

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки**

**Кафедра інформаційних систем та технологій**

До захисту допущено:

Завідувач кафедри

\_\_\_\_\_ Олександр РОЛІК

«\_\_» \_\_\_\_\_ 2025 р.

**Дипломний проєкт**

**на здобуття ступеня бакалавра**

**за освітньо-професійною програмою «Інформаційне забезпечення  
робототехнічних систем»**

**спеціальності 126 «Інформаційні системи та технології»**

**на тему: «Система віддаленого моніторингу та керування роботом через  
вебінтерфейс»**

Виконав:

студент IV курсу, групи ІК-12

Кардашов Юрій Олександрович

\_\_\_\_\_

Керівник:

доцент кафедри ІСТ, к.т.н., доцент,

Кравець Петро Іванович

\_\_\_\_\_

Рецензент:

доцент кафедри ІСТ, к.т.н., доцент,

Полупан Юлія Вікторівна

\_\_\_\_\_

Засвідчую, що у цьому дипломному  
проєкті немає запозичень з праць інших  
авторів без відповідних посилань.

Студент \_\_\_\_\_

Київ – 2025 року

**Національний технічний університет України**  
**«Київський політехнічний інститут імені Ігоря Сікорського»**  
**Факультет інформатики та обчислювальної техніки**  
**Кафедра інформаційних систем та технологій**

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційне забезпечення робототехнічних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Олександр РОЛІК

«\_\_» \_\_\_\_\_ 2025 р.

**ЗАВДАННЯ**

**на дипломний проєкт студенту**

**Кардашову Юрію Олександровичу**

1. Тема проєкту «Система віддаленого моніторингу та керування роботом через вебінтерфейс», керівник проєкту Кравець Петро Іванович, доцент кафедри ІСТ, к.т.н., доцент, затверджені наказом по університету від «23» травня 2025 р. № 1705-с
2. Термін подання студентом проєкту: «20» червня 2025 р.
3. Вихідні дані до проєкту: перелік доступних робіт, можливість редагування та додавання нових працівників, редагування доступних та додавання нових робіт
4. Зміст пояснювальної записки:
  1. Опис предметної області
  2. Аналіз існуючих рішень
  3. Вибір технологій та проектування системи
  4. Розроблення системи

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо): діаграма прецедентів, схема бази даних, діаграма діяльності процесу реєстрації, діаграма діяльності процесу логіну.

6. Дата видачі завдання: «17» лютого 2025 р.

#### Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Аналіз предметної області	10.05.2025	
2	Постановка завдання	12.05.2025	
3	Тестування існуючих рішень	13.05.2025	
4	Аналіз технологій для проєкту	16.05.2025	
5	Розробка системи	17.05.2025	
6	Перевірка системи	25.05.2025	
7	Створення діаграм	02.06.2025	
8	Оформлення пояснювальної записки	05.06.2025	
9	Подання ДП на захист	23.06.2025	

Студент

Юрій КАРДАШОВ

Керівник

Петро КРАВЕЦЬ

## АНОТАЦІЯ

Система віддаленого моніторингу та керування роботом через вебінтерфейс  
Проект містить 60 с. тексту, 7 рисунків, 4 таблиці, посилання на 16  
літературні джерела, додатки та 5 конструкторських документів.

СИСТЕМА МОНІТОРИНГУ, СИСТЕМА КЕРУВАННЯ, КОМП'ЮТЕРНА  
МЕРЕЖА.

Об'єктом розробки є система управління роботами .

Мета розробки — підвищення техніко-економічних показників роботи  
підприємств, де використовують роботів.

У дипломному проекті розроблено фрагменти системи керування  
роботами , а саме: верхній рівень автоматизованої системи керування роботами, а  
також підсистема моніторингу за роботами. Проведено ретельний аналіз та вибір  
системи, котра була б найбільш оптимальною для вирішення задач.

Значну увагу було приділено сполученню верхнього і нижнього рівнів системі  
управління роботами, вибору оптимальної мережі для злагодженої роботи системи.

Отримані результати можуть бути корисними при автоматизації аналогічних  
чи подібних об'єктів.

## SUMMARY

Remote Monitoring and Control System for a Robot via Web Interface

The project contains 60 pages of text, 7 figures, 4 tables, references to 16 literary sources, appendices, and 5 design documents.

Keywords: monitoring system, control system, computer network.

The object of development is a robot control system. The purpose of the development is to improve the technical and economic performance of enterprises that use robots.

The graduation project presents fragments of the robot control system, namely: the upper level of the automated robot control system and the robot monitoring subsystem. A thorough analysis and selection of a system that would be most optimal for solving the tasks was carried out.

Significant attention was paid to the integration of the upper and lower levels of the robot control system and to selecting the optimal network for the coordinated operation of the system.

The obtained results can be useful for the automation of similar or related facilities.

Номер рядка	Формат	Позначення	Найменування	Кільк. аркушів	Номер екзем.	Примітка
1			<u>Документація загальна</u>			
2						
3			Знову розроблена			
4						
5	A4	ІК12.110БАК.006 ПЗ	Пояснювальна записка	60		
6	A3	ІК12.110БАК.006 Д1	Система віддаленого	1		
7			моніторингу та керування			
8			роботом через вебінтерфейс.			
9			Діаграма прецедентів			
10	A3	ІК12.110БАК.006 Д2	Система віддаленого	1		
11			моніторингу та керування			
12			роботом через вебінтерфейс.			
13			Схема бази даних			
14	A3	ІК12.110БАК.006 Д3	Система віддаленого	1		
15			моніторингу та керування			
16			роботом через вебінтерфейс.			
17			Діаграма діяльності процесу			
18			реєстрації			
19	A3	ІК12.110БАК.006 Д4	Система віддаленого	1		
20			моніторингу та керування			
21			роботом через вебінтерфейс.			
22			Діаграма діяльності процесу			
23			логіну			
24						
25						
26						
27						
28						

**ІК12.110БАК.006 ТП**

Зм.	Аркуш	№ докум.	Підпис	Дата				
Розроб.		Кардашов Ю.О			Система віддаленого моніторингу та керування роботом через вебінтерфейс. Відомість проекту	Літ.	Аркуш	Аркушів
Керівн.		Кравець П.І.					1	1
						КПІ ім. Ігоря Сікорського		
Затв.								

**Пояснювальна записка**  
**до дипломного проєкту**  
**на тему: «Система віддаленого моніторингу та**  
**керування роботом через вебінтерфейс»**

Київ – 2025 року

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	4
ВСТУП.....	5
1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ .....	7
1.1 Особливості моніторингу та управління роботизованими системами.....	8
1.2 Типові проблеми та обмеження в управлінні та моніторингу роботизованих систем.....	11
1.3 Постановка задачі.....	13
1.3.1 Призначення системи.....	13
1.3.2 Цілі та задачі розробки.....	14
Висновок до розділу 1 .....	15
2 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ .....	16
2.1 Robot Operating System (ROS).....	16
2.2 RobotIQ Insights .....	17
2.3 Промислові IoT/SCADA рішення.....	18
2.4 Порівняльний аналіз систем .....	19
Висновок до розділу 2.....	21
3 ВИБІР ТЕХНОЛОГІЙ ТА ПРОЄКТУВАННЯ СИСТЕМИ .....	23
3.1 Архітектура системи .....	23
3.2 Клієнтська частина системи моніторингу та управління роботами .....	24
3.3 Компонування системи .....	26
3.4 Вибір мови програмування .....	27
3.5 Вибір середовища розробки.....	28
3.6 Вибір фреймворку бекенду .....	29
3.7 Вибір бази даних .....	31
Висновок до розділу 3 .....	32
4 РОЗРОБЛЕННЯ СИСТЕМИ.....	35
4.1 Структура системи .....	36

					<b>ІК12.110БАК.006 ПЗ</b>			
		№ докум.	Підпис		Система віддаленого моніторингу та керування роботом через вебінтерфейс. Пояснювальна записка	Літ.	Арк.	Аркушів
Розробив	Кардашов Ю.С			Т			2	60
Перевірив	Кравець П.І.							
Затв.								
						КПІ ім. Ігоря Сікорського Група ІК-12		

4.2 Функціональна модель системи .....	37
4.3 Модель бази даних .....	39
4.4 Передавання та обробка даних .....	40
4.5 Архітектура програмного забезпечення.....	41
4.6 Керівництво користувача.....	44
4.8 Безпека системи.....	51
4.9 Тестування системи.....	52
4.9.1 Юніт-тести .....	53
4.9.2 Ручне тестування .....	54
4.9.3 Навантажувальне тестування.....	55
Висновок до розділу 4.....	56
ВИСНОВКИ .....	58
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	60

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД – База даних.

API – Application Programming Interface – прикладний програмний інтерфейс.

					ІК12.110БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		4

## ВСТУП

У сучасну епоху стрімкого технологічного прогресу автоматизовані системи та роботизовані пристрої дедалі активніше впроваджуються в різні галузі людської діяльності — від промисловості й логістики до медицини, агросектора та побуту. Особливого значення при цьому набуває здатність забезпечити ефективне та безпечне керування роботами на відстані, а також отримувати актуальні дані про їхній технічний стан і виконувати дії в режимі реального часу. Це дозволяє не лише оптимізувати робочі процеси, а й мінімізувати участь людини в потенційно небезпечних умовах.

Зростаюча потреба в мобільності, автономності та гнучкості роботизованих систем об'єктивно зумовлює актуальність досліджень, спрямованих на розробку програмно-апаратних комплексів для моніторингу й дистанційного управління роботами. У низці випадків стандартні рішення не задовольняють специфічні вимоги користувачів або мають обмежені можливості щодо адаптації, масштабування чи інтеграції з іншими системами. Враховуючи це, виникає потреба у створенні індивідуалізованих рішень, орієнтованих на конкретні задачі, які мають бути виконані роботизованою платформою.

Метою даного дипломного проєкту є розробка інформаційної системи, яка дозволяє здійснювати моніторинг параметрів роботи автономного робота та керувати його функціональністю на відстані за допомогою зручного інтерфейсу. Передбачається, що така система забезпечить збирання, обробку та візуалізацію даних із сенсорів, а також дозволить формувати команди для віддаленого управління рухом, виконанням завдань або зміною режимів роботи.

У процесі реалізації проєкту вирішуються такі основні завдання:

- розробка архітектури системи з урахуванням модульного принципу;
- вибір апаратної платформи для роботи сенсорів та виконавчих механізмів;
- створення програмного забезпечення для обміну даними між користувачем і роботом;
- тестування системи на предмет її надійності, стабільності та швидкодії.

					ІК12.110БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		5

– тестування працездатності та надійності системи.

У рамках дипломного проєкту розроблено інформаційну систему, яка реалізує механізми моніторингу та дистанційного керування роботом за допомогою веб-інтерфейсу. Система побудована на базі технологій .NET Framework із використанням Master Page як шаблону для формування уніфікованої структури сторінок. Такий підхід дозволяє забезпечити зручну навігацію, узгоджений дизайн та ефективно повторне використання коду в межах застосування.

Програмна реалізація включає серверну частину, що обробляє запити користувачів, взаємодіє з базою даних, формує відповіді у форматі, придатному для відображення у браузері, та клієнтську частину – адаптивну вебсторінку, яка забезпечує виведення даних про стан роботи.

Дипломний проєкт складається з наступних розділів: вступ, основні розділи, висновки, список використаних джерел із 15 найменувань, додатки. Графічна частина включає 7 рисунків, 2 таблиці, 4 конструкторські документи. Загальний обсяг пояснювальної записки складає 60 сторінок.

					ІК12.110БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		6

# 1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

Розвиток робототехнічних технологій протягом останніх десятиліть суттєво вплинув на формування нової техногенної реальності, у якій роботи дедалі частіше виконують завдання, що традиційно потребували участі людини. У різних сферах діяльності — від промисловості та логістики до медицини, сільського господарства й побуту — спостерігається зростання потреби у впровадженні автономних, адаптивних та надійних роботизованих систем. У зв'язку з цим питання ефективного контролю, моніторингу стану та управління роботами набуває першочергового значення.

Сучасна система управління роботами — це багатокomпонентна структура, яка виконує функції координації, аналітики, зворотного зв'язку та взаємодії між окремими роботизованими пристроями. Основними завданнями таких систем є збір телеметричних даних у реальному часі, виявлення відхилень у роботі, прогнозування збоїв, розподіл ресурсів, планування задач та оперативне переналаштування відповідно до змін навколишнього середовища або поставлених цілей. Таким чином, система виконує роль центрального «мозку», який забезпечує синхронну та ефективну взаємодію між усіма елементами автоматизованої інфраструктури.

У багатьох існуючих рішеннях процес керування роботами або частково автоматизований, або обмежений використанням ізольованих модулів, що не дозволяє досягти високого рівня інтеграції та масштабованості. Застарілі підходи призводять до фрагментації даних, зниження ефективності використання обладнання та ускладнюють оперативне втручання у випадку нештатних ситуацій. Саме тому необхідним є створення єдиної інформаційної системи моніторингу та управління, здатної об'єднати всі компоненти в одну цифрову екосистему.

У багатьох існуючих рішеннях процес керування роботами або частково автоматизований, або обмежений використанням ізольованих модулів, що не дозволяє досягти високого рівня інтеграції та масштабованості. Застарілі підходи призводять до фрагментації даних, зниження ефективності використання

					ІК12.110БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		7

обладнання та ускладнюють оперативне втручання у випадку нештатних ситуацій. Саме тому необхідним є створення єдиної інформаційної системи моніторингу та управління, здатної об'єднати всі компоненти в одну цифрову екосистему.

У багатьох існуючих рішеннях процес керування роботами або частково автоматизований, або обмежений використанням ізольованих модулів, що не дозволяє досягти високого рівня інтеграції та масштабованості. Застарілі підходи призводять до фрагментації даних, зниження ефективності використання обладнання та ускладнюють оперативне втручання у випадку нештатних ситуацій. Саме тому необхідним є створення єдиної інформаційної системи моніторингу та управління, здатної об'єднати всі компоненти в одну цифрову екосистему.

Із розвитком технологій штучного інтелекту, машинного навчання та IoT (Інтернету речей), системи управління роботами набувають нових можливостей: адаптивна поведінка, самонавчання, розпізнавання об'єктів, прогнозування технічних несправностей — усе це значно розширює сферу їх застосування. При цьому підвищується потреба у високій обчислювальній потужності, захисті інформації та забезпеченні стійкості до зовнішніх впливів, включаючи кібератаки.

Отже, система моніторингу та управління роботами є не лише допоміжним елементом, а фактично основою для побудови повноцінного роботизованого середовища. Її наявність дозволяє досягти високої точності, безперервності та керованості процесів, знижує витрати, мінімізує людський фактор і забезпечує гнучке масштабування відповідно до змін умов роботи. У сучасних умовах саме такі системи стають запорукою ефективного використання роботизованих рішень у реальному секторі економіки.

### 1.1 Особливості моніторингу та управління роботизованими системами

Функціонування сучасних роботизованих систем неможливо уявити без впровадження ефективних механізмів моніторингу та управління. Ці процеси забезпечують не лише безперебійну роботу окремих пристроїв, а й узгоджену взаємодію між елементами складних технічних комплексів. Система управління

					ІК12.110БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		8

роботами виконує ключову роль у забезпеченні продуктивності, точності виконання завдань, адаптації до змін зовнішніх умов, а також у підтримці стабільного функціонування в умовах багатозадачності та розподіленого середовища.

Типова система управління роботами включає в себе кілька основних підсистем: модуль керування завданнями, систему збору даних у реальному часі, аналітичний модуль, а також інтерфейс взаємодії з оператором. Для повноцінної роботи така система має враховувати типи роботів, специфіку виконуваних операцій (наприклад, монтаж, транспортування, маніпуляції з об'єктами), а також технічні обмеження — діапазон дій, наявність сенсорних даних, енергоспоживання тощо.

Особливу складність в управлінні роботизованими комплексами становить координація дій у багатокомпонентному середовищі. Приклади таких середовищ — виробничі лінії з кількома роботами, склади з автономними транспортними платформами або служби доставки з використанням дронів. В усіх цих випадках система управління повинна забезпечувати уникнення конфліктів у маршрутах, динамічний перерозподіл ресурсів, пріоритезацію завдань та своєчасне реагування на зміну умов.

Ще одним критично важливим аспектом є система моніторингу. Вона включає постійний збір телеметричних даних про стан обладнання: температура, вібрації, навантаження, рівень заряду, положення в просторі тощо. Такі дані дозволяють не лише оперативно виявляти збої, а й проводити превентивну діагностику — прогнозувати ймовірність виходу з ладу та планувати технічне обслуговування. Збір і аналіз подібної інформації в режимі реального часу дозволяє зменшити простої, уникати аварійних ситуацій і загалом підвищити ефективність експлуатації робототехнічних систем.

Не менш важливою є функція протоколювання та аналітики. Сучасні системи мають підтримувати ведення журналів подій, автоматичне створення звітів про виконані дії, виявлення відхилень від стандартних сценаріїв. Це дозволяє здійснювати аудит дій роботів, а також оцінювати їхню ефективність на основі

кількісних показників. У промислових умовах такі дані є основою для прийняття управлінських рішень щодо оптимізації процесів.

Суттєвою особливістю сучасних систем є також здатність до масштабування та адаптації. В умовах зростаючих вимог і швидкої модернізації обладнання, система управління повинна мати відкриту архітектуру та підтримувати розширення функціональності: додавання нових типів роботів, підключення зовнішніх сенсорів, інтеграцію з іншими цифровими платформами (ERP, MES, SCADA тощо). Окрім того, надзвичайно актуальним є забезпечення безпеки: як фізичної (через контроль зон доступу та аварійне зупинення), так і інформаційної — захисту каналів зв'язку, автентифікації користувачів та резервування даних.

Ще одна характерна риса сучасних систем — врахування «розумної поведінки» роботів. Усе більше систем використовують елементи машинного навчання, що дозволяє роботам адаптуватися до змін у середовищі, навчатися на основі історичних даних і вдосконалювати алгоритми виконання завдань. Для підтримки таких функцій інформаційна система повинна забезпечувати обмін даними з нейронними мережами, мати ресурси для обчислень і підтримувати відповідні інтерфейси.

Таким чином, система моніторингу та управління роботами є складною інтелектуальною платформою, що інтегрує в собі механізми контролю, аналізу, прогнозування та взаємодії. Її ефективне функціонування дозволяє досягати високої надійності, продуктивності та адаптивності роботизованих систем у різних сферах діяльності — від виробництва до медицини, транспорту чи сфери обслуговування. Впровадження таких систем є необхідною умовою для подальшого розвитку автоматизації та створення «розумного» середовища нового покоління.

					ІК12.110БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		10

## 1.2 Типові проблеми та обмеження в управлінні та моніторингу роботизованих систем

На сучасному етапі розвитку автоматизованих технологій, ефективно управління та моніторинг роботизованих систем стикається з низкою суттєвих труднощів. Однією з основних проблем є відсутність уніфікованих рішень, які б дозволяли централізовано керувати різнорідними роботами в межах однієї інформаційної інфраструктури. У багатьох випадках кожна система функціонує автономно, з використанням власних протоколів, інтерфейсів та форматів даних, що призводить до фрагментації управління, утруднює інтеграцію та значно ускладнює створення єдиного цифрового простору для роботизованої екосистеми.

Ще однією типовою проблемою є обмежена масштабованість існуючих систем керування. У разі збільшення кількості роботів, розширення функціоналу або змін у виробничому процесі, багато систем виявляються неспроможними адаптуватися без суттєвих змін архітектури або навіть повного переосмислення логіки управління. Це призводить до затримок у впровадженні, підвищених витрат на модернізацію та ризику збоїв при спробах масштабування.

Особливої уваги потребує питання реального часу. Сучасні роботизовані комплекси потребують синхронного обміну даними між окремими елементами, включаючи сенсори, виконавчі механізми та аналітичні модулі. В умовах затримок або нестабільного з'єднання виникають збої в роботі — від втрати точності позиціонування до критичних аварій. У багатьох випадках такі ситуації викликані недостатньою пропускнуою здатністю мереж, застарілим апаратним забезпеченням або неефективними алгоритмами обробки даних.

Ще одним обмеженням є недостатній рівень автономності більшості систем. Хоча деякі роботи здатні виконувати окремі завдання самостійно, у складніших сценаріях вони вимагають постійного втручання оператора або ручного налаштування. Це обмежує продуктивність, підвищує навантаження на персонал і знижує загальну ефективність. Причиною цього часто є недостатня гнучкість

					ІК12.110БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		11

програмного забезпечення або слабка інтеграція інтелектуальних модулів, зокрема систем штучного інтелекту чи машинного навчання.

Варто також згадати про проблему технічної сумісності між роботами різних виробників. У більшості випадків відсутність єдиних стандартів унеможливорює пряме об'єднання таких пристроїв в єдину мережу без додаткових засобів трансляції команд або адаптації протоколів. Це ускладнює розгортання багатокомпонентних систем, підвищує витрати на налаштування і знижує надійність у процесі експлуатації.

Вразливості в програмному забезпеченні, відсутність належного шифрування або слабка автентифікація можуть призвести до несанкціонованого доступу, втрати даних чи навіть прямого впливу на фізичні дії роботизованих пристроїв. Захист таких систем потребує багаторівневої стратегії, яка включає безпеку мережі, логування подій, контроль доступу, а також постійне оновлення програмного забезпечення.

Окремою складністю є підготовка та кваліфікація персоналу. Ефективна експлуатація роботизованих систем вимагає від операторів не лише знань з технічного обслуговування, а й розуміння логіки керування, основ програмування, роботи з інтерфейсами збору та аналізу даних. У багатьох організаціях відсутність відповідної підготовки призводить до неефективного використання можливостей системи або помилок, які можуть спричинити порушення роботи.

Проблеми управління та моніторингу роботизованих систем охоплюють як технічні, так і організаційні аспекти. Їх подолання вимагає застосування комплексного підходу — від стандартизації протоколів до впровадження систем глибокої аналітики, від забезпечення кібербезпеки до розвитку компетенцій персоналу. Тільки за умов усунення вказаних обмежень можливо забезпечити ефективну, надійну та масштабовану роботизовану інфраструктуру.

					ІК12.110БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		12

### 1.3 Постановка задачі

#### 1.3.1 Призначення системи

Проектована система моніторингу та управління роботами призначена для автоматизації керування роботизованими платформами у межах виробничих, логістичних або наукових середовищ. Основними функціональними напрямками автоматизації є:

- централізоване управління діями автономних та напівавтономних роботів;
- виявлення та оповіщення про відхилення від штатного функціонування;
- адміністрування прав доступу операторів, технічного персоналу та системних адміністраторів;
- збір, обробка та візуалізація телеметричних даних з датчиків у режимі реального часу.

Основна мета впровадження такої системи — зниження ризику помилок при взаємодії з роботизованим обладнанням, підвищення продуктивності праці шляхом автоматизації рутинних дій, а також забезпечення оперативного реагування на позаштатні ситуації. Система має забезпечувати ефективну взаємодію між різними модулями робототехнічної інфраструктури та підтримувати масштабування без потреби в повному переналаштуванні.

Ключовою характеристикою розроблюваного рішення є його здатність інтегрувати дані з різнорідних джерел — сенсорних систем, камер спостереження, модулів навігації — у єдиному інформаційному полі. Такий підхід дає змогу здійснювати комплексний моніторинг стану роботизованих одиниць, прогнозувати технічні несправності на основі історичних даних, а також забезпечувати аналітичну підтримку рішень щодо планування навантаження.

Особлива увага приділяється модульності системи, що дозволяє інтегрувати нові типи роботів, алгоритмів або зовнішніх сервісів (наприклад, платформи віддаленого навчання роботів, системи симуляцій або зовнішні ERP/SCADA-компоненти). Завдяки цьому досягається гнучкість в експлуатації та можливість

					ІК12.110БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		13

адаптації системи до різних сценаріїв використання — від виробничих ліній до автономних дронів у польових умовах.

Таким чином, система спрямована на створення універсального цифрового середовища для управління роботами, що забезпечує не лише технічне керування, а й моніторинг, аналіз та прийняття управлінських рішень на основі даних, отриманих у реальному часі.

### 1.3.2 Цілі та задачі розробки

Метою створення системи моніторингу та управління роботами є забезпечення надійного керування групою роботизованих пристроїв у реальному часі, оптимізація обміну даними між користувачем та апаратною частиною, а також автоматизація обробки телеметрії для швидкого прийняття рішень.

Для реалізації цієї мети розробляється інформаційна система на основі технологій .NET Framework [1], яка дозволяє ефективно поєднувати серверну логіку з веб-інтерфейсом. У якості базової структури інтерфейсу застосовано механізм MasterPage, що забезпечує єдиний шаблон для всіх сторінок системи, а також підвищує зручність підтримки та розширення проєкту. Верстка та візуальне оформлення реалізовані з використанням фреймворку Bootstrap [2], що дозволяє досягти адаптивного дизайну і зручної навігації з різних типів пристроїв. В якості системи керування базами даних використовується Microsoft SQL Server, що забезпечує надійне зберігання, запити та обробку великих обсягів даних.

Крім того, при розробці враховується можливість подальшої інтеграції з іншими системами, наприклад, SCADA-платформами або модулями штучного інтелекту для розширення функцій автоматичного аналізу даних.

Таким чином, створення системи спрямоване не лише на технічне керування роботами, а й на побудову єдиної інформаційної екосистеми, яка дозволить підвищити ефективність, безпеку та надійність роботизованих процесів у сучасному виробничому або експериментальному середовищі.

					ІК12.110БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		14

## Висновок до розділу 1

У цьому розділі було детально описано предметну область дослідження, розглянуто її основні характеристики й специфіку, а також окреслено ключові процеси діяльності в освітньому закладі, які підлягають автоматизації. Проведено аналіз існуючих проблем в організації навчального процесу й обґрунтовано необхідність створення інформаційної системи для його ефективного управління. На основі проведеного аналізу сформульовано постановку задачі дипломного проєкту, визначено основні вимоги до системи й окреслено шляхи їх реалізації.

Розробка інформаційної системи управління навчальним процесом дозволить суттєво підвищити ефективність управління освітніми процесами, автоматизувати рутинні операції, знизити вплив людського чинника та забезпечити прозорість прийняття управлінських рішень. Система створюється з урахуванням потреб усіх учасників освітнього процесу — від адміністрації та викладачів до студентів, забезпечуючи їм зручний і безпечний доступ до актуальної інформації. Важливою перевагою є можливість подальшого розширення функціоналу системи відповідно до потреб конкретного закладу освіти, що робить її гнучкою, масштабованою та придатною для використання в закладах різного рівня — від загальноосвітніх шкіл до вищих навчальних закладів.

Таким чином, створення й впровадження такої системи є актуальним і перспективним завданням, що сприятиме підвищенню якості освітнього процесу, покращенню організації роботи навчального закладу та загальному розвитку цифрового середовища в освіті.

					ІК12.110БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		15

## 2 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

У контексті стрімкого розвитку автоматизації та індустрії 4.0, ефективне управління роботизованими системами відіграє ключову роль у забезпеченні безперервності, точності та безпеки виробничих і логістичних процесів. Різноманітні системи моніторингу та управління роботами (Robot Management and Monitoring Systems, RMMS) дозволяють координувати дії роботів, здійснювати діагностику у реальному часі, прогнозувати несправності, забезпечувати збір статистичних даних та керувати задачами з урахуванням зовнішніх і внутрішніх змін. Найбільш поширеними прикладами таких платформ є ROS (Robot Operating System), RobotIQ Insights, а також корпоративні рішення на базі SCADA/ІоТ технологій.

### 2.1 Robot Operating System (ROS)

Robot Operating System (ROS) є відкритим середовищем, що надає набір бібліотек та інструментів для розробки програмного забезпечення керування роботами. Основною перевагою ROS є його модульність, завдяки якій розробники можуть створювати гнучкі системи керування для найрізноманітніших типів роботів — від мобільних платформ до промислових маніпуляторів [15]. Функціональні можливості ROS включають:

- обробку сенсорних даних (LiDAR, камери, ІМУ);
- планування траєкторій та навігацію;
- виявлення та уникнення перешкод;
- підтримку розподіленого керування та багаторівневої взаємодії між модулями.

ROS є універсальною платформою для досліджень і прототипування, проте вона не має вбудованих засобів для масового моніторингу або централізованого адміністрування роботів у масштабах підприємства. Для цього розробникам



Скріншот із програми RobotIQ Insights зображено на рисунку 2.2

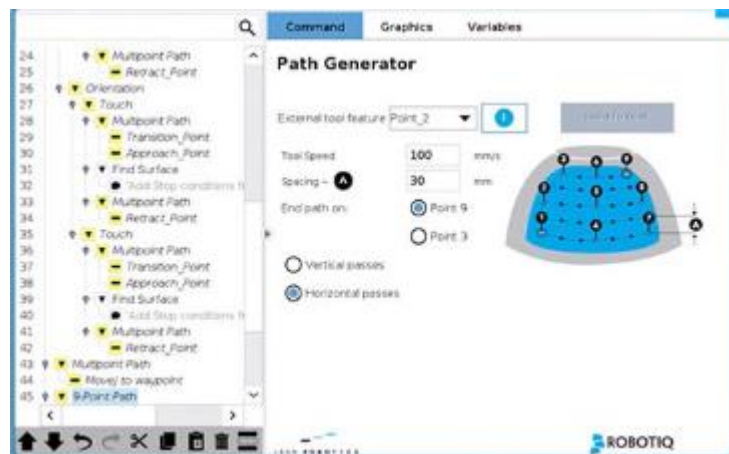


Рисунок 2.2 – Скріншот RobotIQ Insights

### 2.3 Промислові IoT/SCADA рішення

Багато сучасних підприємств використовують системи моніторингу та керування роботами, інтегровані в загальну інфраструктуру SCADA (Supervisory Control and Data Acquisition) або IoT-платформи. Прикладом таких систем є Siemens MindSphere, Bosch Nexeed, Schneider EcoStruxure. Ключові особливості цих систем:

- високий рівень інтеграції з датчиками, контролерами, ERP-системами;
- можливість побудови цифрових двійників та прогнозу аналітики;
- масштабованість для роботи з десятками чи сотнями пристроїв;
- підтримка кібербезпеки, резервного копіювання та адміністрування користувачів.

Скріншот із промислової програми зображено на рисунку 2.3



Рисунок 2.3 – Скріншот SCADA-інтерфейсу контролю стану виробничої лінії з роботами

## 2.4 Порівняльний аналіз систем

З метою об'єктивного оцінювання ефективності різних систем моніторингу та управління роботами, проведено порівняльний аналіз за основними критеріями.

Таблиця 2.1 – Порівняльний аналіз систем

Критерій	ROS	RobotIQ Insights	SCADA/ІoT системи
Тип ліцензії	Відкрита	Комерційна	Комерційна
Гнучкість налаштувань	Висока	Середня	Висока
Масштабованість	Обмежена	Середня	Висока
Вимоги до технічного персоналу	Високі	Низькі	Високі

Зм.	Лист	№ докум.	Підпис	Дата

Критерій	ROS	RobotIQ Insights	SCADA/ІoT системи
Моніторинг у реальному часі	Обмежено (через додатки)	Так	Так
Підтримка аналітики	Через сторонні модулі	Вбудована	Вбудована

ROS є одним із найпотужніших рішень з відкритим кодом для управління роботами, що забезпечує гнучкість конфігурацій, підтримку великої кількості датчиків, контролерів та виконавчих пристроїв. Однак, через складність у налаштуванні та потребу в глибоких технічних знаннях, ROS більше підходить для дослідницьких проєктів, стартапів і лабораторій, ніж для промислових підприємств.

Основною перевагою ROS є його модульна архітектура, завдяки якій можна створити практично будь-яку конфігурацію системи. Проте відсутність нативних засобів аналітики й обмежений рівень безпеки ставлять під сумнів його застосування у критично важливих середовищах без додаткових модулів.

RobotIQ Insights комерційна платформа, розроблена спеціально для підтримки колаборативних роботів (CoBots). Система вирізняється інтуїтивним інтерфейсом та інтегрованими інструментами візуалізації, які не потребують програмування з боку кінцевого користувача. Проте гнучкість налаштувань обмежена, а сумісність із апаратними рішеннями переважно зосереджена на продуктах RobotIQ і Universal Robots.

Insights добре підходить для середніх підприємств, де важливо швидко впровадити рішення без значних витрат часу на конфігурацію. Її слабкою стороною залишається порівняно невисокий рівень кастомізації та потреба у щомісячній підписці.

SCADA/ІoT системи вже давно використовуються у промисловості для віддаленого керування, моніторингу та автоматизації. Серед найпопулярніших — Ignition, Siemens WinCC, Wonderware, AVEVA, які забезпечують широку підтримку протоколів, резервування даних, історизацію та зведену аналітику.

Перевагою SCADA-рішень є висока надійність і масштабованість, що робить їх придатними для управління великими парками роботизованих систем або гібридних систем (роботи + інші пристрої). Проте складність налаштування та висока вартість ліцензування можуть бути стримуючим чинником для невеликих команд або навчальних установ.

## Висновок до розділу 2

На підставі аналізу трьох найпоширеніших типів систем моніторингу та управління роботами можна зробити висновок, що кожна з них має свої унікальні переваги і обмеження, які зумовлюють доцільність використання тієї чи іншої системи залежно від конкретного контексту. Так, ROS (Robot Operating System) забезпечує високу гнучкість, розширюваність та підтримку широкого спектра апаратних рішень завдяки відкритому вихідному коду. Вона ідеально підходить для дослідницьких, прототипувальних і освітніх проєктів, але її ефективне впровадження вимагає високої кваліфікації розробників та належної технічної підготовки.

RobotIQ Insights, зі свого боку, пропонує зручний користувацький інтерфейс, просту інтеграцію з популярними колаборативними роботами та вбудовані аналітичні функції. Це робить її привабливим варіантом для малих і середніх виробничих підприємств, які прагнуть швидкого впровадження без глибоких налаштувань. Водночас, обмеженість у кастомізації та залежність від вузького спектра обладнання знижують її універсальність у більш складних і динамічних виробничих середовищах.

SCADA/IoT-системи являють собою зрілі промислові рішення, здатні підтримувати масштабовані архітектури, централізоване управління, глибоку інтеграцію з іншими IT-інфраструктурами підприємства (ERP, MES тощо), а також високий рівень безпеки й надійності. Вони є незамінними на великих виробництвах з великою кількістю датчиків, роботів і контролерів. Однак значні витрати на

					ІК12.110БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		21

ліцензування, технічну підтримку та впровадження обмежують їх доступність для менш капіталомістких організацій.

Таким чином, проведений аналіз демонструє відсутність універсального рішення, здатного задовольнити одночасно вимоги до простоти, гнучкості, безпеки, масштабованості та інтеграції. Це створює обґрунтовану потребу у розробці нової адаптивної платформи управління роботизованими системами, яка б поєднувала найкращі риси наявних підходів: відкритість та розширюваність ROS, простоту і доступність RobotIQ, надійність та промислову інтеграцію SCADA-рішень. Така система повинна мати модульну архітектуру, підтримку стандартних протоколів взаємодії, масштабовану структуру обробки даних, а також орієнтацію на користувача з різним рівнем технічної підготовки. У підсумку це дозволить забезпечити ефективне управління роботизованими комплексами в умовах цифрової трансформації промисловості та автоматизації виробничих процесів.

					ІК12.110БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		22

### 3 ВИБІР ТЕХНОЛОГІЙ ТА ПРОЄКТУВАННЯ СИСТЕМИ

Проєктування системи моніторингу та управління роботизованими пристроями вимагає зваженого підходу до вибору технологій, архітектури та середовищ розробки. У даному розділі обґрунтовано вибір мови програмування, фреймворків, бази даних, підходів до організації користувацького інтерфейсу та серверної логіки.

#### 3.1 Архітектура системи

Система моніторингу та керування роботизованими пристроями реалізована відповідно до модульної багаторівневої архітектури, що забезпечує логічний поділ відповідальностей між основними компонентами. Така структура дозволяє спростити розробку, масштабування та підтримку програмного забезпечення, а також адаптацію до змін у функціональних вимогах. Архітектура системи передбачає три ключові рівні.

Рівень бізнес-логіки включає класи, що описують поведінку роботів, правила обробки телеметричних даних, формування команд на виконання операцій, а також взаємодію з апаратними інтерфейсами. Тут закладено основну функціональність, пов'язану з плануванням, контролем та аналізом дій роботизованих платформ.

Рівень представлення (інтерфейс користувача) реалізований за допомогою Web Forms з використанням MasterPage для створення уніфікованого макета сторінок. Інтерфейс побудований на основі Bootstrap, що забезпечує адаптивну верстку та зручну взаємодію з елементами керування — панелями моніторингу, таблицями з результатами, графіками роботи систем тощо. Вебінтерфейс дозволяє операторам переглядати статус роботів, ініціювати команди, переглядати журнал подій та отримувати сповіщення [5].

Рівень доступу до даних заснований на використанні SQL Server. Усі дані про стан системи, історію дій, журнали подій, параметри роботи обладнання та результати аналізу зберігаються у централізованій реляційній базі. Доступ до БД

					ІК12.110БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		23

реалізовано через типізовані DataSet та ADO.NET, з чітким відокремленням логіки запитів від користувацького інтерфейсу [6].

Використання .NET Framework як основи для серверної логіки забезпечує сумісність із існуючою інфраструктурою Windows, а також стабільну інтеграцію з службами IIS, механізмами автентифікації Active Directory та системами управління подіями [13].

Поділ коду відповідно до функціональних блоків дозволяє реалізувати принцип єдиної відповідальності для кожного модуля. Наприклад, компоненти, що відповідають за зберігання даних, не залежать від інтерфейсу або сценаріїв взаємодії з користувачем. Це підвищує гнучкість та дозволяє проводити модульне тестування окремих частин системи без необхідності їх глибокого переписування.

Також передбачено можливість підключення зовнішніх сенсорних або програмних систем — для цього реалізовані API-інтерфейси обміну даними. Таке рішення дозволяє використовувати систему в умовах промислових або дослідницьких середовищ, де необхідна інтеграція з іншими модулями автоматизації.

Загалом, архітектура побудована таким чином, щоб бути максимально адаптивною до змін, придатною до розширення та сумісною з сучасними вимогами до систем моніторингу в реальному часі.

### 3.2 Клієнтська частина системи моніторингу та управління роботами

Для реалізації клієнтської частини системи моніторингу та управління роботизованими комплексами було використано функціональні можливості ASP.NET Web Forms у поєднанні з технологією MasterPage, яка забезпечує уніфікований шаблон для всіх сторінок вебзастосунку. Це рішення дозволяє централізовано управляти зовнішнім виглядом інтерфейсу та структурою сторінок, значно спрощуючи підтримку та розширення інтерфейсу користувача. У контексті системи керування роботами це особливо важливо для забезпечення єдиного

стандарту відображення даних, журналів, статусів обладнання та панелей керування [4].

Для побудови адаптивного і зручного у використанні інтерфейсу було інтегровано Bootstrap — сучасну фронтенд-бібліотеку, яка дозволяє швидко створювати вебінтерфейси, що коректно працюють на різних розмірах екранів і пристроях. Завдяки використанню Bootstrap вдалося реалізувати інтуїтивно зрозумілі компоненти, такі як динамічні таблиці станів, графіки навантаження, панелі індикаторів та форми взаємодії з користувачем. Вбудована підтримка сітки та елементів управління забезпечила швидку верстку сторінок, зберігаючи консистентність стилів по всій системі.

Формування динамічного вмісту відбувається з використанням серверних контролів ASP.NET у поєднанні з AJAX-технологіями, що дозволяє оновлювати частини сторінки без повного перезавантаження. Це особливо актуально для модулів, які відображають поточні дані з роботів у режимі реального часу — наприклад, телеметрію, журнал подій, повідомлення про несправності або сигнали тривоги. Внаслідок цього користувачі отримують швидкий та інтерактивний інтерфейс без перевантаження серверних ресурсів.

На стороні сервера реалізовано обробку даних через .NET Framework, що забезпечує стабільність, широкі можливості інтеграції з внутрішніми бібліотеками, а також підтримку складної бізнес-логіки системи керування. Вся інформація зберігається у SQL Server, який виступає в ролі основної СУБД. Це дозволяє організувати ефективну роботу з великим обсягом даних, включаючи зберігання історичних записів про активність роботів, облік команд управління, логування змін стану та аналітичні дані.

Таким чином, поєднання .NET Framework, MasterPage, Bootstrap та SQL Server дозволило створити функціональний, адаптивний і масштабований вебінтерфейс для системи моніторингу робототехнічних пристроїв. Такий підхід забезпечує зручність у використанні, простоту розширення та швидке відображення актуальної інформації, що є критично важливим для надійної

експлуатації роботизованих систем у промисловому або лабораторному середовищі.

### 3.3 Компонування системи

Система моніторингу та управління робототехнічними пристроями проєктується як веборієнтований багаторівневий застосунок, побудований на основі архітектурних принципів розподілення відповідальностей між основними компонентами. Такий підхід дозволяє досягти високої масштабованості, зручності обслуговування, ефективної організації коду та підвищення надійності під час експлуатації системи. Архітектурна модель розділена на кілька логічних рівнів.

Презентаційний рівень, реалізований із використанням ASP.NET Web Forms і технології MasterPage, відповідає за побудову інтерфейсу користувача. Це дозволяє зручно розміщувати шаблонні елементи (меню, хедери, футери) для всіх сторінок, зменшуючи дублювання коду та полегшуючи підтримку дизайну.

Логічний рівень обробляє запити користувачів, виконує бізнес-операції, валідацію та формування відповідей. Основна логіка зосереджена у сервісних класах, що ізольовано обробляють окремі функції (реєстрація, контроль стану робіт, керування сценаріями дій тощо).

Рівень доступу до даних працює через ADO.NET або Entity Framework (залежно від підсистеми), забезпечуючи ефективну взаємодію з SQL Server. Уся структурована інформація, включно з логами, параметрами роботи, планами дій робіт, зберігається в реляційній базі з підтримкою транзакцій та складних запитів [11].

Комунікація між інтерфейсом і серверною логікою здійснюється через Web Forms-контролі та AJAX-запити, які дозволяють оновлювати частини сторінки без перезавантаження. Це особливо важливо для відображення телеметрії в реальному часі або змін в оперативному статусі пристроїв. Система також підтримує механізми автентифікації та авторизації користувачів через стандартні можливості

					ІК12.110БАК.006 ПЗ	Арк.
						26
Зм.	Лист	№ докум.	Підпис	Дата		

.NET Forms Authentication, з розподілом прав доступу до функціоналу залежно від ролі (адміністратор, оператор, технік).

Дані, які надходять від клієнта (наприклад, команди для роботів або параметри конфігурації), проходять перевірку як на стороні браузера (через JavaScript-валідацію), так і на сервері — це дозволяє мінімізувати помилки та убезпечити систему від некоректного вводу. Вся логіка валідації розміщена централізовано у відповідних модулях логічного рівня.

Для зручного відображення й адаптивного дизайну застосовано Bootstrap, який забезпечує коректну роботу інтерфейсу на різних пристроях — від десктопів до смартфонів. Це дозволяє операторам системи моніторити стан обладнання або переглядати події навіть із мобільних пристроїв у польових умовах.

Таким чином, система побудована за принципами чіткої модульності: інтерфейс, логіка обробки, та зберігання даних функціонують незалежно, що спрощує тестування, модернізацію та подальше розширення функціоналу. Завдяки використанню .NET Framework, MasterPage, Bootstrap та SQL Server, реалізовано гнучке, надійне рішення для інтеграції роботів у цифрову інфраструктуру підприємства або лабораторного комплексу.

### 3.4 Вибір мови програмування

Для реалізації серверної логіки системи було обрано мову C# у поєднанні з .NET Framework, що є потужною платформою для створення стабільних і масштабованих прикладних рішень. Вибір саме цього стеку зумовлений його глибокою інтеграцією з Windows-середовищем, широкою підтримкою корпоративних технологій, а також високою надійністю при розробці систем класу RMMS (Robot Monitoring and Management System).

Мова C# забезпечує зручне використання об'єктно-орієнтованих підходів, що особливо важливо при проектуванні модульної архітектури. В рамках системи управління роботами це дозволяє реалізувати окремі компоненти для збору

					ІК12.110БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		27

телеметрії, керування станами роботизованих вузлів, аналітики та взаємодії з користувачами у вигляді незалежних модулів [12].

Крім того, .NET Framework має потужні засоби для створення веб-застосунків із використанням Web Forms і шаблонів MasterPage, що дозволяє створити уніфікований макет для всіх сторінок інтерфейсу. Це забезпечує не тільки візуальну цілісність інтерфейсу, а й спрощує підтримку й розвиток системи в довгостроковій перспективі.

### 3.5 Вибір середовища розробки

Для створення ефективної та масштабованої системи моніторингу й управління роботизованими пристроями критично важливо правильно підібрати середовище розробки. Враховуючи специфіку задач, що включають збір даних у реальному часі, обробку подій, керування пристроями та інтерактивну візуалізацію, вибір інструментів програмування напряму впливає на якість, швидкість розробки та подальшу підтримку системи.

При розробці серверної частини було обрано Visual Studio як основне інтегроване середовище розробки (IDE). Visual Studio забезпечує глибоку підтримку .NET Framework, що є основною технологією реалізації логіки управління роботами, збору телеметричних даних та зберігання інформації у SQL Server. IDE містить усі необхідні інструменти для роботи з Windows Forms, Web Forms (включно з MasterPage-шаблонами), налагодження, створення баз даних і візуального проєктування. Завдяки підтримці відладки в реальному часі та широкій екосистемі плагінів Visual Studio дозволяє швидко створювати та тестувати серверні компоненти системи [7].

Зважаючи на повну сумісність Visual Studio з .NET Framework і SQL Server, а також наявність візуального дизайнера Web Forms та зручного конструктора баз даних, саме це середовище було обрано для основного циклу розробки серверної частини RMMS-системи [8].

Для клієнтської частини системи, яка реалізована у вигляді веб-інтерфейсу з використанням HTML/CSS, JavaScript та Bootstrap, було обрано Visual Studio Code. Цей редактор є легким, швидким, та надзвичайно гнучким завдяки великій кількості доступних розширень. Його можливості включають підтримку автодоповнення, літінгу, Git-інтеграції та зручної роботи з багатьма файлами одночасно, що є актуальним при створенні адміністративної панелі, дашбордів і форм керування [14].

Visual Studio Code також ідеально підходить для розробки фронтенду, який в даному проєкті є відповідальним за відображення поточних станів робіт, журналів подій, керування командами запуску та аварійного зупинення. Завдяки підтримці Bootstrap, вдалося реалізувати адаптивний дизайн, що коректно працює як на настільних комп'ютерах, так і на мобільних пристроях, що критично важливо для оперативного доступу інженерів і технічного персоналу [9].

Таким чином, комбінація Visual Studio для серверної частини та Visual Studio Code для клієнтського інтерфейсу забезпечила ефективний цикл розробки системи моніторингу та управління роботами. Такий вибір дозволив оптимізувати продуктивність, спростити налагодження та надати гнучкі інструменти для подальшого масштабування системи.

### 3.6 Вибір фреймворку бекенду

Розробка серверної частини системи моніторингу та керування роботизованими пристроями вимагає використання надійного фреймворку, здатного ефективно обробляти запити, керувати даними з робототехнічних платформ, а також забезпечувати стабільну роботу в умовах високого навантаження. З огляду на вимоги до сумісності, надійності та підтримки корпоративних стандартів, для реалізації бекенду було обрано ASP.NET Web Forms у складі .NET Framework.

Цей фреймворк забезпечує зручний механізм розробки вебінтерфейсів за допомогою шаблонів MasterPage, що дозволяє легко підтримувати єдиний стиль

					ІК12.110БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		29

представлення даних та зручну навігацію по сторінках. Такий підхід забезпечує швидке створення адміністраторських панелей, інтерфейсів для відображення телеметрії та керування діями роботів.

Однією з ключових переваг використання ASP.NET Web Forms є підтримка подієво-орієнтованої моделі розробки, що спрощує реалізацію інтерактивної логіки на серверному боці. Це особливо актуально для реалізації елементів керування в реальному часі — наприклад, запуску, зупинки чи перемикання режимів роботи окремих пристроїв.

Фреймворк також інтегрується з потужним середовищем Visual Studio, що дозволяє створювати та супроводжувати великі багатокомпонентні проєкти з підтримкою діагностики, журналювання, тестування та інтеграції з системами контролю версій. Робота з SQL Server як основною СУБД забезпечується за допомогою ADO.NET або Entity Framework, що дозволяє зручно зберігати дані з сенсорів, логів роботи пристроїв, інформацію про інциденти та журнали обслуговування.

Безпека є важливою складовою при побудові подібних систем, тому фреймворк забезпечує вбудовані механізми автентифікації, авторизації та управління ролями. Це дає змогу організувати розмежування доступу між різними категоріями користувачів — від операторів до інженерів чи адміністраторів системи.

Крім того, система легко масштабується та адаптується під потреби конкретного підприємства або дослідницької лабораторії. Завдяки розвиненій екосистемі .NET, можлива інтеграція з іншими сервісами — включно з модулями IoT, SCADA або OPC-серверами, що критично важливо для забезпечення повноцінного моніторингу у промисловому середовищі.

Таким чином, ASP.NET Web Forms на базі .NET Framework було обрано як стабільну, перевірену платформу, що дозволяє швидко та ефективно реалізувати серверну логіку системи моніторингу та управління роботами з можливістю її подальшої модернізації, розширення і масштабування.

					ІК12.110БАК.006 ПЗ	Арк.
						30
Зм.	Лист	№ докум.	Підпис	Дата		

### 3.7 Вибір бази даних

У процесі створення інформаційної системи для моніторингу та управління роботизованими пристроями одним із ключових етапів стало визначення оптимальної системи управління базами даних (СУБД), яка відповідала б вимогам надійності, масштабованості, інтеграції з .NET Framework та ефективної обробки великих обсягів структурованої інформації в реальному часі.

З огляду на те, що система оперує чітко структурованими даними — такими як технічні параметри роботів, журнали виконання завдань, телеметрія, інформація про помилки й обслуговування — було прийнято рішення на користь реляційної СУБД. Реляційні системи, на відміну від NoSQL-рішень, забезпечують строгі гарантії цілісності, підтримують транзакції, складні запити та встановлення логічних зв'язків між сутностями — усе це критично важливо для промислових і технічних застосунків.

Для реалізації серверної частини проєкту було обрано Microsoft SQL Server. Цей вибір обґрунтований кількома ключовими факторами. По-перше, SQL Server має глибоку інтеграцію з технологіями Microsoft, зокрема .NET Framework, що дозволяє ефективно реалізовувати взаємодію через ADO.NET або Entity Framework. По-друге, SQL Server забезпечує високий рівень продуктивності при роботі з великими масивами даних, що важливо для обробки телеметрії з численних роботизованих пристроїв.

Серед переваг SQL Server також варто відзначити підтримку збережених процедур, тригерів, складних транзакцій та індексації, що дозволяє оптимізувати роботу системи в умовах високої навантаженості. Крім того, можливості автоматизованого резервного копіювання, реплікації та високої доступності роблять цю СУБД надійним вибором для критично важливих інфраструктур.

Для потреб розробки та тестування в локальному середовищі застосовується SQL Server Express, що дозволяє запускати систему без складного налаштування сервера, але з повною сумісністю з продуктивною версією. Це значно полегшує

					ІК12.110БАК.006 ПЗ	Арк.
						31
Зм.	Лист	№ докум.	Підпис	Дата		

процес розробки, налагодження та перевірки функціональності системи в умовах обмеженого ресурсу.

В рамках системи також реалізовано механізми збору та обробки фонових задач, зокрема — створення звітів, моніторинг стану системи чи періодичне сканування стану обладнання. Для цього було застосовано вбудовані можливості SQL Server Agent, що дозволяє планувати виконання завдань без необхідності підключення додаткових інструментів або сервісів.

Таким чином, поєднання SQL Server у якості основної СУБД із можливістю масштабування та високою інтеграцією з платформою .NET забезпечує стабільну, безпечну і ефективну роботу інформаційної системи моніторингу та керування роботами як у промисловому, так і в дослідницькому середовищі. Такий вибір відповідає сучасним вимогам до надійності, обробки великих обсягів даних і підтримки реального часу.

### Висновок до розділу 3

У даному розділі було проведено обґрунтований вибір технологічної основи для реалізації системи моніторингу та управління роботизованими комплексами, що є ключовим етапом у розробці будь-якого програмного продукту промислового або наукового призначення. На основі глибокого аналізу сучасних технологій, архітектурних підходів та практичного досвіду розробки аналогічних систем було сформовано оптимальний технологічний стек, орієнтований на досягнення високого рівня стабільності роботи, масштабованості рішення, а також забезпечення гнучкості під час інтеграції з іншими програмними й апаратними засобами. Такий підхід дозволяє гарантувати довгострокову життєздатність системи та її відповідність сучасним вимогам до автоматизованих систем управління.

Основою серверної частини системи обрано .NET Framework, що є потужною платформою для створення надійних і масштабованих вебзастосунків. Використання цієї технології забезпечило розробникам доступ до багатого набору

					ІК12.110БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		32

функціоналу для обробки запитів, побудови безпечної архітектури й реалізації складної бізнес-логіки. Крім того, важливою перевагою є наявність широких можливостей для інтеграції з апаратним забезпеченням, що дозволило ефективно реалізувати канали взаємодії між програмною частиною системи та роботизованими комплексами. Важливим чинником на користь вибору .NET Framework також стала його розвинена екосистема та підтримка великої кількості інструментів для розробки, тестування та підтримки вебзастосунків.

У побудові клієнтського інтерфейсу було застосовано ASP.NET Web Forms із використанням шаблонної структури MasterPage, що забезпечило послідовність оформлення сторінок, спрощення підтримки й оновлення зовнішнього вигляду, а також підвищення продуктивності роботи над інтерфейсною частиною. Використання Bootstrap дозволило створити сучасний, адаптивний дизайн, орієнтований на зручність взаємодії користувача з системою на пристроях із різними розмірами екранів — від настільних комп'ютерів і ноутбуків до планшетів та смартфонів. Завдяки цьому користувачі отримують комфортний доступ до функціоналу системи незалежно від обраного пристрою, що підвищує загальний рівень задоволеності та ефективності роботи.

У якості системи управління базами даних було обрано Microsoft SQL Server, який є перевіреним і надійним інструментом для роботи з великими обсягами даних і забезпечує високий рівень безпеки, підтримку транзакцій і складних запитів. Це дозволило реалізувати ефективне й структуроване зберігання даних, включаючи лог-файли, параметри роботи роботів, сценарії виконання завдань, журнали подій та звітну документацію. Наявність розвинених засобів адміністрування та резервного копіювання додатково підвищує надійність збереження й обробки інформації

Загалом, обраний підхід до реалізації системи моніторингу та управління роботами дозволяє забезпечити її високу функціональну надійність, продуктивність, гнучкість і розширюваність. Система демонструє здатність інтегруватися з сучасними технологіями та рішеннями, забезпечуючи при цьому простоту адаптації під специфічні потреби різних користувачів. Прийняті на цьому

					ІК12.110БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		33

етапі технічні й архітектурні рішення формують міцний фундамент для ефективної експлуатації системи, її масштабування та подальшого розвитку, з урахуванням швидкої еволюції цифрових технологій, автоматизації виробничих процесів і потреб ринку. Таким чином, створена система є важливим кроком на шляху до побудови інноваційного середовища управління роботизованими комплексами.

					ІК12.110БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		34

## 4 РОЗРОБЛЕННЯ СИСТЕМИ

Розроблена система орієнтована на автоматизацію основних функцій, пов'язаних з наглядом за технічним станом роботів, збором телеметричних даних, передачею команд управління, візуалізацією роботи пристроїв та забезпеченням віддаленого доступу для операторів і технічного персоналу.

Проект реалізовано з урахуванням сучасних вимог до інформаційних систем, зокрема — масштабованості, стабільності, гнучкості налаштувань, а також простоти у користуванні. У межах розділу детально розглянуто структуру системи, її функціональну модель, логіку взаємодії між компонентами, архітектурні рішення, модель бази даних на основі SQL Server, а також принципи обміну та обробки інформації між сервером і користувачем.

Клієнтська частина побудована з використанням технології ASP.NET Web Forms із шаблонами MasterPage, що забезпечує єдину структуру інтерфейсу і зручну навігацію. Для оформлення інтерфейсу застосовано Bootstrap, який гарантує адаптивність дизайну, швидке компонування елементів і коректну роботу системи на різних пристроях. Серверна частина побудована на основі .NET Framework, що дозволяє досягти високої продуктивності, забезпечити надійне з'єднання з обладнанням та обробку запитів у реальному часі [10].

Також у розділі буде розглянуто створення UML-діаграм, зокрема діаграм класів, які дозволяють відобразити внутрішню логіку та взаємозв'язки між основними компонентами системи. Буде представлено структуру користувацького інтерфейсу, що допоможе зрозуміти логіку взаємодії оператора з платформою моніторингу. Завдяки інтеграції елементів керування, обробці подій та реалізації модулів звітності, система демонструє приклад практичного впровадження ІТ-рішень у сфері автоматизації та робототехніки.

Отже, цей розділ описує цілісний процес створення інформаційної системи, починаючи від її концептуального планування до безпосередньої реалізації функціональних модулів і підготовки до використання в реальних умовах експлуатації.

					ІК12.110БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		35

## 4.1 Структура системи

– Система моніторингу та управління роботизованими пристроями побудована за принципами багаторівневої архітектури, що забезпечує надійність, розширюваність та зручність у підтримці. Основні складові цієї архітектури включають:

- інтерфейс користувача на базі ASP.NET Web Forms із MasterPage; серверна частина (ASP.NET Core Web API);
- серверну частину, реалізовану на .NET Framework;
- реляційну базу даних Microsoft SQL Server;
- служби контролю та обробки подій у фоновому режимі.

Кожен із компонентів системи реалізує власний набір функцій і взаємодіє з іншими через чітко визначені інтерфейси, що забезпечує модульність і гнучкість у розширенні функціональності.

Клієнтська частина реалізована на основі ASP.NET Web Forms з використанням шаблонів MasterPage, що дозволяє зберігати єдиний стиль сторінок, забезпечити зручну навігацію та спростити підтримку інтерфейсу. Для створення адаптивного та привабливого візуального середовища застосовується Bootstrap, який дозволяє швидко реалізовувати функціональні елементи управління, адаптувати інтерфейс до різних типів пристроїв та розмірів екранів, а також підтримувати кросбраузерну сумісність.

Серверна частина розроблена з використанням .NET Framework, який надає стабільну платформу для реалізації бізнес-логіки, обробки даних, автентифікації користувачів та взаємодії з обладнанням через стандартні API. В основі логіки — чіткий поділ на рівні: логіка обробки подій, сервіси моніторингу, модулі управління командами та логіка зберігання і вибірки даних. Завдяки застосуванню шаблонів проектування, таких як «репозиторій» та «фабрика», забезпечено просту заміну або доповнення функціональних блоків без зміни загальної структури системи.

					ІК12.110БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		36

Для збереження інформації про стан обладнання, параметри роботи, журнали подій та користувацькі дії використовується SQL Server — високопродуктивна реляційна СУБД, яка підтримує ACID-транзакції, індексацію великих обсягів даних та безпечне управління правами доступу. Завдяки тісній інтеграції SQL Server з .NET-середовищем розробки, була реалізована ефективна ORM-взаємодія через ADO.NET та Entity Framework, що дозволяє зменшити кількість ручного коду при роботі з базою.

Окрему роль у системі відіграють фонові сервіси, що відповідають за обробку запланованих дій, відправку сповіщень, контроль за виконанням задач роботів, а також формування звітів. Ці служби запускаються автоматично через таймери або зовнішні сигнали, не впливаючи на продуктивність основного інтерфейсу. У разі розширення системи на кілька вузлів — фонові логіка може бути масштабована через окремі служби Windows або планувальники задач.

Загалом запропонована архітектура забезпечує стійку роботу системи у виробничих умовах, дозволяє ефективно обробляти потоки даних у реальному часі, зберігати історію змін і оперативно реагувати на критичні ситуації. Таке розмежування функціональності гарантує гнучкість при подальшій модифікації та спрощує впровадження нових функцій або підтримку додаткового обладнання.

## 4.2 Функціональна модель системи

Функціональна модель інформаційної системи для керування та моніторингу роботизованих пристроїв описує ключові взаємодії користувачів із системою, визначаючи ролі, можливості та типові сценарії використання. Такий підхід дозволяє формалізувати вимоги до програмного забезпечення, визначити функціональні межі компонентів та забезпечити ефективну реалізацію системи у виробничому середовищі.

Основні користувачі системи:

– Оператор – здійснює контроль за станом роботів у реальному часі, отримує сповіщення, надсилає базові команди управління;

					ІК12.110БАК.006 ПЗ	Арк.
						37
Зм.	Лист	№ докум.	Підпис	Дата		

– Адміністратор – створює облікові записи, визначає рівні доступу, керує користувачами та правами.

Кожен із типів користувачів має доступ до певного набору функцій, який відповідає його ролі у загальній структурі управління. Наприклад, оператор має доступ лише до панелі моніторингу й базових команд, тоді як інженер може виконувати повну діагностику технічного стану роботів, включно з переглядом журналів, переглядом сенсорних даних та виконанням тестових сценаріїв.

Основні сценарії використання (Use Case):

- авторизація та розмежування доступу (усі ролі);
- моніторинг стану пристроїв у реальному часі (оператор);
- передача команд на виконання (оператор);
- перегляд та аналіз логів системи (оператор);
- оновлення конфігурацій роботів (оператор);
- керування користувачами та правами доступу (адміністратор);
- перевірка підключення до баз даних та резервне копіювання (адміністратор).

Система реалізована на основі .NET Framework, що забезпечує стабільну серверну логіку, а також використання технології MasterPage для збереження єдиного інтерфейсу на всіх сторінках. Інтерфейс створено з використанням Bootstrap, що дозволяє забезпечити адаптивність і зручну роботу на різних пристроях. Для зберігання структурованих даних використовується SQL Server, який дозволяє ефективно управляти інформацією про пристрої, користувачів, події системи та конфігураційні параметри.

Усі взаємодії користувачів супроводжуються відповідною перевіркою доступу до функцій системи, логуванням подій та валідацією даних, що гарантує надійність та безпечну експлуатацію платформи. Завдяки функціональній моделі система підтримує не лише щоденне керування роботами, але й аналітику роботи пристроїв та аудит дій користувачів.

Таким чином, розроблена функціональна модель дозволяє чітко визначити ролі, функціональні обов'язки та права кожного користувача системи. Це створює

					ІК12.110БАК.006 ПЗ	Арк.
						38
Зм.	Лист	№ докум.	Підпис	Дата		

умови для ефективного керування інфраструктурою робототехнічних пристроїв, підвищує надійність керування і спрощує інтеграцію нових функцій у майбутньому.

#### 4.3 Модель бази даних

Для ефективного зберігання, обробки та доступу до даних у системі моніторингу і керування роботизованими пристроями реалізована реляційна база даних, побудована на основі SQL Server. Ця СУБД дозволяє забезпечити високу швидкодію, масштабованість, а також підтримку складної бізнес-логіки, що необхідна для управління великою кількістю пристроїв, користувачів та взаємозв'язків між ними.

Схема бази даних зображена кресленнику ІК12.110БАК.006 Д2.

Основними таблицями є:

robot — містить інформацію про кожен роботизований пристрій, включаючи технічні параметри, дату створення, компанію-виробника, зображення, поточний стан та кількість на складі;

company — зберігає дані про компанії, що випускають або обслуговують роботів;

employee — включає детальну інформацію про співробітників, які взаємодіють із системою;

robot\_user — фіксує прив'язку між працівниками та пристроями, вказуючи період їхнього використання;

admin\_login — відповідає за адміністрування системи, забезпечуючи доступ до управлінського інтерфейсу.

Зв'язки між таблицями реалізовано за допомогою зовнішніх ключів з підтримкою каскадного оновлення та видалення, що відповідає принципам третьої нормальної форми (3НФ) і дозволяє уникнути надмірного дублювання даних. Такий рівень нормалізації підвищує цілісність інформації та спрощує її обробку.

					ІК12.110БАК.006 ПЗ	Арк.
						39
Зм.	Лист	№ докум.	Підпис	Дата		

Робота з даними реалізована за допомогою Entity Framework (EF) — ORM-платформи, яка дозволяє взаємодіяти з базою даних у вигляді об'єктів C#. Завдяки використанню EF у середовищі .NET Framework, забезпечується гнучке управління міграціями, конфігурацією моделі та логікою доступу до даних, що особливо корисно при масштабуванні або перенесенні на інші середовища [3].

Інтерфейс користувача побудований з використанням ASP.NET WebForms у поєднанні з MasterPage, що забезпечує єдину структуру навігації та стилістичну узгодженість усіх сторінок. Для адаптивності й швидкої розробки інтерфейсу застосовується Bootstrap, який дозволяє формувати зручний вигляд сторінок як для стаціонарних пристроїв, так і для мобільних.

Таким чином, створена модель бази даних забезпечує масштабованість, структурованість та розширюваність. Вона ефективно підтримує усі функції системи — від моніторингу стану роботів у реальному часі до ведення історії їх використання різними співробітниками. Обрана архітектура та інструменти дозволяють надалі розвивати систему, не змінюючи її фундаментальних основ.

#### 4.4 Передавання та обробка даних

Надійна передача та обробка даних є фундаментом ефективної роботи системи моніторингу й управління роботизованими пристроями. У контексті багаторівневої архітектури, реалізованої на основі .NET Framework, обмін даними між клієнтською частиною (реалізованою за допомогою ASP.NET WebForms із використанням MasterPage і Bootstrap) та серверною логікою відбувається із забезпеченням високого рівня безпеки та продуктивності.

Ключовим каналом обміну виступає протокол HTTPS, що дозволяє гарантувати цілісність і конфіденційність інформації при передачі. Формат JSON використовується як універсальний стандарт передачі структурованих даних між фронтендом і бекендом, що дозволяє легко серіалізувати інформацію про роботів, користувачів та операційні події, пов'язані з технікою.

Основними джерелами вхідних даних у системі є:

					ІК12.110БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		40

Веб-форми, які заповнюють користувачі для додавання, оновлення чи перегляду інформації про роботів та їхній статус. Дані з форм передаються у вигляді JSON-об'єктів до серверної частини.

Телеметричні дані про стан роботів, які можуть передаватися у фоновому режимі через спеціалізовані служби моніторингу. Ці дані включають інформацію про активність, температуру, навантаження тощо, що зберігається у таблицях бази даних (наприклад, `robot_user` або `robot`) для подальшого аналізу.

Інформація про доступ, що формується під час автентифікації користувачів. Для цього використовується механізм JWT (JSON Web Token), який зберігає інформацію про роль, права доступу та ідентифікатор користувача.

Уся логіка перевірки вхідних даних на сервері реалізована через серверні валідатори у .NET, що дозволяє уникнути помилок на етапі обробки запитів. Зокрема, верифікація значень, контроль за дублюванням, перевірка типів і діапазонів числових параметрів — усе це забезпечується до моменту збереження в SQL Server. Для зручності обробки даних система використовує внутрішні класи DTO (Data Transfer Objects), які перетворюються в бізнес-моделі з урахуванням внутрішньої структури бази.

Загалом, розроблена модель передачі та обробки інформації повністю відповідає вимогам до сучасних корпоративних систем. Вона забезпечує високу продуктивність, захист даних, масштабованість і простоту інтеграції. Завдяки використанню перевірених технологій (HTTPS, JSON, JWT, SQL Server) та гнучкої архітектури на основі .NET Framework, система легко адаптується до змін і дозволяє ефективно розвивати функціонал у відповідь на потреби підприємства.

#### 4.5 Архітектура програмного забезпечення

Серверна частина системи моніторингу та управління роботами побудована відповідно до принципів модульного проектування із чітким розмежуванням відповідальностей між логічними рівнями. Основу архітектури становить концепція трьохшарового підходу, який включає доменний, логічний та

					ІК12.110БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		41

презентаційний рівні, що дозволяє забезпечити підтримуваність, масштабованість і ефективність роботи з кодом. Проєкт реалізовано на базі .NET Framework, що забезпечує стабільність, інтеграцію з технологіями Microsoft та підтримку для побудови корпоративних вебсистем.

Скріншот архітектури програми зображено на рисунку 4.1

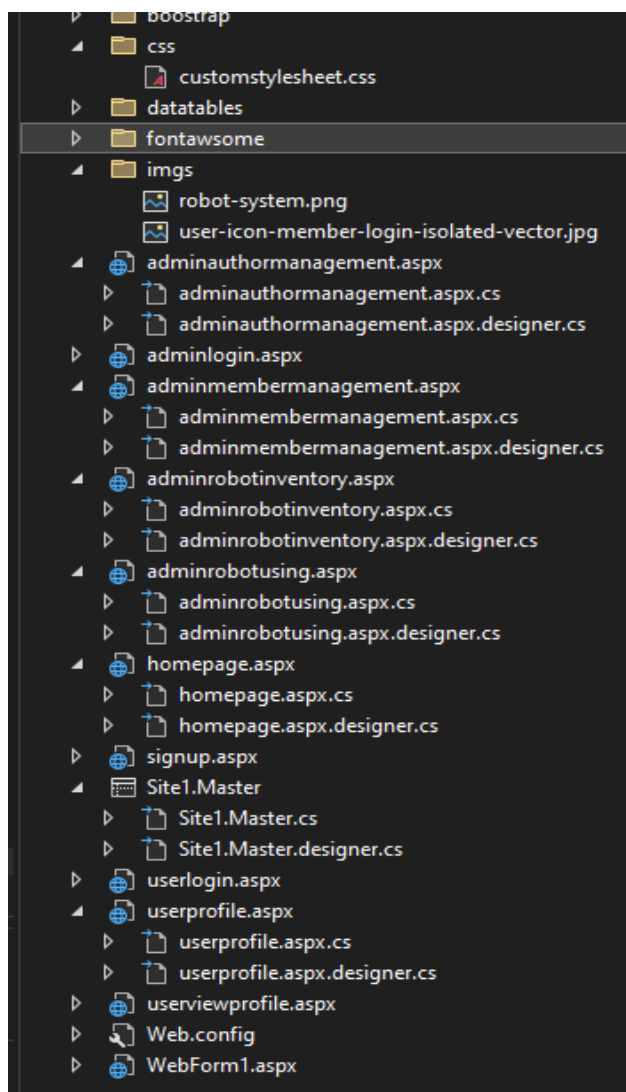


Рисунок 4.1 – Програмна реалізація

#### Структура серверної частини

Domain: містить визначення основних сутностей, що описують предметну область — зокрема, інформацію про роботів, компанії, користувачів системи, історію використання пристроїв. Тут зосереджено бізнес-логіку, інтерфейси доступу до даних та визначення ключових інваріантів.

Presentation (web): побудована з використанням ASP.NET Web Forms і MasterPage для уніфікації інтерфейсу. Тут реалізовано сторінки взаємодії з користувачами, контролери подій, перевірка прав доступу та маршрутизація запитів. Використання MasterPage забезпечує єдину стилістичну структуру та повторне використання UI-елементів.

Application: реалізує логіку роботи сервісів, що взаємодіють із базою даних, DTO для обміну даними між шарами, а також обробку запитів та бізнес-операцій, пов'язаних з реєстрацією роботів, контролем їхнього стану та використанням.

Фронтенд побудований із застосуванням HTML, CSS та фреймворку Bootstrap, що дозволяє створити адаптивний та зручний інтерфейс для моніторингу роботів і керування їхніми параметрами. Сторінки системи, реалізовані у форматі Web Forms, підтримують зворотній зв'язок із сервером та оновлення даних без перезавантаження за допомогою AJAX. Bootstrap дозволяє досягти візуальної узгодженості, забезпечуючи чітку побудову таблиць моніторингу, форм для вводу та індикаторів стану.

Система охоплює ключові таблиці:

robot – містить основну інформацію про пристрої: технічні характеристики, дату створення, тип, енергоспоживання, зображення, поточний та загальний стан на складі.

company – реєструє виробників або власників роботів.

employee – містить дані про співробітників, які взаємодіють із системою.

robot\_user – фіксує історію використання роботів конкретними співробітниками, включаючи дати початку та завершення експлуатації.

admin\_login – реалізує доступ адміністраторів до системи.

Усі зв'язки реалізовано через зовнішні ключі. База даних приведена до третьої нормальної форми (3НФ), що дозволяє уникнути надмірності та забезпечити цілісність даних.

Інтеграція та взаємодія компонентів

					ІК12.110БАК.006 ПЗ	Арк.
						43
Зм.	Лист	№ докум.	Підпис	Дата		

Компоненти системи об'єднані через внутрішні сервіси та інтерфейси, які реалізують залежності через патерн інверсії керування. Це забезпечує слабке зв'язування між модулями та дає можливість масштабувати систему. Наприклад, при зміні типу бази даних достатньо реалізувати новий шар доступу до даних, не змінюючи бізнес-логіку.

Система поділена на функціональні підсистеми:

Реєстрація пристроїв — дозволяє адміністраторам додавати нові моделі робіт, вказуючи детальну інформацію;

Контроль експлуатації — фіксує, хто і коли використовував робота, з можливістю подальшого аудиту;

Моніторинг статусу — в режимі реального часу відображає поточний стан кожного пристрою;

Облік персоналу — керування інформацією про працівників, зокрема їхні права доступу;

Звітність — формування звітів щодо використання робіт, доступне для адміністрації;

Аутентифікація та безпека — підтримка ролей, захищена авторизація через облікові записи.

Цей підхід дозволив створити надійну, модульну та гнучку систему, здатну адаптуватися до потреб організацій, які керують роботизованими пристроями, зберігаючи при цьому простоту обслуговування та легкість розширення. Якщо потрібна візуальна схема, можу допомогти з її оформленням на основі вашої моделі.

#### 4.6 Керівництво користувача

Інформаційна система призначена для ефективного управління роботизованими модулями, з урахуванням різних категорій користувачів. Доступ до функціональних можливостей визначається роллю користувача, яка встановлюється під час авторизації.

					ІК12.110БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		44

Система має адаптивний, зручний для користувача інтерфейс, що забезпечує швидкий доступ до основних можливостей як з комп'ютера, так і з мобільних пристроїв. На зображенні продемонстровано сторінку авторизації, де користувач може увійти до системи або зареєструватися за допомогою інтерфейсу з чіткими полями введення даних і візуально зрозумілими кнопками.

Оператор (користувач):

- входить до системи через особистий ідентифікатор і пароль;
- має змогу переглядати поточний стан роботи
- отримує сповіщення про несправності чи потребу в обслуговуванні;
- завантажує звіти про роботу системи.

Адміністратор:

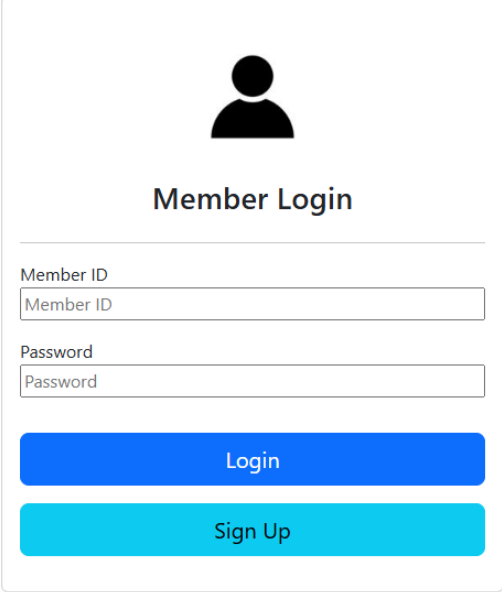
- створює нові облікові записи користувачів;
- керує ролями та правами доступу;
- здійснює загальне адміністрування системи, включаючи оновлення програмного забезпечення.

Діаграма прецедентів зображена на кресленнику ІК12.110БАК.006 Д1.

Інтерфейс інтуїтивно зрозумілий, адаптований під мобільні пристрої, всі поля мають валідацію, дії підтверджуються через спливаючі повідомлення. Далі зображено скріншоти основних сторінок додатку.

Кожна дія підтверджується візуальним повідомленням (наприклад, успішний вхід, помилка в паролі), а всі форми мають вбудовану перевірку правильності введення даних. Сторінка входу, як видно з прикладу на зображенні, забезпечує простий та ефективний початок взаємодії з системою.

Скріншот сторінки входу зображено на рисунку 4.2



Member ID

Member ID

Password

Password

Login

Sign Up

[<< Back to Home](#)

Рисунок 4.2 – Сторінка для входу


Сторінка User Registration призначена для створення нового облікового запису користувача. Вона містить зручну та інтуїтивно зрозумілу форму з полями, необхідними для збору особистих даних. Всі поля мають чітке маркування та структуровані у дві колонки.

Діаграма діяльності процесу входу зображена на кресленнику ІК12.110БАК.006 Д4

Система автоматично перевіряє правильність введених даних, зокрема коректність електронної адреси, формату номера телефону та довжини пароля. У разі успішної реєстрації користувача буде додано до бази даних, після чого він зможе увійти до системи відповідно до своєї ролі (користувач, адміністратор, технічний спеціаліст тощо).

Дизайн сторінки є адаптивним, що забезпечує коректне відображення як на комп'ютерах, так і на мобільних пристроях. Крім того, нижня частина інтерфейсу містить кнопку повернення на головну сторінку, а у верхній панелі доступна навігація до інших розділів системи.

Скріншот сторінки реєстрації зображено на рисунку 4.3



### User Registration

---

Full Name

Contact Number

State

Full Address

User ID

Date of Birth

Email ID

City

Pincode

Password

[<< Back to Home](#)

Рисунок 4.3 – Сторінка для реєстрації

Сторінка Your Profile забезпечує користувача можливістю перегляду та оновлення персональних даних, що зберігаються в системі. Вона структурована у вигляді форми, яка дозволяє редагувати ключову інформацію: ПІБ, дату народження, контактні дані, місце проживання, а також оновлювати пароль доступу до облікового запису.

Діаграма діяльності процесу реєстрації зображена на кресленнику ІК12.110БАК.006 ДЗ

Інтерфейс є адаптивним і зручно відображається як на настільних, так і на мобільних пристроях. У верхньому меню користувач має доступ до переходу на основні сторінки системи: View Robots, User Login, Sign up, Log out, а також бачить персональне привітання.

Сторінка містить візуальний розділ Your Robots, призначений для відображення списку підключених пристроїв або роботизованих модулів, закріплених за користувачем.



Сторінка має адаптивний дизайн, що дозволяє зручно працювати з нею як на стаціонарних, так і на мобільних пристроях. Це забезпечує ефективну взаємодію користувача із системою у будь-який момент.

Скріншот сторінки роботів зображено на рисунку 4.5

The screenshot shows a web application interface for a 'Robot system'. At the top, there is a navigation bar with 'Home', 'About Us', and 'Terms' on the left, and 'View Robots', 'User Login', 'Sign up', 'Log out', and 'Hello user' on the right. The main content area is divided into two sections. The left section, titled 'Robots', contains a form with two columns of input fields. The first column includes 'Member ID', 'Member Name', and 'Start Date' (with a date picker icon). The second column includes 'Robot ID', 'Robot Name', and 'End Date' (with a date picker icon). There are 'Issue' and 'Return' buttons at the bottom of the form. A 'Go' button is positioned between the 'Robot ID' and 'Robot Name' fields. Below the form is a '<< Back to Home' link. The right section, titled 'Robot List', is currently empty.

Рисунок 4.5 – Сторінка пристроїв

Сторінка перегляду даних працівника є важливим компонентом системи, яка дозволяє адміністраторам керувати інформацією про співробітників. На ній відображається детальна форма з полями для введення або редагування ідентифікаційного номера працівника, повного імені, дати народження, контактного номера, адреси електронної пошти, місця проживання (штат, місто), поштового коду та повної поштової адреси. Дані про події, додаючи нові, змінюючи існуючі або видаляючи неактуальні.

Особливу увагу приділено стану облікового запису, який можна змінювати за допомогою візуально зрозумілих кнопок: активація (зелена), тимчасове призупинення (жовта) та видалення (червона). У нижній частині сторінки доступна опція повного й безповоротного видалення користувача.

Права частина сторінки відведена під список працівників, що дозволяє швидко переглядати або обирати користувачів для редагування. Інтерфейс є інтуїтивно зрозумілим і зручним для адміністрування, зберігаючи при цьому адаптивність для різних типів пристроїв.

Скріншот сторінки переліку працівників зображено на рисунку 4.6

					ІК12.110БАК.006 ПЗ	Арк.
						49
Зм.	Лист	№ докум.	Підпис	Дата		

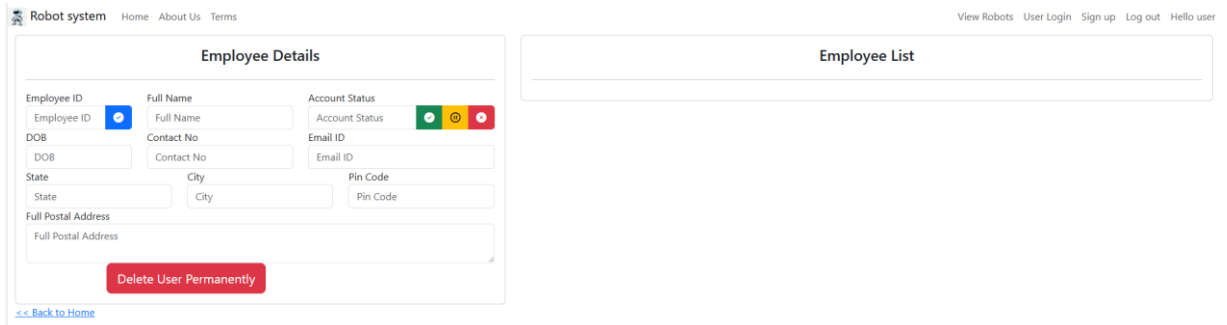


Рисунок 4.6 – Сторінка працівників

Сторінка авторизації адміністратора є захищеним шлюзом доступу до адміністраторської частини системи. Вона призначена для перевірки повноважень і надання доступу лише уповноваженим користувачам.

Інтерфейс реалізований у вигляді компактної форми, що містить два поля для введення логіна та пароля, а також велику кнопку «Login» для підтвердження входу. Над формою розміщено іконку користувача та заголовок «Admin Login», що чітко позначає функціональне призначення сторінки.

Сторінка також має навігаційне посилання для повернення на головну сторінку, що забезпечує зручність навігації. Завдяки адаптивному дизайну форма коректно відображається на різних пристроях, включно зі смартфонами та планшетами. Це забезпечує гнучкість адміністрування в будь-який час і з будь-якого місця.

Скріншот сторінки входу для адміну зображено на рисунку 4.7

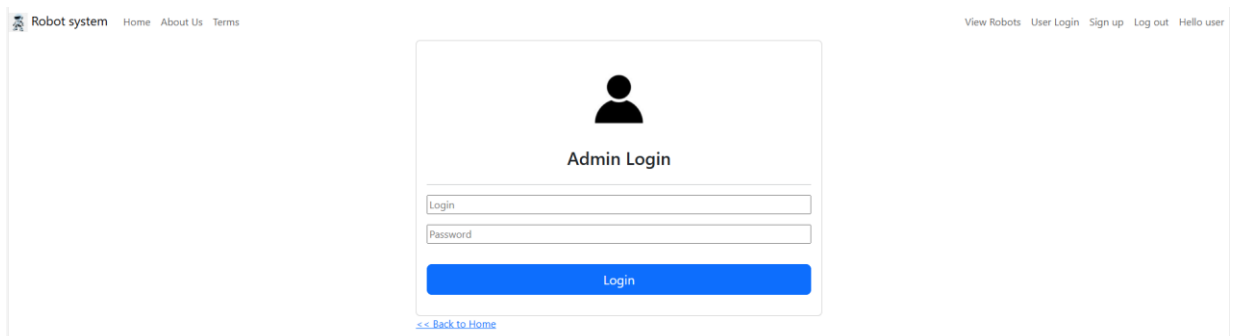


Рисунок 4.7 – Сторінка логіну для адміну

#### 4.7 Безпека системи

Система реалізує функціонал обліку, моніторингу та контролю дій з роботами, включаючи операції видачі, повернення, перегляду поточного статусу робіт та управління користувачами. Завдяки використанню Bootstrap забезпечено адаптивний інтерфейс, який коректно працює як на настільних комп'ютерах, так і на мобільних пристроях. Всі форми введення (наприклад, для ідентифікації працівників, введення даних про робіт або авторизації адміністратора) мають зручне компонування, що відповідає вимогам до сучасного UX/UI дизайну.

Архітектура системи побудована на базі ASP.NET Web Forms, що дозволяє організувати чіткий поділ між логікою обробки подій на сервері та представленням даних. Застосування Master Page сприяє зменшенню дублювання коду та полегшує впровадження глобальних змін до структури сайту.

Для забезпечення захищеного доступу до функціоналу реалізовано механізм автентифікації з перевіркою облікових даних адміністратора та звичайного користувача. Передбачено контроль сесій та обмеження доступу до адміністративної частини лише для авторизованих осіб.

Для обробки запитів до бази даних використовуються стандартні механізми ADO.NET, з урахуванням принципів безпеки при роботі з SQL-запитами. Крім того, впроваджено валідацію вхідних даних на рівні як клієнта (через HTML5 і Bootstrap), так і сервера (через засоби .NET Framework), що дозволяє уникнути типових помилок та потенційних загроз, пов'язаних з неправильним введенням інформації.

Таким чином, система, створена на основі Bootstrap і .NET Framework з використанням Master Page, забезпечує ефективне управління роботизованими пристроями, контроль дій користувачів та зручність адміністрування, що робить її надійним інструментом для автоматизованих облікових процесів.

					ІК12.110БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		51

## 4.8 Тестування системи

Для забезпечення стабільності та коректної роботи системи управління роботами, що розроблена із застосуванням .NET Framework, Master Page та фронтенд-фреймворку Bootstrap, проведено всебічне тестування, яке охоплювало як окремі елементи функціоналу, так і їхню інтеграцію у межах загальної архітектури застосунку.

Тестування виконувалось на всіх етапах життєвого циклу розробки, починаючи з перевірки логіки окремих серверних модулів (наприклад, обробка видачі та повернення роботів), і закінчуючи валідацією взаємодії між інтерфейсом користувача та серверною частиною. Особливу увагу приділено перевірці коректності обробки запитів у рамках реалізованої багаторівневої архітектури на основі Master Page.

Інтерфейс користувача, створений з використанням компонентів Bootstrap, перевірявся на коректність відображення у різних браузерях та на різних пристроях. Тестувалися всі інтерактивні елементи, зокрема форми введення, кнопки, повідомлення про помилки та переходи між сторінками.

На рівні логіки додатку проводилось модульне тестування функцій, пов'язаних з управлінням користувачами, а також з контролем доступу та обробкою даних. Окремі тести були спрямовані на перевірку функціонування системи авторизації, валідації введених даних та захисту від типових помилок, таких як повторне надсилання форми чи введення некоректних даних.

Крім функціонального тестування, було здійснено навантажувальну перевірку для оцінки, як система реагує на велику кількість одночасних запитів. Завдяки цьому вдалося виявити вузькі місця у роботі серверної частини та оптимізувати обробку даних.

Усі виявлені на етапі тестування помилки фіксувались, аналізувались та усувались до впровадження системи в експлуатацію. Це дозволило досягти високої надійності, забезпечити зручність користування інтерфейсом і гарантувати відповідність роботи системи функціональним вимогам.

					ІК12.110БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		52

#### 4.8.1 Юніт-тести

З метою забезпечення надійності серверної логіки в системі управління та моніторингу роботів у рамках проєкту було впроваджено комплекс юніт-тестів, що охоплюють основні функції бізнес-логіки. Тестування здійснювалося на рівні окремих сервісів, реалізованих у середовищі ASP.NET Core, з акцентом на коректну обробку вхідних даних, валідацію, перетворення об'єктів до DTO і обробку виключень.

Основою тестової інфраструктури є бібліотека NUnit — стандартний фреймворк для тестування в .NET, який дозволяє створювати зрозумілі й структуровані тести, підтримує роботу з атрибутами, групуванням і перевіркою очікуваних результатів. Він забезпечує інтеграцію з інструментами візуалізації результатів тестування в середовищі Visual Studio, що полегшує контроль якості коду.

Для імітації зовнішніх залежностей використовується бібліотека Moq, яка дозволяє створювати мок-об'єкти з визначеним набором поведінки. Завдяки цьому досягається ізоляція сервісів під час тестування, що дозволяє зосередитись на перевірці внутрішньої логіки без взаємодії з базою даних або сторонніми API. Moq забезпечує можливість налаштування відповідей методів, перевірку викликів і сценаріїв з винятками.

Генерація тестових даних реалізована за допомогою бібліотеки AutoFixture, яка дозволяє автоматично створювати об'єкти з випадковими, але структурованими значеннями. Це значно пришвидшує процес підготовки до тестування, дає змогу легко перевірити поведінку сервісів у випадках з нетиповими або граничними вхідними даними.

У процесі тестування перевірялись як типові сценарії використання (наприклад, успішна обробка запитів), так і нестандартні ситуації, включаючи обробку винятків, валідацію некоректних даних і неправильні запити. Особлива увага була приділена сервісам, що відповідають за управління роботами, авторизацію користувачів і логування дій.

					ІК12.110БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		53

Завдяки використанню NUnit, Moq та AutoFixture вдалося побудувати гнучке й надійне середовище для тестування серверної частини. Це підвищує стабільність системи, полегшує виявлення помилок на ранніх етапах і забезпечує можливість швидкого масштабування та доопрацювання проєкту в майбутньому.

#### 4.8.2 Ручне тестування

Для забезпечення стабільної та зручної роботи системи моніторингу й дистанційного управління роботами було проведено повноцінне ручне тестування, що охоплює всі основні сценарії взаємодії користувача із системою – від входу до виконання операцій над пристроями та перевірки коректності збереження інформації. Тестування здійснювалось у браузерному середовищі на різних пристроях, що дало змогу виявити як логічні, так і візуальні помилки, а також проблеми у взаємодії між фронтендом і бекендом.

Першим кроком стала перевірка роботи інтерфейсу, створеного з використанням Razor Pages у .NET Core. Було протестовано коректність навігації між сторінками, підвантаження контенту, логіку шаблонів і відповідність верстки принципам адаптивного дизайну, реалізованого через Bootstrap. Тестування охопило всі варіанти відображення: настільні браузери, планшети та мобільні пристрої.

Наступним етапом була перевірка процесу автентифікації. Було протестовано обробку правильних і неправильних облікових даних, захист від повторного надсилання форм, збереження сесії, а також перевірка механізмів авторизації на рівні ролей. Особливу увагу приділено роботі JWT-токенів, їх коректній генерації, передачі в заголовках запитів і збереженню в localStorage або sessionStorage.

Було проведено ручне тестування всіх форм введення даних, пов'язаних із реєстрацією або обліком роботизованих пристроїв. У рамках перевірки використовувались інструменти розробника браузера Chrome DevTools, за допомогою яких перевірялись валідація, підказки, блокування кнопок, візуалізація

					ІК12.110БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		54

помилки і коректність обов'язкових полів. Вводились як типові, так і граничні значення для виявлення некоректної обробки.

Ключові функціональні сценарії, включаючи додавання, зміну та видалення записів про роботи, були перевірені вручну. Тестувалась логіка оновлення інформації, правильність її збереження в базі даних, а також відповідність відображення поточного стану в інтерфейсі. Повторна перевірка сторінок після оновлення браузера підтвердила коректне завантаження оновлених даних.

Для перевірки API-логіки використовувався інструмент Thunder Client — розширення для Visual Studio Code, яке дозволяє швидко створювати та надсилати HTTP-запити. Через нього вручну тестувались всі основні ендпоінти: авторизація, отримання списків, створення й оновлення об'єктів. Оцінювалась структура JSON-відповідей, статус-коди, повідомлення про помилки, а також стабільність взаємодії з базою даних через REST.

Окремо перевірялась система повідомлень користувачу: своєчасне відображення інформативних повідомлень, можливість їх закриття, перевірка сценаріїв заборонених дій (наприклад, доступ без прав). Тестувалась реакція системи на спроби змінити дані без відповідного рівня авторизації.

Проведене ручне тестування підтвердило коректність реалізації ключових функцій, стабільність роботи системи та її зручність для кінцевого користувача. Застосування сучасних інструментів, таких як Chrome DevTools і Thunder Client, дозволило комплексно перевірити як фронтенд, так і серверну частину застосунку, що забезпечує готовність системи до розгортання в реальному середовищі.

#### 4.8.3 Навантажувальне тестування

Для перевірки продуктивності та стійкості роботи системи управління роботами в умовах підвищеного навантаження було проведено навантажувальне тестування з використанням сучасного інструменту Artillery. Цей фреймворк, орієнтований на Node.js, дозволяє моделювати поведінку реальних користувачів і надсилати велику кількість HTTP-запитів до API системи. Artillery підтримує

					ІК12.110БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		55

сценарії на JavaScript, гнучку конфігурацію профілю навантаження та генерацію детальних звітів.

Під час тестування імітувалося одночасне надсилання десятків і сотень запитів до ключових частин серверної логіки, реалізованої на базі .NET Core. Перевірялась робота ендпоінтів авторизації, обліку роботів, обробки запитів на видачу та повернення пристроїв, а також отримання звітності. Сценарії включали як прості операції, так і запити, що передбачають взаємодію з базою даних і складну бізнес-логіку.

У результаті тестування було підтверджено, що система здатна стабільно функціонувати при збільшенні кількості одночасних запитів без істотного зниження продуктивності. Разом з тим, аналіз показав, що окремі частини, пов'язані з великомасштабними вибірками даних, потребували оптимізації. Після внесення змін у запити та покращення індексації бази даних спостерігалось зменшення часу відповіді та стабілізація навантаження.

Таким чином, навантажувальне тестування за допомогою Artillery дозволило виявити потенційні проблеми продуктивності ще до впровадження системи в експлуатацію, що значно підвищило її готовність до роботи у реальних умовах із великою кількістю активних користувачів.

#### Висновок до розділу 4

У цьому розділі було детально розглянуто архітектуру, функціональні можливості та технічну реалізацію системи управління та моніторингу роботів, створеної з використанням сучасних веб-технологій, зокрема .NET Framework, Bootstrap і шаблонної структури Master Page. Ці технології дозволили розробити гнучку, масштабовану й надійну систему, яка відповідає вимогам до сучасного програмного забезпечення для управління складними об'єктами.

Описано ключові компоненти клієнтської та серверної частин системи, що забезпечують коректну взаємодію між користувачем і апаратною частиною. Особливу увагу приділено способам взаємодії з базою даних, зокрема механізмам

					ІК12.110БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		56

збереження, обробки й аналізу даних про роботу пристроїв. Реалізовані механізми авторизації та контролю доступу забезпечують захист інформації й обмеження дій користувачів відповідно до їхніх ролей у системі. Така структура дозволяє централізовано управляти роботами, вести облік їх використання, здійснювати моніторинг стану та оперативно реагувати на зміну параметрів роботи пристроїв.

Представлена функціональна модель системи є комплексною й охоплює всі основні сценарії використання. Вона включає дії як адміністратора, так і рядових користувачів: реєстрацію нових облікових записів, видачу та повернення пристроїв, формування звітів, ведення журналу подій і перегляд стану обладнання в реальному часі. Завдяки використанню адаптивного інтерфейсу на основі Bootstrap система є зручною для роботи як на стаціонарних комп'ютерах, так і на мобільних пристроях, що значно розширює сферу її застосування.

Особливу увагу під час реалізації системи було приділено питанням тестування. Проведено модульну перевірку роботи окремих компонентів за допомогою таких засобів, як NUnit, Moq та Autofixture, що дозволило перевірити коректність логіки обробки даних і взаємодії між модулями. Додатково виконано ручне тестування функціоналу користувацького інтерфейсу та навантажувальне тестування із використанням Postman і кб. Ці заходи дали змогу оцінити продуктивність системи та її здатність працювати стабільно під час високого навантаження, що особливо важливо для промислових рішень.

Завдяки продуманій архітектурі, ретельному вибору технологій і впровадженню багаторівневої системи тестування, розроблена система повністю відповідає сучасним вимогам до веб-застосунків. Вона демонструє високу надійність і стабільність роботи, забезпечує можливість подальшого розвитку й інтеграції нових функцій. Це робить систему гнучкою платформою для впровадження нових рішень в сфері управління й моніторингу роботів у різних галузях.

## ВИСНОВКИ

У межах розробки системи управління та моніторингу роботами було реалізовано комплексне вебрішення, призначене для централізованого обліку, контролю та аналізу виконання робіт у рамках організаційної діяльності. Побудова системи базувалась на перевірених технологіях — серверна частина реалізована за допомогою .NET Framework, що забезпечило стійку архітектуру, можливість ефективної обробки запитів та зручну взаємодію з базою даних. Це дозволило досягти високої надійності функціонування у виробничих або адміністративних умовах.

Інтерфейс користувача створено із застосуванням Bootstrap, що забезпечило адаптивність зовнішнього вигляду для різних пристроїв — від стаціонарних ПК до мобільних платформ. Комбінація стандартних компонентів Bootstrap та індивідуального стилювання надала системі сучасний та зрозумілий вигляд, що значно покращує користувацький досвід.

В основі побудови візуальної структури застосунку використано технологію Master Page, яка дозволяє створити єдину шаблонну структуру для всіх сторінок системи. Завдяки цьому вдалося уніфікувати інтерфейс, спростити підтримку проєкту, а також полегшити його подальше розширення — як у межах додавання нових функцій, так і при адаптації до нових вимог користувачів або бізнес-логіки.

Система забезпечує можливість створення завдань, призначення виконавців, встановлення термінів, а також моніторинг прогресу їх виконання. Інструменти звітності дають змогу оцінювати ефективність праці, виявляти затримки та планувати подальші дії. Користувачі з різними ролями (адміністратор, керівник, виконавець) мають доступ до функціоналу згідно з рівнем прав, що гарантує безпеку даних та дотримання принципу мінімальних привілеїв.

Під час реалізації також було враховано можливість масштабування системи. Завдяки розділенню логіки на окремі шари (презентаційний, логічний, доступу до даних), застосунок можна адаптувати до змін у структурі організації або підключити до інших зовнішніх сервісів без повного перероблення архітектури.

					ІК12.110БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		58

У процесі розробки система була протестована з використанням як ручного, так і автоматизованого тестування. Це дозволило виявити та усунути ряд потенційних помилок ще на етапі впровадження, підвищити стабільність роботи, скоротити ризики помилок у продакшн-середовищі та забезпечити якісне користувацьке тестування.

Таким чином, розроблена система є ефективним інструментом для організації, координації та контролю робіт в умовах реального робочого середовища. Вона поєднує сучасний інтерфейс, надійний бекенд і чітку структуру, що робить її придатною як для внутрішнього використання в організаціях, так і для подальшого масштабування або комерційного впровадження.

					ІК12.110БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		59

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Microsoft Docs – .NET Framework Documentation <https://learn.microsoft.com/en-us/dotnet/framework/> (Офіційне джерело щодо архітектури, структури, безпеки та реалізації систем на .NET Framework)
2. Bootstrap Official Documentation <https://getbootstrap.com/> (Актуальна інформація щодо компонентів, адаптивної верстки, грид-системи тощо)
3. Entity Framework Guide <https://learn.microsoft.com/en-us/ef/> (Документація по ORM для зручного доступу до БД у .NET)Dev D. F. S. JWT, OAuth, OpenID Connect, and Azure AD: Authentication Levels. *DEV Community*. URL: <https://dev.to/dotnetfullstackdev/jwt-oauth-openid-connect-and-azure-ad-authentication-levels-4a4m>
4. ASP.NET Web Forms – Master Pages Overview <https://learn.microsoft.com/en-us/aspnet/web-forms/overview/older-versions-style/master-pages/master-pages>
5. Freeman A., "Pro ASP.NET Web Forms in C#." Apress, 2014. (Практичний посібник із побудови ASP.NET Web Forms-додатків із використанням Master Pages, доступ до даних, UI)Google Classroom: All you need to know | Hiver™. *Hiver: AI-powered customer service platform*. URL: <https://hiverhq.com/blog/google-classroom-basics>
6. Microsoft SQL Server Documentation <https://learn.microsoft.com/en-us/sql/sql-server/>
7. Stack Overflow Developer Survey (2023) <https://survey.stackoverflow.co/2023/>
8. Microsoft Visual Studio Documentation <https://learn.microsoft.com/en-us/visualstudio/?view=vs-2022>
9. CodeProject <https://www.codeproject.com/>
10. ASP.NET Web API Documentation <https://learn.microsoft.com/en-us/aspnet/web-api/>
11. Microsoft Learn Training <https://learn.microsoft.com/en-us/training/>
12. C# Corner <https://www.c-sharpcorner.com/>
13. Internet Information Services (IIS) Docs <https://learn.microsoft.com/en-us/iis/>

					ІК12.110БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		60

14. JetBrains Documentation <https://www.jetbrains.com/help/>

15. ROS Documentation <https://docs.ros.org>

16. RobotIQ <https://insights.robotiq.co>

					ІК12.110БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		61