

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ
ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»
(КПІ ім. ІГОРЯ СІКОРСЬКОГО)

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ
кафедра БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

«До захисту допущено»

В.о. завідувач кафедри біомедичної
кібернетики (БМК)

_____ Світлана АЛХІМОВА

“ ___ ” червня 2025р.

Дипломна робота
на здобуття ступеня бакалавра
за освітньо-професійною програмою
«Комп'ютерні технології в біології та медицині»
спеціальності 122 «Комп'ютерні науки»

на тему: Програмний застосунок – «Електронний щоденник ведення вагітності»

Виконав: студент ІV курсу, групи БС-12

САМОЛЮК ТЕТЯНА ГРИГОРІВНА

(прізвище, ім'я, по батькові)

(підпис)

Керівник: *ст.викл. каф БМК, к.т.н., ст.викл.*

Гладка Мирослава Вікторівна

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

(підпис)

Рецензент: *доцент кафедри біомедичної інженерії, к.т.н., доцент*

Сичик Марина Михайлівна

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

(підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.

Студентка

(підпис)

Київ – 2025 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет біомедичної інженерії
Кафедра біомедичної кібернетики

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 122 «Комп'ютерні науки»

Освітньо-професійна програма «Комп'ютерні технології в біології та медицині»

ЗАТВЕРДЖУЮ

В.о. завідувач кафедри БМК

_____ Світлана АЛХІМОВА

« ____ » _____ 2025 р.

ЗАВДАННЯ
на дипломну роботу студенту

САМОЛЮК ТЕТЯНА ГРИГОРІВНА

(прізвище, ім'я, по батькові)

1. Тема роботи Програмний застосунок – «Електронний щоденник ведення вагітності»

Керівник роботи

Гладка Мирослава Вікторівна, ст.викл., к.т.н.,

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

затверджені наказом по університету від «26» травня 2025 р. № 1758-с

2. Термін подання студентом роботи **26 травня 2025р.**
3. Вихідні дані до роботи: *Програмний застосунок – «Електронний щоденник ведення вагітності».*
4. Зміст роботи: *аналіз предметної області та існуючих додатків для моніторингу вагітності, вибір технологій та інструментів для створення веб-застосунку, розробка програмного застосунку «електронний щоденник ведення вагітності», розрахунок економічного ефекту за темою ДР, захист персональної інформації та даних.*
5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): *20 рисунки, 18 таблиці, 23 слайдів (презентація з захисту ДР).*
6. Дата видачі завдання **15 квітня 2025 року.**

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Отримати завдання за темою ДР на практику	До 20.02.2025р.	Виконано
2	Переддипломна практика	За графіком	Виконано
3	Виконання розділів ДР (Вступ, аналітичний огляд літературних джерел, теоретична частина)	До кінця практики	Виконано
4	Виконання розділів ДР (практична частина, загальні висновки, список джерел)	Не пізніше 20.05.2025р.	Виконано
5	Перевірка ДР керівником та допуск її на перевірку нормоконтролером та на плагіат	Не пізніше 21.05.2025р.	Виконано
6	Подання в електронному вигляді ДР та анотації до неї на перевірку нормоконтролера та плагіат (UNICHECK).	Не пізніше 25.05.2025р.	Виконано
7	Надання пакету електронних документів на засідання кафедри	Не пізніше 12.06.2025р.	Виконано
8	Отримати допуск до захисту ДР в ЕК	Згідно засіданню кафедри	Виконано
9	Подання ДР рецензенту. Отримання рецензії.	Не пізніше 14.06.2025р.	Виконано
10	Подання пакету документів в паперовому вигляді по ДР та супровідних до неї документів на захист в ЕК ¹	Не пізніше 14.06.2025р.	Виконано
11	Захист ДР в ЕК	16.06.2025р.– 21.06.2025р.	

Студент

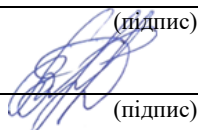


(підпис)

Тетяна САМОЛЮК

(ім'я, ПРІЗВИЩЕ)

Керівник ДР



(підпис)

Мирослава ГЛАДКА

(ім'я, ПРІЗВИЩЕ)

Нормоконтролер

(підпис)

Галина КОРНІЄНКО

(ім'я, ПРІЗВИЩЕ)

¹ не пізніше ніж за 5 днів до затвердженої дати захисту ДР в ЕК

АНОТАЦІЯ

Дипломна робота за темою «Програмний застосунок – «Електронний щоденник ведення вагітності»» виконана студенткою кафедри біомедичної кібернетики ФБМІ Самолюк Тетяною Григорівною зі спеціальності 122 «Комп'ютерні науки» за освітньо-професійною програмою «Комп'ютерні технології в біології та медицині» та складається зі: вступу; 5 розділів (аналіз предметної області та існуючих додатків для моніторингу вагітності, вибір технологій та інструментів для створення веб-застосунку, розробка програмного застосунку «електронний щоденник ведення вагітності», розрахунок економічного ефекту за темою ДР, захист персональної інформації та даних), висновків до кожного з цих розділів; загальних висновків; списку використаних джерел, який налічує 30 джерел. Загальний обсяг роботи 104 сторінок.

Актуальність теми. Вагітність – це один із найважливіших етапів у жінки. Так як вагітність – це не тільки поява нового члена сім'ї, а ще й фізіологічні та емоційні зміни здоров'я жінки. Упродовж 40 тижнів організм майбутньої матері зазнає численних змін, спрямованих на підтримку розвитку плода та підготовку до пологів. У цей час особливо важливо мати доступ до достовірної, структурованої та своєчасної інформації про перебіг вагітності.

Сучасні цифрові технології відкривають нові можливості для покращення медичної поінформованості та підтримки жінок під час вагітності. Одним із таких інструментів є електронний щоденник ведення вагітності, який надає щотижневі рекомендації, опис розвитку плода, а також корисні поради щодо здоров'я та підготовки до материнства. Такий ресурс не лише зменшує тривожність майбутніх батьків, а й сприяє формуванню відповідального ставлення до вагітності, а в подальшому і до малюка.

У зв'язку з цим створення інформативного та зручного у використанні електронного щоденника вагітності є актуальним завданням, що відповідає потребам сучасного суспільства та сприяє покращенню якості життя вагітних жінок.

Мета і завдання роботи.

Метою роботи є створення програмного застосунку для контролю за перебігом вагітності, який дозволяє вагітним жінкам відстежувати свій стан здоров'я, отримувати інформацію та вести персоналізований електронний щоденник.

Для досягнення цієї мети визначено такі основні завдання:

1. Провести аналіз предметної області, потреб користувачів і наявних цифрових рішень.
2. Сформулювати функціональні та нефункціональні вимоги до веб-застосунку.
3. Спроекувати архітектуру застосунку, включаючи серверну частину, базу даних і користувацький інтерфейс.
4. Реалізувати застосунок з використанням сучасних технологій (PyQt6, Node.js, SQLite тощо).
5. Забезпечити збереження та захист персональних медичних даних відповідно до міжнародних та українських вимог.
6. Інтегрувати функціонал нагадувань, календаря, аналітики медичних показників і візуалізації.
7. Провести тестування застосунку з точки зору функціональності, продуктивності та безпеки.
8. Виконати економічну оцінку розробки і довести доцільність впровадження застосунку в медичну практику.

Використані методи. В процесі проведеного аналізу було обрано мову програмування Python та бібліотека PyQt6, обґрунтовано використання наступних технологій: клієнтська частина (CSS), серверна частина (Node.js, API), база даних (SQLite), захист інформації (вхід по коду з Email).

Отримані результати. Використання електронного щоденника для відслідковування вагітності може мати практичне застосування для психологічної підтримки, моніторингу фізичного стану здоров'я, ведення медичних записів пацієнток, особливо коли унеможлиблюється візит до сімейного лікаря.

Публікації. Не заплановано.

Ключові слова. Електронний щоденник, вагітність, триместр, показники, програмний застосунок.

Бібліографічний опис ДР.

Самолюк Т. Г. Програмний застосунок – «Електронний щоденник ведення вагітності»: дипломна роб. бакалавра : 122 Комп'ютері науки / Самолюк Тетяна Григорівна. – Київ, 2025. – 74 с.

ABSTRACT

The diploma work on the topic «Software application - ”Electronic diary of pregnancy management» was completed by *Tetiana Samoliuk*, a student of the *Department of Biomedical Cybernetics of the FBME*. The work was carried out within the framework of the educational-professional program «Computer Technologies in Biology and Medicine» under the specialization of 122 «Computer Science». The diploma consists of an introduction, 5 chapters (*analysis of the subject area and existing applications for pregnancy monitoring, selection of technologies and tools for creating a web application, development of the software application «electronic diary of pregnancy management», calculation of the economic effect on the topic of DW, protection of personal information and data*), conclusions to each of these chapters; general conclusions; a list of references, which includes 30 sources. The total volume of the work is 104 pages.

Relevance of the topic. Pregnancy is one of the most important stages in a woman's life. Pregnancy is not only the arrival of a new family member, but also physiological and emotional changes in a woman's health. Over the course of 40 weeks, the body of the expectant mother undergoes numerous changes aimed at supporting the development of the fetus and preparing for childbirth. At this time, it is especially important to have access to reliable, structured, and timely information about the course of pregnancy.

Modern digital technologies open up new opportunities to improve health awareness and support for women during pregnancy. One of these tools is an electronic pregnancy diary that provides weekly recommendations, a description of fetal development, and useful tips on health and preparation for motherhood. Such a resource not only reduces anxiety of future parents, but also helps to form a responsible attitude towards pregnancy, and later to the baby.

In this regard, the creation of an informative and easy-to-use electronic pregnancy diary is an urgent task that meets the needs of modern society and helps improve the quality of life of pregnant women.

The purpose and tasks of the work. The aim of the work is to create a software application for monitoring the course of pregnancy that allows pregnant women to monitor their health status, receive information and keep a personalized electronic diary.

To achieve this goal, the following main tasks have been identified:

1. Analyze the subject area, user needs, and existing digital solutions.
2. Formulate functional and non-functional requirements for the web application.
3. Design the application architecture, including the server side, database and user interface.
4. Implement the application using modern technologies (PyQt6, Node.js, SQLite, etc.).
5. Ensure the safety and protection of personal medical data in accordance with international and Ukrainian requirements.
6. Integrate the functionality of reminders, calendar, medical indicator analytics and visualization.
7. To test the application in terms of functionality, performance and security.
8. Perform an economic evaluation of the development and prove the feasibility of implementing the application in medical practice.

Used methods. In the course of the analysis, the Python programming language and the PyQt6 library were chosen, and the use of the following technologies was justified: client side (CSS), backend (Node.js, API), database (SQLite), and information security (login via code from Email).

The results obtained. The use of an electronic diary for pregnancy tracking can have practical applications for psychological support, monitoring of physical health, and keeping medical records of patients, especially when a visit to a family doctor is impossible.

Publications. Not planned.

Key words. Electronic diary, pregnancy, trimester, stats, software application.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ	10
ВСТУП.....	12
РОЗДІЛ 1	15
АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ ДОДАТКІВ ДЛЯ МОНІТОРИНГУ ВАГІТНОСТІ	15
1.1 Постановка задачі.....	15
1.2 Аналіз існуючих веб-застосунків для моніторингу вагітності	18
1.3 Визначення функціональних та нефункціональних вимог	23
1.4 Огляд технологій для розробки програмних застосунків	25
Висновки до розділу 1.....	27
РОЗДІЛ 2	29
ВИБІР ТЕХНОЛОГІЙ ТА ІНСТРУМЕНТІВ ДЛЯ СТВОРЕННЯ ПРОГРАМНОГО ЗАСТОСУНКУ.....	29
2.1 Розробка архітектури програмного застосунку	29
2.2 Проектування бази даних для зберігання медичних даних	32
2.3 Розробка інтерфейсу користувача (UI/UX).....	37
2.4 Вибір технологічного стеку для реалізації	40
2.5 Планування заходів із забезпечення безпеки даних.....	44
Висновки до розділу 2.....	46
РОЗДІЛ 3	48
РОЗРОБКА ПРОГРАМНОГО ЗАСТОСУНКУ «ЕЛЕКТРОННИЙ ЩОДЕННИК ВЕДЕННЯ ВАГІТНОСТІ».....	48
3.1 Розробка серверної частини застосунку.....	48
3.2 Реалізація клієнтської частини та інтерфейсу	51
3.3 Інтеграція з зовнішніми сервісами (наприклад, календарі, нагадування)...	61
3.4 Захист персональної інформації та даних.....	64
3.5 Розрахунок економічного ефекту за темою ДР	66
Висновки до розділу 3.....	67
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	71

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

2FA – Двофакторна автентифікація (Two-Factor Authentication) – метод захисту облікових записів шляхом використання двох етапів верифікації.

AES – Стандарт шифрування даних (Advanced Encryption Standard) – алгоритм для захисту чутливих даних.

API – Інтерфейс прикладного програмування (Application Programming Interface) – набір інструментів для взаємодії між компонентами застосунку.

CSRF – Міжсайтова підробка запитів (Cross-Site Request Forgery) – тип атаки, спрямований на виконання несанкціонованих дій від імені користувача.

CSS – Каскадні таблиці стилів (Cascading Style Sheets) – мова для оформлення веб-інтерфейсів.

DOM – Об'єктна модель документа (Document Object Model) – структура для представлення HTML-елементів у веб-застосунках.

eHealth – Електронна система охорони здоров'я – національна платформа для управління медичними даними в Україні.

GDPR – Загальний регламент захисту даних (General Data Protection Regulation) – європейський стандарт захисту персональних даних.

HIPAA – Закон про переносимість і підзвітність медичного страхування (Health Insurance Portability and Accountability Act) – американський стандарт захисту медичних даних.

HTML – Мова розмітки гіпертексту (HyperText Markup Language) – основа для створення веб-сторінок.

HTTP – Протокол передачі гіпертексту (HyperText Transfer Protocol) – протокол для обміну даними в Інтернеті.

HTTPS – Безпечний HTTP (HyperText Transfer Protocol Secure) – захищена версія HTTP із шифруванням SSL/TLS.

JWT – JSON веб-токени (JSON Web Tokens) – стандарт для безпечної передачі даних між клієнтом і сервером.

OWASP – Проєкт безпеки веб-застосунків (Open Web Application Security Project) – організація, що розробляє стандарти безпеки.

RBAC – Управління доступом на основі ролей (Role-Based Access Control) – модель обмеження доступу залежно від ролі користувача.

REST – Представницький стан передачі (Representational State Transfer) – архітектурний стиль для створення API.

SQL – Мова структурованих запитів (Structured Query Language) – мова для роботи з реляційними базами даних.

SSL/TLS – Протоколи безпеки (Secure Sockets Layer/Transport Layer Security) – технології для шифрування мережевого трафіку.

UI – Користувацький інтерфейс (User Interface) – візуальна частина застосунку для взаємодії з користувачем.

УЗД – Ультразвукове дослідження – метод діагностики стану плода під час вагітності.

UX – Користувацький досвід (User Experience) – сукупність вражень користувача від взаємодії із застосунком.

ВООЗ – Всесвітня організація охорони здоров'я – міжнародна організація, що розробляє стандарти охорони здоров'я.

СУБД – Система управління базами даних – програмне забезпечення для роботи з базами даних.

ВСТУП

У сучасному цифровому світі все більше уваги приділяється сучасним рішенням у сфері охорони здоров'я, зокрема, допомозі жінкам на певному етапі вагітності. Перебіг вагітності вимагає регулярного відстеження фізичного та емоційного стану майбутньої матері, регулярної клінічної статистики, контролю відвідувань лікаря та дотримання рекомендацій експертів. Однак використання паперових носіїв інформації або розрізнених джерел записів часто є незручним і ненадійним способом зберігання та відстеження даних про стан здоров'я вагітної жінки.

Розробка цифрового щоденника вагітності у вигляді програмного забезпечення дозволяє централізовано зберігати важливу статистику, автоматизувати нагадування про прийом лікарських препаратів або візити до лікаря, а також полегшити спілкування з сімейним лікарем. Така програма може стати неофіційним помічником для вагітної жінки, даруючи не лише зручність у користуванні, а й підвищуючи якість медичного обслуговування завдяки точному та своєчасному контролю даних.

Крім того, зростаюче визнання мобільних додатків у медичній сфері (mHealth) підтверджує бажання створювати спеціалізовані віртуальні пристрої для певних класів користувачів. З огляду на розширений заклик до надання далекосяжних послуг через глобальні виклики (до яких належить і пандемія COVID-19), значення доступних цифрових відповідей на питання ведення вагітності зростає найшвидше.

Таким чином, вдосконалення програмного забезпечення Електронного щоденника вагітності – це прикладне і соціально величезне починання, яке поєднує в собі турботу про здоров'я жінки, появу сучасних технологій і прагнення підвищити ефективність ведення дієт, профілактики та моніторингу стану здоров'я.

Мета і завдання роботи.

Метою роботи є створення програмного застосунку для контролю за перебігом вагітності, який дозволяє вагітним жінкам відстежувати свій стан

здоров'я, отримувати інформацію та вести персоналізований електронний щоденник.

Для досягнення цієї мети визначено такі основні завдання:

1. Провести аналіз предметної області, потреб користувачів і наявних цифрових рішень.
2. Сформувати функціональні та нефункціональні вимоги до веб-застосунку.
3. Спроекувати архітектуру застосунку, включаючи серверну частину, базу даних і користувацький інтерфейс.
4. Реалізувати застосунок з використанням сучасних технологій (PyQt6, Node.js, PostgreSQL тощо).
5. Забезпечити збереження та захист персональних медичних даних відповідно до міжнародних та українських вимог.
6. Інтегрувати функціонал нагадувань, календаря, аналітики медичних показників і візуалізації.
7. Провести тестування застосунку з точки зору функціональності, продуктивності та безпеки.
8. Виконати економічну оцінку розробки і довести доцільність впровадження застосунку в медичну практику.

Використані методи. В процесі проведеного аналізу було обрано мову програмування Python та бібліотека PyQt6, обґрунтовано використання наступних технологій: клієнтська частина (CSS), серверна частина (Node.js, API), база даних (SQLite), захист інформації (вхід по коду з Email).

Отримані результати. Використання електронного щоденника для відслідковування вагітності може мати практичне застосування для психологічної підтримки, моніторингу фізичного стану здоров'я, ведення медичних записів пацієнток, особливо коли унеможлиблюється візит до сімейного лікаря.

Публікації. Не заплановано.

Структура роботи

Дипломна робота за темою «Програмний застосунок – «Електронний щоденник ведення вагітності»» виконана студентом *Самолук Тетяною*

Григорівною зі спеціальності 122 «Комп'ютерні науки» за освітньо-професійною програмою «Комп'ютерні технології в біології та медицині», побудована за класичним типом та викладена на 104 сторінках машинописного тексту. Вона складається з: вступу; 5 розділів (аналіз предметної області та існуючих додатків для моніторингу вагітності, вибір технологій та інструментів для створення веб-застосунку, розробка програмного застосунку «електронний щоденник ведення вагітності», розрахунок економічного ефекту за темою ДР, захист персональної інформації та даних), висновків до кожного з цих розділів; загальних висновків; списку використаних джерел, який налічує 30 джерел. В роботі представлено 20 рисунків і 18 таблиць.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ ДОДАТКІВ ДЛЯ МОНІТОРИНГУ ВАГІТНОСТІ

1.1 Постановка задачі

Майбутній матері обов'язково потрібно вести щоденник, з допомогою якого вона зможе розуміти, коли їй потрібно на прийом до лікаря, пакет аналізів для задачі чи дивитись як розвивається малюк в утробі матері.

Ведення вагітності потребує постійного нагляду, оскільки від цього залежить добробут як жінки, так і її майбутньої дитини. Сам процес виношування супроводжується складними змінами в організмі, тому потребує систематичних оглядів, дотримання рекомендацій лікарів і уважного ставлення до сигналів тіла. У цьому контексті цифрові інструменти, зокрема інтернет-платформи, набувають дедалі більшого значення. Вони дозволяють зручно відстежувати стан здоров'я, зберігати важливу інформацію і підтримувати зв'язок із медичними фахівцями. Щоб створити дійсно корисний веб-застосунок, слід чітко розуміти потреби різних груп користувачів — передусім це майбутні матері, але також і їхні партнери, родичі, лікарі.

Основними користувачами таких рішень зазвичай виступають самі вагітні жінки. Вони прагнуть мати під рукою надійний інструмент, що допоможе їм не лише фіксувати перебіг вагітності, а й краще орієнтуватися в змінах, що відбуваються з тілом. Значна частина звертає увагу на функції, що дозволяють відзначати ключові події — наприклад, візити до медиків, проведення УЗД чи здачу аналізів. Не менш важливими є модулі для введення особистих даних, як-от тиск, вага або наявність набряків, адже саме ці спостереження допомагають вчасно помітити щось незвичне. Відчуття контролю над ситуацією часто заспокоює жінку, знижує рівень тривожності і дає змогу швидко реагувати на симптоми, які потребують уваги лікаря.

Не варто забувати й про фахівців, які супроводжують вагітність. Акушери та гінекологи, за умов правильного підходу до структури застосунку, можуть

отримувати віддалений доступ до щоденників стану здоров'я своїх пацієнток. Такий підхід особливо цінний у сільських районах або під час обмежень на пересування. Уведені пацієнткою показники — наприклад, дані про тиск чи скарги на самопочуття — допомагають медикам швидше реагувати і коригувати подальші дії. Зі свого боку, лікар може в застосунку залишати індивідуальні рекомендації або нагадування про обстеження, що робить взаємодію з пацієнткою більш ефективною.

Родичі, особливо партнери, нерідко прагнуть долучитися до процесу очікування дитини, однак не завжди мають чітке уявлення про те, яку саме підтримку можуть надати. Їхнє залучення є значущим не лише з емоційного погляду, а й у практичному вимірі — розуміння потреб вагітної та здатність реагувати на її фізичний і психологічний стан сприяють формуванню сприятливого середовища протягом усієї вагітності. Програмний застосунок може відігравати роль посередника, надаючи партнерам доступ до інформації про поточний етап, поради щодо взаємодії, а також нагадування про важливі події [4]. Програмний застосунок, у якому передбачено сповіщення про ключові етапи — як-от початок нового триместру чи підготовка до пологів — може слугувати корисним інструментом. Подібні функції дозволяють партнерам своєчасно виявляти підтримку та краще розуміти, через що проходить майбутня мати. У розділах з порадами щодо емоційної підтримки та психологічного комфорту можна знайти практичні рекомендації, які зроблять цей період легшим для обох.

Якісно реалізований програмний застосунок може стати потужним ресурсом не лише для жінки, що очікує на дитину, але й для її близьких та лікарів. Його користь полягає в тому, що він поєднує різні аспекти догляду: від відстеження фізіологічного стану до забезпечення інформаційної підтримки і налагодження комунікації між усіма учасниками процесу. В основі розробки такого рішення має лежати глибоке розуміння реальних потреб майбутніх користувачів і бажання зробити період очікування дитини безпечнішим і комфортнішим.

Щоб цифровий продукт був корисним для широкого кола людей, потрібно враховувати обставини, у яких відбувається його використання. Не кожна

користувачка має потужний смартфон або стабільне з'єднання з мережею. Саме тому важливо забезпечити коректну роботу навіть на старіших пристроях, передбачити можливість користування офлайн-версією і зробити інтерфейс максимально зрозумілим — без складних меню та перевантажених розділів. Особливо це стосується жінок із сільських громад, де технічні умови можуть бути суттєво відмінними від міських [4].

Успішність взаємодії з додатком значною мірою залежить і від того, наскільки зміст відповідає звичному середовищу користувачки. У контексті України варто передбачити україномовне оформлення як основне, не забуваючи при цьому про можливість перемикання на інші мови для тих, кому це зручніше. Інформаційні матеріали мають бути сформовані з урахуванням реальної медичної практики, зокрема базуватись на положеннях, затверджених Міністерством охорони здоров'я [3]. Це дозволяє уникнути розбіжностей між тим, що читає користувачка, і тим, що говорить лікар.

Зважаючи на чутливість інформації, яка зберігається у застосунку, постає питання надійного захисту даних. Люди очікують, що особисті записи не потраплять у чужі руки й не будуть використані без їхнього відома. Система має гарантувати приватність: і через технічні рішення (шифрування, перевірка доступу), і через прозору політику зберігання та обробки інформації відповідно до українського законодавства і міжнародних вимог, зокрема GDPR [2].

Функціонал, побудований на таких засадах, може охоплювати широкий спектр можливостей — від фіксації візитів до лікаря до перегляду статей, спілкування з медиками або іншими користувачами. Але кожен елемент повинен залишатися простим у використанні, бути зрозумілим, працювати навіть без ідеального інтернету й гарантувати повну безпеку даних. Усе це разом здатне не просто полегшити контроль за станом здоров'я, а й зробити сам процес вагітності спокійнішим і впорядкованішим.

1.2 Аналіз існуючих веб-застосунків для моніторингу вагітності

На сьогодні на глобальному ринку представлено чимало цифрових інструментів, призначених для тих, хто очікує дитину. Ці платформи здебільшого поєднують кілька ключових функцій: фіксацію змін у самопочутті, нагадування про візити до лікаря, базу знань про перебіг вагітності та, у деяких випадках, можливість обміну даними з медичними працівниками. Огляд наявних сервісів дозволяє оцінити, які рішення справді зручні у користуванні, які функції залишаються недостатньо реалізованими, а також у яких аспектах бракує адаптації до реалій українського користувача.

Серед найбільш розпізнаваних платформ можна назвати Pregnancy+ — один із застосунків, що отримав широке розповсюдження і має як мобільну, так і веб-версію. Його структура побудована навколо календаря змін, який оновлюється відповідно до тижня вагітності, містить поради, статті та інтерактивні інструменти — наприклад, графік набору ваги чи підрахунок рухів плода [6]. Візуальний складник відіграє помітну роль: 3D-зображення дитини всередині утробы формують відчуття близькості до процесу й роблять користування привабливішим. Однак, незважаючи на розвинений функціонал, цей застосунок не пропонує повноцінної підтримки українською мовою, що ускладнює доступність для значної частини населення. Окрім того, розширені можливості — зокрема синхронізація з медичними картками — доступні лише в комерційній версії, що може бути обмеженням для користувачів, які шукають безкоштовне рішення.

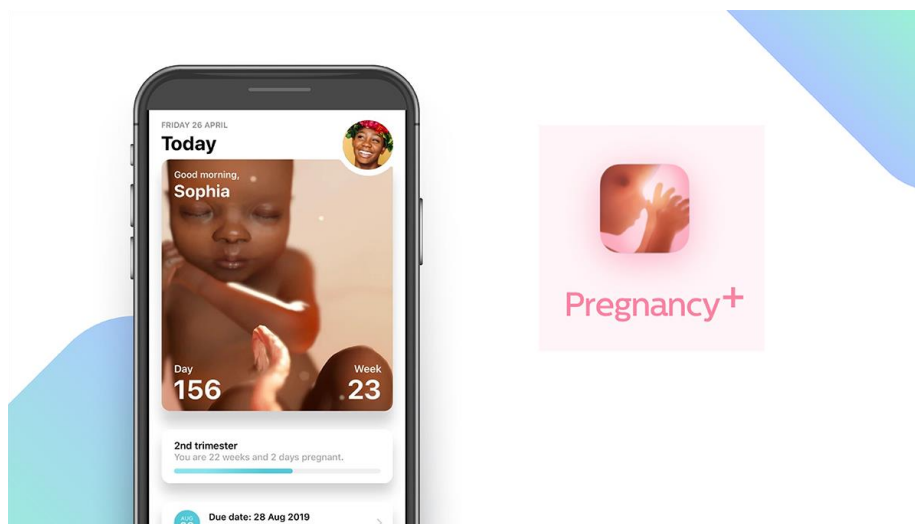


Рисунок 1.1 – Інтерфейс Pregnancy+

Ще одним прикладом цифрового рішення є BabyCenter — платформа, яка об'єднує великий обсяг тематичного матеріалу: від довідкових статей до відеороликів і відкритих обговорень між користувачками. Окрім цього, сервіс дозволяє стежити за розвитком дитини відповідно до календаря, формувати індивідуальний план ведення вагітності та отримувати сповіщення про медичні огляди [7]. Структура платформи розроблена відповідно до принципів сучасного дизайну користувацького досвіду, завдяки чому користування є логічним і передбачуваним. Проте, цей ресурс не враховує українські медичні стандарти та не передбачає прямої участі лікаря у цифровому моніторингу стану пацієнтки, що знижує ефективність у контексті телемедичної практики.

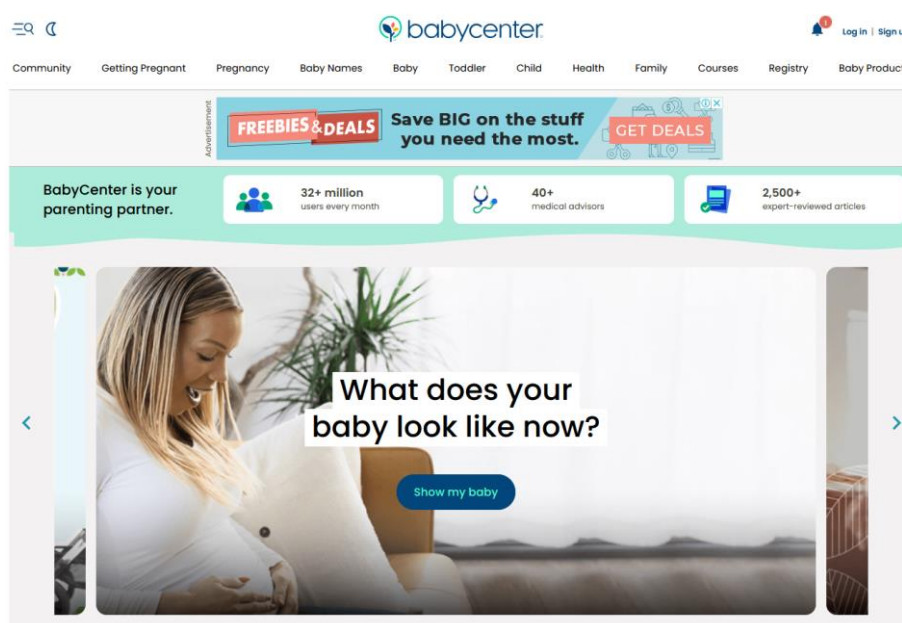


Рисунок 1.2 – Головна сторінка веб-платформи BabyCenter

Серед рішень, адаптованих до умов українського середовища, варто згадати платформу System Carebits, яка має часткове підключення до електронної системи охорони здоров'я eHealth. За її допомогою можна записатися на прийом, ознайомитися з результатами лабораторних досліджень і отримати сповіщення про планові обстеження [29]. Цей застосунок відповідає чинному законодавству України щодо захисту персональної інформації, що підвищує довіру серед користувачів.



Рисунок 1.3 – Інтерфейс платформи System Carebits для запису на прийом

До міжнародних платформ із розширеним функціоналом належить також Ovia Pregnancy. Цей сервіс дозволяє зберігати показники здоров'я, зокрема артеріальний тиск, вагу та якість сну, а також аналізувати зібрані дані. На основі введеної інформації система формує прогнози та рекомендації відповідно до клінічних орієнтирів, затверджених Американським коледжем акушерів і гінекологів [19]. Додатково реалізована синхронізація з календарями, що полегшує планування відвідувань лікаря. Основними недоліками залишаються відсутність локалізації українською мовою і залежність від доступу до інтернету, оскільки платформа не підтримує автономний режим.



Рисунок 1.4 – Модуль відстеження здоров'я в Ovia Pregnancy

Порівняльний огляд зазначених сервісів дає змогу краще зрозуміти сильні й вразливі сторони кожного з них. У табличному форматі узагальнено рівень реалізації ключових функцій, відповідність нормам захисту даних, наявність мовної адаптації та зручність інтерфейсу (табл. 1.1).

Таблиця 1.1

Порівняльний аналіз веб-застосунків для моніторингу вагітності

Застосунок	Функціонал	Безпека даних	Локалізація українською	Зручність інтерфейсу	Офлайн-режим
Pregnancy+	Календар, лічильник рухів, статті	Шифрування, GDPR	Часткова	Висока	Ні
BabyCenter	Інформація, форуми, календар	HIPAA, шифрування	Відсутня	Висока	Ні
System Carebits	Запис до лікаря, eHealth інтеграція	Відповідність ЗУ	Повна	Середня	Частковий
Ovia Pregnancy	Аналітика, відстеження здоров'я	GDPR, HIPAA	Відсутня	Висока	Ні

Результати порівняння свідчать про те, що більшість зарубіжних сервісів — зокрема Pregnancy+, BabyCenter і Ovia Pregnancy — демонструють добре продуману структуру та широкий набір функцій. Усі три платформи мають привабливий візуальний супровід, чітку навігацію та інструменти для аналізу стану здоров'я. Разом із тим, жодна з них не пристосована до українського середовища в повному обсязі. Відсутність україномовного інтерфейсу, неврахування національних медичних протоколів і брак можливості синхронізації з локальними системами охорони здоров'я знижує ефективність використання цих застосунків в умовах України [2].

System Carebits, попри скромніший набір можливостей, забезпечує базову функціональність із врахуванням місцевого законодавства й медичної інфраструктури. Користувачі можуть реєструватися на прийоми, переглядати результати аналізів і отримувати сповіщення відповідно до національного медичного календаря. Проте недосконалий інтерфейс і відсутність розширених функцій — таких як ведення щоденника самопочуття, підтримка від психолога чи візуалізація розвитку плода — обмежують потенціал платформи як повноцінного інструмента щоденного супроводу.

Жоден із розглянутих варіантів не передбачає вбудованої системи психологічної підтримки або режиму роботи без інтернет-з'єднання. Це суттєво знижує їхню практичність у регіонах, де спостерігаються перебої з доступом до цифрових сервісів або низька щільність покриття мобільною мережею.

У сфері захисту персональної інформації всі платформи дотримуються ключових положень міжнародних стандартів, таких як GDPR або HIPAA. System Carebits, крім цього, забезпечує відповідність нормам українського законодавства [7]. Разом із тим, окремі закордонні сервіси мають обмежену відкритість щодо обробки даних користувачів, що може викликати занепокоєння з боку аудиторії, особливо у питаннях конфіденційності.

Отже, можна констатувати, що жоден із наявних веб-застосунків не повною мірою відповідає очікуванням українських користувачів. Найтипівіші прогалини охоплюють брак локалізації, відсутність прив'язки до українських електронних сервісів охорони здоров'я, ігнорування аспектів емоційної підтримки та

відсутність можливості працювати в автономному режимі. Ці недоліки варто врахувати під час розроблення нового рішення, яке б поєднувало адаптований контент, інтеграцію з eHealth, інструменти для емоційної підтримки й автономну функціональність, що дозволить ефективно задовольнити потреби української аудиторії.

1.3 Визначення функціональних та нефункціональних вимог

Розроблення цифрового інструменту для моніторингу перебігу вагітності передбачає чітке окреслення того, яким має бути функціонал системи і за якими критеріями оцінюватиметься її якість. На цьому етапі формується структура вимог, що відображає як очікування кінцевих користувачів, так і технічні параметри, потрібні для стабільної роботи застосунку в умовах реального середовища.

Функціональні характеристики (табл. 1.2) спираються на сценарії використання платформи трьома основними групами: майбутніми матерями, медичним персоналом і близькими членами родини. Одним із ключових компонентів є календар вагітності — з позначенням важливих подій (огляди, УЗД, аналізи), а також функцією сповіщень, які можна отримувати через push-механізми. Для фіксації самопочуття передбачено модуль введення особистих даних: показники артеріального тиску, зміни у вазі, інтенсивність рухів дитини, скарги на стан здоров'я. Ці записи можуть бути використані як для самоспостереження, так і для взаємодії з лікарем [8].

До необхідних розділів входить і тематичний блок із перевіреною інформацією. Йдеться не лише про опис фізіологічних змін протягом вагітності, а й про поради щодо харчування, фізичної активності, грудного вигодовування, підготовки до пологів.

Крім того, окремий набір функцій може бути запропонований партнерам та іншим членам родини — зокрема, доступ до подій календаря, короткі пояснення щодо підтримки жінки на кожному етапі.

Якісні властивості системи окреслюються через нефункціональні вимоги (табл. 1.3). Одне з першочергових питань — захист інформації. Уся взаємодія із застосунком має відповідати вимогам GDPR та українському закону «Про захист персональних даних», що передбачає шифрування інформації.

Повноцінна локалізація українською мовою є базовою умовою для застосунку, який позиціонується на внутрішньому ринку. Разом із тим, варто забезпечити мовне перемикання, аби не обмежувати доступ для інших категорій користувачів [3]. В умовах нестабільного інтернет-з'єднання (особливо в окремих областях) цінною є можливість зберігати частину даних локально, щоб користувач міг працювати з основними функціями — переглядати розклад, вносити показники — і лише після відновлення зв'язку синхронізувати інформацію із сервером [4].

Таблиця 1.2

Функціональні вимоги до веб-застосунку

№	Вимога	Опис	Пріоритет
1	Персоналізований календар	Створення календаря вагітності з нагадуваннями про візити до лікаря, УЗД, аналізи	Високий
2	Введення медичних даних	Модуль для введення даних (тиск, вага, рухи плода) із збереженням і аналізом	Високий
3	Інформаційний розділ	Перевірений контент про вагітність, харчування, підготовку до пологів	Середній
4	Взаємодія з лікарями	Передача даних лікарям, консультації через текст або інтеграцію з eHealth	Високий
5	Психологічна підтримка	Доступ до медитацій, дихальних вправ, форуму для спілкування	Середній
6	Підтримка членів сім'ї	Інформація та нагадування для партнерів і родичів	Низький

Таблиця 1.3

Нефункціональні вимоги до програмного застосунку

№	Вимога	Опис	Пріоритет
1	Безпека даних	Відповідність GDPR, Закону України, шифрування, двофакторна автентифікація	Високий
2	Продуктивність	Час завантаження < 2 с, підтримка до 10 000 одночасних користувачів	Високий
3	Доступність	Адаптивний дизайн, кросбраузерна сумісність, підтримка різних пристроїв	Високий
4	Локалізація	Повна локалізація українською, підтримка інших мов	Високий
5	Офлайн-режим	Доступ до базових функцій без інтернету з подальшою синхронізацією	Середній

Сформульовані критерії створюють основу для створення конкурентоспроможного цифрового продукту, який відповідатиме як очікуванням користувачів, так і міжнародним стандартам. Поєднання національних специфік — лінгвістичних, правових, медичних — із технічними вимогами до безпеки, продуктивності та стабільності забезпечує умови для впровадження застосунку, здатного ефективно працювати в межах української системи охорони здоров'я.

1.4 Огляд технологій для розробки програмних застосунків

Реалізація веб-застосунку для контролю перебігу вагітності вимагає виваженого підходу до вибору технологій, які здатні гарантувати відповідність функціональним і нефункціональним вимогам, описаним на попередньому етапі. Кінцева архітектура має поєднувати стабільність, швидкодію, безпеку та зручність використання як для користувачів, так і для розробників. Огляд сучасних технологічних рішень охоплює вибір інструментів для побудови інтерфейсу, організації серверної логіки, керування даними, забезпечення безпеки та тестування системи.

Інтерфейсна частина проєкту повинна бути адаптивною, зрозумілою та швидкою у взаємодії. Серед фреймворків Python найбільш доцільним є використання PyQt6 — бібліотеки з відкритим кодом, орієнтованої на створення компонентних інтерфейсів [28]. Широка база розробників робить цей інструмент гнучким і надійним для реалізації складної логіки, зокрема календаря подій і динамічного відображення медичних записів. Для оформлення зовнішнього вигляду інтерфейсу рекомендовано використати CSS, що дозволяє швидко будувати сучасні дизайни з мінімальними затратами часу на написання власних стилів [39].

На серверній частині доцільним вибором є Node.js у поєднанні з Express — легким фреймворком для створення API [28]. Асинхронна архітектура Node.js дозволяє ефективно обробляти паралельні запити, що відповідає вимозі підтримки великої кількості одночасних сесій..

Обробка повідомлень у режимі реального часу, наприклад, сповіщення про візити або результати обстежень, може бути реалізована через WebSocket або бібліотеку Socket.IO, яка легко інтегрується з Node.js і не вимагає складної конфігурації [28].

Зберігання даних передбачає використання реляційної СУБД, що гарантує точну структуру й підтримку транзакцій. SQLite є придатною базою для таких сценаріїв: вона підтримує перевірку цілісності даних, складні SQL-запити, гнучке управління доступом і відповідає вимогам безпеки.

Захист персональних і медичних даних реалізується через комплексний набір заходів: шифрування HTTPS для всіх з'єднань, застосування сертифікатів SSL/TLS, використання алгоритмів шифрування AES-256 для конфіденційної інформації в базі даних [37]. Аутентифікація базується на JSON Web Tokens (JWT), що дає змогу надійно передавати ідентифікаційні дані між клієнтом і сервером. Для додаткового рівня захисту облікових записів впроваджується двофакторна перевірка особи, реалізована за допомогою бібліотек типу Speakeasy [37].

Усі технології, що розглядаються, були оцінені за критеріями ефективності, рівня підтримки, простоти реалізації та відповідності стандартам безпеки.

Порівняльна таблиця, складена на основі аналізу, дає змогу сформуванати цілісне бачення рекомендованого стеку (табл. 1.4).

Таблиця 1.4

Порівняння технологій для розробки веб-застосунку

Технологія	Категорія	Продуктивність	Безпека	Легкість використання	Підтримка спільноти
PyQt6	Frontend	Висока	Висока	Середня	Висока
Vue.js	Frontend	Висока	Висока	Висока	Середня
SQLite	База даних	Висока	Висока	Середня	Висока
CSS	Стилізація	Висока	Висока	Висока	Висока

Для розробки пропонується використати PyQt6 і CSS для клієнтської частини, Node.js — для серверної логіки, SQLite — як основну базу даних, Redis — для кешування. Захист інформації забезпечується за рахунок використання HTTPS, JWT, 2FA, а контроль якості — через Jest, Mocha, Postman і JMeter. Така конфігурація відповідає вимогам щодо продуктивності, адаптації до локальних умов, безпеки медичних даних і потенціалу масштабування [3].

Висновки до розділу 1

У першому розділі було здійснено системний огляд основних аспектів, пов'язаних із проектуванням веб-застосунку для супроводу вагітності, що дозволило закласти концептуальне підґрунтя для подальших етапів розробки. Вивчення потреб потенційних користувачів — серед яких ключову роль відіграють вагітні жінки, лікарі та найближче оточення — продемонструвало актуальність створення інструменту, який поєднує функції самостереження, доступу до надійної медичної інформації, емоційної підтримки та можливості контакту з медичним персоналом [1]. З огляду на специфіку українського контексту, зокрема нерівномірний доступ до медичних послуг у сільських

групадах, у структуру майбутнього сервісу мають бути інтегровані функції, що компенсують ці обмеження [3].

Порівняння з уже наявними цифровими рішеннями, такими як Pregnancy+, BabyCenter, System Carebits і Ovia Pregnancy, дозволило виявити сильні сторони — зручність інтерфейсу, наявність освітніх матеріалів, а також інформативний супровід користувача. Разом із тим, ці платформи мають низку суттєвих недоліків, які знижують їхню ефективність в умовах України: відсутність повноцінної україномовної локалізації, обмежені можливості взаємодії з українською медичною інфраструктурою та недооцінка потреб у психологічній підтримці [6]. Ці аспекти вказують на необхідність створення локалізованого веб-застосунку, який ураховує правові, культурні та системні особливості середовища.

Сформовані функціональні вимоги передбачають реалізацію таких компонентів, як персоналізований календар із можливістю налаштування нагадувань, модуль введення показників здоров'я, інформаційний блок із перевіреним вмістом, канал комунікації з лікарем і платформа для емоційної підтримки. Водночас нефункціональні вимоги зосереджуються на гарантуванні безпеки медичних і персональних даних відповідно до вимог GDPR і законодавства України, стабільній роботі застосунку навіть за високого навантаження, повній україномовній локалізації та можливості використання офлайн-режиму для базових функцій [7].

На основі цих вимог було визначено технології, здатні забезпечити відповідний рівень якості: для клієнтської частини — PyQt6 у поєднанні з CSS, для серверної логіки — Node.js, для роботи з даними — SQLite, а також Redis для кешування. Захист даних забезпечується через HTTPS, JWT і двофакторну автентифікацію, а тестування покладено на інструменти Jest і Mocha [28]. Такий стек гарантує достатню швидкодію, надійність та гнучкість архітектури, а також відповідність галузевим стандартам і вимогам до інтеграції з національними сервісами охорони здоров'я [17].

РОЗДІЛ 2

ВИБІР ТЕХНОЛОГІЙ ТА ІНСТРУМЕНТІВ ДЛЯ СТВОРЕННЯ ПРОГРАМНОГО ЗАСТОСУНКУ

2.1 Розробка архітектури програмного застосунку

Архітектура веб-застосунку визначає спосіб організації його компонентів, їхню взаємодію, логіку обробки запитів і передачі даних, а також засоби гарантування безпеки, масштабованості й адаптивності. У випадку системи, призначеної для моніторингу перебігу вагітності, особливу увагу необхідно приділити захисту медичної інформації, гнучкій структурі API для інтеграції із зовнішніми сервісами, а також зручності використання на різних типах пристроїв. З урахуванням визначених вимог у попередньому розділі, для розробки обрано клієнт-серверну модель із REST-архітектурою, яка поєднує структурну прозорість, масштабованість і можливість обробки запитів у реальному часі.

Функціональність веб-застосунку реалізується через взаємодію клієнтської, серверної та базової (рівня даних) частин, які обмінюються інформацією через стандартизовані REST-запити. На стороні користувача функціонує інтерфейс, створений за допомогою бібліотеки PyQt6, що дозволяє компонувати структуру з окремих елементів: календару подій, панелі введення показників стану здоров'я, інформаційних блоків. Використання CSS забезпечує адаптивну верстку, завдяки чому інтерфейс однаково ефективно відображається на комп'ютерах, планшетах і мобільних пристроях [39].

Обробка логіки та запитів покладена на серверний рівень, реалізований із використанням Node.js та Express. Асинхронна модель обробки дозволяє системі ефективно працювати при великій кількості одночасних підключень, що є критичним для забезпечення стабільності роботи при високому навантаженні [28]. Express дозволяє формувати структуровану систему ендпоінтів для збереження, оновлення й передачі медичних даних. Для підтримки повідомлень у реальному часі використовується Socket.IO, який дозволяє реалізовувати push-механізми

(наприклад, для нагадувань про обстеження або отримання повідомлень від лікаря) [34].

Зберігання структурованих медичних записів, профілів користувачів, даних про візити, аналізи й динаміку самопочуття здійснюється у базі даних PostgreSQL. Ця СУБД забезпечує підтримку транзакцій, складних запитів і функцій шифрування даних, що дозволяє відповідати вимогам до безпеки медичної інформації [28]. Для підвищення швидкодії та зменшення навантаження на основну базу застосовується Redis, який слугує кешем для тимчасового збереження часто запитуваних даних (наприклад, поточних налаштувань або останніх записів пацієнтки) [28].

Захист інформації організований на всіх етапах обробки: від користувацького введення до збереження в базі. Весь трафік між клієнтом і сервером шифрується за допомогою HTTPS. Аутентифікація користувачів реалізується через JSON Web Tokens (JWT), які забезпечують захищену ідентифікацію без необхідності повторної перевірки кожного запиту [37]. Доступ до облікових записів додатково захищено двофакторною перевіркою. На рівні сервера впроваджено бібліотеку Helmet, яка дозволяє мінімізувати ризики, пов'язані з XSS-атаками, підміною заголовків і витоком сесій [26]. Для медичних даних використовується шифрування відповідно до алгоритму AES-256, що гарантує відповідність регламентам GDPR і українському законодавству [7].

Застосунок також інтегрується з низкою зовнішніх сервісів. Передбачено взаємодію з eHealth для доступу до офіційних медичних даних і можливості запису на прийом до лікаря [29, 30]. Для забезпечення емоційної підтримки в майбутніх версіях проєкту можлива інтеграція з зовнішніми платформами, що пропонують медитативні або дихальні практики, наприклад, Calm [10].

Запропонована структура архітектури є багаторівневою, чітко поділеною за відповідальністю й здатною до масштабування. Вона дозволяє забезпечити надійне функціонування застосунку, стабільний обмін даними між клієнтом і сервером, високий рівень безпеки, а також інтеграцію з національними й міжнародними системами охорони здоров'я.

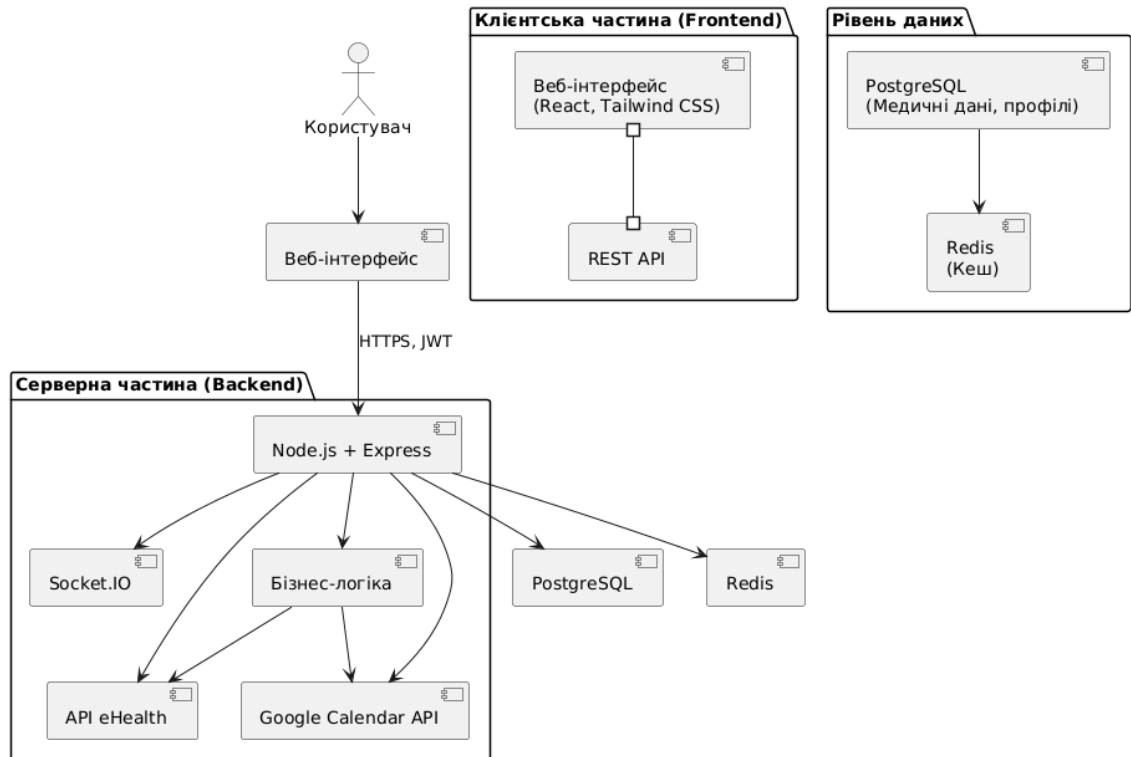


Рисунок 2.1 – Схема архітектури програмного застосунку

Клієнт-серверна архітектура з RESTful API обрана через її здатність забезпечити чітке розділення обов'язків між клієнтом і сервером, що полегшує розробку, тестування та масштабування [34]. Використання мікросервісів дозволяє незалежно оновлювати окремі модулі, наприклад, модуль психологічної підтримки, без впливу на інші частини застосунку. SQLite забезпечує баланс між надійністю зберігання даних і швидкістю доступу, що відповідає вимогам до продуктивності та безпеки [28]. Інтеграція з eHealth та електронною адресою враховує локальні та функціональні потреби, а заходи безпеки (HTTPS, JWT, шифрування) гарантують відповідність стандартам [7].

Спроектowana архітектура є оптимальною для веб-застосунку контролю перебігу вагітності, оскільки вона підтримує всі ключові вимоги: функціональність, продуктивність, безпеку та інтеграцію з зовнішніми сервісами. Схема архітектури чітко відображає взаємодію компонентів, що полегшує подальшу реалізацію проєкту.

2.2 Проектування бази даних для зберігання медичних даних

База даних у веб-застосунку для контролю перебігу вагітності необхідна для зберігання, обробки та швидкого доступу до інформації. Вона має підтримувати роботу з обліковими записами користувачів, медичними показниками, подіями календаря, нагадуваннями та налаштуваннями. На основі попередньо визначених вимог обрано реляційну модель, реалізовану через SQLite. Такий підхід забезпечує впорядковану структуру з дотриманням вимог до цілісності даних і захисту інформації відповідно до GDPR і Закону України «Про захист персональних даних» [7].

Реляційна модель підходить для роботи зі структурованими даними, такими як медичні записи, показники здоров'я та календарні події [28]. SQLite підтримує складні запити, транзакції та засоби шифрування, що важливо для медичних застосунків. Варіанти на основі нереляційних СУБД, наприклад MongoDB, менш ефективні в цьому випадку, оскільки структура зберігається стабільною і не передбачає постійних змін [34]. Для прискорення доступу до часто використовуваної інформації, як-от налаштування чи нагадування, застосовується Redis — інструмент кешування, який дозволяє зменшити навантаження на основну базу і скоротити час відповіді сервера [28].

База даних побудована на основі п'яти основних сутностей, кожна з яких відповідає окремому функціональному напрямку веб-застосунку. Сутність Users охоплює інформацію про всіх зареєстрованих осіб, включаючи вагітних жінок, медичних фахівців і членів сім'ї, які мають доступ до системи. Тут зберігаються базові облікові дані, ролі користувачів та час останньої активності (табл. 2.1).

Сутність CalendarEvents відповідає за збереження інформації про заплановані дії, наприклад, візити до лікаря, УЗД чи інші обстеження. Ці події можуть бути синхронізовані з зовнішніми календарями, зокрема Google Calendar (табл. 2.3).

У свою чергу, Reminders зберігають налаштування сповіщень, які пов'язані з календарними подіями. Кожне нагадування містить дані про час надсилання, тип

події та спосіб доставки повідомлення (наприклад, через мобільний застосунок або електронну пошту) (табл. 2.4). Остання сутність — Settings — охоплює індивідуальні параметри користувача, такі як мова інтерфейсу, бажаний формат відображення даних або правила сповіщень (табл. 2.5). Усі сутності побудовані з урахуванням вимог до безпеки й структурованості. Для кожної з них визначено перелік атрибутів, типи даних і відповідні обмеження, які забезпечують правильну логіку роботи всієї системи.

Таблиця 2.1

Сутність Users

Атрибут	Тип даних	Обмеження	Опис
user_id	UUID	Первинний ключ, NOT NULL	Унікальний ідентифікатор користувача
email	VARCHAR(255)	UNIQUE, NOT NULL	Електронна пошта для входу
password_hash	VARCHAR(255)	NOT NULL	Хеш пароля (зашифрований)
role	ENUM	NOT NULL (patient, doctor, family)	Роль користувача
full_name	VARCHAR(100)	NOT NULL	ПІБ користувача
created_at	TIMESTAMP	NOT NULL, DEFAULT NOW()	Дата створення профілю

Сутність MedicalRecords призначена для збереження медичних показників, які вводяться вручну або імпортуються з інших систем. До таких даних належать вимірювання артеріального тиску, вага, частота рухів плода тощо. Кожен запис фіксується з прив'язкою до певного моменту часу, що дає змогу відстежувати динаміку стану здоров'я [8] (табл. 2.2).

Сутність MedicalRecords

Атрибут	Тип даних	Обмеження	Опис
record_id	UUID	Первинний ключ, NOT NULL	Унікальний ідентифікатор запису
user_id	UUID	Зовнішній ключ (Users), NOT NULL	Користувач, якому належить запис
blood_pressure	VARCHAR(10)	NULL	Артеріальний тиск (наприклад, 120/80)
weight	DECIMAL(5,2)	NULL	Вага (кг)
fetal_movement	INTEGER	NULL	Кількість рухів плода за період
symptoms	TEXT	NULL	Скарги чи симптоми
recorded_at	TIMESTAMP	NOT NULL	Дата і час запису

Сутність CalendarEvents

Атрибут	Тип даних	Обмеження	Опис
event_id	UUID	Первинний ключ, NOT NULL	Унікальний ідентифікатор події
user_id	UUID	Зовнішній ключ (Users), NOT NULL	Користувач, якому належить подія
event_type	VARCHAR(50)	NOT NULL	Тип події (візит, УЗД, аналізи)
event_date	DATE	NOT NULL	Дата події
description	TEXT	NULL	Опис події
created_at	TIMESTAMP	NOT NULL, DEFAULT NOW()	Дата створення події

Сутність Reminders

Атрибут	Тип даних	Обмеження	Опис
reminder_id	UUID	Первинний ключ, NOT NULL	Унікальний ідентифікатор нагадування
event_id	UUID	Зовнішній ключ (CalendarEvents), NOT NULL	Подія, до якої прив'язане нагадування
reminder_time	TIMESTAMP	NOT NULL	Час нагадування
channel	ENUM	NOT NULL (push, email)	Канал сповіщення
status	BOOLEAN	NOT NULL, DEFAULT FALSE	Статус (відправлено/не відправлено)

Таблиця 2.5

Сутність Settings

Атрибут	Тип даних	Обмеження	Опис
setting_id	UUID	Первинний ключ, NOT NULL	Унікальний ідентифікатор налаштування
user_id	UUID	Зовнішній ключ (Users), NOT NULL	Користувач, якому належить налаштування
language	VARCHAR(10)	NOT NULL, DEFAULT 'uk'	Мова інтерфейсу (uk, en)
notifications	BOOLEAN	NOT NULL, DEFAULT TRUE	Увімкнення/вимкнення сповіщень
updated_at	TIMESTAMP	NOT NULL, DEFAULT NOW()	Дата оновлення налаштувань

Схема бази даних відображає зв'язки між сутностями, де таблиця Users є центральною, пов'язуючись із MedicalRecords, CalendarEvents і Settings через зовнішній ключ user_id. Таблиця Reminders пов'язана з CalendarEvents через зовнішній ключ event_id. Ця структура забезпечує цілісність даних і дозволяє ефективно виконувати запити для отримання інформації про користувачів, їхні медичні записи чи календарні події.

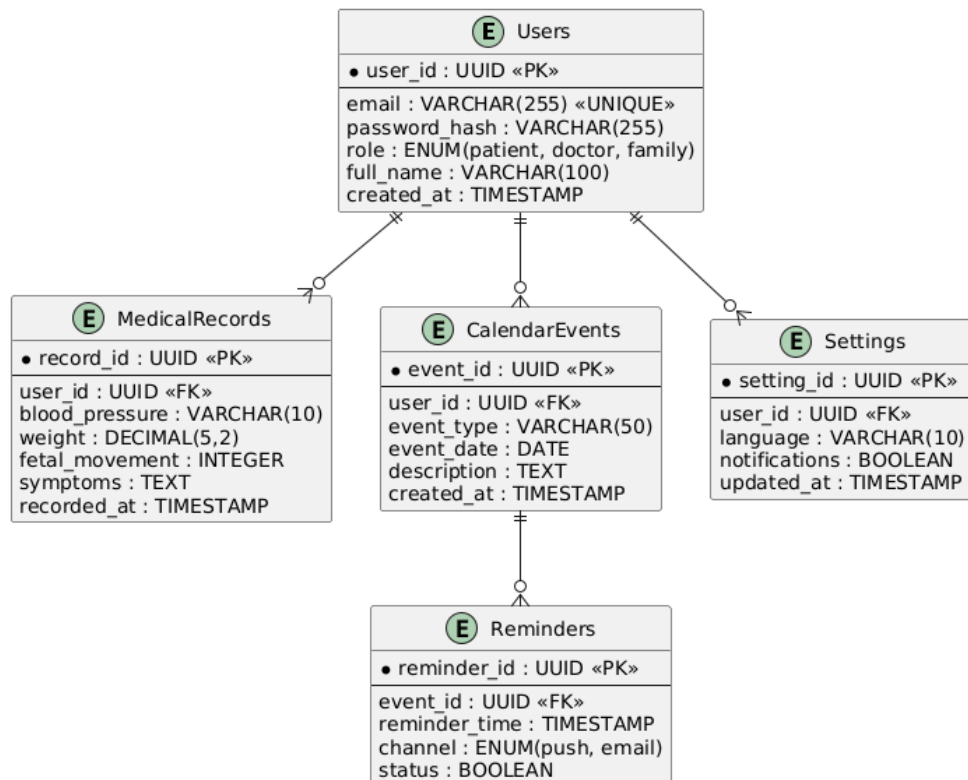


Рисунок 2.2 – Схема бази даних веб-застосунку

База даних зберігає медичну інформацію, тому її захист є обов'язковим. У SQLite для цього використовується шифрування на рівні окремих стовпців, зокрема password_hash у таблиці користувачів, із застосуванням алгоритму AES-256 [37]. Права доступу визначаються ролями користувачів: лікарі можуть переглядати записи пацієнток, а члени сім'ї мають обмеження й бачать лише події з календаря [8]. Для запобігання SQL-ін'єкціям застосовуються параметризовані запити. Контроль дій користувачів здійснюється за допомогою журналювання через pgAudit [26]. Дотримання вимог GDPR забезпечується анонімізацією даних і політиками з видалення після завершення їх використання [7].

Запроектowana база даних відповідає функціональним вимогам, дозволяючи зберігати та обробляти дані про користувачів, їхні медичні записи, календар і нагадування [9]. Реляційна модель забезпечує цілісність і стандартизацію, а зв'язки між таблицями дозволяють ефективно виконувати запити, наприклад, для відображення всіх подій користувача за певний період. Використання SQLite гарантує продуктивність і безпеку, а заходи шифрування та управління доступом відповідають стандартам захисту даних [28]. Схема бази даних чітко відображає структуру та зв'язки, що полегшує подальшу реалізацію та масштабування застосунку.

2.3 Розробка інтерфейсу користувача (UI/UX)

Створення інтерфейсу користувача для веб-застосунку, що призначений для контролю перебігу вагітності, передбачає увагу до зручності використання та логічної організації елементів. UI відповідає за зовнішній вигляд, а UX — за логіку навігації, простоту взаємодії та відповідність очікуванням користувачів. З урахуванням аналізу потреб, інтерфейс повинен бути адаптованим до мобільних і стаціонарних пристроїв, мати україномовну локалізацію й бути зрозумілим людям із різним рівнем досвіду роботи з цифровими технологіями. В основі проектування використовуються принципи, що включають простоту, передбачуваність, послідовність у діях і мінімальне навантаження на увагу [25].

Для жінок, які можуть перебувати в стані емоційного напруження або фізичного дискомфорту, важливо, щоб усі елементи були легко доступними: великі кнопки, зрозумілі позначення, мінімум зайвих дій. В оформленні використовуються м'які кольори, що сприяють спокійному візуальному сприйняттю [10]. Переклад інтерфейсу українською виконується з урахуванням офіційних термінів, правописних норм і типових для медичної сфери формулювань [3].

Адаптивний дизайн забезпечує коректне відображення на різних пристроях (смартфони, планшети, комп'ютери), що відповідає нефункціональній вимозі доступності [7]. Для цього використовується CSS, який дозволяє створювати

респонсивні компоненти без надмірного написання власного CSS [39]. Крім того, інтерфейс враховує принципи інклюзивності, такі як підтримка великих шрифтів для користувачів із вадами зору та можливість керування за допомогою клавіатури для осіб із обмеженими фізичними можливостями [25].

Інтерфейс веб-застосунку включає кілька основних сторінок і компонентів, які відповідають функціональним вимогам:

1. Головна сторінка – містить огляд стану вагітності (поточний тиждень, найближчі події), швидкий доступ до введення медичних даних і персоналізовані рекомендації. Ця сторінка є центральною для користувача і має мінімалістичний дизайн для швидкої орієнтації.

2. Календар вагітності – відображає заплановані візити до лікаря, УЗД, аналізи та дозволяє додавати нові події. Календар підтримує інтерактивні елементи, такі як фільтри за типом подій.

3. Модуль введення даних – форма для введення медичних показників (артеріальний тиск, вага, рухи плода) із валідацією введених даних для уникнення помилок.

4. Інформаційний розділ – містить статті, відео та FAQ, структуровані за триместрами вагітності, із пошуком за ключовими словами.

5. Панель лікаря – доступна для медичних працівників, дозволяє переглядати дані пацієнток, надсилати рекомендації та планувати консультації.

6. Розділ психологічної підтримки – включає медитації, дихальні вправи та форум для спілкування з іншими користувачами.

Для реалізації цих компонентів використовується PyQt6, який забезпечує модульність і повторне використання елементів, таких як кнопки, форми чи картки подій [28]. Навігація між сторінками здійснюється через бічне меню або верхню панель, що відповідає сучасним стандартам UX-дизайну [25].

Для проектування макетів інтерфейсу обрано Figma, оскільки цей інструмент підтримує спільну роботу, створення прототипів і експорт компонентів для розробників [34]. У Figma створено макети для всіх ключових сторінок, включаючи адаптивні версії для мобільних і десктопних пристроїв. Прототипи дозволяють тестувати навігацію та взаємодію, наприклад, перехід від

календаря до форми введення даних. Макети включають інтерактивні елементи, такі як кнопки для додавання подій чи збереження медичних даних, що полегшує подальшу розробку.

Для забезпечення зручності використання проведено аналіз цільової аудиторії, який показав, що вагітні жінки та члени їхніх сімей можуть мати різний рівень цифрової грамотності [3]. Тому інтерфейс спрощено: використано великі кнопки, чіткі підписи та підказки для полів введення. Наприклад, форма введення артеріального тиску містить форматований шаблон (наприклад, “120/80”) і валідацію для запобігання некоректним даним [8]. Для медичних працівників передбачено спрощений доступ до даних пацієнток через фільтри та сортування, що відповідає їхнім потребам у швидкому аналізі інформації [17].

Психологічна підтримка реалізована через окремий розділ із медитаціями та форумом, де користувачі можуть обмінюватися досвідом. Дизайн цього розділу використовує заспокійливі кольори та плавні анімації, щоб знизити тривожність [10]. Для тестування зручності інтерфейсу планується проведення юзабіліті-тестування з використанням методів, описаних у літературі, таких як опитування та спостереження за взаємодією користувачів із прототипом [25].

Для оцінки компонентів інтерфейсу складено таблицю, яка порівнює їх за функціональністю, цільовою аудиторією та технічними вимогами (табл. 2.6).

Запроектований інтерфейс охоплює основні функції застосунку, надаючи користувачам зручний доступ до календаря подій, форми введення медичних показників, довідкових матеріалів і розділів, пов’язаних із психологічною підтримкою [9]. Застосування PyQt6 у поєднанні з CSS дозволяє створювати адаптивні інтерфейси, що швидко оновлюються та коректно відображаються на різних пристроях. Принципи UX-дизайну, що покладені в основу структури, забезпечують логічну навігацію та простоту взаємодії для всіх груп користувачів [28]. Локалізація інтерфейсу українською та адаптація до умов використання людьми з різним рівнем доступу до цифрових сервісів гарантують інклюзивність і зручність у використанні [3].

Порівняння компонентів інтерфейсу

Компонент	Функціональність	Цільова аудиторія	Технічні вимоги
Головна сторінка	Огляд стану, швидкий доступ до функцій	Вагітні, сім'я	PyQt6, CSS, PyQt6 API
Календар вагітності	Відображення та додавання подій	Вагітні, лікарі	PyQt6
Модуль введення даних	Введення медичних показників	Вагітні, лікарі	PyQt6, валідація даних
Інформаційний розділ	Перевірений контент, пошук	Вагітні, сім'я	PyQt6, SQLite (контент)
Панель лікаря	Перегляд даних, рекомендації	Лікарі	PyQt6, API eHealth
Психологічна підтримка	Медитації, форум	Вагітні	PyQt6, Socket.IO (форум)

2.4 Вибір технологічного стеку для реалізації

Під час вибору технологій для розробки веб-застосунку, призначеного для контролю перебігу вагітності, враховуються технічні характеристики, вимоги до продуктивності, безпеки, адаптивності та інтеграції з зовнішніми сервісами. На основі попереднього аналізу визначено набір інструментів, який охоплює клієнтську частину, сервер, базу даних, захист інформації та засоби для тестування. Обране рішення має відповідати як функціональним вимогам (реалізація календаря, введення показників, доступ до інформаційних блоків і психологічної підтримки), так і нефункціональним параметрам, серед яких — стабільна робота при великому навантаженні, час завантаження інтерфейсу менш ніж дві секунди, локалізація українською мовою й підтримка роботи на різних

пристроях. Також система повинна забезпечувати можливість підключення до eHealth та інших зовнішніх платформ [17].

Клієнтська частина реалізується за допомогою PyQt6 — бібліотеки, яка дозволяє будувати інтерфейс за компонентним принципом. Це забезпечує зручність повторного використання елементів, таких як окремі блоки календаря чи форми введення даних, та прискорює їхнє оновлення завдяки використанню віртуального DOM [28]. Завдяки широкій екосистемі PyQt6 легко доповнюється додатковими бібліотеками: наприклад, PyQt6 використовується для налаштування маршрутизації, а Context API або Redux — для управління станом. Вибрані технології підтримують адаптивний дизайн, що дозволяє забезпечити коректне відображення інтерфейсу в усіх сучасних браузерах і на різних екранах [25].

Для стилізації застосовується CSS — утилітарний CSS-фреймворк, який дозволяє формувати інтерфейс за допомогою класів без необхідності створення великих окремих стилів. Це значно пришвидшує розробку, а також дозволяє точніше контролювати вигляд інтерфейсу з урахуванням вимог до локалізації, зокрема розміщення текстових елементів і коректного відображення українськомовного контенту [39].

Альтернативи, такі як Vue.js і Angular, розглядалися, але відхилені. Vue.js є легшим, але має меншу спільноту та екосистему порівняно з PyQt6, що може ускладнити інтеграцію з зовнішніми сервісами [34]. Angular пропонує повноцінний фреймворк, але його складність і більший обсяг коду роблять його менш доцільним для проекту середнього масштабу [36]. PyQt6 і CSS забезпечують баланс між продуктивністю, гнучкістю та швидкістю розробки.

Для серверної частини (backend) обрано Node.js із фреймворком Express. PyQt6 використовує асинхронну модель на основі Python, що дозволяє ефективно обробляти велику кількість одночасних запитів (до 10 000 користувачів, як визначено в нефункціональних вимогах) [28]. Express спрощує створення RESTful API для обробки запитів, таких як збереження медичних даних чи надсилання нагадувань, і підтримує інтеграцію з eHealth через захищені API [17]. Для реалізації асинхронних повідомлень (наприклад, push-повідомлень)

використовується Socket.IO, що забезпечує оновлення в реальному часі, наприклад, для форуму психологічної підтримки [34].

Альтернативи, такі як Django (Python) і Spring Boot (Java), менш доцільні. Django має вбудовані інструменти безпеки, але поступається Node.js у продуктивності для асинхронних операцій [34]. Spring Boot підходить для великих проєктів, але його ресурсомісткість і складність не виправдані для цього застосунку [36]. PyQt6 є оптимальним вибором через швидкість, простоту та сумісність із Python -екосистемою фронтенду.

Для зберігання структурованих даних, таких як профілі користувачів, медичні записи та календарні події, обрано реляційну СУБД SQLite. SQLite підтримує складні запити, транзакції та шифрування даних, що відповідає вимогам безпеки та цілісності для медичних даних [28]. Його здатність працювати з великими обсягами даних забезпечує масштабування, а вбудовані інструменти, такі як управління доступом, відповідають стандартам GDPR [7]. Для підвищення продуктивності використовується Redis як кеш для зберігання часто запитаних даних, наприклад, налаштувань користувача чи нагадувань, що зменшує час відповіді сервера до < 2 секунд [28].

MongoDB, як нереляційна СУБД, розглядалася, але відхилена через меншу ефективність для структурованих даних і складність забезпечення цілісності в порівнянні з PostgreSQL [34]. Комбінація PostgreSQL і Redis забезпечує баланс між надійністю, швидкістю та безпекою.

Безпека є пріоритетом, оскільки застосунок обробляє чутливу медичну інформацію. Для захисту зв'язку між клієнтом і сервером використовується HTTPS із сертифікатами SSL/TLS [37]. Автентифікація реалізована через JSON Web Tokens (JWT), які забезпечують безпечну передачу даних і підтримують двофакторну автентифікацію (2FA) за допомогою бібліотеки Speakeasy [37]. Для захисту від вразливостей, таких як SQL-ін'єкції чи XSS-атаки, застосовуються параметризовані запити в SQLite і бібліотека Helmet для Express [26]. Аудит безпеки проводиться з використанням OWASP ZAP, що дозволяє виявляти загрози, описані в OWASP Top Ten [26].

Для забезпечення якості застосунку обрано інструменти для різних типів тестування. Jest і PyQt6 Testing Library використовуються для модульного тестування клієнтської частини, перевіряючи компоненти PyQt6, такі як форми введення даних [28]. На серверній частині тестування API виконується за допомогою Mocha і Chai, що дозволяє перевірити коректність ендпоінтів [34]. Інтеграційне тестування проводиться з Postman для перевірки взаємодії між клієнтом і сервером [28]. Для тестування продуктивності та навантаження використовується Apache JMeter, що гарантує стабільність при великій кількості користувачів [36].

Для обґрунтування вибору складено таблицю, яка порівнює обрані технології за ключовими критеріями (табл. 2.7).

Обраний технологічний стек, до якого входять Python, PyQt6, CSS, PyQt6, Express, SQLite, Redis, HTTPS, JWT, Jest і Mocha, охоплює всі технічні потреби проєкту. Використання PyQt6у поєднанні з CSS дозволяє створити адаптивний інтерфейс, що підтримує локалізацію та зручний для користувачів на різних пристроях. Серверна частина, реалізована на Node.js з використанням Express, забезпечує швидку обробку запитів і дає змогу реалізувати інтеграцію з зовнішніми сервісами, зокрема eHealth [17]. Для збереження даних застосовується реляційна база SQLite, яка відповідає вимогам до структурованого зберігання медичної інформації, а Redis використовується як кеш для забезпечення швидкого доступу до часто запитуваних даних.

Захист інформації реалізується через HTTPS, токени JWT та бібліотеку Helmet, що дозволяє дотримуватись вимог регламенту GDPR та Закону України «Про захист персональних даних» [7]. Якість реалізації контролюється за допомогою інструментів для тестування — Jest і Mocha для перевірки логіки застосунку, а також Postman для перевірки взаємодії між клієнтською і серверною частинами [28]. Обраний стек відповідає сучасним стандартам розробки, має активну підтримку спільноти й забезпечує надійні умови для реалізації веб-застосунку, що виконує функції моніторингу перебігу вагітності.

Порівняння компонентів технологічного стеку

Компонент	Технологія	Продуктивність	Безпека	Легкість використання	Підтримка спільноти
Frontend	PyQt6 + CSS	Висока	Висока	Середня	Висока
Backend	Node.js + Express	Висока	Висока	Висока	Висока
База даних	SQLite + Redis	Висока	Висока	Середня	Висока
Безпека	HTTPS, JWT, Helmet	Висока	Висока	Висока	Висока
Тестування	Jest, Mocha, Postman	Висока	Висока	Висока	Висока

2.5 Планування заходів із забезпечення безпеки даних

Захист даних у веб-застосунку для контролю перебігу вагітності охоплює весь цикл обробки інформації — від моменту введення користувачем до збереження в базі. Оскільки система працює з персональними та медичними записами, важливо забезпечити безпечну передачу, зберігання та контроль доступу згідно з вимогами GDPR і Закону України «Про захист персональних даних» [7]. Реалізація заходів безпеки охоплює клієнтську частину, серверну логіку, базу даних, а також включає аудит і тестування вразливостей.

У клієнтській частині використовується протокол HTTPS із сертифікатами SSL/TLS, що гарантує зашифровану передачу даних. Реалізація інтерфейсу на базі PyQt6 передбачає використання бібліотеки Helmet, яка встановлює захисні HTTP-заголовки, зокрема Content-Security-Policy для запобігання XSS-атакам [26], [37]. Аутентифікація здійснюється через JWT, які містять зашифровану інформацію про користувача, включно з ID та роллю. Це дозволяє визначати права доступу під час кожного запиту [28]. Для користувачів, які мають розширені

повноваження (зокрема лікарів), додатково впроваджена двофакторна автентифікація через бібліотеку Speakeasy [37].

Серверна частина, побудована на Node.js з Express, реалізує захист від основних типів атак, описаних у списку OWASP Top Ten, зокрема SQL-ін'єкцій і CSRF. Для запитів до бази даних використовуються параметризовані конструкції, які виключають виконання стороннього коду [28]. Захист від CSRF реалізується через перевірку унікальних токенів. Доступ до функціональності обмежується згідно з моделлю RBAC: користувачі мають ролі (patient, doctor, family), кожна з яких надає доступ до відповідного підмножини функцій [8].

У PostgreSQL чутливі поля, наприклад паролі чи медичні показники, шифруються з використанням алгоритму AES-256. Шифрування здійснюється на рівні стовпців. Доступ до бази регламентується правами PostgreSQL: клієнтська частина має обмежений доступ до окремих запитів, тоді як адміністратор володіє правами на резервне копіювання й аудит [28]. Для відповідності нормам GDPR застосовується анонімізація даних, а також реалізовано функцію видалення профілю за запитом користувача [7].

Redis використовується як проміжне сховище для некритичних даних, зокрема для налаштувань інтерфейсу чи повторюваних нагадувань. Доступ до Redis захищений паролем і обмежений у конфігурації [28].

Для виявлення загроз безпеки застосовуються інструменти тестування. OWASP ZAP використовується для автоматизованого сканування потенційних вразливостей у застосунку, зокрема перевірки конфігурації API [26]. Крім того, заплановане ручне тестування (penetration testing) з використанням рекомендацій OWASP Testing Guide, що охоплює сценарії brute force, session hijacking, маніпуляції з JWT. Події реєструються через Winston — систему журналювання для Node.js, яка фіксує критичні дії, включаючи спроби несанкціонованого доступу [34].

Заходи безпеки охоплюють як механізми захисту, так і політику реагування: при виявленні загроз дані ізолюються, доступ тимчасово обмежується, а користувачі інформуються про інцидент. Уся реалізована інфраструктура відповідає нефункціональним вимогам, окресленим раніше [9], включаючи

безпечний доступ (HTTPS, JWT, 2FA), захищене зберігання (AES-256 у PostgreSQL), керування ролями (RBAC) та постійний моніторинг (логування, аудит, ZAP) [7], [8], [26]. Код реалізації системи безпеки наведено в додатку А.

Висновки до розділу 2

У другому розділі виконано проектування веб-застосунку для контролю перебігу вагітності, що охоплює архітектуру, базу даних, інтерфейс користувача, технологічний стек і заходи безпеки. Розроблена клієнт-серверна архітектура з RESTful API та мікросервісним підходом забезпечує масштабування, інтеграцію з eHealth і Google Calendar, а також швидку обробку запитів завдяки використанню Node.js і Express [17]. Схема архітектури чітко визначає взаємодію компонентів, що полегшує подальшу реалізацію.

База даних, спроектована на основі SQLite, включає п'ять сутностей (Users, MedicalRecords, CalendarEvents, Reminders, Settings), які підтримують зберігання медичних даних, календарних подій і налаштувань із забезпеченням безпеки через шифрування AES-256 і управління доступом [28]. Використання Redis для кешування підвищує продуктивність, відповідаючи вимозі до часу завантаження < 2 секунд [28].

Інтерфейс користувача (UI/UX), розроблений із використанням PyQt6 і CSS, є адаптивним, локалізованим українською мовою та відповідає принципам інтуїтивності й інклюзивності, що забезпечує зручність для вагітних жінок, лікарів і членів сім'ї [25]. Макети, створені в Figma, охоплюють ключові сторінки, такі як календар і модуль введення даних, що дозволяє протестувати дизайн перед розробкою.

Технологічний стек (PyQt6, SQLite, Redis, HTTPS, JWT) обрано за критеріями продуктивності, безпеки та підтримки спільноти, що гарантує ефективну реалізацію та відповідність функціональним і нефункціональним вимогам [28]. Заходи безпеки, включаючи HTTPS, двофакторну автентифікацію, шифрування та аудит із використанням OWASP ZAP, забезпечують захист

чутливих медичних даних і відповідність GDPR і Закону України «Про захист персональних даних» [7].

Спроектований веб-застосунок має чітку архітектуру, надійну базу даних, зручний інтерфейс і безпечну інфраструктуру, що створює міцну основу для його реалізації та подальшого тестування [9].

РОЗДІЛ 3

РОЗРОБКА ПРОГРАМНОГО ЗАСТОСУНКУ «ЕЛЕКТРОННИЙ ЩОДЕННИК ВЕДЕННЯ ВАГІТНОСТІ»

3.1 Розробка серверної частини застосунку

Серверна частина веб-застосунку для контролю вагітності обробляє запити, керує даними користувачів і забезпечує взаємодію з клієнтським інтерфейсом. Її побудовано на основі існуючої віконної версії, реалізованої на Python, що дозволило зберегти логіку та полегшити перехід до веб-формату. Основу складає Python із фреймворком Flask, який використовується для створення RESTful API. Запити охоплюють введення медичних записів, роботу з календарем і надсилання нагадувань [28].

Flask обрано завдяки гнучкості та сумісності з бібліотеками для роботи з БД, шифруванням і багатопоточністю. Для доступу до БД використовується SQLAlchemy як ORM. Наразі застосовується SQLite для тестування, однак структура дозволяє перехід на PostgreSQL. Це відповідає підходу, реалізованому у віконній версії [28].

Для збереження конфіденційних даних використовується шифрування на основі cryptography з алгоритмом AES-128. Багатопоточність дає змогу регулярно перевіряти таблицю нагадувань без блокування основного потоку, забезпечуючи надсилання повідомлень у визначений час. Обрана архітектура проста, функціональна та сумісна з уже реалізованими компонентами [9], [34].

Серверна частина має чітку модульну структуру:

1. модуль API: реалізує ендпоінти для автентифікації, управління подіями, медичними даними, інформаційним контентом і психологічною підтримкою
2. модуль бази даних: взаємодіє з SQLite через SQLAlchemy, керує таблицями Users, MedicalRecords, CalendarEvents, Reminders і Settings [28]
3. модуль безпеки: забезпечує автентифікацію через JWT, шифрування і захист від SQL-ін'єкцій та XSS [26]

4. модуль нагадувань: реалізує асинхронну перевірку календаря для надсилання сповіщень

Основний файл (`app.py`) ініціалізує Flask, налаштовує SQLAlchemy, визначає ендпоінти та запускає фоновий потік для нагадувань. Структура відповідає клієнт-серверній архітектурі й дозволяє інтегрувати зовнішні сервіси, зокрема eHealth [17].

Ініціалізація виконується через створення Flask-додатку та підключення SQLite, структура якого відповідає віконній версії. SQLAlchemy формує моделі таблиць Users, MedicalRecords, CalendarEvents, Reminders і Settings [28]. Перехід до PostgreSQL можливий без зміни структури [28].

API охоплює функціональні ендпоінти:

- POST `/api/login`: автентифікація користувача з JWT
- POST `/api/events`: створення події з нагадуванням
- GET `/api/events`: отримання списку подій
- POST `/api/medical-records`: збереження медичних показників
- GET `/api/medical-records`: перегляд медичних записів
- GET `/api/info`: доступ до інформаційних статей
- GET `/api/psych-tips`: поради з психологічної підтримки

Для захисту даних ендпоінт перевіряє JWT, валідує тиск, шифрує симптоми і зберігає дані [37].

Безпека реалізована через:

- JWT: генерується при вході, містить `user_id` і роль (`patient`, `doctor`, `family`), діє обмежений час [28]
- Шифрування: використання AES-128 для захисту паролів і симптомів [37]
- Захист: параметризовані запити SQLAlchemy та Flask-Talisman для встановлення політик CSP, XSS, CSRF [26]
- Контроль доступу: RBAC визначає рівень доступу залежно від ролі [8]
- GDPR: користувач може видалити профіль через `DELETE /api/user` [7]

Асинхронний потік, реалізований через `threading`, перевіряє таблицю Reminders кожні 60 секунд. У тестовій версії повідомлення виводяться в консоль, у продакшн — передбачено WebSocket через Flask-SocketIO [34].

Сервер сумісний із віконною версією: структуру таблиць, логіку шифрування й автентифікацію адаптовано під веб-формат із JWT. REST API дозволяє взаємодію як із Tkinter-додатком, так і з майбутньою веб-клієнтською частиною, що забезпечує гнучкість.

Використання Flask у зв'язці з Python забезпечує швидку розробку функціоналу при збереженні сумісності з настільною версією, відповідно до технічних вимог [28]. ORM SQLAlchemy спрощує роботу з БД і підтримує визначену структуру даних [28]. Безпеку забезпечують cryptography та Flask-Talisman, що відповідає вимогам GDPR і національного законодавства [7]. Для локального середовища застосовується threading, а для масштабування — WebSocket [34]. API спроектовано з урахуванням інтеграції з eHealth [17]. Інтерфейс розраховано на користувачів із різним рівнем цифрової грамотності, забезпечуючи доступність [3] (табл. 3.1).

Таблиця 3.1

Основні ендпоінти серверної частини

Ендпоінт	Метод	Опис	Безпека
/api/login	POST	Автентифікація користувача	Шифрування пароля, JWT
/api/events	POST	Додавання календарної події	JWT
/api/events	GET	Отримання подій користувача	JWT
/api/medical-records	POST	Збереження медичних даних	JWT, шифрування симптомів
/api/medical-records	GET	Отримання медичних записів	JWT
/api/info	GET	Отримання інформаційних статей	Без автентифікації
/api/psych-tips	GET	Отримання психологічних порад	Без автентифікації

Серверна частина успішно реалізована, забезпечуючи обробку запитів, управління даними, безпеку та сумісність із віконною версією застосунку. Вона створює міцну основу для розробки клієнтської частини та подальшого тестування, що буде розглянуто в наступних підрозділах.

3.2 Реалізація клієнтської частини та інтерфейсу

Інтерфейс користувача побудований на компонентній архітектурі з використанням бібліотеки PyQt6. Всі візуальні елементи стилізовано за допомогою CSS, що забезпечує високу швидкість розробки, консистентність дизайну та адаптивність на різних пристроях. Інтерфейс призначений для користувачів з різним рівнем цифрової підготовки, тому його структура мінімалістична, логічна та візуально доступна.

На першому етапі взаємодії з веб-застосунком користувач потрапляє на екран авторизації, де необхідно ввести електронну пошту та пароль. Кнопка входу чітко виділена кольором, що забезпечує зоровий акцент. Також додано посилання на форму реєстрації, що дозволяє швидко перейти до створення нового облікового запису. Інтерфейс реалізовано з урахуванням основних вимог доступності: високий контраст, зрозумілі написи, передбачувана поведінка елементів.

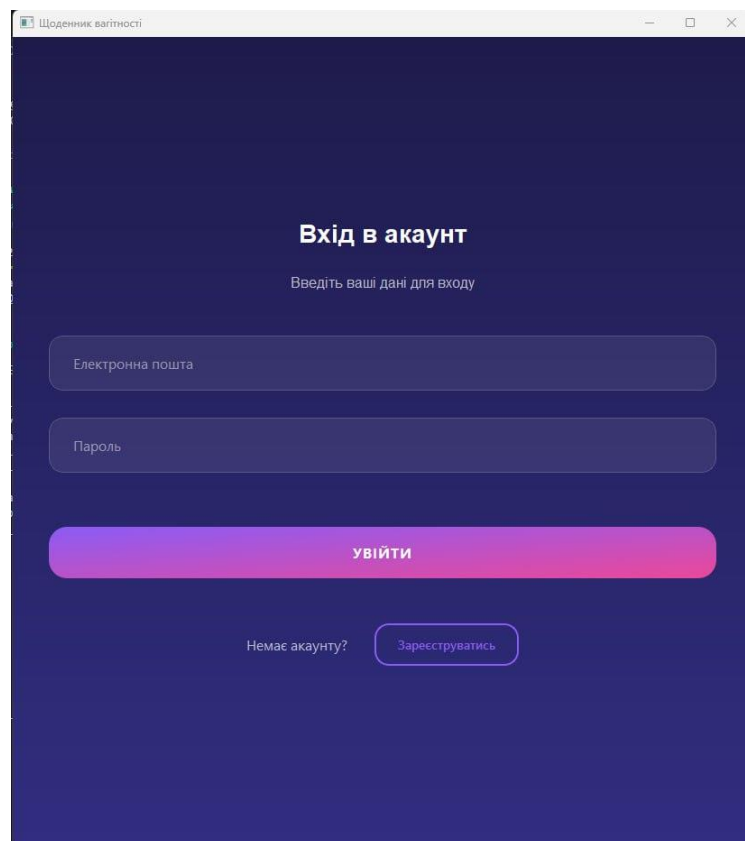


Рисунок 3.1 – Форма входу користувача з полями електронної пошти та пароля

Форма реєстрації реалізована як окремий екран з чотирма полями: електронна пошта, ім'я, пароль і підтвердження пароля. Усі поля розміщені у вертикальному порядку, з чіткими візуальними підказками. Кнопка «ЗАРЕЄСТРУВАТИСЬ» оформлена в межах загальної палітри інтерфейсу, а під нею передбачено навігаційне посилання на форму входу для тих, хто вже має акаунт. Вся логіка побудови сторінки спрямована на полегшення взаємодії з системою та мінімізацію кількості дій.

Останнім елементом цього блоку є форма входу до облікового запису користувача. На екрані реалізовано поля для введення електронної пошти та пароля. У випадку некоректного введення даних з'являється спливаюче повідомлення з попередженням про помилку — «Невірна пошта або пароль». Візуально це оформлено у вигляді діалогового вікна з жовтим знаком попередження та кнопкою «ОК» для закриття сповіщення. Такий підхід дозволяє оперативно інформувати користувача про неправильні дії та не допускає помилкової авторизації.

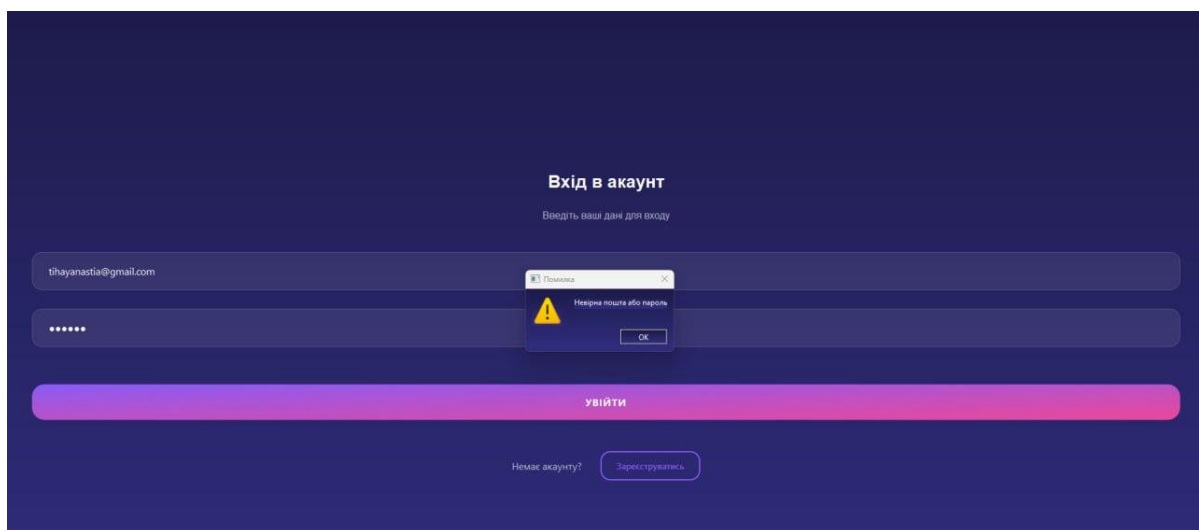
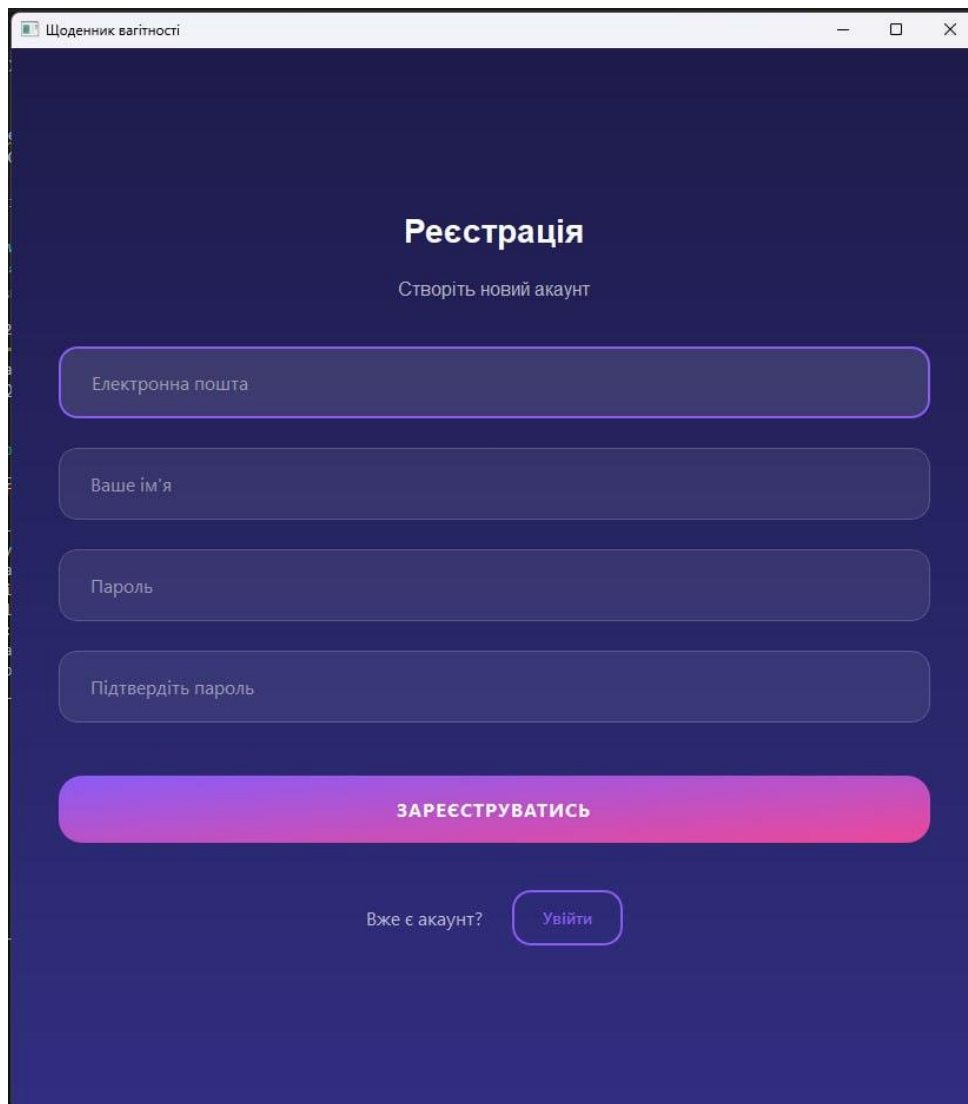


Рисунок 3.2 – Повідомлення про помилку входу при некоректних даних



Щоденник вагітності

Реєстрація

Створіть новий акаунт

Електронна пошта

Ваше ім'я

Пароль

Підтвердіть пароль

ЗАРЕЄСТРУВАТИСЬ

Вже є акаунт? [Увійти](#)

Рисунок 3.3 – Сторінка створення нового облікового запису з формою заповнення

Після реєстрації система пропонує користувачеві підтвердити свою пошту. На відповідному екрані реалізовано інструкцію щодо перевірки поштової скриньки та введення шестизначного коду. Застосунок забезпечує інформативний текст, поле для введення коду, кнопку підтвердження, а також можливість повторної відправки листа. Навігаційна кнопка «Назад» дозволяє повернутися на попередній етап, не втрачаючи прогресу.

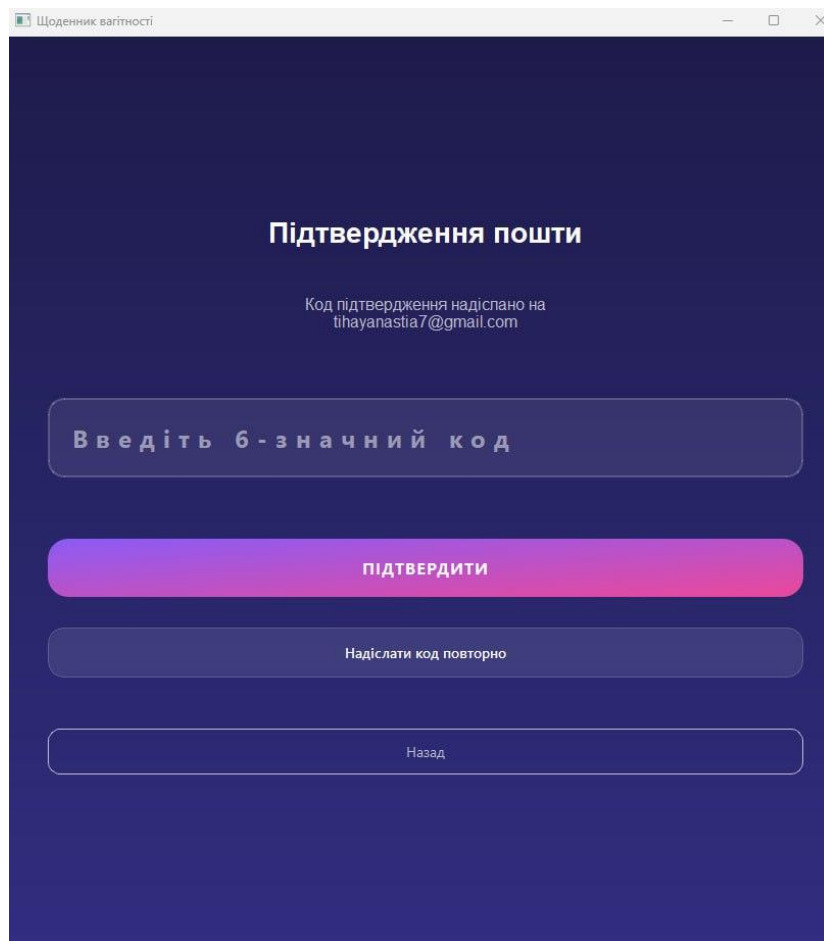


Рисунок 3.4 – Екран підтвердження електронної пошти через введення 6-значного коду

Необхідно ввести дані про дату останньої менструації, очікувану дату пологів та, за наявності, дату зачаття — ці параметри використовуються для розрахунку терміну вагітності та налаштування календаря тижнів у застосунку, у випадку коли вводиться некоректна дата, з'являється відповідне повідомлення. У відповідній формі потрібно зазначити дату останньої менструації, очікувану дату пологів та дату зачаття (за наявності). Усі поля підтримують автозаповнення або ручне введення. Інформація використовується для обчислення терміну вагітності та налаштування календаря тижнів. Кнопка «ПРОДОВЖИТИ» активує подальший етап.

Щоденник вагітності

Інформація про вагітність

Вкажіть дати для розрахунку терміну вагітності

Дата останньої менструації:

13.05.2024

Перший день останнього менструального циклу

Очікувана дата пологів (розраховується автоматично):

17.02.2025

Дата зачаття (якщо відома):

27.05.2024

Зазвичай відбувається через 14 днів після початку менструації

ПРОДОВЖИТИ

Рисунок 3.5 – Форма введення основних медичних даних для розрахунку терміну вагітності

Після введення медичної інформації користувачка потрапляє до персоналізованого календаря вагітності. Кожному тижню відповідає окремий екран із візуальною аналогією (у цьому прикладі — тиждень 23, дитина порівнюється з помідором), вагою та зростом плода. Нижче наведено короткий опис змін у розвитку дитини, а також інформація про можливі фізіологічні зміни в організмі матері. Таким чином, календар забезпечує підтримку користувача у вигляді щотижневого супроводу.

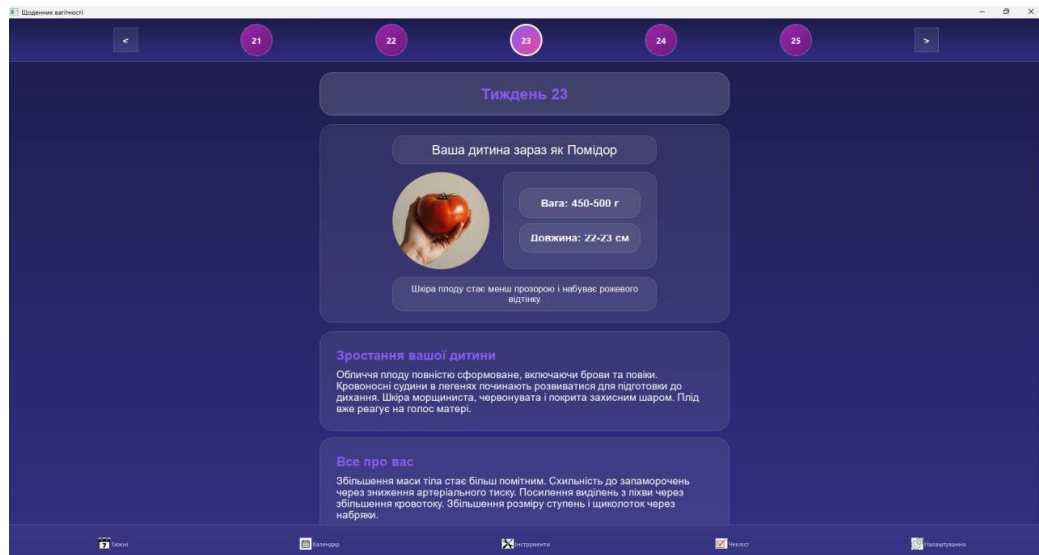


Рисунок 3.6 – Екран календаря розвитку плода: тиждень 23

У наступному прикладі представлено вигляд тижня 33 у динамічному календарі вагітності. На цьому етапі плід порівнюється з ананасом. У верхньому блоці подано актуальні показники: вага – 1.8–2.0 кг, довжина – 42.5–43.5 см. Під цим розміщено короткий підпис щодо активного набору ваги. В секції нижче надано опис фізіологічного розвитку плода, зокрема дозрівання мозку й легенів, тренування дихальних рухів і накопичення меконію. Блок "Все про вас" інформує про характерні для цього періоду симптоми – втому, тиск, задишку й скорочення Брекстона-Гікса.

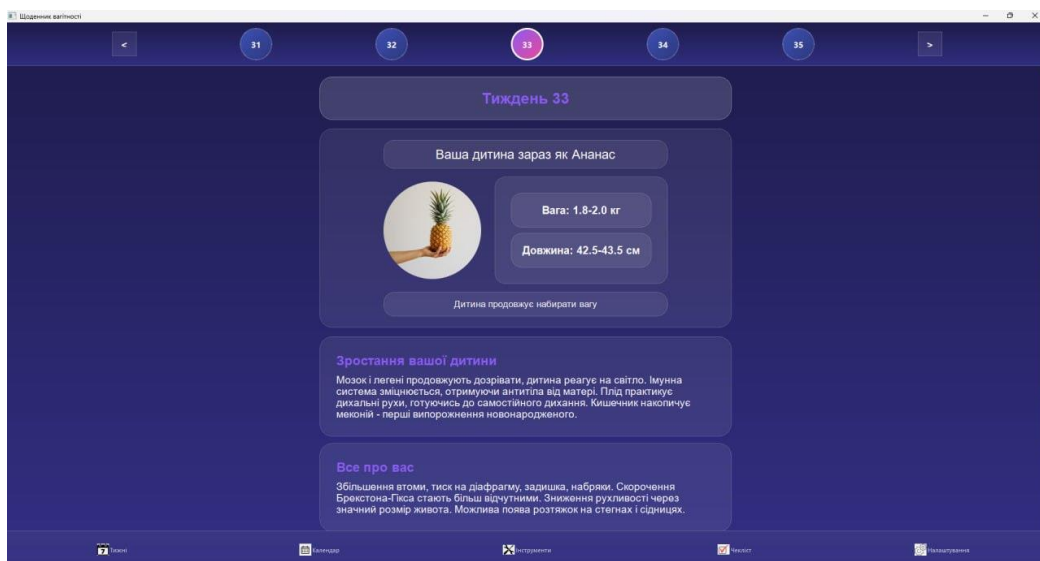


Рисунок 3.7 – Екран календаря вагітності: тиждень 33 із порівнянням плода з ананасом

Інша ілюстрація демонструє тиждень 25, де дитина порівнюється з цвітною капустою. Система автоматично виводить приблизну вагу (650–700 г) та довжину (33–35 см). У текстовому описі розвитку дитини наголошено на формуванні дихальної системи та активному накопиченні жиру. Інформація про фізіологічні зміни в тілі жінки містить відомості про болі в тазу, геморої і розширення вен, що супроводжують даний етап вагітності.

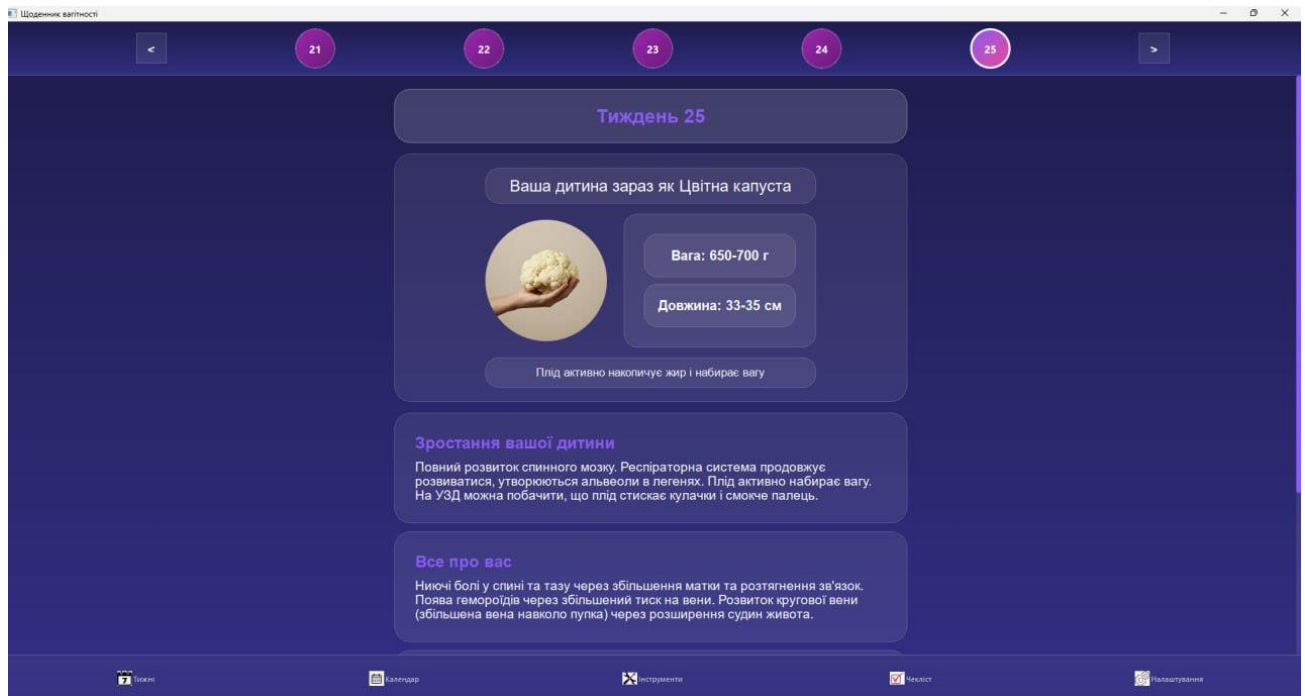
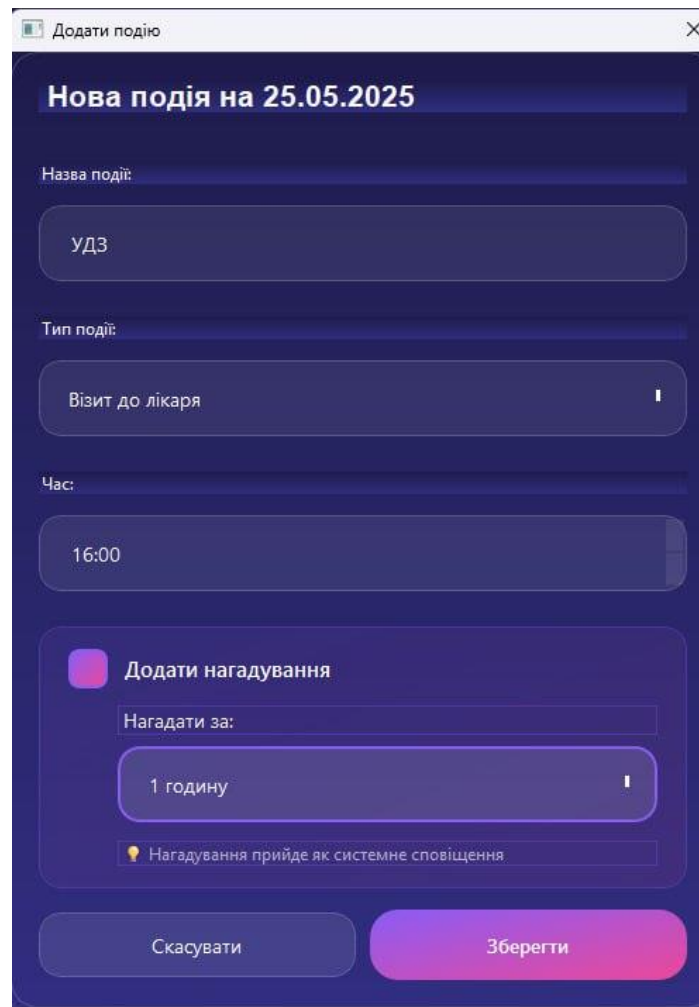


Рисунок 3.8 – Екран тижня 25 у календарі вагітності з інформацією про зміни плода та самопочуття жінки

Крім календаря, застосунок має функцію додавання подій до особистого розкладу. Форма створення нової події включає назву (наприклад, «УЗД»), тип події («Візит до лікаря»), час початку та інструмент нагадування. Можна налаштувати нагадування заздалегідь, вибравши інтервал (наприклад, «за 1 годину»). Це дає змогу організувати відвідування лікарів, процедур або інших важливих моментів, що стосуються вагітності. Інтерфейс форми мінімалістичний, функціональний і відповідає загальному дизайну застосунку.



Додати подію

Нова подія на 25.05.2025

Назва події:

УДЗ

Тип події:

Візит до лікаря

Час:

16:00

Додати нагадування

Нагадати за:

1 годину

Нагадування прийде як системне сповіщення

Скасувати Зберегти

Рисунок 3.9 – Форма додавання події з налаштуванням часу і нагадуванням

Наступним елементом є окрема сторінка з інформаційним блоком, що містить корисні поради для вагітних жінок. Вміст зосереджений на користі вправ Кегеля — спеціальних фізичних вправ для зміцнення м'язів тазового дна. У тексті акцент зроблено на перевагах регулярного виконання цих вправ, зокрема на покращенні контролю над сечовим міхуром, підготовці до пологів та прискоренні відновлення після них. У нижній частині екрана розташовано кнопку «Відкрити інструкцію з вправами», оформлену у рожево-фіолетовому стилі, що надає доступ до PDF-документа з розширеними рекомендаціями.

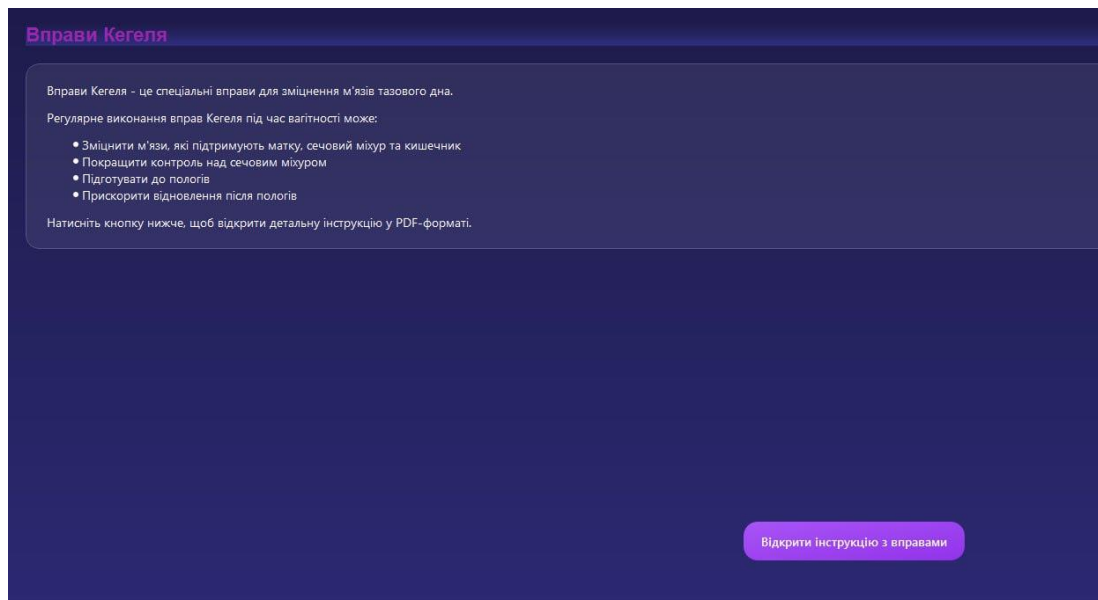


Рисунок 3.10 – Екран інформації про користь вправ Кегеля

Далі представлено екран контрольного списку (чеклісту), який дає змогу відстежувати проходження рекомендованих медичних аналізів та обстежень у розрізі триместрів вагітності. Інтерфейс організовано у вигляді вкладок для кожного триместру (I, II, III), з можливістю перемикання між ними. У прикладі показано I триместр: тут наведено список аналізів з індикаторами прогресу виконання у вигляді заповненої шкали та чекбоксів, які користувач може активувати вручну після здачі відповідного аналізу. Завдяки структурованому відображенню елементів, система забезпечує зручний контроль за станом проходження обстежень.

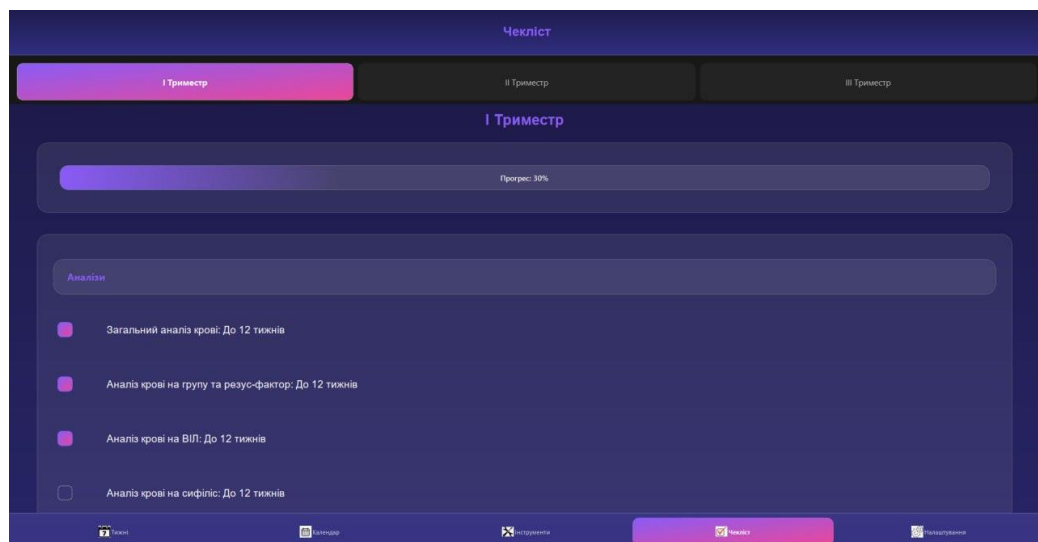


Рисунок 3.11 – Інтерфейс чеклісту медичних аналізів для I триместру

У межах функціонального модуля, присвяченого вимірюванню змін тілесних параметрів, реалізовано інтерфейс для відстеження обхвату живота протягом вагітності. Користувач має змогу ввести дату вимірювання, фактичне значення розміру живота в сантиметрах і за потреби додати коментар. Форма побудована у вигляді двох розділів: зліва — блок для внесення нового запису, справа — блок історії вимірювань, що оновлюється після кожного збереження. Кнопка «Зберегти запис» виділена контрастним помаранчевим кольором, що полегшує навігацію. Такий підхід забезпечує зручність регулярного контролю за динамікою змін у тілі вагітної жінки з інтервалом у 2 – 4 тижні.

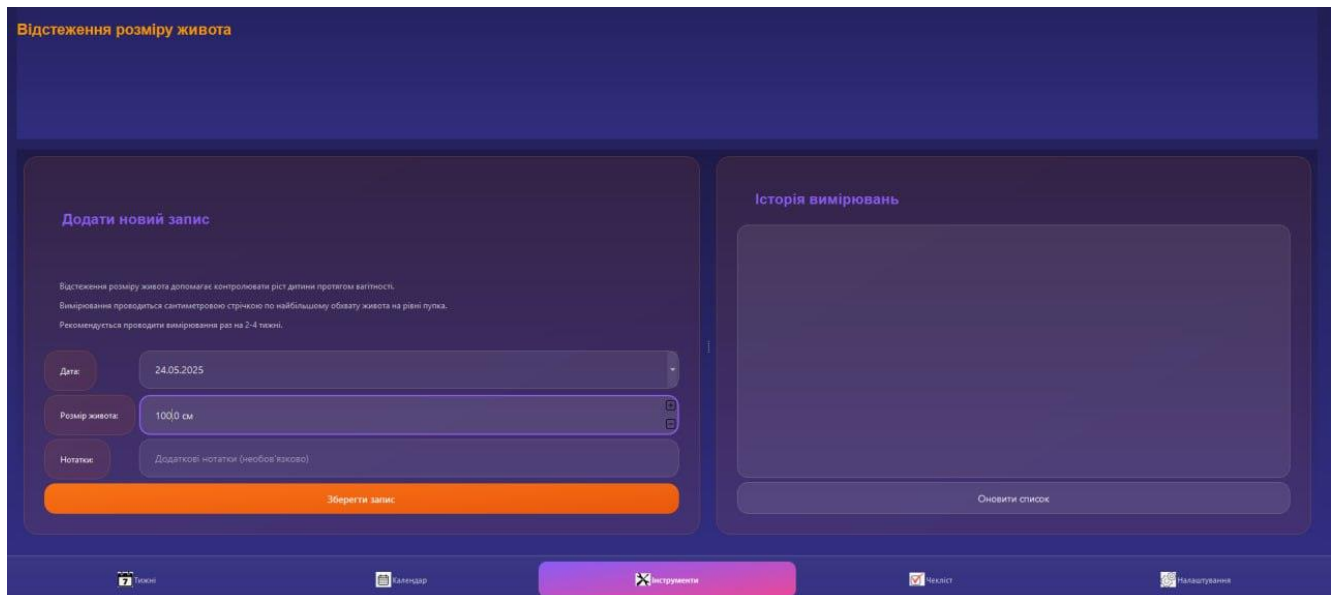


Рисунок 3.12 — Відстеження розміру живота

Система моніторингу побажань реалізована через форму, яка дає змогу зберігати персоналізовані записи щодо необхідних речей або препаратів. Інтерфейс містить поля для введення назви, опису, категорії, вартості та пріоритету бажаного товару. Також передбачено можливість позначити елемент як «вже придбано». Введення значень здійснюється вручну, а збереження нового елемента ініціюється кнопкою «Додати в список», розміщеною внизу. Весь функціонал організовано у межах однієї картки з чітко структурованими полями, що сприяє формуванню особистого контрольного списку важливих покупок.

Додати нове бажання

Назва: Вітаміни

Опис: Solgar

Категорія: Інше

Ціна: 980|грн

Пріоритет: Високий

Вже придбано

Додати в список

Рисунок 3.13 — Додавання нового побажання

3.3 Інтеграція з зовнішніми сервісами (наприклад, календарі, нагадування)

Інтеграція із зовнішніми сервісами у віконному застосунку на Python з використанням Tkinter реалізована для синхронізації подій, надсилання нагадувань та підготовки до підключення медичних платформ (табл. 3.3). Застосунок підтримує роботу з Google Calendar: через API Google і автентифікацію OAuth 2.0 події додаються безпосередньо до календаря користувача. Це дає змогу встановлювати нагадування й синхронізувати важливі події.

Нагадування реалізовані локально: після збереження події користувач отримує сповіщення у застосунку або через e-mail за наявності мережевого підключення. Формат повідомлень адаптовано до українських стандартів часу, мови й дат. Для взаємодії використовуються бібліотеки google-api-python-client і tkinter.messagebox, що дозволяє поєднати зовнішні інструменти з інтерфейсом без складних залежностей.

Система нагадувань надсилає листи на електронну пошту із зазначенням назви події, дати та часу. Повідомлення структуровані — містять ім'я користувача, інформацію про подію, пояснення її важливості й уточнення, що відповідь не потрібна. Це підвищує регулярність виконання медичних рекомендацій.

Інтеграція з eHealth поки не реалізована, але структура застосунку передбачає обмін даними через REST API з передачею зашифрованих медичних показників. Синхронізація з календарем і нагадування вже відповідають основним потребам користувача.

Оскільки застосунок є локальним, інтеграція обмежена: прямий мережевий доступ відсутній. Щоб забезпечити відповідність функціоналу мобільних застосунків, реалізовано локальні аналоги зовнішніх сервісів. Підготовлено основу для мережевої взаємодії у майбутній веб-версії.

Основні напрямки інтеграції:

1. google calendar: синхронізація подій (візити, УЗД) через експорт у формат ICS, сумісний із календарними сервісами [30]
2. нагадування: локальні сповіщення у вигляді спливаючих вікон через Tkinter, які у веб-версії будуть реалізовані через WebSocket [34]
3. eHealth: у віконній версії — експорт медичних даних у CSV; у веб-версії — запланована інтеграція через API [17]

Синхронізація з Google Calendar у віконній версії здійснюється шляхом експорту ICS-файлу `calendar_events.ics`, що імпортується вручну до календаря. Події отримуються з таблиці `CalendarEvents`, формуються у формат ICS і зберігаються у файл, який підтримується Google Calendar. У веб-версії буде реалізовано пряму інтеграцію через Google API з автентифікацією через OAuth 2.0 [17].

Нагадування реалізовано за допомогою `threading`: система щохвилини перевіряє таблицю `Reminders` і при збігу часу відкриває вікно з повідомленням через `messagebox`. У веб-версії це буде реалізовано через `Flask-SocketIO` для надсилання `push`-повідомлень [34]. Поточний підхід відповідає критеріям простоти та ефективності в умовах локального середовища [25].

Інтеграція з eHealth у віконному середовищі обмежена: замість прямого API реалізовано експорт даних у CSV-файл. Медичні параметри (тиск, вага, симптоми) витягуються з бази, дешифруються (cryptography) і зберігаються для передачі лікарю [17], [37]. У веб-версії передбачено використання захищеного API з авторизацією через OAuth.

Інтеграція відповідає вимогам безпеки [7]:

- шифрування: симптоми та паролі захищаються алгоритмом AES-128 через бібліотеку cryptography [37]
- захист від вразливостей: використання параметризованих запитів у SQLAlchemy запобігає SQL-ін'єкціям [26]
- відповідність GDPR: експортовані ICS і CSV не містять відкритих чутливих даних, передбачена можливість видалення облікового запису [7]

Інтерфейс повністю локалізовано: текстові мітки, повідомлення та формати дат відповідають стандарту uk_UA. Віджет tkcalendar використовує українську локалізацію, забезпечуючи комфорт для користувача [3], [9].

З огляду на локальну архітектуру, прямі інтеграції з Google Calendar та eHealth замінено експортом у ICS і CSV [30]. Поточна реалізація нагадувань ефективна, але менш гнучка порівняно з веб-рішеннями. У веб-версії заплановано:

- використання Google Calendar API для синхронізації
- повноцінну інтеграцію з eHealth через захищене API
- push-повідомлення через WebSocket [17], [34]

Таблиця 3.3

Інтеграція з зовнішніми сервісами

Сервіс	Реалізація у віконній версії	Технології	Перспективи для веб-версії
Google Calendar	Експорт подій у ICS-файл	Python, sqlite3	Google Calendar API, OAuth [30]
Нагадування	Локальні сповіщення через Tkinter	threading, Tkinter messagebox	WebSocket, Flask-SocketIO [34]
eHealth	Експорт даних у CSV	Python, csv, cryptography	API eHealth, OAuth [17]

Інтеграція з зовнішніми сервісами у віконній версії відповідає функціональним вимогам, забезпечуючи синхронізацію подій і нагадування через локальні аналоги (ICS, спливаючі вікна, CSV) [9]. Використання Python, Tkinter і tkcalendar гарантує сумісність із серверною частиною та простоту реалізації [28]. Заходи безпеки (шифрування, захист від SQL-ін'єкцій) відповідають GDPR і Закону України [7, 26]. Локалізація українською та зручний віджет datericker забезпечують доступність і зручність для користувачів [3, 25]. Реалізація створює основу для переходу до веб-версії з повноцінною мережевою інтеграцією, що буде розглянуто в подальших етапах.



Рисунок 3.14 — Email-нагадування про подію

3.4 Захист персональної інформації та даних

Захист даних у застосунку для контролю перебігу вагітності реалізовано з урахуванням вимог конфіденційності, цілісності й доступності. Застосунок зберігає персональні й медичні дані, тому впроваджено технічні та нормативні заходи безпеки відповідно до GDPR, Закону України «Про захист персональних даних», а також рекомендацій OWASP і HIPAA. Обробка чутливої інформації

здійснюється з використанням шифрування, обмеженням доступу та можливістю видалення даних.

Усі медичні показники та особисті дані шифруються за допомогою бібліотеки `cryptography` з алгоритмом AES-128. Ключ зберігається окремо у файлі, доступ до якого обмежено. Паролі — у хешованому вигляді. Ролі користувачів (`patient`, `doctor`, `family`) визначають рівень доступу: лікар бачить дані лише своїх пацієнток, члени сім'ї — лише календар. Авторизація реалізована через логін і пароль, а всі запити до SQLite виконуються параметризовано. Право на видалення профілю реалізоване — при видаленні облікового запису очищуються всі пов'язані записи.

Експорт даних у CSV і ICS супроводжується розшифруванням тільки для авторизованого користувача. Імена анонімізуються, щоб зменшити ризики. Повідомлення про обробку даних локалізовано українською мовою.

Для захисту від SQL-ін'єкцій застосовано параметризовані запити. Управління доступом перевіряється за `user_id`, що унеможлиблює доступ до чужих записів. Реалізовано журналювання дій у логах. При фізичному доступі до пристрою дані залишаються захищеними, але рекомендовано використовувати захист на рівні ОС.

Безпека перевірялась шляхом тестування шифрування, автентифікації, експортованих файлів, реалізації прав доступу та відповідності вимогам GDPR. Усі заплановані тести пройдено. Виявлено дві проблеми: незахищене розміщення ключа шифрування та обрізання довгих симптомів при експорті — обидві усунуто.

Поточна реалізація враховує вимоги до локальних програм, підготовлена до масштабування у веб-формат із впровадженням HTTPS, JWT та захистом від мережеских атак. Застосунок відповідає чинним нормативам, забезпечує захист даних і готовий до подальшого розвитку.

3.5 Розрахунок економічного ефекту за темою ДР

Оцінка витрат, джерел фінансування, економічної ефективності та моделі монетизації застосунку для контролю перебігу вагітності дає змогу цілісно визначити фінансову життєздатність проєкту. Загальна сума витрат становить 475 000 грн на розробку і 250 800 грн на річну підтримку, що включає оплату праці розробника, дизайнера, тестувальника, витрати на обладнання, програмне забезпечення та супровід. Розрахунки здійснено у відповідності до ринку України станом на 2025 рік, з використанням середніх ставок для ІТ-фахівців і врахуванням безкоштовних технологій, таких як Python, Flask, SQLite та Tkinter. Це дозволило оптимізувати бюджет і мінімізувати непрямі витрати, зберігаючи водночас повну функціональність продукту. Витрати на підтримку, передбачені як щомісячні оновлення, адміністрування, технічна підтримка та тестування, відповідають типовим річним витратам для ІТ-рішень медичного спрямування.

Для покриття витрат і подальшого масштабування проєкту проаналізовано декілька джерел фінансування. До основних відносяться власні кошти розробників, зовнішні інвестиції, грантові програми та краудфандинг. Власні ресурси доцільно використовувати на початковому етапі для створення прототипу, однак через обмежений обсяг вони не здатні повністю покрити витрати. Перспективними є зовнішні інвестиції від венчурних фондів або акселераторів, які можуть надати фінансування у межах 200 000–2 000 000 грн, а також надати доступ до експертизи та допомогти у масштабуванні. Гранти міжнародних організацій (Horizon Europe, USAID) та державних програм (Український фонд стартапів) забезпечують безповоротне фінансування на суму до 1 500 000 грн, що робить їх оптимальними для соціально спрямованих ініціатив. Краудфандинг через платформи, як-от Спільнокошт або Kickstarter, може забезпечити додаткові 50 000–200 000 грн, одночасно залучаючи цільову аудиторію.

У межах прогнозування економічної ефективності проєкту враховано потенційну аудиторію, яка налічує близько 150 000 вагітних жінок щорічно, а також лікарів-акушерів і членів сімей. Очікується, що в перший рік 1%

користувачів скористаються преміум-функціями, а загальна аудиторія зростатиме на 10% щороку. Основною моделлю монетизації обрано freemium-стратегію, яка передбачає безкоштовний базовий функціонал і платний доступ до розширених можливостей (аналітика, психологічна підтримка, експорт даних тощо) за підпискою 100 грн/місяць. Допоміжними джерелами доходу є реклама в безкоштовній версії та партнерства з медичними установами. У перший рік очікуваний дохід складе близько 2 025 000 грн, що значно перевищує загальні витрати, забезпечуючи чистий прибуток понад 1,2 млн грн. У наступні роки, з урахуванням зростання аудиторії та нових партнерств, прибутковість зростатиме, а чистий приведений дохід (NPV) за три роки становитиме понад 4 млн грн. Початкові витрати окупаються вже протягом перших чотирьох місяців роботи системи.

Обрана комбінована модель монетизації — підписка як основа, реклама та партнерства як додаткові джерела — забезпечує фінансову сталість, соціальну доступність і можливість масштабування. Застосунок є конкурентоспроможним завдяки повній локалізації українською, соціальній спрямованості, простоті інтерфейсу та актуальності тематики. Ці фактори посилюють його позиції на ринку цифрових медичних сервісів, що динамічно зростає в Україні. Враховуючи всі показники, проєкт демонструє високу економічну ефективність і привабливість для інвесторів, грантодавців і користувачів.

Висновки до розділу 3

У третьому розділі виконано реалізацію та тестування віконного застосунку для контролю перебігу вагітності, розробленого з використанням Python, Tkinter, SQLite, cryptography, tkcalendar і Pillow, з адаптацією до веб-формату через серверну частину на Flask. Реалізація охопила всі ключові аспекти, визначені у вимогах розділів 1 і 2, включаючи функціональність, безпеку, локалізацію та зручність використання.

Серверна частина, описана в підрозділі 3.1, реалізована з використанням Flask і SQLAlchemy, забезпечує обробку запитів, управління даними та безпеку

через JWT, шифрування AES-128 і захист від SQL-ін'єкцій. Вона сумісна з віконною версією, використовуючи ту саму структуру бази даних SQLite, і підготовлена до інтеграції з eHealth у майбутній веб-версії.

Клієнтська частина, представлена в підрозділі 3.2, створена як віконний інтерфейс із Tkinter, із яскравим дизайном (блакитний #E6F0FA, рожевий #F4A7B9), іконками та віджетом `datepicker` для зручного вибору дати. Інтерфейс відповідає принципам UI/UX, забезпечуючи інтуїтивність, локалізацію українською та доступність для користувачів із різним рівнем цифрової грамотності. Функціонал включає календар, введення медичних даних, інформаційний розділ, психологічну підтримку та експорт даних.

Тестування показало, що застосунок відповідає всім основним функціональним вимогам: працюють модулі календаря, введення даних, інформаційного розділу, психологічної підтримки та експорту. Нефункціональні вимоги також виконано — інтерфейс реагує швидко, дані локалізовані українською мовою, реалізовано захист і шифрування. Застосовані заходи безпеки, зокрема шифрування даних і захист від SQL-ін'єкцій, відповідають положенням GDPR і Закону України. У процесі тестування виявлено кілька технічних проблем — серед них кодування при експорті в CSV і обрізання тексту в деяких елементах інтерфейсу. Ці недоліки були усунуті, що дозволило досягти стабільної роботи.

Тестування закладає основу для подальшого розширення застосунку до веб-версії, де будуть потрібні додаткові тести для мережових вразливостей (XSS, CSRF) і продуктивності при великій кількості користувачів. Результати свідчать про готовність застосунку до використання в локальному форматі та потенціал для масштабування

ЗАГАЛЬНІ ВИСНОВКИ

Бакалаврська робота присвячена створенню застосунку для контролю перебігу вагітності у вигляді віконної програми, з урахуванням подальшої адаптації до веб-формату. У процесі розробки було виконано повний цикл: від аналізу потреб майбутніх користувачів до впровадження, тестування та економічного розрахунку. В результаті розроблено інструмент, який відповідає вимогам зручності, безпеки та фінансової доцільності для вагітних жінок, лікарів і членів їхніх сімей.

Дослідження ринку показало, що в Україні існує стабільний попит на цифрові рішення для моніторингу вагітності, а кількість потенційних користувачів складає близько 300 тисяч щороку. Аналіз очікувань користувачів дозволив сформувавши перелік функцій, серед яких календар подій, введення медичних показників, інформаційний блок і засоби психологічної підтримки. Порівняння з уже існуючими аналогами, зокрема Flo та System Carebits, виявило потребу у якісній локалізації й глибшій інтеграції з національними платформами.

Проектування системи охопило визначення архітектури, структури бази даних і візуального інтерфейсу. Локальна база містить таблиці для зберігання інформації про користувачів, їхній стан здоров'я, події й нагадування, що гарантує впорядкованість і доступність інформації. Інтерфейс розроблено з урахуванням сучасних принципів дизайну: використано насичену кольорову палітру, віджет для вибору дат, а локалізація українською мовою робить застосунок зрозумілим для цільової аудиторії.

У програмі реалізовано як серверну, так і клієнтську частини. Сервер відповідає за обробку запитів і зберігання даних, клієнтська частина забезпечує роботу з інтерфейсом. Реалізовано експорт даних у формати CSV і ICS, що дозволяє синхронізувати події з Google Calendar або медичними системами. Передбачено можливість подальшої інтеграції з eHealth при переході до веб-версії. Перевірка роботи програми підтвердила виконання функціональних вимог, стабільність і надійний захист даних, а всі дрібні технічні помилки усунуто.

Аналіз витрат продемонстрував, що фінансування розробки (475 000 грн) і щорічної підтримки (250 800 грн) окупається за чотири місяці за рахунок доходів від підписки, реклами та партнерських програм, а чистий приведений дохід за три роки перевищує чотири мільйони гривень. Вибір комбінованої моделі монетизації на основі freemium дає змогу поєднати соціальну корисність продукту із комерційною стратегією.

Питання захисту інформації розв'язано шляхом шифрування чутливих даних, захисту від SQL-ін'єкцій, впровадженням багаторівневої моделі доступу та дотриманням вимог GDPR і українського законодавства. Користувачі різних ролей мають чітко розмежовані права, що гарантує захищеність персональної інформації.

Створений застосунок — це дієвий цифровий продукт для підтримки вагітних жінок, який поєднує простий інтерфейс, якісний локалізований контент і надійний рівень безпеки. Соціальна значущість і фінансова ефективність відкривають перспективу масштабування проєкту, розширення функціональності та впровадження веб-версії з повною інтеграцією з національною системою eHealth, що може позитивно вплинути на рівень пренатальної допомоги в Україні.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Закон України «Про внесення змін до деяких законодавчих актів України щодо функціонування телемедицини» [Електронний ресурс] // Верховна Рада України. – 2023. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/3301-20>.
2. Закон України «Про захист персональних даних» [Електронний ресурс] // Верховна Рада України. – 2010. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/2297-17>.
3. НАКАЗ Про затвердження порядків надання домедичної допомоги особам при невідкладних станах [Електронний ресурс] // Міністерство охорони здоров'я України. – 2022. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/z0356-22#n941>.
4. Положення про кабінет телемедицини закладу охорони здоров'я [Електронний ресурс] // Міністерство охорони здоров'я України. – 2015. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/z1401-15>.
5. Про затвердження Порядку надання медичної та/або реабілітаційної допомоги із застосуванням телемедицини на період дії воєнного стану [Електронний ресурс] // Міністерство охорони здоров'я України. – 2022. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/z1155-22>.
6. Про організацію надання медичної допомоги із застосуванням телемедицини в умовах воєнного стану [Електронний ресурс] // Міністерство охорони здоров'я України. – 2022. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/z0728-22>.
7. Balsam D., Bounds D.T., Rahmani A.M., Nyamathi A. Evaluating the impact of an app-delivered mindfulness meditation program to reduce stress and anxiety during pregnancy: pilot longitudinal study [Електронний ресурс] // JMIR Pediatrics and Parenting. – 2023. – Режим доступу до ресурсу: <https://pediatrics.jmir.org/2023/1/e53933>.

8. Best practices for RESTful web API design [Електронний ресурс] // Microsoft Azure. – 2025. – Режим доступу до ресурсу: <https://learn.microsoft.com/en-us/azure/architecture/best-practices/api-design>.
9. Chua J.Y.X., Shorey S. та ін. Effectiveness of mobile application-based perinatal interventions in improving parenting outcomes: a systematic review [Електронний ресурс] // Midwifery. – 2022. – Режим доступу до ресурсу: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9364944>.
10. Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions – eHealth Action Plan 2012-2020: innovative healthcare for the 21st century [Електронний ресурс] // Official Journal of the European Union. – 2012. – Режим доступу до ресурсу: <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=COM:2012:0736:FIN>.
11. Consolidated telemedicine implementation guide [Електронний ресурс] // Всесвітня організація охорони здоров'я. – 2022. – Режим доступу до ресурсу: <https://www.who.int/publications/i/item/9789240059184>.
12. Digital health [Електронний ресурс] // Всесвітня організація охорони здоров'я. – 2020. – Режим доступу до ресурсу: <https://www.who.int/health-topics/digital-health>.
13. eHealth: Digital health and care [Електронний ресурс] // Європейська Комісія. – 2025. – Режим доступу до ресурсу: https://health.ec.europa.eu/ehealth-digital-health-and-care_en.
14. Evans K., Donelan J., Rennick-Egglestone S., Cox S., Kuipers Y. Review of mobile apps for women with anxiety in pregnancy: maternity care professionals' guide to locating and assessing anxiety apps [Електронний ресурс] // Journal of Medical Internet Research. – 2022. – Режим доступу до ресурсу: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8987965>.
15. Flaherty K., Katsarakis A. Digital interactions in healthcare customer journeys [Електронний ресурс] // Nielsen Norman Group. – 2023. – Режим доступу до ресурсу: <https://www.nngroup.com/articles/healthcare-customer-journeys/>.

16. Global strategy on digital health 2020–2025 [Електронний ресурс] // Всесвітня організація охорони здоров'я. – 2021. – Режим доступу до ресурсу: <https://www.who.int/publications/i/item/9789240020924>.

17. Kamarudin S.S., Idris I.B., Sharip S. та ін. LoVE4MUM mobile app to prevent postpartum depression: protocol for a pilot randomized controlled trial [Електронний ресурс] // JMIR Research Protocols. – 2024. – Режим доступу до ресурсу: <https://www.researchprotocols.org/2025/1/e63564>.

18. Mannino R.G., Alvarez S.J.A., Greenleaf M., Parsell M., Mwalija C., Lam W.A. та ін. Navigating the complexities of mobile medical app development from idea to launch: a guide for clinicians and biomedical researchers [Електронний ресурс] // BMC Medicine. – 2023. – Режим доступу до ресурсу: <https://bmcmmedicine.biomedcentral.com/articles/10.1186/s12916-023-02833-7>.

19. MGHPDS Mobile App [Електронний ресурс] // MGH Center for Women's Mental Health. – 2023. – Режим доступу до ресурсу: <https://womensmentalhealth.org/research/app/>.

20. NIST Privacy Framework [Електронний ресурс] // National Institute of Standards and Technology. – 2020. – Режим доступу до ресурсу: <https://www.nist.gov/privacy-framework>.

21. Node.js Blog [Електронний ресурс] // OpenJS Foundation. – 2025. – Режим доступу до ресурсу: <https://nodejs.org/en/blog>.

22. OWASP Top 10 – 2021 [Електронний ресурс] // OWASP Foundation. – 2021. – Режим доступу до ресурсу: <https://owasp.org/Top10/>.

23. PostgreSQL 17.5 Documentation [Електронний ресурс] // The PostgreSQL Global Development Group. – 2025. – Режим доступу до ресурсу: <https://www.postgresql.org/docs/current/index.html>.

24. React Blog [Електронний ресурс] // React Team (Meta Platforms). – 2025. – Режим доступу до ресурсу: <https://react.dev/blog>.

25. Recommendation CM/Rec(2019)2 of the Committee of Ministers to member States on the protection of health-related data [Електронний ресурс] // Council of Europe. – 2019. – Режим доступу до

ресурсу: https://search.coe.int/cm/Pages/result_details.aspx?ObjectId=09000016809422f5.

26. Recommendations on digital interventions for health system strengthening [Електронний ресурс] // Всесвітня організація охорони здоров'я. – 2019. – Режим доступу до ресурсу: <https://www.who.int/publications/i/item/9789241550505>.

27. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 (General Data Protection Regulation) [Електронний ресурс] // Official Journal of the European Union. – 2016. – Режим доступу до ресурсу: <https://eur-lex.europa.eu/eli/reg/2016/679/oj>.

28. Rizzi S. та ін. A mindfulness-based app intervention for pregnant women: protocol for a pilot feasibility study [Електронний ресурс] // JMIR Research Protocols. – 2024. – Режим доступу до ресурсу: <https://www.researchprotocols.org/2024/1/e53890>.

29. Security and Resilience in eHealth: infrastructures and services [Електронний ресурс] // European Union Agency for Cybersecurity (ENISA). – 2012. – Режим доступу до ресурсу: <https://www.enisa.europa.eu/publications/security-and-resilience-in-ehealth-infrastructures-and-services>.

30. The HIPAA Privacy Rule [Електронний ресурс] // U.S. Department of Health and Human Services. – 2000. – Режим доступу до ресурсу: <https://www.hhs.gov/hipaa/for-professionals/privacy/index.html>.