.9 – Видяд таблиці «project_partition»

Структура таблицы [Связи](#)

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/> 1	prog_id 🔑	int			Нет	Нет		AUTO_INCREMENT	Ещё ▾
<input type="checkbox"/> 2	pp_id	int			Нет	Нет			Ещё ▾
<input type="checkbox"/> 3	progress	int			Нет	Нет			Ещё ▾
<input type="checkbox"/> 4	date_added	date			Нет	Нет			Ещё ▾
<input type="checkbox"/> 5	partition_img	varchar(100)	latin1_swedish_ci		Нет	Нет			Ещё ▾

Рисунок 3.10 – Видяд таблиці «project_progress»

Структура таблицы [Связи](#)

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/> 1	tid 🔑	int			Нет	Нет		AUTO_INCREMENT	Ещё ▾
<input type="checkbox"/> 2	date_added	date			Нет	Нет			Ещё ▾
<input type="checkbox"/> 3	eid	int			Нет	Нет			Ещё ▾
<input type="checkbox"/> 4	pio	int			Нет	Нет			Ещё ▾

Рисунок 3.11 – Видяд таблиці «project_team»

Структура таблицы [Связи](#)

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/> 1	tm_id 🔑	int			Нет	Нет		AUTO_INCREMENT	Ещё ▾
<input type="checkbox"/> 2	tid	int			Нет	Нет			Ещё ▾
<input type="checkbox"/> 3	eid	int			Нет	Нет			Ещё ▾

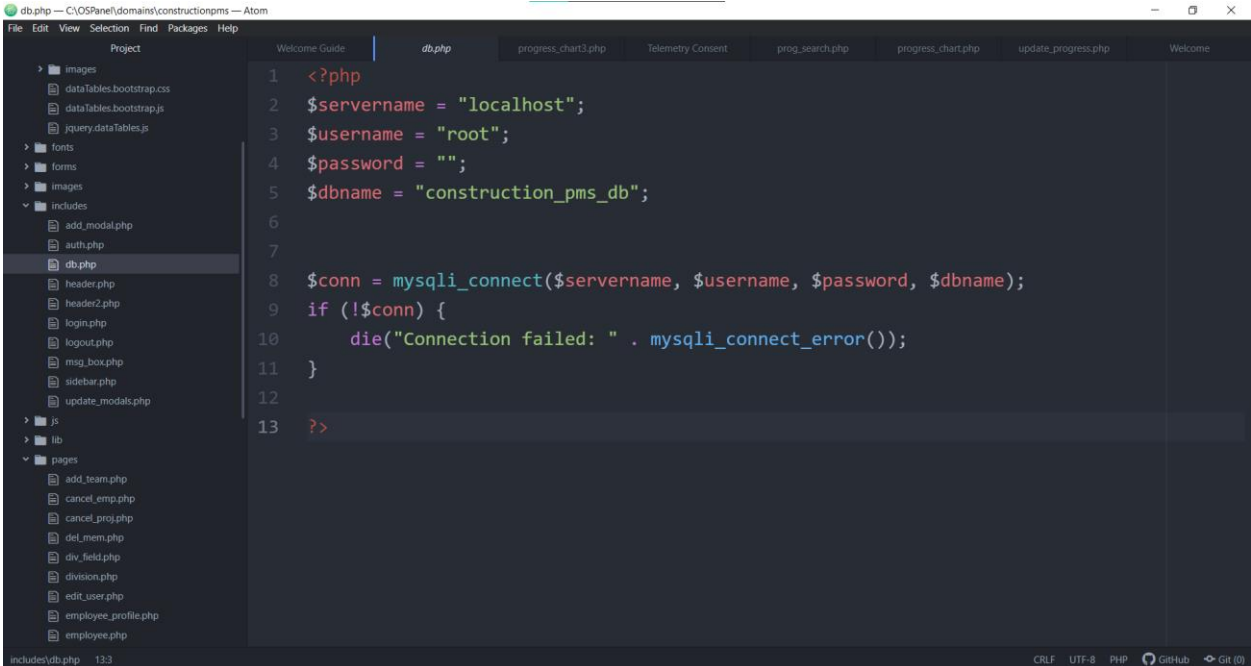
Рисунок 3.12 – Видяд таблиці «project_team»

Структура таблицы [Связи](#)

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/> 1	uid 🔑	int			Нет	Нет		AUTO_INCREMENT	Ещё ▾
<input type="checkbox"/> 2	eid	int			Нет	Нет			Ещё ▾
<input type="checkbox"/> 3	user_type	int			Нет	Нет			Ещё ▾
<input type="checkbox"/> 4	username	varchar(50)	latin1_swedish_ci		Нет	Нет			Ещё ▾
<input type="checkbox"/> 5	password	varchar(50)	latin1_swedish_ci		Нет	Нет			Ещё ▾
<input type="checkbox"/> 6	io	int			Нет	Нет			Ещё ▾

Рисунок 3.13 – Видяд таблиці «users»

Після створення бази даних відбувається інтеграція в систему. На рисунку 3.14 представлено результат підключення БД до інформаційної системи.



```

1  <?php
2  $servername = "localhost";
3  $username = "root";
4  $password = "";
5  $dbname = "construction_pms_db";
6
7
8  $conn = mysqli_connect($servername, $username, $password, $dbname);
9  if (!$conn) {
10     die("Connection failed: " . mysqli_connect_error());
11 }
12
13 ?>

```

Рисунок 3.14 – Результат підключення БД до інформаційної системи

3.3 Реалізація інформаційної технології управління будівельними процесами

Для реалізації інформаційної технології було використано PHP для створення серверної частини додатку, JavaScript для створення скриптів, Bootstrap для стилів користувацького інтерфейсу. В якості бази даних було обрано MySQL.

Першим чином було реалізовано технологію авторизації користувача. Для захисту особистих даних було створено систему хешування особистих даних. На рисунку 3.15 представлено результат створення системи авторизації користувача в системі. Також після авторизації користувач може змінити особисті дані. Повний лістинг коду «user_edit.php» представлено в додатку Б.

```

68     </div>
69 </body>
70 <script>
71     jQuery(document).ready(function(){
72         jQuery(document).ready(function(){
73             $("#correct").hide();
74             $("#error").hide();
75             jQuery("#login_form").submit(function(e){
76                 e.preventDefault();
77                 var formData = jQuery(this).serialize();
78                 $.ajax({
79                     type: "POST",
80                     url: "includes/login.php",
81                     data: formData,
82                     success: function(html){
83                         if(html=="true" )
84                         {
85                             $("#error").hide();
86                             $("#correct").slideDown();
87                             var delay = 2000;
88                             setTimeout(function(){ window.location = 'pages/index.php?page=home'; }, delay);
89                         }else{
90                             $("#error").slideDown();
91                             var delay = 2000;
92                             setTimeout(function(){ $("#error").slideUp(); }, delay);
93                         }
94                     }
95                 });
96                 return false;
97             });
98         });
99     });
100
101 </script>

```

Рисунок 3.15 – Результат створення технології авторизації користувача в системі

Для автоматизації обчислювальних процесів щодо відображення прогресу виконання будівельного проекту було розроблено систему обчислення (рис 3.16). За допомогою неї адміністратор на основі отриманих даних про процес виконання кожного етапу зможе отримати графік успішності. Це допоможе в врегулюванні дедлайнів та виконанні додаткових планувань. Повний лістинг коду `progress_chart3.php` представлено в додатку В.


```

1 <?php
2 include '../includes/header.php';
3 $servername = "localhost";
4 $username = "root";
5 $password = "";
6 $dbname = "fpgs_rs";
7 $conn = new mysqli($servername, $username, $password, $dbname);
8 if ($conn->connect_error) {
9     die("Connection failed: " . $conn->connect_error);
10 }
11
12 $id = $_GET['id'];
13 $prog = $conn->query("SELECT * FROM project_partition natural join project_division where project_id = '$id' ");
14 while($progress = $prog ->fetch_assoc()){
15
16     $name = $progress['division'];
17     $pid = $progress['pp_id'];
18     $i= 1;
19     $prog3 = $conn->query("SELECT SUM(progress) as total_prog FROM project_progress where pp_id = '$pid' ");
20     $row_prog = $prog3->fetch_assoc();
21     if ($prog && $prog->num_rows > 0)
22     {
23
24         if($row_prog['total_prog'] <= 50){
25             $color='rgba(251, 159, 118, 0.53)';
26         }elseif ($count > 50 ) {
27             $color='rgba(120, 151, 239, 0.53)';
28         }
29         $array[$id][] = '{"progress":' . $row_prog['total_prog'] . ',' . '"name":' . ucfirst($name) . ',' . '"color":"' . $color . '"}';
30     }
31 }
32
33
34 }
35

```

Рисунок 3.16 – Результат створення системи обчислення

Також для інтуїтивно зрозумілого інтерфейсу було створено шапку та навігаційне меню інформаційної системи. На рисунках 3.17 – 3.18 представлено результати створення header та sidebar інформаційної системи.

```

1 <head>
2
3 <meta charset="utf-8">
4 <meta http-equiv="X-UA-Compatible" content="IE=edge">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <meta name="description" content="">
7 <meta name="author" content="">
8 <link rel="icon" href="images/logo.jpg">
9 <title>Project Management System</title>
10
11 <!-- Bootstrap Core CSS -->
12 <link href="..asset/css/bootstrap.min.css" rel="stylesheet">
13 <!-- Custom CSS -->
14 <link href="..asset/css/sb-admin.css" rel="stylesheet">
15
16 <!-- Morris Charts CSS -->
17 <link href="..asset/css/plugins/morris.css" rel="stylesheet">
18
19 <!-- Custom Fonts -->
20 <link href="..asset/font-awesome/css/font-awesome.min.css" rel="stylesheet" type="text/css">
21 <!-- am chart -->
22 <script type="text/javascript" src="..am_chart/amcharts.js"></script>
23 <script type="text/javascript" src="..am_chart/pie.js"></script>
24 <script type="text/javascript" src="..am_chart/none.js"></script>
25 <script type="text/javascript" src="..am_chart/serial.js"></script>
26 <script type="text/javascript" src="..am_chart/light.js"></script>
27 <script src="..am_chart/jquery.min.js"></script>
28 <style>
29 .chartdiv a {
30     display: none !important;
31 }
32 .chartdiv {
33     background-color: white;
34 }
35 </style>

```

Рисунок 3.17 – Результат створення шапки інформаційної системи

```

1 <ul class="nav navbar-nav side-nav">
2 <li>
3 <a href="index.php?page=home"><i class="fa fa-fw fa-dashboard"></i> Dashboard</a>
4 </li>
5
6 <li class="">
7 <a href="index.php?page=employee&i=1"><i class="fa fa-fw fa-users"></i> Employee List</a>
8 </li>
9 <li>
10 <a href="index.php?page=project_list&i=1"><i class="fa fa-fw fa-files-o"></i> Project List</a>
11 </li>
12 <?php if($SESSION['TYPE'] == 1): ?>
13 <li class="">
14 <a href="index.php?page=user_list&i=1"><i class="fa fa-users"></i> Users</a>
15 </li>
16 <?php endif; ?>
17 <li>
18 <a id="demo3" href="javascript:;" data-toggle="collapse" data-target="#demo3"><i class="fa fa-fw fa-gears"></i> Maintenance
19 <ul id="demo3" class="collapse">
20
21 <li>
22 <a href="index.php?page=position"><i class="fa fa-fw fa-file-o"></i>Position</a>
23 </li>
24 <li>
25 <a href="index.php?page=division"><i class="fa fa-fw fa-file-o"></i>Project Divisions</a>
26 </li>
27 <li style="">
28 <a href="index.php?page=project_team"><i class="fa fa-fw fa-users"></i>Project Team</a>
29 </li>
30 </ul>
31 </li>
32 </ul>
33 <script>
34 var _h = location.href

```

Рисунок 3.18 – Результат створення навігаційного меню інформаційної технології

Також було інтегровано технологію мережевого планування. Для цього за допомогою JavaScript було описано сценарії взаємодії функцій обчислення. При встановленні термінів виконання проекту технологія автоматично запропонує більш вигідну дату для завершення проекту. На рисунку 3.19 представлено результат інтеграції технології мережевого планування в інформаційну технологію. Повний лістинг коду надано в додатку Г.

```

24 "#####",containerBackgroundOpacity:["containerBackgroundOpacity",1],containerClassName:["containerClassName","fusioncharts-container"],chartType:["ty
25 0],baseChartMessageFont:["baseChartMessageFont","Verdana,sans"],baseChartMessageFontSize:["baseChartMessageFontSize","10"],baseChartMessageColor:["basec
26 100],baseChartMessageImageScale:["baseChartMessageImageScale",100],dataLoadStartMessage:["dataLoadStartMessage","Retrieving data. Please wait."],dataLo
27 supported:"");
28 loadMessages:["loadMessage","loading chart. Please wait."],renderErrorMessage:["renderErrorMessage","Unable to render chart."],attributes:{lang:["lang",
29 0],n:nb[a][1];
30 n:nb[a][1];
31 n:nb[a][1];
32 n:nb[a][1];
33 n:nb[a][1];
34 n:nb[a][1];
35 n:nb[a][1];
36 n:nb[a][1];
37 n:nb[a][1];
38 n:nb[a][1];
39 n:nb[a][1];
40 n:nb[a][1];
41 n:nb[a][1];
42 n:nb[a][1];
43 n:nb[a][1];
44 n:nb[a][1];
45 n:nb[a][1];
46 n:nb[a][1];
47 n:nb[a][1];
48 n:nb[a][1];
49 n:nb[a][1];
50 n:nb[a][1];
51 n:nb[a][1];
52 n:nb[a][1];
53 n:nb[a][1];
54 n:nb[a][1];
55 n:nb[a][1];
56 n:nb[a][1];
57 n:nb[a][1];
58 n:nb[a][1];
59 n:nb[a][1];
60 n:nb[a][1];
61 n:nb[a][1];
62 n:nb[a][1];
63 n:nb[a][1];
64 n:nb[a][1];
65 n:nb[a][1];
66 n:nb[a][1];
67 n:nb[a][1];
68 n:nb[a][1];
69 n:nb[a][1];
70 n:nb[a][1];
71 n:nb[a][1];
72 n:nb[a][1];
73 n:nb[a][1];
74 n:nb[a][1];
75 n:nb[a][1];
76 n:nb[a][1];
77 n:nb[a][1];
78 n:nb[a][1];
79 n:nb[a][1];
80 n:nb[a][1];
81 n:nb[a][1];
82 n:nb[a][1];
83 n:nb[a][1];
84 n:nb[a][1];
85 n:nb[a][1];
86 n:nb[a][1];
87 n:nb[a][1];
88 n:nb[a][1];
89 n:nb[a][1];
90 n:nb[a][1];
91 n:nb[a][1];
92 n:nb[a][1];
93 n:nb[a][1];
94 n:nb[a][1];
95 n:nb[a][1];
96 n:nb[a][1];
97 n:nb[a][1];
98 n:nb[a][1];
99 n:nb[a][1];
100 n:nb[a][1];

```

Рисунок 3.19 – Результат інтеграції технології мережевого планування

3.4 Приклад роботи інформаційної технології

Перш ніж почати використовувати інформаційну технологію, користувачу необхідно виконати авторизацію в системі. На рисунку 3.20 представлено вигляд сторінки авторизації. Для успішного входу в систему необхідно ввести пароль «admin» та логін «admin».

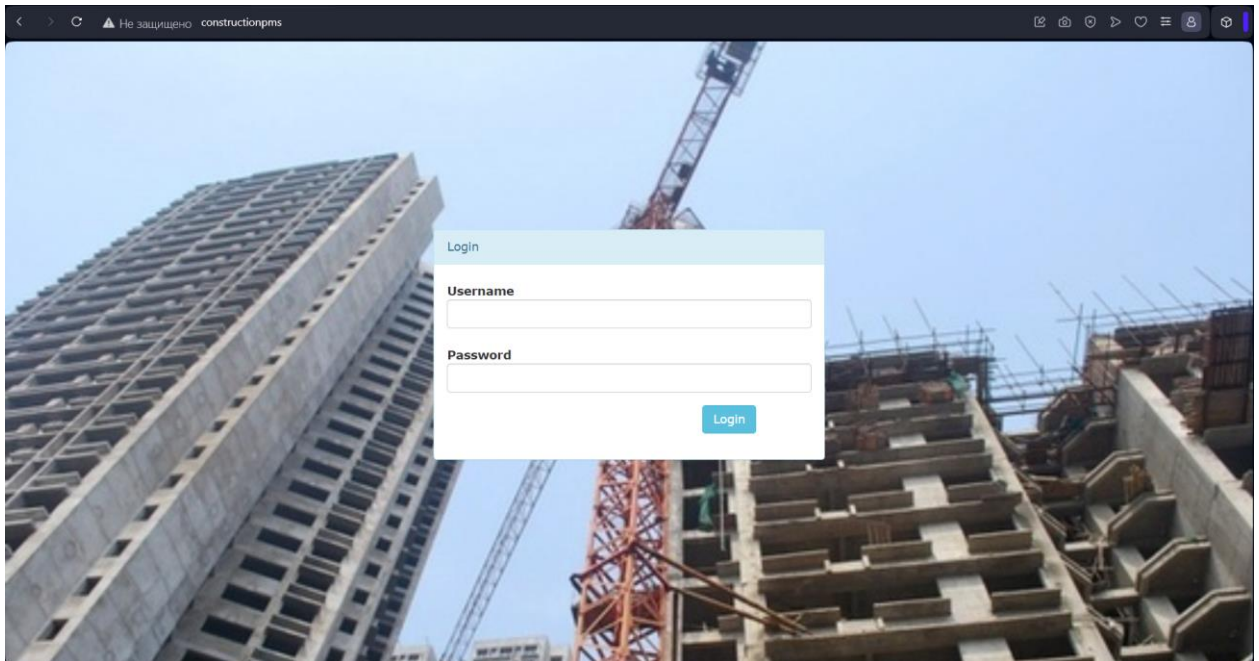


Рисунок 3.20 – Вигляд сторінки авторизації

Дана інформаційна технологія організовує записи проектів будівельної компанії зі статусом кожної частини категорії розробки, такої як планування, ландшафт і т.д. Ця функція є динамічною для цієї системи, що означає, що користувач системи може керувати категорією або списком підрозділів. Технологія управління проектами також керує командами/працівниками для кожного проекту. Відображення прогресу кожної категорії/підрозділу в кожному проекті також буде управлятися користувачем системи, оскільки ця функція допоможе компанії відстежувати прогрес проекту по підрозділах і в цілому.

Також інформаційна технологія відображає просте нагадування/сповіщення на головній сторінці про майбутні дедлайни, яке

відображається лише тоді, коли проект все ще виконується і поточна дата знаходиться в межах 15 днів до дедлайну, а також відображає проект, який все ще виконується і вже пройшов дедлайн.

Після успішної авторизації користувач потрапляє на головну сторінку, на якій відображені актуальні будівельні проекти та дедлайни їх виконання (рис. 3.21).

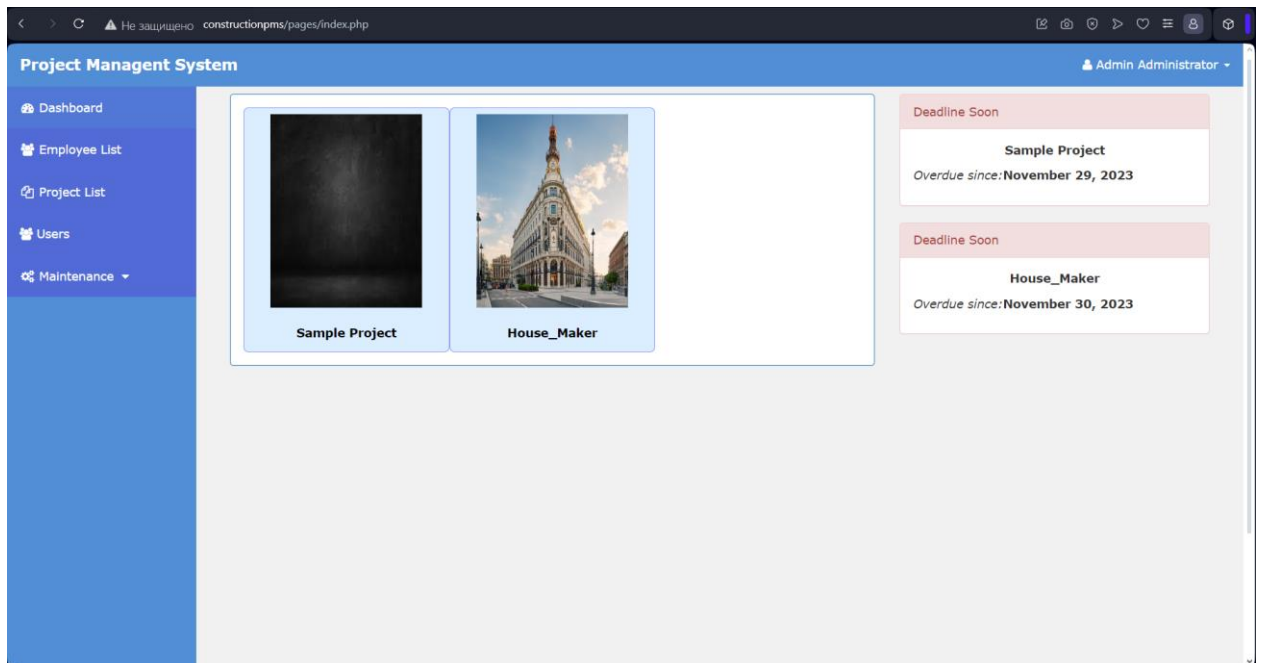


Рисунок 3.21 – Головна сторінка

Інформаційна технологія надає наступні можливості:

-Сторінка зі списком співробітників

-Сторінка, на якій перераховані всі співробітники компанії та здійснюється управління ними. Адміністратор отримує право змінювати інформацію про співробітників та виконувати операції з додаванням та видаленням робочого персоналу. На рисунку 3.22 представлено вигляд сторінки зі списком робітників.

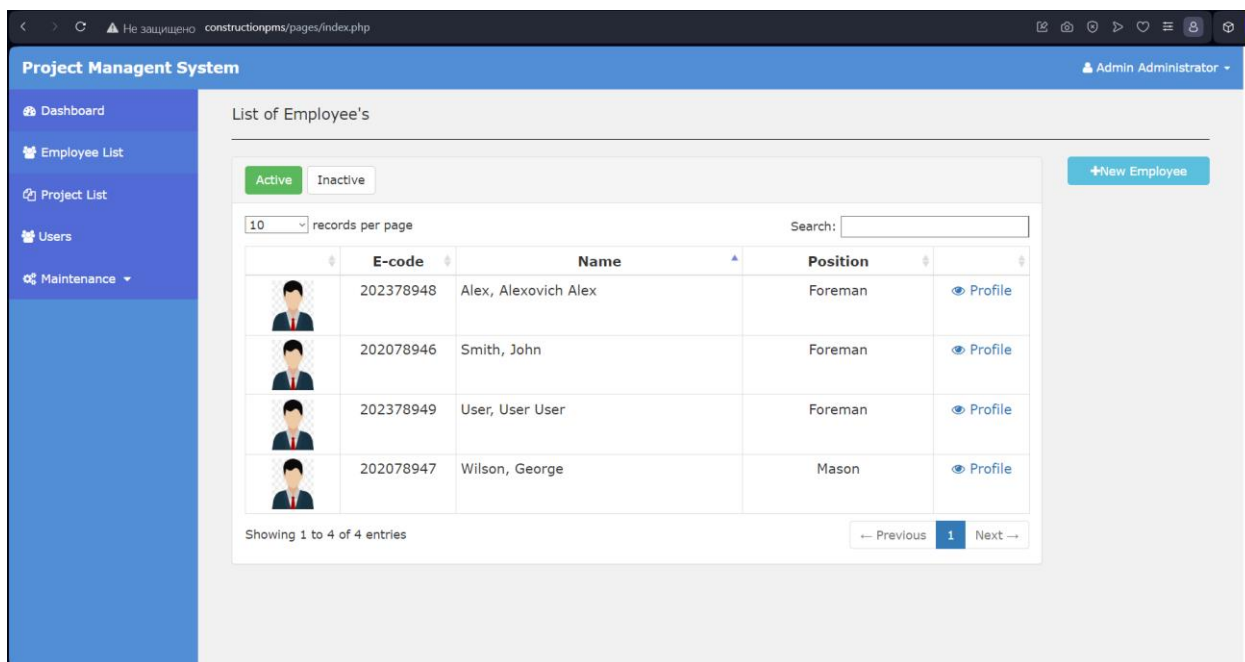


Рисунок 3.22 – Сторінка зі списком робітників

Для додавання нового співробітника необхідно заповнити форму. На рисунках 3.23 – 3.24 представлено форму додавання та результат заповнення форми.

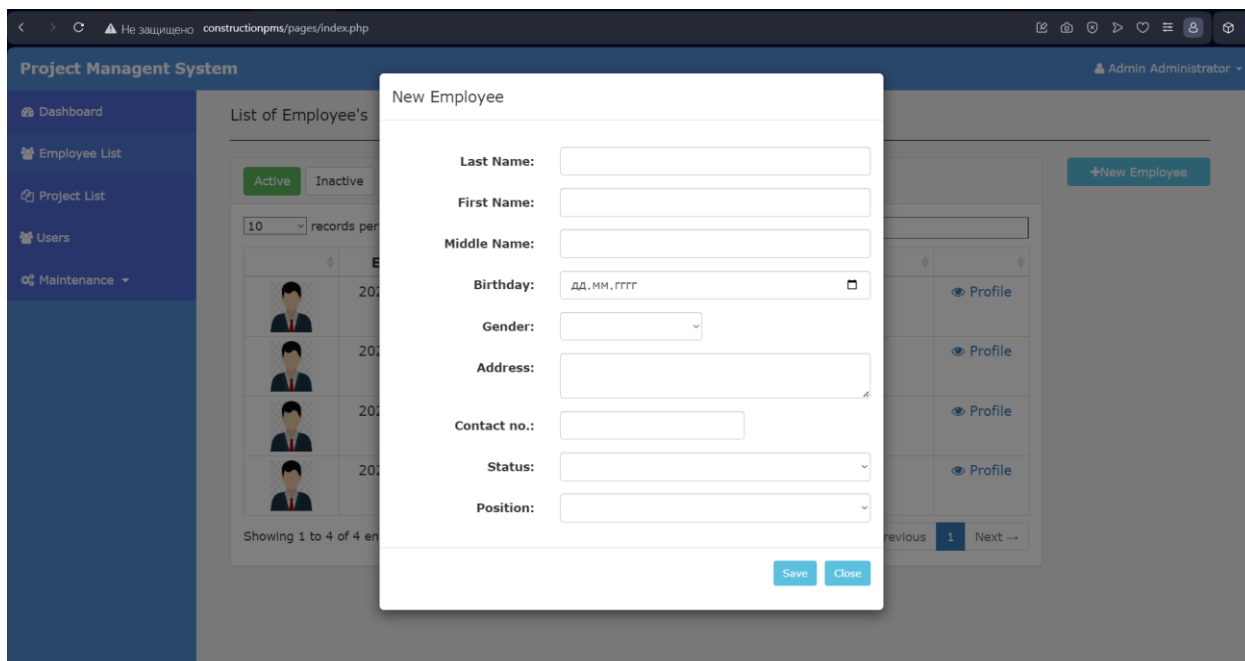


Рисунок 3.23 – Форма додавання нового співробітника

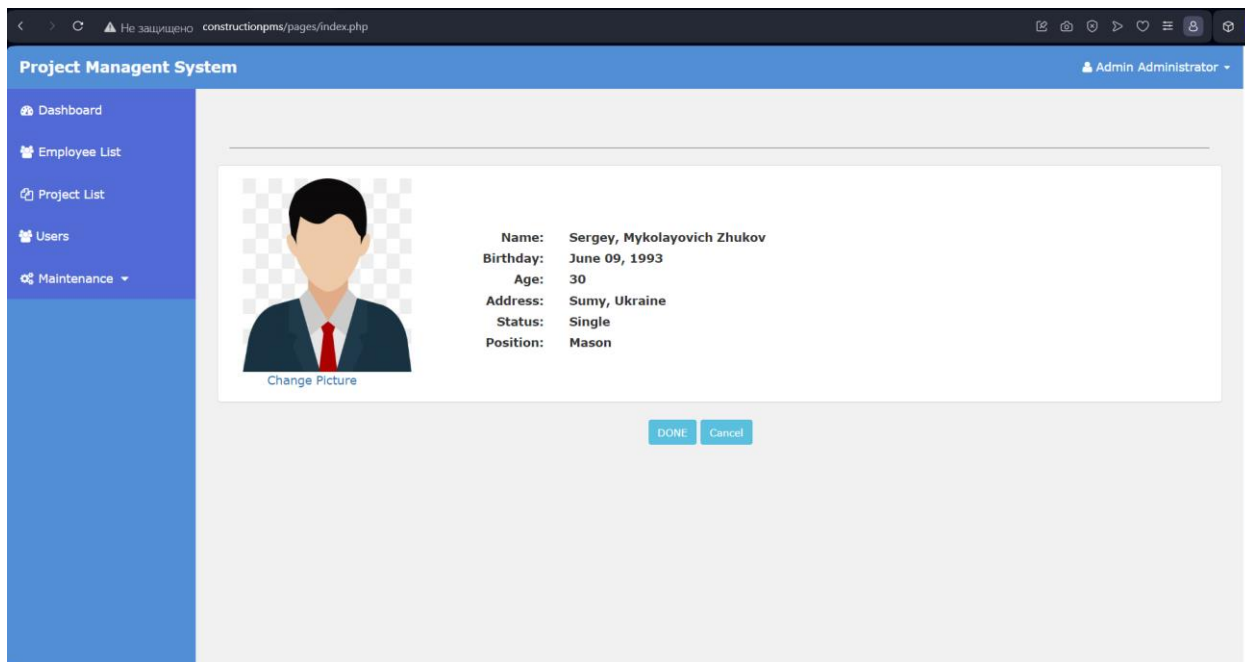


Рисунок 3.24 – Результат додавання нового співробітника

-Сторінка посади

-Сторінка, на якій користувач системи може керувати списком посад співробітників. Адміністратор може змінити посаду або додати нову. На рисунку 3.25 представлено вигляд сторінки посади.

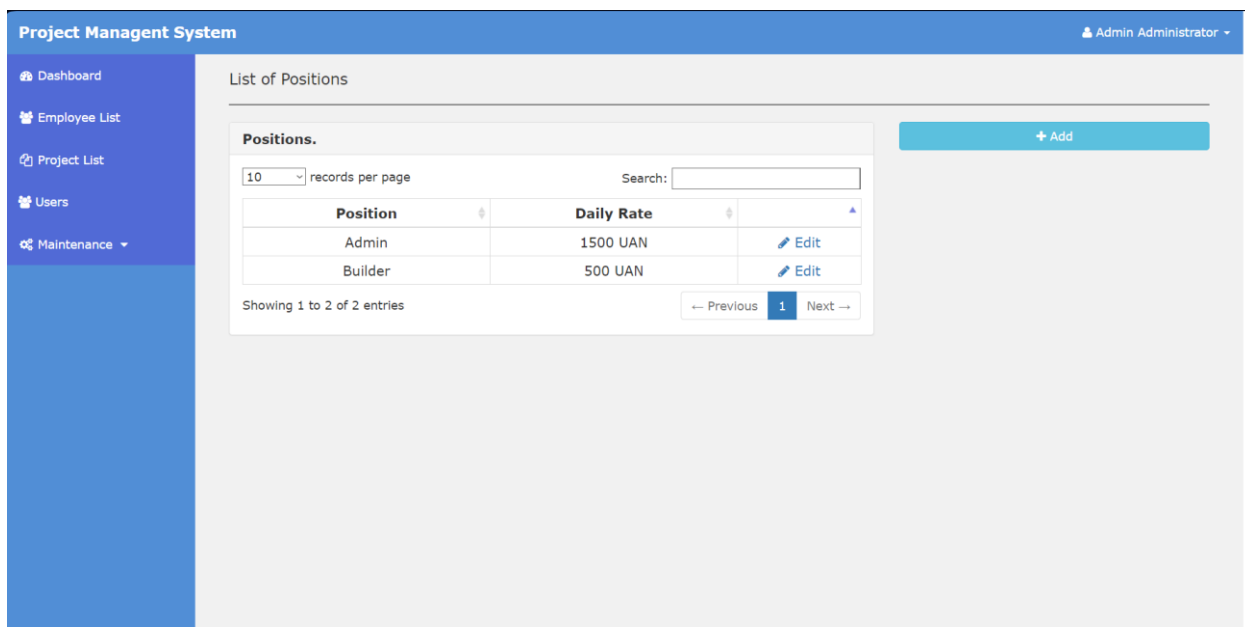


Рисунок 3.25 – Вигляд сторінки посади

-Сторінка відділу проекту

-Сторінка, на якій користувач системи може керувати списком підрозділів проекту (рис. 3.26).

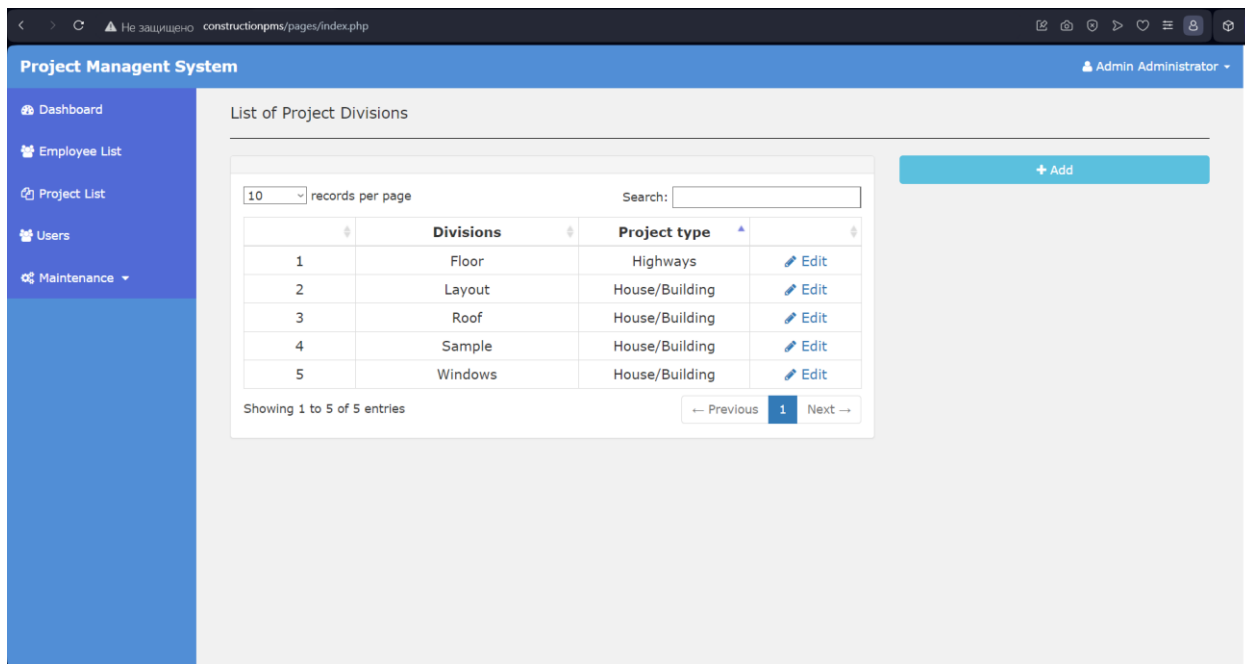


Рисунок 3.26 – Вигляд сторінки підрозділів проекту

-Сторінка команди проекту

-Сторінка, на якій перераховані команди проекту, якими може керувати користувач системи (рис. 3.27).

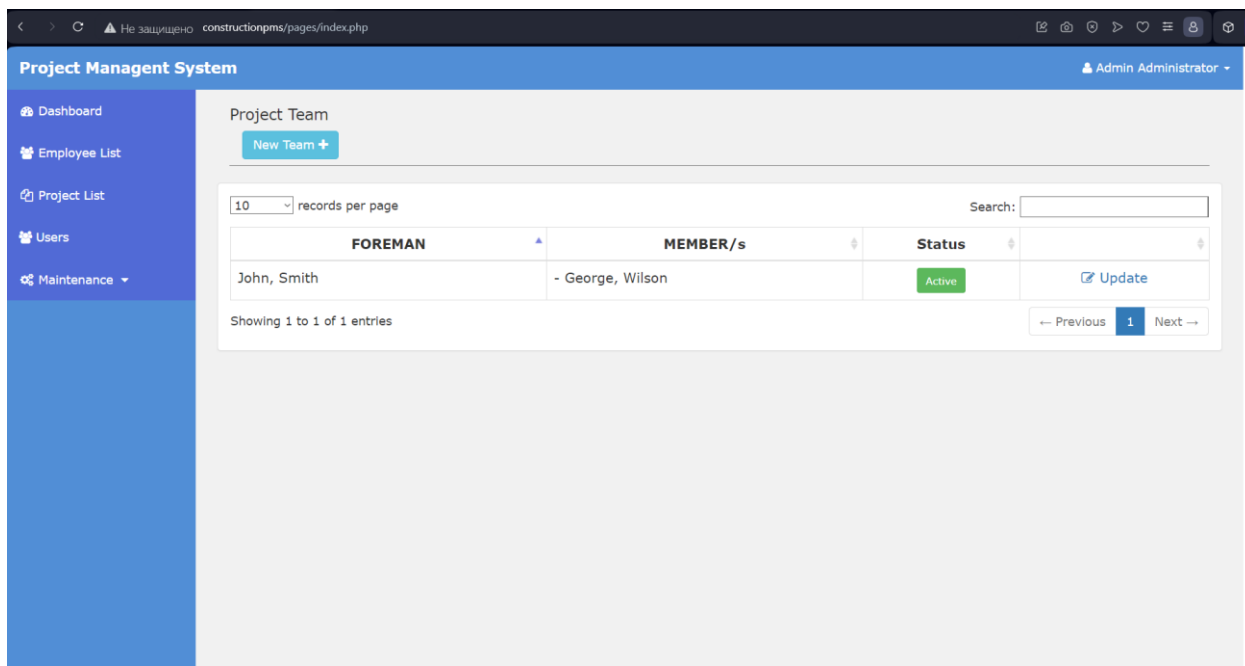


Рисунок 3.27 – Вигляд сторінки команд проекту

-Сторінка списку проектів

-Сторінка, на якій перераховані всі проекти, якими можна керувати разом з їх деталями та статусом (рис. 3.28).

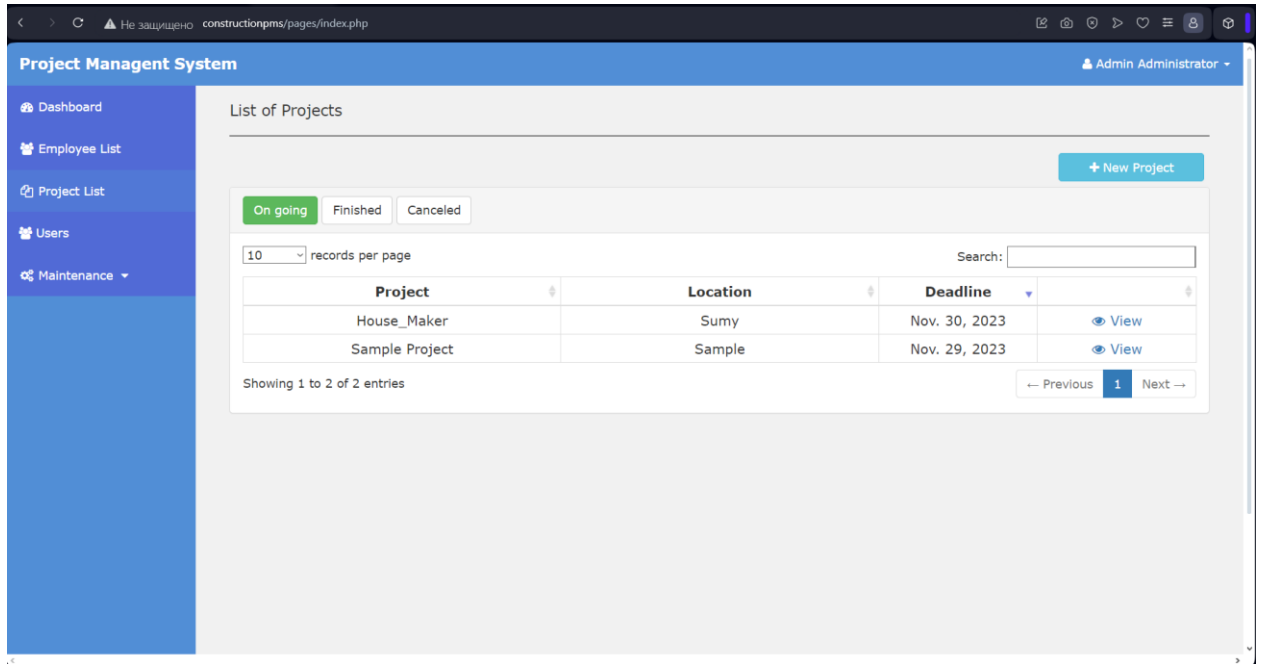


Рисунок 3.28 – Вигляд сторінки проектів

Також можна дізнатися більш детальну інформацію про проект, просто натиснувши на нього (рис 3.29).

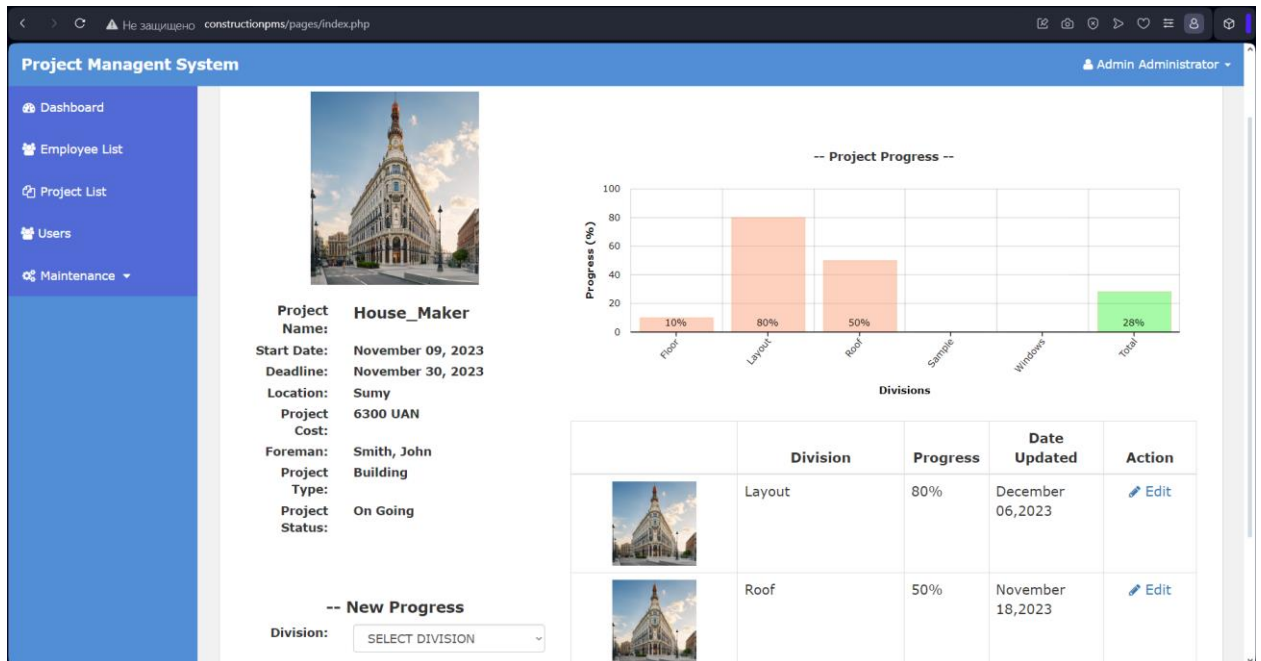


Рисунок 3.29 – Сторінка детальної інформації про проект

-Сторінка користувачів

-Сторінка, де системний адміністратор може керувати списком користувачів системи (рис. 3.30).

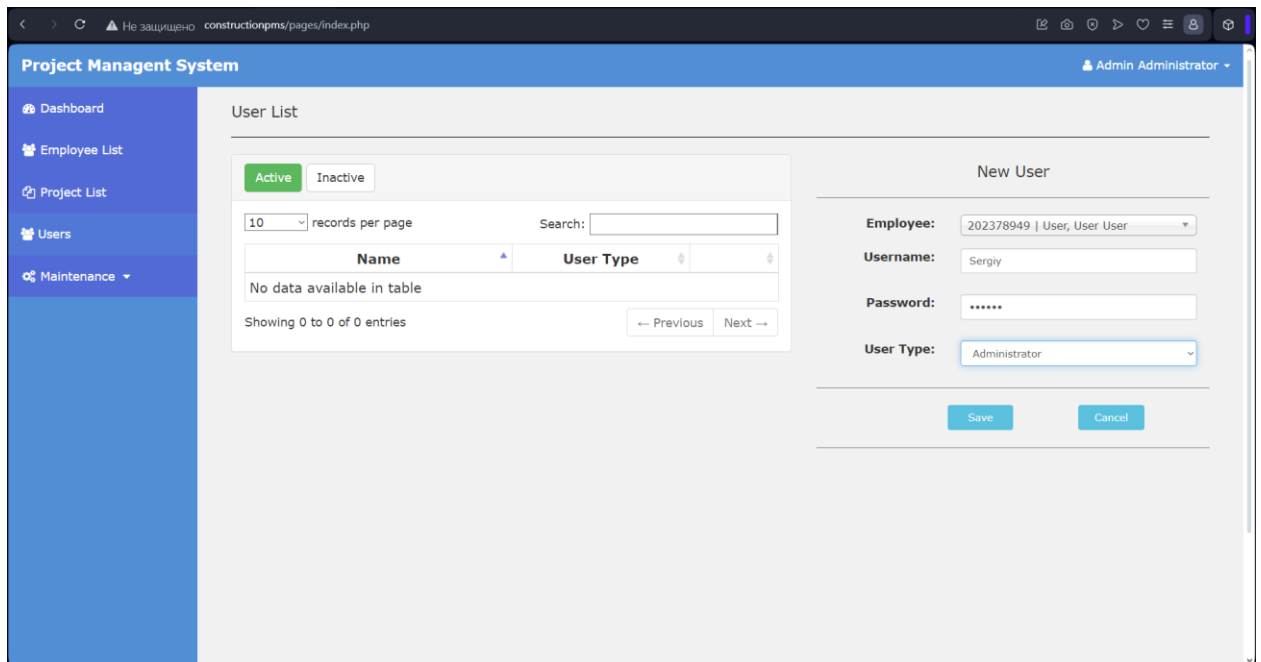


Рисунок 3.30 – Сторінка користувачів

ВИСНОВКИ

В результаті проведеного дослідження та розробки інформаційної технології системи управління будівельними процесами вдалося визначити оптимальні шляхи оптимізації бізнес-процесів та підвищення якості контролю виконання проекту. Аналіз цільового домену та порівняння з аналогічними рішеннями підтвердили актуальність інформаційної технології управління будівельними процесами. Основні особливості та вимоги до системи були визначені з урахуванням потреб бізнесу, а порівняння з існуючими рішеннями підкреслило її потенційні переваги в оптимізації обслуговування.

Дослідження та впровадження новітніх технологій веб-розробки, баз даних та методів управління проектами дозволили досягти оптимального балансу між функціональністю, продуктивністю та безпекою системи. Складені робочі плани та аналіз ризиків сприяли ефективній реалізації проекту, забезпечивши підготовку до можливих викликів.

Розроблена інформаційна технологія управління будівельними процесами відкриває нові можливості для служби управлінням процесом побудови будинків та сприяє підвищенню його конкурентоспроможності. Впровадження інформаційних технологій у бізнес-процеси не лише покращило ефективність роботи системи, а й забезпечило його готовність до викликів і можливостей сучасного ринку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. 10 Benefits Of Web Based Applications & Systems [Електронний ресурс] – URL: <https://rapidcrews.com/10-benefits-of-web-based-applications-systems/> (дата звернення 27.11.2023).
2. What are the benefits of Web Service [Електронний ресурс] – URL: <https://www.geodatasource.com/resources/tutorials/what-are-the-benefits-of-web-service/> (дата звернення 27.11.2023).
3. CRM система CRMSTROY [Електронний ресурс] – URL: <https://crm.crmstroy.com/> (дата звернення 27.11.2023).
4. Система для управління бізнес процесами OneBoxCRM [Електронний ресурс] – URL: <https://1b.app/ua/partner/shebo/> (дата звернення 27.11.2023).
5. Network planning Автор Graham K. Rand & Luis Valadares Tavares [Електронний ресурс] – URL: https://link.springer.com/referenceworkentry/10.1007/1-4020-0611-X_665 (дата звернення 27.11.2023).
6. What is cloud architecture? [Електронний ресурс] – URL: <https://cloud.google.com/learn/what-is-cloud-architecture> (дата звернення 27.11.2023).
7. What are the Advantages of Object-Oriented Programming? Автор Sriram 19.09.2023 [Електронний ресурс] – URL: <https://www.upgrad.com/blog/what-are-the-advantages-of-object-oriented-programming/> (дата звернення 27.11.2023).
8. Client Server Architecture: Types, Examples, & Benefits [Електронний ресурс] – URL: <https://www.redswitches.com/blog/client-server-architecture/> (дата звернення 27.11.2023).
9. Client Server Architecture: Components, Types, Benefits Автор Abhresh Sugandhi від 15.09.2023 [Електронний ресурс] – URL:

- <https://www.knowledgehut.com/blog/cloud-computing/client-server-architecture> (дата звернення 27.11.2023).
10. Why MERN Stack Developers: Benefits, Cost, Hiring Автор Irfan A Rehman від 22.06.2023 [Електронний ресурс] – URL: <https://clickysoft.com/why-should-hire-mern-stack-developers/> (дата звернення 27.11.2023).
 11. Create Interactive Applications with PHP MySQL Development Services [Електронний ресурс] – URL: <https://webxloo.com/technology/php-mysql.html> (дата звернення 27.11.2023).
 12. C++ for Web Development: Why It's a Game-Changer Автор Stefan [Електронний ресурс] – URL: <https://teamcubate.com/blogs/c-plus-plus-for-web-development> (дата звернення 27.11.2023).
 13. What Are the Benefits of Using PHP and MySQL [Електронний ресурс] – URL: <https://wpcrux.com/blog/what-are-the-benefits-of-using-php-and-mysql> (дата звернення 27.11.2023).
 14. The Complete Guide To Understand IDEF Diagram [Електронний ресурс] – URL: <https://www.edrawmax.com/article/the-complete-guide-to-understand-idef-diagram.html> (дата звернення 28.11.2023).
 15. Features and Benefits of the IDEF0 Activity Modeling Method [Електронний ресурс] – URL: <https://www.informit.com/articles/article.aspx?p=2123714&seqNum=3> (дата звернення 28.11.2023).
 16. Benefits of Use Cases [Електронний ресурс] – URL: <https://www.archimetric.com/benefits-use-cases/> (дата звернення 28.11.2023).
 17. Key Benefits of MySQL Development Services [Електронний ресурс] – URL: <https://blog.flatworldsolutions.com/key-benefits-of-mysql-development-services/> (дата звернення 28.11.2023).
 18. How to Write SMART Goals: SMART Goal Examples [Електронний ресурс] – URL: <https://www.projectmanager.com/blog/how-to-create-smart-goals> (дата звернення 28.11.2023).

19. What Is a Work Breakdown Structure (WBS)? [Электронный ресурс] – URL: <https://www.projectmanager.com/guides/work-breakdown-structure> (дата звернення 28.11.2023).
20. What is Organization Breakdown Structure (OBS) in Construction? [Электронный ресурс] – URL: <https://www.getpowerplay.in/blog/what-is-organization-breakdown-structure-obs-in-construction> (дата звернення 28.11.2023).
21. Gantt Charting: Definition, Benefits, and How They're Used [Электронный ресурс] – URL: <https://www.investopedia.com/terms/g/gantt-chart.asp> (дата звернення 28.11.2023).
22. What Is Project Risk Management: Benefits, Challenges, Best Practices [Электронный ресурс] – URL: <https://status.net/articles/project-risk-management-benefits-challenges/> (дата звернення 28.11.2023).

ДОДАТОК А

Деталізація мети проекту методом SMART

SMART - це метод, який встановлює критерії ідеальних цілей і завдань проекту. SMART означає "конкретні, вимірювані, досяжні, реалістичні та пов'язані з часом". Це означає, що мета повинна відповідати цим критеріям, щоб вважатися SMART-ціллю.

SMART-цілі допомагають керівникам проектів, бізнес-менеджерам і будь-яким іншим типам керівників команд визначити чіткі завдання, які повинні бути виконані їхніми командами. Найкраще те, що SMART-цілі можна використовувати для вимірювання ефективності практично будь-якого проекту або завдання [18].

Цей простий набір критеріїв є керівництвом, яке допоможе командам бути більш продуктивними, завжди пам'ятаючи про цілі.

Результати SMART-методу щодо web-орієнтованої системи управління будівельними процесами наведено у таблиці Б.1.

Таблиця Б.1 – Формалізація мети за допомогою методу SMART

Specific (Конкретна)	Створити web-систему, яка забезпечить ефективне управління будівельними процесами, включаючи планування, розподілення ролей та моніторинг ресурсів.
Measurable (Вимірювана)	Забезпечення можливості вимірювання ефективності системи через тайм-менеджмент часу виконання проектів, аналіз витрат та покращення комунікації між виконавцями проекту.
Achievable (Досяжна)	Реалізувати основні функціональності системи з використанням сучасних технологій та методів веб-розробки.

Продовження таблиці Б.1 – Формалізація мети за допомогою методу SMART

Relevant (Відповідна)	Система повинна відповідати потребам будівельної галузі, забезпечуючи інструменти для ефективного управління проектами та ресурсами.
Time-bound (Обмежена в часі)	Проект має бути завершений та впроваджений впродовж визначеного терміну, згідно з розробленим графіком робіт.

Work Breakdown Structure (WBS)

Структура розбиття робіт (WBS) - це система управління проектами, яка розбиває проекти на менші, більш керовані компоненти або завдання. Це візуальний інструмент, який розбиває весь проект на частини, щоб полегшити його планування, організацію та відстеження прогресу. WBS присвоює кожному завданню унікальний ідентифікатор, а потім розміщує їх в ієрархічній структурі, яка показує взаємозв'язок між кожним завданням і пов'язаними з ним результатами [19].

Основними цілями WBS є забезпечення чіткого розуміння обсягу проекту, визначення всіх робіт, які необхідно виконати, а також забезпечення ефективного планування та управління проектом.

На рисунку Б.1 представлено WBS проекту щодо web-орієнтованої системи управління будівельними процесами

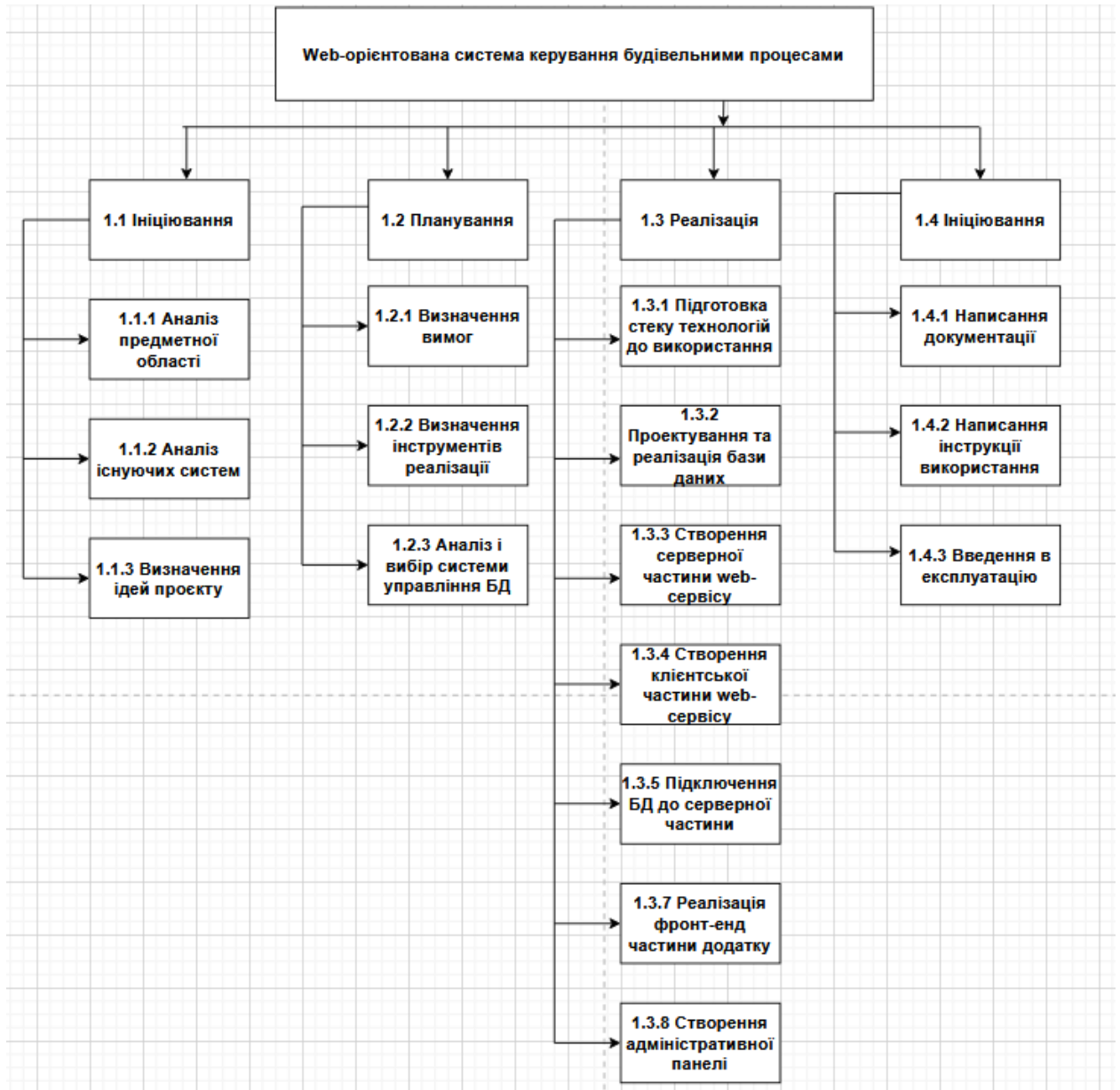


Рисунок Б.1 - WBS-структура робіт проекту

Organization Breakdown Structure (OBS)

Організаційна структура є широко використовуваним інструментом управління проектами для представлення організації проекту. Зазвичай вона починається зі спонсора проекту та всіх ключових зацікавлених сторін.

OBS має тісний зв'язок з WBS, оскільки OBS визначає організацію, відповідальну за виконання певного сегменту в WBS. Цей взаємопов'язаний

процес отримує допомогу від методології матриці розподілу відповідальності (RAM) і допомагає у визначенні обсягу проекту [20].

Структура OBS повністю залежить від цілей проекту, беручи до уваги об'єкт, наданий субпідрядникам, а також архітектурний і технічний персонал. Ця структура готується кожного разу менеджером проекту і включає зміни відповідно до складності, обсягу та розміру проекту.

На рисунку Б.2 представлено організаційну структуру планування проекту.

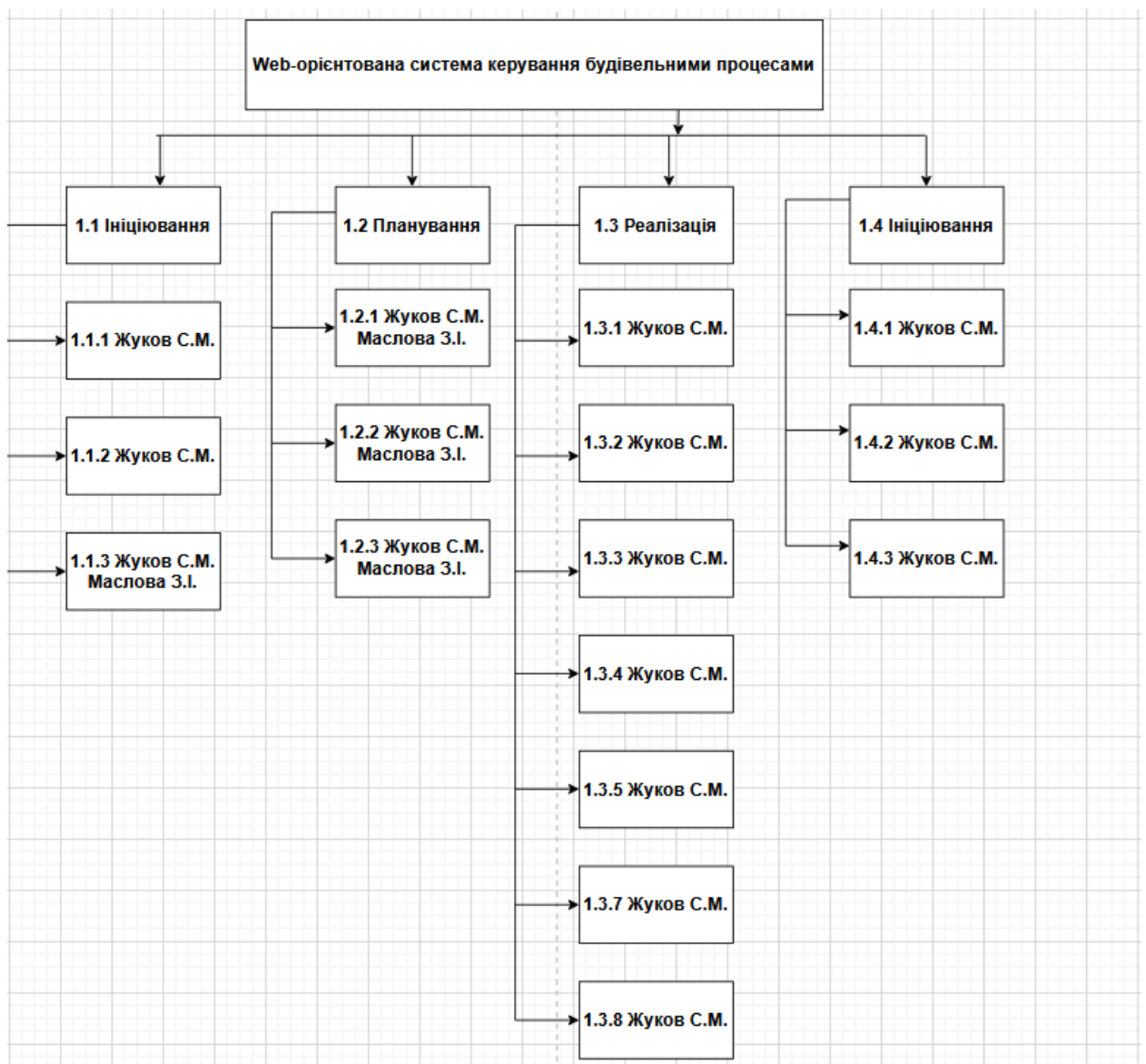


Рисунок Б.2 – OBS структура робіт проекту

Діаграма Ганта

Діаграма Ганта - це візуальне представлення проектних завдань і робіт, запланованих у часі. Вона допомагає планувати проекти і дозволяє візуалізувати прогрес у реальному часі.

Як правило, діаграма Ганта складається зі списку завдань зліва і горизонтальних смуг прогресу справа. Кольорові смуги прогресу відображають дати початку і закінчення завдання, а також вказують на відсоток виконаної роботи в рамках завдання [21].

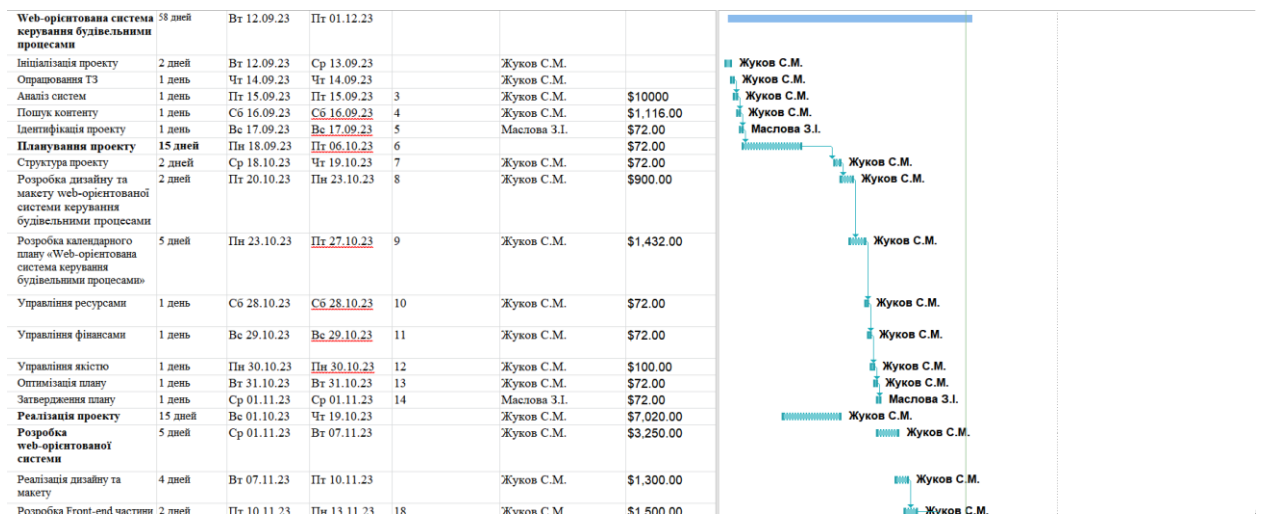


Рисунок Б.3 – Діаграма Ганта, частина 1

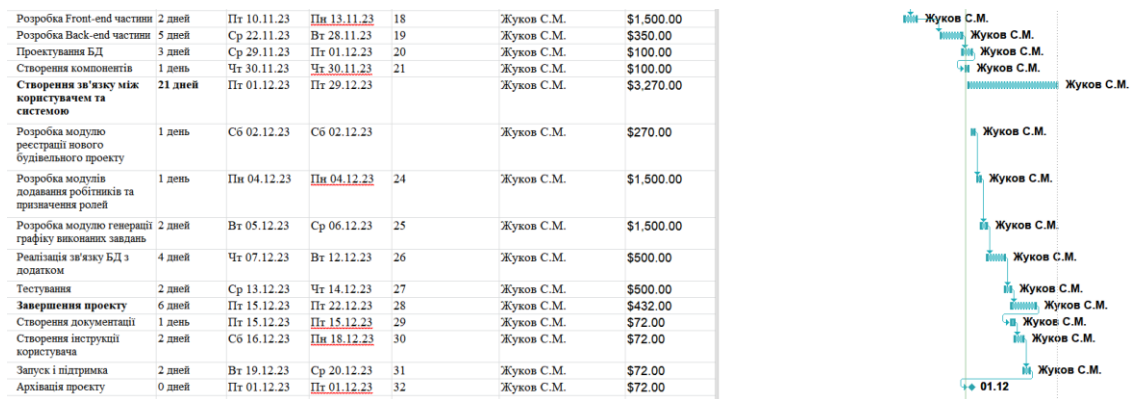


Рисунок Б.4 – Діаграма Ганта, частина 2

Управління ризиками проекту

Управління проектними ризиками - це розробка стратегій, спрямованих на запобігання або мінімізацію впливу тривожних загроз на проект. Завжди існує невизначеність щодо позитивного результату проекту. Все може швидко піти не так, і коли менеджери проектів розуміють значення управління ризиками, вони стратегічно готуються до управління цими ризиками [22].

Коли менеджер береться за довгострокове завдання, виникає багато питань, які потребують уваги, починаючи від формування бюджету і закінчуючи тим, як утримати персонал на робочому місці. Тому цілком ймовірно, що стратегії управління ризиками ніколи не увійдуть до планування проекту. Однак вивчення цих підходів є необхідністю, оскільки переваги розробки планів для вирішення непередбачуваних проблем є численними.

Таблиця А.2. Ймовірність виникнення і величина ризику

№	Ризики	Виникнення	Втрати
1	Відсутність резервної копії даних	2	4
2	Поява додатку - альтернативи	3	3
3	Технічні несправності під час розробки	3	5
4	Проблеми з захистом особистих даних користувачів	2	3
5	Помилки з оновленням версії web-сервісу	4	5
6	Застарілість та неактуальність інформації	2	3

Таблиця А.3 – Матриця впливу

Вірогідність виникнення	Матриця впливу				
5			3	5	
4		4			
3		6	2		
2				1	
1					
Ступінь впливу	1	2	3	4	5

ДОДАТОК Б

Лістинг коду «user_edit.php»

```

<center><h4>Edit User</h4></center>

<?php
    include '../includes/db.php';

    $query2= mysqli_query($conn, "SELECT
*,CONCAT(lastname,', ',firstname,', ',midname) as
name,users.io as status FROM users left join employee
on users.eid = employee.eid where uid =
'".$_GET['uid']."' ");
    $row2 = mysqli_fetch_assoc($query2);
    ?>

<hr style="border-bottom:1px solid grey"></hr>
<div id="retCode1"></div>
    <form id="update_user_form" method="POST">
    <div class="form-group">
    <div class="col-sm-12 text-right"><center><label
for="emp"><?php echo ucwords($row2['name'])
?></label></center></div>
        <input type="hidden" value="<?php echo
$row2['uid'] ?>" name="uid">

    </div>
    </div>

<br>
<br>
    <div class="form-group">

```

```

    <div class="col-sm-4 text-right"><label
for="us">Username:</label></div>
    <div class="col-sm-8">
    <input type="text" class="form-control input-sm"
id="us" value="<?php echo $row2['username'] ?>"
name="user">
    </div>
    </div>
<br>
<br>
    <div class="form-group">
    <div class="col-sm-4 text-right"><label
for="pass">Password:</label></div>
    <div class="col-sm-8">
    <input type="password" class="form-control input-sm"
id="pass" value="<?php echo $row2['password'] ?>"
name="pass">
    </div>
    </div>
<br>
<br>
    <div class="form-group">
    <div class="col-sm-4 text-right"><label
for="u_type">User Type:</label></div>
    <div class="col-sm-8">
    <select type="text" class="form-control input-sm"
id="u_type" name="u_type">
    <option value="<?php echo $row2['user_type'] ?>" >
    <?php if($row2['user_type'] == '1'){
    echo 'Administrator';

```

```

    }else{
        echo 'Contractor';
    }

    ?>
</option>
<?php if($row2['user_type'] == '2'){?><option
value="1">Administrator</option> <?php } ?>
    <?php if($row2['user_type'] == '1'){?><option
value="2">Contractor</option><?php } ?>
</select>
</div>
</div>
<br>
<br>
    <div class="form-group">
        <div class="col-sm-4 text-right"><label
for="status">Status:</label></div>
        <div class="col-sm-8">
            <select type="text" class="form-control input-sm"
id="status" name="status">
                <option value="<?php echo $row2['status'] ?>" >
                <?php if($row2['status'] == '1'){
                    echo 'Active';
                }else{
                    echo 'Inactive';
                }

                ?>
            </option>

```

```

    <?php if($row2['status'] == '2'){?><option
value="1">Active</option> <?php } ?>
    <?php if($row2['status'] == '1'){?><option
value="2">Inactive</option><?php } ?>
</select>
</div>
</div>
<br>
<hr style="border-bottom:1px solid grey"></hr>

<div class="form-horizontal">
  <div class="form-group">
    <div class="col-sm-12">
      <center>
        <div class="col-sm-4"></div>
        <button class="col-sm-2 btn btn-info btn-sm"
id="save_pos">Save</button>
        <div class="col-sm-2"></div>
        <a class="col-sm-2 btn btn-info btn-sm"
onclick="window.location.reload()">Cancel</a>
      </center>
    </div>
  </div>
</div>
<hr style="border-bottom:1px solid grey"></hr>
</form>

<script>
  $(document).ready(function(){
    $('#msg2').hide();
  });

```

```
});  
jQuery("#update_user_form").submit(function(e){  
    e.preventDefault();  
    var formData =  
jQuery(this).serialize();  
    $.ajax({  
        type: "POST",  
        url:  
"../forms/update_forms.php?action=user",  
        data: formData,  
        success: function(html){  
            $('#retCode1').append(html);  
            var delay = 2000;  
            setTimeout(function(){  
location.replace(document.referrer);    }, delay);  
  
            }  
        });  
        return false;  
    });  
</script>
```


ДОДАТОК В

Лістинг коду «progress_chart3»

```
<?php
include '../includes/header.php';
    $servername = "localhost";
$username = "root";
$password = "";
$dbname = "fggs_rs";
$conn = new mysqli($servername, $username, $password,
$dbname);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$id = $_GET['id'];
    $prog = $conn->query("SELECT * FROM
project_partition natural join project_division where
project_id = '$id' ");
    while($progress = $prog ->fetch_assoc()){

        $name = $progress['division'];
        $pid = $progress['pp_id'];
        $i= 1;
        $prog3 = $conn->query("SELECT SUM(progress) as
total_prog FROM project_progress  where pp_id = '$pid'
");
        $row_prog = $prog3->fetch_assoc();
            if ($prog && $prog->num_rows > 0)
                {
```

```

        if($row_prog['total_prog'] <= 50){
            $color='rgba(251, 159, 118, 0.53)';
        }elseif ($count > 50 ) {
            $color='rgba(120, 151, 239, 0.53)';
        }
        $array[$id][]
    = '{"progress":' . $row_prog['total_prog'] . ', ' . "name":' . ucfirst($name) . ', ' . "color":' .
    $color . '"}';

    }
    else
    {
        $array[] = '{"progress":'0', "name":'0' }';
    }
}

$prog2 = $conn->query("SELECT SUM(progress) as
total FROM project_progress natural join
project_partition where project_id = '$id' ");
$progress2 = $prog2 ->fetch_assoc();
$total = $progress2['total'] / $prog->num_rows ;
$tots= number_format($total,0);
$colors='rgba(0, 241, 5, 0.39)';

    $data2 = ', {"progress":' . $tots .
    ', "name":' . "Total" . ', "color":' .
    $colors . '"}';

```

```

    $data= implode(',', $array[$id]);
?>
<div class="chartdiv" id="chartdiv<?php echo
$_GET['id'] ?>" style="width:100% ; height:40%;"></div>
<script>

    jQuery(document).ready(function() {
chart.exportConfig = {
    menuItems: [{
        icon: '../am_chart/images/export.png',
        format: 'png',
        onclick: function(a) {
            var output = a.output({
                format: 'png',
                output: 'datastring'
            }, function(data) {
                console.log(data)
            });
        }
    }
    ]
}
});

var chart = AmCharts.makeChart("chartdiv<?php echo
$_GET['id'] ?>", {
    "type": "serial",

```

```

    "theme": "none",

    "pathToImages": "http://localhost/new_admin/am_chart/ima
    ges/export.png",
    "dataProvider": [<?php echo $data.$data2 ?>],
    "title": "Project Progress",
    "valueAxes": [{
        "axisAlpha": 0,
        "position": "left",
        "title": "Progress (%)",

    }],

    "startDuration": 1,
    "graphs": [{
        "balloonText": "<b style='text-
transform:capitalize'>[[name]]: [[value]]%</b>",
        "colorField": "color",
        "fillAlphas": 0.9,
        "lineAlpha": 0.2,
        "type": "column",
        "valueField": "progress",
        "labelText": "[[progress]]%",
        "labelPosition": "inside",

    }],

    "chartCursor": {
        "categoryBalloonEnabled": false,
        "cursorAlpha": 0,
        "zoomable": false
    },

```

```
"categoryField": "name",
"categoryAxis": {
  "text-transform": "capitalize",
    "gridPosition": "start",
    "labelRotation": 50,
    "title": "Divisions"
  },
});

</script>
<script src="../../assets/js/jquery.min.js"></script>
```

ДОДАТОК Г

Лістинг коду «prog_search.php»

```

<?php

include '../includes/db.php';

$name = $_GET['name'];
if($name == ''){
    $sql = mysqli_query($conn,"SELECT
*,CONCAT(lastname,', ',firstname, ' ',midname) as name
from projects left join project_team on projects.tid =
project_team.tid left join employee on project_team.eid
= employee.eid order by deadline ASC");
}else{
    $sql =mysqli_query($conn,"SELECT
*,CONCAT(lastname,', ',firstname, ' ',midname) as name
from projects left join project_team on projects.tid =
project_team.tid left join employee on project_team.eid
= employee.eid where project like '%" . $name . "%' order
by deadline ASC");
}

if(mysqli_num_rows($sql) > 0){
while($proj = mysqli_fetch_assoc($sql)){
    $id= $proj['project_id'];
?>

<div class="panel panel-deafault">
    <div class="panel panel-body">
        <div class="col-sm-4">

```

```

        <center></center><br>
        <center><h4><b><?php echo
$proj['project'] ?></b></h4></center>
    </div>
    <div class="col-sm-8"> <?php include
'progress_chart.php'; ?> </div><br>
    <div class="col-sm-12 col-sm-offset-11"><a
href="index.php?page=update_progress&id=<?php echo $id
?>" class="btn btn-sm btn-info"><i class="fa fa-
edit"></i> Update</a></div>

    </div>
</div>

<?php }}else{?>
    <div class="panel panel-deafault">
        <div class="panel panel-body">
            <div class="col-sm-12">

                <center><h4><b>No result
..</b></h4></center>
            </div>

        </div>
    </div>
</div>
<?php } ?>

```