Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Кафедра ПІ

СПЕЦИФІКАЦІЯ

ДО КОМПЛЕКСНОЇ КУРСОВОЇ РОБОТИ

Тема роботи: Веб-система контролю зберігання вакцин

Виконав:

ст. гр. ПЗПІ-22-1  
  
  
Керівник  
доцент каф. ПІ

Головін С.Ю.

Мельнікова Р.В.

Комісія:

доц. каф. ПІ  
ст. викл. каф. ПІ  
ст. викл. каф. ПІ

Каук В.І.  
Олійник О.В.

Онищенко К.Г.

Харків, 2025 р.

**ЗМІСТ**

1 Вступ…………….…………………………………….………………………………4

1.1 Огляд продукту…………..………...……..……………………………..……….4

1.2 Мета……………………...………...……………………………………………..4

1.3 Межі………………………………………………..……………...……………...5

1.4 Посилання..………………………………………..……………...……………...6

1.5 Означення та абревіатури.………………………..……………...……………...6

2 Загальний опис….…………………………………….……………………………..8

2.1 Перспективи продукту…..………………………..……………...……………...8

2.2 Функції продукту………..…….…………………..…………….……………….8

2.3 Характеристики користувачів...…………………..……………...……………...9

2.4 Загальні обмеження…………...…………………..……………...……………...9

2.5 Припущення та залежності…...…………………..……………...…………….10

3 Конкретні вимоги……….……………………………………………….....…...…...11

3.1 Вимоги до зовнішніх інтерфейсів.……………………………...……………11

3.1.1 Інтерфейс користувача……..……………………………...……………11

3.1.2 Апаратний інтерфейс……..………………….…………...……………..11

3.1.3 Програмний інтерфейс……..………..…………………...……………..12

3.1.4 Комунікаційний протокол.....………..…………………...……………..12

3.1.5 Обмеження памʼяті……........………..…………………...……………..12

3.1.6 Операції…………..……........………..…………………...……………..13

3.1.7 Функції продукту...……........………..…………………...……………..13

3.1.8 Припущення й залежності....………..…………………...……………..14

3.2 Властивості програмного продукту………………………….…………...…..14

3.3 Атрибути програмного продукту………………………………………....…..15

3.3.1 Надійність……………….......………..…………………...……………..15

3.3.2 Доступність.…………….......………..…………………...……………..15

3.3.3 Безпека…….…………….......………..…………………...……………..16

3.3.4 Супроводжуваність.…….......………..…………………...……………..16

3.3.5 Переносимість…....…….......………..…………………...……………..16

3.3.6 Продуктивність.......…….......………..…………………...……………..17

3.4 Вимоги бази даних……………...………………………………………....…..17

3.5 Інші вимоги……………………...………………………………………....…..18

4 Додаткові матеріали….……………………………………………………………...19

**1 ВСТУП**

1.1 Огляд продукту

Програмний продукт “Веб-система контролю зберігання вакцин” розроблено з метою забезпечення високої якості та безпеки зберігання вакцин. Сучасні вимоги до зберігання вакцин передбачають підтримання оптимальних умов температури та вологості, що є критично важливими для збереження їх ефективності. Недостатній контроль цих параметрів може призвести до псування вакцин та значних фінансових втрат, а також створити ризик для здоров’я населення.

Запропонована система інтегрує інноваційні IoT-сенсори, що в режимі реального часу здійснюють моніторинг умов зберігання, автоматично збираючи дані про температуру, вологість та інші ключові параметри. Отримані дані передаються на серверну частину системи, де відбувається їх аналіз, зберігання та візуалізація через зручний веб-інтерфейс. Крім того, система автоматизує процес сповіщення відповідального персоналу у випадку виявлення поточних чи потенційних відхилень від встановлених норм, що дозволяє оперативно реагувати на потенційні загрози та запобігати псуванню вакцин.

Цей підхід забезпечує комплексне управління запасами вакцин, підвищує оперативність реагування на критичні ситуації та сприяє оптимізації логістичних процесів у медичних закладах. Розроблена система відповідає сучасним вимогам ринку, поєднуючи високий рівень технологічності з простотою використання, що робить її корисною для транспортних компаній будь-якого розміру.

1.2 Мета

Основною метою системи є забезпечення ефективного контролю умов зберігання вакцин за допомогою сучасних технологій IoT, що дозволить оперативно виявляти поточні чи потенційні відхилення від встановлених норм. Це сприятиме збереженню якості вакцин, зниженню фінансових втрат та мінімізації ризиків для здоров’я населення.

Крім того, система орієнтована на автоматизацію процесів моніторингу, аналізу даних та сповіщення відповідного персоналу, що дозволить забезпечити надійність зберігання запасів. Завдяки зручному веб-інтерфейсу, розроблена система стає корисною для транспортних компаній будь-якого розміру, забезпечуючи швидкий доступ до інформації та прийняття обґрунтованих рішень у сфері управління запасами вакцин.

1.3 Межі

Межі системи вартує розглянути з погляду на внутрішній функціонал, такий що лежить у відповідальності системи, та зовнішній функціонал, такий що є дотичним, але не належить до системи напряму.

До внутрішнього функціоналу відноситься:

* реальний моніторинг умов зберігання вакцин за допомогою IoT-сенсорів;
* автоматизація аналізу зібраної інформації та визначення поточних чи потенційних відхилень від встановлених норм;
* генерація сповіщень відповідальному персоналу у випадку виявлення критичних змін умов зберігання;
* графічна візуалізація даних та звітність, з можливістю експорту, через веб-інтерфейс, для оперативного прийняття рішень;
* зручний веб-інтерфейс, для проведення облікових дій з користувачами, вакцинами та іншими складниками системи, що буде мати системи валідації вводу та інший фокус на якість досвіду користувача.

До зовнішнього функціоналу відноситься:

* фізичне управління процесами зберігання вакцин, наприклад контроль за охолоджувальними системами, не є частиною системи;
* інтеграція з існуючими апаратними засобами або системами логістики здійснюється через стандартизовані API, але їх внутрішня логіка не контролюється системою;
* система може бути розширена мобільним додатком або іншим каналом доступу до інформації, як окремими модулями, проте основний акцент робиться на веб-інтерфейсі;
* система сфокусована на автоматизації збору та аналізу даних про умови зберігання вакцин і не включає фізичну організацію зберігання або ремонт обладнання.

1.4 Посилання

У цьому розділі наведені посилання на нормативні документи, стандарти, технологічні ресурси та інші джерела, які стали основою для розробки системи.

Серед них:

1. React.js. URL: https://react.dev/ (дата звернення 01.03.2025).
2. JavaScript. URL: https://developer.mozilla.org/uk/docs/Web/JavaScript (дата звернення 01.03.2025).
3. Express. URL: https://expressjs.com/ (дата звернення 01.03.2025).
4. Arduino. URL: https://expressjs.com/ (дата звернення 01.03.2025).
5. ВООЗ. URL: https://www.who.int/ (дата звернення 01.03.2025).

Ці ресурси становлять основу для розробки системи та забезпечують відповідність сучасним вимогам у сфері веб-розробки, IoT-технологій та стандартів зберігання вакцин.

1.5 Означення та абревіатури

Нижче наведено основні терміни та абревіатури, що використовуються у даній специфікації:

* IoT (Internet of Things) - Інтернет речей, технологія, що дозволяє пристроям взаємодіяти та обмінюватися даними через інтернет;
* React.js - бібліотека JavaScript для створення інтерфейсів користувача, що забезпечує швидку та ефективну розробку веб-застосунків;
* JS - JavaScript, мова програмування, що використовується для розробки веб-додатків та інтерактивних елементів;
* Express - фреймворк для JS, який дозволяє створювати масштабовані серверні застосунки;
* Arduino - апаратна платформа для створення IoT-пристроїв, яка використовується для збору даних з сенсорів;
* SQL - Structured Query Language, мова програмування для управління та маніпуляції даними в реляційних базах даних;
* Реляційна база даних - система керування базами даних, що організовує дані у вигляді таблиць з чітко визначеними зв’язками між ними;
* CRUD - абревіатура для операцій Create, Read, Update, Delete (створення, читання, оновлення, видалення даних);
* API - Application Programming Interface, набір протоколів та інструментів для розробки програмного забезпечення та інтеграції систем;
* Backend - серверна частина системи, яка відповідає за обробку даних, реалізацію бізнес-логіки та взаємодію з базою даних;
* Frontend - клієнтська частина системи, яка відповідає за інтерфейс користувача та взаємодію через веб-додаток.

Ці терміни та абревіатури допомагають забезпечити чіткість формулювань у документації та сприяють кращому розумінню функціоналу системи.

**2 ЗАГАЛЬНИЙ ОПИС**

2.1 Перспективи продукту

Програмна система для контролю зберігання вакцин з використанням технології IoT має значний потенціал для подальшого розвитку та розширення. Завдяки використанню стандартизованих API система може інтегруватися з іншими інформаційними платформами, що забезпечує єдиний інформаційний простір для оперативного аналізу даних. Також передбачено розширення функціоналу шляхом додаткових модулів для розширеної аналітики, прогнозування потреб у вакцинації та створення мобільних додатків для доступу до даних у режимі реального часу.

Постійне вдосконалення IoT-сенсорів та алгоритмів аналізу сприятиме підвищенню точності моніторингу умов зберігання вакцин, а розширені можливості валідації та захисту даних забезпечать високий рівень безпеки відповідно до сучасних стандартів. Гнучкість та адаптивність системи дозволяють швидко реагувати на ринкові зміни та вимоги регуляторних органів, що відкриває можливості її використання не лише у медичних установах, але й у суміжних галузях, де потрібен контроль за умовами зберігання чутливих матеріалів.

2.2 Функції продукту

Програмна система забезпечує повний спектр функцій, спрямованих на моніторинг умов зберігання вакцин та управління запасами. Система здійснює реальний моніторинг за допомогою IoT-сенсорів, що автоматично збирають дані про температуру та вологість. Отримані дані аналізуються для визначення поточних чи потенційних відхилень від встановлених норм, що дозволяє системі генерувати сповіщення відповідальному персоналу у випадку виявлення критичних змін.

Також, система надає можливості візуалізації отриманих даних для обраної локації чи вакцини, у формі графіків та формування звітів з наявною інформацією із подальшим експортом у форматів PDF, що сприяє оперативному прийняттю рішень. Завдяки зручному вебінтерфейсу користувачі, адміністратори та оператори, можуть проводити облікові операції, а саме управляти інформацією про локації, керувати користувачами, замінювати дані про вакцини та їх запаси та переглядати історичні сповіщення персоналу, з підтримкою коректності даних, завдяки механізмам валідації вводу користувача. Ці функції забезпечують комплексний підхід до контролю умов зберігання вакцин, сприяючи збереженню їх якості та мінімізації фінансових втрат.

2.3 Характеристики користувачів

Основними ролями користувачів є адміністратори, які відповідають за управління системою та керування користувачами, та відповідальний персонал, що здійснює безпосереднє управління записами, моніторинг умов зберігання вакцини за допомогою даних, отриманих від IoT-сенсорів. Користувачі повинні мати базові навички роботи з комп'ютером та розуміння принципів роботи веб-додатків. Завдяки інтуїтивно зрозумілому веб-інтерфейсу та вбудованим механізмам валідації даних, система забезпечує простоту використання, що сприяє ефективному контролю та управлінню запасами вакцин незалежно від місця їх зберігання. Користувачі повинні мати смартфон та вміти їм користуватися для приєднання до системи автоматичних сповіщень.

2.4 Загальні обмеження

Система має декілька загальних обмежень, які варто врахувати під час її впровадження та експлуатації. По-перше, система залежить від надійного інтернет-з'єднання для передачі даних між IoT-сенсорами, серверною частиною та веб-інтерфейсом. Будь-які перебої у з'єднанні можуть впливати на точність та оперативність отримання інформації. По-друге, для користування системою необхідно, щоб користувачі володіли базовими навичками роботи з комп'ютерами та смартфонами, оскільки автоматичні сповіщення надходять на мобільні пристрої. Також система орієнтована на веб-інтерфейс, тому сумісність з різними браузерами та мобільними додатками повинна забезпечуватися через регулярні оновлення та підтримку стандартів веб-розробки.

Окремо варто зазначити, що система не включає фізичне управління умовами зберігання вакцин. Вона також залежить від правильності налаштувань IoT-сенсорів, а точність їх роботи може впливати на результати моніторингу. Ці обмеження потрібно враховувати при розгортанні системи, щоб забезпечити максимальну ефективність контролю та управління запасами вакцин.

2.5 Припущення та залежності

Розробка та впровадження системи базуються на низці припущень і залежностей, які необхідно врахувати для забезпечення її коректної роботи. Серед основних припущень можна виділити те, що у користувачів є базові навички роботи з комп'ютерами та смартфонами, а також забезпечено надійне інтернет-з'єднання для стабільної передачі даних між IoT-сенсорами, серверною частиною та веб-інтерфейсом. Передбачається, що IoT-сенсори будуть встановлені та правильно налаштовані, що дозволить отримувати точні дані про температуру та вологість.

Щодо залежностей, система спирається на сучасний технологічний стек, який включає React для створення інтерфейсу користувача, Express для розробки серверної частини та PostgreSQL для роботи з реляційною базою даних. Інтеграція з IoT-пристроями здійснюється через апаратну платформу Arduino, а також стандартні API для взаємодії з іншими інформаційними системами. Ці припущення та залежності є критично важливими для забезпечення максимальної ефективності контролю умов зберігання вакцин та надійного управління запасами.

**3 КОНКРЕТНІ ВИМОГИ**

3.1 Вимоги до зовнішніх інтерфейсів

3.1.1 Інтерфейс користувача

Інтерфейс користувача є основним каналом взаємодії з системою, тому повинен бути інтуїтивно зрозумілим та зручним. Він забезпечує швидкий доступ до основних функцій: обліку запасів вакцин, моніторингу даних з IoT-сенсорів, перегляду звітів та налаштування сповіщень. Інтерфейс підтримує адаптивну верстку, що гарантує комфортну роботу як на стаціонарних комп’ютерах, так і на мобільних пристроях. Форми вводу даних містять вбудовану валідацію для запобігання основних помилок та забезпечення високої якості інформації. Крім того, система дозволяє користувачам переглядати історичні дані у зручному форматі графіків та генерувати звіти й проводити їх експорт, що сприяє оперативному прийняттю рішень. Розробка інтерфейсу базується на сучасних принципах UI/UX, щоб забезпечити комфортне використання навіть для користувачів з мінімальним досвідом роботи з подібними системами.

3.1.2 Апаратний інтерфейс

Система взаємодіє з апаратними компонентами через IoT-сенсори, що використовують платформу Arduino. Сенсори встановлюються в місцях зберігання вакцин для безперебійного моніторингу температури та вологості. Налаштування є швидким, легким та зрозумілим і відбувається через консольний інтерфейс. Зібрані дані передаються через стабільне інтернет-з'єднання до серверної частини, де вони аналізуються та обробляються. Апаратний інтерфейс забезпечує сумісність із стандартними протоколами зв’язку, що дозволяє легко інтегрувати систему з існуючими апаратними засобами. Крім того, пристрої підтримують функції самодіагностики, що сприяють своєчасному виявленню проблем з зарядом і попередженню про необхідність обслуговування.

3.1.3 Програмний інтерфейс

Програмний інтерфейс забезпечує обмін даними між серверною частиною системи та іншими компонентами через реалізацію REST API. Він дозволяє здійснювати операції створення, читання, оновлення та видалення (CRUD) даних, пов’язаних з вакцинами, користувачами, локаціями та характеристиками цих сутностей системи. Інтерфейс використовує стандартні HTTP-протоколи та формати передачі даних JSON, що спрощує інтеграцію з іншими інформаційними системами. Розробка API відбувається з урахуванням високої продуктивності та безпеки, що гарантує оперативну обробку запитів і захист даних. Механізми аутентифікації та авторизації забезпечують обмежений доступ кожного типу користувачів.

3.1.4 Комунікаційний протокол

Система забезпечує надійну передачу даних між IoT-сенсорами, серверною частиною та веб-інтерфейсом за допомогою протоколів TCP/IP. Для обміну даними IoT-сенсори використовують HTTP протокол, що дозволяє передавати інформацію у форматі JSON.

3.1.5 Обмеження памʼяті

Система має незначні обмеження щодо пам’яті, які залежать від апаратних характеристик серверної частини та IoT-сенсорів. Обсяг пам’яті, доступний для зберігання даних, визначає кількість та обсяг історичних записів, які можуть зберігатися до архівації чи обробки. Також важливо враховувати обмеження пам’яті на пристроях IoT, де ресурси можуть бути обмеженими. Розробка системи передбачає оптимізацію використання пам’яті шляхом ефективної обробки та зберігання даних, що дозволяє підтримувати високу продуктивність при зростанні кількості записів.

3.1.6 Операції

Система підтримує низку операцій, що забезпечують ефективне управління даними та взаємодію з користувачами. До основних операцій належать операції створення, читання, оновлення та видалення (CRUD) записів, що стосуються вакцин, користувачів, записів від IoT сенсорів, повідомлень, локацій та їх характеристик.

Крім того, система автоматично здійснює збирання та обробку даних від IoT-сенсорів у режимі реального часу для моніторингу параметрів зберігання. Етапом обробки даних сенсорів є перевірка на відповідність нормам зберігання вакцини, з можливістю сповіщення відповідального персоналу, у випадку, коли параметри перевищують норму або є негативна тенденція в системі.

Виконання запитів здійснюється через REST API, що дозволяє інтегрувати систему з іншими інформаційними платформами. Також реалізовано операції генерації та експорту звітів і графічної візуалізації даних, що дозволяє користувачу легко отримувати потрібну йому інформацію. Усі операції виконуються з урахуванням високої продуктивності та безпеки, завдяки використанню механізмів аутентифікації та авторизації для обмеження доступу відповідно до ролей користувачів.

3.1.7 Функції продукту

Продукт забезпечує комплексний набір функцій, спрямованих на контроль умов зберігання вакцин та управління даними. Серед основних функцій - моніторинг параметрів зберігання за допомогою IoT-сенсорів, що автоматично збирають інформацію про температуру та вологість, а також аналіз цих даних з перевіркою на відповідність нормам зберігання вакцини. У випадку виявлення відхилень або негативної тенденції система генерує сповіщення для відповідального персоналу.

До інших функцій відноситься можливість управління записами про вакцини, користувачів, даними від IoT-сенсорів, повідомленнями, локаціями та їх характеристиками за допомогою операцій CRUD, що забезпечуються через REST API. Інтерфейс користувача дозволяє здійснювати облік, перегляд історичних даних, генерувати звіти та здійснювати їх експорт, а також використовувати графічну візуалізацію даних для оперативного прийняття рішень. Всі функції інтегровані в систему з урахуванням високої продуктивності, безпеки та відповідності ролям користувачів через механізми аутентифікації та авторизації.

3.1.8 Припущення й залежності

Розробка зовнішніх інтерфейсів системи базується на ряді припущень і залежностей, які забезпечують їх коректну роботу та інтеграцію з іншими компонентами. Передбачається, що всі апаратні та програмні засоби, зокрема IoT-сенсори на базі Arduino, налаштовані коректно і забезпечують стабільну передачу даних через TCP/IP та HTTP протокол, а також що інтернет-з'єднання є надійним для безперебійного обміну інформацією. REST API, який використовується для взаємодії між серверною частиною системи та іншими інформаційними платформами, покладається на відповідність стандартам безпеки та сумісності, а також на вбудовані механізми аутентифікації та авторизації. Крім того, інтерфейси орієнтовані на підтримку різних пристроїв користувачів, таких як комп'ютери та смартфони, що вимагає регулярного оновлення та дотримання сучасних стандартів веб-розробки. Ці припущення та залежності є критично важливими для забезпечення ефективного функціонування зовнішніх інтерфейсів системи.

3.2 Властивості програмного продукту

Програмний продукт характеризується рядом властивостей, що забезпечують його ефективне функціонування та інтеграцію в існуючі інформаційні середовища. Основна властивість є висока продуктивність та надійність, що дозволяють системі обробляти великий обсяг даних в режимі реального часу без збоїв. Продукт розроблено з урахуванням масштабованості, що забезпечує можливість розширення функціоналу та інтеграції з іншими системами через стандартизовані API.

Безпека є одним із ключових аспектів, тому реалізовані механізми аутентифікації та авторизації гарантують обмежений доступ відповідно до ролей користувачів. Інтерфейс користувача побудовано за сучасними принципами UI/UX, що забезпечує інтуїтивно зрозуміле управління даними.

3.3 Атрибути програмного продукту

3.3.1 Надійність

Надійність програмного продукту є ключовим атрибутом, що гарантує його стабільну роботу навіть при високих навантаженнях. Система спроєктована так, щоб забезпечувати безперебійну обробку великого обсягу даних, завдяки використанню високопродуктивних серверів та ефективного алгоритму обробки інформації. Автоматичне повторне встановлення з’єднання вебклієнта та нотифікація про некоректний стан батареї сенсора сприяють мінімізації можливих збоїв. Також реалізовано контроль цілісності даних, що забезпечує правдивість інформації та стабільність роботи всіх компонентів системи.

3.3.2 Доступність

Програмний продукт розроблено таким чином, щоб забезпечити високу доступність для користувачів у будь-який час. Система гарантує безперервний доступ завдяки використанню надійних серверних рішень. Вебінтерфейс адаптивний і сумісний з різними пристроями, що сприяє зручному доступу в не залежності від браузера користувача. Крім того, система використовує оптимізоване управління ресурсами для мінімізації часу відповіді та забезпечення швидкого завантаження даних, що є критично важливим для оперативного прийняття рішень.

3.3.3 Безпека

Безпека є одним із ключових атрибутів системи. Програмний продукт використовує передові механізми аутентифікації та авторизації для забезпечення обмеженого доступу до даних, гарантуючи, що лише уповноважені користувачі можуть отримувати доступ до конфіденційної інформації.. Крім того, система регулярно перевіряє цілісність даних. Ці заходи сприяють високому рівню захисту як інформації користувачів, так і самої інфраструктури системи.

3.3.4 Супроводжуваність

Програмний продукт розроблено з урахуванням принципів супроводжуваності, що забезпечує легке оновлення, розширення та модифікацію системи у майбутньому. Архітектура продукту побудована за модульним принципом, що дозволяє ізольовано вносити зміни без порушення роботи інших компонентів. Крім того, використання стандартизованих технологій і чистому коду сприяє швидкому виявленню та виправленню помилок, а також інтеграції нових функціональних можливостей без значних зусиль з боку розробників.

3.3.5 Переносимість

Програмний продукт розроблено з урахуванням можливості перенесення на різні апаратні та програмні платформи. Архітектура системи забезпечує її незалежність від конкретного апаратного забезпечення, а використання стандартних технологій дозволяє легко адаптувати продукт до нових середовищ експлуатації. Це дає можливість встановлювати систему як на сучасних дата-центрах, так і на менш потужних серверах, забезпечуючи ефективну роботу в різних умовах. Крім того, завдяки використанню вебтехнологій та API, продукт може бути запущений на будь-якому клієнтському пристрою.

3.3.6 Продуктивність

Програмний продукт розроблено з урахуванням високих вимог до продуктивності, що дозволяє системі ефективно обробляти великий обсяг даних у режимі реального часу. Оптимізоване використання серверних ресурсів та алгоритмів кешування на вебклієнті сприяє мінімізації часу на виконання операцій. Завдяки сучасним технологічним рішенням система здатна підтримувати стабільну роботу навіть за високих навантажень, що є критично важливим для оперативного контролю умов зберігання вакцин.

3.4 Вимоги бази даних

Програмний продукт використовує PostgreSQL як систему управління реляційною базою даних. База даних повинна підтримувати основні сутності: вакцина, користувач, запис від IoT-сенсора, повідомлення, локації та їх характеристики. Архітектура бази даних розроблена з урахуванням нормалізації даних до третьої нормальної форми для уникнення надлишковості інформації та забезпечення її цілісності.

Записи про вакцину містять інформацію про температурні обмеження та обмеження вологості. Дані користувачів включають контактну інформацію та ролі в системі, які визначають рівень доступу до функцій. Записи від IoT-сенсорів зберігають дані про показники температури та вологості, що регулярно оновлюються для моніторингу умов зберігання. Повідомлення містять інформацію про сповіщення, які генеруються системою у разі виявлення відхилень, а локації - дані про місце розташування вакцинних складів.

Використання PostgreSQL забезпечує масштабованість та високу продуктивність, підтримуючи складні запити та аналітичні операції, що є критично важливими для оперативного прийняття рішень.

3.5 Інші вимоги

Система повинна бути спроєктована з урахуванням спрощеного розгортання та налаштування, що дозволяє швидко інтегрувати продукт в існуючу інфраструктуру без значних зусиль. Вона має бути сумісною з різними операційними системами, які використовуються на клієнтських пристроях, та підтримувати стандартні веббраузери. Інтерфейс користувача повинен надавати зрозумілі сповіщення про виниклі помилки чи збої.

**4 ДОДАТКОВІ МАТЕРІАЛИ**

1. PostgreSQL. URL: https://www.postgresql.org/ (дата звернення 08.03.2025).
2. TCP/IP. URL: https://en.wikipedia.org/wiki/TCP/IP (дата звернення 08.03.2025).
3. HTTP. URL: https://developer.mozilla.org/en-US/docs/Web/HTTP (дата звернення 08.03.2025).
4. REST API. URL: https://restfulapi.net/ (дата звернення 08.03.2025).
5. UI/UX. URL: https://www.nngroup.com/articles/definition-user-experience/ (дата звернення 08.03.2025).
6. CRUD. URL: https://en.wikipedia.org/wiki/Create,\_read,\_update\_and\_delete (дата звернення 08.03.2025).
7. SQL. URL: https://en.wikipedia.org/wiki/SQL (дата звернення 08.03.2025).