

–

Я, Фомін Олексій Геннадійович як здобувач вищої освіти ХНУРЕ, розумію і підтримую політику закладу з академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Я не використовував штучний інтелект для підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

25 листопада 2025 р.



Олексій ФОМІН

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет _____ АКТ _____
Кафедра _____ КІТАР _____
Рівень вищої освіти _____ другий (магістерський) _____
Спеціальність 174 Автоматизація, комп'ютерно-інтегровані технології та
робототехніка _____
Тип програми _____ Освітньо-професійна _____
Освітня програма Комп'ютерно-інтегровані технологічні процеси і
виробництва _____
(шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри КІТАР _____
(підпис)

« ____ » _____ 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Фоміну Олексію Геннадійовичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення автоматизованої системи технологічних процесів з виготовлення будівельних конструкцій _____

Затверджена наказом по університету від 10.11.2025 р. № 1029 Ст _____

2. Термін подання здобувачем роботи до екзаменаційної комісії 17.12.2025 р. _____

3. Вихідні дані до роботи _____

3.1 Платформи: АРМ, Сервер; 3.2 Багатофункціональна система; 3.3 Технології розробки: PHP, React, Python _____

4. Перелік питань, що потрібно опрацювати в роботі _____

4.1 Аналіз предметної області; 4.2 Аналіз структури основних компонентів системи; 4.3 Аналіз та вибір інструментальних засобів для створення системи; _____

4.4 Опис системи; 4.5 Інструкція користувача; 4.6 Висновки _____

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій

Графічний матеріал у вигляді презентації – 12 арк. ф. А 4

6. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Вступ. Аналіз предметної області	10.11.2025	виконано
2	Аналіз структури основних компонентів системи	15.11.2025	виконано
3	Аналіз та вибір інструментальних засобів для створення системи. Опис системи	25.11.2025	виконано
4	Інструкція користувача	30.11.2025	виконано
5	Безпека життєдіяльності	01.12.2025	виконано
6	Оформлення пояснювальної записки	05.12.2025	виконано
7	Подання роботи на нормоконтроль	06.12.2025	виконано
8	Подання роботи для перевірки на академічну доброчесність	07.12.2025	виконано
9	Подання роботи на рецензію	10.12.2025	виконано
10	Подання роботи на підпис зав. кафедри	12.12.2025	виконано
11	Подання кваліфікаційної роботи в ЕК	15.12.2025	виконано

Дата видачі завдання 01.09.2025 р.

Здобувач _____ Олексій ФОМІН
(підпис)

Керівник роботи _____ доц. Наталія ДЕМСЬКА
(підпис)

РЕФЕРАТ

Пояснювальна записка: 72 с., 6 табл., 14 рис., 2 дод., 25 джерел.

REACT JS, PHP, MYSQL, HTTP, MQTT, МОНИТОРИНГ, ЗБЕРЕЖЕННЯ ДАНИХ, АНАЛІЗ, СЕРВЕР.

Метою кваліфікаційної роботи є розроблення автоматизованої системи технологічних процесів бетонного заводу, яка забезпечує збереження, обробку та аналіз даних щодо виконання виробничих операцій і використаних матеріалів. Основною задачею є створення веб-застосунку для ведення звітності, контролю технологічних параметрів та оптимізації процесу виготовлення бетонних сумішей і будівельних конструкцій.

У ході роботи було виконано аналіз виробничих процесів бетонного заводу, визначено ключові показники ефективності та розроблено структуру бази даних для зберігання інформації про виконання технологічних операцій, витрати сировини та якість готової продукції; для реалізації системи використано веб-технології: React для інтерфейсу користувача, PHP і Python для серверної частини та MySQL як сховище даних. Отримані результати можна віднести до Цілі сталого розвитку 9 “Промисловість, інновації та інфраструктура”, зокрема п. 9.3, що передбачає розвиток високотехнологічних секторів промисловості на основі інтеграції освіти, науки та виробництва, впровадження інновацій та інформаційно-телекомунікаційних технологій.

Тестування прототипу підтвердило працездатність системи та її здатність до масштабування під потреби виробництва. Використання сучасних технологій підвищило рівень автоматизації, забезпечило точність обліку матеріалів, оптимізувало витрати та покращило якість кінцевої продукції, що сприяє зростанню продуктивності та конкурентоспроможності бетонного заводу.

ABSTRACT

Explanatory note: 72 pages, 6 tabl., 14 pic., 2 applications, 25 sources.

REACT JS, PHP, MYSQL, HTTP, MQTT, MONITORING, DATA STORAGE, ANALYSIS, SERVER.

The purpose of this qualification work is to develop an automated system for the technological processes of a concrete plant, which ensures the storage, processing, and analysis of data related to the execution of production operations and the materials used. The main objective is to create a web application for reporting, monitoring technological parameters, and optimizing the process of manufacturing concrete mixtures and construction elements.

During the work, an analysis of the concrete plant's production processes was carried out, key performance indicators were identified, and a database structure was developed to store information on the execution of technological operations, raw material consumption, and the quality of finished products. To implement the system, modern web technologies were used: React for the user interface, PHP and Python for the backend, and MySQL as the data storage. The results obtained correspond to Sustainable Development Goal 9 "Industry, Innovation, and Infrastructure," particularly target 9.3, which aims at developing high-tech industrial sectors through the integration of education, science, and production, as well as the implementation of innovations and information and telecommunication technologies.

Testing of the prototype confirmed the system's functionality and its capability to scale according to production needs. The use of modern technologies increased the level of automation, ensured accurate material tracking, optimized costs, and improved the quality of the final product, thereby contributing to increased productivity and competitiveness of the concrete plant.

ЗМІСТ

Перелік скорочень	9
Вступ.....	12
1 Аналіз предметної області.....	10
1.1 Актуальність теми.....	10
1.2 Інформаційні та функціональні потреби користувача	13
1.3 Цілі та задачі автоматизованої системи.....	13
1.4 Вимоги до системи.....	14
1.5 Вимоги до дизайну.....	14
1.6 Структура web-додатку системи	15
1.7 Аналоги автоматизованих систем технологічних процесів бетонних заводів.....	16
1.7.1 Система Siemens SIMATIC	16
1.7.2 Система ELKON.....	17
1.8 Підсумки аналізу предметної області	17
2 Вибір інструментальних засобів для створення системи	27
2.1 Вибір серверу та процедура розміщення системи.....	27
2.2 Аналіз програмних засобів реалізації системи	27
2.2.1 Мова розмітки гіпертексту HTML та мова розмітки CSS	28
2.2.2 Мова програмування PHP	28
2.2.3 Графічний редактор Adobe Photoshop CC.....	30
2.2.4 Середовище розробки Visual Studio Code	31
2.2.5 Онлайн-редактор Figma.....	32
2.2.6 Фреймворк React Native	33
2.2.7 Бібліотека стилів Bootstrap.....	34
2.2.8 Брокер Mosquitto	35
2.2.9 Мова програмування JavaScript.....	36
2.2.10 СУБД PhpMyAdmin	38

	8
3 Структура основних компонентів системи	19
3.1 Спосіб зберігання даних.....	19
3.2 Програмна реалізація системи.....	20
3.2.1 Обґрунтування вибору системи керування базами даних	20
3.2.2 Опис таблиць	21
3.2.3 Опис Web-додатку системи технологічних процесів.....	24
3.3 Підсумки по розробленню структури компонентів системи.....	26
4 Реалізація системи автоматизації технічних процесів	39
4.1 Складові системи автоматизації технологічних процесів	39
4.2 Програмна реалізація підключення та зберігання даних з топіка через MQTT протокол.....	40
4.3 Програмна реалізація web-додатку	41
4.3.1 Front-end модуль web-додатку.....	41
4.3.2 Back-end модуль web-додатку	42
4.4 Розрахунок стійкості системи.....	43
4.5 Забезпечення безпечних умов праці при створенні автоматизованої системи.....	48
5 Інструкція користувача.....	49
5.1 Інтерфейс Web-додатку	49
5.2 Розділ реалізація товарів і послуг.....	49
5.3 Розділ звіти	53
5.4 Розділ замовлення	54
5.5 Підсумки аналізу інтерфейсу системи.....	55
Висновки	56
Перелік джерел посилання	58
Додаток А Апробація наукових результатів досліджень.....	63
Додаток Б Програмна реалізація підключення та зберігання даних з топіка через MQTT протокол.....	67
Додаток В Демонстраційний матеріал.....	73

ПЕРЕЛІК СКОРОЧЕНЬ

ПЗ – програмне забезпечення

СУБД – системи управління базами даних

API – Application Programming Interface

Cron – Command Run On

HTTP – HyperText Transfer Protocol

IoT – Internet of Things

JSX – JavaScript XML

MQTT – Message Queue Telemetry Transport

UI – User interface

UX – User Experience

XML – eXtensible Markup Language

ВСТУП

На сучасному етапі розвитку інформаційних технологій Інтернет є потужним середовищем для обміну, збереження та обробки даних. Разом із глобальним розширенням мережі швидко розвивається напрям веб-програмування, який став основою для створення різноманітних автоматизованих систем і веб-застосунків [1]. Сьогодні для ефективної організації виробничих процесів підприємствам необхідно мати власні інформаційні системи, що дозволяють здійснювати контроль, аналітику та управління технологічними процесами у зручній цифровій формі.

Особливо це актуально для підприємств будівельної галузі, зокрема бетонних заводів, де точність і оперативність виконання технологічних операцій безпосередньо впливають на якість кінцевої продукції. Веб-застосунок у такій системі виконує роль централізованого інструмента для зберігання звітів про виробничі процеси, контролю використання матеріалів і відстеження ефективності роботи обладнання [2-3].

У даній роботі розглядається розроблення системи для контролю якості та автоматизації технологічних процесів бетонного заводу. Актуальність теми зумовлена потребою в підвищенні рівня цифровізації промислових підприємств, скороченні паперового документообігу та покращенні контролю над витратами матеріалів.

Об'єктом дослідження є оцінка ефективності бетонних заводів та прогнозування їх розширення, аналіз складських залишків і прогноз поповнення матеріалів, оцінка завантаженості заводів і вибір оптимального майданчика для виконання замовлення.

Предметом дослідження є програмний комплекс системи контролю якості, методи структурування та візуалізація даних, алгоритми підрахунку використаних матеріалів.

В роботі запропоновано програмний комплекс контролю якості

виробництва, аналіз та збереження даних. ПЗ дозволяє зберігати та аналізувати дані з використаних матеріалів, що дозволяє лабораторіям при бетонних заводах, визначати якість і міцність сумішей, використаних при змішуванні бетону. ПЗ представляє базу з результатами експериментальних випробувань, що можна віднести до отриманих наукових результатів.

Для досягнення поставленої мети необхідно виконати наступні завдання:

Робота виконана згідно [4-7]. Результати роботи опубліковані в [1].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Актуальність теми

Розробка веб-застосунків для управління технологічними процесами [8] сьогодні стає невід'ємною складовою розвитку сучасних підприємств. Зокрема, для бетонних заводів, де важливо контролювати кожен етап виробництва, створення веб-системи є необхідною умовою для ефективної роботи. Веб-застосунок виступає не лише інструментом комунікації між працівниками, а й засобом централізованого зберігання та обробки даних, що дозволяє оперативно реагувати на зміни у виробничому процесі [9].

Інтернет сьогодні є не просто середовищем для обміну інформацією, а цілою індустрією, яка активно впроваджується у виробничу та технологічну сфери. Сучасні підприємства дедалі частіше використовують веб-технології для автоматизації процесів, підвищення точності обліку матеріалів та покращення управління ресурсами. Завдяки постійному розвитку інформаційних технологій підвищується швидкість доступу до даних, розширюються можливості аналітики та знижується вартість впровадження таких систем.

Розроблення автоматизованої системи технологічних процесів бетонного заводу на основі веб-технологій дозволяє не лише відстежувати стан виконання виробничих операцій, а й формувати звіти, аналізувати використання матеріалів та оптимізувати роботу обладнання у реальному часі. Такий підхід забезпечує ефективне управління виробництвом, мінімізує людський фактор і сприяє підвищенню якості кінцевої продукції. Саме тому створення подібних систем є актуальним напрямом розвитку сучасних підприємств будівельної галузі [8].

1.2 Інформаційні та функціональні потреби користувача

Для того щоб чітко зрозуміти, якою хоче бачити майбутню систему замовник, були проведені перемови та з'ясовано, які функції повинна виконувати система і що хоче отримати замовник від нього. Після перемов, можна визначити, що головними функціями, які повинні бути присутні у системі є можливість створювати нові замовлення на обраний завод, створення довідників з параметрами, створення заявок, та отримання і збереження даних про стан технічного процесу у реальному часі, протокол подій, внесення змін до заявок, і головне збереження звітів, а також розрахунків використаних матеріалів по спеціальному математичному алгоритму.

1.3 Цілі та задачі автоматизованої системи

Ціль автоматизованої системи технологічних процесів бетонного заводу полягає у створенні ефективного інструменту для керування виробничими операціями, моніторингу виконання технологічних етапів та аналізу використаних матеріалів. Система має забезпечити централізоване збирання, обробку та збереження інформації про процеси виготовлення бетонних сумішей, що сприятиме підвищенню точності обліку й оптимізації роботи підприємства.

Основні задачі системи:

- розробити веб-застосунок для відображення та моніторингу даних про виконання технологічних процесів у реальному часі;
- реалізувати механізм збереження звітів і параметрів виробництва у базі даних для подальшого аналізу та формування статистичних звітів;
- забезпечити можливість перегляду історії виконаних операцій, контролю витрат матеріалів та оцінки ефективності виробничих етапів.

1.4 Вимоги до системи

Основою вимогою для системи є розробка зручного інтерфейсу веб-додатку та можливість зберігати переглядати і аналізувати дані.

Головна сторінка додатку (рис. 1.1) повинна містити навігаційне меню, схему елементів управління, а також самі елементи управління. Також мають бути розроблені додаткові екрани для зручного перегляду більш детальної інформації системи.

Головний екран додатку має містити список замовлень і його стан. Натиснувши на одне із замовлень користувача має переадресувати на екран обраного замовлення.

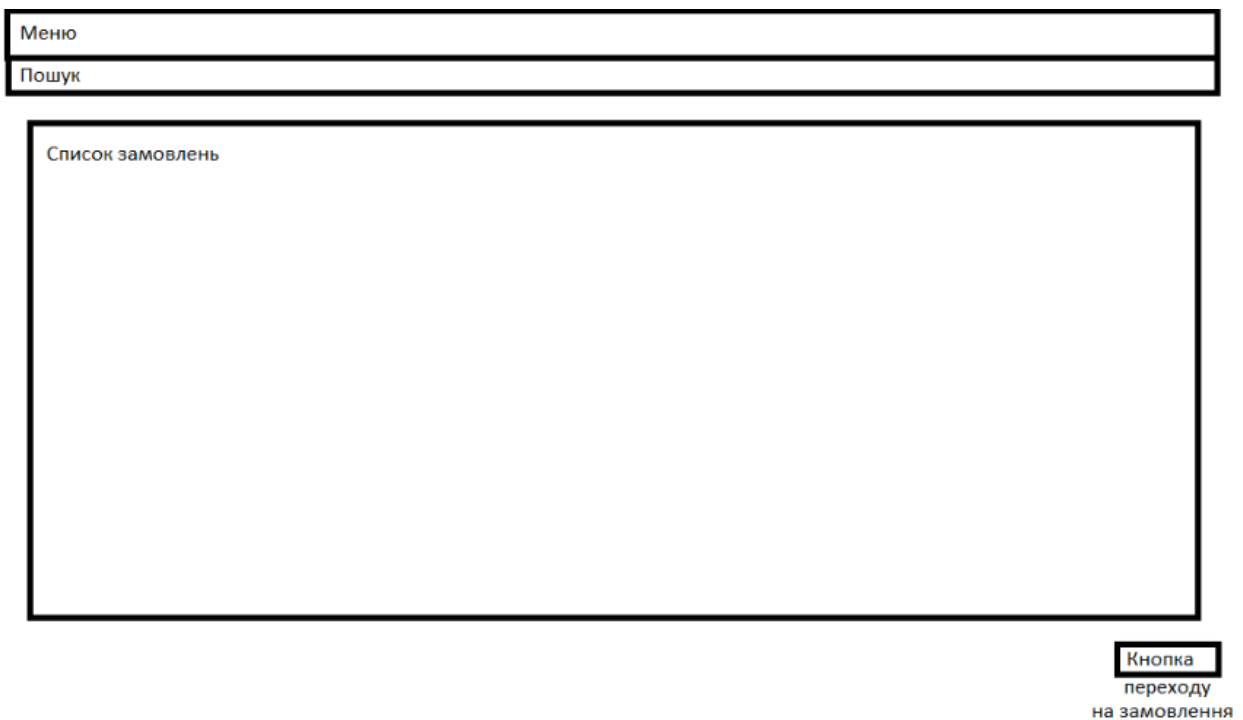


Рисунок 1.1 – Приклад розміщення елементів головної сторінки

1.5 Вимоги до дизайну

Web-додаток може розтягуватися на будь-яку ширину екрану, забезпечуючи оптимальне використання простору на пристроях різного розміру. Всі сторінки мають загальну структуру та загальну кольорову гаму:

світлий фон допомагає зменшити напругу очей користувача, області меню чітко відокремлені від основного контексту за допомогою контрастного синього кольору, а текст розділів меню та заголовки забезпечують зручну читабельність. Кожна сторінка має шапку з основними посиланнями на найважливіші функції додатку, що спрощує навігацію для користувачів. Вище розташоване тіло додатку, де представлена основна інформація або доступ до необхідних функцій. У хедері додатку, розташовано навігаційне меню, що спрощує взаємодію користувача з інтерфейсом і виконує функції навігації між розділами застосунку. Такий дизайн додатку дозволяє забезпечити зручний та швидкий доступ до основних розділів з першого екрану, зменшуючи необхідність в додатковій навігації.

1.6 Структура web-додатку системи

Структура додатку – це його розділи, підрозділи та сторінки. Навігація забезпечує доступ до них, тобто різні меню та перехресні посилання. На етапі планування додатку та написання технічного завдання необхідно ретельно її продумати. Додаток створюється з особливою структурою: зручна та зрозуміла структура додатку допомагає користувачеві легко знаходити потрібну йому інформацію.

На рисунку 1.2 представлена структура майбутнього додатку, яка дає загальне представлення розташування розділів та взаємозв'язок між ними. Слід розуміти, що структурна схема додатку враховує всю специфіку додатку, що розробляється.



Рисунок 1.2 – Структурна додатку

1.7 Аналоги автоматизованих систем технологічних процесів бетонних заводів

Розглянемо кілька реальних прикладів сучасних систем керування виробничими процесами, які мають функціональність моніторингу, збору даних та аналітики у реальному часі, а також можливість віддаленого доступу через веб- або мобільні застосунки.

1.7.1 Система Siemens SIMATIC PCS 7

Siemens SIMATIC PCS 7 – це потужна платформа для автоматизації технологічних процесів, яка широко використовується у виробництві будівельних матеріалів, зокрема на бетонних заводах. Система дозволяє комплексно керувати всіма етапами виробництва: від дозування сировини до контролю якості готового продукту.

Основні режими роботи [10]:

- автоматичний режим: система самостійно керує виконанням технологічних операцій відповідно до заданих параметрів і рецептур;
- напівавтоматичний режим: оператор може втручатися в окремі етапи процесу, коригуючи роботу обладнання;
- режим моніторингу: відображення в реальному часі показників роботи установок, стану обладнання та використаних матеріалів.

Система має веб-інтерфейс та мобільний доступ, що дозволяє:

- переглядати виробничі дані та історію процесів;
- формувати звіти про використання матеріалів;
- оперативно реагувати на відхилення або аварійні ситуації;
- контролювати ефективність роботи обладнання дистанційно.

1.7.2 Система ELKON

ELKON – це спеціалізоване рішення для керування бетонними заводами, яке забезпечує точне дозування компонентів, автоматичний контроль процесу змішування та збереження усіх технологічних параметрів.

Основні режими роботи [11]:

- режим рецептури: автоматичне виконання технологічного циклу відповідно до обраного складу бетонної суміші;
- режим звітності: автоматичне формування звітів про кожен виробничий цикл з інформацією про витрати матеріалів;
- режим ручного керування: можливість оператора вносити зміни у параметри виробництва у реальному часі.

Система має веб-застосунок і підтримку Android, що дозволяє:

- віддалено керувати виробничими процесами;
- переглядати аналітику споживання матеріалів;
- зберігати архів даних для подальшого аналізу;
- отримувати сповіщення про помилки або відхилення від норм.

1.8 Підсумки аналізу предметної області

Аналіз предметної області показав, що створення веб-системи для автоматизації процесів бетонного заводу є актуальним рішенням для підвищення ефективності виробництва та якості контролю. Визначені інформаційні та функціональні потреби користувачів дозволили сформулювати основні вимоги до системи: зручний інтерфейс, створення замовлень і ТТН,

моніторинг процесів у реальному часі, ведення звітності та аналітики.

Сформульовані цілі та задачі передбачають централізоване збирання й аналіз даних, контроль технологічних параметрів та підтримку прийняття управлінських рішень. Окреслені вимоги до структури та дизайну забезпечують зручну навігацію й доступність інформації.

Розгляд аналогічних систем (Siemens PCS 7, ELKON) підтверджує актуальність розробки та задає орієнтир для впровадження сучасних підходів в автоматизації бетонних виробництв.

2 ВИБІР ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ ДЛЯ СТВОРЕННЯ СИСТЕМИ

2.1 Вибір серверу та процедура розміщення системи

Сервер – це комп'ютер або програмне забезпечення, яке забезпечує надання певних послуг, ресурсів чи даних іншим пристроям у мережі, що називаються клієнтами.

Основне призначення сервера – обробка запитів клієнтів і передача їм необхідної інформації. Наприклад, web-сервер зберігає web-сторінки та відправляє їх користувачам через інтернет, файловий сервер – керує зберіганням і передачею файлів, а базовий сервер баз даних – обробляє запити до сховища даних [13].

2.2 Програмні засоби реалізації системи

Список сучасних інструментів та технологій, доступних для розробки додатків, є нескінченним, оскільки індустрія розробки швидко розвивається, і регулярно з'являються нові платформи. Вибір відповідного інструменту для розробки десктопних програм є складною задачею, яка як правило полягає в тому, що розробник повинен розуміти, які функції реалізує додаток. Для реалізації системи було обрано такі програмні засоби:

- JavaScript React.js;
- мова розмітки гіпертексту HTML та мова стилю сторінок CSS;
- бібліотека стилів Bootstrap;
- мова програмування PHP;
- система керування базами даних Phpmyadmin;
- графічний редактор Adobe Photoshop CC;
- онлайн-редактор Figma;

- середовище розробки IDE Visual studio code;
- брокер Mosquitto.

2.2.1 Мова розмітки гіпертексту HTML та мова розмітки CSS

HTML (HyperText Markup Language) є стандартною мовою розмітки, яка використовується для створення та структурування веб-сторінок [16]. HTML дозволяє визначати різноманітні елементи веб-сторінки, такі як заголовки, абзаци, списки, посилання, зображення та інші мультимедійні елементи, через використання тегів. Теги HTML зазвичай складаються з відкриваючого та закриваючого тегів, які оточують контент, який вони форматують або визначають. Наприклад, тег `<p>` використовується для визначення абзацу, а тег `<a>` для створення гіперпосилання.

HTML також підтримує вбудовані скрипти та стилі, що дозволяє розширювати функціональність та зовнішній вигляд веб-сторінок. Загалом, HTML є фундаментальною технологією для веб-розробки, оскільки забезпечує базову структуру та семантику будь-якого веб-документа.

CSS – це формальна мова опису зовнішнього вигляду документа, написаного з використанням мови розмітки. Вона переважно використовується як засіб опису, оформлення зовнішнього вигляду веб-сторінок, написаних за допомогою мов розмітки HTML та XHTML, але може також застосовуватися до будь-яких XML-документах, наприклад, до SVG або XUL.

3.2.2 Мова програмування PHP

PHP [17] – це скриптова мова загального призначення, яка інтенсивно застосовується для розробки веб-додатків. В даний час підтримується переважною більшістю хостинг-провайдерів і є однією із лідерів серед мов, що застосовуються для створення динамічних веб-сайтів.

Мова та її інтерпретатор (Zend Engine) розробляються групою ентузіастів в рамках проєкту з відкритим кодом, який поширюється під

власною ліцензією, несумісною з GNU GPL.

В області веб-програмування, зокрема серверної частини, мова PHP – одна із популярних сценарних мов (разом з JSP, Perl та мовами, використовуваними). Популярність в області побудови веб-сайтів визначається наявністю великого набору вбудованих засобів та додаткових модулів для розробки веб-додатків. Основні з них:

- автоматичне вилучення POST та GET-параметрів, а також змінних оточення веб-сервера в зумовлені масиви;

- взаємодія з великою кількістю різних систем управління базами даних через додаткові модулі (MySQL, MySQLi, SQLite, PostgreSQL, Oracle (OCI8), Oracle, Microsoft SQL Server, Sybase, ODBC, mSQL, IBM DB2, Cloudscape і Apache Derby, Informix, Ovrimos SQL , Lotus Notes, DB ++, DBM, dBase, DBX, FrontBase, FilePro, Ingres II, SESAM, Firebird/InterBase, Paradox File Access, MaxDB, Інтерфейс PDO), Redis;

- автоматизована відправка HTTP-заголовків;

- робота з HTTP-авторизацією;

- робота з cookies та сесіями;

- робота з локальними та віддаленими файлами, сокетамі;

- обробка файлів, що завантажуються на сервер;

- робота з XForms.

Мова автоматично підтримує HTTP Cookies, згідно з міжнародними стандартами Netscape, що дозволяє проводити установку та читання невеликих сегментів даних на стороні клієнта. Робота з Cookies організована за допомогою сеансів (сесій). У сесій є термін дії (після його закінчення, дані видаляються), в сесіях можна зберігати та редагувати різні типи даних, в тому числі серіалізовані – пропущені через serialize (процес відбувається автоматично), PHP-об'єкти.

2.2.3 Графічний редактор Adobe Photoshop CC

Adobe Photoshop є не лише багатофункціональним графічним редактором розробленим і поширюваним гігантом галузі – фірмою Adobe Systems. У своїй основі ця програма спеціалізується на роботі з растровими зображеннями, водночас пропонуючи користувачам деякі інструменти для векторної графіки [18]. Завдяки своїм неперевершеним можливостям, Photoshop визнаний лідером на ринку комерційного редагування растрових зображень та є, певною мірою, обличчям фірми Adobe.

Що стосується особливостей Adobe Photoshop, то варто зазначити, що, незважаючи на те, що спочатку програма створювалася як інструмент для професійного редагування зображень, орієнтований переважно на поліграфічну сферу, з часом її функціональні можливості значно розширилися. Photoshop швидко вийшов за межі традиційних задач ретуші та корекції, ставши універсальним середовищем для роботи з графічними матеріалами різного типу. Завдяки своїй гнучкості та широкому набору інструментів він отримав важливе місце у веб-дизайні, де використовується для створення макетів сторінок, інтерфейсів, елементів навігації та графічних компонентів сайтів [18].

Таким чином, Adobe Photoshop не лише постійно змінюється та вдосконалюється, але й активно адаптується до нових вимог індустрії. Програма відображає тенденції розвитку цифрового дизайну, інтегруючи інноваційні інструменти та технології. Завдяки цьому Photoshop залишається одним із ключових рішень для професіоналів, які працюють із графікою, мультимедіа та веб-інтерфейсами, забезпечуючи високу якість роботи та гнучкість при створенні сучасних цифрових продуктів.

2.2.4 Середовище розробки Visual Studio Code

Visual Studio Code – це потужний інструмент для створення, редагування та налагодження різноманітних веб-додатків та програм для хмарних систем [19]. Розроблений компанією Microsoft, Visual Studio Code є безкоштовним і має версії для платформ Windows, Linux та OS X, що робить його доступним для широкого кола користувачів.

Вперше Visual Studio Code був представлений у квітні 2015 року на конференції Build 2015, відзначивши собою перше кросплатформне середовище розробки у родині продуктів Visual Studio. Цей інструмент базується на відкритому проєкті Atom, що розробляється компанією GitHub, використовуючи Atom Shell, який в свою чергу використовує браузерний движок Chromium та Node.js. Цікаво, що хоча Visual Studio Code використовує кодову базу Atom, ця інформація не відображається на офіційному веб-сайті, у прес-релізах або в офіційному блозі.

Редактор Visual Studio Code має вбудований зневаджувач, інструменти для роботи з Git, а також засоби для рефакторингу, навігації по коду, автодоповнення та контекстної підказки [19]. Він підтримує розробку для різних платформ, включаючи ASP.NET та Node.js, і позиціонується як легковагове рішення, яке не вимагає повноцінного інтегрованого середовища розробки. Серед підтримуваних мов та технологій включаються JavaScript, C++, C#, TypeScript, jade, PHP, Python, XML, Batch, F#, DockerFile, Coffee Script, Java, HandleBars, R, Objective-C, PowerShell, Luna, Visual Basic, Markdown, JSON, HTML, CSS, LESS та SASS, Naxe.

Visual Studio Code постійно оновлюється та розвивається, випускаючи нові версії з новими функціями та поліпшеннями. Команда розробників активно співпрацює зі спільнотою користувачів, щоб забезпечити відповідь на їхні потреби та вимоги.

Однією з ключових переваг Visual Studio Code є його розширюваність. Інші розробники можуть створювати власні розширення для покращення функціональності редактора, що дозволяє адаптувати його до конкретних

потреб користувачів та робочих процесів. Завдяки цій гнучкості, Visual Studio Code широко використовується як для роботи над великими проєктами, так і для невеликих скриптів, і він завоював популярність серед розробників з усього світу.

2.2.5 Онлайн-редактор Figma

Figma [20] – це інноваційний векторний онлайн-сервіс, призначений для розробки інтерфейсів та прототипування, що розвивається компанією з такою ж назвою. Він став справжньою революцією в індустрії дизайну, завдяки своїм унікальним можливостям спільної роботи та інтуїтивно зрозумілому інтерфейсу. Перевагою Figma є можливість працювати як у веб-браузері, так і у вигляді клієнтського додатку на десктопі користувача, що робить його доступним для роботи в будь-який зручний час і місце.

Цей сервіс дозволяє зберігати всі файли онлайн, що значно спрощує процес спільної роботи та забезпечує безпеку даних. Крім того, Figma може бути використаний як для створення простих прототипів і дизайн-систем, так і для складних проєктів, таких як розробка мобільних додатків або створення веб-порталів [21].

У світі дизайну та розробки програмного забезпечення Figma завоював шалену популярність завдяки своїй потужній функціональності та швидкому темпу розвитку. У 2018 році він став одним із інструментів, які найбільш швидко розвиваються для розробників та дизайнерів, і з тих пір лише підтверджує своє лідерство в цій сфері.

Ключовою особливістю Figma є можливість спільної роботи над проєктами в реальному часі. Кілька користувачів можуть одночасно редагувати документ, бачати зміни, які роблять інші учасники. Це робить процес командної роботи більш ефективним та продуктивним, а також дозволяє швидко здійснювати обговорення та затвердження ідей.

Крім того, Figma має велику бібліотеку готових компонентів та елементів дизайну, які можна використовувати для швидкої побудови

інтерфейсу [20]. Це значно економить час на створення нових елементів з нуля і дозволяє швидше переходити до стадії прототипування та тестування. В цілому, Figma став необхідним інструментом для будь-якого дизайнера чи розробника, який цінує швидкість, зручність та можливість ефективної спільної роботи.

2.2.6 Фреймворк React Native

React.js [22] – це фреймворк з відкритим вихідним кодом, розроблений компанією Facebook, який призначений для створення динамічних вебзастосунків із використанням JavaScript. Основною перевагою React є компонентний підхід, який дозволяє створювати інтерфейси користувача з незалежних, багаторазових компонентів. Завдяки цьому розробка стає більш структурованою, а підтримка та масштабування проєктів – простішими та ефективнішими.

React.js використовує віртуальний DOM (Virtual DOM), що дає змогу мінімізувати кількість оновлень реального DOM і значно підвищує продуктивність застосунків. Це особливо важливо для великих вебсистем із великою кількістю інтерактивних елементів. React автоматично оновлює лише ті частини інтерфейсу, які змінилися, що забезпечує швидку реакцію інтерфейсу та комфорт користувачів.

Однією з ключових можливостей React.js є використання JSX – спеціального синтаксису, який поєднує JavaScript та HTML-подібну розмітку. Це спрощує написання коду, оскільки розробники можуть описувати структуру інтерфейсу безпосередньо в коді логіки. Такий підхід робить процес розробки більш інтуїтивним і зручним [23].

React.js також має потужну екосистему бібліотек і додаткових інструментів, таких як React Router для керування маршрутизацією, Redux або Recoil для управління станом застосунку, а також безліч компонентних бібліотек для швидкого створення інтерфейсів. Використання таких інструментів значно прискорює розробку та дозволяє створювати

масштабовані веб-рішення корпоративного рівня.

Ще однією перевагою React є його активна спільнота. Тисячі розробників у всьому світі постійно вдосконалюють бібліотеку, створюють плагіни, розширення та навчальні матеріали. Це сприяє швидкому вирішенню проблем і полегшує старт для новачків [24].

React.js широко використовується великими компаніями, такими як Facebook, Netflix, Uber, Airbnb та багатьма іншими, що свідчить про його стабільність, гнучкість і здатність працювати під великими навантаженнями. Крім того, React постійно розвивається, отримуючи нові можливості, як-от React Hooks, що спрощують роботу з логікою стану та життєвим циклом компонентів.

У поєднанні з серверним рендерингом через Next.js або схожі фреймворки React.js дозволяє створювати високопродуктивні веб-застосунки з чудовою SEO-оптимізацією. Це робить React.js універсальним інструментом як для створення односторінкових застосунків (SPA), так і для комплексних веб-платформ. Завдяки своїй гнучкості, продуктивності та великій спільноті React.js залишається одним із найпопулярніших інструментів сучасної веб-розробки [24].

2.2.7 Бібліотека стилів Bootstrap

Bootstrap – це не лише зручний і швидкий спосіб зробити сайт адаптивним, але й ефективний з точки зору управління проектом [25]. Завдяки стандартизації компонентів і стилів, розробники можуть швидко впроваджувати нові функції та покращення без необхідності переписувати величезні частини коду. Це особливо корисно в командній роботі, коли кілька розробників працюють над одним проектом.

Проте, важливо зазначити, що Bootstrap має свої обмеження та недоліки. Один із них – це ризик використання стандартних компонентів, які можуть зробити сайт схожим на інші. Для того, щоб відрізнити сайт від інших, іноді потрібно використовувати інші рішення або налаштовувати Bootstrap

компоненти. Крім того, завдяки популярності Bootstrap, існує ризик, що сайт може виглядати застарілим або масовим, особливо якщо використовується стандартні стилі без значних модифікацій [26].

Ще однією проблемою, пов'язаною з використанням Bootstrap, може бути його вага. Так як у нього включено багато компонентів та стилів для різних потреб, файл CSS Bootstrap може бути великим і сповненим непотрібними стилями для конкретного проєкту. Це може вплинути на швидкість завантаження сторінки, особливо на мобільних пристроях з повільним інтернет-з'єднанням.

Нарешті, важливо розуміти, що Bootstrap – це всього лише інструмент. Він може значно полегшити вам роботу, але не замінить професійного дизайну та розробки. Щоб створити дійсно привабливий та ефективний веб-сайт, важливо мати розуміння основ дизайну та веб-розробки, а також уміння застосовувати їх у практиці. Таким чином, хоча Bootstrap є потужним і корисним інструментом, він має бути використаний з обережністю та розумінням його можливостей та обмежень [25].

2.2.8 Брокер Mosquitto

Mosquitto – це легкий брокер протоколу MQTT, створений для обміну повідомленнями між пристроями в мережі Інтернету речей (IoT). Mosquitto надає просту та ефективну платформу для обміну даними між різними пристроями, забезпечуючи надійну та масштабовану комунікацію [27].

MQTT був розроблений для пристроїв з обмеженими ресурсами та низькою пропускну здатністю мережі. Брокер Mosquitto відповідає цим вимогам, оскільки споживає мінімальні ресурси та може працювати навіть на малопотужних пристроях, таких як Raspberry Pi. Ця легкість і ефективність роблять Mosquitto ідеальним рішенням для проєктів IoT, де необхідно з'єднати численні датчики, контролери та інші пристрої [28].

Однією з основних функцій Mosquitto є можливість керування підключеннями клієнтів. Він підтримує одночасне підключення великої

кількості клієнтів, дозволяючи їм підписуватися на певні теми та отримувати повідомлення, коли вони публікуються іншими клієнтами. Це створює гнучку та динамічну систему комунікації, де пристрої можуть швидко та ефективно обмінюватися інформацією.

Mosquitto підтримує всі три рівні якості обслуговування (QoS) в MQTT, що дозволяє користувачам вибирати між швидкістю передачі та надійністю доставки повідомлень. QoS "0" забезпечує найшвидшу передачу без гарантії доставки, QoS "1" гарантує доставку принаймні одного разу, а QoS "2" – найнадійніший рівень, який гарантує, що повідомлення буде доставлено рівно один раз.

Нарешті, Mosquitto має активну та підтримуючу спільноту, а також добре документоване API, що спрощує інтеграцію та використання в різних проєктах. Завдяки відкритому вихідному коду та широкій підтримці, Mosquitto став популярним вибором серед розробників IoT-систем, забезпечуючи ефективну та надійну платформу для обміну повідомленнями в різних середовищах та додатках [29].

2.2.9 Мова програмування JavaScript

JavaScript [30] – це динамічна, високорівнева мова програмування, яка використовується для розробки інтерактивних та функціональних веб-сторінок. Вона є однією з трьох основних технологій, на яких базується сучасний веб (поряд з HTML та CSS), і забезпечує роботу більшості інтерактивних елементів на веб-сторінках, таких як форми, кнопки, анімації та інші інтерактивні функції.

JavaScript вперше з'явився в 1995 році завдяки розробнику Brendan Eich з компанії Netscape. Спочатку мова була створена для браузера Netscape Navigator, але з часом стала стандартом для всіх основних веб-браузерів. Стандартом для JavaScript є ECMAScript, який регулярно оновлюється, додаючи нові функції та можливості [30].

JavaScript відрізняється від багатьох інших мов програмування тим, що

є інтерпретованою мовою, тобто код виконується безпосередньо браузером без необхідності попередньої компіляції. Це робить його дуже гнучким та зручним для швидкого розробки та тестування веб-аплікацій.

Мова підтримує об'єктно-орієнтоване, імперативне та функціональне програмування, що дозволяє розробникам використовувати різні підходи в залежності від вимог проекту [31]. JavaScript широко використовується для створення як простих веб-сторінок, так і складних веб-додатків з багатою функціональністю.

Важливою особливістю JavaScript є можливість взаємодії з HTML та CSS, що дозволяє динамічно змінювати вміст і стиль веб-сторінок на основі дій користувачів. Це робиться через Document Object Model (DOM), який є представленням структури HTML документа в пам'яті браузера.

З розвитком технологій JavaScript став також використовуватися для серверної частини додатків завдяки таким платформам як Node.js. Це дозволяє використовувати одну мову програмування як для клієнтської, так і для серверної частини додатків, що спрощує процес розробки та підтримки коду.

Важливу роль у розвитку JavaScript відіграють різноманітні бібліотеки та фреймворки, такі як jQuery, React, Angular та Vue.js, які полегшують розробку складних веб-додатків та забезпечують високу продуктивність і масштабованість. Ці інструменти надають розробникам можливість використовувати готові рішення для типових задач, що значно скорочує час розробки [31].

JavaScript постійно еволюціонує, отримуючи нові можливості та вдосконалення. Останні версії мови додають підтримку нових синтаксичних конструкцій, функцій та API, що дозволяє розробникам створювати ще більш потужні та ефективні додатки. Завдяки своїй універсальності та широкому спектру застосувань, JavaScript залишається однією з найпопулярніших мов програмування у світі.

2.2.10 СУБД PhpMyAdmin

phpMyAdmin [32] – це безкоштовний інструмент з відкритим кодом, написаний на PHP, який призначений для адміністрування MySQL та MariaDB через веб-інтерфейс. Він дозволяє користувачам легко виконувати різноманітні операції з базами даних, такі як управління базами даних, таблицями, індексами, користувачами, привілеями тощо.

phpMyAdmin зазвичай використовується системними адміністраторами, розробниками та іншими ІТ-фахівцями для спрощення управління базами даних без необхідності використання командного рядку. Він широко використовується завдяки своїй зручності та потужним можливостям [33].

3 СТРУКТУРА ОСНОВНИХ КОМПОНЕНТІВ СИСТЕМИ

3.1 Спосіб зберігання даних

Для зберігання даних було використано базу даних, а саме MySQL. Вся інформація зберігається в ній – записи, показники з датчиків, інформація про користувачів та ін. Для управління базою даних використовується система керування базами даних phpMyAdmin [12] (рис. 2.1), яка надає графічний веб-інтерфейс для управління базою MySQL. У більшості хостинг-провайдерів phpMyAdmin [13] можна знайти в панелі управління хостингом, що дозволяє користувачам швидко отримати доступ до бази даних і виконувати операції з нею. Структура бази даних проста і зручна.

Таблиця	Тип	Заставлення	Розмір	Фрагментовані
auth_user	int(11) unsigned		32.0 КБ	-
BSU	int(11) unsigned		16.0 КБ	-
car	int(11) unsigned		8.0 КБ	-
classRecipe	int(11) unsigned		7.0 КБ	-
comp	int(11) unsigned		17.0 КБ	-
compMain	int(11) unsigned		4.0 КБ	-
customer	int(11) unsigned		3.0 КБ	-
dispatcher	int(11) unsigned		5.0 КБ	-
django_migrations	int(11) unsigned		19.0 КБ	-
driver	int(11) unsigned		7.0 КБ	-
mainState	int(11) unsigned		19.0 КБ	-
midstate	int(11) unsigned		3.0 КБ	-
mtgt_processor_product	int(11) unsigned		13.0 КБ	-
mtgt_processor_report	int(11) unsigned		11.0 КБ	-
mtgt_processor_reportcurrentloop	int(11) unsigned		8.0 КБ	-
mtgt_processor_reportcurrentloop	int(11) unsigned		28.0 КБ	-
order	int(11) unsigned		17.0 КБ	-
orderState	int(11) unsigned		897.0 КБ	-
plants	int(11) unsigned		6.0 КБ	-
product	int(11) unsigned		19.0 КБ	-
recComp	int(11) unsigned		3.0 КБ	-
recFrost	int(11) unsigned		3.0 КБ	-
recFrost	int(11) unsigned		4.0 КБ	-
recMobility	int(11) unsigned		2.0 КБ	-
recMasks	int(11) unsigned		9.0 КБ	-
recMasks	int(11) unsigned		12.0 КБ	-
recStrength	int(11) unsigned		11.0 КБ	-
recWat	int(11) unsigned		5.0 КБ	-
report	int(11) unsigned		316.0 КБ	-
reportCurrentLoop	int(11) unsigned		21.196.0 КБ	-

Рисунок 3.1 – Таблиці бази даних

Управління та обслуговування бази даних проходить за допомогою спеціальної мови SQL. З її допомогою можливе виконання наступних основних операцій в ній: створення нової таблиці; додавання нових записів до таблиці; зміна та видалення записів; зміна структур таблиць БД; вставка нових рядків; зміна значень полів рядка або набору рядків; видалення окремого рядка або набору рядків.

3.2 Програмна реалізація системи

Програмна реалізація системи передбачає створення комплексного серверно-клієнтського рішення, яке забезпечує обробку даних у реальному часі, їх синхронізацію та подальше відображення у web-інтерфейсі. На серверній частині функціонує модуль взаємодії з базою даних, що приймає, обробляє та зберігає інформацію, що надходить від виробничих систем, а також від користувачів через web-додаток. Для цього застосовано архітектурний підхід, заснований на API-ендпоінтах, що дозволяє чітко розмежувати логіку бізнес-процесів і шар доступу до даних. Клієнтська частина системи формує запити до серверу, отримує актуальні дані та відображає їх у зручному інтерфейсі диспетчера. Такий підхід забезпечує гнучкість, масштабованість і стабільність роботи програмного комплексу, а також дозволяє легко розширювати функціональність системи відповідно до виробничих потреб.

3.2.1 Обґрунтування вибору системи керування базами даних

Метою написання даного проєкту є автоматизованої системи технологічних процесів “бетонного заводу”. Для реалізації цієї задачі була обрана реляційна модель даних. Усі запити до бази даних виконуються у програмному коді проєкту за допомогою ORM, зручну роботу з якою дає мова програмування на PHP [14].

Для взаємодії веб-застосунком, було розроблено низку програмних запитів у базу даних в форматі API, що формуються зі сторони клієнта.

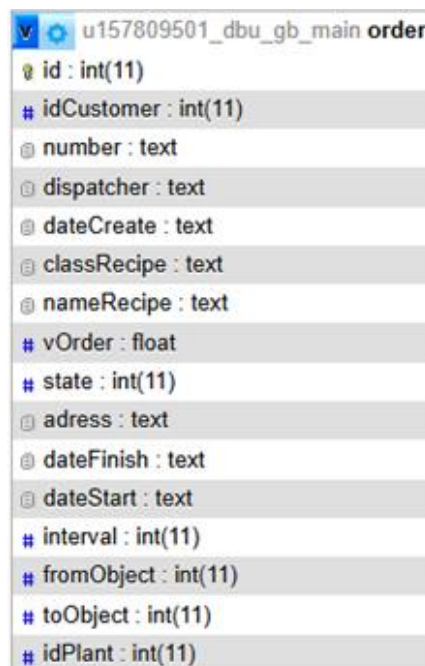
Маючи список таблиць в базі даних і об'єктів в програмі, система автоматично перетворить запити з одного вигляду в інший. В результаті запиту будь-якого об'єкта необхідний SQL-запит буде сформований та виконаний, а результати "магічним" чином перетворені в об'єкти "номер телефону" всередині програми.

З погляду програміста система повинна виглядати як постійне сховище

об'єктів. Він може просто створювати об'єкти та працювати з ними як завжди, а вони автоматично зберігатимуться в реляційній базі даних [15].

3.2.2 Опис таблиць

Схема бази даних – це її структура, описана на формальній мові, яка підтримується у СУБД [15]. В реляційних базах даних схема визначає таблиці, поля в кожній таблиці (зазвичай із зазначенням їх назви, типу, обов'язковості), і обмеження цілісності (первинний, потенційні та зовнішні ключі, інші обмеження). Поняття схеми бази даних відіграє ту ж роль, що і поняття теорії у численні предикатів. Модель цієї "теорії" точно відповідає базі даних, яку можна представити як математичний об'єкт. Таким чином, схема може містити формули, що представляють обмеження цілісності спеціально для застосунків і обмеження спеціально для типу бази даних, які виражені однією мовою баз даних. Хоча схема визначена на мові бази даних у вигляді тексту, термін часто використовується для позначення графічного представлення структури бази даних. Основними об'єктами графічного представлення схеми є таблиці (рис. 2.3) та зв'язки, які визначаються зовнішніми ключами, наприклад візьмемо таблицю Order.



Column Name	Data Type	Constraints
id	int(11)	Primary Key
idCustomer	int(11)	Foreign Key
number	text	
dispatcher	text	
dateCreate	text	
classRecipe	text	
nameRecipe	text	
vOrder	float	
state	int(11)	
adress	text	
dateFinish	text	
dateStart	text	
interval	int(11)	
fromObject	int(11)	
toObject	int(11)	
idPlant	int(11)	

Рисунок 3.3 – Таблиця Order бази даних

Розглянемо опис таблиць у базі даних. У проєкті, що розробляється, використовується 29 таблиць. Оскільки переважна кількість таблиць це довідники, котрі мають стандартну структуру полів, ми розглянемо тільки основні, а саме: Order, Product, а також TTN, отже розглянемо їх детальніше.

Таблиця "Orders" (рис. 2.4) містить інформацію про замовлення на замішування бетону. У цій таблиці зберігаються різноманітні дані про замовлення на заводі, замовника, обсяг, суміщі, рецепти бетону, статус, цех, диспетчер, номер замовлення, тощо. Ця таблиця є невід'ємною складовою структури бази даних проєкту, і є однією із основних таблиць.


#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково
1	id 	int(11)			Ні	Немає		AUTO_INCREMENT
2	idCustomer	int(11)			Так	NULL		
3	number	text	utf8mb4_unicode_ci		Так	NULL		
4	dispatcher	text	utf8mb4_unicode_ci		Так	NULL		
5	dateCreate	text	utf8mb4_unicode_ci		Так	NULL		
6	classRecipe	text	utf8mb4_unicode_ci		Так	NULL		
7	nameRecipe	text	utf8mb4_unicode_ci		Так	NULL		
8	vOrder	float			Так	NULL		
9	state	int(11)			Так	0		
10	adress	text	utf8mb4_unicode_ci		Так	NULL		
11	dateFinish	text	utf8mb4_unicode_ci		Так	NULL		
12	dateStart	text	utf8mb4_unicode_ci		Так	NULL		
13	interval	int(11)			Так	NULL		
14	fromObject	int(11)			Так	NULL		
15	toObject	int(11)			Так	NULL		
16	idPlant	int(11)			Так	NULL		

Рисунок 3.4 – Таблиця "Order"

Наступною таблицею розглянемо "TTN" (рисунок 2.5), за ієрархією вона є дочірньою таблицею Order, і зберігає дані заявок на змішування сумішей. Заявки це окремий домен, котрий має певну кількість станів, такі як: "Відправлено в АРМ БСУ", "Отримано АРМ БСУ", "У черзі", "У виробництві", "Виконано", "Звіт відправлено", "Звіт отримано", "Звіт отримано", "Звіт збережено", "Квітування збережено", "Корекція", "Квітування корекція",

"Видалено", "Квітування видалено".

Таблиця також містить інші параметри зарезервовані у посібниках. Після вдалого проходження виробничого циклу, успішно виконана заявка, має відсилати на відповідний звіт, дані з якого описані в таблиці "Product".


#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково
1	id 	int(11)			Ні	Немає		AUTO_INCREMENT
2	date	datetime			Так	NULL		
3	idPlant	int(11)			Так	NULL		
4	idOrder	int(11)			Так	NULL		
5	dispatcher	text	utf8mb4_unicode_ci		Так	NULL		
6	vProduct	int(11)			Так	NULL		
7	driver	text	utf8mb4_unicode_ci		Так	NULL		
8	car	text	utf8mb4_unicode_ci		Так	NULL		
9	finishAdress	text	utf8mb4_unicode_ci		Так	NULL		
10	finishDate	datetime			Так	NULL		
11	state	int(11)			Так	21		
12	isPause	int(11)			Так	0		
13	idProduct	int(11)			Так	NULL		
14	idBsu	int(11)			Так	NULL		

Рисунок 3.5 – Таблиця "TTN"

Таблиця "Product" (рис. 3.6) містить інформацію про звіти, отримані в результаті проходження технологічного процесу. Окрім полів з даними які відсилають на посібники, вона також зберігає дані використаних матеріалів по циклам на виробництві, тобто в сирому виді, дані з арі мають велику кількість масивів даних, котрі система розпаршує і утворює певну кількість записів у таблицях використаних матеріалів, з котрих, в тому числі, і формується потім остаточний вигляд звіту з використаних матеріалів, а також даних з посібників.

Таким чином була розроблена детальна та продумана архітектура бази даних для автоматизованої системи управління технологічними процесами "бетонного заводу". Вона забезпечує зручне, надійне та ефективне зберігання всієї необхідної інформації, яка використовується під час роботи виробництва.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково
<input type="checkbox"/>	1 id	int(11)			Hi	Немає		AUTO_INCREMENT
<input type="checkbox"/>	2 dateStart	datetime			Hi	Немає		
<input type="checkbox"/>	3 timeEnd	datetime			Hi	Немає		
<input type="checkbox"/>	4 vProduct	float			Hi	Немає		
<input type="checkbox"/>	5 loopNumber	int(11)			Hi	Немає		
<input type="checkbox"/>	6 vLoop	float			Hi	Немає		
<input type="checkbox"/>	7 driver	text	utf8mb4_unicode_ci		Hi	Немає		
<input type="checkbox"/>	8 car	text	utf8mb4_unicode_ci		Так	NULL		
<input type="checkbox"/>	9 classRecipe	text	utf8mb4_unicode_ci		Hi	Немає		
<input type="checkbox"/>	10 nameRecipe	text	utf8mb4_unicode_ci		Hi	Немає		
<input type="checkbox"/>	11 recipe	text	utf8mb4_unicode_ci		Hi	Немає		
<input type="checkbox"/>	12 idTtn	int(11)			Hi	Немає		
<input type="checkbox"/>	13 timeStart	datetime			Hi	Немає		
<input type="checkbox"/>	14 num_loop	int(11)			Hi	Немає		
<input type="checkbox"/>	15 idPlant	text	utf8mb4_unicode_ci		Hi	Немає		
<input type="checkbox"/>	16 indProduct	int(11)			Hi	Немає		

Рисунок 3.6 – Таблиця "Product"

Усі дані, що надходять у систему, фіксуються та зберігаються у базі даних протягом визначеного проміжку часу відповідно до встановлених параметрів і правил. Структура бази даних організована таким чином, щоб забезпечити швидкий доступ до потрібної інформації, можливість її обробки, аналізу та подальшого використання для контролю й оптимізації технологічних процесів на бетонному заводі.

3.2.3 Опис Web-додатку системи технологічних процесів

У межах даного проєкту розробляється web-додаток, який виступає основним інструментом диспетчера виробничої системи. Додаток призначений для організації, контролю та управління основними процесами, пов'язаними з обробкою замовлень і заявок, а також веденням довідкової інформації. Використання web-технологій забезпечує зручний доступ до системи з будь-якого робочого місця в межах підприємства.

Диспетчер працює у спеціально розробленій інформаційній зоні

відповідальності, що забезпечує ефективне виконання його функціональних обов'язків. Основним призначенням системи є автоматизація управлінських процесів, зниження кількості ручних операцій і підвищення точності обліку виробничих даних.

До функціональних можливостей диспетчера належать:

- створення, редагування та видалення замовлень;
- створення, редагування та видалення заявок, пов'язаних із конкретними замовленнями;
- створення, редагування та видалення довідників, що містять службу та нормативну інформацію;
- перегляд замовлень, заявок, довідників і формування звітів за заданими параметрами.

Інтерфейс web-додатка побудований із урахуванням принципів зручності, логічної структури та швидкого доступу до основних функцій. Він включає такі основні сторінки:

- сторінку зі списком замовлень та історією їх виконання;
- сторінку окремого замовлення зі списком пов'язаних заявок і хронологією їх змін;
- сторінку звітів, що містить узагальнені дані для аналізу ефективності роботи;
- сторінки довідників, які забезпечують централізоване керування службовою інформацією.

Розроблений web-додаток покликаний оптимізувати роботу диспетчера, зробити процес управління замовленнями більш прозорим і контрольованим, а також забезпечити швидкий доступ до актуальної інформації для прийняття ефективних управлінських рішень.

3.3 Підсумки по розробленню структури компонентів системи

Було визначено структуру основних компонентів системи та обґрунтовано вибір технологій для її реалізації. Для зберігання даних використано реляційну базу MySQL, що забезпечує надійність, швидку обробку запитів та зручне керування через phpMyAdmin. Архітектура БД побудована таким чином, щоб гарантувати цілісність даних, логічні зв'язки між таблицями та ефективну роботу з великими обсягами виробничої інформації.

Було розроблено основні таблиці – Orders, TTN і Product, які формують ядро системи та містять ключові дані щодо замовлень, заявок і виробничих звітів. Їх структура забезпечує повний цикл фіксації та обробки процесів бетонного виробництва.

Програмна реалізація базується на використанні API-запитів, що спрощує роботу з БД. Такий підхід забезпечує масштабованість, надійність і підтримку системи.

Розроблений web-додаток виступає інструментом диспетчера, надаючи можливості управління замовленнями, заявками, довідниками та звітністю. Інтерфейс побудовано таким чином, щоб забезпечити швидкий доступ до ключових функцій і підвищити ефективність роботи користувача.

4 РЕАЛІЗАЦІЯ СИСТЕМИ АВТОМАТИЗАЦІЇ ТЕХНОЛОГІЧНИХ ПРОЦЕСІВ

4.1 Складові системи автоматизації технологічних процесів

Система поділяється на 3 важливі складові такі як: серверна, служба, котра працює постійно, отримує і обробляє дані та web-додаток для виведення інформації. Служба являє собою посередника між сервером і web-додатком, що підключена по протоколу MQTT та брокеру Mosquitto, і зберігає дані в базу даних по HTTP.

Серверна частина проекту реалізована на одній фізичній машині, яка виконує три основні функціональні ролі.

- комунікація;
- збереження даних;
- відображення інформації.

Модуль комунікації відповідає за взаємодію між сервером та системою збору даних.

Основні завдання:

- підтримка постійного з'єднання з MQTT-брокером;
- приймання повідомлень, які надсилаються з панелі керування (контролера або IoT-пристрою);
- відстеження актуальності та цілісності отриманих даних;
- передача повідомлень у внутрішні сервіси для подальшої обробки.

Функція збереження даних відповідає за обробку та структурування вхідних повідомлень.

Основні завдання:

- розпізнавання типу даних, отриманих з MQTT-брокера;
- попередня валідація та нормалізація даних;

- збереження інформації у базі даних відповідно до визначеної структури зберігання (MySQL);
- логування операцій запису для забезпечення надійності та можливості аудиту.

Модуль відображення інформації (Web-інтерфейс) відповідає за візуалізацію даних та забезпечує доступ користувачів до системи через веб-інтерфейс.

Основні завдання:

- відображення поточних та історичних даних згідно з вимогами користувача;
- реалізація відображення звітів та поточних таблиць для моніторингу стану системи;
- можливість фільтрації та пошуку даних;
- інтеграція з API серверної частини для динамічного оновлення інформації.

4.2 Програмна реалізація підключення та зберігання даних з топика через MQTT протокол

У додатку Б представлено код реалізації підключення та збереження даних у системі.

Цей код реалізує MQTT-клієнт у середовищі Django, який приймає та обробляє повідомлення з брокера, зберігаючи отримані дані у базі даних. Після підключення до MQTT-сервера програма підписується на низку тем, що містять інформацію про звіти та виробництво бетонного заводу. Коли надходить нове повідомлення, воно розбирається, з нього виділяються дані про транспортні накладні, стан виробництва, продукцію та поточні цикли. Скрипт перевіряє наявність записів у базі, створює або оновлює їх, а також надсилає зворотні повідомлення у відповідні теми при певних станах. Передбачено автоматичне відновлення підключення до бази та захист від

запуску кількох копій процесу за допомогою блокувального файлу.

Даний код оновлюється кожен хвилину, завдяки Cron задачі, що по часовому триггеру активує скрипт.

4.3 Програмна реалізація web-додатку

Web-додаток у даному проєкті ділиться на front-end та back-end частину і представляє собою застосунок, для відображення, фільтрації, зберігання, та редагування даних. Він забезпечує зручний доступ до звітів про виконання технологічних операцій, аналіз використаних матеріалів та візуалізацію динаміки виробничих показників, а також список замовлень і заявок по даним замовленням.

Додатково система реалізує механізми автоматичного оновлення інформації на основі даних, що надходять із зовнішніх джерел, зокрема через MQTT-брокер. Це дозволяє забезпечити оперативне відображення змін у статусах замовлень, параметрах виконання робіт та руху матеріалів. Завдяки цьому користувачі отримують актуальну інформацію у режимі реального часу, що значно підвищує ефективність контролю та прийняття рішень у виробничому процесі.

4.3.1 Front-end модуль web-додатку

Цей модуль розроблено з використанням сучасних технологій JavaScript, з використанням бібліотеки React.js. Структура модуля, обумовлена стандартами написання clean code, передбаченою React. При написанні додатку було застосовано компонентний підхід.

Структура front-end модулю має вид:

- Src – головна папка з усім вихідним кодом проєкту, у якій зберігаються всі інші модулі;
- Config – містить файли налаштувань застосунку (шляхи, змінні середовища, API-ключі, базові константи);

- Domain – логіка предметної області: бізнес-моделі, сервіси, інтерфейси, правила обробки даних;
- Infrastructure – реалізація технічних частин: робота з API, базами даних, мережею, зовнішніми сервісами;
- Pages – окремі сторінки застосунку (компоненти, що відповідають маршрутам, наприклад Orders, Reports);
- Public – статичні файли, доступні напряму (index.html, favicon, зображення);
- Ui – набір універсальних інтерфейсних компонентів (кнопки, модальні вікна, форми, панелі).

4.3.2 Back-end модуль web-додатку

Цей модуль розроблено з використанням мови PHP. Структура модуля, обумовлена стандартами написання clean code. При написанні додатку було застосовано компонентний підхід, основною задачею back-end у даному проєкті є надання API, для front-end, шляхом застосування заданих запитів.

Структура back-end модулю має вид:

- API – містить логіку обробки запитів від клієнтів: контролери, маршрути, ендпоінти, що забезпечують взаємодію між фронтендом і сервером;
- Database – відповідає за роботу з базою даних: підключення, моделі, запити, міграції та збереження даних;
- MQTT – модуль для роботи з протоколом MQTT: підписка на топіки, отримання та обробка повідомлень, взаємодія із клієнтами або сервісами в реальному часі.

4.4 Розрахунок стійкості системи

Автоматизована система технологічних процесів це програмне забезпечення (ПЗ) де є сервер і зовнішній пристрій, котрий по mqtt надсилає на сервер звіти з виконання замовлень. Замовлення попередньо створюються через web-інтерфейс.

Вхідні припущення (використані для всіх сценаріїв):

- а) середній розмір одного повідомлення (payload) = 1 КБ = 1024 байт;
- б) повідомлення зберігаються в БД (один INSERT = одне повідомлення);
- в) розглядається 3 сценарії:

1) низький: активних замовлень = 100, звітів на замовлення – 1 / година = 24 / доба;

2) середній: активних замовлень = 1000, звітів на замовлення – 1 / 10 хв = 144 / доба;

3) високий: активних замовлень = 5000, звітів на замовлення – 1 / хв = 1440 / доба;

г) день = 86400 секунд, година = 3600 секунд.

Результати розрахунків стійкості та навантаження автоматизованої системи.

1. Розрахунок кількості повідомлень на добу.

Кількість повідомлень визначається за формулою (4.1):

$$N_{\text{день}} = N_{\text{замовлення}} \cdot k_{\text{звіти}}, \quad (4.1)$$

де $N_{\text{замовлення}}$ – кількість замовлень;

$k_{\text{звіти}}$ – кількість звітних повідомлень на одне замовлення за добу.

Результати розрахунку подано в табл. 4.1.

Таблиця 4.1 – Кількість повідомлень на добу

Сценарій	Розрахунок	Значення, повід./добу
Низький	100×24	2 400
Середній	1000×144	144 000
Високий	5000×1440	7 200 000

2. Обсяг даних на добу.

Формула для обчислення обсягу даних:

$$V_{\text{байт}} = N_{\text{день}} \cdot 1024. \quad (4.2)$$

Переведення у мегабайти та гігабайти:

$$1 \text{ МБ} = 1\,048\,576 \text{ байт}, 1 \text{ ГБ} = 1024 \text{ МБ}.$$

Результати розрахунків обчислення обсягу даних зведені до таблиці 4.2.

Таблиця 4.2 – Обсяг даних на добу

Сценарій	Обсяг, байт/добу	Обсяг, МБ/добу	Обсяг, ГБ/добу
Низький	2 457 600	2,34	–
Середній	147 456 000	140,6	0,137
Високий	7 372 800 000	7 031,25	6,87

3. Інтенсивність надходження повідомлень (повідомлень/секунду)

Інтенсивність розраховується за формулою:

$$I = \frac{N_{\text{день}}}{86400} , \quad (4.3)$$

де 86400 – кількість секунд у добі.

Результати розрахунків середньої інтенсивності повідомлень та INSERT-операцій зведені до таблиці 4.3.

Таблиця 4.3 – Середня інтенсивність повідомлень та INSERT-операцій

Сценарій	повідомлень/с	Характеристика
Низький	0,028	≈ одне повідомлення за 36 с
Середній	1,67	≈ одне повідомлення за 0,6 с
Високий	83,33	Пікове навантаження ~83 INSERT/с

Результати розрахунків показують, що навантаження 80–90 INSERT/с є допустимим для оптимізованої SQL-БД.

4. Пропускна здатність MQTT-брокера.

Пропускна здатність MQTT-брокера розраховується за формулою:

$$B_c = \frac{V_{\text{байт}}}{86400} . \quad (4.4)$$

Результати розрахунків пропускної здатності брокера зведені до таблиці 4.4.

Таблиця 4.4 – Пропускна здатність брокера

Сценарій	Байт/с	КБ/с	МБ/с
Низький	28,44	0,0278	–
Середній	1706,67	1,67	–
Високий	85 312,5	83,33	0,081

5. Розмір буфера для 1 години простою.

Обчислення кількості повідомлень за годину:

$$N_{\text{год}} = \frac{N_{\text{день}}}{24}. \quad (4.5)$$

Обсяг буфера:

$$V_{\text{буфер}} = N_{\text{год}} \cdot 1024. \quad (4.6)$$

Результати розрахунків розміру буфера для 1 години простою зведені до таблиці 4.5.

Таблиця 4.5 – Буфер для збереження повідомлень (1 година)

Сценарій	Повідомлення/год	Обсяг, байт	Обсяг, МБ
Низький	100	102 400	0,1
Середній	6 000	6 144 000	5,86
Високий	300 000	307 200 000	293

Результати розрахунків показують, що максимальний необхідний буфер – близько 300 МБ.

6. Прогноз росту обсягу архіву за місяць

$$V_{\text{міс}} = V_{\text{день}} \cdot 30. \quad (4.7)$$

Таблиця 4.6 – Прогноз використання диску (30 діб)

Сценарій	ГБ/добу	ГБ/міс
Низький	0,002285	0,0686
Середній	0,137	4,119
Високий	6,868	206

Результати розрахунків показують, що у високому сценарії архів за місяць може досягати ≈ 206 ГБ.

7. Оцінка доступності системи.

Доступність одного сервера:

$$A = 99,9\% = 0,999. \quad (4.8)$$

Недоступність на рік:

$$T_{\text{недост}} = 8760 \cdot (1 - A) = 8,76 \text{ год.} \quad (4.9)$$

Для двох незалежних серверів:

$$A_{\text{сист}} = 1 - (1 - A)^2 = 0,999999. \quad (4.10)$$

Недоступність:

$$T_{\text{недост}} = 8760 \cdot 10^{-6} = 0,00876 \text{ год} \approx 31,5 \text{ с.} \quad (4.11)$$

Результати розрахунків показують, що використання реплікації дозволяє зменшити річну недоступність із 8,76 год до ≈ 32 секунд, що суттєво підвищує надійність автоматизованої системи.

Отже проведений розрахунок демонструє, що запропонована автоматизована система технологічних процесів бетонного заводу є стійкою до високих навантажень, забезпечує резервування даних у разі тимчасового простою, має прийнятні вимоги до пропускну здатності мережевих компонентів та може функціонувати з високим рівнем доступності при застосуванні серверної реплікації. Обчислені параметри підтверджують можливість надійної експлуатації системи у реальних виробничих умовах.

4.5 Забезпечення безпечних умов праці при створенні автоматизованої

системи

Під час розроблення автоматизованої системи технологічних процесів особлива увага приділяється забезпеченню безпечних умов праці як для працівників, що займаються створенням програмного продукту, так і для персоналу підприємства, який буде користуватися системою. Створення веб-застосунку передбачає роботу за комп'ютером, тому важливо організувати робоче місце відповідно до вимог охорони праці: забезпечити правильне розташування монітора, достатнє освітлення, ергономічне крісло та регулярні перерви для запобігання перевтомі. Безпека також стосується інформаційного середовища, адже система працює з виробничими даними підприємства, тому необхідно передбачити заходи захисту інформації, включаючи резервне копіювання, обмеження доступу та використання надійних засобів автентифікації [34].

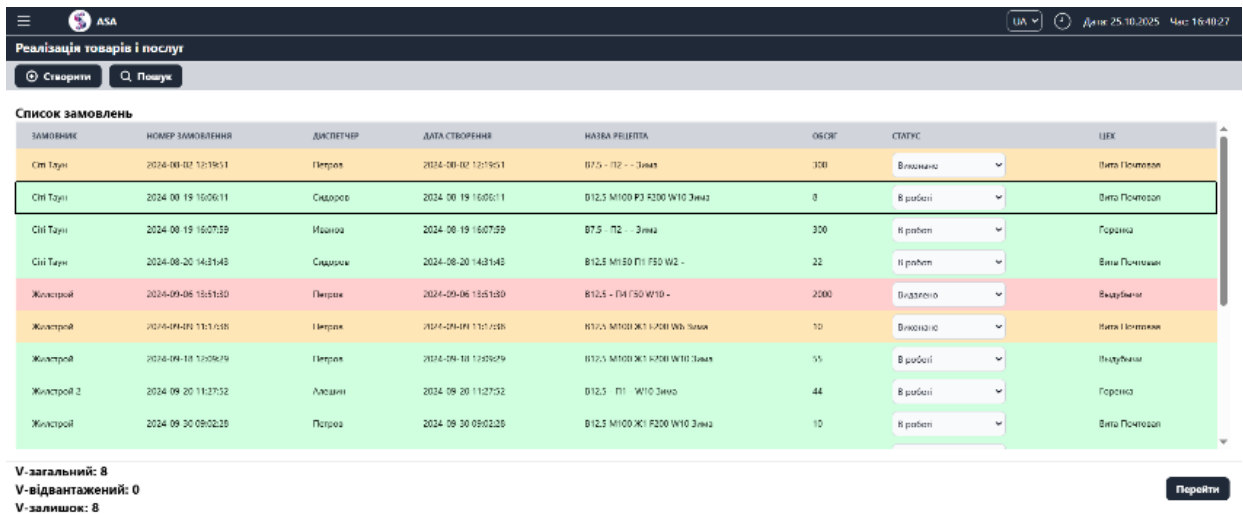
Для користувачів системи важливо мінімізувати ризики неправильного внесення даних, що може вплинути на технологічні процеси. Це забезпечується завдяки інтуїтивному інтерфейсу, повідомленням про помилки та надійним механізмам валідації. Веб-застосунок також має працювати стабільно, не створюючи загроз для персоналу, який залежить від коректності відображення технологічних даних. Використання сучасних програмних засобів, тестування та дотримання стандартів розробки сприяють підвищенню безпеки системи та комфортним умовам її експлуатації.

Крім того, під час впровадження автоматизованої системи враховуються питання технічної та експлуатаційної безпеки. Серверне обладнання повинно розміщуватися в умовах, що відповідають вимогам електробезпеки та пожежної безпеки, а доступ до нього має бути обмеженим. Передбачені засоби безперебійного живлення, моніторинг стану серверів та своєчасне технічне обслуговування. Такий підхід забезпечує стабільність роботи програмного комплексу та захищає працівників від можливих технічних ризиків [34].

5 ІНСТРУКЦІЯ КОРИСТУВАЧА

5.1 Інтерфейс Web-додатку

Перша сторінка, на який потрапляє користувач під час використання АРМу системи є розділ Реалізація товарів і послуг (рисунок 5.1). Диспетчер має перед собою список актуальних замовлень, а також навігаційне меню системи.



ЗАМОВНИК	НОМЕР ЗАМОВЛЕННЯ	ДИСПЕТЧЕР	ДАТА СТВОРЕННЯ	НАЗВА РЕЦЕПТА	ОБ'ЄКТ	СТАТУС	ЦЕНА
Сті Таун	2024-08-02 12:19:51	Петров	2024-08-02 12:19:51	B7.5 - П2 - Зміна	300	Виконано	Вито Почтовий
Сті Таун	2024-08-19 16:06:11	Сидоров	2024-08-19 16:06:11	B12.5 M100 P3 P200 W10 Зміна	8	В роботі	Вито Почтовий
Сті Таун	2024-08-19 16:07:59	Иванов	2024-08-19 16:07:59	B7.5 - П2 - Зміна	300	В роботі	Горська
Сті Таун	2024-08-20 14:31:43	Сидоров	2024-08-20 14:31:43	B12.5 M150 П1 P50 W2 -	22	В роботі	Вито Почтовий
Жилстрой	2024-09-06 13:51:20	Петров	2024-09-06 13:51:20	B12.5 - П1 P30 W10 -	2000	Виконано	Видубини
Жилстрой	2024-09-09 11:51:53	Петров	2024-09-09 11:51:53	B12.5 M100 Ж1 P200 W10 Зміна	10	Виконано	Вито Почтовий
Жилстрой	2024-09-10 12:09:19	Петров	2024-09-10 12:09:19	B12.5 M100 Ж1 P200 W10 Зміна	15	В роботі	Видубини
Жилстрой 2	2024-09-20 11:27:52	Лавин	2024-09-20 11:27:52	B12.5 - П1 - W10 Зміна	44	В роботі	Горська
Жилстрой	2024-09-30 09:02:28	Петров	2024-09-30 09:02:28	B12.5 M100 Ж1 P200 W10 Зміна	10	В роботі	Вито Почтовий

V-загальний: 8
V-відвантажений: 0
V-залишок: 8

Рисунок 5.1 – Інтерфейс web-додатку (Реалізація товарів і послуг)

5.2 Розділ реалізація товарів і послуг

Розглянемо детальніше складові сторінки Реалізація товарів і послуг, системи автоматизованої системи технологічних процесів “бетонного заводу” (рис. 5.2).

The screenshot shows the 'Реалізація товарів і послуг' application interface. At the top, there is a header with a logo, a search bar, and a language selector (UA) and date/time display (25.10.2025, 16:48:27). Below the header, there are buttons for 'Створити' and 'Пошук'. The main content area is titled 'Список замовлень' and contains a table with columns: 'ЗАМОВНИК', 'НОМЕР ЗАМОВЛЕННЯ', 'ДИСПЕТЧЕР', 'ДАТА СТВОРЕННЯ', 'НАЗВА РЕЦЕПТА', 'ОБСЯГ', 'СТАТУС', and 'ЦЕХ'. The table lists several orders with their respective details. At the bottom left, there are summary statistics: 'V-загальний: 8', 'V-відвантажений: 0', and 'V-залишок: 8'. At the bottom right, there is a 'Перейти' button. Callouts 1-9 point to specific UI elements: 1 (header menu), 2 (create button), 3 (filter button), 4 (selected row), 5 (status dropdown), 6 (language selector), 7 (date/time), 8 (summary statistics), and 9 (transition button).

ЗАМОВНИК	НОМЕР ЗАМОВЛЕННЯ	ДИСПЕТЧЕР	ДАТА СТВОРЕННЯ	НАЗВА РЕЦЕПТА	ОБСЯГ	СТАТУС	ЦЕХ
Спі Таун	2024-08-02 12:19:51	Петров	2024-08-02 12:19:51	B7.5 - П2 - - Зима	300	Виконано	Вита Почтовая
Спі Таун	2024-08-19 16:06:11	Сидоров	2024-08-19 16:06:11	B12.5 M100 P3 F200 W10 Зима	8	В роботі	Вита Почтовая
Спі Таун	2024-08-19 16:07:59	Иванов	2024-08-19 16:07:59	B7.5 - П2 - - Зима	300	В роботі	Горенка
Спі Таун	2024-08-20 14:31:43	Сидоров	2024-08-20 14:31:43	B12.5 M150 P11 F50 W2 -	22	В роботі	Вита Почтовая
Жилстрой	2024-09-06 13:51:30	Петров	2024-09-06 13:51:30	B12.5 - П4 F50 W10 -	2000	Видалено	Вьдубачи
Жилстрой	2024-09-09 11:17:38	Петров	2024-09-09 11:17:38	B12.5 M100 Ж1 F200 W6 Зима	10	Виконано	Вита Почтовая
Жилстрой	2024-09-18 12:09:29	Петров	2024-09-18 12:09:29	B12.5 M100 Ж1 F200 W10 Зима	55	В роботі	Вьдубачи
Жилстрой 2	2024-09-20 11:27:52	Алешин	2024-09-20 11:27:52	B12.5 - П1 - W10 Зима	44	В роботі	Горенка
Жилстрой	2024-09-30 09:02:28	Петров	2024-09-30 09:02:28	B12.5 M100 Ж1 F200 W10 Зима	10	В роботі	Вита Почтовая

V-загальний: 8
 V-відвантажений: 0
 V-залишок: 8

Перейти

Рисунок 5.2 – Складові сторінки Реалізація товарів і послуг

У цьому розділі (рисунок 5.2) можна виділити такі компоненти:

- вказівник 1: кнопка, що відкриває меню додатку (рисунок 5.3);
- вказівник 2: кнопка, що відкриває розділ для створення нового замовлення (рисунок 5.4);
- вказівник 3: кнопка, що відкриває модальне вікно для фільтрації замовлень (рисунок 5.5);
- вказівник 4: обране замовлення;
- вказівник 5: актуальний стан замовлення;
- вказівник 6: перемикач мови (українська/англійська);
- вказівник 7: актуальна дата та час;
- вказівник 8: обсяг продукції в обраному замовленні;
- вказівник 9: кнопка переходу на замовлення.

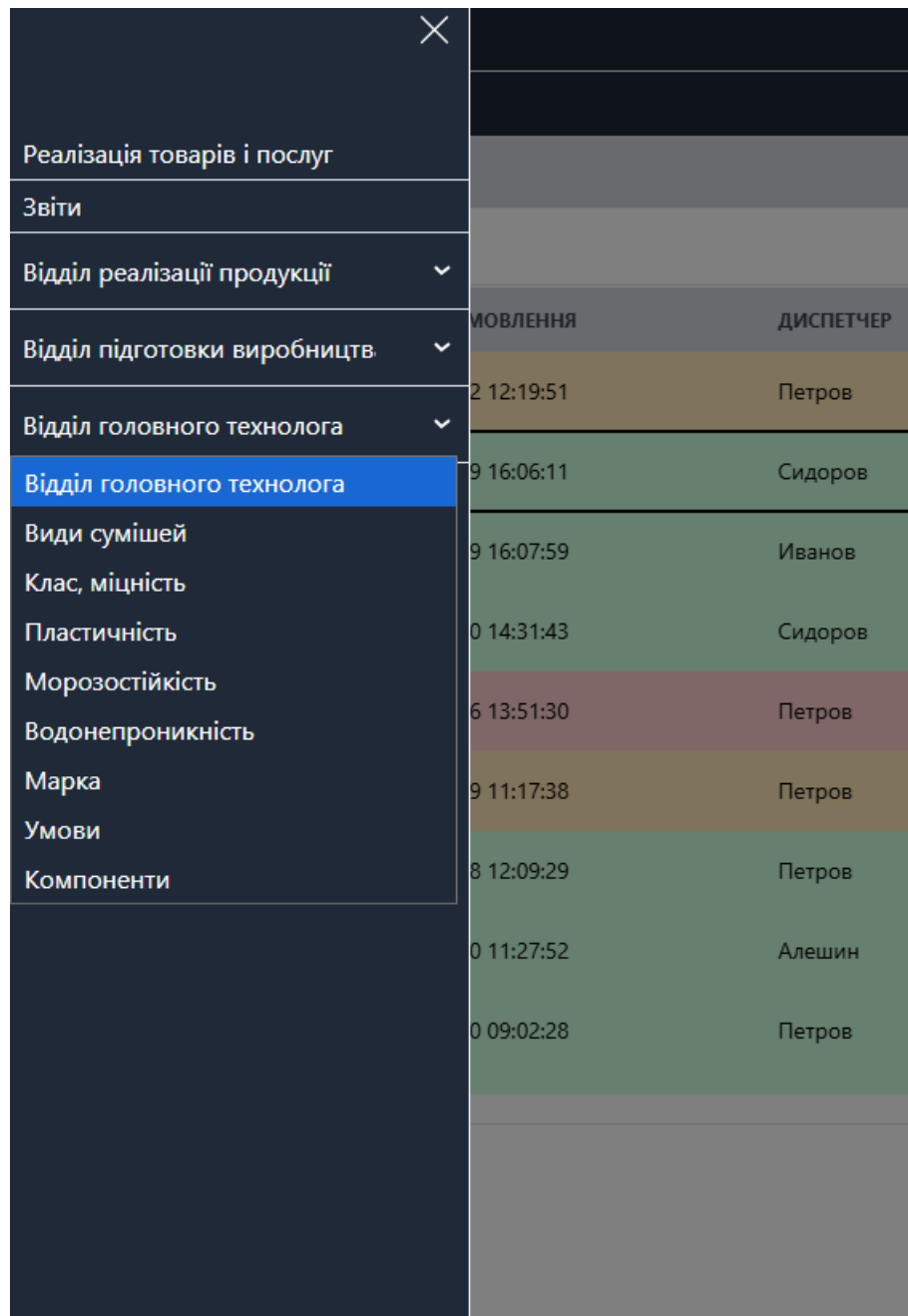


Рисунок 5.3 – Кнопка, що відкриває додаткове меню додатку

У додатковому меню можна виділити такі компоненти:

- посилання на розділ Реалізація товарів і послуг (рисунок 5.1);
- посилання на розділ Звіти (рисунок 5.6);
- посилання на довідники категорії Відділ реалізації продукції;
- посилання на довідники категорії Відділ підготовки виробництва;
- посилання на довідники категорії Відділ головного технолога.

Рисунок 5.4 – Розділ для створення нового замовлення

У даному розділі відбувається створення нового замовлення, шляхом вводу, або вибору доступних параметрів із довідників. Із складових цього інтерфейсу можна виділити:

- вказівник 1: кнопка, що відкриває модальне вікно створення додаткового параметру у відповідному довіднику;
- вказівник 2: кнопка, що відкриває випадаючий список для вибору параметру відповідного довідника (створення параметра у цьому розділі обмежено);
- вказівник 3: кнопка, що відкриває модальне вікно для вибору дати та часу;
- вказівник 4: поле вводу значення (у числовому форматі).

Рисунок 5.5 – Модальне вікно для фільтрації замовлень

У даному модальному вікні, можна ввести як один, так і усі, або декілька необхідних параметрів, для фільтрації списку замовлень

5.3 Розділ звіти

Цей розділ являє собою список отриманих звітів від PLC, що були сформовані під час виконання виробничих циклів, та відправлені на АРМ сервер на робоче місце диспетчеру.

1

2

3

4

5

6

Рисунок 5.6 – Розділ Звіти

Розглянемо детальніше складові сторінки Звіти, системи автоматизованої системи технологічних процесів “бетонного заводу”:

- вказівник 1: секція вводу параметрів для фільтрації звітів;
- вказівник 2: обраний звіт;
- вказівник 3: кнопка, що виконує пошук по заданим фільтрам;
- вказівник 4: кнопка, що очищає усі фільтри;
- вказівник 5: сформована калькуляція по використаним матеріалам та загальний обсяг в кубах;
- вказівник 6: сформована калькуляція розділена по циклам виробничого

процесу, що було проведено для даної заявки.

5.4 Розділ замовлення

У цьому розділі (рисунок 5.6) можна виділити такі компоненти:

- вказівник 1: кнопка збереження змін після редагування параметрів замовлення;
- вказівник 2: назва заводу, по якому створено замовлення;
- вказівник 3: секція параметрів замовлення та їх редагування;
- вказівник 4: секція з історією замовлення;
- вказівник 5: кнопка сортування заявок;
- вказівник 6: кнопка, що відкриває модальне вікно створення, редагування або видалення заявок;
- вказівник 7: обрана заявка;
- вказівник 8: випадаючий список зміни стану заявки;
- вказівник 9: кнопка встановлення паузи заявки;
- вказівник 10: кнопка посилення на звіт;
- вказівник 11: історія обраної заявки.

The screenshot displays the 'Деталі замовлення' (Order Details) page. At the top, there are navigation buttons and a search bar. Below that, the order details are shown, including the factory name 'Вита Почтова' (Vita Pochtova) and various parameters like volume and price. A table below lists the orders (TTN) with columns for order number, machine, date, dispatcher, driver, address, time, and status. The status history on the right shows the progression of the order from 'Створено' (Created) to 'Відправлено в виробництво' (Sent to production) and finally 'Виконано, звіт відправлено' (Completed, report sent).

НОМЕР ЗАЯВКИ	МАШИНА	ДАТА	ДИСПЕЧЕР	ВОДИЙ	АДРЕСА ДОСТАВКИ	ЧАС ДОСТАВКИ	ЦЕХ	СТАН	ОБСЯГ
612	ак 2239 мн	2025-09-25 18:22:14	Сидоров	Коваль	адреса	2025-09-25 18:22:00	ГБ Почайна	Звіт збережено	2
613	ак 2239 мн	2025-09-25 18:57:24	Петров	ОТЛАДКА	адреса	2025-09-25 18:57:00	ГБ Почайна	Звіт збережено	2
614	ак 2239 мн	2025-09-26 09:41:10	Петров	9	адреса	2025-09-26 09:40:00	ГБ Почайна	Квитування «Видалено»	2
615	ак 2239 мн	2025-09-26 09:42:06	Петров	9	адреса	2025-09-26 10:40:00	ГБ Почайна	Звіт збережено	2
616	ак 2239 мн	2025-09-29 10:00:46	Петров	Щеченко	адреса	2025-09-29 10:00:00	ГБ Почайна	Відправлено в вироб	2
617	ак 2239 мн	2025-09-29 10:34:32	Петров	Щеченко	адреса	2025-09-29 10:34:00	ГБ Почайна	Звіт збережено	6
618	ак 2239 мн	2025-09-29 10:43:05	Петров	Біньковський	адреса	2025-09-29 10:43:00	ГБ Почайна	Звіт збережено	2
619	ак 2239 мн	2025-09-29 10:58:36	Петров	ОТЛАДКА	адреса	2025-09-29 10:58:00	ГБ Почайна	Квитування «Видалено»	6

Рисунок 5.6 – Розділ Замовлення

5.5 Підсумки аналізу інтерфейсу системи

У цьому розділі було представлено детальний огляд інтерфейсу web-додатку автоматизованої системи технологічних процесів бетонного заводу. На основі проведеного опису можна зробити висновок, що інтерфейс є логічно організованим, інтуїтивно зрозумілим і орієнтованим на зручність роботи диспетчера. Усі основні розділи – від перегляду та фільтрації замовлень до формування звітів – реалізовані таким чином, щоб забезпечити швидкий доступ до потрібної інформації та можливість оперативного управління виробничими процесами.

Система дозволяє створювати, редагувати та контролювати замовлення, переглядати їхній стан, обсяг продукції та історію змін. Розділ звітів забезпечує доступ до даних, отриманих від PLC, що гарантує точне відображення ходу виробничих циклів та споживання матеріалів. Наявність фільтрів, модальних вікон і зручних інструментів вибору параметрів робить роботу у системі максимально ефективною та мінімізує ймовірність помилок.

Таким чином, інтерфейс web-додатку повністю відповідає функціональним вимогам системи, сприяє підвищенню продуктивності та забезпечує диспетчеру повний контроль над процесами реалізації продукції та обробки виробничих даних.

ВИСНОВКИ

У результаті роботи над кваліфікаційною роботою було досягнуто поставлену мету – розроблено автоматизовану систему керування технологічними процесами бетонного заводу, яка забезпечує збереження, обробку та аналіз даних щодо виконання виробничих операцій і використаних матеріалів. Було проведено роботу зі стендами АРМів та сервера, аналогічними тим, що використовуються на проекті з реальними задачами.

На робочому місці було проведено дослід і налагодження, відповідно вимогам до технологічних процесів бетонного заводу. Було пройдено усі можливі варіанти виконання циклів виробництва бетону, протестовано дозування сумішей, підраховано та порівняно з даними у звітах, що формуються на місці диспетчера.

Під час дослідження проведено детальний аналіз виробничих процесів, визначено ключові показники ефективності та побудовано структуру бази даних для зберігання технологічної інформації. Використання сучасних web-технологій – React, PHP та MySQL, що дало змогу створити функціональний веб-застосунок, який поєднує зручний інтерфейс користувача з надійною серверною логікою та ефективним управлінням даними. Розроблена система дозволяє диспетчеру здійснювати моніторинг виробничого процесу, контролювати використання матеріалів, швидко виявляти відхилення, формувати звіти та статистичні дані.

Проведене тестування підтвердило стабільність, масштабованість і гнучкість системи, що дозволяє адаптувати її під потреби конкретного виробництва. Застосування розробленої системи забезпечує підвищення рівня автоматизації, покращення якості готової продукції та зростання конкурентоспроможності бетонного заводу.

Отримані результати роботи можна віднести до Цілі сталого розвитку 9 “Промисловість, інновації та інфраструктура”, а саме п.9.3 “Сприяти

прискореному розвитку високо- та середньовисокотехнологічних секторів переробної промисловості, які формуються на основі використання ланцюгів «освіта – наука – виробництво» та кластерного підходу за напрямками: розвиток інноваційної екосистеми; розвиток інформаційно-телекомунікаційних технологій (ІКТ); застосування ІКТ в АПК, енергетиці, транспорті та промисловості; високотехнологічне машинобудування; створення нових матеріалів; розвиток фармацевтичної та біоінженерної галузей”.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Fomin O. Development of an automated system for the technological processes of a "concrete plant" (a company for the production of construction components) // Sustainable smart cities and communities: business and innovation solutions 2025: Proceedings of the I International Conference (SSC&C2025), Kharkiv, April 21, 2025. – Kharkiv: Kharkiv National University of Radio Electronics, 2025. – P. 11–12. Jones A., Smith, B. Mastering HTTP: From Fundamentals to Advanced. Apress, 2018. 320 p.
2. Reichenbach, S., & Kromoser, B. (2021). State of practice of automation in precast concrete production. *Journal of Building Engineering*, 43, 102527.
3. Flanagan D. JavaScript: The Definitive Guide. O'Reilly Media, USA, 7th edition, 2020. 600 p.
4. ДСТУ 3008: 2015. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання / Нац. стандарт України. – Вид. офіц. – [Чинний від 2017 – 07 – 01]. – Київ: ДП «УкрНДНЦ», 2016. – 26 с.
5. Методичні вказівки з підготовки та захисту кваліфікаційної роботи здобувачами другого (магістерського) рівня вищої освіти спеціальності 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка, освітньо-професійних програм: «Комп'ютерно-інтегровані технологічні процеси і виробництва», «Комп'ютеризовані та робототехнічні системи» / Упоряд. І. Ш. Невлюдов, Р. В. Артюх, В. В. Безкоровайний, Н. П. Демська, В. В. Євсєєв, О. І. Филипенко, О. М. Цимбал. Харків: ХНУРЕ, 2024. 57 с.
6. Основи наукових досліджень : підручник / І. Ш. Невлюдов, Ю. М. Олександров, А. О. Андрусевич, О. О. Чала ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – Prague : OKTAN PRINT, 2024. – 468 с. DOI <https://doi.org/10.46489/ONDNP> Режим доступу на ресурсі бібліотеки ХНУРЕ <https://openarchive.nure.ua/handle/document/28574>
7. Невлюдов І. Ш. Техніко-економічне обґрунтування інженерних

рішень в інтелектуальному виробництві : підручник / І. Ш. Невлюдов. - Кривий Ріг: Чернявський Д. О., 2024. – 388 с. : іл. Режим доступу на ресурсі бібліотеки ХНУРЕ <https://openarchive.nure.ua/handle/document/27408>

8. Molina-Ríos, J., & Pedreira-Souto, N. (2020). Comparison of development methodologies in web applications. *Information and Software Technology*, 119, 106238.

9. Wu, C., Shang, G., Wu, L., & Zhao, M. (2021, March). Design of Control System for Concrete Mixing Plant. In *2021 4th International Conference on Electron Device and Mechanical Engineering (ICEDME)* (pp. 319-324). IEEE.

10. Chugcho-Barroso, D., Villacres, B. D., Villacres, E. F., Bustos-Pulluquitin, S., Garcia, C. A., & Garcia, M. V. (2025). Native Implementation of IEC-61499 Standard in Siemens Industrial PCs. In *AI and Computing in Industrial Education Handbook: A Multidisciplinary Analysis from Early to Higher Education and Specialized Sectors* (pp. 743-760). Cham: Springer Nature Switzerland.

11. Idrees, S., Nugraha, J. A., Tahir, S., Choi, K., Choi, J., Ryu, D. H., & Kim, J. H. (2024). Automatic concrete slump prediction of concrete batching plant by deep learning. *Developments in the Built Environment*, 18, 100474.

12. Rawat, B., & Purnama, S. (2021). Mysql database management system (dbms) on ftp site lapan bandung. *International Journal of Cyber and IT Service Management*, 1(2), 173-179.

13. Arman, M. (2025). Rancang Bangun Web Server Blog Dengan Layanan Vps Dan Navigasi. *BETRIK*, 16(01), 59-73.

14. Al Hajj, J., & Sah, M. (2023, November). Assessing the impact of ChatGPT in a PHP programming course. In *2023 7th International Symposium on Innovative Approaches in Smart Technologies (ISAS)* (pp. 1-10). IEEE.

15. Aftab, Z., Iqbal, W., Almustafa, K. M., Bukhari, F., & Abdullah, M. (2020). Automatic NoSQL to relational database transformation with dynamic schema mapping. *Scientific programming*, 2020(1), 8813350.

16. Ahmad, D. K., Ahmad, M. F., Ahmad, M. N., & Ahmad, A. S. (2020). An experiment of animation development in hypertext preprocessor (PHP) and

hypertext markup language (HTML). *Int. J. Sci. Res. in Computer Science and Engineering* Vol, 8(2).

17. Sotnik S. Overview: PHP and MySQL Features for Creating Modern Web Projects / S. Sotnik, V. Manakov, V. Lyashenko // *International Journal of Academic Information Systems Research (IJASIR)*. – 2023. – Vol. 7(1). – P. 11-17.

18. Padova, T. (2022). *Adobe Photoshop Elements Advanced Editing Techniques and Tricks: The Essential Guide to Going Beyond Guided Edits*. New Riders.

19. Husni, N. L., Sari, P. A. R., Dewi, T., Handayani, A. S., Sartika, D., & Mirza, A. (2022, February). Visual Studio Code for Activity Monitoring Interface. In *5th FIRST T1 T2 2021 International Conference (FIRST-T1-T2 2021)* (pp. 380-386). Atlantis Press.

20. Staiano, F. (2023). *Designing and Prototyping Interfaces with Figma: Elevate your design craft with UX/UI principles and create interactive prototypes*. Packt Publishing Ltd.

21. Huang, T. (2024). FEAD: Figma-Enhanced App Design Framework for Improving UI/UX in Educational App Development. *arXiv preprint arXiv:2412.06793*.

22. Irawan, A. J., Tobing, F. A. T., & Surbakti, E. E. (2021, October). Implementation of gamification Octalysis method at design and build a react native framework learning application. In *2021 6th International Conference on New Media Studies (CONMEDIA)* (pp. 118-123). IEEE.

23. Suri, B., Taneja, S., Bhanot, I., Sharma, H., & Raj, A. (2022, December). Cross-platform empirical analysis of mobile application development frameworks: Kotlin, react native and flutter. In *Proceedings of the 4th International Conference on Information Management & Machine Intelligence* (pp. 1-6).

24. Zahra, H. A., & Zein, S. (2022, October). A systematic comparison between flutter and react native from automation testing perspective. In *2022 International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)* (pp. 6-12). IEEE.

25. Krause, J. (2020). Introduction to bootstrap. In *Introducing Bootstrap 4: Create Powerful Web Applications Using Bootstrap 4.5* (pp. 1-17). Berkeley, CA: Apress.
26. Huang, K., & Qiu, X. (2022, December). Bootstrapping library-based synthesis. In *International Static Analysis Symposium* (pp. 272-298). Cham: Springer Nature Switzerland.
27. Hunkeler U., Truong H.-L., Stanford-Clark A. MQTT-S—A Publish/Subscribe Protocol for Wireless Sensor Networks. In: *3rd International Conference on Communication Systems Software and Middleware (COMSWARE)*, 2008. – P. 791–798.
28. Longo, E., Redondi, A. E. C., Cesana, M., & Manzoni, P. (2022). BORDER: A benchmarking framework for distributed MQTT brokers. *IEEE Internet of Things Journal*, 9(18), 17728-17740.
29. Staglianò, L., Longo, E., & Redondi, A. E. (2021, August). D-MQTT: design and implementation of a pub/sub broker for distributed environments. In *2021 IEEE International Conference on Omni-Layer Intelligent Systems (COINS)* (pp. 1-6). IEEE.
30. Shukla, A. (2023). Modern javascript frameworks and javascript's future as a full-stack programming language. *Journal of Artificial Intelligence & Cloud Computing*, 144(2), 2-5.
31. Ravshanov, A. (2024). Data Types in Javascript Programming Language. Introduction of new innovative technologies in education of pedagogy and psychology, 1(3), 143-150.
32. Ramadhan, R. F., & Mukhaiyar, R. (2020). Penggunaan Database Mysql dengan Interface PhpMyAdmin sebagai Pengontrolan Smarthome Berbasis Raspberry Pi. *JTEIN: Jurnal Teknik Elektro Indonesia*, 1(2), 129-134.
33. Rawat, B., & Purnama, S. (2021). Mysql database management system (dbms) on ftp site lapan bandung. *International Journal of Cyber and IT Service Management*, 1(2), 173-179.

34. Bochkovskyi, A. P., & Sapozhnikova, N. Y. (2021). Development of system of automated occupational health and safety management in enterprises. *Journal of Achievements in Materials and Manufacturing Engineering*, 107(1).