

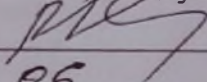
Бакалаврська кваліфікаційна робота на тему:

WEB-ЗАСТОСУНОК ДЛЯ ТЕСТУВАННЯ МОБІЛЬНОЇ МЕРЕЖІ

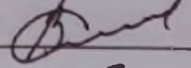
Виконав: студент 4-го курсу, групи ПЗТ-21
спеціальності 172 – Телекомунікації та
радіотехніка

 Дмитрук М.К.

Керівник: д.т.н., завідувач кафедри ІКТ

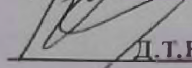
 Кичак В.М.
« 12 » 06 2025 р.

Рецензент: д.т.н., доц. каф. ІРТ

 Воловик А.К.
« 12 » 06 2025 р.

Допущено до захисту

Завідувач кафедри ІКТ

 д.т.н., проф. Кичак В.М.

« 12 » 06 2025 р.

Факультет Інформаційних електронних систем
Кафедра Інфокомунікаційних систем і технологій
Рівень вищої освіти перший (бакалаврський)
Галузь знань – 17 Електроніка та телекомунікації

(шифр і назва)

Спеціальність – 172 – Телекомунікації та радіотехніка

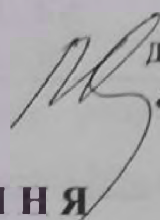
(шифр і назва)

Освітньо-професійна програма – Програмне забезпечення телекомунікаційних систем

ЗАТВЕРДЖУЮ

Завідувач кафедри ІКСТ

д.т.н., професор В.М. Кичак

 «5» 05 2025 року

З А В Д А Н Н Я
НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ

Дмитрук Максим Юрійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи. Web - застосунок для тестування мобільної мережі
керівник роботи: Кичак Василь Мартинович, док. техн. наук, професор

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом ВНТУ від “ 20 ” березня 2025 року № 97

2. Строк подання студентом роботи 09 червня 2025 року

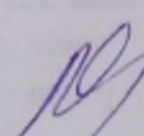
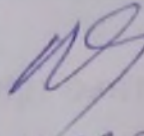
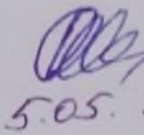
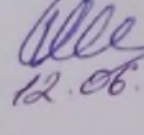
3. Вихідні дані до роботи: розробити програму для тестування телекомунікаційних мереж, розглянути принцип побудови системи, провести тестування програмного забезпечення.

4. Зміст текстової частини: Вступ. Аналіз предметної області та обґрунтування вибору технології. Розрахунки навантажень і параметрів телекомунікаційної мережі. Проектування web-застосунку. Розробка та реалізація web-застосунку. Охорона праці. Висновки. Список використаних джерел. Додатки.

5. Перелік ілюстративного матеріалу:

Результати тестування здійсненні Web-застосунком. Відкритий код програмного забезпечення.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
Основна частина	Кичак В.М. завідувач кафедри ІКСТ, д.т.н., професор	 5.05.2025	 12.06.2025
Охорона праці	Дембіцька С. В. професор кафедри БЖДПБ, д.п.н.,	 5.05.2025	 12.06.2025

7. Дата видачі завдання «05» травня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів бакалаврської дипломної роботи	Строк виконання етапів роботи	Примітки
1.	Формування та затвердження теми та індивідуального завдання бакалаврської дипломної роботи (ДБР)	06.05.2025р.	
2.	Виконання спеціальної частини БДР. Перший рубіжний контроль виконання БДР	19.05.2025р.	
3.	Виконання спеціальної частини БДР. Другий рубіжний контроль виконання БДР	31.05.2025р.	
4.	Виконання розділу «Охорона праці»	04.06.2025р.	
5.	Нормоконтроль БДР	09.06.2025р.	
6.	Попередній захист БДР	10.06.2025р.	
7.	Рецензування БДР	11.06.2025р.	
8.	Захист БДР ЕК	13.06.2025р.	

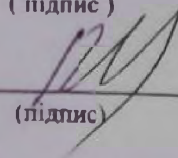
Студент



Дмитрук М.Ю

(підпис)

Керівник роботи



Кичак В.М.

(підпис)

АНОТАЦІЯ

УДК 621.391

Дмитрук М.Ю. Web - застосунок для тестування мобільної мережі. Бакалаврська дипломна робота зі спеціальності 172 - Телекомунікації та радіотехніка, освітня програма – програмне забезпечення телекомунікаційних систем. Вінниця: ВНТУ, 2025. 108 с. Укр. мовою. Бібліогр.: 56 назв; рис.: 12; табл. 7.

Бакалаврська дипломна робота присвячена експериментальному дослідженню та розробці програмного забезпечення для тестування мобільних телекомунікаційних мереж. Робота включає розробку тестування мобільної мережі та подання результатів у вигляді карт, таблиць та графіків.

У бакалаврській дипломній роботі проведено аналіз вимог до інтерфейсу користувача, обґрунтовано вибір HTML, CSS та TypeScript як основних засобів реалізації. Описано процес побудови адаптивного інтерфейсу, обробки подій, формаму входу, а також модулі для візуалізації статистики. На основі цього аналізу розроблено програмне забезпечення, яке дозволяє тестувати мобільну мережу.

Ключові слова: тестування, телекомунікаційні мережі, мобільна мережа, статистика, пропускна здатність.

ABSTRACT

UDK 621.391

Dmytruk M.Yu. Web - an application for testing a mobile network. Bachelor's thesis in the specialty 172 - Telecommunications and radio engineering, educational program - software for telecommunication systems. Vinnytsia: VNTU, 2025. 108 p. In Ukrainian. Bibliography: 56 titles; Fig.: 12; Table. 7.

The bachelor's thesis is devoted to experimental research and development of software for testing mobile telecommunication networks. The work includes the development of mobile network testing and presentation of results in the form of maps, tables and graphs.

The bachelor's thesis analyzes the requirements for the user interface, justifies the choice of HTML, CSS and TypeScript as the main means of implementation. The process of building an adaptive interface, event processing, login form, as well as modules for visualizing statistics is described. Based on this analysis, software has been developed that allows testing a mobile network.

Keywords: testing, telecommunication networks, mobile network, statistics, bandwidth.

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	7
ВСТУП	11
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОБҐРУНТУВАННЯ ВИБОРУ ТЕХНОЛОГІЇ	15
1.1. Сучасний стан мобільних мереж та їх тестування	15
1.2. Параметри мобільної мережі, їх ефективність та оцінювання	16
1.3. Існуючі програмні засоби для аналізу мобільної мережі	19
1.4. Переваги та недоліки існуючих рішень.....	20
1.5. Постановка задачі та обґрунтування вибору технологій.....	22
1.5.1 Обґрунтування вибору технологій для реалізації клієнтської частини	23
1.5.2 Обґрунтування вибору бібліотеки для реалізації інтегральної карти.....	25
1.6 Обґрунтування вибору технологій для реалізації серверної частини	26
1.7 Висновки до розділу “Аналіз предметної області та обґрунтування вибору технологій”	27
2 РОЗРАХУНКИ НАВАНТАЖЕНЬ І ПАРАМЕТРІВ ТЕЛЕКОМУНІКАЦІЙНОЇ МЕРЕЖІ.....	29
2.1 Оцінка кількості одночасних користувачів і навантаження на систему.....	29
2.2 Вимоги до каналу зв’язку	31
2.3 Розрахунок навантаження на сервер.....	32
2.4 Розрахунок енергоспоживання сервера.....	34
2.5 Розрахунок пропускної здатності оптичного каналу	36
2.6 Розрахунок загасання в оптичному волокні	38
2.7 Висновки до розділа “Розрахунки навантажень і параметрів телекомунікаційної мережі”	39
3 ПРОЄКТУВАННЯ WEB-ЗАСТОСУНКУ.....	41
3.1 Архітектура системи.....	41
3.2. Функціональні вимоги до застосунку	42
3.3. Принципи побудови клієнт-серверної взаємодії	43
3.4 Візуалізація елементів мобільної мережі	44
3.5 Висновки до розділу “Проектування web-застосунку”	44
4 РОЗРОБКА ТА РЕАЛІЗАЦІЯ WEB-ЗАСТОСУНКУ.....	46
4.1 Візуалізація компонентів web-застосунку	46
4.2 Реалізація алгоритму тестування мобільної мережі.....	55
4.3 Реалізація відображення даних на карту	57
4.4 Реалізація серверної частини	58
4.5 Висновки до розділу “Розробка та реалізація web-застосунку”	61

5 ОХОРОНА ПРАЦІ.....	63
5.1. Технічні рішення щодо безпечного виконання роботи	64
5.1.1 Технічні рішення з безпечної організації робочих місць	64
5.1.2 Технічні рішення з електробезпеки	65
5.2 Технічні рішення з гігієни праці та виробничої санітарії.....	67
5.2.1 Мікроклімат.....	67
5.2.2. Склад повітря робочої зони	68
5.2.3 Виробниче освітлення	69
5.2.4 Виробничий шум	71
5.2.5. Виробничі випромінювання	72
5.3.1. Технічні рішення системи запобігання пожежі.....	75
5.4 Висновки до розділу “Охорона праці”	77
ВИСНОВКИ	78
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	80
Додаток А ІЛЮСТРАТИВНИЙ МАТЕРІАЛ.....	86
Додаток Б ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ.....	90
Додаток В ЛІСТИНГ WEB-ЗАСТОСУНКУ ДЛЯ ТЕСТУВАННЯ МОБІЛЬНОЇ МЕРЕЖІ.....	91

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

API - Application Programming Interface - набір правил і протоколів для взаємодії між програмними компонентами.

AWS - Amazon Web Services - хмарна платформа від компанії Amazon, яка надає широкий спектр сервісів для зберігання, обробки та доставки даних, розгортання додатків і інфраструктури.

CE - *conformité européenne* - сертифікат що свідчить про те, що продукція відповідає вимогам безпеки, охорони здоров'я та захисту довкілля, встановленим законодавством Європейського Союзу

DTO - Data Transfer Object - об'єкт, який використовується для передачі даних між різними шарами програми або між системами, зазвичай без логіки, лише з полями для зберігання інформації.

GNS3 - Graphical Network Simulator 3 - емулятор мережевого обладнання, дозволяє емулювати реальне мережеве обладнання (Cisco, MikroTik, Juniper, Linux тощо) для побудови віртуальної мережі, тестування конфігурацій, лабораторних робіт.

HDD - Hard Disk Drive - накопичувач даних із магнітними пластинами і механічною голівкою для читання/запису інформації.

HTTP - HyperText Transfer Protocol - основний протокол для передачі даних у мережі Інтернет, що забезпечує обмін інформацією між клієнтом (браузером) і сервером.

IoT - Internet of Things - це мережа фізичних пристроїв, об'єднаних в мережу Інтернет, яка дозволяє їм взаємодіяти та обмінюватися даними.

JSON - JavaScript Object Notation - текстовий формат обміну даними.

JWT - JSON Web Token - стандарт відкритого формату для безпечної передачі інформації у вигляді JSON-об'єкта між сторонами, який можна перевірити і довіряти за допомогою цифрового підпису.

NS-3 - Network Simulator 3 - симулятор мережевого рівня що використовується для досліджень і симуляцій мережевого протоколу, поведінки мережі на рівні пакетів

OMNeT++ - Objective Modular Network Testbed in C++ - це модульний дискретно-подійний симулятор, призначений для моделювання мереж різного типу, з акцентом на дослідження протоколів, телекомунікацій, мереж сенсорів, транспортних і розподілених систем.

Open5GS - Open Source 5G System - це відкрита реалізація ядра мобільної мережі 4G/5G (EPC/5GC), написана мовами C та C++.

ORM - Object-Relational Mapping - технологія (або підхід) для відображення об'єктів програмної мови на записи в реляційній базі даних, що дозволяє працювати з базою даних через об'єкти.

QoS - Quality of Service - сукупність технологій і механізмів, які забезпечують гарантований рівень продуктивності мережевих сервісів.

RAM - Random Access Memory - тип комп'ютерної пам'яті, яка використовується для тимчасового зберігання даних і програм, що активно використовуються під час роботи системи.

REST - Representational State Transfer - архітектурний стиль для побудови веб-сервісів із використанням HTTP.

RSRP - Reference Signal Received Power - потужність прийнятого опорного сигналу

RSSI - Received Signal Strength Indicator - індикатор рівня прийнятого сигналу

RxJS - Reactive Extensions for JavaScript - бібліотека для реактивного програмування, яка дозволяє працювати з асинхронними потоками даних у JavaScript і TypeScript.

SINR - Signal to Interference plus Noise Ratio - відношення сигналу до інтерференції та шуму

SLA - Service Level Agreement - офіційний договір між постачальником послуг і клієнтом, що визначає стандарти якості, доступності та інших параметрів обслуговування.

srsRAN - Software Radio Systems Radio Access Network - це відкрите програмне забезпечення для реалізації LTE/5G радіомережі (RAN).

SSD - Solid State Drive - пристрій для зберігання даних, який використовує флеш-пам'ять для швидкого доступу до інформації без рухомих частин.

TCO - tjänstemännens centralorganisation - міжнародний сертифікат якості та екологічної безпеки для електронного обладнання

WebSocket - протокол зв'язку, який забезпечує двонаправлений канал зв'язку між клієнтом і сервером через єдине TCP-з'єднання, що дозволяє обмінюватися даними в реальному часі.

БД - база даних

ВЦ - вичислювальний центр

ГДК - гранично допустима концентрація

ГІС - геоінформаційні системи

ДБН - державні будівельні норми

ДСанПіН - державні санітарні правила і норми

ДСТУ - державний стандарт України

MIMO - Multiple Input Multiple Output - системи зв'язку з рознесеними передавальними і приймальними антенами.

НПАОП - нормативно-правовий акт з охорони праці

ОВ - оптичне волокно

ПЗ - програмне забезпечення

ПК - персональний компютер

ВСТУП

Актуальність роботи. У сучасному світі мобільні телекомунікаційні мережі є фундаментом цифрової інфраструктури, яка забезпечує доступ до інтернету, обмін даними, голосовий та відеозв'язок, функціонування Інтернету речей (IoT), автономного транспорту, віддаленого навчання, медицини та багатьох інших сфер. Зі зростанням кількості користувачів та інтенсивністю трафіку виникає потреба у

забезпеченні стабільного, швидкого та безпечного зв'язку. Зокрема, з розгортанням мереж п'ятого покоління (5G) відбувається значне ускладнення архітектури мереж та зростання вимог до контролю якості їх функціонування.

Актуальність теми даної роботи зумовлена обмеженою доступністю універсальних інструментів, які дозволяли б ефективно тестувати параметри мобільних мереж, виконувати симуляції й аналізувати якість покриття в реальному часі. Більшість існуючих засобів є або надто складними у налаштуванні, або мають закриту архітектуру, що унеможлиблює їх інтеграцію у власні дослідницькі чи виробничі процеси. У той час як веб-технології демонструють значний потенціал у створенні кросплатформених застосунків, здатних до інтерактивного збирання, обробки та візуалізації телекомунікаційних даних без необхідності встановлення додаткового програмного забезпечення.

Створення спеціалізованого web-застосунку, що дозволяє проводити моделювання мережевих елементів, тестування базових станцій, візуалізацію оптичних з'єднань та відображення результатів на графіках і картах, є надзвичайно актуальним як для наукових досліджень, так і для комерційного використання. Такий підхід дозволяє значно підвищити прозорість і точність аналізу роботи мереж, забезпечити раннє виявлення потенційних проблем, проводити навчання та тестування для інженерів, а також покращити планування мережевої інфраструктури.

Крім того, актуальність проєкту посилюється потребою в побудові інструментів для автономного та хмарного моніторингу, які є стійкими до навантажень, масштабованими та легкими у впровадженні. У цьому контексті web-застосунок виступає універсальним рішенням, яке може бути використане операторами мобільного зв'язку, ІТ-компаніями, викладачами в освітньому процесі, а також студентами для моделювання сценаріїв та експериментального дослідження параметрів мобільної мережі.

Таким чином, розробка веб-застосунку для тестування мобільної мережі не лише відповідає вимогам сучасних технологій, а й сприяє формуванню цифрових

рішень майбутнього, що поєднують гнучкість, відкритість, ефективність та інтерактивність.

Аналіз останніх досліджень. Останні роки спостерігається активний розвиток у сфері дослідження мобільних мереж п'ятого покоління (5G) та інструментів для їхнього тестування. Вітчизняні та зарубіжні науковці приділяють велику увагу питанням моделювання мобільних мереж, аналізу покриття, якості обслуговування та ефективності використання радіочастотного спектра [1].

У наукових публікаціях розглядаються різні методи симуляції роботи базових станцій, зокрема із врахуванням таких параметрів, як потужність передавання, щільність користувачів, тип модуляції, затримки в мережі тощо. Зокрема, в роботі Nguyen H. C. - "Simulation of 5G base stations with user behavior and network parameters" запропоновано модель симуляції базових станцій з урахуванням просторових характеристик трафіку [2]. Крім того, у роботі Wang Z. "Journal of Optical Communications and Networking" аналізують вплив топології оптичної транспортної мережі на продуктивність телекомунікаційної інфраструктури [3]. Для візуалізації результатів широко використовуються геоінформаційні системи, про що свідчить дослідження Бондаренка І. С. , у якому розглядаються способи інтеграції ГІС в системи аналізу покриття [4].

Разом з тим, більшість існуючих рішень є або комерційними продуктами з обмеженим доступом, або складними у впровадженні програмними комплексами, що вимагають глибоких технічних знань. Це створює попит на інтуїтивно зрозумілі, веб-орієнтовані системи для моделювання та тестування мобільних мереж, які б поєднували наочність, функціональність та можливість розширення [5].

Таким чином, запропонована в даній роботі реалізація Web-застосунку відповідає актуальним науково-технічним тенденціям і продовжує лінію досліджень у напрямку створення зручних інструментів для аналізу та візуалізації роботи мобільної мережі.

Метою даної роботи є розробка web-застосунку для тестування мобільної мережі, що дозволяє здійснювати:

збір даних про параметри роботи мобільної мережі у режимі реального часу;
збереження результатів досліджень у базі даних для подальшого порівняльного аналізу.

візуалізація зібраних даних на графіках, діаграмах та інтегральній карті.

Для досягнення поставленої мети необхідно розв'язати наступні задачі:

розробити концепцію Web-застосунку для тестування базових станцій та оптичних з'єднань.;

реалізувати модуль симуляції оптичних з'єднань із візуалізацією параметрів каналу зв'язку;

розробити функціонал для візуалізації результатів тестування у вигляді графіків, таблиць та інтерактивної карти.

Об'єктом дослідження є процес тестування мобільних мереж, зокрема симуляція роботи базових станцій та оптичних з'єднань, а також методи візуалізації результатів тестування у вигляді графіків, таблиць та на інтерактивних картах.

Предметом дослідження є методи та інструменти для розробки Web-застосунку, що симулює тестування мобільної мережі, включаючи моделювання базових станцій, оптичних з'єднань та візуалізацію результатів тестування в графічній та табличній формах.

Новизна роботи полягає в розробці алгоритму та програмного забезпечення у вигляді інтерактивного Web-застосунку для симуляції тестування мобільних мереж. Запропонований алгоритм забезпечує моделювання роботи базових станцій та оптичних з'єднань з урахуванням основних параметрів мережі. Реалізована програма поєднує функції симуляції та візуалізації результатів у вигляді графіків, таблиць і на інтерактивній карті, що дозволяє проводити тестування в реальному часі, оперативно моніторити параметри та аналізувати ефективність мережі.

Практичне значення роботи полягає в тому, що розроблений web-застосунок для тестування мобільної мережі дозволяє підвищити ефективність моніторингу, аналізу та оптимізації телекомунікаційних систем.

Матеріали БДР обговорювалися на науковому семінарі кафедри ІКСТ у 2025 році та доповідались на LIV Всеукраїнській науково-технічній конференції та опубліковані в матеріалах конференції, а тези представлені в списку літератури [6].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОБҐРУНТУВАННЯ ВИБОРУ ТЕХНОЛОГІЇ

1.1. Сучасний стан мобільних мереж та їх тестування

Сьогодні мобільні мережі є ключовою складовою глобальної телекомунікаційної інфраструктури. Вони забезпечують бездротову передачу даних і голосу, що є основою для функціонування смартфонів, Інтернету речей

(IoT), автономних систем, хмарних сервісів та багатьох інших технологій. Станом на 2025 рік, у світі активно впроваджуються та вдосконалюються мережі п'ятого покоління (5G), які забезпечують значно вищу швидкість передавання даних, знижену затримку та здатність обслуговувати велику кількість пристроїв на одиницю площі порівняно з мережами попередніх поколінь (4G/LTE) [7].

З переходом до 5G виникає потреба у нових методах проектування, тестування та моніторингу мережі. Традиційні інструменти стають менш ефективними через збільшення складності інфраструктури, використання міліметрових хвиль, масиву антен (MIMO), beamforming та гібридних архітектур з використанням як радіоканалів, так і оптичних ліній зв'язку.

Тестування мобільної мережі охоплює як апаратні, так і програмні аспекти. Основні параметри, що підлягають перевірці, включають: якість сигналу (RSSI, RSRP, SINR), пропускну здатність, затримку, рівень втрат пакетів, ефективність хендоверу між базовими станціями, надійність оптичних з'єднань та інші [8]. Сучасне тестування часто здійснюється за допомогою комбінації програмного моделювання (симуляції) та натурних вимірювань з використанням спеціалізованого обладнання.

Значного поширення набули також віртуалізовані підходи до моделювання мобільних мереж, зокрема за допомогою програм-емуляторів, платформ з відкритим вихідним кодом (наприклад, ns-3, GNS3) та Web-інтерфейсів [9]. Це дозволяє проводити попередню оцінку параметрів мережі, виявляти «вузькі місця» у плануванні покриття або навантаження, а також швидко тестувати гіпотези без фізичного розгортання інфраструктури.

Отже, актуальність створення Web-застосунку для симуляції та візуалізації тестування мобільних мереж зумовлена потребою в доступних, гнучких і наочних інструментах, які б дозволяли фахівцям швидко аналізувати якість роботи мережі та приймати ефективні технічні рішення.

1.2. Параметри мобільної мережі, їх ефективність та оцінювання

Для забезпечення високої якості обслуговування в мобільних мережах необхідно комплексно оцінювати їх ключові параметри, що впливають на продуктивність і стабільність роботи.

Серед основних характеристик, які визначають ефективність мобільної мережі, виділяють пропускну здатність, затримку, рівень сигналу, швидкість передачі даних, коефіцієнт втрат пакетів та час безперервної роботи мережі.

Пропускна здатність визначає максимальний обсяг інформації, який може передаватися через мережу за одиницю часу, що є важливим для підтримки великої кількості користувачів та швидкого обміну даними, її величину можна розрахувати за виразом наведеним в [10]

$$C = S / t, \quad (1.1)$$

де C — пропускна здатність (bps); S — обсяг переданих даних (біт);

t — час передачі (сек).

Затримка відображає час передачі пакетів від відправника до отримувача і є критичною для додатків реального часу, таких як голосові дзвінки або відеоконференції, її величину можна розрахувати за виразом наведеним в [11]

$$D = D_{prop} + D_{trans} + D_{proc} + D_{queue}, \quad (1.2)$$

де D — загальна затримка; D_{prop} — затримка поширення; D_{trans} — час передачі; D_{proc} — час обробки; D_{queue} — час у черзі.

Рівень сигналу впливає на стабільність з'єднання і якість прийому, а швидкість передачі даних визначає фактичну продуктивність при завантаженні і відвантаженні інформації. Важливим параметром також є коефіцієнт втрат пакетів,

оскільки висока частка втрат негативно позначається на якості зв'язку, викликаючи переривання або спотворення сигналу.

Розрахувати рівень сигналу можна розрахувати за виразом наведеним в [12]

$$RSSI = P_t + G_t + G_r - L_p, \quad (1.3)$$

де P_t — потужність передавача; G_t і G_r — підсилення антен; L_p — втрати на шляху сигналу.

Швидкість передачі даних характеризує ефективність обміну інформацією між користувачем і мережею, що особливо важливо при перегляді відео або завантаженні файлів.

Швидкість передачі можна розрахувати за виразом наведеним в [13]

$$R = L / t, \quad (1.4)$$

де R — швидкість передачі; L — обсяг даних; T — час передачі.

Коефіцієнт втрат пакетів відображає надійність з'єднання, оскільки втрата пакетів призводить до зниження якості сервісу, зокрема під час відеодзвінків або потокової передачі, його розрахунок здійснюється за виразом наведеним в [14]

$$RLR = (N_{lost} / N_{send}) \cdot 100\%, \quad (1.5)$$

де N_{lost} — кількість втрачених пакетів; N_{send} — кількість відправлених пакетів.

Час безперервної роботи мережі є індикатором її надійності та доступності для користувача, що особливо важливо для критичних сервісів.

Коефіцієнт втрат пакетів можна розрахувати за виразом наведеним в [15]

$$U_{ptime} = (T_{total} / T_{work}) \cdot 100\%, \quad (1.6)$$

де T_{work} — час роботи мережі; T_{total} — загальний час спостереження.

Крім того, час безперервної роботи або доступність мережі відображає її надійність і готовність обслуговувати користувачів без перерв.

Для оцінювання ефективності мобільних мереж розглядаються різні покоління технологій, що відрізняються за рівнем розвитку та можливостями.

Мережі другого покоління (2G), що базуються на стандартах GSM та CDMA, забезпечували в основному голосовий зв'язок та передачу даних низької швидкості, що сьогодні вважається недостатнім для сучасних потреб.

Мережі третього покоління (3G) з технологіями UMTS та HSPA пропонують покращену швидкість передачі даних і підтримку мобільного Інтернету, однак їх ефективність поступається більш сучасним рішенням.

Четверте покоління (4G, LTE) характеризується значним підвищенням швидкості, зменшенням затримок та покращеною якістю обслуговування, що робить його домінуючою технологією для більшості користувачів.

Найсучаснішим стандартом є п'яте покоління (5G), яке забезпечує надвисоку пропускну здатність, мінімальні затримки та високу надійність, що відкриває нові можливості для розвитку Інтернету речей, автономних транспортних засобів та інших інноваційних застосунків [16].

Таким чином, оцінювання параметрів мобільної мережі та порівняння різних поколінь дозволяє визначити ступінь їх ефективності та актуальність для вирішення завдань сучасного зв'язку.

1.3. Існуючі програмні засоби для аналізу мобільної мережі

Сучасні мобільні мережі потребують постійного моніторингу, тестування та оптимізації. Для цього на ринку представлено широкий спектр програмних засобів, які дозволяють аналізувати якість сигналу, пропускну здатність, рівень затримки, надійність з'єднання та інші параметри, що безпосередньо впливають на якість обслуговування користувача.

Серед найбільш відомих інструментів, які використовуються операторами зв'язку та інженерами для тестування, можна виділити TEMS Investigation, Nemo Outdoor та Nemo Analyze [17-20]. Ці програмно-апаратні комплекси дозволяють здійснювати вимірювання мережі, збирати статистику щодо параметрів покриття, оцінювати якість передачі голосу та даних. Вони забезпечують високу точність результатів, однак вимагають дорогого обладнання та спеціальної підготовки персоналу.

Для моделювання та симуляції мобільних мереж активно застосовуються програмні середовища з відкритим кодом, зокрема NS-3 та OMNeT++ [21]. Ці платформи дозволяють створювати програмні моделі мобільних мереж, модулювати різноманітні сценарії навантаження, аналізувати поведінку протоколів, вивчати ефективність алгоритмів розподілу ресурсів. Незважаючи на високу гнучкість, використання подібних симуляторів часто потребує глибоких знань у галузі програмування та мережевих технологій.

Крім того, існують спеціалізовані рішення для розгортання тестових мереж, як-от Open5GS та srsRAN. Ці інструменти дозволяють створювати власні віртуальні або фізичні мобільні мережі покоління 4G або 5G з повним стеком протоколів. Вони активно використовуються в наукових дослідженнях, проте мають складний процес налаштування [22,23].

Для отримання інформації про розміщення базових станцій користувачі часто звертаються до онлайн-сервісів, таких як CellMapper. Цей веб-інструмент

використовує дані користувачів для побудови карти мобільного покриття різних операторів, що може бути корисним під час планування мереж або їх аналізу [24].

Не менш важливою є наявність мобільних застосунків, зокрема NetMonster, які дозволяють збирати телеметрію про поточні параметри підключення без потреби у складному обладнанні. Вони дають змогу фахівцям оперативно отримувати дані про частоту, ідентифікатор стільника, тип модуляції, рівень сигналу тощо [25].

Однак, попри наявність великої кількості програмних рішень, більшість з них не забезпечують зручної, інтерактивної візуалізації даних та моделювання у браузері. Бракує інструментів, що дозволяли б здійснювати симуляцію мережевих процесів, аналізувати зв'язки між базовими станціями та відображати результати у вигляді графіків, таблиць і карт у єдиному інтерфейсі. Це створює потребу у розробці Web-застосунку нового типу, що поєднає в собі функції симуляції, аналізу та візуалізації в реальному часі.

1.4. Переваги та недоліки існуючих рішень

На сучасному етапі розвитку телекомунікаційної галузі існує широкий спектр програмних рішень для тестування та аналізу мобільних мереж, кожне з яких має свої переваги та обмеження. Аналіз цих рішень дозволяє виявити основні напрямки для подальшого вдосконалення систем моніторингу та тестування.

До основних переваг професійних програмно-апаратних комплексів, таких як TEMS Investigation, Nemo Outdoor, Keysight UXM Wireless Tester, належить висока точність вимірювань, комплексність збору параметрів, підтримка сучасних стандартів мобільного зв'язку (LTE, 5G), а також наявність потужних аналітичних інструментів. Вони забезпечують глибоке розуміння процесів, що відбуваються в реальній мережі, дозволяють досліджувати якість зв'язку з урахуванням радіотехнічних та мережевих характеристик. Такі рішення широко застосовуються операторами мобільного зв'язку та дослідницькими організаціями [26,27].

Разом з тим, подібні інструменти мають і низку суттєвих недоліків. Насамперед, це висока вартість ліцензій та необхідність у спеціалізованому обладнанні. Крім того, процес збору та обробки даних є складним і потребує відповідної підготовки персоналу. Їх використання часто є доцільним лише у великих компаніях або у рамках науково-дослідних проєктів.

Щодо програмних засобів симуляції, таких як NS-3, OMNeT++, Open5GS чи srsRAN, їх перевагою є можливість створення віртуального середовища, що дозволяє досліджувати роботу мережі без фізичного доступу до обладнання. Такі інструменти широко використовуються в навчальних цілях та при розробці нових алгоритмів функціонування мереж. Їх відкритий код і модульність забезпечують високу гнучкість налаштування та розширення функціональності [28-31].

Однак для використання подібних систем необхідні глибокі знання у галузі комп'ютерних мереж, програмування, а також значний час для розгортання та тестування. Інтерфейс таких систем часто є низькорівневим, що ускладнює роботу нефахівцям. Відсутність інтерактивної графічної візуалізації також знижує їхню доступність для широкого кола користувачів.

Таким чином, незважаючи на наявність численних програмних засобів для аналізу мобільної мережі, жоден із них не забезпечує повноцінної інтеграції таких можливостей, як зручне відображення результатів у вигляді графіків, таблиць та інтегральних карт у веб-середовищі без необхідності встановлення. Це створює передумови для розробки нового типу Web-застосунку, що об'єднає переваги існуючих рішень та усуне їхні ключові недоліки.

1.5. Постановка задачі та обґрунтування вибору технологій

У сучасних умовах стрімкого розвитку мобільних технологій та впровадження нових стандартів зв'язку, зокрема 4G та 5G, постає нагальна потреба у створенні ефективних засобів тестування та аналізу мобільної мережі. Існуючі інструменти або є надто складними й дорогими у використанні, або ж обмежуються поверхневим збором даних без належної можливості глибокої

візуалізації та моделювання. У зв'язку з цим виникає актуальне завдання розробки Web-застосунку, що дозволить не лише здійснювати тестування роботи базових станцій та оптичних з'єднань, а й забезпечуватиме доступну та зрозумілу візуалізацію результатів тестування у вигляді графіків, таблиць і інтерактивної карти.

Поставлена задача передбачає створення зручного та інтуїтивного інтерфейсу, який дозволяє моделювати розташування базових станцій, отримувати ключові параметри, а також формувати структуру оптичних з'єднань. Отримані в процесі тестування дані повинні зберігатися у структурованій базі даних для подальшого аналізу, порівняння і виведення статистичних характеристик. Однією з ключових вимог до застосунку є його доступність у веб-браузері, що дозволяє користувачам працювати з системою незалежно від типу пристрою чи операційної системи.

Отже, сформульована задача охоплює побудову повноцінного веб-застосунку для симуляції, аналізу та візуалізації мобільної мережі. Обрані технології відповідають сучасним вимогам до продуктивності, масштабованості та зручності користування, що робить їх доцільними для реалізації поставленої мети.

1.5.1 Обґрунтування вибору технологій для реалізації клієнтської частини

Для реалізації клієнтської частини системи моніторингу мобільної мережі було розглянуто кілька популярних сучасних фреймворків, зокрема Angular, React та Vue.js. Кожен із них має власні переваги, архітектурні особливості та специфіку використання, проте, зважаючи на вимоги до проєкту, найбільш доцільним виявився вибір Angular [32].

React — це бібліотека, розроблена компанією Meta (Facebook), що фокусується переважно на створенні інтерфейсів користувача. React надає розробникам гнучкість і свободу у виборі підходів до архітектури, однак така свобода часто вимагає від команди самостійно обирати додаткові інструменти для

маршрутизації, управління станом, форм, валідації тощо. У проєктах великої складності це може призвести до фрагментації рішень, залежності від сторонніх бібліотек та збільшення часу на розробку, підтримку й уніфікацію коду. Крім того, для реалізації серверного рендерингу в React необхідно використовувати Next.js, що створює ще один додатковий шар абстракції [33].

Vue.js, у свою чергу, є прогресивним фреймворком, який поєднує у собі елементи, властиві як Angular, так і React. Він вирізняється простотою у вивченні, легкістю інтеграції в існуючі проєкти, а також зручним двостороннім зв'язком даних. Проте, попри свою гнучкість, Vue поступається Angular у масштабованості великих проєктів, особливо в тих випадках, коли потрібна чітка структура, стандартизація, офіційна підтримка корпоративного рівня та повноцінна екосистема «під ключ» [34].

Для реалізації клієнтської частини для тестування мобільної мережі було прийнято рішення використати сучасний фреймворк Angular, який динамічних і ефективних клієнтських застосунків. Angular, створений та підтримуваний компанією Google, має ряд переваг, які роблять його особливо привабливим для реалізації комплексних інтерфейсних рішень.

Однією з основних причин вибору Angular стала його реактивна архітектура, що дозволяє ефективно управляти станом застосунку, обробляти події у реальному часі, працювати з потоками даних (через RxJS), а також оновлювати інтерфейс без необхідності ручного втручання. Це особливо актуально для проєкту, в якому дані можуть змінюватися динамічно внаслідок взаємодії з користувачем або отримання нової інформації з сервера.

Важливою перевагою Angular є його модульна структура, що дозволяє розділяти застосунок на незалежні компоненти та модулі, спрощуючи як розробку, так і тестування, підтримку та масштабування проєкту. Такий підхід значно полегшує командну роботу над програмним забезпеченням, оскільки кожен модуль може розроблятися окремо з чітко визначеними інтерфейсами [35].

Ще однією суттєвою перевагою Angular є наявність вбудованого механізму маршрутизації, що забезпечує просте й ефективне управління навігацією між сторінками. Також Angular має потужний інструментарій для форм, валідації, міжнародалізації, а також офіційно підтримує TypeScript, що додає до розробки строгість типізації, спрощує відлагодження коду, підвищує його надійність та полегшує супровід.

Завдяки інтегрованій підтримці серверного рендерингу через Angular Universal, з'являється можливість реалізовувати SEO-оптимізовані застосунки, що особливо важливо у випадках, коли доступ до інтерфейсу може відбуватися через пошукові системи або зовнішні платформи [36].

Angular також має широку екосистему інструментів та бібліотек, включаючи Angular CLI — офіційний інструмент для генерації проєктів, автоматизації компіляції, тестування, збірки та деплою. Крім того, завдяки великій спільноті, Angular має потужну документаційну базу, постійно оновлюється, а також підтримує інтеграцію з сучасними інструментами розробки, такими як Webpack, ESLint, Prettier та інші [37].

Таким чином, вибір Angular як основи для розробки клієнтської частини був повністю обґрунтованим з урахуванням потреб проєкту у гнучкості, надійності, масштабованості та сучасному підході до побудови веб-застосунків. Angular забезпечив ефективне середовище для реалізації інтерфейсу користувача з підтримкою складної логіки, інтерактивних карт, взаємодії з сервером у реальному часі та відображення аналітичної інформації в зручному для користувача вигляді.

1.5.2 Обґрунтування вибору бібліотеки для реалізації інтегральної карти

Одним із ключових елементів у візуалізації результатів системи тестування мобільної мережі є побудова інтерактивної карти, яка дозволяє візуалізувати розташування базових станцій, оптичних з'єднань між ними, а також наочно відображати поточний стан мережевих об'єктів. Для цього було необхідно обрати

відповідну бібліотеку, яка б забезпечувала гнучке управління елементами карти, підтримку кастомних маркерів, можливість роботи з географічними координатами, інтерактивну взаємодію з елементами (наприклад, спливаючі вікна, обробка подій), а також мала хорошу продуктивність і легко інтегрувалася з Angular-застосунком. Також важливим критерієм була наявність відкритої ліцензії, що дозволяє безперешкодне використання інструменту у навчальних та дослідницьких цілях.

Серед можливих варіантів були розглянуті такі популярні картографічні бібліотеки як Google Maps API, Leaflet.js, OpenLayers та Mapbox GL JS. Після аналізу функціональних можливостей, складності інтеграції, продуктивності та гнучкості було прийнято рішення зупинитися на бібліотеці Leaflet.js. Вона вирізняється відкритою ліцензією (MIT), високою продуктивністю при роботі з великою кількістю об'єктів, широкими можливостями кастомізації та легко інтегрується з Angular-додатком без потреби у сторонніх обгортках або залежностях. Leaflet також дозволяє легко додавати маркери, полігони, лінії, а також спливаючі інформаційні вікна з динамічним вмістом [38].

Іншою перевагою Leaflet є велика кількість додаткових плагінів і прикладів у відкритому доступі, що дає можливість масштабувати функціонал карти у майбутньому, не змінюючи обрану технологію. Таким чином, бібліотека Leaflet.js була обрана як оптимальне рішення для побудови інтерактивної карти у дипломному проєкті, що забезпечило зручне та ефективне відображення геопросторової інформації в реальному часі.

1.6 Обґрунтування вибору технологій для реалізації серверної частини

Для реалізації серверної частини інформаційної системи моніторингу мобільної мережі було проведено аналіз сучасних технологій і фреймворків, що дозволяють ефективно працювати з API, базами даних, забезпечувати масштабованість, безпеку, простоту підтримки та гнучкість. Серед найбільш популярних рішень були розглянуті такі технології як Node.js (у поєднанні з

Express.js або NestJS), Django (Python), Spring Boot (Java), а також ASP.NET Core (C#). Після порівняння особливостей кожної з платформ, було прийнято рішення на користь NestJS як найбільш придатного інструменту для реалізації серверної частини проєкту [39].

На відміну від традиційного Express.js, який є мінімалістичним і потребує додаткової конфігурації для структурування великих проєктів, NestJS є повноцінним фреймворком, що побудований поверх Express (або Fastify) та використовує TypeScript за замовчуванням. NestJS наслідує принципи архітектури, подібні до Angular — модульність, ін'єкція залежностей, декларативність, що забезпечує високу організованість коду, зменшує залежність від зовнішніх бібліотек та пришвидшує розробку за рахунок готових рішень.

Однією з ключових переваг NestJS є його повна сумісність з TypeScript, що дозволяє підтримувати єдність технологій між клієнтською та серверною частинами. Це значно спрощує розробку, уніфікує підходи, полегшує комунікацію між командами, а також дозволяє повторно використовувати загальні типи, DTO (Data Transfer Object), інтерфейси та структури даних на обох кінцях. Такий підхід особливо ефективний у масштабних системах, де важливо зберігати цілісність типів та структур.

Крім того, NestJS надає широкі можливості для побудови REST та GraphQL API, підтримку WebSocket, мікросервісів, middleware, а також інтерфейси для взаємодії з базами даних через ORM (наприклад, TypeORM або Prisma). У даному проєкті була обрана база даних MySQL, яка добре інтегрується з NestJS за допомогою TypeORM. MySQL — це перевірене часом рішення з високою продуктивністю, широкими можливостями масштабування, підтримкою складних SQL-запитів, транзакцій і високою надійністю.

Для автентифікації користувачів у NestJS реалізовано підтримку механізмів авторизації через JWT (JSON Web Token), OAuth2, Google Login тощо. Це дозволяє легко реалізовувати захищені ендпоїнти, розмежування доступів, а також забезпечити безпечну інтеграцію з іншими сервісами.

З огляду на вимоги до швидкодії, масштабованості, підтримки сучасних стандартів безпеки, легкості обслуговування та можливості подальшого розширення, NestJS у поєднанні з Node.js, TypeScript і MySQL було обрано як оптимальну технологічну платформу для реалізації серверної частини системи. Такий вибір забезпечує ефективну обробку запитів, зручну структуру коду, надійність у роботі з даними та легку інтеграцію з фронтендом, реалізованим на Angular.

1.7 Висновки до розділу “Аналіз предметної області та обґрунтування вибору технологій”

У результаті аналізу предметної області та наявних технологій було здійснено ґрунтовне дослідження сучасного стану мобільних мереж, методів їхнього тестування, а також доступних програмних рішень для аналізу та візуалізації параметрів зв'язку. Проведене порівняння переваг і недоліків уже наявних інструментів дало змогу критично оцінити їх функціональність, масштабованість та придатність до реальних умов тестування мереж. Це дозволило не лише виявити актуальні технічні потреби, але й сформулювати чітке бачення майбутнього програмного рішення.

Особливу увагу було приділено обґрунтуванню вибору відповідних технологій для реалізації клієнтської, серверної частин, а також засобів інтерактивної візуалізації (наприклад, інтеграції з картографічними бібліотеками). Цей вибір відіграє ключову роль у забезпеченні стабільної роботи та високої продуктивності розробленого застосунку, що критично важливо в умовах великої кількості користувачів і високої частоти запитів. Таким чином, даний розділ заклав концептуальну та технологічну основу для розробки системи, що має не лише практичну цінність, а й потенціал до подальшого масштабування й адаптації до інших видів мережевих інфраструктур.

2 РОЗРАХУНКИ НАВАНТАЖЕНЬ І ПАРАМЕТРІВ ТЕЛЕКОМУНІКАЦІЙНОЇ МЕРЕЖІ

2.1 Оцінка кількості одночасних користувачів і навантаження на систему

Передбачається, що застосунок буде використовуватись групою моніторингу, де одночасно може працювати до 20 працівників, а також передбачимо можливість того, що всі ці користувачі будуть одночасно надсилати запити до сервера для запуску тестів мобільної мережі, отримання результатів та візуалізації даних. Припустимо, що кожен користувач генерує запит до сервера кожні 15 секунд, що є типовим інтервалом між інтерактивними діями користувача в тестувальній системі.

Визначимо кількість запитів на хвилину можна за допомогою виразу наведеним в [39]

$$Q = N \cdot (60 / T), \quad (2.1)$$

де N - це кількість запитів на хвилину; T - довжина інтервалу в с;

Підставивши потрібні значення у формулу отримаємо:

$$Q = 20 \cdot (60 / 15) = 80 \text{ запитів / хв}$$

Обсяг переданих даних в одному запиті, включно з JSON-відповіддю, графіками, метаданими та частковими оновленнями інтерфейсу, становить приблизно 120 КБ. Це значення є середнім для застосунків, які відображають результати з використанням клієнтських фреймворків на кшталт Angular, які активно взаємодіють з REST або GraphQL API.

Обсяг трафіку на хвилину розраховується за допомогою виразу наведеним в [40]

$$V_{хв} = Q \cdot S \quad (2.2)$$

де Q - це кількість запитів на хвилину; S - приблизний обсяг переданих даних в одному запиті;

Підставивши потрібні дані отримаємо:

$$V_{хв} = 80 \cdot 120 = 9600 \text{ КБ або } 9,6 \text{ МБ}$$

Тепер визначимо обсяг трафіку на годину за виразом, що наведений в [40]:

$$V_{год} = V_{хв} \cdot 60 \quad (2.3)$$

Підставивши потрібні нам дані отримаємо:

$$V_{год} = 9,6 \cdot 60 = 576 \text{ МБ}$$

Це означає, що при безперервній роботі з активною взаємодією, сервер буде змушений обробляти близько 576 МБ вхідних та вихідних даних на годину, не враховуючи додаткових навантажень на БД, кеш або логування. У масштабах дня (наприклад, 6 годин) ця цифра зростає до понад 3,5 ГБ. Це вимагає стабільного з'єднання, сучасної інфраструктури та продуманої архітектури кешування, аби уникнути перевантажень системи [40].

Таким чином, розрахунок кількості користувачів та обсягу генерованого трафіку дозволяє сформулювати об'єктивні вимоги до серверної частини, налаштувати обмеження частоти запитів та виявити потенційні вузькі місця у роботі Web-застосунку.

2.2 Вимоги до каналу зв'язку

Одним з найважливіших факторів, що впливають на стабільність роботи Web-застосунку для тестування мобільної мережі, є пропускна здатність каналу передачі даних. При розрахунку вимог до каналу зв'язку необхідно враховувати як обсяг даних, що передається між клієнтом і сервером, так і частоту запитів, кількість одночасних користувачів, а також характер інформації (текст, таблиці, графіки, карти).

Згідно з розрахунками, проведеними у підрозділі 2.1, годинне навантаження на серверну частину становить приблизно 576 МБ трафіку. Це лише частина обсягу, який повинен обробити канал зв'язку, оскільки також мають враховуватись супутні системні запити, перезапиту, оновлення сторінок, завантаження статичних ресурсів (скриптів, стилів, зображень) тощо. У реальній експлуатації фактичне навантаження на канал зв'язку може бути на 25–30% більшим, ніж розрахункове. Отже, оцінку слід проводити з запасом.

Для коректного функціонування Web-застосунку потрібно розрахувати мінімальну пропускна здатність каналу за виразом представленим в [41]

$$R = (V_{\text{год}} \cdot N) / t \quad (2.4)$$

де $V_{\text{год}}$ - це обсяг трафіку на годину; t - тривалість у секундах; N - кількість байтів в одному біті;

Підставивши дані у формулу отримаємо:

$$R = (576 \cdot 8) / 3600 = 1,28 \text{ Мбіт/с}$$

З урахуванням резерву на пікові навантаження, рекомендована пропускна здатність повинна бути не меншою за 10 Мбіт/с. Це дозволить уникнути затримок при одночасному обслуговуванні 20 користувачів, забезпечити коректну відправку та отримання запитів без втрат пакетів, затримок при візуалізації графіків та проблем із оновленням інтерфейсу користувача.

Також слід враховувати розподіл трафіку вхідного та вихідного. У застосунку для тестування мобільної мережі передача даних відбувається здебільшого з сервера до клієнта (тобто вихідний трафік з точки зору сервера). Тому саме вихідна смуга пропускання має бути більшою або рівною середньому обсягу завантаження даних клієнтами. Для корпоративних середовищ, де застосунок розгорнутий локально, варто враховувати обмеження внутрішньої мережі та параметри маршрутизаторів і комутаторів.

Крім того, при виборі хостингу або VPS-провайдера доцільно звертати увагу на гарантії швидкості з'єднання, наявність SLA (Service Level Agreement), що підтверджує мінімальний гарантований рівень доступу до мережі. Також важливим є наявність систем моніторингу каналу, які дозволяють відстежувати трафік, виявляти вузькі місця та забезпечити масштабованість інфраструктури в разі зростання навантаження [42].

Отже, канал зв'язку має забезпечувати передачу даних зі швидкістю не менше 10 Мбіт/с із можливістю динамічного масштабування або резервування, що

є критично важливим для забезпечення безперебійної роботи системи при інтенсивному навантаженні.

2.3 Розрахунок навантаження на сервер

Для забезпечення стабільної роботи Web-застосунку, особливо у випадку одночасної взаємодії багатьох користувачів, необхідно оцінити навантаження на серверну частину системи. Це включає обробку вхідних запитів, зберігання та обробку даних, генерацію графіків та карт, а також підтримку клієнтських сесій. Сервер повинен мати достатній обсяг оперативної пам'яті, обчислювальну потужність процесора, пропускну здатність шини та швидкодію файлової системи.

Перш за все, слід оцінити кількість запитів, які надходять на сервер. Як зазначено у підрозділі 2.1, при 20 одночасних користувачах, які надсилають запити кожні 15 секунд, генерується 80 запитів на хвилину або 4800 запитів на годину. Якщо припустити, що кожен запит обробляється за 50–100 мілісекунд, то сервер повинен виконувати щонайменше 4 запити щосекунди без затримок. Це відповідає базовим вимогам для застосунків середнього навантаження.

Обсяг оперативної пам'яті, що споживається кожною сесією користувача, залежить від складності запитів і обсягу проміжних даних. Для Web-застосунку, який генерує графіки, інтерактивні карти та виконує обробку аналітики, середній обсяг споживання пам'яті становить 50–80 МБ на одного користувача [43].

Розрахувати необхідний обсяг оперативної пам'яті можна за допомогою виразу [43]:

$$RAM = N \cdot S \quad (2.5)$$

де N - кількість запитів що надсилаються одночасно; S - середнє споживання оперативної пам'яті одним користувачем.

Підставивши дані у формулу отримаємо:

$$RAM = 20 \cdot 80 = 1600 \text{ МБ} = 1,6 \text{ ГБ}$$

З урахуванням додаткового навантаження з боку операційної системи, бази даних, кешування, логування та буферизації, рекомендований обсяг оперативної пам'яті для серверної частини має становити не менше 4 ГБ.

Щодо процесора, враховуючи середній час обробки запиту (100 мс) та необхідність багатопоточності, мінімальна конфігурація повинна включати щонайменше 4 логічних ядра з тактовою частотою від 2,4 ГГц. Цього достатньо для рівномірного розподілу навантаження та запобігання «вузькому місцю» при обробці одночасних запитів.

З точки зору зберігання даних, використання традиційних HDD може створювати затримки при читанні/записі даних, особливо у випадках з високою кількістю одночасних звернень. Тому рекомендується використовувати SSD з швидкістю читання/запису не менше 500 МБ/с. Загальний обсяг дискового простору залежить від того, чи зберігаються результати тестів, історія запитів, користувацькі дані та лог-файли. Для базового функціонування достатньо 100–150 ГБ SSD-накопичувача.

Важливим також є використання системи кешування (наприклад, Redis або Memcached), яка дозволяє зменшити навантаження на базу даних при повторному запиті одних і тих самих даних. Це особливо ефективно для візуалізацій, які часто повторюються (наприклад, ті самі графіки на різних пристроях або у різних користувачів).

Отже, для нормальної роботи Web-застосунку у середовищі з навантаженням до 20 користувачів необхідний сервер із процесором на 4 ядра, 4 ГБ оперативної пам'яті, SSD-диском обсягом від 100 ГБ і підтримкою кешування. Така конфігурація дозволить уникнути затримок, збоїв і забезпечить надійну роботу системи в реальному часі.

2.4 Розрахунок енергоспоживання сервера

Енергоспоживання є важливим фактором при плануванні експлуатаційних витрат Web-застосунку, особливо якщо серверне обладнання функціонує в режимі 24/7. Постійна робота серверів потребує надійного енергопостачання та обґрунтованого розрахунку споживаної електроенергії, що безпосередньо впливає на загальну вартість володіння інфраструктурою.

Для розрахунку візьмемо типову конфігурацію сучасного віртуального або фізичного сервера із такими параметрами:

- процесор із 4 ядрами (2,4–3,2 ГГц);
- 4 ГБ оперативної пам'яті;
- SSD-диск 100–200 ГБ;
- активне з'єднання з мережею та щонайменше 1 віртуальна машина.

Середнє енергоспоживання такої конфігурації становить орієнтовно 80–100 Вт/год. Для розрахунків візьмемо середнє значення — 90 Вт. Це враховує не тільки споживання основних компонентів, а й втрати на блок живлення, охолодження (активне повітряне) та інші внутрішні вузли системи [44].

Обчислити енергоспоживання за добу можна за допомогою виразу що представлений в [44]

$$W_{\text{д}} = P \cdot t \quad (2.6)$$

де P - споживана потужність пристрою; t - тривалість роботи за добу.

Підставивши дані отримаємо:

$$W_{\text{д}} = 90 \cdot 24 = 2\,160 \text{ Вт} \cdot \text{год} = 2,16 \text{ кВт} \cdot \text{год}$$

Розрахувати скільки кВт споживатиме сервер а місяць можна за допомогою виразу що представлений в [44]:

$$W_{\text{м}} = W_{\text{д}} \cdot 30 \quad (2.7)$$

Підставимо дані у формулу та отримаємо:

$$W_m = 2,16 \cdot 30 = 64,8 \text{ кВт} \cdot \text{год}$$

За тарифом 5 грн/кВт·год (типовий тариф для юридичних осіб або хмарних дата-центрів в Україні): Вартість енергії = $64,8 \cdot 5 = 324$ грн/міс

Крім основного споживання, потрібно враховувати втрати на кондиціонування приміщення (при використанні фізичного сервера), резервне живлення (ІБП), комутаційне обладнання. В середньому, додаткові елементи інфраструктури додають 20–30% до загального енергоспоживання. Отже згідно виразу що представлений в [44]:

$$W_{\text{заг}} = W_m + (0,3 \cdot W_m) \quad (2.8)$$

Підставивши дані у формулу отримаємо:

$$W_{\text{заг}} = 64,8 + 0,3 \cdot 64,8 = 64,8 + 19,44 = 84,24 \text{ кВт} \cdot \text{год}$$

Остаточна вартість експлуатації з урахуванням усіх втрат: Вартість = $85 \cdot 5 = 425$ грн/міс

Цей показник є важливим при виборі між власною серверною інфраструктурою та хмарним середовищем. У випадку використання VPS-платформи, як-от Hetzner, DigitalOcean або AWS, енергоспоживання вже враховано у вартості оренди. Проте, при розгортанні на локальному сервері чи в навчальному закладі, ці витрати необхідно враховувати окремо.

Враховуючи середньомісячну кількість користувачів (припустимо, 20), собівартість енергії на одного користувача складе: $425 / 20 = 21,25$ грн/міс

Таким чином, витрати на енергоживлення серверного обладнання в масштабах одного місяця є помірними й не становлять істотного навантаження на бюджет організації. Це дозволяє ефективно використовувати Web-застосунок у режимі постійної доступності та високої продуктивності без необхідності значного резервування ресурсів.

2.5 Розрахунок пропускної здатності оптичного каналу

У сучасних телекомунікаційних системах, особливо в умовах передачі великого обсягу даних, оптичні канали займають провідне місце завдяки своїй високій пропускній здатності, надійності та енергоефективності. У контексті даного Web-застосунку, який оперує симульованими даними мобільної мережі, доцільно враховувати характеристики оптичних каналів, що використовуються для передавання сигналів між базовими станціями та серверами або дата-центрами. Це дає змогу моделювати параметри, які максимально наближені до реальних умов.

Пропускна здатність одного волоконно-оптичного каналу може змінюватися залежно від використовуваної технології, типу передавача (лазер, світлодіод), модуляції та довжини хвилі. Для розрахунків приймемо стандартне значення, яке використовується в більшості комерційних систем — 10 Гбіт/с на одне волокно. Це значення типове для одномодового волокна з підтримкою 10G Ethernet.

Якщо в моделі застосовується мультиплексування або паралельне використання кількох каналів, то загальна пропускна здатність визначається як сума пропускних здатностей усіх каналів, згідно виразу представленого в [44]:

$$C_{\text{заг}} = C_{\text{од}} \cdot N \quad (2.9)$$

де $C_{\text{од}}$ — пропускна здатність одного каналу (Гбіт/с); N — кількість оптичних каналів;

При $N = 5$:

$$C_{\text{заг}} = 10 \cdot 5 = 50 \text{ Гбіт/с}$$

Цього показника достатньо для одночасного обслуговування понад 1000 активних користувачів із середнім навантаженням у 5–10 Мбіт/с кожен. Така конфігурація є характерною для регіональних вузлів операторів мобільного зв'язку та великих навчальних/наукових центрів.

Однак для повноти картини варто врахувати, що фактична пропускна здатність може знижуватися через втрати сигналу, відбиття, погіршення якості з'єднань, а також через обмеження комутаційного обладнання. У деяких випадках оператори впроваджують QoS (Quality of Service) та пріоритетність трафіку, що впливає на доступну смугу пропускання.

Також варто врахувати резервування каналів (наприклад, використання схеми 4+1), коли один канал використовується виключно як резервний. У цьому випадку активне навантаження розподіляється на 4 канали, а п'ятий канал активується у випадку збою або перевищення допустимого навантаження.

У підсумку, при моделюванні систем передачі даних у рамках Web-застосунку для тестування мобільної мережі доцільно закладати технічну можливість масштабування до 50–100 Гбіт/с, з можливістю розширення кількості каналів. Це забезпечить відповідність моделі сучасним вимогам до пропускної здатності магістральних та доступових мереж у мобільному зв'язку.

2.6 Розрахунок загасання в оптичному волокні

Одним із ключових параметрів, що впливають на якість передавання сигналів в оптичних лініях зв'язку, є загасання (атенюація). Воно характеризує зменшення потужності оптичного сигналу при його проходженні через волоконно-оптичну лінію і вимірюється в децибелах на кілометр (дБ/км) [44].

Основні причини загасання:

- Поглинання світла матеріалом волокна, яке виникає внаслідок внутрішніх процесів у скла або домішок у матеріалі;
- Розсіювання Релея, що пов'язане з мікроскопічними неоднорідностями в структурі скла;
- Зовнішні втрати, зумовлені вигинами, стиками, з'єднаннями та мікродефектами у волокні.

Обчислити загасання можна за допомогою виразу представленого в [44]:

$$\alpha = (10 / L) \cdot \log_{10} \cdot (P_{\text{вх}} / P_{\text{вих}}) \quad (2.10)$$

де α - загасання, дБ/км; L - довжина волокна, км; $P_{\text{вх}}$ - потужність сигналу на вході, мВт; $P_{\text{вих}}$ — потужність сигналу на виході, мВт.

У сучасних оптичних волокнах загасання залежить також від довжини хвилі переданого сигналу.

Визначити залежність загасання від довжини хвилі можна за допомогою виразу [44]:

$$\alpha = \alpha_{\text{поглин}}(\lambda) + \alpha_{\text{розс}}(\lambda) + \alpha_{\text{доод}}(\lambda) \quad (2.11)$$

де $\alpha_{\text{поглин}}(\lambda)$ - втрати через поглинання; $\alpha_{\text{розс}}(\lambda)$ - втрати через розсіювання; $\alpha_{\text{доод}}(\lambda)$ - додаткові втрати (механічні дефекти, з'єднання, вигини тощо).

2.7 Висновки до розділа “Розрахунки навантажень і параметрів телекомунікаційної мережі”

У другому розділі було проведено комплексні інженерні розрахунки, які мають вирішальне значення для забезпечення надійності та ефективності функціонування web-застосунку. Аналіз навантаження на систему, кількості одночасних користувачів, пропускну здатності каналів зв'язку, а також енергоспоживання серверної інфраструктури дозволив не лише оцінити поточні

ресурси, але й передбачити потенційні вузькі місця під час масштабування системи.

Результати розрахунків стали основою для створення оптимальної архітектури застосунку. Зокрема, обґрунтовано вибір потужності серверів, необхідної кількості ресурсів пам'яті, а також характеристик оптичного волокна, що має використовуватись як передавальне середовище. Це забезпечує стабільну роботу системи в умовах великого потоку даних та знижує ризик втрат через перевантаження чи обмеження пропускної здатності. Таким чином, цей розділ дозволяє побудувати застосунок не тільки як функціональний інструмент, але і як стійку, надійну платформу для моніторингу й аналізу мережевої інфраструктури.

3 ПРОЄКТУВАННЯ WEB-ЗАСТОСУНКУ

3.1 Архітектура системи

Розроблена система для тестування мобільної мережі побудована за принципами трирівневої клієнт-серверної архітектури, що забезпечує ефективну взаємодію між користувачем і обчислювальними модулями, а також гнучкість масштабування та модернізації в майбутньому. Архітектура поділена на три основні рівні: інтерфейс користувача (frontend), серверна логіка (backend) та база даних.

На рівні інтерфейсу користувача реалізовано зручне і доступне середовище взаємодії з системою, розроблене з використанням фреймворку Angular. Користувач має змогу протестувати мережу та переглядати результати моделювання у вигляді графіків, діаграм, таблиць та інтерактивної карти [45].

Серверна частина, реалізована за допомогою NestJS — сучасного backend-фреймворку на платформі Node.js, відповідає за логіку обробки запитів, взаємодію з базою даних та формування відповіді на різні види запитів. В основі серверної частини лежить модульна архітектура, яка дозволяє ефективно розподіляти відповідальність між різними компонентами, зменшувати зв'язність коду та забезпечувати легке розширення функціоналу [46,47].

База даних реалізована на основі MySQL і розміщена на хмарному хостингу Railway, що забезпечує стабільний доступ, масштабованість і зручне адміністрування. Структура бази даних оптимізована для швидкого доступу до потрібних даних, забезпечення цілісності та логічного зв'язку. У ній зберігається інформація про базові станції, оптичні лінії, результати тестів та облікові записи користувачів [48,49].

Зв'язок між frontend та backend реалізований за допомогою REST API, що дає змогу стандартизувати обмін даними між клієнтом і сервером у форматі JSON. Це спрощує взаємодію між компонентами системи, забезпечує розширюваність, безпеку та зручність при інтеграції з іншими сервісами або мобільними додатками [50].

Таким чином, архітектура системи відповідає сучасним вимогам до web-застосунків, забезпечуючи продуктивність, надійність, масштабованість і простоту в обслуговуванні. Вона створює надійну основу для моделювання тестування мобільної мережі, збору та візуалізації результатів, а також подальшого розвитку системи в межах наукових або прикладних досліджень.

3.2. Функціональні вимоги до застосунку

Розроблений web-застосунок для тестування мобільної мережі повинен відповідати низці функціональних вимог, які визначають його основні можливості та забезпечують досягнення поставленої мети. Основною вимогою до системи є забезпечення можливості проведення моделювання та виведення результатів без безпосереднього втручання у параметри мережі. Користувач має змогу запускати тестування мережі з використанням фіксованих або попередньо сформованих даних про базові станції, оптичні з'єднання, географічне розташування об'єктів та технічні характеристики сигналу. Застосунок виконує тестування мережі і на основі отриманих результатів цього формує візуальні та табличні результати, такі як карти покриття, діаграми показників якості зв'язку, діаграми продуктивності тощо. Уся взаємодія з користувачем зосереджена на перегляді результатів, без можливості змінювати структуру або конфігурацію мережі зсередини інтерфейсу. Крім того, передбачено механізм авторизації користувачів для надання персоніфікованого доступу до історії тестувань, а також можливість збереження і порівняння результатів між різними симуляціями. Такий підхід дозволяє зосередити функціональність системи на аналізі ефективності мобільної мережі, забезпечуючи простоту використання, наочність та цілісність тестових даних.

3.3. Принципи побудови клієнт-серверної взаємодії

Клієнт-серверна архітектура є основою розробленого web-застосунку, що забезпечує розділення відповідальностей між користувацьким інтерфейсом (клієнтом) та обробкою запитів і збереженням даних (сервером). Такий підхід дозволяє забезпечити масштабованість, підтримуваність та зручність розвитку системи.

Клієнтська частина реалізована з використанням фреймворку Angular, який відповідає за відображення інтерфейсу, взаємодію з користувачем, побудову графіків, таблиць та відображення карти з елементами мережі. Angular надсилає HTTP-запити до серверної частини через REST API для отримання або надсилання даних.

Серверна частина створена на основі Node.js із використанням фреймворку NestJS, що дозволяє організувати структуру проєкту за принципами модульності та інверсії залежностей. Сервер обробляє запити, виконує необхідні обчислення або моделювання, звертається до бази даних для зберігання або отримання інформації, а потім повертає відповідь клієнту у форматі JSON.

Обмін даними між клієнтом і сервером побудований на основі REST-архітектури, що забезпечує прозорість, повторне використання ресурсів і простоту масштабування. Кожен об'єкт або ресурс (наприклад, базова станція, тестування, результати) має свій маршрут, через який до нього можна отримати доступ або внести зміни.

Такий підхід дозволяє відокремити логіку відображення даних від логіки їх обробки, що покращує продуктивність розробки, підвищує безпеку системи та дає змогу легко інтегрувати нові компоненти або зовнішні сервіси у майбутньому.

3.4 Візуалізація елементів мобільної мережі

У розробленому web-застосунку реалізовано візуалізацію основних елементів мобільної мережі, таких як базові станції та оптичні з'єднання. Кожна базова станція відображається на карті у вигляді мітки з відповідними параметрами, що включають географічні координати, назву, покоління мережі, IP-адресу, оператора зв'язку, швидкість з'єднання та статус роботи (наприклад, "Не працює"). Такий підхід дозволяє наочно відображати технічні характеристики та стан базових станцій безпосередньо на географічній карті.

Оптичні з'єднання відображаються на інтегрованій карті та мають такі параметри, як маршрут з географічними координатами початку та кінця з'єднання, довжина траси у кілометрах, тип волоконно-оптичного кабелю, робоча частота, пропускна здатність та рівень загасання сигналу. Для візуального відображення маршрути оптичних ліній позначаються кольорами, наприклад зеленим якщо волокно працює і не пошкоджене, жовтим є пошкодження а червоним якщо критично пошкоджене, це дозволяє легко орієнтуватися на інтерактивній карті та приймати відповідні рішення.

Для обробки і збереження цих даних використовується серверна частина застосунку, яка формує структуровані моделі та передає їх клієнтській частині у форматі JSON. Це дозволяє відтворити реальні умови роботи мережі та провести комплексний аналіз її ефективності.

3.5 Висновки до розділу “Проектування web-застосунку”

Третій розділ присвячено проектуванню архітектури web-застосунку, що є критично важливим етапом у розробці будь-якого складного ПЗ. У результаті було реалізовано клієнт-серверну архітектуру з чітким поділом обов'язків між компонентами системи. Це дозволяє досягти високого рівня масштабованості, зручності супроводу та розширення функціоналу без ризику порушення роботи інших модулів.

Описані функціональні вимоги, що були сформульовані на основі попереднього аналізу, дозволили чітко визначити ключові сценарії взаємодії користувача з системою, включно з тестуванням мобільної мережі, візуалізацією даних та доступом до інтерактивної карти. Це має особливу значущість для забезпечення інтуїтивної роботи системи, що безпосередньо впливає на якість прийняття рішень у процесі мережевого моніторингу. Висновки цього розділу забезпечують цілісне розуміння функціонального дизайну системи й підтверджують її відповідність поставленим задачам.

4 РОЗРОБКА ТА РЕАЛІЗАЦІЯ WEB-ЗАСТОСУНКУ

4.1 Візуалізація компонентів web-застосунку

Користувацький інтерфейс web-застосунку побудований відповідно до класичної трисекційної структури: Header (заголовок), Main (основна частина) та Footer (підвал). Такий підхід забезпечує логічну організацію контенту, зручність навігації та уніфікований вигляд сторінки на різних пристроях.

Секція Header розташована у верхній частині сторінки і виконує функцію глобального навігаційного меню. Вона містить три основні навігаційні елементи для перемикавання між ключовими розділами застосунку — такими як "Тестування мережі", "Статистика" та "Карта оптичних з'єднань".

Основна частина сторінки (Main) динамічно змінюється в залежності від обраного розділу. Тут відображаються головна сторінка на якій відбувається тестування мережі, статистика по проведеним тестуванням та карта оптичних з'єднань на якій відображаються результати тестування.

Секція Footer знаходиться в нижній частині сторінки та виконує роль підвалу сторінки.

Вигляд головної сторінки та її позиціонування можна побачити на рисунку 3.1.

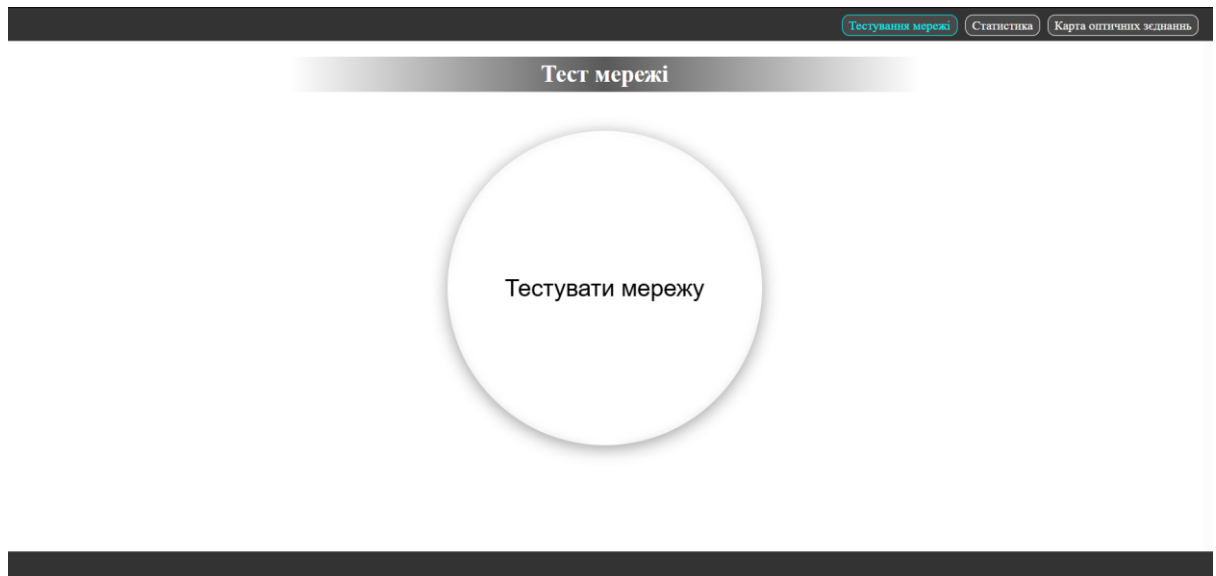


Рисунок 4.1 - Головна сторінка

На рисунку 4.1 можна побачити зазначені вище секції header, main та footer, у секції main на головній сторінці користувача зустрічає напис “Тестування мережі” а також кнопка яка виконує функцію що виконує тестування мережі, код даної функції представлений у Додатку Б.

Структуру та вигляд у якому подаються результати тестування можна побачити на Рисунку 4.2.

Результат							
Результат тестування Базових станцій							
Uci оператори		LifeCell		Kyivstar		Vodafone	
Категорія стану	Кількість	Категорія стану	Кількість	Категорія стану	Кількість	Категорія стану	Кількість
Протестовано БС	40	Протестовано БС	14	Протестовано Kyivstar БС	7	Протестовано Vodafone	17
Всього працює	30	Всього працює	11	Всього працює	7	Всього працює	12
Всього тимчасово не працює	0	Всього тимчасово не працює	0	Всього тимчасово не працює	0	Всього тимчасово не працює	0
Всього не працює	10	Всього не працює	3	Всього не працює	2	Всього не працює	5
БС		Покладання	Оператор	Швидкість	IP	Стан	
Базова Станція м. Вінниця		5G	Vodafone	300 Mб/с	192.168.0.1	Працює	
Базова Станція м. Луцьк		5G	Vodafone	400 Mб/с	192.168.0.2	Працює	
Базова Станція м. Дніпро		5G	Kyivstar	300 Mб/с	192.168.0.3	Працює	
Базова Станція м. Житомир		4G	Kyivstar	100 Mб/с	192.168.0.5	Працює	
Базова Станція м. Ужгород		4G	Lifecell	80 Mб/с	192.168.0.6	Працює	
Базова Станція м. Запоріжжя		4G	Lifecell	90 Mб/с	192.168.0.7	Працює	
Базова Станція м. Івано-Франківськ		5G	Vodafone	300 Mб/с	192.168.0.8	Працює	
Базова Станція м. Київ		5G	Vodafone	400 Mб/с	192.168.0.9	Працює	
Базова Станція м. Кропивницький		4G	Lifecell	100 Mб/с	192.168.0.10	Працює	
Базова Станція м. Львів		4G	Vodafone	100 Mб/с	192.168.0.12	Працює	
Базова Станція м. Миколаїв		5G	Lifecell	300 Mб/с	192.168.0.13	Працює	
Базова Станція м. Одеса		4G	Vodafone	100 Mб/с	192.168.0.14	Працює	
Базова Станція м. Полтава		5G	Lifecell	350 Mб/с	192.168.0.15	Працює	

Рисунок 4.2 - Результат тестування базових станцій

На рисунку 4.2 можна побачити що три кнопки зверху а саме кнопка що відповідає за повторне тестування, та дві кнопки для перемикання між результатами по базовим станціям та результатами по оптичним волокнам а також сам результат тестування.

Розглянемо детальніше результати тестування по базовим станціям, результати складаються із 5 таблиць перші 4 таблиці містять дані по кількості протестованих базових станцій та їх статуси тобто користувач може побачити кількість всього протестованих базових станцій, кількість базових станцій що працює, кількість базових станцій що тимчасово не працює та кількість базових станцій що не працює, такі дані представлені у 4 екземплярах тобто для всіх операторів, для оператора lifecell, для оператора kyivstar, та для оператора vodafone.

Нижче під 4 таблицями можна побачити одну велику таблицю у якій зібрана інформація по кожній базовій станції а саме її назва, покоління, оператор, швидкість (пропускна здатність), IP, та її стан. Якщо комірка таблиці має зелений колір заднього фону це свідчить про те що ця базова станція працює, якщо ж задній фон комірки таблиці має оранжевий колір це свідчить про те що дана базова станція тимчасово не працює, тобто на цій базовій станції можуть проводитися планові роботи тобто може відбуватися заміна обладнання або ж ремонт поточного обладнання і так далі. Якщо колір комірки червоний це свідчить про те що дана базова станція не працює через значні пошкодження та не підлягає швидкому ремонту.

Тепер розглянемо інші результати тестування які були отримані а саме результати тестування по оптичним волокнам на Рисунку 4.3

Тестування мережі							
Статистика							
Карта оптичних з'єднань							
Тестувати ще раз							
Результат по БС							
Результат по ОВ							
Результат							
Результат тестування Оптичних волокон							
Категорія стану				Кількість			
Протестовано ОВ				52			
Всього працює				36			
Потрачено				1			
Всього не працює				15			
Від	До	Фізична довжина	Тип волокна	Довжина хвилі	Пропускна здатність	Максимальне загасання регенераційної ділянки	Стан
Луцьк	Львів	150 км	Одномодове волокно	1550 нм	65 Gbps	34 дБ	Працює
Львів	Ужгород	270 км	Одномодове волокно	1550 нм	40 Gbps	61 дБ	Працює
Ужгород	Чернівці	160 км	Одномодове волокно	1550 нм	60 Gbps	36 дБ	Працює
Львів	Івано-Франківськ	130 км	Одномодове волокно	1550 нм	70 Gbps	30 дБ	Працює
Івано-Франківськ	Тернопіль	120 км	Одномодове волокно	1550 нм	70 Gbps	27 дБ	Працює
Луцьк	Рівне	80 км	Одномодове волокно	1550 нм	75 Gbps	18 дБ	Працює
Рівне	Коростень	150 км	Одномодове волокно	1550 нм	65 Gbps	34 дБ	Працює
Рівне	Житомир	130 км	Одномодове волокно	1550 нм	70 Gbps	30 дБ	Працює
Тернопіль	Хмельницький	100 км	Одномодове волокно	1550 нм	75 Gbps	23 дБ	Працює
Тернопіль	Чернівці	170 км	Одномодове волокно	1550 нм	60 Gbps	39 дБ	Працює
Чернівці	Кам'янець-Подільський	110 км	Одномодове волокно	1550 нм	65 Gbps	25 дБ	Працює
Хмельницький	Кам'янець-Подільський	50 км	Одномодове волокно	1550 нм	80 Gbps	12 дБ	Працює
Хмельницький	Вінниця	90 км	Одномодове волокно	1550 нм	75 Gbps	21 дБ	Працює
Вінниця	Умань	70 км	Одномодове волокно	1310 нм	80 Gbps	23 дБ	Працює

Рисунок 4.3 - Результати тестування оптичних волокон

На рисунку 4.3 схожі результати що і для базових станцій а саме у першій таблиці, бачимо загальну кількість протестованих оптичних волокон, скільки оптичних волокон працює, скільки пошкоджено та скільки не працює.

У нижній таблиці можна побачити бачити загальну інформацію по кожному оптичному волокну, ми бачимо звідки і до якого місця прокладене оптичне волокно, його фізичну довжину, його тип, довжину хвилі, пропускну здатність, максимальне згасання регенераційної ділянки та стан волокна. Відповідно до того як було у таблиці по базовим станціям реалізована таблиця по оптичним з'єднанням якщо волокно не пошкоджене задній фон комірок таблиці має зелений колір якщо воно пошкоджене то колір комірок на задньому фоні буде оранжевим а якщо волокно не працює тобто стався обрив то колір комірки на задньому фоні буде червоним.

Тепер розглянемо структуру сторінки “Статистика” на Рисунку 4.4

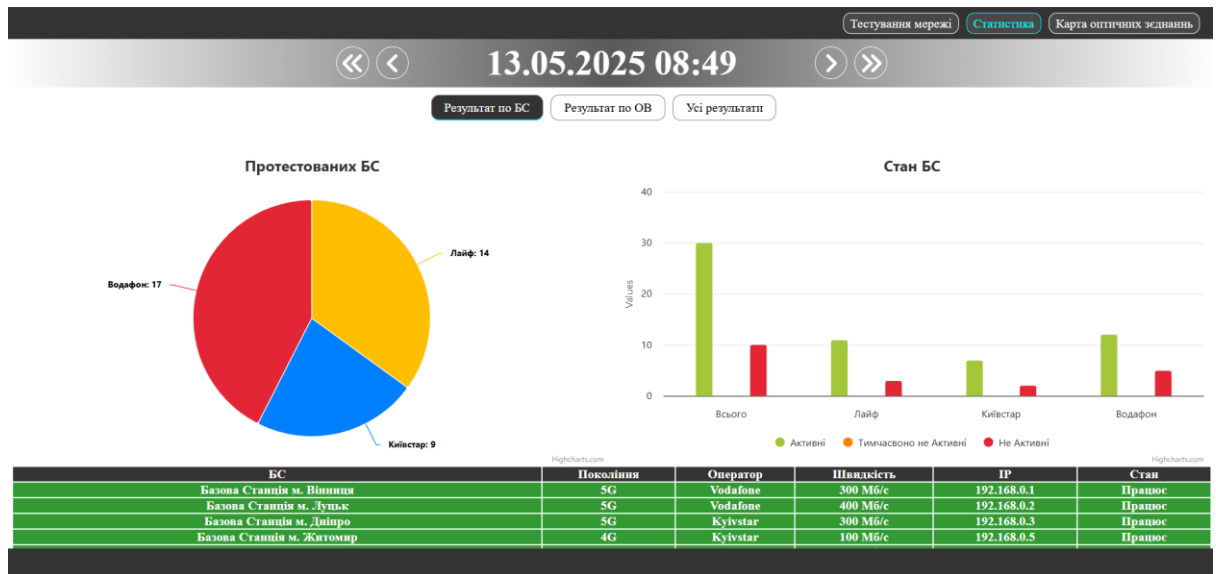


Рисунок 4.4 - Сторінка зі статистикою проведених тестувань

На рисунку 4.4 можна побачити декілька елементів а саме з самого верху знаходиться дата проведення тестування мережі та кнопки для перемикання між результатами тестувань що знаходяться збоку.

Під часом знаходиться 3 кнопки перша кнопка відображає результати тестувань базових станцій друга кнопка відображає результати тестувань оптичних волокон а третя відображає таблицю з усіма результатами тестування (Рисунок 4.5) при натисканні на яке користувача перенесе на потрібні за датою результати тестування.

Тестування мережі

Статистика

Карта оптичних зв'язків

Дата	Протестовано БС	Працює БС	Тимчасово не працює БС	Не працює БС	Протестовано ОВ	Працює ОВ	Пошкоджено ОВ	Не працює ОВ
12.05.2025 13:22	40	30	0	10	52	35	2	15
13.05.2025 08:04	40	30	0	10	52	35	2	15
13.05.2025 08:49	40	30	0	10	52	35	2	15
20.05.2025 13:17	40	30	0	10	52	36	1	15

Рисунок 3.5 - Результати усіх проведених користувачем тестувань

Під трьома кнопка знаходиться дві діаграми, на першій діаграмі відображена кількість протестованих базових станцій кодного оператора, на другій діаграмі відображений стан базових станцій тобто скільки базових станцій доступно, скільки тимчасово недоступно та скільки недоступно. Нижче під цими діаграмами можна побачити таку саму таблицю як і ту що користувач отримує при тестуванні на головній сторінці.

Схожі результати можна отримати по оптичним волокнам натиснувши на кнопку Результати ОВ, отримані результати можна побачити на рисунку 4.6

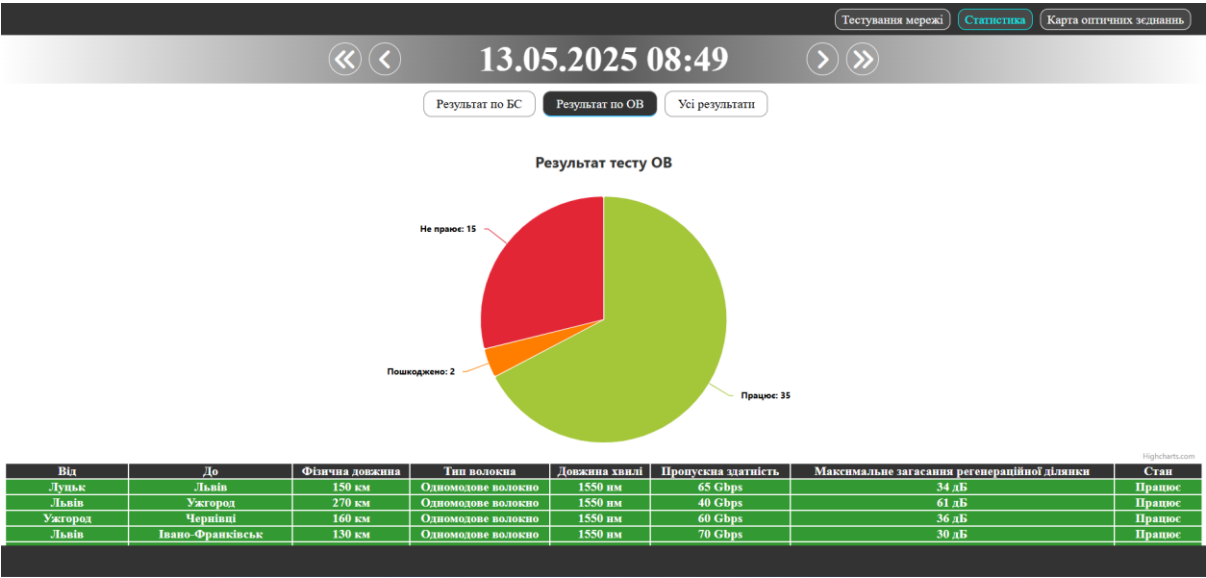


Рисунок 4.6 - Отриманні результати по оптичним волокнам на сторінці статистика

На рисунку 4.6 ми бачимо що порівняно з результатами по базовим станціям тут вже одну діаграму на якій відображено кількість оптичних з'єднань що працює, кількість пошкоджених та яка кількість не працює. Під цією діаграмаю ми бачимо таку ж саму таблицю як і на головній сторінці після проведення тестування.

Перейдемо до сторінки на якій знаходиться інтегральна карта вигляд цієї сторінки можна побачити на рисунок 4.7

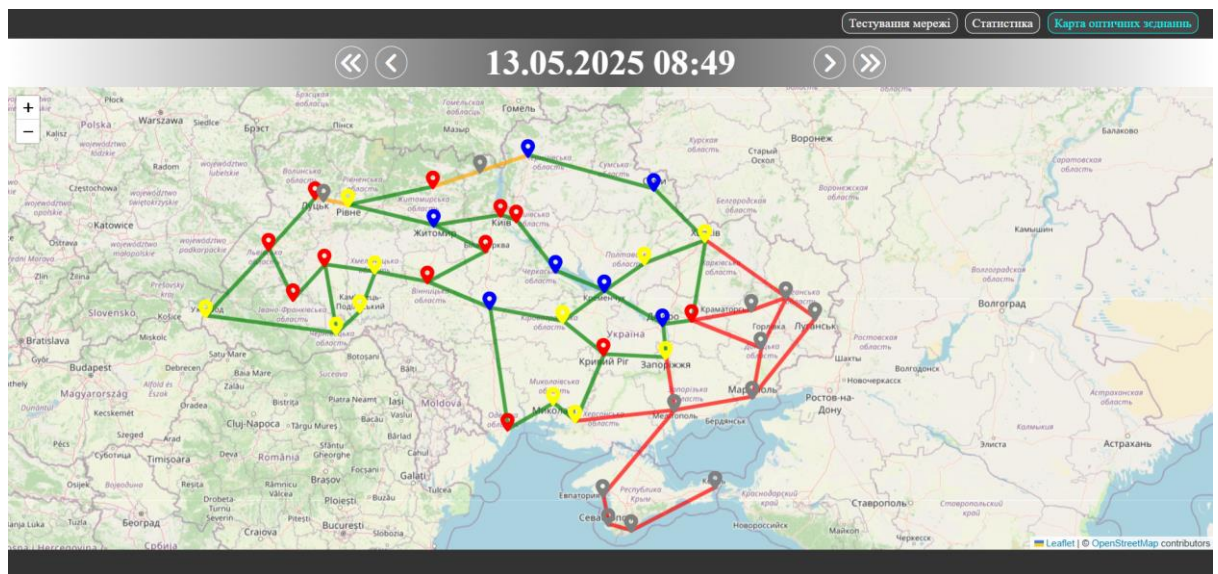


Рисунок 4.7 Вміст сторінки “Карта оптичних волокон”

На рисунку 4.7 можна побачити інтегральну карту на якій розміщено базові станції які позначені мітками та оптичні волокна.

Як ми бачимо на рисунку 4.7 мітки мають різні кольори, червоного кольору мітки це базові станції оператора vodafone, жовтого кольору це базові станції оператора lifecell а синього кольору базові станції оператора kyivstar, також можна побачити мітки сірого кольору це мітки які показують у якому місці пошкоджено оптоволоконну лінію зв'язку та відображають базові станції які не працюють або тимчасово недоступні.

При натисканні на мітку користувач бачить таке вікно як на рисунку 4.8

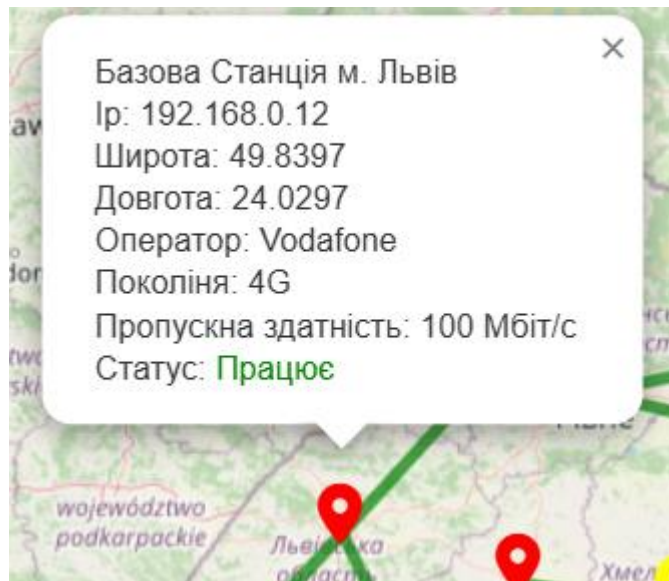


Рисунок 4.8 - Інформаційне вікно базової станції

На рисунку 4.8 можна побачити такі параметри базової станції як її назву, ір адресу, розташування по широті та довготі, оператора, покоління мережі, пропускну здатність та її стан.

Крім міток бачимо також і оптичні волокна трьох кольорів, зеленого кольору відображенні оптичні волокна що працюють, жовтого кольору оптичні волокна відображенні ті волокна що мають пошкодження а червоним кольором ті що не працюють через обрив волокна. При натисканні на оптичне з'єднання користувач може побачити інформаційне вікно як на рисунок 4.9

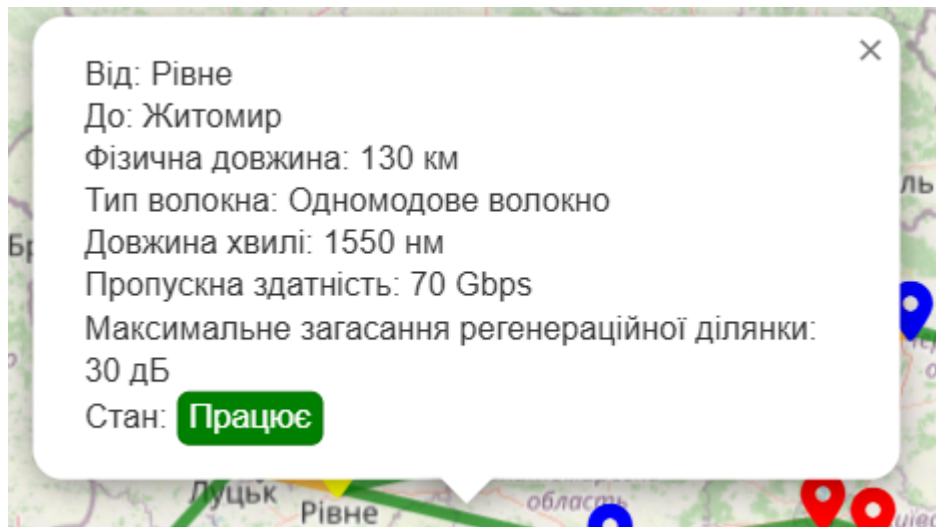


Рисунок 4.9 - Інформаційне вікно оптичного з'єднання

На рисунку 4.9 можна побачити інформаційне вікно оптичного з'єднання на якому користувач може побачити звідки та до якого пункту було прокладено оптичне волокно та такі параметри волокна як фізична довжина, тип волокна, довжина хвилі, пропускна здатність, максимальне згасання регенераційної ділянки та стан волокна.

Усі параметри оптичних волокон є максимально наближені до реалістичних а максимальне згасання регенераційної ділянки розраховано за виразом [51]

$$A = [aL + (0,1N) + 0,4 \cdot 2] \quad (4.1)$$

де a – максимальне значення коефіцієнта загасання на даному оптичному кабелі, дБ/км;

L – оптична довжина кабелю

$0,1$ – номінальне значення загасання в місці зварювання для оптичного модуля оптичного волокна, дБ;

N – кількість зрощувань оптичного волокна на регенераційній ділянці;

$0,4$ – згасання в роз'ємних з'єднувачах, дБ;

4.2 Реалізація алгоритму тестування мобільної мережі

Алгоритм тестування мобільної мережі є основною частиною системи, що забезпечує перевірку якості з'єднання, визначення параметрів зв'язку та оцінку продуктивності мережі. Реалізований алгоритм тестування мобільної мережі виконує симуляцію перевірки її параметрів, зокрема доступності базових станцій, якості оптичних з'єднань та пропускної здатності. Оскільки фактичного доступу до реального обладнання мобільної мережі та каналів зв'язку немає, тестування виконується у вигляді візуалізованого запиту до бази даних, яка містить заздалегідь підготовлені стимульовані дані, тому у межах реалізації системи було розроблено алгоритм стимульованого тестування мобільної мережі, який ґрунтується на використанні заздалегідь підготовлених даних, що зберігаються в базі даних. Розроблений алгоритм дозволяє ефективно імітувати процес опитування елементів мережі та відображення результатів у зручному для користувача форматі. Даний алгоритм представлений на рисунку 4.10

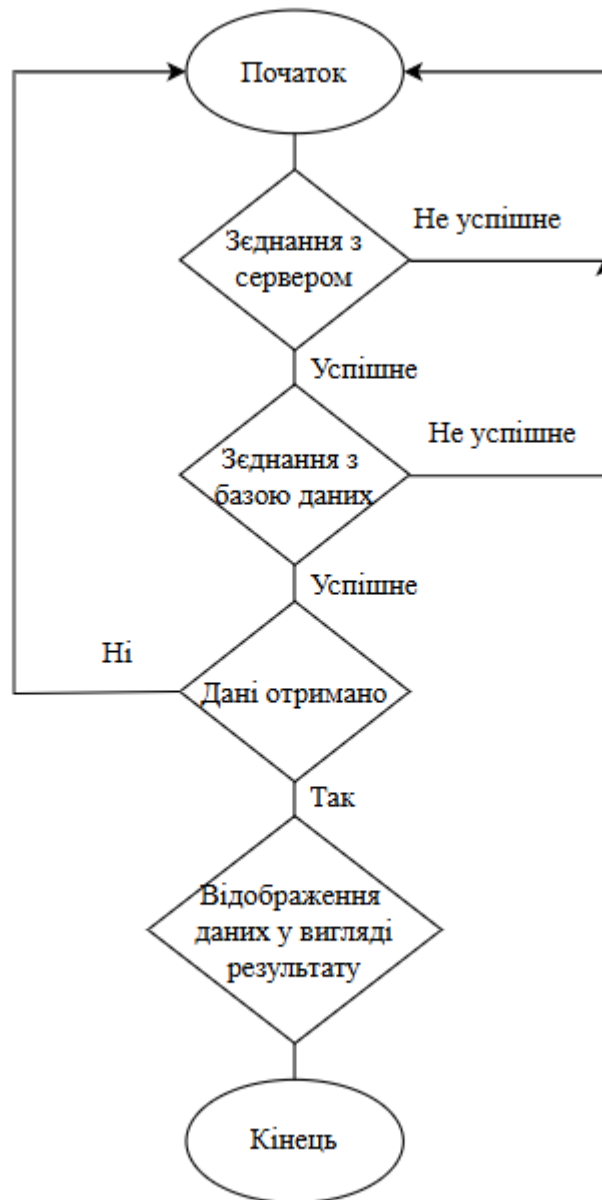


Рисунок 4.10 - Алгоритм тестування мережі

Процес тестування ініціюється через інтерфейс користувача, де передбачено відповідну кнопку, що запускає симуляцію. Після цього клієнтська частина застосунку формує запит до серверної частини. Сервер обробляє цей запит і повертає відповідні дані з бази даних.

У базі даних міститься інформація про основні елементи мобільної мережі, зокрема базові станції та оптичні з'єднання. Для кожної базової станції зберігаються координати, назва, покриття, потужність сигналу та інші технічні характеристики. Для оптичних ліній передбачено збереження даних про маршрут

(у вигляді послідовності координат), довжину з'єднання в кілометрах, тип волокна, довжину хвилі та інші технічні характеристики..

Після отримання цих даних клієнтська частина виконує їх обробку та візуалізацію. Зокрема, базові станції відображаються на інтерактивній карті, з можливістю ознайомитися з детальною інформацією при наведенні або натисканні. Оптичні з'єднання відображаються у вигляді ліній між містами, з кольоровим маркуванням та спливаючими підказками, що містять ключові параметри з'єднання.

Окрім картографічного представлення, результати симульованого тестування виводяться також у вигляді таблиць та діаграм. Це дозволяє користувачу швидко оцінити стан мережі, порівняти характеристики різних елементів мережі та швидко побачити проблему якщо вона виникла.

Таким чином, розроблений алгоритм забезпечує можливість виконання умовного тестування мобільної мережі без необхідності підключення до реального телекомунікаційного обладнання. Такий підхід є оптимальним для демонстраційних, навчальних та дослідницьких цілей, а також може слугувати основою для подальшої інтеграції з реальними джерелами даних у майбутньому.

4.3 Реалізація відображення даних на карту

Інтерактивна карта є важливим елементом web-застосунку, оскільки забезпечує наочне відображення просторової структури мережі мобільного зв'язку, дозволяючи оперативно аналізувати стан базових станцій та оптичних з'єднань між ними. Її реалізація значно підвищує інформативність інтерфейсу користувача, спрощує процес діагностики несправностей і сприяє прийняттю обґрунтованих рішень щодо оптимізації мережевої інфраструктури.

Для реалізації відображення даних на карту було використано бібліотеку Leaflet — одну з найпопулярніших JavaScript-бібліотек для створення інтерактивних карт. Завдяки її широким можливостям та гнучкості, стало можливим інтегрувати геопросторову візуалізацію в Angular.

Процес відображення карти на сторінку починається з отримання даних з бази даних, отримавши всі необхідні дані виконується перевірка на наявність існуючого екземпляра, який при необхідності видаляється. Після цього ініціалізується нова карта за допомогою бібліотеки Leaflet.

Далі відбувається побудова міток для кожної базової станції. Для кожної з них створюється власна іконка з використанням бібліотеки FontAwesome, колір якої залежить від значення властивості color. Мітки додаються на карту та супроводжуються інформаційними спливаючими вікнами (popups), які містять розширену інформацію: IP-адресу, координати, оператора, покоління зв'язку, пропускну здатність та статус станції. Колір тексту статусу динамічно змінюється залежно від його значення (працює — зелений, тимчасово не працює — помаранчевий, не працює — червоний).

Оптичні з'єднання візуалізуються за допомогою ліній (Polyline), колір яких також відображає поточний стан: зелений — справне з'єднання, помаранчевий — пошкодження, червоний — несправність. У разі наявності проблемної точки, вона позначається окремою міткою з відповідним описом причини. При відкритті спливаючого вікна над лінією з'єднання динамічно формується текстовий опис з параметрами з'єднання (довжина, тип волокна, довжина хвилі, пропускну здатність, загасання), а також кнопка зі стилем, що відповідає статусу.

Таким чином, дана реалізація забезпечує інтерактивну візуалізацію мобільної мережі з урахуванням її просторової структури, технічних параметрів і поточного стану, що значно підвищує ефективність моніторингу та аналізу мережевої інфраструктури.

4.4 Реалізація серверної частини

Серверна частина відіграє ключову роль у функціонуванні веб-системи, оскільки забезпечує логіку обробки запитів, взаємодію з базою даних, симуляцію тестування мобільної мережі та передачу результатів клієнтській частині для подальшої візуалізації.

Для реалізації серверної частини було обрано фреймворк NestJS — сучасний прогресивний фреймворк, заснований на Node.js і TypeScript. Його модульна структура, підтримка декораторів, інтеграція з сучасними ORM та підтримка REST API роблять його зручним для створення масштабованих серверних застосунків.

У якості ORM (Object-Relational Mapping) у системі використовується Prisma — сучасна, типізована та зручна у використанні бібліотека для взаємодії з базами даних. Prisma дозволяє визначати структуру таблиць через Prisma schema, автоматично генерує типи та забезпечує безпечну роботу з даними за допомогою автодоповнення і перевірки типів.

Основні функції серверної частини включають обробку HTTP-запитів від клієнтської частини, реалізованої на Angular, з'єднання з базою даних (наприклад, PostgreSQL або MySQL) за допомогою Prisma, читання з бази даних інформації про базові станції та оптичні з'єднання, які використовуються для моделювання роботи мережі, генерацію відповідей у форматі JSON для подальшого відображення на клієнтській частині, а також симуляцію тестування мобільної мережі шляхом імітації опитування базових станцій та волокон на основі наявних даних.

Оскільки у цій реалізації відсутній доступ до реального обладнання мобільних операторів, тому процес тестування представлено у вигляді симуляції. Сервер, отримуючи запит від клієнта, звертається до бази даних через Prisma Client, отримує попередньо внесені дані, обробляє їх і формує відповідь, яка виводиться у вигляді карти, таблиць і графіків.

Серверна частина веб-системи базується на чіткій структурі, що включає контролери та сервіси. Контролер відповідає за обробку HTTP-запитів, які надходять від клієнта. Він містить методи для перевірки працездатності сервера, а також для опитування та отримання інформації про базові станції, оптичні волокна та результати тестування. Запити можуть бути як GET, які приймають параметри через рядок запиту, так і POST, що отримують дані у тілі запиту. Для роботи з базою даних використовується Prisma Client, який інтегрований у сервіс AppService. Сервіс виконує бізнес-логіку, включаючи перевірку облікових даних

користувачів, отримання інформації з бази та оновлення результатів тестування. Такий поділ ролей забезпечує логічну організацію коду, де контролер відповідає за маршрутизацію і прийом запитів, а сервіс — за їх обробку і взаємодію з базою даних.

На рисунку 3.11 можна побачити приклада використання запита GET для отримання усіх результатів уже проведених тестувань та взаємодію цього метода з сервісом `app.service.ts`.

```
// Отримання усіх результатів тестування
@Get('get-Res')
async getRes(
  @Query('name') name: string,
  @Query('type') type: string,
) {
  return this.appService.getRes(name, type);
}
```

Рисунок 4.11 - GET запит для отримання усіх результатів тестування

На рисунку 4.11 представлений метод `getRes` у контролері, що обробляє HTTP GET-запит за маршрутом `/get-Res`. Він приймає два параметри через рядок запиту: `name` — назва логіну користувача який увійшов у систему, та `type` — тип інформації який буде виведений. Ці параметри передаються у відповідну функцію сервісу `getRes`, який здійснює запит до бази даних за допомогою `Prisma` для отримання результатів тестування мобільної мережі, що відповідають заданим параметрам.

Тепер розглянемо функцію `getRes` що знаходиться у сервісі та відображена на рисунку 4.12

```
// Отримую дані з усіх тестувань
async getRes(name: string, type: string, data?: any) {
  const user = await this.databaseService.post.findFirst({
    where: {
      name: name,
      type: type,
    },
  });
  return user?.data;
}
```

Рисунку 4.12 - Функція для отримання даних з усіх тестувань

На рисунку 4.12 ми бачимо асинхронну функцію `getRes`, що відповідає за отримання даних з усіх тестувань мобільної мережі за переданими параметрами `name` та `type`. За допомогою `Prisma` здійснюється пошук першого запису у таблиці `post`, де поля `name` та `type` співпадають з переданими значеннями. Якщо такий запис знаходиться, метод повертає поле `data`, яке містить результати тестування у форматі, готовому для подальшої обробки та відображення на клієнтській частині. Такий підхід дозволяє ефективно отримувати дані з бази даних для візуалізації результатів тестування.

Таким чином, серверна частина реалізує повноцінну логіку веб-застосунку для моделювання та аналізу мобільної мережі в умовах симуляції та забезпечує стабільну взаємодію з клієнтською частиною.

4.5 Висновки до розділу “Розробка та реалізація web-застосунку”

Цей розділ є практичним втіленням усіх попередніх етапів дослідження, де здійснено безпосередню реалізацію функціональних компонентів web-застосунку. Було детально описано реалізацію користувацького інтерфейсу, алгоритмів тестування, обробки даних, а також серверної логіки. Особливо важливим є те, що система була протестована на реальних або наближених до реальних умовах, що підтверджує її працездатність і придатність до впровадження.

Завдяки створеному модулю тестування мобільної мережі, користувач отримує можливість у реальному часі проводити перевірку параметрів зв'язку, а

результати виводяться у вигляді зрозумілих візуалізацій. Це підвищує ефективність аналізу та дозволяє оперативно виявляти проблемні ділянки. Система інтегрується з картою, що забезпечує просторовий контекст і дозволяє враховувати географічну специфіку мережевого покриття. Таким чином, розділ 4 підтверджує технічну зрілість розробки та її реальну практичну користь.

5 ОХОРОНА ПРАЦІ

Охорона праці є ключовим елементом забезпечення безпечних умов життєдіяльності людини в процесі її професійної діяльності. У сучасних умовах розвитку науки й техніки, коли впроваджуються новітні технології, комп'ютеризовані системи та автоматизовані платформи, значення охорони праці лише зростає. Це пояснюється як підвищеною складністю технологічних процесів, так і необхідністю забезпечення безпеки персоналу, що працює з високотехнологічним обладнанням і програмним забезпеченням.

У галузі телекомунікацій та інформаційних технологій, зокрема у сфері створення web-застосунків для моніторингу та тестування мобільних мереж, охорона праці має специфічне значення. Незважаючи на те, що така діяльність не пов'язана безпосередньо з важким фізичним виробництвом, вона включає роботу з електронним обладнанням та мережевими пристроями. Це створює низку потенційно небезпечних та шкідливих виробничих факторів, які вимагають відповідального підходу до організації безпеки праці.

Предметом проєктування є web-застосунок для тестування мобільної мережі, який розробляється як частина інформаційно-аналітичної системи для збору, обробки та візуалізації даних щодо якості мобільного покриття. Дослідний характер розробки полягає у тестуванні застосунку в реальних умовах експлуатації, з використанням мобільних пристроїв, інтернет-з'єднання та інтерфейсів доступу до баз даних.

Під час виконання дипломної роботи на тему «Web-застосунок для тестування мобільної мережі» на дослідника згідно Гігієнічної класифікацією праці за показниками шкідливості та небезпечності факторів виробничого середовища, важкості та напруженості трудового процесу, мали вплив такі небезпечні та шкідливі виробничі фактори:

1. Фізичні: підвищена запиленість та загазованість повітря робочої зони; підвищений рівень шуму на робочому місці; підвищена чи понижена вологість

повітря; підвищений рівень статичної електрики; підвищений рівень електромагнітного випромінювання; пряма і відбита блискість; підвищена яскравість; понижена контрастність; недостатня освітленість робочої зони.

2. Психофізіологічні: розумове перевантаження; перенапруга аналізаторів; статичне перевантаження.

3. Хімічні: шкідливі речовини в повітрі робочої зони.

На основі зазначених факторів розробляємо заходи з гігієни праці та виробничої санітарії, які допоможуть забезпечити безпечні та комфортні умови праці.

5.1. Технічні рішення щодо безпечного виконання роботи

5.1.1 Технічні рішення з безпечної організації робочих місць

У рамках виконання дипломної роботи, яка передбачає розробку web-застосунку для тестування мобільної мережі, всі етапи проєктування, програмування та тестування здійснюються за персональним комп'ютером в офісному середовищі. Основним чинником забезпечення безпечних умов праці є правильна організація робочого місця відповідно до чинних нормативних документів з охорони праці.

Згідно з вимогами ДСТУ 8604:2015 «Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи», комп'ютерний стіл має площину заввишки 750 мм і глибину не менше 800 мм, що забезпечує достатній простір для розміщення клавіатури, миші, документів та інших необхідних матеріалів. Стілець має регульовану висоту сидіння, спинку з підтримкою поперекового відділу хребта. Це дозволяє працівнику змінювати положення тіла без значних зусиль, що є важливим для зниження статичного навантаження на опорно-руховий апарат.

Згідно з вимогами НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», монітор розміщується на відстані 600–700 мм від очей користувача так, щоб його верхній

край знаходився на рівні горизонтальної лінії зору. Кут зору до центра екрана становить приблизно 15–20°, що відповідає фізіологічним нормам та зменшує напруження очей. Екран розташований перпендикулярно до джерела природного світла, що мінімізує відблиски та покращує якість зорового сприйняття.

Клавіатура встановлюється на висоті 700–750 мм від рівня підлоги з можливістю опори для зап'ястків, що знижує ризик розвитку тунельного синдрому. Мишка знаходиться на одному рівні з клавіатурою, що забезпечує зручність у користуванні та знижує м'язове напруження руки. Усі кабелі та дроти від техніки заховані в кабель-канали, що унеможливорює спотикання та не порушує естетику робочого місця.

Для попередження перевтоми та зниження рівня психоемоційного навантаження в роботі передбачено чергування видів діяльності: робота за ПК не повинна тривати більше 60 хвилин без перерви, після чого рекомендовано виконувати коротку гімнастику або змінювати вид діяльності на 10–15 хвилин. Такий режим організації праці базується на положеннях СанПіН 3.3.2.007-98 та спрямований на збереження працездатності користувача протягом усього робочого дня.

Також дотримано мінімально допустимі габарити вільного простору: міжрядні проходи мають ширину не менше 1 м, а відстань від тильної сторони одного монітора до наступного — не менше 2 м, що відповідає вимогам ДБН В.2.2-3:2014. Поверхня підлоги — антиковзаюча, покрита лінолеумом

5.1.2 Технічні рішення з електробезпеки

Усі етапи роботи над web-застосунком для тестування мобільної мережі виконуються з використанням персонального комп'ютера та периферійного обладнання (монітор, клавіатура, мишка, маршрутизатор, блок живлення, тощо), що потребує надійного електроживлення. У зв'язку з цим важливим аспектом безпечного виконання роботи є дотримання вимог електробезпеки. Всі технічні

рішення в цьому підрозділі спрямовані на запобігання електротравмам та забезпечення безперервної безпечної експлуатації електроустаткування.

Робоче місце розташоване в адміністративно-побутовому приміщенні, яке підключене до стандартної чотирьохпровідної трифазної електромережі з глухозаземленою нейтраллю, напругою 380/220 В. Згідно з ДСТУ EN 61140:2017, така мережа забезпечує фазну напругу 220 В, що використовується для живлення електронного та комп'ютерного обладнання. Умови експлуатації характеризуються з підвищеною небезпекою через можливість одночасного контакту з металевими корпусами пристроїв.

У дипломній роботі передбачено використання захисного занулення, яке полягає у навмисному з'єднанні металевих корпусів обладнання з заземленим нульовим захисним провідником електромережі. Це забезпечує миттєве відключення обладнання в разі пошкодження ізоляції та переходу напруги на корпус, завдяки спрацюванню автоматичних вимикачів або плавких запобіжників. Такий підхід відповідає вимогам ПУЕ та ДСТУ ІЕС 60364-4-41:2017 щодо організації системи електрозахисту при напрузі до 1000 В.

З метою підвищення рівня електробезпеки передбачено використання автоматичних вимикачів з диференційним струмом (ПЗВ), які реагують на витік струму понад 30 мА. Це дозволяє відключити живлення ще до настання небезпечного рівня впливу електричного струму на організм людини, відповідно до ДСТУ ІЕС 61008-1:2017.

Для запобігання електротравмам від контакту з відкритими струмоведучими частинами застосовано комплекс технічних рішень:

- Ізоляція провідників та використання сертифікованих подовжувачів і мережевих фільтрів з відповідним перетином кабелю (не менше 0,75 мм²), що відповідає вимогам ДСТУ EN 50525-2-31:2018;
- Закриті розетки із заземленням, оснащені шторками, що запобігають випадковому доторканню до контактів;

- Організація електропроводки у пластикових кабель-каналах, що унеможлиблює її пошкодження та захищає персонал від випадкового контакту.

Уся електротехнічна інфраструктура, яку передбачено в дипломній роботі, відповідає чинним санітарним нормам, технічним регламентам і забезпечує надійний рівень електробезпеки при виконанні завдань з розробки Web-застосунку для тестування мобільної мержі.

5.2 Технічні рішення з гігієни праці та виробничої санітарії

5.2.1 Мікроклімат

Мікроклімат виробничих приміщень нормується в залежності від теплових характеристик виробничого приміщення, категорії робіт по важкості і періоду року. Категорія виконуваних робіт під час розробки web-застосунку для тестування мобільної мережі - 1а [52] (табл.4.2.1).

Таблиця 5.2.1 – Параметри мікроклімату

Період року	Параметр мікроклімату	Величина
Холодний	Температура повітря в приміщенні	21 ... 25 ° C
	Відносна вологість	40 ... 60%
	Швидкість руху повітря	до 0,1 м / с
Теплий	Температура повітря в приміщенні	22 ... 28 ° C
	Відносна вологість	40 ... 60%
	Швидкість руху повітря	0,1 ... 0,2 м / с

Для підтримання у виробничих приміщеннях метеорологічних умов, які задовольняють нормативні вимоги використовують систему вентиляції.

Приміщення обладнано системою загальнообмінної припливно-витяжної вентиляції. На кожен вентиляційну установку складений паспорт з технічною характеристикою та схемою установки.

Крім того, для підтримання температури в холодний період року використовують загальну систему опалення.

5.2.2. Склад повітря робочої зони

Якість повітря у приміщеннях, де виконуються роботи за персональним комп'ютером, є важливим фактором, що впливає на самопочуття, працездатність і здоров'я користувача. Робота над Web-застосунок для тестування мобільної мережі виконується в умовах закритого адміністративного приміщення. В таких умовах повітря робочої зони повинно відповідати санітарно-гігієнічним нормам за складом, чистотою та рівнем кисню.

Відповідно до ДСТУ-Н Б А.3.2.1:2007, основними показниками якості повітря є вміст шкідливих речовин, зокрема вуглекислого газу (CO_2), пилу, озону, а також оксид азоту. При відсутності джерел токсичних випарів та небезпечних матеріалів основним фактором, що впливає на якість повітря, є концентрація CO_2 , який виділяється внаслідок дихання. У непровітрюваних приміщеннях вона може перевищувати гранично допустиму норму та викликати головний біль, втому, зниження уваги [53].

ГДК шкідливих речовин, які знаходяться в досліджуваному приміщенні, наведені в таблиці 5.2.2.

Таблиця 5.2.2 – ГДК шкідливих речовин у повітрі

Назва речовини	ГДК, мг/м ³		Клас небезпечності
	Максимально разова	Середньодобова	
Оксид азоту	0,085	0,085	2

Продовження таблиці 5.2.2 – ГДК шкідливих речовин у повітрі

Вуглекислий газ	3	1	4
Пил нетоксичний	0,5	0,15	4
Озон	0,16	0,03	1

Під час роботи на ПК важливо, щоб повітря мало певний іонний склад. Рівні позитивних і негативних іонів у повітрі приміщень з ПК мають відповідати санітарно-гігієнічним нормам (табл.4.2.3).

Таблиця 5.2.3 – Рівні іонізації повітря приміщень при роботі на ПК

Рівні	Кількість іонів в 1 см ³	
	n+	n-
Мінімально необхідні	400	600
Оптимальні	1500-3000	3000-5000
Максимально необхідні	5000	5000

Забезпечення складу повітря робочої зони здійснюється за допомогою системи кондиціонування та вологого прибирання.

5.2.3 Виробниче освітлення

Раціональне виробниче освітлення є важливою умовою створення безпечного та комфортного середовища праці. Зорове навантаження, пов'язане з розробкою Web-застосунку для тестування мобільної мережі, передбачає постійну роботу за комп'ютером, тому забезпечення нормативного рівня освітленості є критичним фактором профілактики зорового та психофізіологічного перевантаження працівників.

У приміщенні, де організовано робочі місця розробників, застосовується комбінована система освітлення, що включає природне бокове освітлення через

вікна у зовнішній стіні та загальне штучне освітлення світлодіодними світильниками. Такий підхід дозволяє забезпечити нормовану освітленість протягом усього робочого дня, незалежно від метеоумов чи часу доби.

Освітлення нормується відповідно до вимог ДБН В.2.5-28:2018 «Природне і штучне освітлення». Для виконання зорових робіт середньої точності (категорія IVa) необхідно забезпечити мінімальну освітленість на робочій поверхні не менше 300 лк (при комбінованій системі освітлення) та коефіцієнт природної освітленості (КПО) не нижче 1,5%, дані зазначені у таблиці 5.2.4:

Таблиця 5.2.4 - Норми освітленості в приміщенні

Характеристика зорової роботи	Найменший або еквівалентний розмір об'єкта розрізнення, мм	Розряд зорової роботи	Підряд зорової роботи	Контраст об'єкту з фоном	Характеристика фону	Штучне при системі комбінованого освітлення		Природне E_n пр	Суміщене E_n сум
						всього	у т. ч. від загального		
Середньої точності	Від 0,5 до 1	ІМ	а	Середній	Середній	750	200	4,0	2,4

Для максимального використання природного освітлення в приміщенні слід систематично очищувати вікна від пилу та встановити жалюзі. Віконні прорізи не затемнюються іншими будівлями.

Як джерела світла для штучного освітлення в приміщенні застосовуються люмінесцентні лампи типу ЛБ. Допускається застосування ламп розжарювання у світильниках місцевого освітлення.

5.2.4 Виробничий шум

Рівні шуму на робочих місцях визначаються за ДСН 3.3.6.037-99 «Санітарні норми виробничого шуму, ультразвуку та інфразвуку» (табл.5.2.5) [54].

Таблиця 5.2.5 Допустимі рівні звуку, еквівалентні рівні звуку і рівні звукового тиску в октавних смугах частот

Вид трудової діяльності, робочі місця	Рівні звукового тиску в дБ в октавних смугах із середньгеометричними частотами, Гц									Рівні звуку, еквівалентні і рівні звуку, дБА/дБАек в.
	31,5	63	125	250	500	1000	2000	4000	8000	
Творча діяльність, обробка даних,	86	71	61	54	49	45	42	40	38	60

Рівень шуму на робочих місцях не має перевищувати 60 дБА, відповідно до вимог ДСН 3.3.6.037-99 для приміщень, де виконується програмування, проектування та обробка інформації. Досягнення нормативного рівня шуму забезпечується за рахунок використання малOSHумного обладнання, зокрема: безвентиляторних систем охолодження в ноутбуках і ПК, твердотільних

накопичувачів (SSD) замість традиційних жорстких дисків (HDD), тихохідних офісних принтерів, енергоефективних блоків живлення та LED-моніторів з пасивним охолодженням. Для зниження рівня шумів в приміщенні застосовуються звукопоглинальні акустичні панелі, встановлені на стінах. Додаткове зниження шуму забезпечується за допомогою звукоізоляційних прокладок під технікою, меблевих перегородок з поглинаючим шаром та м'якого покриття підлоги (акустичний лінолеум).

5.2.5. Виробничі випромінювання

Під час розробки Web-застосунку для тестування мобільної мережі працівники значний час працюють за персональними комп'ютерами, дисплеями та іншою офісною технікою. Основними джерелами виробничих випромінювань у таких умовах є відеомонітори, системні блоки, периферійні пристрої (принтери, сканери), а також телекомунікаційне обладнання – зокрема Wi-Fi-модулі та мобільні пристрої. Виробничі випромінювання, які можуть впливати на організм людини, включають електромагнітні поля низької та високої частоти, електростатичні поля та, в деяких випадках, рентгенівське випромінювання [55].

Для запобігання негативному впливу випромінювань на здоров'я персоналу необхідно дотримуватися гігієнічних нормативів згідно з ДСанПіН 3.3.6.096-2002 та ДСанПіН 3.3.2.007-98. Відповідно до таблиці 5.2.6, напруженість електричної складової електромагнітного поля на відстані 50 см від поверхні відеомонітора не повинна перевищувати 10 В/м, а магнітної складової – 0,3 А/м. Щодо електростатичного поля, його рівень для дорослих користувачів не повинен перевищувати 20 кВ/м, а для дітей – 15 кВ/м.

Таблиця 5.2.6 – Допустимі параметри електромагнітних випромінювань

Найменування параметра	Допустимі значення
Напруженість електричної складової електромагнітного поля на відстані 50 см від поверхні відеомонітору	10 В/м
Напруженість магнітної складової електромагнітного поля на відстані 50 см від поверхні відеомонітору	0,3 А/м
Напруженість електростатичного поля не повинна перевищувати:	для дорослих користувачів 20кВ/м для дітей 15кВ/м

Для забезпечення відповідності зазначеним нормам у роботі передбачено використання сучасних рідкокристалічних (LCD/LED) моніторів, сертифікованих за стандартами безпеки TCO та CE. Вони мають низький рівень електромагнітного та електростатичного випромінювання, який у кілька разів нижчий за гранично допустимі значення. Робочі місця розміщуються на безпечній відстані від електронного обладнання: не менше 50 см від екранів моніторів і щонайменше 1 м від точок доступу Wi-Fi.

5.3 Пожежна безпека

Пожежна безпека – стан об’єкта, при якому виключається можливість пожежі, а в разі його виникнення запобігається вплив на людей небезпечних факторів пожежі і забезпечується захист матеріальних цінностей. Пожежна безпека забезпечується системою запобігання пожежі і системою пожежного захисту.

Пожежна безпека приміщення повинна відповідати вимогам Кодексу цивільного захисту України та «Правила пожежної безпеки в Україні».

За вибуховою і пожежною небезпекою приміщення належить до категорії Д, згідно з нормами технологічного проектування «Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною

небезпекою». У приміщенні знаходяться ПК та інша оргтехніка, що можуть спричинити пожежу.

Згідно даних наведених у таблиці 3.1 будівля, в якій знаходиться приміщення, має II ступінь вогнестійкості приміщення можна віднести до категорії вибухопожежонебезпеки В (табл.5.3.1).

Таблиця 5.3.1. – Визначення ступеня вогнестійкості будівельних конструкцій

Ступінь вогнестійкості будинків	Мінімальні межі вогнестійкості будівельних конструкцій (у хвилинах) та максимальні межі поширення вогню по них (см)								
	стіни				колони	сходові площадки, косоури, сходи, балки, марші сходових кліток	перекриття міжповерхові (у т.ч. горищні та надпідвалами)	елементи суміщених покриттів	
	несучі та сходових кліток	самонесучі	зовнішні несучі	внутрішні несучі (перегородки)				плити, настили, прогони	балки, ферми, арки, рами
III	REI 120 M0	REI 60 M0	E15, M0 E30, M1	EI 15 M1	R 120 M0	R 60 M0	REI 45 M1	Не нормуються	

Клас приміщення і зон з вибухо- і пожежонебезпеки II-IIa (приміщення, у якому знаходяться тверді горючі речовини та матеріали.)

5.3.1. Технічні рішення системи запобігання пожежі

Для запобігання виникнення пожежі в приміщенні застосовують такі заходи:

- щорічне проведення повторних протипожежних інструктажів та занять за програмою пожежно-технічного мінімуму з особами, що відповідальні за пожежну безпеку;
- утримання в справному стані засобів протипожежного захисту;
- своєчасне інформування про несправність пожежної техніки, систем протипожежного захисту, водопостачання тощо.

Електромережі, електроприлади і апаратура експлуатується тільки у справному стані з урахуванням вказівок та рекомендацій підприємств-виготовлювачів. У разі виявлення пошкоджень електромереж, вимикачів, розеток та інших електровиробів слід негайно вимкнути їх та вжити необхідних заходів щодо приведення в пожежобезпечний стан.

5.3.2. Технічні рішення системи протипожежного захисту

Протипожежний захист – це комплекс організаційних і технічних заходів, спрямованих на забезпечення безпеки людей, запобігання пожежі, обмеження її розповсюдження, а також на створення умов для успішного гасіння пожежі.

У всіх службових приміщеннях обов'язково повинен бути «План евакуації людей при пожежі», що регламентує дії персоналу у разі виникнення вогнища загоряння і в якому зазначено місця розташування пожежної техніки. Як відомо пожежа може виникнути при взаємодії горючих речовин, окислення і джерел запалювання. У робочому приміщенні присутні всі три основні чинники, необхідні для виникнення пожежі. Горючими компонентами є: будівельні матеріали для акустичної і естетичної обробки приміщень, перегородки, двері, підлоги, ізоляція кабелів і ін

Джерелами запалювання у ВЦ можуть бути електронні схеми від ПК, прилади, застосовувані для технічного обслуговування, пристрої електроживлення, кондиціонування повітря, де внаслідок різних порушень утворюються перегріті елементи, електричні іскри та дуги, здатні викликати загоряння горючих матеріалів. У сучасних ПК дуже висока щільність розміщення елементів електронних схем. У безпосередній близькості один від одного розташовуються сполучні дроти, кабелі. При протіканні по них електричного струму виділяється значна кількість теплоти. При цьому можливо оплавлення ізоляції. Для відведення надлишкової теплоти від ПК служать системи вентиляції та кондиціонування повітря. При постійному дії ці системи представляють собою додаткову пожежну небезпеку.

При проведенні обслуговуючих, ремонтних і профілактичних робіт використовуються різні мастильні речовини, легкозаймисті рідини, прокладаються тимчасові електропровідниками, ведуть пайку та чистку окремих вузлів. Виникає додаткова пожежна небезпека, яка потребує додаткових заходів пожежного захисту. Зокрема, при роботі з паяльником слід використовувати неспалену підставку з нескладними пристроями для зменшення споживаної потужності в неробочому стані.

Однією з найбільш важливих завдань пожежної захисту є захист будівельних приміщень від руйнувань та забезпечення їх достатньої міцності в умовах впливу високих температур при пожежі. З огляду на високу вартість електронного устаткування приміщення, а також категорію його пожежної небезпеки, будинки для ВЦ і частини будинку іншого призначення, в яких передбачено розміщення ПК повинні бути 1 і 2 ступеня вогнестійкості.

Для виготовлення будівельних конструкцій використовуються, як правило, цегла, залізобетон, скло, метал та інші негорючі матеріали.

До засобів гасіння пожежі, призначених для локалізації невеликих загорянь, відносяться внутрішні пожежні водопроводи, вогнегасники, сухий пісок, азбестові ковдри і т. д. Застосування води в машинних залах ПК, сховищах носіїв

інформації, приміщеннях контрольно-вимірювальних приладів, зважаючи на небезпеку пошкодження або повного виходу з ладу дорогого устаткування можливо у виняткових випадках, коли пожежа приймає загрозливо великі розміри. При цьому кількість води повинна бути мінімальною, а пристрої ПК необхідно захистити від попадання води, накриваючи їх брезентом або полотном [56].

Для гасіння пожеж на початкових стадіях широко застосовуються вогнегасники. У приміщенні застосовуються головним чином вуглекислотні вогнегасники, перевагами якого є висока ефективність гасіння пожежі, діелектричні властивості вуглекислого газу, що дозволяє використовувати ці вогнегасники навіть у тому випадку, коли не вдається знеструмити електроустановку відразу.

5.4 Висновки до розділу “Охорона праці”

П'ятий розділ роботи має прикладне значення з точки зору створення безпечних умов праці для розробників, що працюють із системами тестування мобільних мереж. Оцінка виробничих ризиків та розробка технічних рішень з охорони праці є критично важливими для довготривалої експлуатації ІТ-інфраструктури. Встановлено низку потенційних шкідливих факторів: електромагнітне випромінювання, мікроклімат, шумове навантаження, а також недостатня іонізація повітря в приміщеннях з ПК.

Впроваджені рекомендації, зокрема щодо забезпечення мікроклімату, пожежної безпеки, освітлення й акустики, не лише підвищують рівень безпеки, а й позитивно впливають на продуктивність працівників та зменшують ймовірність професійного вигорання. Таким чином, цей розділ має не лише інженерну, але й соціальну цінність, адже демонструє інтеграцію гуманітарних і технічних аспектів у процес розробки ПЗ.

ВИСНОВКИ

У ході виконання бакалаврської дипломної роботи було розроблено Web-застосунок для тестування мобільної мережі. Основною метою цього дослідження було розробити Web-застосунок що тестує мережі та виводить результати у вигляді графіків, діаграм та інтерактивних карт.

У процесі роботи було проаналізовано основні технічні характеристики мобільних мереж, а також інструменти, які можуть бути використані для моделювання та візуалізації параметрів таких мереж. На основі цього аналізу було сформовано вимоги до функціоналу майбутнього застосунку, а також обґрунтовано вибір інструментальних засобів та технологій для його реалізації. Було розглянуто сучасні підходи до візуалізації даних, організації обміну між клієнтською та серверною частинами застосунку, а також забезпечення інтерактивності інтерфейсу користувача.

В результаті реалізовано Web-застосунок, який дозволяє у зручній формі тестувати мобільну мережу, перегляд результатів у вигляді діаграм, інтегральних карт та таблиць. Програмний продукт створено із застосуванням сучасного технологічного стеку, зокрема Angular для фронтенду та NestJS для бекенду, що забезпечує надійність, масштабованість і хорошу продуктивність.

Окрему увагу було приділено ергономічним, безпековим та технічним вимогам щодо безпечного виконання роботи при роботі з комп'ютерною технікою. Розробка проходила з урахуванням нормативів ДСТУ, вимог електробезпеки, норм мікроклімату, освітленості та іонізації повітря, що дозволяє використовувати систему в умовах навчального закладу або підприємства без шкоди для здоров'я користувачів.

Отримані результати можуть бути використані як у практичних цілях — для первинного аналізу якості мобільного зв'язку, так і в навчальних дисциплінах, пов'язаних з телекомунікаційними системами.

Отже, розроблена програма для тестування мобільної мережі є важливим інструментом для тестування та моніторингу мобільної мережі. Вона надає можливість інженерам та адміністраторам мереж оперативно та точно аналізувати

стан мережі, виявляти проблемні місця та розробляти стратегії для їх усунення, що в кінцевому підсумку підвищує надійність і доступність телекомунікаційних систем.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Zhang J., Wang M., Wang Y. 5G Mobile Communication Technology. – Cham : Springer, 2020. – 385 p.

2. Nguyen H. C., Islam S. M. R., Tran X. N., et al. Simulation of 5G base stations with user behavior and network parameters // IEEE Access. – 2021. – Vol. 9. – P. 15876–15889.
3. Wang Z., Zhou X., Li J., et al. Optical transport network architecture for 5G // Journal of Optical Communications and Networking. – 2019. – Vol. 11, No. 4. – P. B1–B9.
4. Бондаренко І. С. Геоінформаційні системи як інструмент аналізу покриття мобільних мереж // Вісник НТУУ "КПІ". – 2020. – №2. – С. 45–52.
5. Mahmood A., Seah W. K. G., Welch I., et al. A survey on 5G network simulators: opportunities, challenges and future research trends // Simulation Modelling Practice and Theory. – 2020. – Vol. 96. – Article ID 101933.
6. Дмитрук М.Ю., Онищук О.В. Впровадження 5G мобільних телекомунікацій // Тези доповідей LIV Всеукраїнської науково-технічної конференції факультету інформаційних електронних систем. – Вінниця: ВНТУ, 2025. – с.2.
7. Дубовий С. М., Литвиненко А. Ю. Розробка веб-застосунків для візуалізації параметрів мобільних мереж // Наукові праці ОНАЗ ім. О. С. Попова. – 2022. – №3. – С. 61–66.
8. Науковий журнал "Комп'ютерно-інтегровані технології: освіта, наука, виробництво" Луцьк, 2024. Випуск № 56 11 с.
9. Dariusz Zmysłowski and Jan M. Kelner Mobile Network Operators' Assessment Based on Drive-Test Campaign in Urban Area for iPerf Scenario. Available from <https://www.mdpi.com/2076-3417/14/3/1268>
10. Таненбаум Е. С., Везеролл Д. Дж. Комп'ютерні мережі. — К.: Вільямс, 2010. — 864 с.
11. Курган М. В., Чухрай Н. І. Мережі передачі даних: навч. посібник. — Львів: Видавництво Львівської політехніки, 2015. — 312 с.
12. Раппорт Т. С. Основи бездротового зв'язку. — К.: Діалектика, 2005. — 704 с.

13. Колодяжний С. О., Петухов С. В. *Основи телекомунікацій*. — Х.: ХНУРЕ, 2016. — 278 с.
14. Cisco Systems. *Understanding Packet Loss in a Network* [Електронний ресурс].
— Режим доступу: <https://www.cisco.com/c/en/us/support/docs/quality-of-service-qos/qos-packet-loss.html>
15. ITU-T Recommendation E.800: Definitions of terms related to quality of service.
— International Telecommunication Union, 2008.
16. Горбатий І. В., Бондарєв А. П. Телекомунікаційні системи та мережі. Принципи функціонування, технології та протоколи: навч. посібник. — Львів: Видавництво Львівської політехніки, 2016. — 336 с.
17. Яковенко К. О. Переваги та обмеження середовищ емуляції та моделювання комунікаційних мереж / К. О. Яковенко // Теоретичні та практичні дослідження молодих вчених : матеріали XVIII-ої Міжнар. наук.-практ. конф. магістрантів та аспірантів, 19–22 листопада 2024 р. — Харків : НТУ «ХПІ», 2024. —С. 181-182.
18. Infovista. TEMS Investigation – Network Testing and Troubleshooting. [Електронний ресурс]. — Режим доступу: <https://www.infovista.com/products/tems-investigation/network-testing-and-troubleshooting>
19. Tele Tools. Nemo Outdoor – Real-time field measurement solution. [Електронний ресурс]. — Режим доступу: <https://tele-tools.com/product/nemo-outdoor>
20. Keysight Technologies. Nemo Analyze – Drive Test Post-Processing Solution. [Електронний ресурс]. — Режим доступу: <https://www.keysight.com/us/en/product/NTN50046C/nemo-analyze-drive-test-post-processing-solution.html>
21. Дячук О.Ю., Колощук М.С., Окунькова О.О., Воротніков В.В. Комплексний аналіз програмного забезпечення для моделювання та емуляції комп'ютерних мереж: інструменти, застосування та майбутні напрями. —

- Житомир: Державний університет «Житомирська політехніка», 2024. – [Електронний ресурс]. – Режим доступу: <http://eztuir.ztu.edu.ua/123456789/8589>
22. Open5GS. Офіційний сайт проекту Open5GS [Електронний ресурс]. – Режим доступу: <https://open5gs.org/>
 23. srsRAN Project. srsRAN gNB with srsUE — srsRAN Project documentation [Електронний ресурс]. – Режим доступу: <https://docs.srsran.com/projects/project/en/latest/tutorials/source/srsUE/source/>
 24. CellMapper. Cellular Coverage and Tower Map. [Електронний ресурс]. – Режим доступу: <https://www.cellmapper.net/>
 25. NetMonster. Офіційний сайт мобільного додатку. [Електронний ресурс]. – Режим доступу: <https://netmonster.app/>
 26. Infovista. TEMS Investigation – Network Testing and Troubleshooting. [Електронний ресурс]. – Режим доступу: <https://www.infovista.com/products/tems-investigation/network-testing-and-troubleshooting>
 27. Keysight Technologies. Nemo Outdoor 5G NR Drive Test Solution. [Електронний ресурс] – Режим доступу: <https://www.keysight.com/il/en/product/NTA50000B/nemo-outdoor-5g-nr-drive-test-solution.html>
 28. Порівняння JavaScript фреймворків: React проти Angular проти Vue. JS Communities . [Електронний ресурс] – Режим доступу: <https://javascript.org.ua/porivnyannya-javascript-frejmworkiv-react-proti-angular-proti-vue/>
 29. React — Overview, Advantages, and Disadvantages. GeeksforGeeks. [Електронний ресурс] – Режим доступу: <https://www.geeksforgeeks.org/react-js-introduction/>
 30. Vue.js — офіційна документація [Електронний ресурс]. – Режим доступу: <https://vuejs.org/guide/introduction.html>

31. Модульна архітектура в Angular: переваги та застосування / ProIT. [Електронний ресурс] – Режим доступу: <https://proit.ua/arkhitekturni-mietodologhiyi-dlia-frontiend-rozrobki-riznovidi-pierievaghi-niedoliki/>
32. Переваги серверного рендерингу у Angular для SEO / IT Expert Blog [Електронний ресурс]. – Режим доступу: <https://itexpert.com.ua/seo-angular-universal>
33. Екосистема Angular: інструменти, бібліотеки та підтримка спільноти / IT блог «DevUA». [Електронний ресурс] – Режим доступу: <https://dev.ua/blog/ekosistema-angular>
34. Poranen, J. «Comparing Web Mapping Libraries: Google Maps API, Leaflet, OpenLayers, Mapbox GL JS» / GIS Stack Exchange. [Електронний ресурс] – Режим доступу: <https://gis.stackexchange.com/questions/leaflet-vs-google-maps-api>
35. Порівняння популярних серверних фреймворків для веб-розробки / Medium, стаття «Node.js vs Django vs Spring Boot vs ASP.NET Core» [Електронний ресурс]. – Режим доступу: <https://medium.com/@author/server-frameworks-comparison>
36. Оптимізація та математичне моделювання мереж зв'язку: навч. посібник / В. М. Безрук, О. М. Буханко, Д. В. Чеботарьова. - Х. : Компанія СМІТ, 194 с.
37. Телекомунікаційні системи передачі: підручник / В. М. Кичак, ТЗІ О. М. Шинкарук, Г. Г. Бортник, І. І. Чесановський, О. В. Стальченко. - Хмельницький : Видавництво НАДПСУ, 2016. - 424 с.
38. Математичні основи теорії телекомунікаційних систем / В.В Поповський, С. О. Сабурова, В. Ф. Олійник, Ю. І. Лосєв, Д. В. Агєєв та ін.: За загальною редакцією В. В. Поповського. - Харків: ТОВ “Компанія СМІТ”, 2006. - 564 с.
39. Телекомунікаційні системи / М. Ю Ільченко, С. О. Кравчук. - Київ : Наукова думка, 2017 - 697 с.
40. Oracle Corporation. Sun GlassFish Communications Server 1.5 Deployment Planning Guide. — Oracle, 2010. — Розділ: Estimating Load on Application

- Server Instances. — [Електронний ресурс]. — Режим доступу: <https://docs.oracle.com/cd/E19355-01/820-4288/abfce/index.html>
41. Горбатий І. В., Бондарєв А. П. Телекомунікаційні системи та мережі. Принципи функціонування, технології та протоколи: навч. посібник. — Львів: Видавництво Львівської політехніки, 2016. — 336 с.
 42. Моделирование та оптимізація систем : підручник / Дубовой В. М, Кветний Р. Н., Михальов О. І., А. В. Усов - Вінниця : ПП ТД “Едельвейс”, 2017. - 804 с.
 43. Simcentric. How to Calculate Max Concurrent Users for the Web Server? — [Електронний ресурс]. — Режим доступу: <https://www.simcentric.com/america-dedicated-server/how-to-calculate-max-concurrent-users-for-the-web-server/>
 44. Горбатий І. В., Бондарєв А. П. Телекомунікаційні системи та мережі. Принципи функціонування, технології та протоколи: навч. посібник. — Львів: Видавництво Львівської політехніки, 2016. — 336 с.
 45. Angular Documentation [Електронний ресурс]. — Режим доступу: <https://angular.io/docs>
 46. K. S. K. NestJS Documentation [Електронний ресурс]. — Режим доступу: <https://docs.nestjs.com>
 47. Node.js Documentation [Електронний ресурс]. — Режим доступу: <https://nodejs.org/en/docs/>
 48. MySQL Documentation [Електронний ресурс]. — Режим доступу: <https://dev.mysql.com/doc/>
 49. Railway — Cloud Platform for Developers [Електронний ресурс]. — Режим доступу: <https://railway.app/>
 50. REST API Tutorial [Електронний ресурс]. — Режим доступу: <https://restfulapi.net/>
 51. Напрямні системи електричного та оптичного зв'язку : лабораторний практикум / Онищук О. В., Городецька О. С., Стець О. С. — Вінниця : ВНТУ, 2016. — 84 с.

52. Гігієнічна класифікація праці (за показниками шкідливості і небезпеки факторів виробничого середовища від 12.08.1986 № 4137-86. - [Електронний ресурс] - Режим доступу: <http://zakon4.rada.gov.ua/laws/show/v4137400-86>
53. ДСН 3.3.6.037-99 Санітарні норми виробничого шуму, ультразвуку та інфразвуку. - [Електронний ресурс] - Режим доступу: <http://document.ua/sanitarni-normi-virobnichogo-shumu-ultrazvuku-ta-infrazvuku-nor4878.html>
54. ДСНіПЗ.3.6.096-2002. Державні санітарні норми і правила при роботі з джерелами електромагнітних полів. URL: <http://zakon2.rada.gov.ua/laws/show/z0203-03>
55. Директива № 90/270/ЄЕС «Про мінімум вимог безпеки і гігієни праці при роботі з екранними пристроями - [Електронний ресурс] - Режим доступу: <https://osha.europa.eu/en/legislation/directives/provisions-on-workload-ergonomical-and-psychosocial-risks/osh-directives/5>
56. ДБН В.1.1-7:2016 Пожежна безпека об'єктів будівництва. Загальні вимоги. URL: http://www.poliplast.ua/doc/dbn_v.1.1-7-2

Додаток А
(обов'язковий)

ІЛЮСТРАТИВНИЙ МАТЕРІАЛ

РЕЗУЛЬТАТИ ТЕСТУВАННЯ МОБІЛЬНОЇ МЕРЕЖІ ЗА ДОПОМОГОЮ WEB-ЗАСТОСУНКА

Тестувати мережі

Статистика

Карта оптичних з'єднань

Тестувати ще раз

Результат по БС

Результат по ОВ

Результат

Результат тестування Базових станцій

Усі оператори		LifeCell		Kyivstar		Vodafone	
Категорія стану	Кількість	Категорія стану	Кількість	Категорія стану	Кількість	Категорія стану	Кількість
Протестовано БС	40	Протестовано БС	14	Протестовано Kyivstar БС	7	Протестовано Vodafone	17
Всього працює	30	Всього працює	11	Всього працює	7	Всього працює	12
Всього тимчасово не працює	0	Всього тимчасово не працює	0	Всього тимчасово не працює	0	Всього тимчасово не працює	0
Всього не працює	10	Всього не працює	3	Всього не працює	2	Всього не працює	5

БС	Покліління	Оператор	Швидкість	IP	Стан
Базова Станція м. Вінниця	5G	Vodafone	300 Mб/с	192.168.0.1	Працює
Базова Станція м. Луцьк	5G	Vodafone	400 Mб/с	192.168.0.2	Працює
Базова Станція м. Дніпро	5G	Kyivstar	300 Mб/с	192.168.0.3	Працює
Базова Станція м. Житомир	4G	Kyivstar	100 Mб/с	192.168.0.5	Працює
Базова Станція м. Ужгород	4G	Lifecell	80 Mб/с	192.168.0.6	Працює
Базова Станція м. Запоріжжя	4G	Lifecell	90 Mб/с	192.168.0.7	Працює
Базова Станція м. Івано-Франківськ	5G	Vodafone	300 Mб/с	192.168.0.8	Працює
Базова Станція м. Київ	5G	Vodafone	400 Mб/с	192.168.0.9	Працює
Базова Станція м. Кропивницький	4G	Lifecell	100 Mб/с	192.168.0.10	Працює
Базова Станція м. Львів	4G	Vodafone	100 Mб/с	192.168.0.12	Працює
Базова Станція м. Миколаїв	5G	Lifecell	300 Mб/с	192.168.0.13	Працює
Базова Станція м. Одеса	4G	Vodafone	100 Mб/с	192.168.0.14	Працює
Базова Станція м. Полтава	5G	Lifecell	350 Mб/с	192.168.0.15	Працює

Рисунок 1 - Результат тестування базових станцій

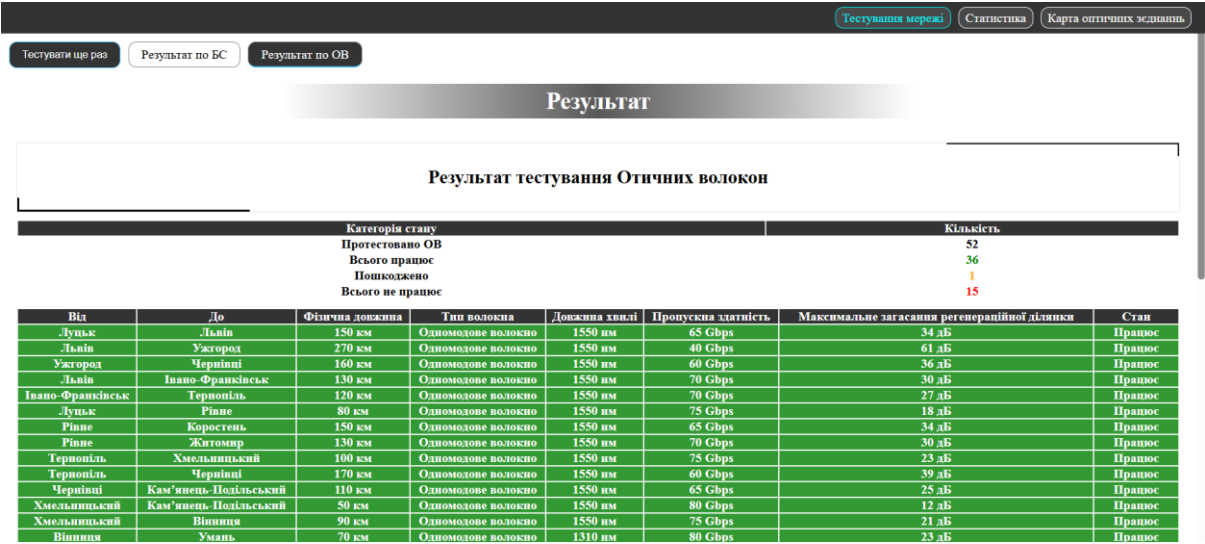


Рисунок 2 - Результат тестування оптичних волокон

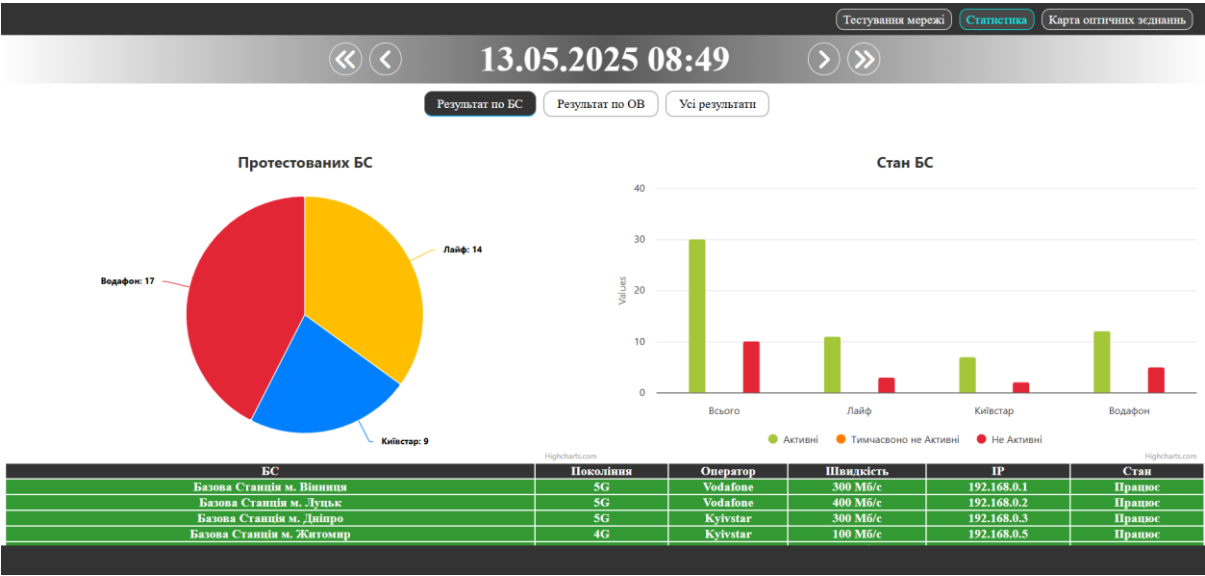


Рисунок 3 - Збережений результат тестування по базовим станціям у розділі статистика

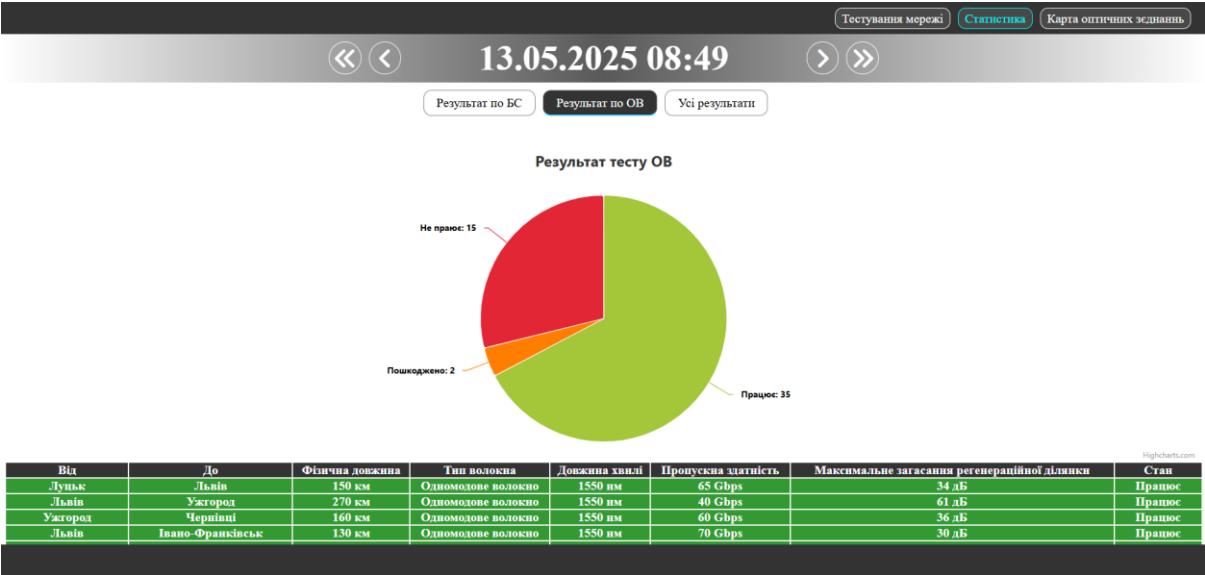


Рисунок 4 - Збережений результат тестування по оптичних волокнам у розділі статистика

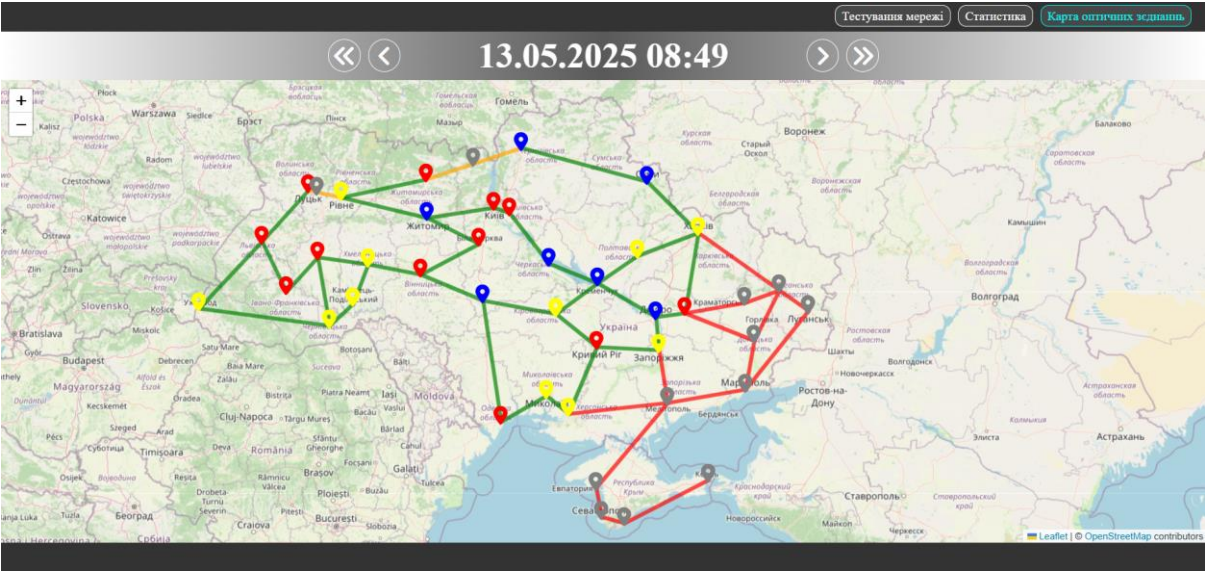


Рисунок 5 - Карта оптичних волокон за результатами тестувань

Додаток Б

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: Web-застосунок для тестування мобільної мережі

Тип роботи: бакалаврська кваліфікаційна робота

(бакалаврська кваліфікаційна робота / магістерська кваліфікаційна робота)

Підрозділ кафедра ІКСТ, факультет ФІЕС, група ПЗТ-216
(кафедра, факультет, навчальна група)

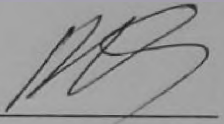
Коефіцієнт подібності текстових запозичень виявлених у роботі системою
StrikePlagiarism 2.08 %

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

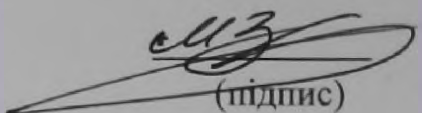
Запозичення виявлені у роботі, є законними і не містять ознак плагіату,
фабрикації, фальсифікації. Роботу прийняти до захисту

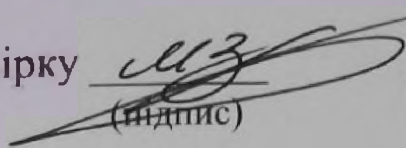
Експертна комісія:

Кичак В.М., завідувач кафедри ІКСТ
(прізвище, ініціали, посада)

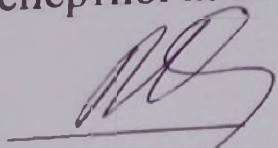

(підпис)

Васильківський М.В. гарант ОПП
(прізвище, ініціали, посада)

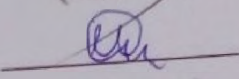

(підпис)

Особа відповідальна за перевірку  Васильківський М.В.
(підпис) (прізвище, ініціали)

З висновком експертної комісії ознайомлений(-на)

Керівник 
(підпис)

Кичак В.М., завідувач кафедри ІКСТ
(прізвище, ініціали, посада)

Здобувач 
(підпис)

Дмитрук М.Ю
(прізвище, ініціали)

ЛІСТИНГ WEB-ЗАСТОСУНКУ ДЛЯ ТЕСТУВАННЯ МОБІЛЬНОЇ МЕРЕЖІ

html файл компонента на якому відображається кнопка для тестування мережі та результати тестування.

```
<div class="box">
  <app-header></app-header>
  <div class="main">
    <h2 class="title" *ngIf="!testComplite">Тест мережі</h2>
    <div class="content">
      <div class="button-wrapper" *ngIf="!buttonHide">
        <button class="network-test-btn" (click)="runTest()" ">
          Тестувати мережу
        </button>
        <span class="spinner-ring"></span>
        <span class="spinner-ring-2"></span>
        <span class="spinner-ring-3"></span>
      </div>
    </div>
  </div>
```

```

</div>

<div class="start-box">
  <div class="res-box" *ngIf="isTestRunning">
    <div class="loader-box">
      <span class="loader">Отримуємо дані</span>
    </div>
  </div>

  <div class="res-box" *ngIf="testComplite">
    <div class="menu-box">
      <button (click)="runTest()">Тестувати ще раз</button>
      <ul>
        <li [ngClass]="{ active: getResBS == true }" (click)="getRes('BS')">
          Результат по БС
        </li>
        <li [ngClass]="{ active: getResOW == true }" (click)="getRes('OW')">
          Результат по ОВ
        </li>
      </ul>
    </div>

    <h2 class="title">Результат</h2>

    <div class="res-box-cont">
      <!-- Результат тесту по БС -->
      <div class="res-block" *ngIf="getResBS">
        <div class="res-miniBlok">
          <h2>Результат тестування Базових станцій</h2>
          <p>
            <span class="line bottom-left-horizontal"></span>
            <span class="line bottom-left-vertical"></span>
            <span class="line top-right-horizontal"></span>
            <span class="line top-right-vertical"></span>
          </p>
        </div>
        <div class="res-cont">
          <div class="res-cont-block-1">
            <div class="table-box">
              <table style="width: 25%">
                <tr class="mini-title-table">
                  <th colspan="2">Усі оператори</th>
                </tr>

```

```

<tr class="title-table">
  <th>Категорія стану</th>
  <th>Кількість</th>
</tr>
<tr>
  <th>Протестовано БС</th>
  <th>{{ markers.length }}</th>
</tr>
<tr>
  <th>Всього працює</th>
  <th style="color: green">{{ activeBS }}</th>
</tr>
<tr>
  <th>Всього темчасово не працює</th>
  <th style="color: orange">{{ timeNoActiveBS }}</th>
</tr>
<tr>
  <th>Всього не працює</th>
  <th style="color: red">{{ noActiveBS }}</th>
</tr>
</table>

<!-- Таблиця по Лайф -->
<table style="width: 25%">
  <tr class="mini-title-table">
    <th colspan="2" style="color: yellow">LifeCell</th>
  </tr>
  <tr class="title-table">
    <th>Категорія стану</th>
    <th>Кількість</th>
  </tr>
  <tr>
    <th>Протестовано БС</th>
    <th>{{ lifeTested }}</th>
  </tr>
  <tr>
    <th>Всього працює</th>
    <th style="color: green">{{ lifeActiveBS }}</th>
  </tr>
  <tr>
    <th>Всього темчасово не працює</th>
    <th style="color: orange">{{ lifeTimeNoActiveBS }}</th>
  </tr>

```

```

<tr>
  <th>Всього не працює</th>
  <th style="color: red">{{ lifeNoActiveBS }}</th>
</tr>
</table>

<!-- Таблиця по Київстар -->

<table style="width: 25%">
  <tr class="mini-title-table">
    <th colspan="2" style="color: rgb(37, 153, 255)">
      Kyivstar
    </th>
  </tr>
  <tr class="title-table">
    <th>Категорія стану</th>
    <th>Кількість</th>
  </tr>
  <tr>
    <th>Протистовано Kyivstar БС</th>
    <th>kyivstarTested</th>
  </tr>
  <tr>
    <th>Всього працює</th>
    <th style="color: green">{{ kyivstarActiveBS }}</th>
  </tr>
  <tr>
    <th>Всього темчасово не працює</th>
    <th style="color: orange">{{ kyivstarTimeNoActiveBS
}}</th>

  </tr>
  <tr>
    <th>Всього не працює</th>
    <th style="color: red">{{ kyivstarNoActiveBS }}</th>
  </tr>
</table>

<!-- Таблиця по Водафону -->
<table style="width: 25%">
  <tr class="mini-title-table">
    <th colspan="2" style="color: rgb(197, 0, 0)">
      Vodafone
    </th>

```

```

</tr>
<tr class="title-table">
  <th>Категорія стану</th>
  <th>Кількість</th>
</tr>
<tr>
  <th>Протистовано Vodafone</th>
  <th>{{ vodafoneTested }}</th>
</tr>
<tr>
  <th>Всього працює</th>
  <th style="color: green">{{ vodafoneActiveBS }}</th>
</tr>
<tr>
  <th>Всього тимчасово не працює</th>
  <th style="color: orange">{{ vodafoneTimeNoActiveBS
}}</th>

</tr>
<tr>
  <th>Всього не працює</th>
  <th style="color: red">{{ vodafoneNoActiveBS }}</th>
</tr>
</table>
</div>
</div>
<div class="res-cont-block-2">
  <!-- Таблиця з повною інформацією -->
  <table style="width: 100%">
    <tr class="title-table">
      <th>БС</th>
      <th>Покоління</th>
      <th>Оператор</th>
      <th>Швидкість</th>
      <th>IP</th>
      <th>Стан</th>
    </tr>
    <tr
      *ngFor="let marker of markers"
      [ngClass]="{
        'bg-green': marker.status === 'Працює',
        'bg-orange': marker.status === 'Тимчасово не працює',
        'bg-red': marker.status === 'Не працює'
      }"

```

```

    >
    <th>{{ marker.title }}</th>
    <th>{{ marker.generation }}</th>
    <th>{{ marker.operator }}</th>
    <th>{{ marker.speed }} Мб/с</th>
    <th>{{ marker.ip }}</th>
    <th>{{ marker.status }}</th>
  </tr>
</table>
</div>
</div>
</div>
<!-- Результат тесту ОВ -->
<div class="res-block" *ngIf="getResOW">
  <div class="res-miniBlok">
    <h2>Результат тестування Оптичних волокон</h2>
    <p>
      <span class="line bottom-left-horizontal"></span>
      <span class="line bottom-left-vertical"></span>
      <span class="line top-right-horizontal"></span>
      <span class="line top-right-vertical"></span>
    </p>
  </div>
  <div class="res-cont">
    <div class="res-cont-block-1">
      <table style="width: 100%">
        <tr class="title-table">
          <th>Категорія стану</th>
          <th>Кількість</th>
        </tr>
        <tr>
          <th>Протестовано ОВ</th>
          <th>{{ connections.length }}</th>
        </tr>
        <tr>
          <th>Всього працює</th>
          <th style="color: green">{{ activeOW }}</th>
        </tr>
        <tr>
          <th>Пошкоджено</th>
          <th style="color: orange">{{ timeNoActiveOW }}</th>
        </tr>
        <tr>

```

```

        <th>Всього не працює</th>
        <th style="color: red">{{ noActiveOW }}</th>
    </tr>
</table>
</div>

<div class="res-cont-block-2">
    <table style="width: 100%">
        <tr class="title-table">
            <th>Від</th>
            <th>До</th>
            <th>Фізична довжина</th>
            <th>Тип волокна</th>
            <th>Довжина хвилі</th>
            <th>Пропускна здатність</th>
            <th>Максимальне загасання регенераційної ділянки</th>
            <th>Стан</th>
        </tr>
        <tr>
            <th>
                *ngFor="let conection of connections"
                [ngClass]="{
                    'bg-green': conection.color === 'green',
                    'bg-orange': conection.color === 'orange',
                    'bg-red': conection.color === 'red'
                }"
            >
            <th>{{ conection.from }}</th>
            <th>{{ conection.to }}</th>
            <th>{{ conection.lengthKm }} км</th>
            <th>{{ conection.fiberType }}</th>
            <th>{{ conection.frequency }}</th>
            <th>{{ conection.bandwidth }} Gbps</th>
            <th>{{ conection.attenuation }} дБ</th>
            <th *ngIf="conection.color === 'green'">Працює</th>
            <th *ngIf="conection.color === 'orange'">Пошкоджено</th>
            <th *ngIf="conection.color === 'red'">Не працює</th>
        </tr>
    </table>
</div>
</div>
</div>
</div>
</div>

```

```

        </div>
    </div>
    <app-footer></app-footer>
</div>

```

typescript файл у якому реалізовані функції для отримання даних

```

import { CommonModule } from '@angular/common';
import { Component, OnInit } from '@angular/core';
import { HeaderComponent } from '../../component/header/header.component';
import { FooterComponent } from '../../component/footer/footer.component';
import { SpeedTestModule } from 'ng-speed-test';
import { SpeedTestService } from 'ng-speed-test';
import { HttpClient } from '@angular/common/http';
import { DataService } from '../../service/data/data.service';
import { SendInfoService } from '../../service/sendInfo/send-info.service';

@Component({
  selector: 'app-test',
  imports: [CommonModule, HeaderComponent, FooterComponent, SpeedTestModule],
  templateUrl: './test.component.html',
  styleUrls: ['./test.component.css'],
})
export class TestComponent {
  isTestRunning: boolean = false;
  buttonHide: boolean = false;
  testComplite: boolean = false;
  userLogin: any;

  getResBS: boolean = true;
  getResOW: boolean = false;

  markers: any = [];
  connections: any = [];
  allRes: any = [];

  // BC
  lifeTested: any;
  kyivstarTested: any;
  vodafoneTested: any;

  activeBS: any;

```

```

timeNoActiveBS: any;
noActiveBS: any;

lifeActiveBS: any;
lifeTimeNoActiveBS: any;
lifeNoActiveBS: any;

kyivstarActiveBS: any;
kyivstarTimeNoActiveBS: any;
kyivstarNoActiveBS: any;

vodafoneActiveBS: any;
vodafoneTimeNoActiveBS: any;
vodafoneNoActiveBS: any;

// OB

activeOW: any;
timeNoActiveOW: any;
noActiveOW: any;

constructor(
  private speedTestService: SpeedTestService,
  private http: HttpClient,
  private dataService: DataService,
  private sendInfoService: SendInfoService
) {}

runTest() {
  this.buttonHide = true;
  this.testComplite = false;
  this.isTestRunning = true;

  this.sendInfoService.userLogin$.subscribe((data) => {
    this.userLogin = data;

    this.dataService.getBSInfo(this.userLogin).subscribe((data) => {
      this.markers = data;

      // BC
      const life = this.markers.filter(
        (marker: { operator: string }) => marker.operator == 'Lifecell'
      );

```

```

const kyivstar = this.markers.filter(
  (marker: { operator: string }) => marker.operator == 'Kyivstar'
);
const vodafone = this.markers.filter(
  (marker: { operator: string }) => marker.operator == 'Vodafone'
);

this.lifeTested = life.length;
this.kyivstarTested = kyivstar.length;
this.vodafoneTested = vodafone.length;

this.activeBS = this.markers.filter(
  (marker: { status: string }) => marker.status == 'Працює'
).length;
this.noActiveBS = this.markers.filter(
  (marker: { status: string }) => marker.status == 'Не працює'
).length;
this.timeNoActiveBS = this.markers.filter(
  (marker: { status: string }) => marker.status == 'Тимчасово не працює'
).length;

this.lifeActiveBS = life.filter(
  (marker: { status: string }) => marker.status == 'Працює'
).length;
this.lifeNoActiveBS = life.filter(
  (marker: { status: string }) => marker.status == 'Не працює'
).length;
this.lifeTimeNoActiveBS = life.filter(
  (marker: { status: string }) => marker.status == 'Тимчасово не працює'
).length;

this.kyivstarActiveBS = kyivstar.filter(
  (marker: { status: string }) => marker.status == 'Працює'
).length;
this.kyivstarNoActiveBS = kyivstar.filter(
  (marker: { status: string }) => marker.status == 'Не працює'
).length;
this.kyivstarTimeNoActiveBS = kyivstar.filter(
  (marker: { status: string }) => marker.status == 'Тимчасово не працює'
).length;

this.vodafoneActiveBS = vodafone.filter(
  (marker: { status: string }) => marker.status == 'Працює'

```

```

).length;
this.vodafoneNoActiveBS = vodafone.filter(
  (marker: { status: string }) => marker.status == 'Не працює'
).length;
this.vodafoneTimeNoActiveBS = vodafone.filter(
  (marker: { status: string }) => marker.status == 'Тимчасово не працює'
).length;

this.dataService.getOWInfo(this.userLogin).subscribe((data) => {
  this.connections = data;

  // OB

  this.activeOW = this.connections.filter(
    (conection: { color: string }) => conection.color == 'green'
  ).length;
  this.timeNoActiveOW = this.connections.filter(
    (conection: { color: string }) => conection.color == 'orange'
  ).length;
  this.noActiveOW = this.connections.filter(
    (conection: { color: string }) => conection.color == 'red'
  ).length;

  this.dataService.getAllRes(this.userLogin).subscribe((data) => {
    this.allRes = data;
    this.allRes.shift();
    const now = new Date();
    const formattedDate = now
      .toLocaleString('uk-UA', {
        day: '2-digit',
        month: '2-digit',
        year: 'numeric',
        hour: '2-digit',
        minute: '2-digit',
        hour12: false,
      })
      .replace(',', ' ');

    const sendData = {
      time: formattedDate,
      allBS: this.markers,
      allOW: this.connections,
      testBSRes: {

```

```

        lifeTested: this.lifeTested,
        kyivstarTested: this.kyivstarTested,
        vodafoneTested: this.vodafoneTested,
        activeBS: this.activeBS,
        timeNoActiveBS: this.timeNoActiveBS,
        noActiveBS: this.noActiveBS,
        lifeActiveBS: this.lifeActiveBS,
        lifeTimeNoActiveBS: this.lifeTimeNoActiveBS,
        lifeNoActiveBS: this.lifeNoActiveBS,
        kyivstarActiveBS: this.kyivstarActiveBS,
        kyivstarTimeNoActiveBS: this.kyivstarTimeNoActiveBS,
        kyivstarNoActiveBS: this.kyivstarNoActiveBS,
        vodafoneActiveBS: this.vodafoneActiveBS,
        vodafoneTimeNoActiveBS: this.vodafoneTimeNoActiveBS,
        vodafoneNoActiveBS: this.vodafoneNoActiveBS,
    },
    testOWRes: {
        activeOW: this.activeOW,
        timeNoActiveOW: this.timeNoActiveOW,
        noActiveOW: this.noActiveOW,
    },
};

if (Array.isArray(this.allRes)) {
    this.allRes.push(sendData);
    this.dataService
        .sendAllRes(this.userLogin, this.allRes)
        .subscribe((data) => {});
}
});

this.isTestRunning = false;
this.testComplite = true;
});
});
}

getRes(param: string) {
    this.getResBS = false;
    this.getResOW = false;

    if (param == 'BS') {

```

```

        this.getResBS = true;
    } else if (param == 'OW') {
        this.getResOW = true;
    } else {
        this.getResBS = true;
    }
}
}
}

```

Серверна частина

Компонент у якому налаштовані get та post запити

```

import { Body, Controller, Get, Param, Post, Query } from '@nestjs/common';
import { AppService } from '../app.service';

```

```

@Controller('app')
export class AppController {
    constructor(private readonly appService: AppService) {}

    @Get('hello')
    getHello(): string {
        return this.appService.getHello();
    }

    // Перевірка на вхід
    @Get('check-credentials')
    async checkCredentials(
        @Query('name') name: string,
        @Query('password') password: string,
        @Query('type') type: string,
    ): Promise<boolean> {
        return this.appService.checkCredentials(name, password, type);
    }

    // Отримання БС
    @Get('get-BS')
    async getBS(
        @Query('name') name: string,
        @Query('type') type: string,
    ) {

```

```

    return this.appService.getBSInfo(name, type);
}

// Отримання ОВ
@Get('get-OW')
async getOW(
  @Query('name') name: string,
  @Query('type') type: string,
) {
  return this.appService.getOWInfo(name, type);
}

// Отримання усіх результатів тестування
@Get('get-Res')
async getRes(
  @Query('name') name: string,
  @Query('type') type: string,
) {
  return this.appService.getRes(name, type);
}

// Оновлення інформації по всіх результатах тестування
@Post('send-Res')
async sendRes(
  @Body() params: { name: string, type: string, data: object[] } // Отримуємо
тіло запиту
) {
  return this.appService.sendRes(params.name, params.type, params.data);
}
}

```

Сервіс у якому знаходяться функції для отримання даних з бази даних та передачу даних на базу даних

```

import { Injectable } from '@nestjs/common';
import { DatabaseService } from '../database/database.service';

@Injectable()
export class AppService {
  constructor(private readonly databaseService: DatabaseService) {}

  getHello(): string {

```

```

    return 'Hello World!';
}

// Перевірки чи є користувач в БД
async checkCredentials(
  name: string,
  password: string,
  type: string,
): Promise<boolean> {
  if (!name || !password) {
    return false;
  }
  const user = await this.databaseService.post.findFirst({
    where: {
      name: name,
      password: password,
      type: type,
    },
  });

  if (user == null) {
    return false;
  } else {
    return !!user;
  }
}

// Отримую дані з БД по ВС
async getBSInfo(name: string, type: string) {
  const user = await this.databaseService.post.findFirst({
    where: {
      name: name,
      type: type,
    },
  });
  return user?.data;
}

// Отримую дані з БД по ОВ
async getOWInfo(name: string, type: string) {
  const user = await this.databaseService.post.findFirst({
    where: {
      name: name,

```

```

        type: type,
      },
    ));
    return user?.data;
  }

  // Отримую дані з усіх тестувань
  async getRes(name: string, type: string, data?: any) {
    const user = await this.databaseService.post.findFirst({
      where: {
        name: name,
        type: type,
      },
    });
    return user?.data;
  }

  // Оновлення даних з усіх тестувань
  async sendRes(name: string, type: string, data: any) {
    console.log(data)
    const user = await this.databaseService.post.findFirst({
      where: {
        name: name,
        type: type,
      },
    });
    if (user) {
      return this.databaseService.post.update({
        where: {
          id: user.id,
        },
        data: {
          data: data
        }
      });
    }
  }
}

```