

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління  
(повна назва)

Кафедра електронних обчислювальних машин  
(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА

### Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Функціональна модель інформаційної системи моніторингу  
процесів на елеваторному комплексі

(тема)

Виконав:

студент 2 курсу, групи СПм 21-1  
Скачков І. В.  
(прізвище, ініціали)

Спеціальність 123 «Комп'ютерна інженерія»  
(код і повна назва спеціальності)

Тип програми освітньо-професійна  
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування  
(повна назва освітньої програми)

Керівник: ст.викл. Піскаръов О.М.  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ

(підпис)

Коваленко А.А.

(прізвище, ініціали)

2022 р.

Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ комп'ютерної інженерії та управління \_\_\_\_\_

Кафедра \_\_\_\_\_ електронних обчислювальних машин \_\_\_\_\_

Рівень вищої освіти \_\_\_\_\_ другий (магістерський) \_\_\_\_\_

Спеціальність \_\_\_\_\_ 123 «Комп'ютерна інженерія» \_\_\_\_\_  
(код і повна назва)

Тип програми \_\_\_\_\_ освітньо-професійна \_\_\_\_\_  
(освітньо-професійна або освітньо-наукова)

Освітня програма \_\_\_\_\_ Системне програмування \_\_\_\_\_  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

“ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ р.

## ЗАВДАННЯ

### НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту \_\_\_\_\_ Скачкову Ігорю Вячеславовичу \_\_\_\_\_  
(прізвище, ім'я, по батькові)

1. Тема роботи \_\_\_\_\_ Функціональна модель інформаційної системи моніторингу процесів на елеваторному комплексі \_\_\_\_\_

затверджена наказом по університету від “ 07 ” листопада 2022 р. № 1454 Ст \_\_\_\_\_

2. Термін подання студентом роботи до екзаменаційної комісії \_\_\_\_\_ 13 грудня 2022 р. \_\_\_\_\_

3. Вхідні дані до роботи \_\_\_\_\_

1) документація інструментів HTML, CSS, PHP \_\_\_\_\_

2) документація мови програмування JavaScript \_\_\_\_\_

3) інформація про архітектури розробки \_\_\_\_\_

4) перелік використаних програмних засобів \_\_\_\_\_

4. Перелік питань, що потрібно опрацювати у роботі \_\_\_\_\_

1. Аналіз предметної області \_\_\_\_\_

2. Огляд і вибір технологій та інструментів розробки \_\_\_\_\_

3. Програмна реалізація \_\_\_\_\_

4. Висновки \_\_\_\_\_

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Слайд-презентація – 11 слайдів

---

---

---

---

---

---

---

---

---

---

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1 )

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

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної області та існуючих	8.11.22-16.11.22	
2	Розробка функціональної моделі	17.11.22-27.11.22	
3	Аналіз технологій реалізації моделі	28.11.22	
4	Обґрунтування комплексу технічних засобів	29.11-01.12	
5	Огляд сучасних середовищ розробки та технологій для створення програмних модулів	02.12-03.12	
6	Тестування та оцінка надійності функціонування програмних рішень	04.12	
7	Подання кваліфікаційної роботи керівникові та її попередній захист	5.12.22-9.12.22	
8	Подання кваліфікаційної роботи на рецензування	10.12.22-11.12.22	

Дата видачі завдання 07 листопада 2022 р

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_  
(підпис)

ст.викл. Піскарьов О.М.  
(посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 83 сторінок, 25  
рисуноків, 2 додатка, 21джерел.

СЕРВЕР, МОНІТОРИНГ, СИСТЕМА МОНІТОРИНГУ, ІНТЕРНЕТ, ВЕБ  
ДОДАТОК, ІНФОРМАЦІЙНА БЕЗПЕКА, МЕРЕЖА, МОДЕЛЬ, ПРОЦЕСИ.

Метою кваліфікаційної роботи є розробка та дослідження моделі інформаційної системи моніторингу на елеваторному комплексі, визначення переваг та недоліків таких систем, а також вдосконалення існуючих систем за рахунок додавання сучасних елементів інформаційного захисту та розробки програмних модулів.

У ході виконання кваліфікаційної роботи було проведено аналітичний огляд існуючих рішень інформаційних систем, їх масштабування, актуальність розробки. Розроблено оригінальну інформаційну модель системи моніторингу, з застосування серверних технологій й засобів та інструментів Front-End розробки. Розглянуто функціональні вимоги до створення таких систем, а також здійснено вибір та обґрунтування технічних засобів реалізації функціональної моделі.

Виконано програмну реалізацію окремих елементів функціональної моделі з розробкою структури серверної частини, алгоритмів вирішення завдань. Здійснено тестування на оцінка надійності розробленої системи.

## ABSTRACT

Master's thesis: 83 pages, 25 figures, 2 appendices, 21 sources.

SERVER, MONITORING, MONITORING SYSTEM, INTERNET, WEB APPLICATION, INFORMATION SECURITY, NETWORK, MODEL, PROCESSES.

The purpose of the qualification work is to develop and research a model of the information system of monitoring at the elevator complex, to determine the advantages and disadvantages of such systems, as well as to improve existing systems by adding modern elements of information protection and developing software modules.

In the course of the qualification work, an analytical review of the existing solutions of information systems, their scaling, and the relevance of the development was carried out. An original information model of the monitoring system has been developed, using server technologies and tools and Front-End development tools. The functional requirements for the creation of such systems were considered, as well as the selection and justification of the technical means of implementing the functional model.

The software implementation of individual elements of the functional model with the development of the structure of the server part, algorithms for solving tasks was carried out. Testing was carried out to assess the reliability of the developed system.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	7
ВСТУП.....	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	12
1.1 Аналіз предметної області та визначення основних понять .....	12
1.2 Актуальність розробки.....	15
1.3 Масштабування проекту.....	16
1.4 Огляд існуючих аналогів.....	17
2 ОГЛЯД ВИКОРИСТАНИХ ТЕХНОЛОГІЙ.....	21
2.1 Функціональна модель системи.....	21
2.2 Технології розробки веб-додатків.....	29
2.3 Засоби Front-end розробки.....	34
2.4 Інструменти розробки Back-end.....	42
2.5 Функціональні вимоги.....	44
2.6 Вибір і обґрунтування комплексу технічних засобів.....	49
3 ПРОГРАМНА РЕАЛІЗАЦІЯ.....	52
3.1 Структура системи.....	52
3.2 Розробка, обґрунтування та оцінка алгоритмів вирішення постановленої задачі.....	53
3.3 Реалізація додатку.....	55
3.4 Тестування та оцінка надійності функціонування програмних рішень.....	60
ВИСНОВКИ.....	69
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	70
ДОДАТОК А.....	72
ДОДАТОК Б.....	79

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БД – база даних

СУБД – система управління базами даних

ПЗ – програмне забезпечення

CSS – Cascade Stylesheet

HTML – Hypertext Markup Language

HTTP – HyperText Transfer Protocol

ICT – Information and communications technology

JS – JavaScript

JSON – JavaScript Object Notation

PDF – Portable Document Format

PHP – Hypertext Preprocessor

SQL – Structured Query Language

URI – Uniform Resource Identifier

URL – Uniform Resource Locator

XML – eXtensible Markup Language

MVC – Model View Controller

DOM – Document object model

## ВСТУП

До відомих видів ресурсів будь-якої організації – матеріальних, трудових, енергетичних, фінансових – додався новий, що раніше не враховується – інформаційний. Інформація дозволяє орієнтуватися в загальній обстановці на підприємстві, зменшити фінансовий ризик, стежити за довкіллям, ринком та змінами ринкової кон'юнктури, оцінювати свою діяльність, виробляти та коригувати стратегію підприємства, приймати якісні управлінські рішення.

Сьогодні не часто якась компанія не використовує технології більшою чи меншою мірою. У багатьох випадках належне функціонування обладнання, мереж та систем має вирішальне значення для підтримки працездатності бізнесу. Ще частіше клієнту пропонується пряме технологічне рішення.

Але навіть якщо технологія незамінна для роботи компанії, це не означає, що вона незламна. Збої можуть статися будь-коли, приводячи часом до критичних ситуацій. Тому в будь-якій компанії, де важлива ІТ-інфраструктура, необхідно стежити за її правильним функціонуванням, щоб можлива помилка не вплинула на послуги, що надаються користувачам.

Для виявлення та запобігання збоям дуже зручно мати гарний інструмент моніторингу. Системи моніторингу контролюють технології, що використовуються компанією (процеси, обладнання, мережі та комунікації, операційні системи або програми, тощо), для аналізу їх роботи та продуктивності, а також для виявлення та попередження про можливі помилки. Хороша система моніторингу здатна контролювати пристрої, інфраструктуру, програми, послуги і навіть бізнес-процеси.

Успішна діяльність підприємства у сучасних умовах неможлива без інформаційного забезпечення з питань, які стосуються безпосередньо до діяльності підприємства, і до загальної економічної та політичної ситуації у країні.

Під інформаційним забезпеченням розуміють:

1) Сукупність процесів з підготовки та доведення до споживачів інформації для вирішення управлінських, виробничо-технічних завдань відповідно до етапів їх виконання.

2) Сукупність єдиної системи класифікації та кодування інформації та єдиної системи документації, сукупність методів та засобів організації машинних масивів інформації.

Дослідження показують, що від 50 до 90% робочого часу сучасний керівник витрачає обмін інформацією, що відбувається у процесі нарад, зборів, зустрічей, розмов, переговорів, прийому відвідувачів, складання та читання різних документів тощо. І це – життєва необхідність, оскільки володіння інформацією означає володіння реальною владою, тому особи, причетні до неї, часто прагнуть її приховувати, щоб згодом на ній спекулювати – адже брак інформації дезорієнтує, тому завжди необхідно вміти відокремити потрібну інформацію від непотрібної, корисної від марної.

Передача інформації про стан та діяльність організації на вищий рівень управління та взаємний обмін інформацією між усіма взаємопов'язаними підрозділами організації здійснюється, як правило, на базі сучасної електронно-обчислювальної техніки та інших технічних засобів зв'язку.

Неабиякою мірою успіх будь-якої справи залежить від того обсягу інформації, якою володіє особа, яка приймає рішення. Генрі Форд говорив: «Основа успіху у бізнесі – своєчасно отримана, правильно зрозуміла та вміло використана інформація».

Автоматизація та створення інформаційних систем моніторингу є на даний момент однією з найбільш ресурсоемних сфер діяльності техногенного суспільства. Однією з причин активного розвитку даної галузі є те, що автоматизація є основою докорінної зміни процесів управління, які відіграють важливу роль у діяльності людини та суспільства. Виникають системи управління, дія яких спрямована на підтримку або покращення роботи об'єкта за допомогою пристрою управління (комплекс засобів збору,

обробки, передачі інформації та формування сигналів, що управляють, або команд).

Інформаційна система - це система, що забезпечує уповноважений персонал даними чи інформацією, що стосуються організації. Інформаційна система управління, загалом, складається з чотирьох підсистем: системи обробки транзакцій, системи управлінських звітів, офісної інформаційної системи та системи підтримки прийняття рішень, включаючи інформаційну систему керівника, експертну систему та штучний інтелект.

Автоматизована інформаційна система моніторингу - взаємопов'язана сукупність засобів, методів та персоналу, що використовуються для зберігання, обробки та видачі інформації на користь досягнення поставленої мети.

Таким чином, автоматизована інформаційна система моніторингу є сукупністю інформації, економіко-математичних методів та моделей, технічних, програмних, технологічних засобів та фахівців, призначена для обробки інформації та прийняття управлінських рішень.

Сучасні ІТ дозволяють:

- перетворити підприємство в інформаційно-кероване. Тобто стає можливим керувати підприємством спираючись на інформаційний ресурс, який, на відміну від інших (Вартість, Час, Оперативність реакції, Гнучкість. Якість, Інновація) можуть бути використані багаторазово;

- сприймати підприємство як одне ціле. Тобто, якщо компанія складається з безлічі підприємств, що ведуть бізнес у різних сферах, або розташовані віддалено один від одного, керівництво може ефективно керувати ними як одним цілим, не турбуючись про сумісність додатків у тих чи інших підрозділах. Також з'являється можливість об'єднання інформаційних підсистем в одну, усуваючи при цьому дублювання процесів;

- керувати підприємством в режимі реального часу. Найбільша цінність - актуальна інформація. ІТ дозволяють дати моментальний доступ до неї всім учасникам процесів. Результатом є збільшення ефективності та

пропускної спроможності каналів інформації, та можливість здійснювати процеси не тільки послідовно, а й паралельно;

- ставати основою для бізнес-стратегії підприємства. ІТ свого часу давали і дають можливість швидко проводити стратегічні зміни для підприємства, полегшуючи впровадження нових систем;

- використовуючи одну програмну платформу, працювати з урахуванням усіх особливостей конкретного підприємства;

- орієнтуватися на масових користувачів. Усі користувачі, яким це необхідно, можуть бути включені до єдиної ІС підприємства. При цьому система забезпечує максимально можливий «дружелюбний» інтерфейс, допомагаючи людям виконувати свою роботу, а не заважаючи.

Відповідно до вищевикладеного, автоматизація систем моніторингу є актуальним та перспективним процесом. Об'єктом дослідження у цій роботі виступають особливості інформаційного забезпечення прийняття оперативних управлінських рішень, що виникають в аспекті цієї теми на елеваторному комплексі.

Предметом цього дослідження є методи та засоби підвищення ефективності управління виробничим процесом.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Аналіз предметної області та визначення основних понять

У ході виконання дипломної роботи необхідно спроектувати та розробити систему моніторингу виробничої діяльності підприємства. Це необхідно для вдосконалення інформаційного обміну на підприємстві, яке дозволить скоротити час при прийнятті оперативних рішень, необхідних для координації виробничого процесу у реальному режимі часу з урахуванням отриманих вчасно необхідних даних. Відсутність даного рішення здатна викликати збій у системі роботи підприємства, що спричинить небажані та непередбачені зміни, при виконанні розроблених планів. За наявності будь-яких відхилень у роботі всього процесу виробництва, починаючи з поставки матеріалів, закінчуючи відвантаженням продукції, необхідно знати повною мірою всю інформацію та мати можливість прийняття управлінських рішень конкретному керівнику для коригування затверджених планів, що неможливо у відсутності інформації та перегляду загальної картини в цілому.. Найголовніше, що при виникненні будь-яких форс-мажорних ситуацій з'являється поспіх, необдумані вчинки, які можуть призвести до різних небажаних наслідків.

Існує безліч способів зберігання та передачі інформації, але вся їх якість, доступність і надійність знаходиться на досить низькому рівні.

Порівняння засобів зберігання інформації без використання інформаційних систем наведено в таблиці. 1.1.

Найефективнішим способом організації роботи виробництва та її працівників є автоматизована система моніторингу виробничої діяльності підприємства. При використанні такої системи всі виробничі ділянки працюють як єдине ціле, в якому всі учасники мають права доступу та дій, а керівництво - повний цілодобовий доступ до управління.

Таблиця 1.1 - Порівняння засобів зберігання інформації

	Доступність даних:	Резервне копіювання:	Можливість втрати даних:	Можливість автоматизації:
Записна книжка/Щоденник	Ні	Ні	Так	Ні
В паперових папках	Так	Ні	Так	Ні
Графік роботи на А4	Так	Ні	Так	Ні
Excel	Ні	Так	Так	Ні

Автоматизація виробничих процесів – найважливіша складова автоматизації управління виробництвом. Автоматизація процесів на виробництві дає такі переваги:

- усі виробничі процеси стають упорядкованими та оперативними;
- контроль, аналіз та планування процесів стають простішими;
- знижується роль людського фактору;
- усі процеси інтегруються в єдину систему.

Автоматизація управління виробництвом на підприємстві надає можливість швидко на будь-якому етапі проводити аналіз діяльності, гнучко керувати процесами та оперативно змінювати ситуацію; рядовим працівникам працювати найефективніше у сфері виробництва, постачання, збуту.

Автоматизація – це швидкість та ефективність [1].

Ефективність виробничої діяльності значно збільшується за рахунок впровадження на підприємстві комплексної автоматизації процесів роботи та управління підприємством.

Автоматизація виробництва може дозволити контролювати виробничий процес на кожному етапі, виявляти та усувати «вузькі місця» в

усіх напрямках:

- контроль за прийомом товару;
- керування лабораторією;
- керування складами ;
- керування замовленнями (експортом).

Моніторинг або моніторинг (укр. спостереження) – система постійного спостереження за явищами і процесами, що проходять в навколишньому середовищі, суспільстві, результати якого слугують для обґрунтування управлінських рішень по забезпеченню безпеки людей та об'єктів/суб'єктів (наприклад, економіки).[2] При автоматичному контролі відбувається отримання і обробка інформації про стан об'єкта/суб'єкта та зовнішніх умов для виявлення подій, що визначають управлінські дії. Подією може бути будь-який якісний результат: поява деталі з розмірами, що виходять за допустимі межі, коротке замикання, вихід температури за встановлене значення, аварія обладнання та інші.

Елеватор – зерносховище, обладнане пристосуваннями для підймання, сушіння, зберігання і т. ін. великої кількості зерна.

Веб-розробка – процес програмування веб-проектів і додатків. Це дизайн та верстання сторінок, робота з клієнтською та серверною частиною, конфігурація серверів.

Верстання веб-сторінок – це процес створення веб-сторінки із попередньо створеного макету дизайну сайту, заздалегідь намальованого за допомогою графічних редакторів.

Верстка веб-сторінки – це структура вже створеної сторінки, клієнта і сервера, а також конфігурування веб-сервера.

Веб-дизайн – це процес виробництва сайтів, який складається з технічної розробки, створення зручної структури веб-сторінки, графічного оформлення і передачі інформації в мережу.

## 1.2 Актуальність розробки.

Актуальність створення наведеного додатку на сьогоднішній день досить висока, на це є багато причин та приводів. Можна розглянути багато переваг з різних аспектів, областей і розділити їх на групи:

- відновлення інфраструктури/підприємств;
- соціальна;
- економічна.

### Відновлення інфраструктури/підприємств

На плечі України впав важкий тягар. Спочатку 20 року по всьому світу пройшлася пандемія COVID-19, яка і триває до цього дня, хоч і не так сильно як раніше, поширившись вона забрала як життя, так і торкнулася фінансову структуру. Безліч підприємств закрилися на якийсь час, а деякі з них назавсім. Але на цьому негативні сторони не закінчилися, на початку 2022 року Україна зазнала відкритого військового вторгнення з боку Росії, внаслідок чого все повторилося знову, тільки в набагато більших масштабах. Але у всіх негативних моментів має бути позитивна сторона, розглядаючи тільки фінансову та бізнес-аналітичну складову можна прийти до всім зрозумілого висновку. Після закінчення всіх цих подій почнуть з'являтися нові організації, підприємства, тощо, які будуть захоплювати різноманітні втрачені частини ринку, будуть намагатися розвиватися та масштабуватися, внаслідок чого вони будуть потребувати використовувати новітні технології, до яких входять і системи моніторингу, для свого росту.

З попереднього пункту можна підкреслити те, що попит повинен бути, а як всім відомо “попит породжує пропозицію”, а не навпаки. Та якщо відкинути сказане раніше, то кожен день з'являються нові організації, підприємства, тощо, які потенційно будуть потребувати вирішення своїх проблем. Також не слід забувати ще о рості технологій, які кожен день роблять шаг вперед та о тих місцях, які ще не знаходяться на так мовити “попередньому етапі технологій”.

Соціальна сторона теж має місце бути тут, хоча можливо це і дуже грубо назвати "соціальним" тому що те, що віднесено сюди дуже різне. Наприклад відеоспостереження за співробітниками так і для запобігання різноманітним злочинним діям важко назвати соціальними, але якоюсь мірою, якщо поглянути на рівень глибше можна назвати їх такими. Різний моніторинг персоналу також у теорії можна віднести до цієї категорії тому, що хоч це і не пряма взаємодія двох і більше людей, але одна людина чи група осіб отримує інформацію також про одну людину чи групу людей, яка тим чи іншим чином впливає на їх вирішення, думки тощо.

### 1.3 Масштабування проекту

Будь-яка компанія, організація чи підприємство з часом буде мати бажання продовжувати розширюватися та рости. Іноді це буває доволі складно. В процесі зростання доходів, буде зростати кількість робітників, кількість матеріалів (бізнес паперів) з якими буде доводитися працювати та ін. Шляхи вирішення проблем в даній ситуації доволі легкі, тому що масштабування з цього проекту може бути майже нескінченне.

Масштабування зі сторони "hardware", та підключення усіх "hardware" елементів до (розробленого) частини "software" проекту, а саме:

- завантаження/відвантаження- електронні ваги з підключенням до мережі, камера для аналізу номерів машин;
- сховище- підключення датчиків температур, вологи, заповненості сховища тощо;
- встановлення електронної системи пропуску робітників.

Робота з базами даних. При збільшенні масштабів з часом з'являються проблеми з роботою над даними. Будь це їх постійна доступність, їх кількість тощо. Вирішенням цих проблем є наступні дії:

- перехід на хмарні сервіси, які забезпечують більшу надійність даних;
- використовувати реплікування даних для сто відсоткової

безперебійної роботи та збільшенню надійності даних.

#### 1.4 Огляд існуючих аналогів

В ситуації з елеваторами та системами моніторингу складається доволі складно, бо в відкритому доступі аналогів нема, всі аналоги які існують та доступні на ринку продаються та налагоджуються під готовий проект.

KMZ INDUSTRIES – провідний постачальник комплексних рішень для зберігання та обробки зерна[3].

Перше та найбільше в Україні серійне виробництво елеваторного обладнання.

Проектування, монтаж та сервіс – за європейськими стандартами.

Автоматизація обладнання та технологічних процесів на елеваторах.

Інтеграція розрізнених робочих ділянок зерносховища (вагова, лабораторія, сушіння, очищення, зберігання, відвантаження) в єдину систему автоматизованого управління елеватором.

Вони позиціонують себе як «ЄДИНЕ В УКРАЇНІ ВИРОБНИЦТВО ПОВНОГО КОМПЛЕКСУ ЕЛЕВАТОРНОГО ОБЛАДНАННЯ».

Виробничі потужності компанії знаходяться в м. Карлівка Полтавської області. Виробництво оснащене обладнанням європейських, японських і американських виробників, має власне конструкторське бюро, монтажний підрозділ, сервісну службу та відділ автоматизації елеваторів.

У 2012 році KMZ Industries придбав англійську компанію Brice-Baker - одного з провідних європейських виробників обладнання для зерносховищ.

29 квітня 2021 року KMZ Industries та Variant Agro Build підписали угоду про злиття елеваторного напрямку ТОВ «Варіант Агро Буд» із ПрАТ «Карлівський машинобудівний завод».

Вони надають повний спектр услуг, можна сказати «повна автоматизація» починаючи з шаф енергопостачання, завершуючи встановленням власного програмного забезпечення та апаратної частини.

З інформації про її рішення відомо тільки із тексту та деяких візуальних моментів.

РІШЕННЯ KMZ INDUSTRIES включає:

- автоматизацію елеваторного обладнання (апаратна частина і ПЗ);
- моніторинг термометрії;
- контроль і управління ваговим комплексом;
- пробовідбірник і лабораторію;
- бухгалтерський і технічний облік.

ІТ-рішення, які візуалізують і об'єднують ці процеси в єдину систему управління зображено на рисунку 1.1.

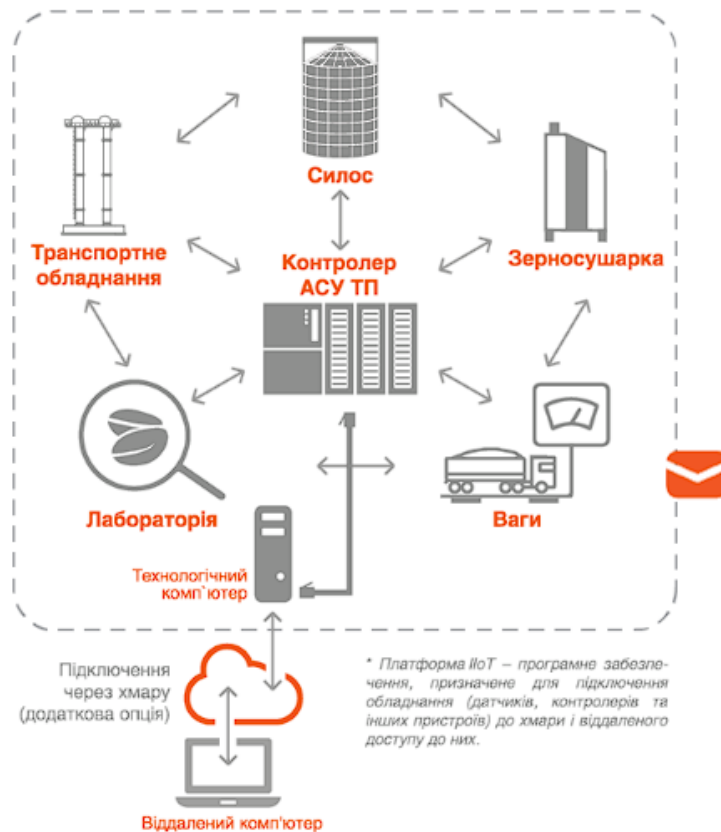


Рисунок 1.1 - Візуалізація процесів рішення компанії KMZ INDUSTRIES

Програмне рішення організоване на платформі PLCnext Technology – німецького електротехнічного гіганта Phoenix Contact.

Особливості ПОТ-рішення наведені на рисунку 1.2.



Рисунок 1.2 - Особливості ПОТ-рішення

Також по особливостям можна виділити:

- модульність. Загальна програмна платформа з можливістю поступового або вибіркового підключення модулів управління за бажанням замовника. Кожне наступне підключення нового обладнання не потребують окремого робочого місця для управління – все керується в одній програмі;
- універсальність. Готовий програмний комплекс адаптується до вимог конкретного елеватора. Це значно знижує витрати під час пускових-налагоджувальних робіт й розробці ПЗ. Об'єднання в одній програмі всіх етапів роботи елеватора: ваговий комплекс, відбір проб і лабораторія, очищення та сушіння зерна, транспортування в силос, контроль зберігання і відвантаження. Виключає потребу у великій кількості програм під кожен ділянку елеватора;
- простота. Максимально простий інтерфейс програми зрозумілий навіть персоналу без спеціальної підготовки;
- можливість розширення. Контролер Phoenix Contact PLCnext може бути розширений до 63 модулів в одному них, що дозволяє заощадити на придбанні апаратної частини при збільшенні модулів, що підключаються до

рішення.

Відсутність прихованих витрат на ліцензії за кількість робочих місць.

Таким чином, на основі проведеного аналітичного огляду існуючих рішень інформаційних систем, їх масштабування, актуальності розробки необхідно:

- розробити оригінальну інформаційну модель системи моніторингу;
- розглянути питання застосування серверних технологій;
- розглянути засоби та інструменти Front-End розробки;
- розглянути функціональні вимоги до створення таких систем;
- здійснити вибір та обґрунтування технічних засобів реалізації функціональної моделі;
- виконати програмну реалізацію елементів функціональної моделі з розробкою структури серверної частини, алгоритмів вирішення завдань;
- здійснити тестування на оцінку надійності розробленої системи.

## 2 ОГЛЯД ВИКОРИСТАНИХ ТЕХНОЛОГІЙ

### 2.1 Функціональна модель системи

Функціональна модель призначена для вивчення особливостей роботи (функціонування) системи та її призначення у взаємозв'язку з внутрішніми та зовнішніми елементами.

Функція – найістотніша характеристика будь-якої системи, що відображає її призначення, те, заради чого вона була створена. Подібні моделі оперують насамперед із функціональними параметрами. Графічним уявленням цих моделей є блок-схеми. Вони відбивають порядок дій, вкладених у досягнення заданих цілей (т. зв. функціональна схема). Функціональною моделлю є абстрактна модель.

Більшість із тих, хто займається реалізацією проектів, пов'язаних із створенням чи розвитком корпоративних інформаційних систем, згодні з тезою, що замовнику потрібна інформаційна система, яка підвищує ефективність діяльності підприємства. Проте замовники та розробники інформаційних систем досі розмовляють різними мовами: вони по-різному розуміють – що означає підвищити ефективність підприємства.

Розробники інформаційних систем дуже часто під підвищенням ефективності розуміють зростання кількості робочих станцій у локальній обчислювальній мережі (ЛВС) підприємства, зростання пропускної спроможності ЛОМ, зростання кількості документів, обробка яких здійснюється на автоматизованих робочих місцях (АРМах) тощо.

Замовники під підвищенням ефективності підприємства розуміють зростання продуктивності праці, зменшення собівартості та підвищення якості вироблених продуктів та послуг. Останнім часом у життя замовників стрімко вриваються нові умови та нові поняття: стратегія розвитку, ключова компетенція, бізнес-архітектура, бізнес-правила та багато іншого.

Для того, щоб замовник та розробник інформаційної системи розуміли один одного, необхідно, щоб розробник переорієнтувався з вирішення технічних завдань щодо створення або розвитку інформаційної системи на вирішення комплексних завдань щодо підвищення ефективності діяльності підприємства замовника. При такому підході на перший план постає проблема ефективного способу вивчення сфери діяльності замовника:

- обстеження існуючої бізнес-архітектури, ділових процесів, бізнес-правил, інформаційних потоків;
- ідентифікація проблем, "вузьких" місць, які негативно впливають на ефективність діяльності підприємства;
- розробка та реалізація заходів щодо усунення наявних проблем та зміни бізнес-архітектури підприємства, розбудови ділових процесів;
- розробка конкретного проекту корпоративної інформаційної системи, реалізація цього проекту та супровід у майбутньому.

У рамках такого підходу підвищити ефективність роботи розробника інформаційних систем покликані інструментальні засоби, призначені для моделювання підприємства та реінжинірингу ділових процесів. Одним із представників такого сімейства інструментальних засобів є CASE-інструменти для функціонального моделювання ділових процесів.

IDEF0 – методологія функціонального моделювання.

У ході реалізації програми інтегрованої комп'ютеризації виробництва (ICAM), запропонованої свого часу ВПС для аерокосмічної промисловості США, було виявлено потребу у розробці методів аналізу взаємодії процесів у виробничих системах.[4] Для задоволення цієї потреби було розроблено методологію IDEF0 (Integrated Definition Function Modeling), яка в даний час прийнята як федеральний стандарт США. Методологія успішно застосовувалася в різних галузях, продемонструвавши себе як ефективний засіб аналізу, проектування та подання ділових процесів. Нині методологія IDEF0 широко застосовується у США, а й у світі. IDEF0 по роботі сайту наведена на рисунку 2.1.



Рисунок 2.1 – IDEF0 коректна робота сайту

У світі технологій, починаючи від маленьких дітей до молодих людей і починаючи від молодих до людей похилого віку, кожен використовує свої смартфони, ноутбуки, комп'ютери, КПК тощо для вирішення будь-яких простіших чи складніших завдань в Інтернеті за допомогою деяких програм, все виглядає дуже просто. до користувача. Крім того, це мета хорошого програмного забезпечення для забезпечення високої якості послуг у зручному для користувача середовищі. Там загальна абстракція будь-якого програмного продукту робить його простим і дуже легким для використання користувачем. Але назад, якщо ми побачимо, що створення складної програмної програми включає складні процеси, які складаються з кількох елементів, невеликою частиною яких є кодування.

Після збору бізнес-вимог бізнес-аналітиком, а потім команда розробників починає працювати над специфікацією вимог до програмного

забезпечення (SRS) , яка послідовно проходить різні етапи, такі як тестування, прийняття, розгортання, обслуговування тощо. Кожен процес розробки програмного забезпечення виконується за допомогою кількох послідовних кроків який відповідає цьому життєвому циклу розробки програмного забезпечення (SDLC).

На етапі проектування життєвого циклу розробки програмного забезпечення архітектура програмного забезпечення визначається та документується. Тож у цій статті ми чітко обговоримо один із важливих елементів життєвого циклу розробки програмного забезпечення (SDLC), тобто архітектуру програмного забезпечення.

Архітектура програмного забезпечення: архітектура програмного забезпечення визначає фундаментальну організацію системи та простіше визначає структуроване рішення. Він визначає, як компонуються компоненти програмної системи, їхній взаємозв'язок і зв'язок між ними. Він служить схемою програмного застосування та основою розробки для команди розробників.

- архітектура програмного забезпечення визначає перелік речей, що полегшує процес розробки програмного забезпечення;
- архітектура програмного забезпечення визначає структуру системи;
- архітектура програмного забезпечення визначає поведінку системи;
- архітектура програмного забезпечення визначає взаємозв'язок компонентів;
- архітектура програмного забезпечення визначає структуру зв'язку;
- архітектура програмного забезпечення врівноважує потреби зацікавлених сторін;
- архітектура програмного забезпечення впливає на структуру команди;
- архітектура програмного забезпечення зосереджена на важливих елементах;
- архітектура програмного забезпечення фіксує ранні дизайнерські

рішення.

Перед запуском проекту потрібно перш за все визначити архітектуру яка краще за все підходить до нього та залежить від вимог проекту та її поведінки. Для цього потрібно знати різноманітні шаблони архітектури та проектування.

З написаного вище можна зрозуміти, що архітектура для проєкту є дуже важливим компонентом. Вона може допомогти мінімізувати код, запобігти сильному зв'язуванню та допоможе стандартизувати написання коду, який забезпечуватиме рішення для ситуації, що повторюється при розробці протягом всього часу.

Для цього можна детально розглянути деякі з них: Model-View-Presenter (MVP) , Model-View-ViewModel (MVVM) та Model-View-Controller (MVC).

MVC (Model-View-Controller) – це шаблон у розробці програмного забезпечення, який зазвичай використовується для реалізації інтерфейсів користувача, даних і логіки керування[5]. Це підкреслює розмежування між бізнес-логікою програмного забезпечення та дисплеєм. Такий «розподіл завдань» забезпечує кращий розподіл праці та покращення технічного обслуговування. Деякі інші шаблони проектування засновані на MVC, наприклад MVVM (модель-вид-вид-модель), MVP (модель-вид-представник) і MVW (модель-вид-що завгодно).

Три частини шаблону проектування програмного забезпечення MVC можна описати наступним чином:

- модель: керує даними та бізнес-логікою;
- перегляд: керує макетом і відображенням;
- контролер: направляє команди до частин моделі та виду.

Приклад контролера представлення моделі

Простий додаток зі списком покупок. Усе, що потрібно, це список із назвами, кількістю та ціною кожного товару, який ми маємо купити цього тижня. Нижче ми опишемо, як ми можемо реалізувати деякі з цих функцій за

допомогою MVC зображено на рисунку 2.2.

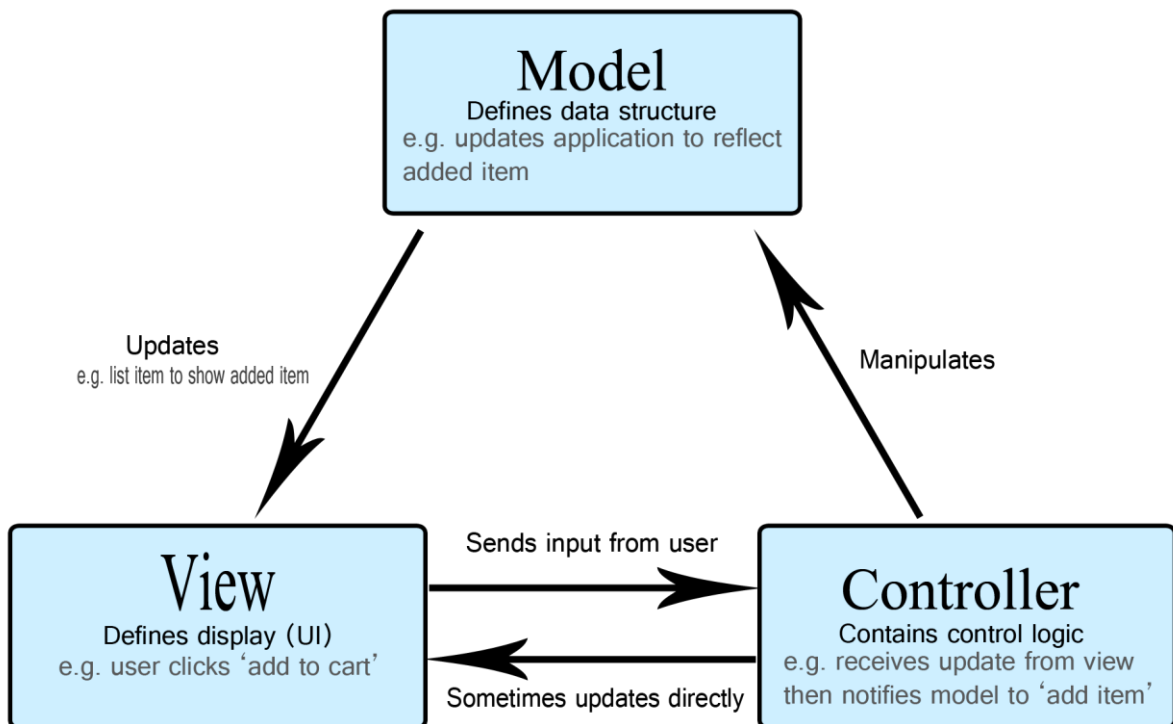


Рисунок 2.2– Реалізація функції за допомогою MVC

Модель визначає, які дані має містити додаток. Якщо стан цих даних змінюється, то модель зазвичай сповіщає про це перегляд (тому дисплей може змінюватися за потреби), а іноді й контролер (якщо потрібна інша логіка для керування оновленим видом).

Повертаючись до нашої програми зі списком покупок, модель вказує, які дані мають містити елементи списку – товар, ціна тощо – і які елементи списку вже присутні.

Він визначає, як мають відобразитися дані програми. У нашій програмі зі списком покупок представлення визначатиме, як список буде представлено користувачеві, і отримуватиме дані для відображення з моделі.

Контролер містить логіку, яка оновлює модель та/або представлення у відповідь на введення від користувачів програми.

Так, наприклад, наш список покупок може мати форми введення та кнопки, які дозволяють нам додавати або видаляти елементи. Ці дії вимагають оновлення моделі, тому вхідні дані надсилаються до контролера, який потім маніпулює моделлю належним чином, а потім надсилає оновлені дані до представлення.

Однак ви також можете просто оновити подання, щоб відобразити дані в іншому форматі, наприклад, змінити порядок елементів на алфавітний або від найнижчої до найвищої ціни. У цьому випадку контролер може впоратися з цим безпосередньо без необхідності оновлення моделі.

MVC в Інтернеті. Як веб-розробник, цей шаблон, ймовірно, буде досить знайомим, навіть якщо ви ніколи раніше його свідомо не використовували. Ваша модель даних, імовірно, міститься в якійсь базі даних (чи то традиційна база даних на стороні сервера, як-от MySQL, чи рішення на боці клієнта, як-от IndexedDB [en-US] .) Керуючий код вашої програми, ймовірно, написаний на HTML/JavaScript , і ваш інтерфейс користувача, ймовірно, написаний з використанням HTML/CSS/що завгодно. Звучить дуже схоже на MVC, але MVC робить ці компоненти більш жорсткими.

На початку існування Інтернету архітектура MVC була здебільшого реалізована на стороні сервера, коли клієнт запитував оновлення через форми або посилання та отримував оновлені перегляди для відображення у браузері. Однак у наші дні більше логіки передається клієнту з появою сховищ даних на стороні клієнта та XMLHttpRequest, що дозволяє частково оновлювати сторінку за потреби.

Такі веб-фреймворки, як AngularJS і Ember.js , реалізують архітектуру MVC, хоча й дещо по-різному.

Також існує Spring MVC – структура для створення слабо пов'язаних веб-додатків, що розділяє основні аспекти їхньої розробки: об'єкти, бізнес-логіку та зовнішній вигляд програми. Основна перевага архітектури MVC - можливість змінювати один із компонентів програми, суттєво не впливаючи на інші[6].

Dispatcher Servlet – серце Spring MVC, фронт-контролер, що повністю настраюється, координує всі дії з обробки запитів зображених на рисунку 2.3.

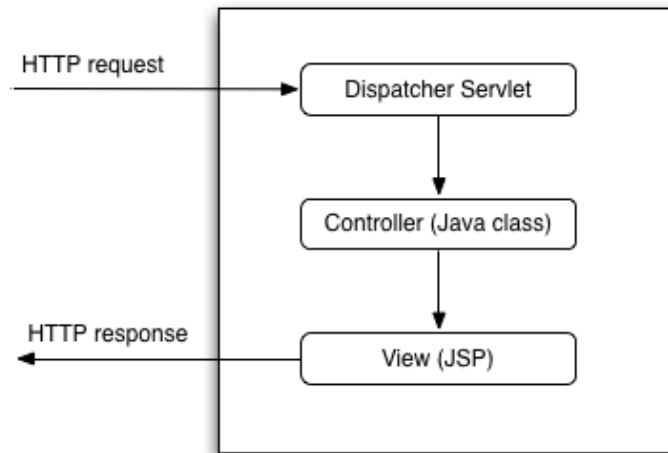


Рисунок 2.3 – Шаблон дизайну при розробці у Spring MVC

Це типовий шаблон дизайну при розробці веб-застосунків, що розподіляє запити користувача і керує роботою інших компонентів програми, відсилаючи команди кожному конкретному контролеру Spring MVC.

Якщо зробити приблизний шаблон MVC яка буде відповідати навеленому проекту, то вона повинна виглядати так як зображено на рисунку 2.4.

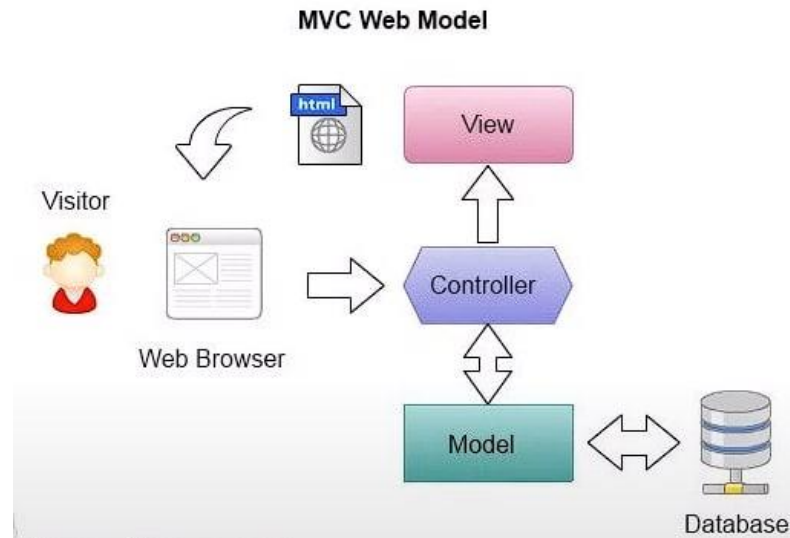


Рисунок 2.4 – Шаблон MVC проекту

## 2.2 Технології розробки веб-додатків

Web-розробка – процес створення веб-сайтів або веб-додатків. Основними етапами процесу є веб-дизайн, верстка сторінок, програмування для веб на стороні клієнта і сервера, а також конфігурування веб-сервера.

Розробка веб-додатків дозволяє веб-проектам працювати та діяти аналогічно до мобільного додатка. Веб-застосунки залучають користувачів, віддаючи пріоритет чуйним взаємодіям, але, як і раніше, доставляються через Інтернет по мережі.

Можна сміливо сказати, що розробка веб-додатків – це потомство веб-розробки. Хоча веб-розробка та розробка веб-застосунків мають спільні риси, розробка веб-додатків за визначенням більш динамічна, ніж може запропонувати традиційна веб-розробка[7].

Десять років тому веб-сторінки були переважно статичними, на них було розкидано кілька зображень та відео. У 2005 році завдяки Ажах стало можливим створювати якісніші, швидші та інтерактивні веб-додатки.

Web-App, також відомий як веб-додаток, є не що інше, як комп'ютерна програма, яка використовує веб-браузер для виконання певної функції. Веб-

додаток - це клієнт-серверна програма, що складається з клієнтської та серверної частин. Користувач вводить дані через клієнтську частину (зовнішній інтерфейс), а серверна частина (внутрішня частина) програми зберігає та обробляє інформацію. Наприклад, типовими веб-додатками є кошики для покупок, системи керування контентом та онлайн-форми.

Веб-застосунки зазвичай пишуться з використанням JavaScript, HTML5 або каскадних таблиць стилів (CSS) для клієнтського програмування (зовнішнього інтерфейсу). ASP.NET, PHP, Java та Ruby зазвичай використовуються для серверного програмування.

Доступ до веб-додатків здійснюється через мережу, і їх не потрібно завантажувати. Натомість користувачі можуть отримувати доступ до веб-застосунків через такі браузерери, як Google Chrome, Mozilla Firefox, Opera або Safari.

Веб-додаток складається з трьох компонентів: веб-сервера, сервера додатків та бази даних. Веб-сервер керує запитамі від клієнта, сервер додатків обробляє запити, а база даних зберігає інформацію. Загальна схема наведена на рисунку 2.5.

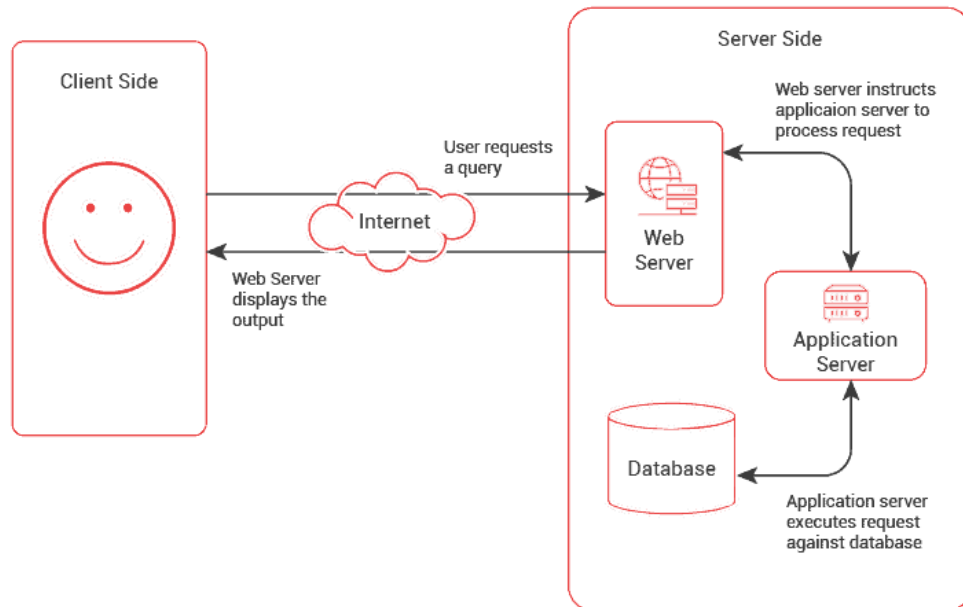


Рисунок 2.5 – Взаємодія сервера і клієнта та БД

Тепер, коли були розглянуті різні аспекти веб-застосунків, можна заглибитися в різні категорії веб-додатків. Веб-програми класифікуються на основі їх функціональних можливостей, інструментів та технологій[8].

Статичний веб-додаток не передбачає жодної взаємодії між користувачем та сервером. Він безпосередньо відображає вміст у браузері кінцевого користувача без вилучення будь-яких даних із боку сервера. Статичні веб-програми створюються за допомогою простих HTML, CSS та JavaScript для відображення відповідного контенту. У деяких випадках для залучення відвідувачів також використовуються GIF-файли, відео та анімація. Статичні веб-програми досить прості і зручні в управлінні.

Веб-сайт компанії або сайт портфоліо людини є прикладами статичних веб-додатків.

Веб-додаток, який взаємодіє з клієнтом і генерує дані в реальному часі на основі запитів користувача, називається динамічним веб-додатком. Він включає різні інтерактивні елементи і функції для залучення відвідувача. Динамічні веб-додатки значно складніші та складніші на технічному рівні.

Для створення динамічних веб-застосунків використовується широкий спектр мов програмування, найбільш поширеними з яких є PHP і ASP.NET.

Прикладом динамічного веб-програми є Facebook, де ви можете легко увійти в систему і безперешкодно спілкуватися зі своїми друзями.

Веб-додаток у вигляді магазину або магазину, який просуває купівлю або продаж чогось в Інтернеті, класифікується як електронна комерція. Ця веб-програма вимагає деяких основних функцій, таких як інтеграція електронних платежів, інтеграція транзакцій, ефективна панель керування для адміністраторів (список/оновлення/видалення продуктів та керування замовленнями) та особистий кабінет для користувачів. Програми для електронної комерції зосереджені навколо таких технологій, як електронні платіжні шлюзи, системи керування запасами, мобільна комерція, керування ланцюжками постачання та інтернет-ринок.

Деякі популярні веб-сайти електронної комерції включають Amazon, eBay, Walmart, Swiggy, Zomato і т.ін.

Веб-програми CMS. Програмне забезпечення системи керування контентом (CMS) дозволяє користувачам створювати, керувати та змінювати контент на веб-сайті, не володіючи будь-якими технічними знаннями в галузі веб-програмування чи мов розмітки. CMS популярна завдяки використанню в особистих блогах, корпоративних блогах, медіа-джерелах і т. д. Найчастіше використовувані системи управління контентом:

WordPress: це найпопулярніший і найнадійніший вибір серед приватних осіб та професіоналів для розробки веб-сайтів. Існує безліч плагінів, тем та онлайн-підручників, що дозволяють створювати унікальні та інтуїтивно зрозумілі веб-сайти без будь-якої технічної підтримки.

Joomla: це платформа CMS з відкритим вихідним кодом за WordPress. Він поставляється з інтуїтивно зрозумілим інтерфейсом, який допомагає користувачам створювати, керувати та змінювати контент на веб-сайті та в блогах.

Drupal: безкоштовна CMS з інтерфейсом для створення онлайн-

спільнот. Люди широко використовують його для особистих блогів, сторінок новин в Інтернеті, ЗМІ, корпоративних блогів, професійних блогів тощо.

Веб-програми порталу належать до програм, які дозволяють авторизованим користувачам отримувати доступ до репозиторію даних компанії. Портали найкраще підходять для компаній та підприємств, які дозволяють користувачам створювати особистий профіль та включати різні відомості, такі як чати, електронні листи та форуми для публікації контенту. Тільки учасники порталу мають винятковий доступ до даних. Щоразу, коли користувач входить на портал, постачальник послуг відстежує дії користувача на веб-сайті. Існують різні типи веб-порталів, які часто пов'язані з різними галузями.

Освітні портали, студентські портали, портали співробітників, банківські портали, портали пацієнтів тощо є прикладами веб-додатків порталу.

Одно сторінковий додаток або SPA - це динамічний додаток, який дозволяє відвідувачам спілкуватися в браузері без будь-яких перешкод у певному темпі. Запити та відповіді користувачів виконуються ефективніше та швидше, ніж звичайні веб-програми, оскільки SPA виконує логіку в інтернет-браузері, а не на сервері. SPA відносно прості, їх легко створювати та налагоджувати при розгортанні.

Наприклад, при перегляді вашої поштової програми заголовки та бічні панелі залишаються недоторканими при доступі до вашої поштової скриньки. Більшість сайтів соціальних мереж, поштових служб, онлайн-відео/аудіо плеєрів швидко виграють від використання окремих веб-додатків та гнучкості. Gmail, Paypal, Netflix, Twitter, Google Maps і т.д. реалізують одно сторінкові програми.

Багатосторінкові програми або MPA складаються з декількох сторінок і перезавантажують всю сторінку з сервера щоразу, коли користувач переходить на іншу сторінку. MPA створюються з використанням різних мов, таких як HTML, CSS, JavaScript, AJAX, jQuery і т. д. Ці програми цінні

своєю масштабованістю без обмежень за кількістю сторінок і надають велику інформацію про продукти або послуги компанії.

Приклади МРА включають каталоги, веб-портали, бізнес-додатки та інтернет-магазини.

Багатофункціональні веб-програми для Інтернету. Багаті інтернет-додатки - це додаток з багатьма з тих же функцій та зовнішнього вигляду, що й настільний додаток. В першу чергу він має безліч функцій і є більш привабливим та швидким, ніж стандартні веб-програми. Вони покладаються на клієнтські плагіни через обмеження свого браузера. Ці програми створюються за допомогою таких інструментів, як AJAX, Java, JavaFX, Adobe Flash, Adobe Flex, та їх також можна використовувати в автономному режимі. Інтернет-програми з широкими можливостями візуально привабливі і забезпечують відмінний інтерфейс користувача.

Документи Google, Youtube, карти Google – популярні приклади багатофункціональних інтернет-додатків.

Прогресивні веб-застосунки (PWA) – це найбільш популярні та добре розроблені типи веб-додатків, зовні схожі на мобільні програми. Вони також відомі як крос-платформні веб-додатки, в яких використовуються найновіші API-інтерфейси браузера та методи прогресивного покращення, що забезпечують native мобільну програму. Основна р PWA – підвищити швидкість та універсальність веб-додатків в умовах повільного інтернет-з'єднання.

Деякі яскраві приклади прогресивних веб-додатків - MakeMyTrip, Forbes, OLX, Starbucks.

## 2.3 Засоби Front-end розробки

### 2.3.1 HTML

HTML розшифровується як HyperText Markup Language . Це стандартна мова розмітки для створення веб-сторінок. Він дозволяє створювати та структурувати розділи, абзаци та посилання за допомогою елементів HTML (будівельних блоків веб-сторінки), таких як теги та атрибути.[9]

HTML має багато варіантів використання, а саме:

- веб-розробка. Розробники використовують HTML-код для проектування того, як браузер відображає елементи веб-сторінки, такі як текст, гіперпосилання та медіа-файли.

- інтернет-навігація. Користувачі можуть легко переходити та вставляти посилання між пов'язаними сторінками та веб-сайтами, оскільки HTML широко використовується для вставлення гіперпосилань.

- веб-документація. HTML дає змогу впорядковувати та формувати документи, подібно до Microsoft Word.

Також варто зазначити, що HTML не вважається мовою програмування, оскільки він не може створювати динамічні функції. Тепер він вважається офіційним веб-стандартом. Консорціум Всесвітньої павутини (W3C) підтримує та розробляє специфікації HTML, а також забезпечує регулярне оновлення.

Звичайний веб-сайт містить кілька різних HTML-сторінок. Наприклад, домашня сторінка, сторінка про програму та сторінка контактів матимуть окремі файли HTML.

Документи HTML – це файли, які закінчуються розширенням .html або .htm. Веб-браузер читає файл HTML і відображає його вміст, щоб користувачі Інтернету могли його переглядати.

Усі сторінки HTML мають ряд елементів HTML, що складаються з

набору тегів і атрибутів. Елементи HTML є будівельними блоками веб-сторінки. Тег повідомляє веб-браузеру, де починається і закінчується елемент, тоді як атрибут описує характеристики елемента.

Три основні частини елемента:

Відкриваючий тег – використовується для вказівки, де елемент починає діяти. Тег загорнутий у відкриваючі та закриваючі кутові дужки. Наприклад, використовуйте початковий тег `<p>`, щоб створити абзац.

Контент – це вихідні дані, які бачать інші користувачі.

Закриваючий тег – те саме, що відкриває тег, але з похилою рискою перед назвою елемента. Наприклад, `</p>` для завершення абзацу.

Поєднання цих трьох частин створить елемент HTML відображено на лістингу 2.1.

### Лістинг 2.1 – Структурування елементів HTML

```
< p > Ось як ви додаєте абзац у HTML. < /p >
```

Іншою важливою частиною елемента HTML є його атрибут, який складається з двох розділів – імені та значення атрибута. Ім'я визначає додаткову інформацію, яку користувач хоче додати, тоді як значення атрибута надає додаткові специфікації.

Наприклад, елемент стилю, який додає фіолетовий колір і сімейство шрифтів `verdana`, зображений на лістингу 2.2.

### Лістинг 2.2 – Наприклад, елемент стилю, який додає фіолетовий колір

```
< p style= "color:purple;font-family:verdana" > Ось як ви  
додаєте абзац у HTML. < /p >
```

Інший атрибут, клас HTML, є найважливішим для розробки та програмування. Атрибут `class` додає інформацію про стиль, яка може працювати з різними елементами з однаковим значенням класу.

Наприклад, ми будемо використовувати той самий стиль для заголовка `<h1>` і абзацу `<p>`. Стиль включає колір фону, колір тексту, межі, поля та

відступи в класі `.important`. Щоб досягти однакового стилю між `<h1>` і `<p>`, додайте `class= «important»` після кожного початкового тегу, зображено на лістингу 2.3.

### Лістинг 2.3 – Додання класу «important»

```
< html >
< голова >
< стиль >
.important {
  колір фону: синій;
  колір: білий;
  рамка: 2 пікселя суцільний чорний;
  поле: 2px;
  відступ: 2 пікселя;
}
< /style >
< /head >
< тіло >
< h1 class = "important" > Це заголовок < /h1 >
< p class = "important" > Це абзац. < /p >
< /body >
< /html >
```

Більшість елементів мають відкриваючий і закриваючий теги, але деякі елементи не потребують закриваючих тегів для роботи, наприклад порожні елементи. Ці елементи не використовують кінцевий тег, оскільки вони не мають вмісту, зображено на лістингу 2.4.

### Лістинг 2.4 – Приклад елементів які не використовують кінцевий тег

```
< img src= "/" alt= "Зображення" >
```

Цей тег зображення має два атрибути: атрибут `src`, шлях до зображення, і атрибут `alt`, текст опису. Однак він не має ні вмісту, ні кінцевого тегу.

Нарешті, кожен документ HTML має починатися з оголошення `<!DOCTYPE>`, щоб повідомити веб-браузеру про тип документа. У HTML5 публічна декларація `doctype HTML` буде такою(лістинг 2.5).

## Лістинг 2.5 – Публічна деклараціяdoctype

```
< !DOCTYPE html >
```

Наразі доступно 142 теги HTML, які дозволяють створювати різні елементи . Незважаючи на те, що сучасні браузери більше не підтримують деякі з цих тегів, вивчати всі різні доступні елементи все ще корисно.

Як і будь-яка інша комп'ютерна мова, HTML має свої переваги та обмеження. Ось плюси та мінуси HTML.

Плюси:

- зручний для початківців. HTML має чітку та узгоджену розмітку, а також неглибоку криву навчання;
- підтримка. Мова широко використовується, за нею стоїть багато ресурсів і велика спільнота;
- доступний. Він є відкритим і абсолютно безкоштовним. HTML працює в усіх веб-браузерах;
- гнучкий. HTML легко інтегрується з серверними мовами, такими як PHP і Node.js.

Мінуси:

- статичний. Ця мова в основному використовується для статичних веб-сторінок. Для динамічної функціональності вам може знадобитися використовувати JavaScript або серверну мову, таку як PHP;
- скрема HTML-сторінка. Користувачі повинні створювати окремі веб-сторінки для HTML, навіть якщо елементи однакові;
- сумісність з браузером. Деякі браузери повільно впроваджують нові функції. Іноді старі браузери не завжди відтворюють нові теги.

### 2.2.2 CSS

Каскадні таблиці стилів (CSS) – це мова, яка дозволяє керувати зовнішнім виглядом веб-сторінки. Мова CSS є рекомендацією Консорціуму Всесвітньої павутини (W3C) , як і HTML або XML.

Стилі дають змогу визначати правила, які застосовуються до одного

або кількох документів HTML . Ці правила впливають на розташування елементів, вирівнювання, шрифти, кольори, поля та інтервали, рамки, фонові зображення тощо.

Метою CSS є відокремлення структури документа HTML від його представлення[10]. Дійсно, за допомогою HTML ми можемо визначити як структуру (вміст та ієрархію між різними частинами документа), так і презентацію. Однак це викликає деякі проблеми. За допомогою пари HTML/CSS ми можемо створювати веб-сторінки, де структура документа міститься у файлі HTML, а презентація – у файлі CSS.

За допомогою CSS ви можете, наприклад, в. визначити набір стилістичних правил , які застосовуються до всіх сторінок веб-сайту. Це полегшує зміну зовнішнього вигляду всього сайту. CSS також дозволяє визначати різні правила для кожної підтримки дисплея (класичний браузер , телевізор, мобільна підтримка, зчитувач Брайля...). CSS також можна використовувати для покращення доступності веб-документів.

Крім того, порівняно з HTML, CSS пропонує нові функціональні можливості з точки зору стилю. Насправді HTML дозволяє керувати документами досить коротким стилем.

Приклад правила CSS зображено на лістингу 2.6.

#### Лістинг 2.6 – Приклад правила CSS

```
p font-color: red; font-size: 12pt
```

Пояснення: це правило визначає колір і розмір тексту для елементів <p> (абзаців) документа HTML. Тож усі абзаци матимуть червоний колір шрифту та розмір 12 пунктів.

Незважаючи на те, що можна використовувати інші мови, існує в основному 4 мови, які формують веб сьогодні: HTML , CSS, Javascript і PHP. HTML і CSS – це мови опису, Javascript і PHP – мови програмування.

HTML і CSS використовуються для опису того, як відображається веб-

сторінка. HTML є серцем мережі, її основою. Він структурує різні елементи веб-сайту. CSS використовується для форматування цих елементів, щоб надати їм візуальний стиль.

CSS - це абревіатура від Cascading Style Sheets. Слово «каскад» нагадує нам, що стилі можна класифікувати за ступенем важливості. Отже, за допомогою CSS ви можете контролювати, як саме відобразатиметься веб-сайт.

Як і HTML, CSS є мовою, яку легко вивчити. Стиль веб-документа можна реалізувати безпосередньо в коді HTML або вставити в інший документ.

Можна застосувати CSS безпосередньо в тезі HTML елемента за допомогою атрибута style, приклад зображено на лістингу 2.7.

Лістинг 2.6 – Надання абзацу «Мій текст» призначено білий фон

```
<p style="background-color: #FFFFFF;">My text</p>
```

Внутрішні, вбудовані та зовнішні стилі CSS.

Внутрішній стиль. Стилі CSS які створені так будуть завантажуватися кожен раз при оновленні веб-сторінки, що швидше за все збільшить час завантаження сторінки. Окрім того, не є можливим використовувати той же самий стиль CSS на декількох сторінках, так як він знаходиться на одній сторінці. Але переваги з цього такого маються. Розміщення його тільки на одній сторінці буде спрощувати спільне використання шаблону для попереднього перегляду.

Зовнішній метод скоріш за все буде найбільш зручним, бо все буде робитися ззовні в файлі .css. Це значить, що можна зробити всі стилі в окремому файлі, потім застосувати CSS на будь-яку сторінку, яку буде потрібно. Також зовнішній стиль повинен покращити час завантаження.

Вбудований стиль CSS. Inline буде працювати з певними елементами, які мають tag <style>. Кожен з компонентів має бути стилізованим, в наслідок

цього це може бути не найкращим чи найшвидшим способом обробки у CSS. Хочу він не є найкращим, але він може стати в нагоді. Наприклад, якщо буде бажання змінити один елемент, швидко переглянути зміни або, можливо, у вас немає доступу до файлів CSS.

### 2.2.3 JavaScript

JavaScript – це динамічна мова комп'ютерного програмування. Він легкий і найчастіше використовується як частина веб-сторінок, реалізації яких дозволяють сценарію на стороні клієнта взаємодіяти з користувачем і створювати динамічні сторінки. Це інтерпретована мова програмування з об'єктно-орієнтованими можливостями.

Спочатку JavaScript був відомий як LiveScript, але Netscape змінив його назву на JavaScript, можливо, через хвилювання, викликане Java. JavaScript вперше з'явився в Netscape 2.0 у 1995 році під назвою LiveScript. Ядро мови загального призначення було вбудовано в Netscape, Internet Explorer та інші веб-браузери[11].

Специфікація ECMA-262 визначила стандартну версію основної мови JavaScript:

- JavaScript – це легка інтерпретована мова програмування;
- призначений для створення мережевих додатків;
- доповнює та інтегрується з Java;
- доповнює та інтегрується з HTML;
- відкрита та крос платформна.

JavaScript на стороні клієнта

JavaScript на стороні клієнта є найпоширенішою формою мови. Сценарій має бути включений до HTML-документа або посилання на нього для того, щоб код інтерпретувався браузером.

Це означає, що веб-сторінка не обов'язково має бути статичним HTML, але може містити програми, які взаємодіють з користувачем, керують

браузером і динамічна створюють вміст HTML.

Клієнтський механізм JavaScript надає багато переваг перед традиційними серверними скриптами CGI. Наприклад, ви можете використовувати JavaScript, щоб перевірити, чи користувач ввів дійсну адресу електронної пошти в поле форми.

Код JavaScript виконується, коли користувач надсилає форму, і лише якщо всі записи дійсні, вони надсилаються на веб-сервер.

JavaScript можна використовувати для перехоплення ініційованих користувачем подій, таких як натискання кнопок, навігація за посиланням та інші дії, які користувач ініціює явно чи неявно.

### Переваги JavaScript

Перевагами використання JavaScript є:

- менше взаємодії з сервером – Ви можете перевірити введені користувачем дані перед тим, як відправити сторінку на сервер. Це економить трафік сервера, що означає менше навантаження на ваш сервер;
- миттєвий зворотний зв'язок з відвідувачами – їм не потрібно чекати перезавантаження сторінки, щоб побачити, чи не забули вони щось ввести;
- підвищена інтерактивність – можна створювати інтерфейси, які реагуватимуть, коли користувач наведе на них мишу або активує їх за допомогою клавіатури;
- зозширені інтерфейси – можна використовувати JavaScript, щоб включити такі елементи, як компоненти перетягування та скидання та повзунки, щоб надати відвідувачам вашого сайту розширений інтерфейс.

### Обмеження JavaScript.

Ми не можемо розглядати JavaScript як повноцінну мову програмування. У ньому відсутні наступні важливі функції:

- JavaScript на стороні клієнта не дозволяє читати або записувати файли. Це збережено з міркувань безпеки;
- JavaScript не можна використовувати для мережових програм, оскільки така підтримка відсутня;

- JavaScript не має багатопоточності чи багатопроцесорності.

Знову ж таки, JavaScript – це легка, інтерпретована мова програмування, яка дозволяє вбудовувати інтерактивність у статичні сторінки HTML.

#### 2.4 Інструменти розробки Back-end

Back-end (або «серверна») – це та містична й неловима частина програми, яка залишається непоміченою. Кожна значна, але складна програма містить більше коду “Non-user interface code”, ніж коду інтерфейсу користувача.

Код інтерфейсу користувача визначає частину зовнішньої розробки програми, пов’язану з безпосередньою взаємодією з користувачем.

Код “Non-user interface code” має різну логіку, яка виконується у фоновому режимі, включаючи зберігання та отримання даних, бізнес-правила та логічне обчислення. Усе це відбувається за лаштунками, і все це відбувається за допомогою бекенд-розробника.

Бекенд-розробник головним чином займатиметься створенням логіки програми, над якою він працює. Фронтальні розробники відповідають за створення інтерфейсу користувача, а бекенд-розробники мають писати коди, які забезпечують роботу всього цього[12].

Розробники проектують архітектуру системи, вирішують проблеми та впроваджують алгоритми. Простіше кажучи, бекенд-розробка передбачає написання коду, який не видно безпосередньо.

Розглядаючи загальну картину потрібно навести найкращі інструменти бекенд-розробки для веб-розробників.

Мови програмування:

- PHP, аббревіатура Hypertext Preprocessor, є серверною мовою сценаріїв і є однією з найпоширеніших мов програмування для серверної веб-розробки. Ми можемо підтвердити це твердження, повідомивши вам, що такі платформи, як Wikipedia, WordPress, Facebook та багато інших, покладаються

на PHP. Цій конкретній мові віддається перевага для веб-розробки через ряд важливих причин, таких як сумісність між платформами, функції ООП, легка інтеграція з HTML, CSS, JavaScript тощо, величезна підтримка спільноти, краща гнучкість і безпека та багато іншого. Крім того, мову досить легко вивчати та використовувати. Крім того, існують різні відомі фреймворки PHP, такі як Laravel, Symfony, CodeIgniter тощо, які ви можете розглянути;

- Python, іншою мовою, якій надають перевагу для розробки бекенда, особливо в наш час, є Python. Це мова програмування високого рівня загального призначення, яка підтримує кілька парадигм програмування, таких як об'єктно-орієнтоване, процедурне та функціональне. Instagram, Spotify, Google тощо є одними з популярних платформ, які використовують Python у своїх технологічних пакетах. Ця мова надає вам такі чудові функції, як підтримка багатих бібліотек, легка інтеграція з іншими мовами, підтримка графічного програмування, сумісність із популярними технологіями тощо. Крім того, Python має дуже простий синтаксис і забезпечує кращу читабельність коду, що згодом робить його легко вивчити та використовувати. Кілька популярних веб-фреймворків Python, які ви можете взяти до уваги, це Django, Flask тощо;

- JavaScript не потребує представлення через його велику роль у веб-розробці. Насправді це одна з основних одиниць веб-розробки поряд з HTML і CSS. Величезна кількість розробників надає перевагу цій мові для веб-розробки, і вона займає перші позиції, а також у різних відомих індексах найкращих мов програмування. Окрім веб-розробки, цю мову також можна використовувати для розробки ігор, розробки мобільних додатків тощо. Деякі з важливих особливостей цієї конкретної мови: імперативна та структурована, легка, підтримка об'єктно-орієнтованого програмування, незалежність від платформи та багато інших. Крім того, найкраща частина полягає в тому, що JavaScript можна використовувати як для розробки фронтенду, так і для серверної розробки. Також є інші мови, які можна розглянути: Java, Go, Ruby тощо;

- Laravel є одним із найпопулярніших фреймворків PHP у світі технологій. Ця веб-платформа з відкритим вихідним кодом відповідає архітектурному шаблону Model-View-Controller (MVC) і надає вам численні додаткові функції, як-от вбудований інструмент командного рядка під назвою Artisan, попередньо встановлені об'єктно-орієнтовані та модульні бібліотеки, Eloquent ORM ( Object Relational Mapping), Template Engine та багато інших. Крім того, фреймворк надає вам різноманітні інструменти для виконання певних завдань, таких як впровадження залежностей, модульне тестування тощо. Готові до використання пакети, надані Laravel, включають Cashier для керування послугами виставлення рахунків за передплатою, Envoy, який забезпечує чистий і мінімальний синтаксис для визначення загального завдань, які ви виконуєте на своїх віддалених серверах, Socialite для спрощеного механізму автентифікації OAuth за допомогою таких постачальників, як Facebook, GitHub, Google тощо;

- Django – це веб-фреймворк із відкритим вихідним кодом на основі Python, який дозволяє вам робити веб-розробку більш ефективно та без будь-яких проблем. Django слідує архітектурному шаблону модель-шаблон-перегляди (MTV). Причиною величезної популярності та попиту на цей конкретний фреймворк є деякі його помітні особливості, такі як розширюваність, швидкий розвиток, масштабованість, безпека, велика спільнота та багато іншого. Підприємства використовують Django для різноманітних окремих областей веб-розробки, таких як платформи соціальних мереж, науково-обчислювальні платформи, системи керування контентом тощо. Деякі з популярних веб-сайтів, які використовують Django: Instagram, Mozilla, Pinterest тощо. Якщо ви хочете почати веб-розробку за допомогою Python, ви можете спробувати Django. Інші фреймворки, які ви можете розглянути: Angular, Meteor, Spring, Ruby on Rails тощо.

## 2.5 Функціональні вимоги

Visual Studio Code (відомий як VS Code ) – це безкоштовний текстовий редактор із відкритим кодом від Microsoft[13]. VS Code доступний для Windows, Linux і macOS. Хоча редактор відносно легкий, він містить деякі потужні функції, які зробили VS Code одним із найпопулярніших інструментів середовища розробки останнім часом (рисунок 2.6).

### Most Popular Development Environments

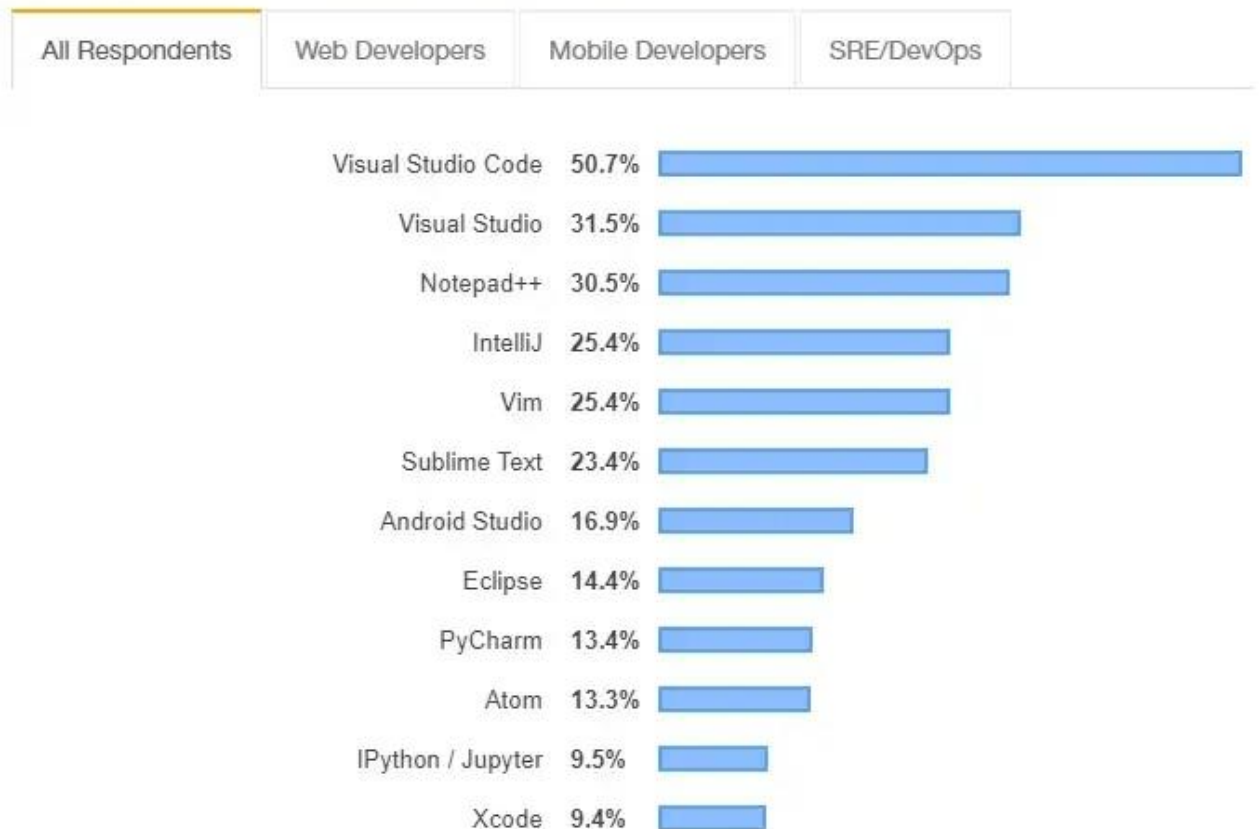


Рисунок 2.6 – Рівень популярності середовищ розробки

Його популярність пояснюється розвитком галузі веб-розробки в ці роки та потребою розробників у легкому добре виконаному редакторі з невеликою кількістю функцій, але менш складним, ніж інші, доступні на ринку.

Він також безкоштовний, і його розробляє та підтримує корпорація Майкрософт із сучасним підходом із використанням Electron .

Це деякі характеристики, які виводять VSCode на перше місце серед редакторів кодування, але є й інші.

### 1 З відкритим кодом

Безпрецедентною перевагою є той факт, що код VS здебільшого відкритий .

Це означає не лише те, що програмне забезпечення безкоштовне для використання, але й те, що ви можете допомогти його покращити.

Перехід на відкрите програмне забезпечення – чудовий вибір, якщо ви хочете збільшити залучення спільноти .

Хоча не всі користувачі VS Code роблять свій внесок у кодову базу, у них виникає певне відчуття єдності.

Інші редактори коду, такі як Atom , і навіть цілі IDE, такі як Eclipse або NetBeans , також є відкритими.

### 2 Простота

Від перших кроків до встановлення нових розширень, усе у VS Code здається простим та інтуїтивно зрозумілим .

Зважаючи на це, я не маю на увазі, що коду VS бракує будь-яких можливостей, зовсім ні.

Завдяки розширюваній архітектурі (про яку ми поговоримо трохи пізніше), VS Code, навіть будучи лише редактором коду, може бути цінною альтернативою іншим складнішим IDE.

Але, на відміну від цих IDE, це все ще вдається зробити це компактно, зручно для користувача .

Ця простота також проявляється через продуктивність VS Code , яка є напроцуд хорошою, особливо для програми Electron.

### 3 Знайомство

VS Code створено за допомогою Electron – фреймворку для створення настільних додатків із JavaScript за допомогою Chromium і Node.js.

Багато веб-розробників, які використовують VS Code, усвідомлюють і цінують цей факт, але не всі це роблять, головним чином через горезвісне

високе використання пам'яті та низьку продуктивність програм Electron .

З іншого боку, VS Code використовує свою вражаючу інтеграцію TypeScript для активації автозавершення та інших корисних функцій редагування як для JS , так і для TS .

Якщо ви використовуєте TypeScript, можна сказати, що VS Code – ваш найкращий вибір.

Отже, той факт, що VS Code побудовано на основі веб-технологій, а також забезпечує їх чудову підтримку, робить його знайомим і майже вибором за замовчуванням для значної частини його зростаючої бази користувачів, веб-розробників.

#### 4 Мінімальний дизайн

Дизайн, як правило, дуже суб'єктивна річ, але майбутнє є мінімальним, і VSCode вирішив прийняти принципи цього підходу до дизайну.

#### 5 Розширення

На ринку VS Code є буквально тисячі розширень, і нові з'являються щодня.

Розширення можуть служити багатьом цілям.

Від тем інтерфейсу користувача, схожих на розширення, до підтримки мов програмування, налагодження, інтеграції Git і навіть програвачів Spotify.

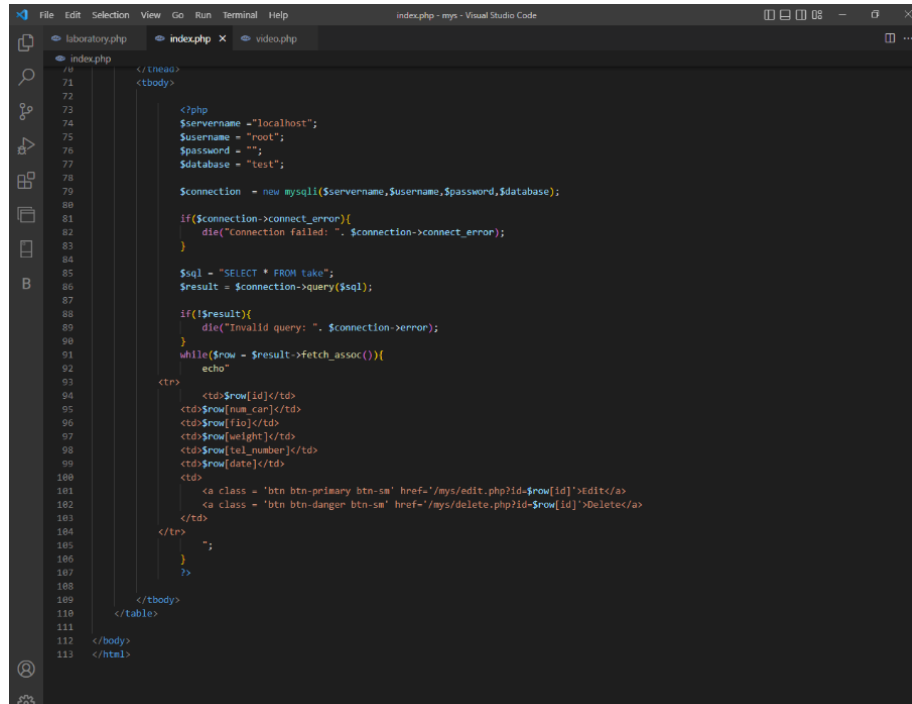
Головне вікно Visual Studio Code представлено на рисунку 2.7:

В UML діаграмі варіантів використання моделюють поведінку системи та допомагають охопити вимоги системи.

Діаграми варіантів використання описують функції високого рівня та область застосування системи. Ці діаграми також ідентифікують взаємодію між системою та її акторами. Варіанти використання та актори на діаграмах варіантів використання описують, що робить система та як актори її використовують, але не те, як система працює всередині.

Діаграми варіантів використання ілюструють і визначають контекст і вимоги всієї системи або важливих частин системи. Ви можете змоделювати складну систему за допомогою однієї діаграми варіантів використання або

створити багато діаграм варіантів використання для моделювання компонентів системи. Зазвичай ви розробляєте діаграми варіантів використання на ранніх етапах проекту та посилаєтеся на них протягом усього процесу розробки.



```
71 </thead>
72 <tbody>
73
74 <?php
75 $servername = "localhost";
76 $username = "root";
77 $password = "";
78 $database = "test";
79
80 $connection = new mysqli($servername,$username,$password,$database);
81
82 if($connection->connect_error){
83     die("Connection failed: ". $connection->connect_error);
84 }
85
86 $sql = "SELECT * FROM take";
87 $result = $connection->query($sql);
88
89 if(!$result){
90     die("Invalid query: ". $connection->error);
91 }
92 while($row = $result->fetch_assoc()){
93     echo "
94     <tr>
95         <td>".$row[id]</td>
96         <td>".$row[num_car]</td>
97         <td>".$row[fio]</td>
98         <td>".$row[weight]</td>
99         <td>".$row[tel_number]</td>
100        <td>".$row[date]</td>
101        <td>
102            <a class = "btn btn-primary btn-sm" href="/mys/edit.php?id=".$row[id]'>Edit</a>
103            <a class = "btn btn-danger btn-sm" href="/mys/delete.php?id=".$row[id]'>Delete</a>
104        </td>
105    </tr>
106    ";
107 }
108 </tbody>
109 </table>
110
111 </body>
112 </html>
```

Рисунок 2.7 – Головне вікно Visual Studio Code

Розроблена контекстна uml use case diagram (рисунок 2.8), дозволяє стисло сформулювати вимоги від замовника. Для опису існує зручний спосіб відображення у вигляді User Stories [14].

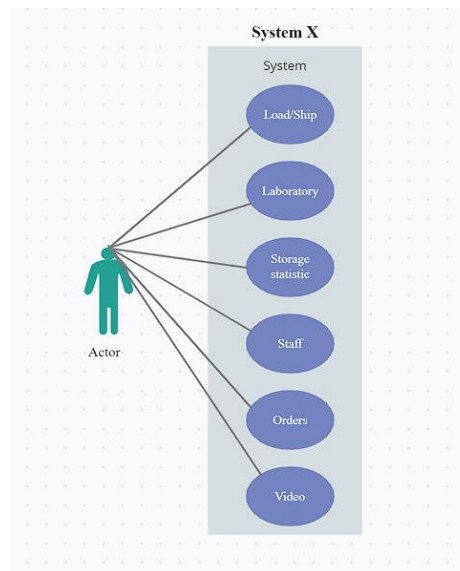


Рисунок 2.8 – Uml use case diagram

Веб-додаток повинен мати сучасний дизайн, зрозумілий та приємний інтерфейс, також повинен мати розділи з кількістю прийнятого та надісланого товару, відображенням графічної статистики по місяцям, розділ лабораторії та результатами їх аналізів над товаром, розділ персоналу, заказів та камер. У розділах з інформацією, а саме, в першому, третьому та п'ятому, користувач повинен мати змогу переглянути усю інформацію по товару (зерну) відповідно перший розділ – хто привіз, скільки, точна дата, другий – результати тестів над товаром (зерном) по трьом головним критеріям -це зараженість шкідниками, вологість, домішки та відображення типу товару.

У розробленому веб застосунку буде виконано багато сторінковий сайт заповнений матеріалами теми завдання.

Перша відображена сторінка буде називатися “Loading/Shipping”, на цій сторінці буде відображена інформація яка була додана до бази даних заздалегідь. Вона призначена для відображення інформації в форматі таблиці про “Завантаження/Відвантаження” продукції. Користувач побачить порядковий номер, номер машини, яка привезла товар, ФІО, кількість перевезеного вантажу в тонах, номер телефону та точний час у форматі datetime.

Наступна вкладка “Storage statistics” відобразить графічну діаграму по місяцям 1-12 зі статистикою по об’єму в тонах чи продаж в гривнях.

В розділі “Laboratory” схоже з першою сторінкою буде відображено дані про результати тестів описаного типу зерна в лабораторії на привід наявності шкідників, рівень вологості та рівень домішок.

Розділ “Staff” в якому буде відображена інформація про персонал.

Схоже з першим та третім розділом піде розділ “Orders” в якому буде відображена інформація стосовно заказів (тип, об’єм, ціна, час, замовник)

Останній розділ буде містити камери спостереження.

## 2.6 Вибір і обґрунтування комплексу технічних засобів

Схема комплексу технічних засобів функціональної задачі «Система адміністрування підготовки спортсменів» зображена на рисунку 2.9.

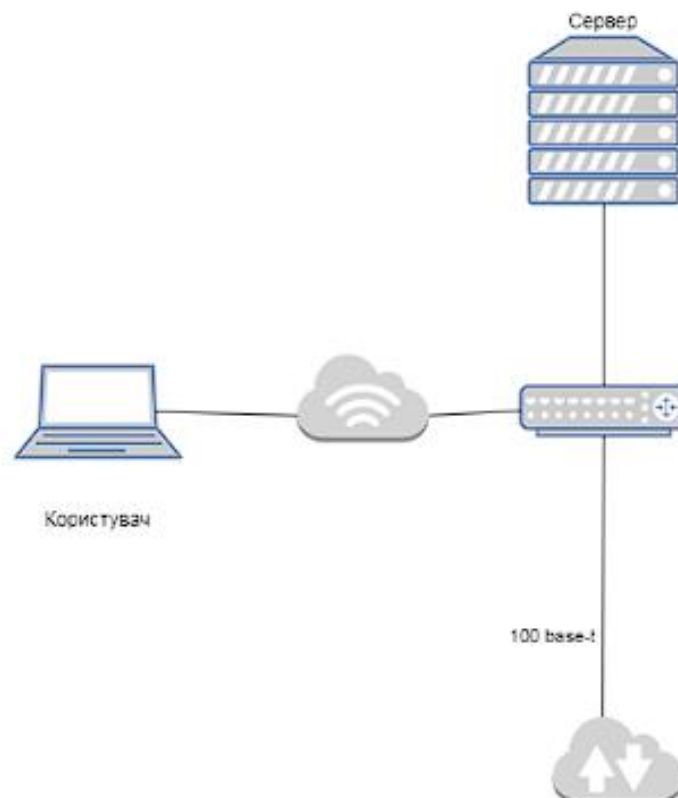


Рисунок 2.9 – Схема фрагменту КТЗ

Для якісної роботи «користувач» повинен використовуватися такі технічні засоби: ПК, доступ до інтернету.

Мінімальні потреби до ПК «Контент менеджера»: процесор Intel core i3 4 ядра; RAM 8 Гб; HDD 500 Гб; Інтернет 100+ мб/с; ОС Windows 10.

Мінімальні характеристики сервера: ОС Windows Server 2012; Процесор: Intel Xeon W3565; Оперативна пам'ять: 16 Гб; Hdd: 100 гб.

Для коректного відображення сайту та запобігання яких небудь можливих помилок чи некоректного відображення елементів бажано використовувати останні версії актуальних браузерів: Google Chrome; Opera; Fire Fox; Safari; Edge/IE (останньої версії).

Розширення екрану: рекомендоване: 1440 x 900 або вище; Також, можна працювати і при дозволі 1024 x 768, але швидше за все, це буде не зовсім зручно.

На прикладі Google Chrome користувач має мати пристрій в наведеними нижче критеріями. Важливо! Браузер Chrome у Windows 7 продовжить отримувати важливі виправлення системи безпеки та оновлення для стабільної роботи до 15 січня 2023 року[15].

Вимоги до Windows-системи: Windows 7, Windows 8, Windows 8.1, Windows 10 або більш пізньої версії; Процесор Intel Pentium 4 або новішої версії з підтримкою SSE3; Примітка. На серверах повинна бути встановлена ОС Windows Server 2008 R2, Windows Server 2012 Windows Server 2012 R2 або Windows Server 2016.

Вимоги до macOS -системи: macOS High Sierra 10.13 або пізнішої версії.

Вимоги до Linux - системи: Ubuntu 18.04 (64-розрядна версія) або пізнішої версії, Debian 10 або пізнішої версії, openSUSE 15.2 або пізнішої версії, Fedora Linux 32 або пізнішої версії. Процесор Intel Pentium 4 або пізнішої версії із підтримкою SSE3.

Вимоги до Android системи: Android 7.0+ Nougat.

Таким чином розроблено оригінальну інформаційну модель системи моніторингу, з застосування серверних технологій й засобів та інструментів

Front-End розробки. Розглянуто функціональні вимоги до створення таких систем, а також здійснено вибір та обґрунтування технічних засобів реалізації функціональної моделі.

## 3 ПРОГРАМНА РЕАЛІЗАЦІЯ

### 3.1 Структура системи

Згідно з технічним завданням необхідно розробити веб-додаток для моніторингу процесів на елеваторному комплексі. Він буде призначений для швидкого, вільного та зручного доступу до інформації яка проходить по підприємству. До цієї інформації входять багато розділів та критеріїв:

- основні особливості додатку;
- велика кількість даних, яка повинна бути групована належним чином;
- відображення великих масивів даних згідно розділів;
- простий і зручний інтерфейс для роботи користувача;
- відображення візуальних графіків для кращого розуміння та візуалізації картини відносно оборотів в цілому.

#### Вибір методу рішення

На даний момент часу існує велика кількість способів створення програмного забезпечення, схожого на \*.

Для вирішення проблеми можна надати такі програмні засоби як:

- мова гіпертекстової розмітки (HTML);
- мова опису (CSS);
- редактор Visual studio code;
- PHP;
- JavaScript;
- XAMMP (MySQL, Apache).

За допомогою усіх цих засобів (інструментів) буде повністю забезпечено коректне виконання завдання на проектування.

Розроблений веб-додаток призначається як сайт – автоматизована система моніторингу підприємства по типу «елеваторного комплексу».

Структура сайту ставиться до того, як організується контент всього сайту. Веб-додаток зазвичай складається з контенту по великій кількості пов'язаних тем, наведених в повідомленнях і на веб-сторінках. Структура сайту виглядає так, як був групуваний контент, в результаті пов'язаний та показаний користувачеві[16]. Структура сайту представлена на рисунку 3.1:

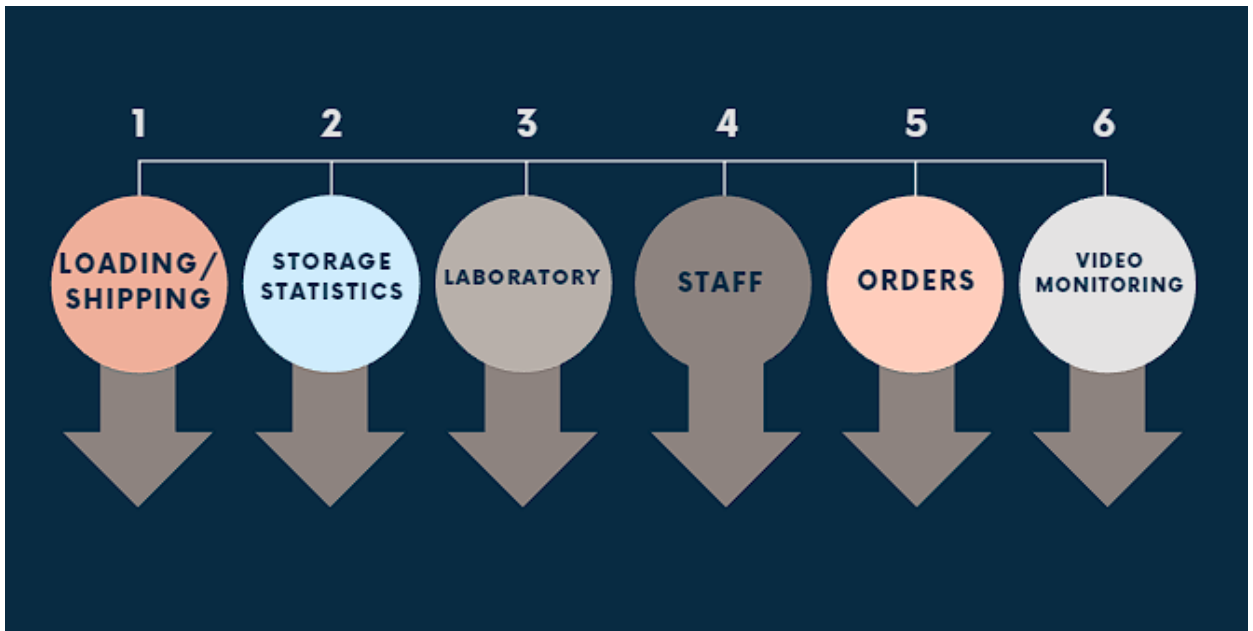


Рисунок 3.1 – Структура сайту

3.2 Розробка, обґрунтування та оцінка алгоритмів вирішення постановленої задачі

Слово «алгоритм» походить від «algorithmi» – латинської форми написання імені великого математика аль-Хорезмі, який сформулював правила виконання арифметичних дій. Тому спочатку під алгоритмом розуміли тільки правила виконання чотирьох арифметичних дій над багатоцифровими числами в десятковій системі числення. Зараз він є одним із фундаментальних понять інформатики.

Алгоритм – це скінчена послідовність команд (вказівок), що визначає, які дії та у якому порядку потрібно виконати, щоб досягти поставленої мети.

Алгоритм складається із команд – окремих указівок виконавцеві виконати деякі конкретні дії. Команди алгоритму виконуються одна за одною, і на кожному кроці відомо, яка команда повинна виконуватися. Почергове виконання команд за кінцеве число кроків приводить до розв’язання задачі. Для того щоб виконавець міг розв’язати задачу за заданим алгоритмом, він повинен уміти виконувати кожен з дій, що вказується командами алгоритму [17].

Схема алгоритму задачі для веб-додатку «Система моніторингу процесів на елеваторному комплексі» наведена на рисунку 3.2.

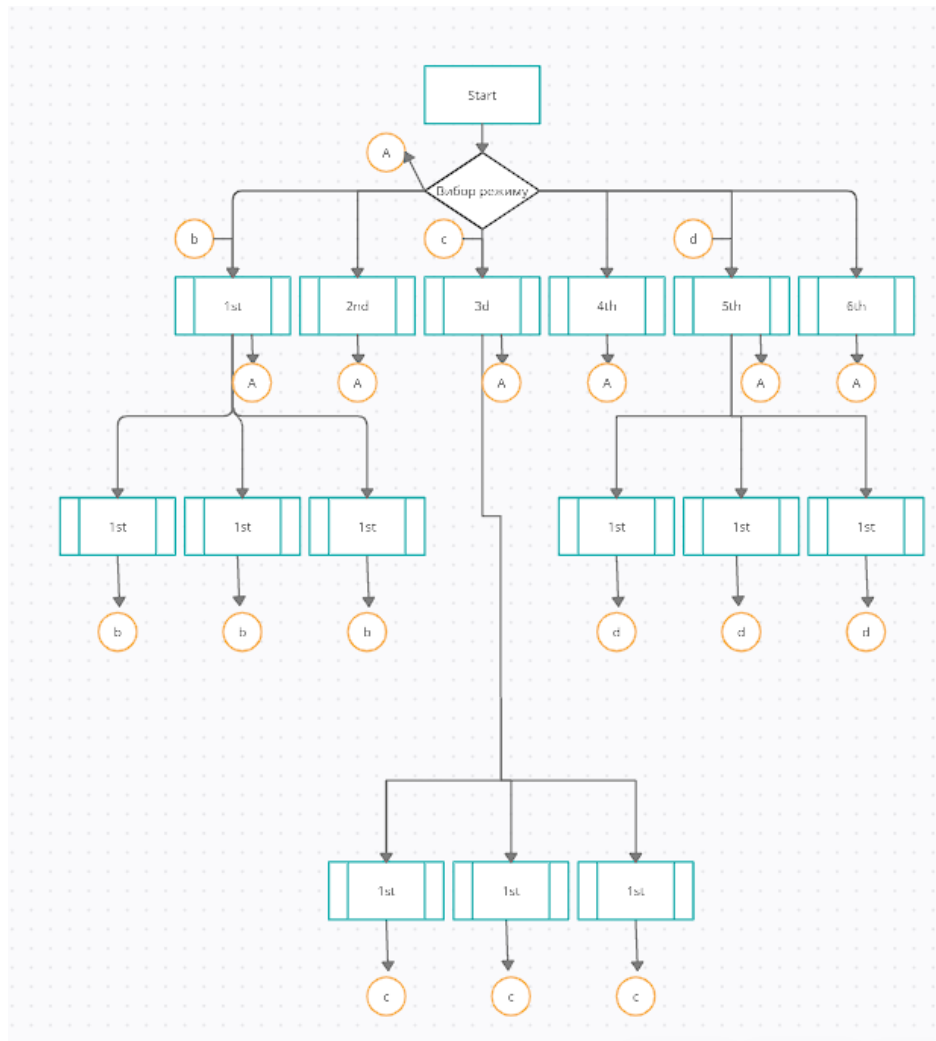


Рисунок 3.2 – Схема алгоритму роботи

### 3.3 Реалізація додатку

Згідно з попередньо розробленою структурою був створений Web-сайт. До його складу входять всі оговорені структурні елементи, перехід по яким виконується шляхом гіперпосилань.

У розробку сайту входить процес створення бази даних та її заповнення, створення Веб-сторінок, на яких в часом будуть вставлені всі інші елементи сайту. В процесі цього проходить формування структурних блоків сайту – відокремлених модулів, в кожного з яких є певна роль і кожен відповідатиме за певний функціонал створеного ресурсу.

#### 3.3.1 Створення та заповнення бази даних

Для адміністрування бд було вибрано phpMyAdmin – mySQL з допомогою АРАСНЕ, які в свою чергу пряцюють за допомоги ХАМРР, рисунок 3.3.

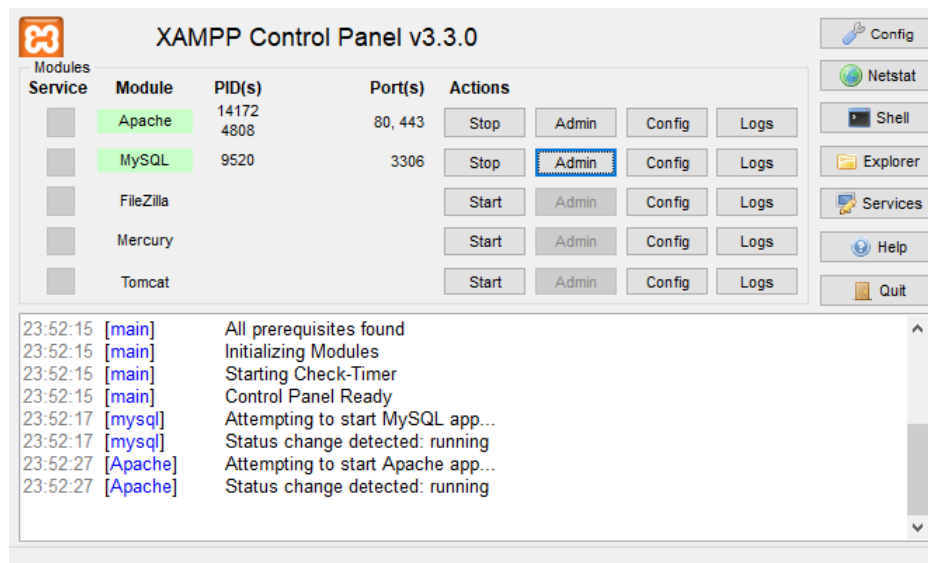


Рисунок 3.3 – Використання ХАМРР

Була створена 1 бд з назвою «test» та 5 таблиць. Зразок першої таблиці з назвою «take», яка відповідає першому розділу «Loading/Shipping» (розділ відповідаючий за прийом товару) зображен нижче на рисунку 3.4.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
1	id	int(10)		UNSIGNED	Нет	Hem		AUTO_INCREMENT	Изменить Удалить Ещё
2	num_car	text	utf8mb4_general_ci		Нет	Hem			Изменить Удалить Ещё
3	fio	text	utf8mb4_general_ci		Нет	Hem			Изменить Удалить Ещё
4	weight	int(11)			Нет	Hem			Изменить Удалить Ещё
5	tel_number	bigint(20)			Нет	Hem			Изменить Удалить Ещё
6	date	datetime			Нет	Hem			Изменить Удалить Ещё

Рисунок 3.4 – Врзрок таблиці даних для першого розділу сайту

Заповнення таблиці «take» стартовими занченнями: «id» - порядковий номер, «num\_car» - номер машини, яка доставила товар, «fio» - прізвище ім'я по батькові, «weight» - вага привезеного товару в тоннах, «tel\_number» - телефонний номер людини, яка доставила товар, «date» - дата та час формату «datetime» для точного відображення часу загрузки чи виврузки товару. Приклад заповнення наведено нижче на рисунку 3.5:

```

INSERT INTO `take` (`id`, `num_car`, `fio`, `weight`, `tel_number`, `date`) VALUES (NULL, 'AX 2121 AA', 'Agat Albert Albertovich', '30', '380988342999', '2022-12-15 03:15:36'), (NULL, 'AA 3333 AE', 'Shevchenko Taras Grigorovich', '10', '380660212323', '2022-12-15 03:15:36'), (NULL, 'BM 2323 AM', 'Sanin Dima Albertovich', '12', '380787878787', '2022-12-15 03:15:36');

```

id	num_car	fio	weight	tel_number	date
1	AX 2121 AA	Agat Albert Albertovich	30	380988342999	2022-12-15 03:15:36
2	AA 3333 AE	Shevchenko Taras Grigorovich	10	380660212323	2022-12-15 03:15:36
3	BM 2323 AM	Sanin Dima Albertovich	12	380787878787	2022-12-15 03:15:36
4	AX 2121 AA	Agat Albert Albertovich	30	380988342999	2022-12-15 03:15:36
5	AA 3333 AE	Shevchenko Taras Grigorovich	10	380660212323	2022-12-15 03:15:36
6	BM 2323 AM	Sanin Dima Albertovich	12	380787878787	2022-12-15 03:15:36

Рисунок 3.5 – Вигляд таблиці «take» першого розділу

Таким самим чином додається ще 4 таблиці «laba» - статистика сховища, «orders» - заклази, «staff» - персонал, «skald» - результати тестві

лабораторії і заповнюється відповідними даними рисунок 3.6.

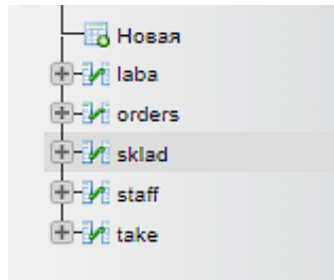


Рисунок 3.6 – База даних з таблицями

### 3.3.2. Розробка візуальної частини

Першим елементом у створеному веб-додатку став “ navbar ”, який вображається на всіх головних сторінках, заповнений відповідними назвами «Loading/Shipping», «Storage statistics», «Laboratory», «Staff», «Orders», «Video monitoring» та виглядає наступним образом(рисунок 3.7).

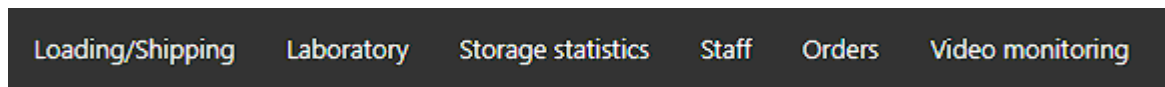


Рисунок 3.7 – Зовнішній вигляд navbar

Далі була виконана сторінка «Loading/Shipping», на якій відображаються данні, які містяться у базі даних "test" у таблиці "take". На сторінці строїться таблиця з стовбцями «id» - для id користувача, «Car Number» - для номеру машини, людини яка привезла товар, «Name Surname Patronymic» - для ім'я прізвища та по батькові, людини яка привезла товар, «Cargo(t.)» - для відображення ваги, «Phone number» - для відображення номеру людини, яка привезла товар, «Date and time» - точна дата та час.

Також після створення таблиці треба достати данні з бд, для цього треба підключитися до неї, робиться це за допомогою коду наведеного в лістингу 3.1.

### Лістинг 3.1 – Підключення до бд

```
<?php
    $servername = "localhost";
    $username = "root";
    $password = "";
    $database = "test";

    $connection = new
mysqli($servername, $username, $password, $database);?>
```

Наступним кроком було додано 2 кнопки біля кожного рядку таблиці:

- редагувати;
- видалити.

Відповідно назв перша з них бере данні з таблиці, редагує їх та повертає обратно до таблиці. Друга же видаляє запис з таблиці та бд.

Завершувальним кроком на сторінці стало додавання ще однієї кнопки для додання нового запису до таблиці.

Також при редагуванні та створенні додані перевірки на поля для коректного вводу даних користувачем.

Ітоговий вигляд сторінки зображений на рисунку 3.8.

ID	Car Number	Name Surname Patronymic	Cargo (t.)	Phone number	Date and time	Edit	Delete
1	AX 2121 AA	Agat Albert Albertovich	30	380988342999	2022-12-15 03:15:36	Edit	Delete
2	AA 3333 AE	Shevchenko Taras Grigorovich	10	380660212323	2022-12-15 03:15:36	Edit	Delete
3	BM 2323 AM	Sanin Dima Albertovich	12	380787878787	2022-12-15 03:15:36	Edit	Delete
4	AX 2121 AA	Agat Albert Albertovich	30	380988342999	2022-12-15 03:15:36	Edit	Delete
5	AA 3333 AE	Shevchenko Taras Grigorovich	10	380660212323	2022-12-15 03:15:36	Edit	Delete
6	BM 2323 AM	Sanin Dima Albertovich	12	380787878787	2022-12-15 03:15:36	Edit	Delete

Рисунок 3.8 – Вигляд першої сторінки «Loading/Shipping»

На наступній сторінці «Laboratory» відображення даних таким же самим чином, різницею є тільки їх критерії.

Вкладка «Storage statistic» призначена для візуалізації даних завдяки яким користувач матиме змогу наглядно оцінити динаміки чи різницю результатів фінансової складової та об'єму матеріалів обороту. Діаграмами створюватимуться за допомогою Chart.js безкоштовної бібліотеки JavaScript з відкритим вихідним кодом.

Додання діаграм на сторінку виконується наступним спомсобом, наведеним в лістингу 3.2.

### Лістинг 3.2. – Створення елемента діаграми на сторінці

```
<script>
  const ctx = document.getElementById('myChart');
  new Chart(ctx, {
    type: 'bar',
    data: {
      labels: ['Red', 'Blue', 'Yellow', 'Green', 'Purple',
'Orange'],
      datasets: [{
        label: '# of Votes',
        data: [12, 19, 3, 5, 2, 3],
        borderWidth: 1
      }]
    },
    options: {
      scales: {
        y: {
          beginAtZero: true
        }
      }
    }
  });
</script>
```

Окрім діаграми на сторінці додано поля для вибору типу діаграми з вибором року за який вона відобразиться також користувач може вибрати по якому критерію він хоче створити діаграму ( на вибір 2 критерія: вага по місяцям чи оборот грошей ). Вигляд сторінки зображений на рисунку 3.9.

Сторінка з заказами схожа по структурі на першу сторінку та схожа за дизайном та функціоналом зі сторони відображення та редагування данимих.

Остання сторінка має назву Відео моніторніг, вона доволі просто по

структурі та функціоналу, бо має лише WebRTC-плеєри. Це веб-плеєр для відтворення Live-стрімів та трансляцій із низькою затримкою, у якого під капотом технологія WebRTC.

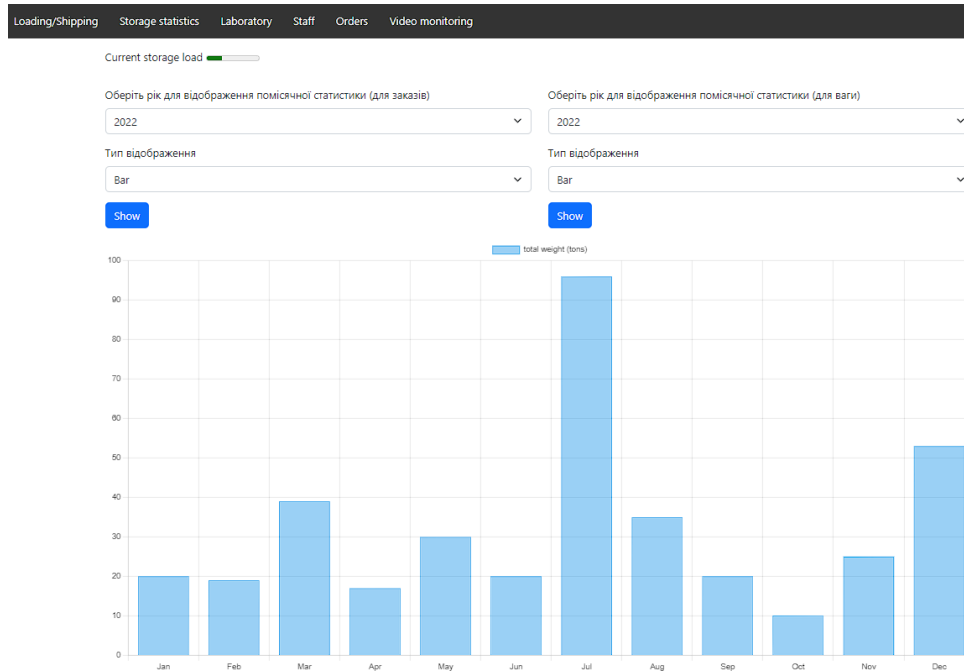


Рисунок 3.9 – Відображенні діаграми

З переваг можна підкреслити:

- може грати відео з низькою затримкою до 500 мілісекунд;
- плеєр може грати стриму з веб-камер, RTSP потоки з IP-камер, RTMP потоки;
- якщо WebRTC не підтримується у браузері, то в цьому випадку плеєр переключиться на іншу технологію, доступну в цьому браузері.

### 3.4 Тестування та оцінка надійності функціонування програмних рішень

Тестування програмного забезпечення - це процес технічного

дослідження, призначений для виявлення інформації про якість продукту відносно контексту, в якому його мають використовувати. Техніка тестування також включає як процес пошуку помилок або інших дефектів, так і випробування програмних складових із метою оцінки.

Тестування програмного забезпечення - процес перевірки відповідності заявлених до продукту вимог і реально реалізованої функціональності, який здійснюють шляхом спостереження за його роботою в штучно створених ситуаціях і на обмеженому наборі тестів, обраних певним чином.

Видів та типів тестування веб додатків існує велика кількість (зображено на рисунку 3.10), кожен із них є важливим для виявлення будь-яких недоліків або помилок роботи веб-додатків.

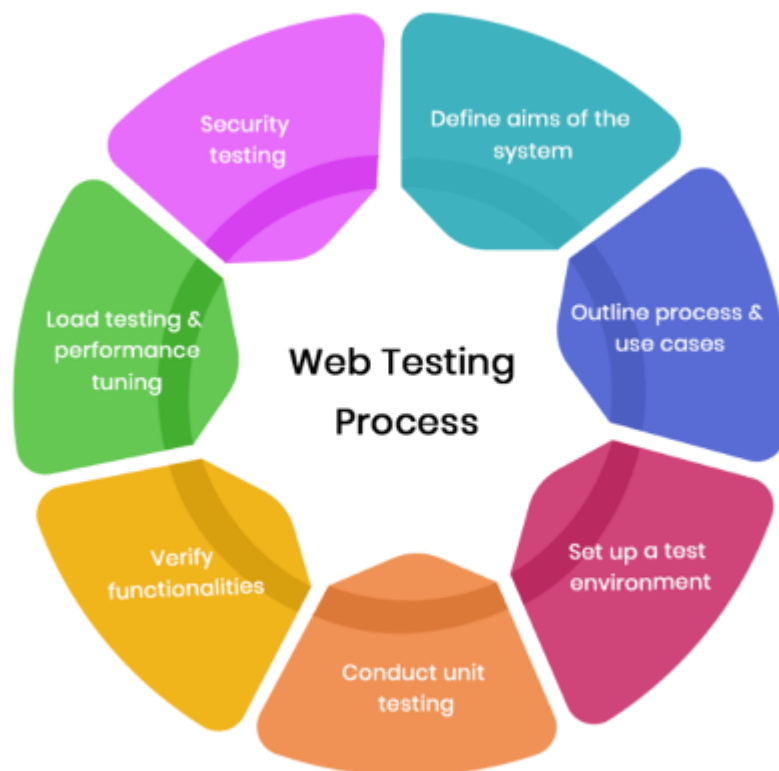


Рисунок 3.10 – Види тестування системи

Тестування безпеки – це тип тестування програмного забезпечення, який виявляє вразливі місця, загрози, ризики в програмному додатку та запобігає зловмисним атакам зловмисників. Метою тестів безпеки є

виявлення всіх можливих лазівок і слабких місць програмної системи, які можуть призвести до втрати інформації, доходу, репутації з боку співробітників або сторонніх осіб Організації [18].

Основна мета тестування безпеки – виявити загрози в системі та виміряти її потенційні вразливості, щоб можна було зустріти загрози, а система не перестала функціонувати або не могла бути використана. Це також допомагає виявити всі можливі загрози безпеці в системі та допомагає розробникам виправляти проблеми за допомогою кодування.

Існує сім основних типів тестування безпеки відповідно до посібника з методології тестування безпеки з відкритим кодом, зображені на рисунку 3.8.

Вони пояснюються наступним чином:

- сканування вразливостей, виконується за допомогою автоматизованого програмного забезпечення для сканування системи на відомі сигнатури вразливостей;

- сканування безпеки: передбачає виявлення слабких місць мережі та системи, а потім пропонує рішення для зменшення цих ризиків. Це сканування можна виконувати як для ручного, так і для автоматичного сканування;

- тестування на проникнення, імітує атаку зловмисного хакера. Це тестування передбачає аналіз конкретної системи для виявлення потенційної вразливості до спроби зовнішнього злому;

- оцінка ризиків, передбачає аналіз ризиків безпеки, які спостерігаються в організації. Ризики класифікуються як низькі, середні та високі. Це тестування рекомендує засоби контролю та заходи для зменшення ризику;

- аудит безпеки, внутрішня перевірка додатків і операційних систем на наявність недоліків у безпеці. Аудит також можна проводити шляхом построчної перевірки коду;

- етичний злам систем програмного забезпечення організації. На відміну від зловмисних хакерів, які крадуть заради власної вигоди. Метою є

виявлення недоліків безпеки в системі;

- оцінка стану, поєднує сканування безпеки, етичне хакерство та оцінку ризиків, щоб показати загальний стан безпеки організації.

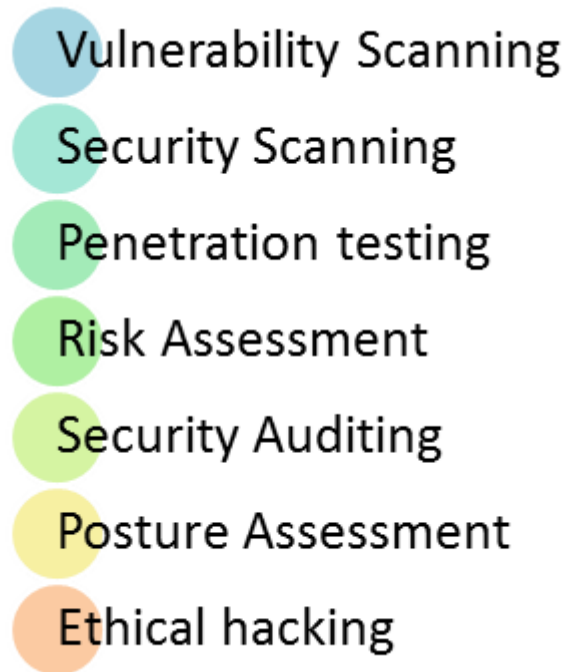


Рисунок 3.11 – Сім основних типів тестування безпеки

Тестування навантаження – це нефункціональний процес тестування програмного забезпечення, під час якого продуктивність програмного додатку перевіряється під певним очікуваним навантаженням. Він визначає, як програмне забезпечення поводить себе під час одночасного доступу кількох користувачів. Метою навантажувального тестування є покращення вузьких місць продуктивності та забезпечення стабільності та безперебійної роботи програмного додатку перед розгортанням.

Це тестування зазвичай визначає:

- максимальна продуктивність програми;
- визначте, чи достатньо поточної інфраструктури для запуску програми;

- стійкість програми щодо пікового навантаження користувача;
- кількість одночасних користувачів, яких може підтримувати програма, і масштабованість, щоб надати до неї доступ більшій кількості користувачів.

Це різновид нефункціонального тестування. У розробці програмного забезпечення навантажувальне тестування зазвичай використовується для клієнт/серверних веб-додатків – як внутрішньої мережі, так і Інтернету.

Use case тестування – це техніка тестування програмного забезпечення, яка допомагає ідентифікувати тестові випадки, які охоплюють всю систему на основі кожної транзакції від початку до кінця. Тестові випадки – це взаємодія між користувачами та програмним додатком. Тестування варіантів використання допомагає виявити прогалини в програмному застосуванні, які можуть бути не знайдені під час тестування окремих програмних компонентів.

Випадок використання під час тестування – це короткий опис конкретного використання програми актором або користувачем. Варіанти використання створюються на основі дій користувача та реакції програмного забезпечення на ці дії користувача. Він широко використовується при розробці тестових випадків на системному рівні або рівні прийняття.

Середовище тестування – це набір програмного та апаратного забезпечення для команд тестування для виконання тестів. Іншими словами, він підтримує виконання тестів із налаштованим обладнанням, програмним забезпеченням і мережею[19].

Тестовий стенд або тестове середовище налаштовано відповідно до потреб програми, що тестується. У деяких випадках тестовий стенд може являти собою комбінацію тестового середовища та тестових даних, які він використовує.

Налаштування правильного тестового середовища гарантує успіх тестування програмного забезпечення. Будь-які недоліки в цьому процесі можуть призвести до додаткових витрат і часу для клієнта.

Функціональне тестування – це тип тестування програмного забезпечення, який перевіряє систему програмного забезпечення на відповідність функціональним вимогам/специфікаціям. Метою функціональних тестів є тестування кожної функції програмного додатку шляхом надання відповідних вхідних даних і перевірки вихідних даних на відповідність функціональним вимогам.

Функціональне тестування в основному включає тестування чорного ящика, і воно не стосується вихідного коду програми. Це тестування перевіряє інтерфейс користувача, API, базу даних, безпеку, зв'язок між клієнтом і сервером та інші функції програми, що тестується. Тестування можна проводити як вручну, так і за допомогою автоматизації.

Перевірка сервісом PageSpeed Insights (рисунок 3.12) показала результати швидкості завантаження контенту та оптимізації сторінки є на достатньо високому рівні.

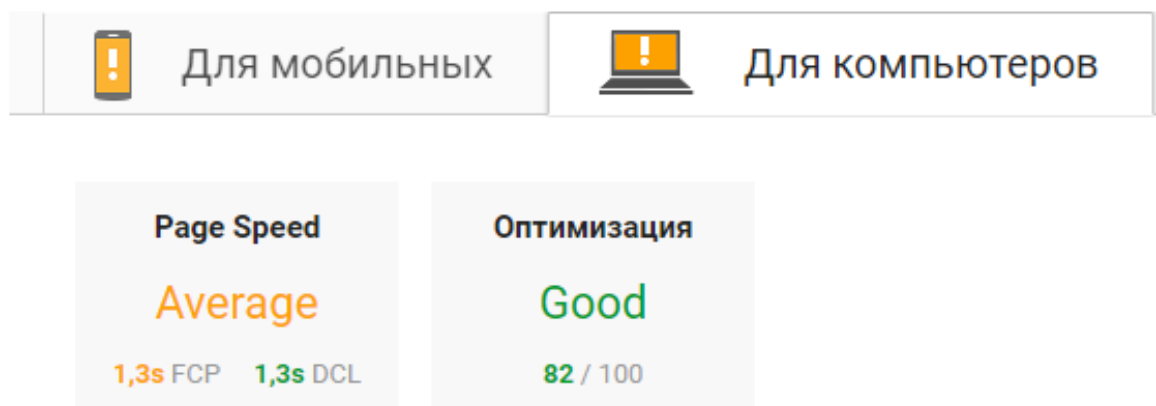


Рисунок 3.12 – Результати перевірки веб-додатку сервісом PageSpeed Insights

Тестування продуктивності (Performance Testing) – визначає наскільки тестований додаток стабільний, а також досліджує показники швидкості реакції програми на зовнішні впливи при різному за характером та інтенсивністю навантаження. Метою тестування є виявлення недоліків і

вразливостей в системі, визначення швидкості завантаження даних і їх обробки, надійності програми.

Стресове тестування (Stress Testing) – це дослідження поведінки додатка при несподіваних змінах навантаження, які перевищують розрахунковий рівень. Ця перевірка визначає як працюють програми та система в нестандартних умовах, а також дозволяє оцінити чи здатна система продовжити свою роботу після нестандартної ситуації. Стресом може бути збільшена інтенсивність виконання операцій, що досягає досить високих значень, або аварійна зміна конфігурації сервера.

Тестування сайту на уразливості. Це ключ до надійності веб-сайтів. Основні правила цього тестування – це перевірка на уразливість різних видів атак.

Кросплатформність або міжплатформність – це здатність програмного забезпечення працювати з декількома платформами або операційними системами [20]. Кросплатформність досягається за допомогою мов високого рівня програмування, середовищ розробки і виконання, що підтримують умовну компіляцію, компонування і виконання коду для різних платформ.

Таким чином результат відображення на інших пристроях буде виглядати як показано на рисунку 3.13 – сторінка з таблицею та рисунок 3.14 – сторінка з діаграмою.

XSS ін'єкцій та SQL ін'єкцій при тестуванні на уразливості не виявлено.

При проведенні різноманітних тестів якихось помилок чи недоліків не виявлено, що говорить про надійність та якість розробленого веб-додатку.

Інтерфейс розробленого веб-додатку для моніторингу процесів на елеваторному комплексі складається за декількох пунктів меню. В першому loading/shipping користувач зайшовши на сторінку побачить таблицю з відображенням інформації о поставках товару та відвантаження товару з детальною інформацією. Відображені будуть ті данні які зараз знаходяться в базі даних, далі користувач має змогу відредагувати чи видалити вже

виведені на сторінку дані або просто створити новий запис.

Схожа ситуація з пунктом лабораторії, будуть відображені інформативні дані стосовно аналізів на шкідників, вологу та домішки, які так само користувач може змінити, видалити чи створити нові данні та вести їх до бази.

### Загрузка Вигрузка

Додати інформацію

ID	Car Number	Name Surname Patronymic	Cargo(t.)	Phone number	Date and time	
1	AX 2121 AA	Agat Albert Albertovich	30	380988342999	2022-12-15 03:15:36	Edit Delete
2	AA 3333 AE	Shevchenko Taras Grigorovich	10	380660212323	2022-12-15 03:15:36	Edit Delete
3	BM 2323 AM	Sanin Dima Albertovich	12	380787878787	2022-12-15 03:15:36	Edit Delete
4	AX 2121 AA	Agat Albert Albertovich	30	380988342999	2022-12-15 03:15:36	Edit Delete
5	AA 3333 AE	Shevchenko Taras Grigorovich	10	380660212323	2022-12-15 03:15:36	Edit Delete
6	BM 2323 AA	Sanin Dima Albertovich	12	380787878787	2022-12-15 03:15:36	Edit Delete

Рисунок 3.13 – Відображення на пристроях з меншим екраном

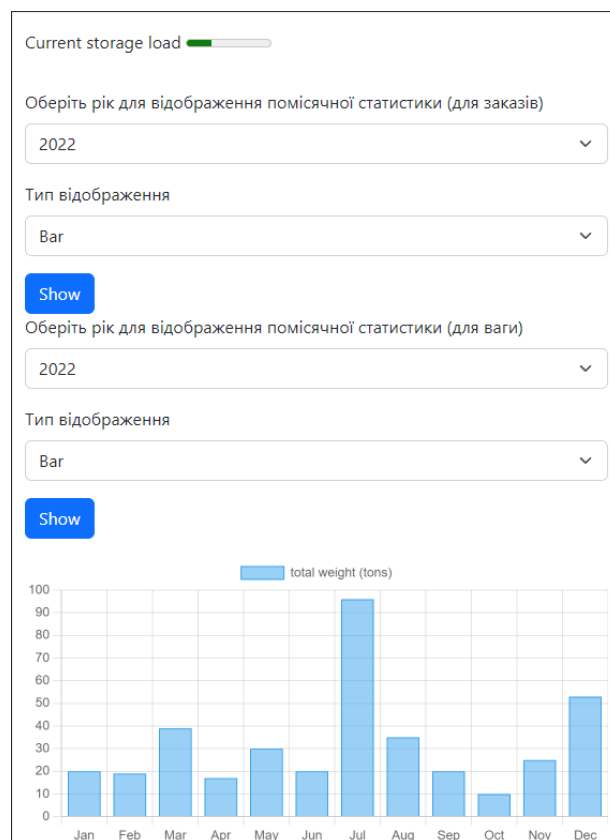


Рисунок 3.14 – Відображення на пристроях з меншим екраном діаграми

Інформація в розділі сховища буде відображена в форматі діаграми, вона може бути побудована по двом критерія кількість ваги (тон) товару (обороту) за місяць протягом року, чи так само але по кількості грошового обороту по місячно, протягом всього року. Також в верхньому куту користувач має змогу побачити поточну завантаженість.

Розділ з персоналом відображає список персоналу, при глибшому розгляданні та переході до конкретного робітника можна побачити його статус, час коли він приходить та уходить з роботи.

Сторінка з фінансами по аналогії з першою сторінкою – це інформаційна таблиця з можливістю корегувати, видаляти чи додавати данні до таблиці.

Та останній розділ має декілька WebRTC-плеєрів завдякі яким можна буде побачити становище на певних ділянках підприємства.

Таким чином виконана програмна реалізація окремих елементів функціональної моделі з розробкою структури серверної частини, алгоритмів вирішення завдань. Здійснено тестування на оцінка надійності розробленої системи.