

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

(повна назва інституту/факультету)

кафедра БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

(повна назва кафедри)

«До захисту допущено»

В.о.завідувачки кафедри БМК

Світлана АЛХІМОВА

(підпис)

(ініціали, ПРІЗВИЩЕ)

«__» __червня__ 2025р.

Дипломна робота

на здобуття ступеня бакалавра

за освітньо-професійною програмою

«Комп'ютерні технології в біології та медицині»

зі спеціальності 122 «Комп'ютерні науки»

на тему:

Веб-застосунок моніторингу стану здоров'я

інсулінозалежних пацієнтів

Виконав: студент IV курсу, групи БС-13

ГРІДІН БОГДАН СЕРГІЙОВИЧ

(прізвище, ім'я, по батькові)

(підпис)

Науковий керівник: *старший викладач каф. біомедичної
кібернетики (БМК), Кисляк Сергій Володимирович*

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

(підпис)

Рецензент: *доцент каф. біомедичної інженерії, д-р філософії
(PhD), Рудніцька Олена Володимирівна*

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

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

Студент

(підпис)

Київ – 2025 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
 Інститут (факультет) БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ
 (повна назва)

Кафедра БІОМЕДИЧНОЇ КІБЕРНЕТИКИ
 (повна назва)

Рівень вищої освіти – перший (бакалаврський)

Спеціальність 122 «Комп'ютерні науки»

Освітньо-професійна програма «Комп'ютерні технології в біології та медицині»
 (код і назва)

ЗАТВЕРДЖУЮ

В.о.завідувачки кафедри БМК

Світлана АЛХІМОВА

« » квітень 2025 р.

ЗАВДАННЯ на дипломну роботу студенту

ГРІДІНА БОГДАНА СЕРГІЙОВИЧА

(прізвище, ім'я, по батькові)

1. Тема роботи

Веб–застосунок моніторингу стану здоров'я інсулінозалежних пацієнтів

керівник роботи

Кисляк Сергій Володимирович

(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

22–23 травня 2025 року

3. Вихідні дані: Відсутні

4. Зміст роботи: Аналіз предметної області, Засоби реалізації веб–застосунку, Реалізація веб–застосунку, Узагальнення результаті роботи.

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): презентація на 15 слайдах, 32 рисунків, 8 таблиць.

6. Консультанти розділів роботи^{1*}

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

^{1*} Якщо визначені консультанти. Консультантом не може бути зазначено наукового керівника магістерської дисертації.

7. Дата видачі завдання **16 квітня 2025р.**

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Отримати завдання за темою ДР на практику	До 24.12.2024р.	<i>виконано</i>
2	Переддипломна практика. Виконання практичної частини ДР та додаткових розділів	1–4 тиждень за графіком практики	<i>виконано</i>
3	Виконання основних розділів ДР (Вступ, Аналіз предметної області, Засоби реалізації веб–застосунку, Реалізація веб–застосунку, Узагальнення результатів роботи)	Протягом усієї практики	<i>виконано</i>
4	Перевірка ДР науковим керівником	до 20.05.2025	<i>виконано</i>
5	Подання в електронному вигляді ДР на перевірку нормоконтролера та плагіат (StrikePlagiarism.com). Доопрацювання ДР	Не пізніше 23.05.2025	<i>виконано</i>
6	Отримати відгук від керівника ДР	Не пізніше 05.06.2025	<i>виконано</i>
7	Надати пакету документів по ДР та супровідних до неї документів на засідання кафедри		<i>виконано</i>
8	Подання ДР рецензенту. Отримання рецензії.	09.06 – 13.06.2025	<i>виконано</i>
9	Захист ДР в ЕК	16.06 – 21.06.2025	

Студент

_____ (підпис)

Богдан ГРІДІН

_____ (ім'я, ПРІЗВИЩЕ)

Керівник ДР

_____ (підпис)

Сергій КИСЛЯК

_____ (ім'я, ПРІЗВИЩЕ)

Нормоконтролер

_____ (підпис)

Галина КОРНІЄНКО

_____ (ім'я, ПРІЗВИЩЕ)

АНОТАЦІЯ

Дипломна робота за темою «Веб–застосунок моніторингу стану здоров'я інсулінозалежних пацієнтів» виконана студентом кафедри біомедичної кібернетики ФБМІ Грідіним Богданом Сергійовичем зі спеціальності 122 «Комп'ютерні науки» за освітньо–професійною програмою «Комп'ютерні технології в біології та медицині» та складається зі: вступу; 4 розділів (*Аналіз предметної області, Засоби реалізації веб–застосунку, Реалізація веб–застосунку, Узагальнення результатів роботи*), висновків до кожного з цих розділів; загальних висновків; списку використаних джерел, який налічує 50 джерел та додатків. Загальний обсяг роботи 70 сторінок.

Актуальність теми.

Цукровий діабет потребує постійного контролю рівня глюкози та інсулінової терапії, що створює щоденні виклики для пацієнтів. Існуючі застосунки або складні у використанні, або не враховують індивідуальні потреби. Тема роботи є актуальною, адже може полегшити самоконтроль та покращити якість життя інсулінозалежних хворих.

Мета і завдання роботи.

Метою роботи є покращення якості моніторингу інсулінозалежних пацієнтів шляхом аналізу сучасних засобів, структуризації медичних даних та виявленні потенційно небезпечних змін.

Її досягнення передбачає вирішення наступних завдань:

1. Проаналізувати наявні аналоги
2. Обрати засоби реалізації
3. Реалізувати веб–застосунок

Використані методи. У дипломній роботі було використано мову програмування Python та веб фреймворк Django.

Отримані результати. Результатом виконаною роботи є веб–застосунок моніторингу стану здоров'я інсулінозалежних пацієнтів.

Публікації. За результатами виконаної роботи не було опубліковано матеріалів конференцій та статей.

Ключові слова. діабет, Python, Django, веб–застосунок, лікар, пацієнт.

Бібліографічний опис ДР

Грідін Б. С. Веб–застосунок моніторингу стану здоров’я інсулінозалежних пацієнтів: дипломна роб. бакалавра: 122 Комп’ютері науки / Грідін Богдан Сергійович. – Київ, 2025. – 70 с

ANNOTATION

Diploma work on the topic ‘Web application for monitoring the health status of insulin-dependent patients’ was performed by a student of the Department of Biomedical Cybernetics of FBMI *Hridin Bohdan Serhiyovych*, speciality 122 ‘Computer Science’ under the educational and professional programme ‘Computer Technologies in Biology and Medicine’ and consists of introduction; 4 chapters (*Analysis of the subject area, Means of implementing a web application, Implementation of a web application, Summarising the results of the work*), conclusions to each of these chapters; general conclusions; a list of references, which includes 50 sources, and appendices. The total volume of the work is 70 pages.

Relevance of the topic.

Diabetes mellitus requires constant glucose monitoring and insulin therapy, which creates daily challenges for patients. Many existing applications are either difficult to use or do not take into account individual needs. My topic is relevant because it can facilitate self-control and improve the quality of life of insulin-dependent patients.

Purpose and objectives of the work.

The aim of the work is to improve the quality of monitoring of insulin-dependent patients by analysing modern tools, structuring medical data and identifying potentially dangerous changes.

Its achievement involves solving the following tasks:

1. Analyse existing analogues
2. Select the means of implementation
3. Implement the web application

Methods used. The Python programming language and the Django web framework were used in the thesis.

Results obtained. The result of the work is a web application for monitoring the health status of insulin.

Publications. Based on the results of the work done, no conference materials or articles were published.

Keywords. diabetes, Python, Django, web-application, doctor, patient

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	10
ВСТУП	11
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	13
1.1. Хронічне захворювання діабет.....	13
1.2. Лікування інсулінозалежних пацієнтів.....	18
1.2.1. Інсулінотерапія	18
1.2.2. Контроль рівня цукру в крові.....	20
1.2.3. Спосіб життя при діабеті	21
1.3. Огляд аналогів	24
1.3.1. Glooko	24
1.3.2. mySugr.....	25
1.3.3. TidePool.....	26
1.3.4. Blood Sugar Tracker – Diabetes	27
Висновки з розділу 1	28
РОЗДІЛ 2 ЗАСОБИ РЕАЛІЗАЦІЇ ВЕБ-ЗАСТОСУНКУ	29
2.1. Мова програмування PYTHON.....	29
2.2. Бібліотека PILLOW.....	29
2.3. Бібліотека PLOTLY	30
2.4. Бібліотека DJANGO–WIDGET–TWEAKS.....	30
2.5. Середовище розробки PYSCHARM	31
2.6. Фреймворк DJANGO	31
2.7. Бази даних DJANGO. SQLITE	32
2.8. Шаблони DJANGO	33
2.9. Захист інформації в DJANGO	34
Висновок до розділу 2	35

РОЗДІЛ 3 РЕАЛІЗАЦІЯ ВЕБ–ЗАСТОСУНКУ	37
3.1. ПРОЕКТУВАННЯ БАЗИ ДАНИХ	37
3.2. ЛОГІКА РОБОТИ ВЕБ–ЗАСТОСУНКУ	42
3.3. ІНТЕРФЕЙС ВЕБ–ЗАСТОСУНКУ	46
3.3.1. Реєстрація та автентифікація.....	46
3.3.2. Вхід як ‘пацієнт’	48
3.3.3. Вхід як «лікар».....	55
3.4. РОЗРАХУНОК ЕКОНОМІЧНОГО ЕФЕКТУ	59
Висновок до розділу 3	61
РОЗДІЛ 4 УЗАГАЛЬНЕННЯ РЕЗУЛЬТАТІВ РОБОТИ.....	62
4.1. ПОРІВНЯННЯ ВЕБ–ЗАСТОСУНКУ З АНАЛОГАМИ	62
Таблиця 4.1	62
4.2. ВИКОРИСТАННЯ ВЕБ–ЗАСТОСУНКУ	64
4.3. ПОДАЛЬШИЙ РОЗВИТОК ВЕБ–ЗАСТОСУНКУ.....	64
Висновок до розділу 4	65
ЗАГАЛЬНІ ВИСНОВКИ	66
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	68

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БД – база даних

ПЗ – програмне забезпечення

ЦД – цукровий діабет

BGM – Blood Glucose Monitoring – моніторинг рівня глюкози в крові

CGM – Continuous Glucose Monitoring – безперервний моніторинг

глюкози

HbA1c – показник крові, що відображає середній вміст глюкози в крові за період до 3 місяців

JPG – Joint Photographic Group – формат стиснення зображень

PNG – Portable Network Graphics – переносний графічний формат

HTML – HyperText Markup Language – мова розмітки гіпертексту

CSS – Cascading Style Sheets – каскадні таблиці стилів

IDE – Integrated Development Environment – інтегроване середовище

розробки

MVT – Model–View–Template – модель–представлення–шаблон

ORM – Object–Relational Mapping – об’єктно реляційне відображення

SQL – Structured Query Language – мова структурованих запитів

XSS – Cross–Site Scripting – міжсайтове скриптування

CSRF – Cross–Site Request Forgery – підробка міжсайтового запиту

PBKDF2 – Password–Based Key Derivation Function 2 – шифр для
пароллю

ВСТУП

Цукровий діабет є досить складним захворюванням, з яким щодня стикається велика кількість людей. Для підтримання стабільного стану пацієнтам потрібно постійно контролювати рівень цукру в крові, вводити інсулін, а також стежити за іншими показниками. Усе це займає багато часу і вимагає зосередженості. В наш час, коли цифрові технології швидко розвиваються, виникає можливість створити засоби, які б могли оптимізувати цей процес. Проте більшість готових рішень або складні у використанні, або не зовсім підходять для повсякденного застосування. Через це постає потреба у простому, зрозумілому веб–застосунку, який допоможе людям краще слідкувати за своїм станом здоров'я.

Мета і завдання роботи

Метою роботи є покращення якості моніторингу інсулінозалежних пацієнтів шляхом аналізу сучасних засобів, структуризації медичних даних та виявленні потенційно небезпечних змін.

Її досягнення передбачає вирішення наступних *завдань*:

1. Проаналізувати наявні аналоги
2. Обрати засоби реалізації
3. Реалізувати веб–застосунок

Використані методи. У дипломній роботі було використано мову програмування Python та веб фреймворк Django.

Отримані результати. У процесі виконання роботи вдалося створити веб–застосунок, який дозволяє інсулінозалежним пацієнтам фіксувати показники свого стану, зокрема рівень глюкози в крові, а також переглядати ці дані у вигляді графіків. Застосунок має простий інтерфейс, тож ним легко користуватися навіть тим, хто не має технічних знань. У програмі реалізовано реєстрацію, авторизацію, внесення нових записів і можливість аналізувати дані.

Публікації. За результатами виконаної роботи не було опубліковано матеріалів статей та конференцій.

Структура роботи

Дипломна робота за темою «Веб–застосунок моніторингу стану здоров’я інсулінозалежних пацієнтів» виконана студентом *Гридіним Богданом Сергійовичем* зі спеціальності 122 «Комп’ютерні науки» за освітньо–професійною програмою «Комп’ютерні технології в біології та медицині», побудована за класичним типом та викладена на 69 сторінках машинописного тексту. Вона складається з: вступу; 4 розділів (*Аналіз предметної області, Засоби реалізації веб–застосунку, Реалізація веб–застосунку, Узагальнення результатів роботи*), висновків до кожного з цих розділів; загальних висновків; списку використаних джерел, який налічує 50 джерел та додатків (12 – на кирилиці, 38 – на латиниці). В роботі представлено 32 рисунків і 8 таблиць

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Розробка веб–застосунку для моніторингу стану здоров’я інсулінозалежних пацієнтів потребує ґрунтовного аналізу предметної області. У даному розділі здійснюється огляд елементів, пов’язаних із захворюванням на цукровий діабет, зокрема його класифікації, особливостей перебігу та впливу на якість життя пацієнтів. Розглядаються сучасні програмні рішення, які виконують функції моніторингу та підтримки пацієнтів, з метою визначення їхніх можливостей, переваг і недоліків.

1.1. Хронічне захворювання діабет

За підрахунками у 2022 році 828 мільйонів дорослих (віком від 18 років) мали діабет, що на 630 мільйонів перевищує показник 1990 року. З 1990 по 2022 рік поширеність діабету, скоригована за віком, збільшилася в 131 країні серед жінок та в 155 країнах серед чоловіків. У більшості країн, зокрема в тих, що мають низький та середній рівень життя, лікування діабету збільшилося недостатньо, якщо порівнювати зі зростанням його поширення. Вплив діабету на здоров’я населення, особливо недіагностованого, все частіше відчувають саме країни з низьким і середнім рівнем доходу на душу населення. Розширення медичного страхування та первинної медичної допомоги має супроводжуватися програмами боротьби з діабетом, спрямовані на перебудову системи медичних послуг та ефективне використання ресурсів задля покращення раннього виявлення та ефективного лікування діабету [9].

За останнє десятиліття поширеність діабету в Україні зросла вдвічі. 7.1% дорослого населення України страждають на діабет, а неінфекційні хвороби є головною причиною передчасної смерті [34]. Майже 3% основних

причин інвалідності серед дорослого населення України пов'язані з цукровим діабетом [42].

Цукровий діабет (надалі ЦД) – це хронічне захворювання, яке розвивається, коли підшлункова залоза перестає продукувати достатню кількість інсуліну, або організм не може належно використовувати вироблений інсулін [46].

Основними симптомами ЦД є:

- відчуття спраги;
- надмірне сечовипускання;
- погіршення зору;
- відчуття втоми, слабкість;
- свербіж;
- втрата ваги (при ЦД 1–го типу);
- надмірна вага (при ЦД 2–го типу)[46].

Діабет збільшує ймовірність ускладнень, таких як ампутація, втрата зору та ниркова недостатність, крім того хворобу пов'язують із серцево–судинними захворюваннями, деменцією, деякими типами раку та інфекціями, зокрема туберкульозом та важкою формою COVID–19. Діабет можна виявити на рівні первинної медичної допомоги. Лікування пероральними гіпоглікемічними ліками або інсуліном, а також використання нових ін'єкційних препаратів зменшують ризик ускладнень. Відсутність лікування або затримка з лікуванням підвищує ймовірність ускладнень та летальності. Поширеність діабету та охоплення лікуванням діабету, а також те, як змінилося лікування, є важливими показниками здоров'я населення та результативності систем охорони здоров'я [45].

ЦД є одним з перших захворювань які відомі людству. Незважаючи на його сучасну поширеність та складність лікування, згадки про хворобу зустрічаються вже у найдавніших історичних джерелах. Вивчення останків первісних людей засвідчує, що вони теж страждали від різних хвороб, хоча

інформація про ураження внутрішніх органів, таких як підшлункова залоза, вкрай обмежена через особливості археологічних знахідок. Незважаючи на це, беручи до уваги описані симптоми і спосіб життя стародавніх людей, можна припустити, що діабет існував і тоді, хоча й мав, певно, нижчу поширеність, ніж у майбутньому. Це могло бути викликано фізично активним способом життя, важкою працею та обмеженим доступом до калорійної їжі [45].

Перші описи симптомів, які відповідають діабету, згадуються 1550 роком до нашої ери, а саме у папірусі Еберса – давньому медичному творі, який був складений в Стародавньому Єгипті. Також у роботах лікарів Стародавньої Індії були описані події надмірного сечовипускання, солодкий смак сечі, відчуття надмірної спраги та виснаження організму. Відомий грецький лікар Аретей Каппадокійський, який жив у I столітті н.е., був одним із перших, хто узагальнив симптоми діабету і запропонував сам термін «діабет», що походить від грецького слова, яке означає «проходити крізь». Він описував хворобу як «розчинення» організму через надмірне сечовиділення, підкреслюючи її тяжкий перебіг і невпинну втрату рідини [45]. Його бачення хвороби: «Діабет – це жахлива недуга, не дуже поширена серед чоловіків, що розплавляє плоть та кінцівки у сечу. Пацієнти постійно мочаться безперервним потоком, як із отвору акведуку. Життя коротке, неприємне та сповнене болю, спрага невгамовна, пиття надмірне і непропорційне до великого об'єму сечі, оскільки її випускається навіть більше... Якщо деякий час вони утримуються від пиття, їхні роти пересихають, а тіла висихають; внутрішні органи здаються випаленими, пацієнти страждають від нудоти, неспокою та пекучої спраги і невдовзі помирають» [45].

У Середньовіччі великий внесок у вивчення діабету зробили представники арабської медицини. Авіценна (Ібн Сіна) у своєму «Каноні лікарської науки» описав клінічну картину хвороби, запропонував диференціацію форм діабету, вказував на важливість дієти, фізичної

активності, та застосування рослинних засобів – таких, як топінамбур, люпин і пажитник. Особливу увагу він приділяв ускладненням діабету, зокрема гангрені стопи [45].

У другій половині сімнадцятого століття було відкрито, що сеча хворих на діабет має солодкий присмак. Це стало основою для діагностики. Наступним ключовим кроком у відкритті діабету стало виявлення у 1869 вченим Паулем Лангергансом скупчення клітин у підшлунковій залозі, які пізніше назвали «острівці Лангерганса» – особливі кластери гормон–продукуючих клітин, що регулюють рівень глюкози в крові [20]. Далі вже в 1889 році, вченим Йозифом фон Мерінгом та Оскаром Мінковським, шляхом видалення підшлункової залози у собаки було вперше експериментальна відтворено ЦД [45].

Наступною важливою подією у історії діабету стала зустріч вчених Фредерика Бентінга, Чарльза Герберта Беста та лікаря Джона Маклеода. Саме 17 травня 1921 року Бантінг, Бест і Маклеод вперше зібралися разом, щоб розпочати дослідження і з'ясувати, як вилучити інсулін з підшлункової залози собаки. Їхній метод полягав у перев'язуванні панкреатичної протоки для знищення речовин, що руйнують інсулін, залишаючи острівці Лангерганса неушкодженими. Отриманий екстракт вводили собакам, які не виробляли власного інсуліну через видалення підшлункової залози, щоб оцінити його вплив на рівень цукру в крові. Спочатку їхні дослідження просувалися повільно і багато експериментів завершувалися невдачею. Однак Бантінг і його команда помітили, що введення їхнього екстракту стабільно знижувало рівень цукру в крові, що зміцнило їхню впевненість у значущості відкриття. До листопада їм вдалося підтримувати життя собаки з діабетом, застосовуючи інсуліновий екстракт протягом 70 днів. Успіх їхнього експерименту можна вважати за відкриття інсуліну[15].

Інсулін – це білковий гормон, що синтезується підшлунковою залозою та контролює рівень глюкози в крові. Його вироблення відбувається в бета–клітинах, які є частиною острівців Лангерганса[43]. Оскільки інсулін відіграє

ключову роль у регуляції рівня глюкози в крові, його недостатність або порушення функціонування призводить до розвитку цукрового діабету. Залежно від причин цього порушення, виділяють різні типи діабету, кожен з яких має свої особливості перебігу.

Виділяють три основних види діабету у людей:

- ЦД 1-го типу – виникає внаслідок руйнування β -клітин підшлункової залози через аутоімунний процес, що запускається впливом тригерних факторів навколишнього середовища у людей з генетичною схильністю. У процесі розвитку захворювання залучаються антитіла до острівців Лангерганса, які реагують на різні антигени β -клітин. Вони можуть виявлятися за багато місяців або навіть років до появи перших симптомів цукрового діабету. У цей час поступово знижується секреторна функція β -клітин, що врешті-решт призводить до клінічного прояву хвороби з абсолютним дефіцитом інсуліну;
- ЦД 2-го типу – виникає переважно через інсулінорезистентність (знижену чутливість до інсуліну) та відносну недостатність інсуліну. Зазвичай розвивається у дорослому віці, а ризик захворювання значно підвищується у людей із зайвою вагою, шкідливими звичками, такими як куріння, або малорухливим способом життя;
- гестаційний діабет – вперше виявляється під час вагітності та характеризується підвищеним рівнем глюкози в крові, який перевищує норму, але не досягає порогу, необхідного для діагностики цукрового діабету [48].

У роботі основна увага приділяється саме інсулінозалежним пацієнтам, зокрема особам із цукровим діабетом 1-го типу та деякими випадками діабету 2-го типу [47], які потребують інсулінотерапії. Саме для цієї категорії хворих моніторинг стану здоров'я є критично важливим, оскільки дозволяє уникати гострих ускладнень, таких як гіпоглікемія та гіперглікемія, і забезпечувати стабільний контроль рівня глюкози в крові.

1.2. Лікування інсулінозалежних пацієнтів

Інсулінозалежний цукровий діабет – це хвороба, що вимагає довічного замісного лікування інсуліном. Головна мета лікування – утримувати рівень глюкози в межах, максимально наближених до фізіологічних показників норми, аби запобігти як гострим, так і хронічним ускладненням. З часу відкриття інсуліну методи лікування значно покращились: від застосування тваринного інсуліну до використання нинішніх аналогів людського інсуліну й інсулінових pomp. На сьогодні лікування інсулінозалежних хворих ґрунтується на засадах інтенсивної інсулінотерапії, самоконтролю рівня глюкози, корекції способу життя тощо. Лікування інсуліном вкрай важливе для інсулінозалежних людей, оскільки відмінна риса лікування – відсутність або майже повна відсутність функції β -клітин.

1.2.1. Інсулінотерапія

Інсулінотерапія ділиться на два головні типи введення інсуліну. Найпоширеніший тип, коли інсулін вводять шприцом, ручкою або пластирем для діабетиків. Менш поширений, але більш ефективний тип інсулінотерапії проводиться помпою, за допомогою якої глюкоза вимірюється в один момент часу (BGM) або безперервного моніторингу глюкози протягом дня і ночі (CGM) [24]. Відтак ліки швидше надходять у кров та починають діяти майже відразу, з найвищою ефективністю. Саме тому інсулінові препарати у формі крапель, інгаляторів, розчинів та таблеток так і не здобули широкого поширення: за таких способів введення неможливо точно визначити об'єм ліків, що потрапляє в кров. Через це пацієнти вимушені робити додатковий аналіз крові на цукор через деякий час після вживання ліків, що не сприяє зменшенню кількості уколів.

Розглянемо перший тип введення інсуліну, а саме шприцом, шприцом–ручкою тощо. Використання інсулінового шприца є зручним через його доступність і точність дозування. Він дозволяє чітко бачити кількість

препарату, що вводиться, та дає змогу змінювати голки за потребою. Водночас такий спосіб менш зручний для щоденного застосування: потрібно мати інсулін у флаконі, набирати його вручну, що займає більше часу та вимагає уважності. Шприц-ручка, навпаки, зручніша в користуванні, особливо поза домом. Вона дозволяє вводити інсулін швидше й непомітніше, але може бути дорожчою й менш універсальною щодо вибору доз [8].

Щодо використання приладів CGM, тобто pomp, то варто розглядати їх використання з перших днів встановлення діагнозу цукровий діабет, що потребує регулювання інсуліну. Інсулінова помпа – це сучасний медичний пристрій, який забезпечує цілодобове дозоване введення інсуліну в організм людини. Її головна функція – імітація природного режиму вироблення інсуліну, який у здорової людини продукується підшлунковою залозою постійно, з невеликими збільшення вироблення перед їжею. Це дає можливість уважно стежити за рівнем глюкози та вчасно коригувати дозування інсуліну та зміни у способі життя, а також зменшує навантаження через часті емоційні тривоги. Зовні помпа нагадує невеликий електронний пристрій, який пацієнт носить при собі. Через спеціальну інфузійну систему (тонку трубку з голкою або канюлею) інсулін потрапляє під шкіру. Такий спосіб введення дозволяє безперервно постачати інсулін у фоновому режимі та додатково вводити разові дози перед вживанням їжі або при підвищеному рівні цукру. Пристрій дозволяє програмувати дуже малі дози, що особливо корисно для дітей або пацієнтів, які мають низьку потребу в інсуліні. Це сприяє стабільнішому рівню глюкози в крові й знижує ризик виникнення як гіпоглікемії, так і гіперглікемії. Ще однією значною перевагою використання помпи є зменшення кількості ін'єкцій. Канюля встановлюється під шкіру лише один раз на кілька днів, тому пацієнт позбавлений необхідності щоденно робити кілька уколів, як це потрібно при введенні інсуліну за допомогою шприца або ручки, що значно підвищує комфорт і зменшує психологічний тиск, особливо у дітей або людей, які бояться уколів. Попри всі переваги, інсулінова помпа має і свої недоліки. Щоб ефективно

користуватися пристроєм, потрібно розуміти принципи його роботи, вміти налаштовувати дози, правильно змінювати інфузійну систему та слідкувати за рівнем цукру в крові. Це вимагає від пацієнта відповідальності й постійного самоконтролю. Ще одним недоліком використання помпи є регулярне технічне обслуговування, адже інфузійну систему потрібно змінювати кожні 2–3 дні, і при цьому важливо уважно слідкувати, щоб не було збоїв у подачі інсуліну, адже це може призвести до швидкого зростання рівня глюкози в крові. Крім того, вагомим фактором є вартість. Як сам пристрій, так і витратні матеріали до нього коштують чимало, і не всі пацієнти можуть дозволити собі постійне користування помпою [6].

1.2.2. Контроль рівня цукру в крові

Систематичний моніторинг глікемії є головним елементом у лікуванні інсулінозалежного діабету. Регулярні вимірювання рівня глюкози крові за допомогою глюкометра або інших приладів необхідні для корекції терапії та досягнення цільових показників глікемії [4],[2]. Навіть при застосуванні CGM пацієнти мають мати можливість зробити BGM. Використання глюкометра є невід’ємною складовою інтенсивних втручань, що підтверджує його користь для контролю глікемії [41]. Зазвичай вимірювання цукру в крові проводиться:

- натще (перед сніданком);
- перед кожним прийомом їжі;
- через 1–2 години після прийому їжі (особливо при застосуванні швидкодіючих інсулінів);
- перед сном;
- під час та після фізичних навантажень;
- при підозрі на гіпоглікемію або гіперглікемію (за симптомами);
- після корекції гіпоглікемії (для контролю відновлення рівня глюкози) [41].

CGM дозволяє відстежувати динаміку глікемії цілодобово. Пристрої CGM використовують сенсори, встановлені під шкірою, які постійно вимірюють концентрацію глюкози. Це дає змогу краще розуміти вплив їжі та фізичних навантажень на рівень цукру та швидко виявляти небезпечні коливання за допомогою звукових чи вібраційних сигналів [30][2]. Рекомендується розглядати початок використання CGM з моменту встановлення діагнозу, що вимагає інсулінотерапії [41]. Часте використання CGM пов'язане з кращим контролем глікемії – наприклад, значним зниженням рівня HbA1c у хворих на ЦД [3][2].

Належний самоконтроль глікемії передбачає не лише регулярні вимірювання, а й аналіз результатів та корекцію терапії за потреби. Правильно призначена терапія у поєднанні з адекватним самоконтролем глікемії та досягненням цільових рівнів запобігає розвитку ускладнень діабету [10]. Лікування діабету складається з інсулінотерапії та її корекції на основі результатів моніторингу і самоконтролю [10]. Пацієнту рекомендується вести щоденник глікемії (записувати виміряні значення), коригувати дозу інсуліну та раціон відповідно до результатів, своєчасно попереджувати високий або низький рівень глюкози, і регулярно консультуватися з лікарем щодо досягнення глікемічних цілей.

1.2.3. Спосіб життя при діабеті

Здоровий спосіб життя є невід'ємною частиною комплексного лікування інсулінозалежного ЦД. Завдяки дієті та фізичним навантаженням можна контролювати діабет і відтерміновувати розвиток ускладнень [17]. Раціон хворого на ЦД має бути збалансованим, він повинен містити багато овочів, фруктів, бобових, горіхів та цільнозернових продуктів [37]. Рекомендовано заповнювати половину тарілки неглюкозними овочами, чверть – білковими стравами (пісне м'ясо, риба, рослинні білки), а решту – «якісними» вуглеводами (цілі крупи, крохмалисті овочі, фрукти, нежирні молочні продукти) [33]. При цьому слід обмежувати споживання насичених жирів і простих цукрів та надавати перевагу клітковині. Важливий контроль

за кількістю вуглеводів, адже їх розподіл по днях і підрахунок дозволяє уникати різких коливань рівня глюкози. Продукти з низьким глікемічним індексом допомагають забезпечити більш стабільний глікемічний профіль [49].

Регулярна фізична активність значно покращує чутливість до інсуліну, сприяє нормалізації маси тіла і зниженню ризику серцево-судинних ускладнень. Рекомендується щонайменше 150–300 хвилин помірної або 70–140 хвилин інтенсивної аеробної фізичної активності протягом тижня [33]. До таких активностей можна віднести: швидку ходьбу, біг, плавання або велопрогулянки. Додатково рекомендують виконувати силові вправи 2–3 рази на тиждень для зміцнення м'язів. Такі навантаження, поєднані зі збалансованим харчуванням, дозволяють ефективніше контролювати глікемію і запобігати ускладненням діабету [17][33].

Якісний сон і психоемоційна рівновага також є важливими. Регулярний сон (7–9 годин щоночі) забезпечує гомеостаз обміну речовин. Недосипання погіршує толерантність до глюкози й підвищує інсулінорезистентність. Крім того, хронічний стрес підвищує рівень кортизолу і, як наслідок, глюкози в крові [14]. Тому хворим на ЦД рекомендують дотримуватися режиму відпочинку, використовувати техніки релаксації та по можливості уникати ситуацій психоемоційного перенапруження. Важливо піклуватися про психологічний комфорт: тривога, депресія чи хронічного стресу можуть погіршувати самоконтроль і якість життя. Фахівці радять підтримувати позитивний настрій, за необхідності звертатися за психологічною підтримкою і долучатися до груп пацієнтів – загалом «піклуватися про свій емоційний стан» у поєднанні з іншими здоровими звичками [14].

Не менш важливо повністю відмовитися від шкідливих звичок. Куріння ускладнює перебіг діабету: воно знижує чутливість тканин до інсуліну і підвищує ризик серцево-судинних ускладнень [14]. Також потрібно уникати вживання алкоголю, оскільки алкоголь може спричиняти небезпечні стрибки цукру (гіпоглікемію під час метаболізму алкоголю і гіперглікемію після

відновлення). Відмова від тютюну та спиртного є стандартною частиною змін способу життя при діабеті [14].

1.3. Огляд аналогів

1.3.1. Glooko

Glooko – це платформа для моніторингу цукрового діабету, яка дозволяє пацієнтам і лікарям автоматично збирати, зберігати та аналізувати дані про рівень глюкози (рис. 1.1). Вона підтримує синхронізацію з багатьма пристроями, зокрема глюкометрами, інсуліновими помпами та системами безперервного моніторингу глюкози (CGM). Дані зберігаються у хмарному середовищі, що забезпечує доступ до них у будь-який момент. Платформа формує звіти та графіки, які допомагають відстежувати зміни рівня глюкози та аналізувати вплив різних факторів на стан здоров'я користувача.

Glooko має функцію віддаленого доступу для лікарів, що дає змогу здійснювати дистанційний контроль та коригувати терапію. Також передбачена інтеграція з деякими фітнес-трекерами й іншими медичними застосунками, що може бути корисним для комплексного моніторингу. Однак, платформа має певні обмеження – частина функціоналу доступна лише за підпискою, а її ефективність залежить від сумісності з конкретними пристроями, які використовує пацієнт.

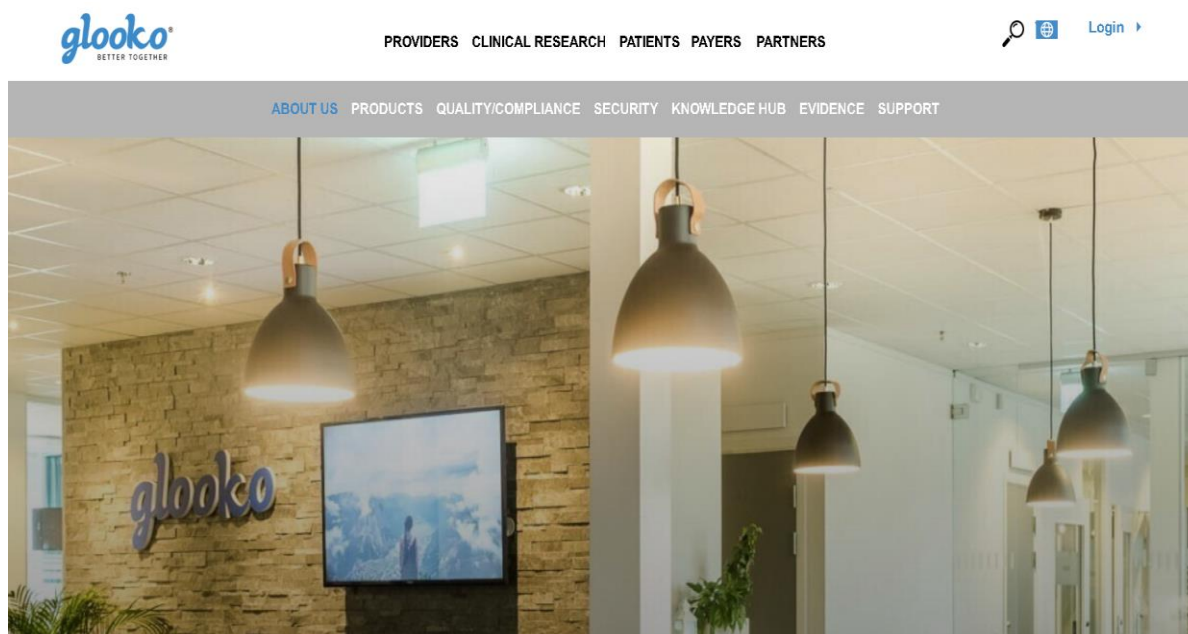


Рисунок 1.1 – Платформа Glooko

1.3.2. mySugr

mySugr – це мобільний застосунок для самоконтролю цукрового діабету, який дозволяє користувачам вести цифровий щоденник, реєструвати рівень глюкози в крові та аналізувати вплив різних факторів на перебіг захворювання (рис. 1.2).

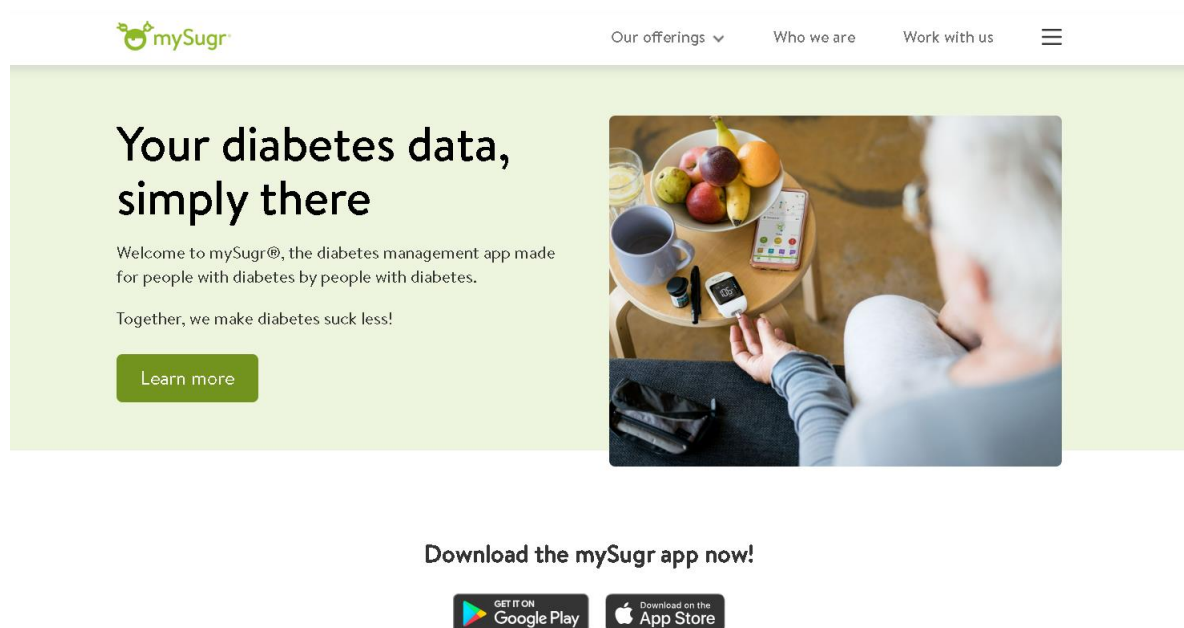


Рисунок 1.2 – Головний сайт застосунку mySugr

Програма підтримує як ручне введення даних, так і автоматичну синхронізацію з деякими глюкометрами. Окрім рівня глюкози, застосунок дає змогу записувати інформацію про прийом інсуліну, спожиті вуглеводи, фізичну активність та загальне самопочуття, що сприяє комплексному підходу до моніторингу стану здоров'я. Усі дані зберігаються у застосунку та візуалізуються у вигляді графіків і звітів, що полегшує аналіз трендів змін рівня глюкози. Додаток має додаткові функції, серед яких створення звітів для лікарів, розрахунок активного інсуліну, прогнозування глікемії та нагадування про необхідність вимірювань. Для підвищення мотивації користувачів застосунок використовує ігровий підхід – виконання певних завдань винагороджується внутрішніми бонусами. Також передбачена

можливість інтеграції з іншими сервісами для розширення функціоналу. Основним обмеженням є те, що більшість розширених можливостей, включно з детальним аналізом показників і персоналізованими рекомендаціями, доступні лише у преміум-версії.

1.3.3. TidePool

Tidepool – це відкрита платформа для збору, зберігання та аналізу даних, яка допомагає пацієнтам із цукровим діабетом та їхнім лікарям отримувати детальну інформацію про рівень глюкози та ефективність терапії (рис. 1.3).

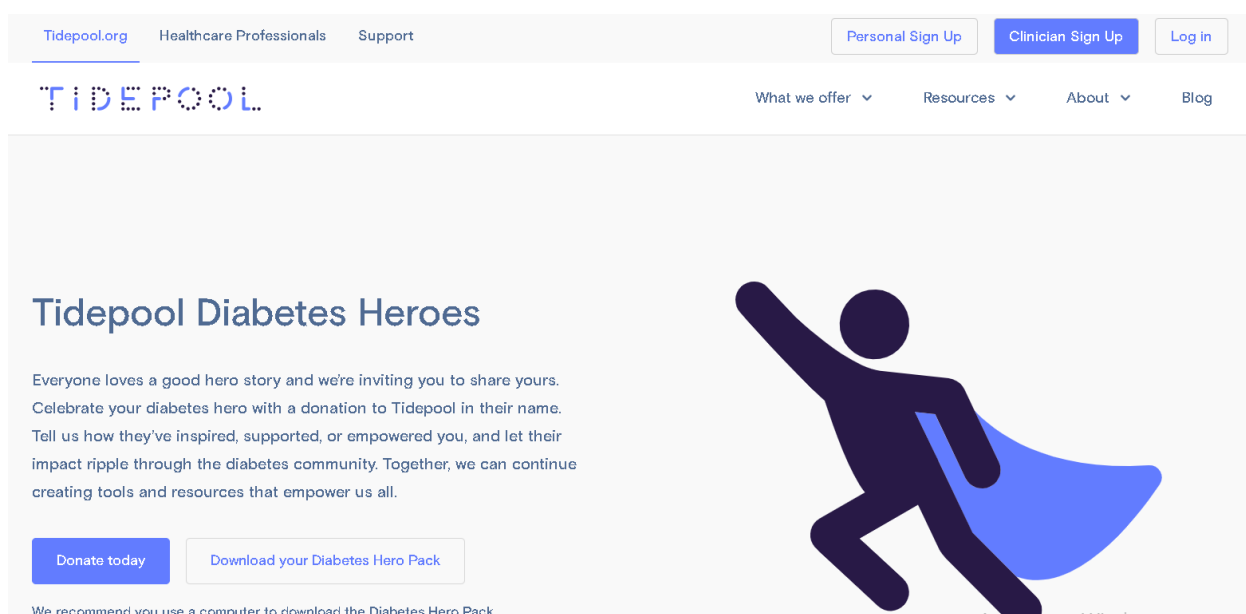


Рисунок 1.3 – Головна сторінка платформи Tidepool

Вона підтримує інтеграцію з різними пристроями для контролю діабету, зокрема глюкометрами, інсуліновими помпами та системами безперервного моніторингу глюкози (CGM), такими як Dexcom, Medtronic та Tandem. Зібрані дані зберігаються у хмарному середовищі та відображаються у вигляді графіків, таблиць і звітів, що допомагає аналізувати зміни рівня глюкози, визначати закономірності та виявляти потенційні ризики. Платформа також дозволяє користувачам вручну додавати додаткову інформацію, таку як прийом їжі, фізична активність та емоційний стан, що

сприяє більш комплексному аналізу. Завдяки відкритому коду та співпраці з медичними компаніями, Tidepool постійно розширює сумісність із новими пристроями. Однак, для ефективного використання платформи необхідне стабільне інтернет-з'єднання, а також сумісність із конкретними медичними пристроями, що може бути обмеженням для деяких користувачів.

1.3.4. Blood Sugar Tracker – Diabetes

Blood Sugar Tracker – Diabetes – це мобільний застосунок для моніторингу рівня глюкози в крові, який допомагає пацієнтам із цукровим діабетом вести щоденник своїх показників і аналізувати зміни у своєму стані (рис. 1.4). Додаток дозволяє користувачам швидко фіксувати рівень цукру, переглядати історію вимірювань у вигляді графіків і таблиць, а також додавати позначки про прийом їжі, інсуліну та фізичну активність.

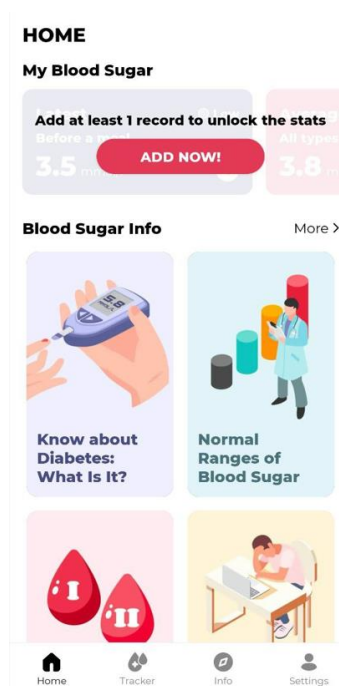


Рисунок 1.4 – Інтерфейс програми Blood Sugar Tracker

Blood Sugar Tracker підтримує ручне введення даних та інтеграцію з деякими пристроями для контролю діабету, що спрощує ведення обліку без використання паперових журналів. Усі внесені показники зберігаються у

мобільному застосунку, що дозволяє відстежувати динаміку змін рівня глюкози та виявляти можливі закономірності.

Однією з ключових переваг застосунок є можливість аналізу показників та надання рекомендацій щодо контролю діабету. Завдяки простому та інтуїтивному інтерфейсу Blood Sugar Tracker підходить як для новачків, так і для досвідчених користувачів. Однак, для повноцінного використання застосунку необхідне регулярне внесення даних вручну, оскільки автоматична синхронізація з медичними пристроями підтримується не для всіх моделей.

Висновки з розділу 1

Було здійснено всебічний аналіз предметної області, що охоплює питання природи, історії розвитку, видів та методів лікування цукрового діабету, з особливим акцентом на інсулінозалежну форму захворювання. Розглянуто сучасні технології лікування, включаючи інсулінотерапію та використання інсулінових помп, а також проаналізовано переваги й недоліки кожного з підходів.

Проведено огляд популярних веб та мобільних застосунків для моніторингу стану здоров'я пацієнтів із діабетом. Виявлено, що більшість наявних рішень мають певні обмеження, зокрема платний доступ до розширених функцій або обов'язкову синхронізацію з медичними пристроями.

РОЗДІЛ 2

ЗАСОБИ РЕАЛІЗАЦІЇ ВЕБ–ЗАСТОСУНКУ

2.1. Мова програмування Python

Python – це інтерпретована, високорівнева, об’єктно–орієнтована мова програмування з динамічною типізацією, яка була створена Гвідо ван Россумом у 1991 році. Вона вирізняється своєю простотою синтаксису, що робить її доступною для розробників різного рівня досвіду, а також потужними можливостями для реалізації складних програмних систем. Однією з основних переваг Python є його читабельність. Код, написаний цією мовою, нагадує псевдокод, що значно полегшує розробку, тестування та підтримку програмних продуктів [36].

Мова програмування має доволі багату стандартну бібліотеку з великою кількістю інструментів для роботи з файлами, базами даних та іншими напрямками. Завдяки цьому Python активно використовується в різних сферах: веб–розробці, аналізі даних, штучному інтелекті, автоматизації систем, наукових дослідженнях тощо. Особливо важливим для мене є те, що Python має велику кількість веб–фреймворків, таких як Django, Flask, FastAPI та інші [32].

Навколо Python сформована велика спільнота, що підтримує тисячі відкритих бібліотек через сховище PyPI. Постійно вдосконалюються існуючі рішення, виправляє помилки та створюються нові бібліотеки з відкритим вихідним кодом для розширення можливостей Python [23].

2.2. Бібліотека Pillow

Pillow – це сучасна бібліотека для роботи з растровими зображеннями в середовищі Python. У веб–застосунку Pillow використовується для обробки

зображень, зокрема – для збереження, зміни розміру або перевірки правильності завантаженого зображення профілю користувача. Ця бібліотека підтримує широкий спектр форматів зображень, включно з JPEG, PNG тощо, що дозволяє гнучко обробляти графічні файли без використання сторонніх сервісів. Pillow інтегрується з Django завдяки підтримці стандартних типів файлів, і його зручно використовувати спільно з моделями, які містять поля типу ImageField [25].

2.3. Бібліотека Plotly

Plotly – це бібліотека для побудови інтерактивних графіків і візуалізацій даних, яка дозволяє створювати графіки без необхідності у складному фронтенді. Plotly використовується для візуального представлення динаміки рівня глюкози в крові, дозувань інсуліну або інших медичних показників. Ще одна особливість бібліотеки – можливість інтерактивної взаємодії з графіками прямо в браузері, з масштабуванням, наведенням на точки даних і панеллю керування. Вона легко інтегрується з Django, оскільки створює HTML код та JavaScript скрипти, які можна вставляти в шаблони [26].

2.4. Бібліотека django-widget-tweaks

Бібліотека django-widget-tweaks розширює можливості стандартної системи шаблонів Django, дозволяючи динамічно змінювати атрибути HTML форм безпосередньо в шаблонах. Це зручно у випадках, коли потрібно застосувати індивідуальні стилі CSS або JavaScript до конкретних полів форми, не змінюючи сам клас форми. django-widget-tweaks підвищує зручність для користувача, дозволяючи швидко змінювати вигляд форм без дублювання коду [13].

2.5. Середовище розробки PyCharm

Для розробки застосунку було обрано інтегроване середовище розробки PyCharm створене компанією JetBrains. PyCharm – це інтегроване середовище розробки Python, що надає великий вибір необхідних інструментів для створення продуктів у різних областях програмування [27]. Для веб–застосунку було використано Pro версію IDE, адже саме для розробки проєктів на Django необхідна саме ця версія, вона надає більший спектр потрібних ресурсів і прискорює розробку самого продукту. Незважаючи на те, що Pro є платною версією, PyCharm надає безкоштовну підписку студентам акредитованих навчальних закладів.

PyCharm пропонує інтелектуальне автодоповнення коду, навігацію по проєкту, статичний аналіз коду тощо. Завдяки цим можливостям середовище дозволяє прискорити процес написання коду, мінімізувати кількість помилок та забезпечити високу якість програмного продукту [19].

Однією з ключових переваг PyCharm є повна підтримка фреймворку Django. IDE надає спеціалізовані шаблони проєктів для Django, автоматично розпізнає конфігураційні файли проєкту, допомагає у створенні моделей, міграцій баз даних, шаблонів і представлень, а також підтримує запуск та налагодження серверів розробки [12].

PyCharm забезпечує зручне управління кодом, пришвидшує процес розробки, допомагає контролювати якість проєкту та ефективно налагоджує застосунок на всіх етапах життєвого циклу.

2.6. Фреймворк Django

Django – це відкритий високорівневий фреймворк для веб–розробки на Python. Фреймворк був створений з метою полегшення розробки складних веб–систем та мінімізації повторення коду, що безпосередньо сприяє підвищенню продуктивності та якості програмного забезпечення [11].

В архітектурному плані Django дотримується моделі MVT (модель–представлення–шаблон), яка дозволяє логічно розділити компоненти системи (рис. 2.1). Модель описує структуру даних і правила взаємодії з базою даних, представлення відповідає за обробку логіки запитів користувача, а шаблон формує зовнішній вигляд сторінки у браузері. Такий поділ сприяє чистоті коду, його повторному використанню та легшому тестуванню [16].

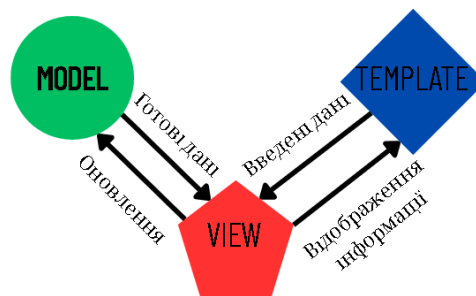


Рисунок 2.1 – Архітектура MVT [16]

Перевага Django у тому, що фреймворк забезпечує широкий набір інтегрованих інструментів. Наприклад, він включає власну ORM систему для роботи з базою даних, систему маршрутизації запитів, шаблонізатор для генерації HTML, а також вбудований адміністративний інтерфейс. Це дозволяє зосередитися безпосередньо на логіці застосунку, не витрачаючи зайвий час на налаштування інфраструктури [1].

Такі популярні веб–сервіси як: Pinterest, Instagram, Spotify використовують Django, що свідчить про його надійність і масштабованість.

2.7. Бази даних Django. SQLite

У процесі розробки веб–застосунку, особливо тих, що передбачають збереження, обробку та виведення користувацьких даних, питання вибору та роботи з базою даних є одним із ключових. Для веб–застосунку, база даних відіграє роль централізованого сховища різних записів.

У фреймворку Django підтримуються різні типи систем управління базами даних, серед яких найбільш поширеними є PostgreSQL, MySQL, MariaDB та SQLite. За замовчуванням у Django використовується легка та вбудована база даних SQLite, яка є оптимальним вибором для невеликих проєктів, локальної розробки тощо [22].

Інтеграція Django з SQLite реалізована через вбудовану систему об'єктно-реляційного відображення. Це означає, що взаємодія з базою даних відбувається не через SQL запити, а через Python класи та методи. В основі побудови структури бази даних у Django лежить оголошення моделей. Кожна модель – це Python клас. Поля класу відповідають колонкам у таблиці бази даних. Після створення моделей Django автоматично генерує SQL таблиці, що дозволяє уникнути ручного проєктування бази даних та дає змогу швидко змінювати її структуру під час розробки [21].

Переваги SQLite полягають у її простоті використання, невеликому розмірі та можливості повноцінного функціонування без потреби в окремому сервері баз даних. Вона зберігає всі дані в одному файлі, що робить процес резервного копіювання, розгортання та перенесення даних надзвичайно зручним. У невеликих проєктах, де не передбачається велике навантаження на базу даних або одночасна робота багатьох користувачів, SQLite демонструє високу ефективність і стабільність [29].

2.8. Шаблони Django

Для сучасного веб-застосунку представлення інформації є чи не найголовнішим аспектом. Веб-інтерфейс повинен бути не лише функціональним, а й зручним та інтуїтивно зрозумілим. Для реалізації користувацького інтерфейсу в Django використовуються веб технології HTML та CSS у поєднанні з власною системою шаблонів.

HTML є стандартною мовою розмітки вебсторінок, яка визначає структуру елементів інтерфейсу: заголовки, форми, таблиці, кнопки тощо.

CSS, у свою чергу, відповідає за зовнішній вигляд цих елементів: кольори, шрифти, відступи, макет сторінки [35]. Використовуючи HTML та CSS, розробник формує вигляд сторінок веб-застосунку таким чином, щоб інтерфейс відповідав вимогам ергономіки та зручності використання для людей із різним рівнем цифрової компетентності [18].

У Django для створення HTML сторінок використовується власна система. Шаблон – це файл з розширенням `.html`, у якому, крім стандартного HTML коду, можуть міститися спеціальні конструкції шаблонізатора Django: змінні у подвійних фігурних дужках, блоки керування, вставки інших шаблонів тощо. Завдяки цьому забезпечується динамічне формування веб-сторінок залежно від контексту [31].

Шаблони працюють у тісному зв'язку з представленнями, які формують логіку взаємодії з користувачем. Представлення передає до шаблону набір даних, які потрібно відобразити. У шаблоні ж ці дані вставляються в HTML за допомогою тегів шаблонізатора. Такий підхід дає змогу розділити логіку та інтерфейс, що відповідає принципу розділення відповідальності та сприяє чистоті коду.

2.9. Захист інформації в Django

Забезпечення інформаційної безпеки є одним із ключових аспектів розробки веб-систем, що працюють із чутливими даними, зокрема медичними. Django, як веб-фреймворк високого рівня, пропонує вбудовані механізми захисту, що дозволяють розробникам реалізовувати безпечні застосунки без потреби вручну впроваджувати низькорівневі перевірки.

Одним із перших рівнів захисту є автоматична нейтралізація спроб впровадження стороннього коду у веб-сторінку. Механізм екранування змінних у шаблонах Django запобігає атакам типу XSS. Усі дані, які передаються у HTML код із серверної частини, проходять фільтрацію, внаслідок чого вставлений JavaScript не виконується, а відображається у

вигляді звичайного тексту. Така поведінка встановлена за замовчуванням і не потребує додаткового втручання з боку розробників [39].

Ще однією критичною загрозою є міжсайтове підроблення запитів – CSRF. Django реалізує ефективний захист від такого виду атак через генерацію унікальних токенів автентичності, які вставляються у кожен форму. Під час надсилання POST запитів сервер перевіряє відповідність токена поточній сесії користувача. У випадку відсутності або фальсифікації токена система автоматично відхиляє запит, що значно ускладнює реалізацію атак такого типу [7].

Значну роль у забезпеченні стійкості до шкідливих SQL запитів відіграє механізм ORM. Замість ручного формування SQL коду розробник працює з абстрактними моделями, які генерують запити у безпечному вигляді. Цей підхід практично усуває можливість SQL ін'єкцій, оскільки всі введені користувачем дані проходять обробку і не можуть змінити структуру запиту [28].

Щодо збереження облікових даних, то Django впроваджує хешування паролів із використанням сучасних криптографічних алгоритмів, зокрема PBKDF2. Таким чином, навіть у разі витоку бази даних зловмисник не отримає доступ до справжніх паролів. Стандартна система автентифікації забезпечує перевірку особи користувача, дозволяючи створювати групи, ролі, обмеження на рівні перегляду або редагування окремих елементів інтерфейсу. У межах веб-застосунку це дозволяє розділяти доступ між пацієнтами й лікарями, забезпечуючи при цьому недоступність персональних медичних записів стороннім особам [5].

Висновок до розділу 2

У другому розділі описано, чому саме було обрано певні технології для створення веб-застосунку. Основною мовою стала Python, оскільки вона досить проста для читання і має багато готових рішень, що добре підходять

для проєктів подібного типу. Для написання коду використовувалося середовище PyCharm, яке добре працює з Django і дозволяє швидко тестувати функціонал.

Фреймворк Django дав змогу логічно розподілити структуру проєкту, що зробило його зручним у підтримці. Також він має власні засоби для роботи з базою даних, що дозволило легко зберігати записи пацієнтів через SQLite. Також була приділена увага на безпеку, адже при роботі з медичними даними це важливо. У Django вже є вбудований базовий захист, який допомагає уникнути типових проблем. Інтерфейс зроблено на HTML і CSS із використанням шаблонів Django. Все виглядає просто і зручно, тож користуватися сайтом легко.

РОЗДІЛ 3

РЕАЛІЗАЦІЯ ВЕБ–ЗАСТОСУНКУ

3.1. Проектування бази даних

Для збереження та обробки даних, що використовуються в системі, створено базу даних з продуманою структурою. Таблиці бази даних описані за допомогою ORM, у вигляді моделей Django (рис. 3.1). Кожна модель описує певний тип сутності у системі: користувача, його щоденник спостережень, записи про вимірювання, коментарі лікаря тощо. Для того, щоб встановити зв'язки між сутностями, були використані зовнішні ключі (ForeignKey).

```
class Measurement(models.Model):
    daily_log = models.ForeignKey(DailyLog, on_delete=models.CASCADE, related_name='measurements')
    period = models.CharField(max_length=20, choices=PERIOD_CHOICES)
    time = models.TimeField()
    glucose_level = models.DecimalField(max_digits=5, decimal_places=2, blank=True, null=True)
    insulin_dose = models.CharField(max_length=50, blank=True)
    description = models.TextField(blank=True)
    activity = models.TextField(blank=True)
```

Рисунок 3.1 – Приклад моделі

База даних веб–застосунку складається з таких таблиць: CustomUser, DailyLog, Measurement, DoctorRequest, PatientProfile, DoctorProfile, DairyComment.

Розглянемо таблицю CustomUser у якій відображенні зареєстровані користувачі з такими полями (табл. 3.1): їхні особисті дані, їхню роль (пацієнт чи лікар) та, у випадку якщо за пацієнтом закріплений лікар, посилення на лікаря.

Таблиця 3.1

Поля таблиці CustomUser

Поле	Тип	Опис
id	Primary Key	Унікальний ідентифікатор користувача
username	CharField	Унікальне ім'я користувача (логін)
email	EmailField	Унікальна електронна пошта
first_name	CharField	Ім'я користувача
last_name	CharField	Прізвище користувача
date_of_birth	CharField	Дата народження користувача
address	CharField	Адреса проживання
city	CharField	Місто проживання
country	CharField	Країна проживання
postal_code	CharField	Поштовий індекс
role	CharField (choices)	Роль користувача (пацієнт/лікар)
assigned_doctor	ForeignKey (self)	Прикріплений лікар (для пацієнта)

Наступною таблицею розглянемо PatientProfile у якій відображена більш детально інформація про пацієнта з полями (табл. 3.2): стать, тип діабету, ріст та вага.

Таблиця 3.2

Поля таблиці PatientProfile

Поле	Тип	Опис
id	PrimaryKey	Унікальний ідентифікатор профілю
user	OneToOneField (CustomUser)	Пов'язаний користувач
gender	CharField	Стать пацієнта
diabetes_type	CharField	Тип діабету
height	PositiveIntegerField	Ріст

weight	DecimalField	Вага
--------	--------------	------

Далі розглянемо таблицю БД DoctorProfile у якій зберігається більш детальна інформація про лікаря, а саме такі поля (табл. 3.3): спеціалізація, місце роботи, освіта, стаж роботи, категорія або науковий ступінь, опис лікаря, фото лікаря, телефон та робоча пошта.

Таблиця 3.3

Поля таблиці DoctorProfile

Поле	Тип	Опис
id	Primary Key	Унікальний ідентифікатор профілю
user	OneToOneField (CustomUser)	Пов'язаний користувач
specialization	CharField	Спеціалізація
workplace	CharField	Місце роботи
education	TextField	Освіта
experience_years	PositiveIntegerField	Стаж роботи
degree	CharField	Категорія або науковий ступінь
bio	TextField	Про себе
photo	ImageField	Фото
phone	CharField	Телефон
email_for_patients	EmailField	Email для пацієнтів

У таблиці DailyLog зберігається інформація про щоденні записи користувача з такими полями (табл. 3.4): дата створення запису та нотатку.

Таблиця 3.4

Поля таблиці DailyLog

Поле	Тип	Опис
id	Primary Key	Унікальний ідентифікатор запису
user	ForeignKey (CustomUser)	Користувач, якому належить запис
date	DateField	Дата запису

note	TextField	Примітка до запису
------	-----------	--------------------

Розглянемо таблицю Measurement. У цій таблиці зберігаються заміри за конкретний запис щоденника. Таблиця має такі поля (табл. 3.5): часовий період вимірювання, час вимірювання, рівень глюкози, доза інсуліну, опис прийому їжі та фізична активність.

Таблиця 3.5

Поля таблиці Measurement

Поле	Тип	Опис
Id	Primary Key	Унікальний ідентифікатор вимірювання
Daily_log	ForeignKey (DailyLog)	Щоденник, до якого належить запис
Period	CharField	Часовий період вимірювання
Time	TimeField	Час вимірювання
Glucose_level	DecimalField	Рівень глюкози
Insulin_dose	CharField	Доза інсуліну
description	TextField	Опис прийому їжі
activity	TextField	Фізична активність

За допомогою таблиці DoctorRequest ми можемо переглядати заявки пацієнтів на прикріплення до лікаря. Поля (табл. 3.6) цієї таблиці містять інформацію про відправника запиту, одержувача, статус запиту і дату створення.

Таблиця 3.6

Поля таблиці DoctorRequest

Поле	Тип	Опис
id	Primary Key	Унікальний ідентифікатор запиту
patient	ForeignKey (CustomUser)	Пацієнт, який створив запит
doctor	ForeignKey (CustomUser)	Лікар, якому надіслано запит
status	CharField	Статус запиту

created_at	DateTimeField	Дата створення запиту
------------	---------------	-----------------------

У таблиці DiaryComment ми можемо переглянути коментарі до щоденників пацієнтів. Таблиця містить поля (табл. 3.7), які зберігають текст, автора та дату створення коментаря.

Таблиця 3.7

Поля таблиці DiaryComment

Поле	Тип	Опис
id	Primary Key	Унікальний ідентифікатор коментаря
log	ForeignKey (DailyLog)	Щоденник, до якого належить коментар
author	ForeignKey (CustomUser)	Автор коментаря
text	TextField	Текст коментаря
created_at	DateTimeField	Дата створення

Усі моделі утворюють цілісну структуру (рис. 3.2), де кожна з них відповідає за певний аспект роботи системи.

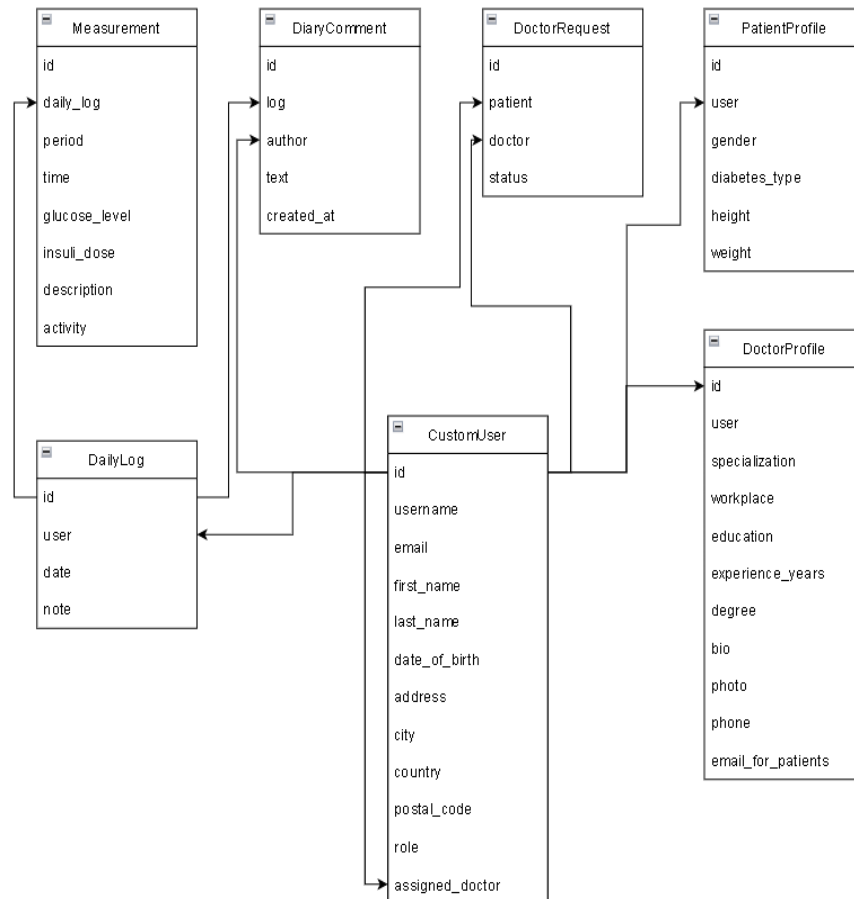


Рис. 3.2 – Діграма бази даних системи

Завдяки чітко визначений зв'язкам і продуманій організації даних, база даних забезпечує надійну основу для зберігання, обробки та взаємодії з інформацією.

3.2. Логіка роботи веб–застосунку

Одним із головних компонентів застосунку є представлення (Django.views). Представлення є елементом, який відповідає за обробку HTTP запитів, що надходять від користувача, та повернення відповідних HTTP відповідей. Представлення виступає логічним посередником між моделлю, яка відповідає за збереження та структуру даних, і шаблоном, що формує зовнішній вигляд інтерфейсу користувача.

Коли користувач заходить на сайт або взаємодіє з якоюсь його частиною, наприклад, натискає кнопку чи заповнює форму, браузер надсилає

спеціальний запит до сервера. У Django ці запити обробляються представленнями. Вони виконують потрібні дії: можуть завантажити дані, зберегти нову інформацію або просто показати відповідну сторінку. Зазвичай використовуються два типи запитів – GET і POST. GET запити потрібні для того, щоб просто отримати дані або відкрити сторінку. POST запити, навпаки, передають дані на сервер, наприклад, коли зберігається форма або створюється новий запис. Представлення аналізує, який саме запит отримано, і діє відповідно.

У веб-застосунку моніторингу стану здоров'я інсулінозалежних пацієнтів всі представлення реалізовані у вигляді функцій. Це дозволяє чітко та зрозуміло описувати логіку кожного процесу в системі. Для обмеження доступу до певних сторінок використовується декоратор `@login_required`, що забезпечує доступ до відповідного представлення лише після автентифікації користувача (рис. 3.3).

```
@login_required
def accept_request_view(request, request_id):
    if request.user.role != 'doctor':
        return redirect('home')

    dr_request = DoctorRequest.objects.get(id=request_id, doctor=request.user)

    dr_request.status = 'accepted'
    dr_request.save()

    patient = dr_request.patient
    patient.assigned_doctor = request.user
    patient.save()

    messages.success(request, "Запит прийнято.")
    return redirect('doctor_requests')
```

Рисунок 3.3 – Приклад представлення у застосунку

Функція `home` відображає головну сторінку застосунку. Вона не містить додаткової логіки, а просто рендерить відповідний HTML шаблон.

Функція `choose_role` відповідає за надання користувачеві можливості вибрати роль перед реєстрацією, зокрема «пацієнт» або «лікар».

Представлення `register` реалізує логіку реєстрації нового користувача з урахуванням обраної ролі. Якщо роль не відповідає дозволеним значенням, здійснюється перенаправлення на сторінку вибору ролі. У випадку, коли запит є POST запитом, перевіряється валідність форми створення користувача. У разі успішної реєстрації користувач автоматично авторизується в системі, після чого перенаправляється на головну сторінку.

Функція `login_view` реалізує механізм входу в обліковий запис. Вона перевіряє правильність введених облікових даних і виконує авторизацію за допомогою стандартного методу `authenticate`. У разі помилки виводиться відповідне повідомлення про помилку.

Функція `profile_view` забезпечує перегляд та редагування особистого профілю зареєстрованого користувача. Якщо запит є POST запитом, збереження даних відбувається через спеціальну форму. У разі успіху виводиться повідомлення про успішне оновлення даних профілю.

Представлення `diary_list_view` відповідає за відображення списку щоденників пацієнта. Відображаються записи у хронологічному порядку, а також доступна форма для вибору дати створення або редагування щоденника.

Функція `edit_log_view` реалізує механізм редагування щоденника за певну дату. Вона дозволяє не лише редагувати основну інформацію, а й вимірювання, а також додавати коментарі. Обробка даних здійснюється за допомогою кількох форм: основної форми щоденника, форми набору вимірювань і форми коментаря.

Функція `delete_log_view` дозволяє видалити запис щоденника за вказану дату. Після видалення користувача перенаправить на загальний список щоденників.

Функція `choose_doctor_view` доступна лише для користувачів із роллю пацієнта. Вона надає можливість переглянути список доступних лікарів та, за

необхідності, створити або переглянути існуючий запит на призначення лікаря.

Функція `send_doctor_request` забезпечує надсилання пацієнтом запиту на призначення конкретного лікаря. Якщо вже існує активний запит (зі статусом «очікується» або «прийнято»), користувач отримує відповідне повідомлення. У разі скасованого раніше запиту система дозволяє повторно його активувати.

Функція `cancel_doctor_request` дозволяє пацієнту скасувати активний запит до лікаря. При цьому статус відповідного запиту змінюється на «скасовано».

Функція `doctor_requests_view` є доступною лише для лікарів і відображає всі запити пацієнтів, які очікують на розгляд. Дані запити можна або прийняти, або відхилити.

Функція `accept_request_view` змінює статус запиту на «прийнято», після чого пацієнт отримує прив'язку до конкретного лікаря через відповідне поле моделі користувача.

Функція `reject_request_view` дозволяє лікарю відхилити запит пацієнта, змінюючи його статус на «відхилено».

Представлення `doctor_patients_view` надає лікарю доступ до переліку усіх закріплених за ним пацієнтів.

Функція `patient_diary_view` надає лікарю змогу переглядати всі щоденники конкретного пацієнта, який закріплений за ним. Перегляд здійснюється з урахуванням хронології та пов'язаної інформації (вимірювання, коментарі тощо).

Функція `health_view` використовується пацієнтом для заповнення анкети стану здоров'я. Окрім основної інформації, система автоматично обчислює вік користувача на основі дати народження.

Функція `patient_log_detail_view` дозволяє лікарю переглянути окремий запис щоденника пацієнта за вказану дату. У цьому представленні також можливо залишити коментар до запису.

Функція `doctor_business_card_view` надає можливість лікарю редагувати свій професійний профіль, завантажити фотографію, вказати опис, контактні дані та іншу релевантну інформацію.

Останньою функцією є `analytics_view`, яка відповідає за формування сторінки аналітики. Тут візуалізуються основні медичні показники пацієнта, такі як рівень глюкози, динаміка показників після прийому їжі, взаємозв'язок між інсуліном і глюкозою. Графіки формуються за допомогою бібліотеки Plotly та передаються у шаблон у вигляді інтерактивних HTML елементів.

Кожна функція виконує чітко окреслену роль у функціонуванні застосунку, забезпечуючи повний життєвий цикл взаємодії користувача з інтерфейсом застосунку.

3.3. Інтерфейс веб–застосунку

3.3.1. Реєстрація та автентифікація

Для роботи веб–застосунку потрібно запустити його у PyCharm та перейти за посиланням, яка з'явиться у терміналі. Після натискання на посилання нас перенаправляє на головну сторінку з можливістю зареєструватися або увійти до системи (рис.3.4).

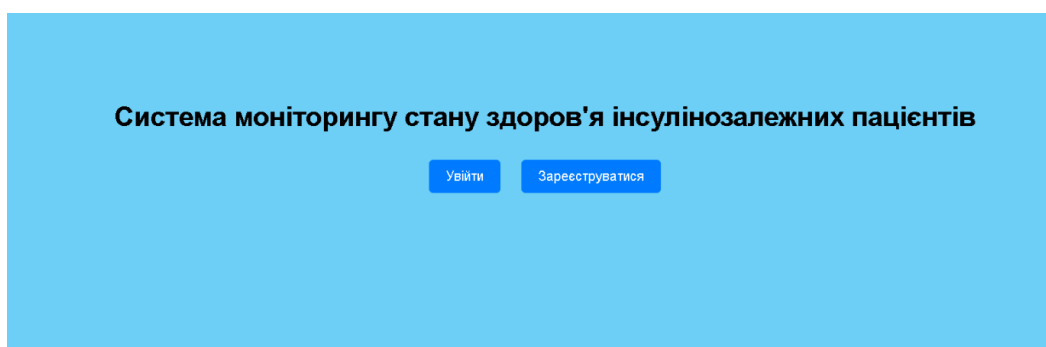


Рисунок 3.4 – Початкова сторінка сайту

Якщо користувач хоче зареєструватися, то при натисканні на «Зареєструватися» його перенаправить на меню вибору, де він зможе обрати роль пацієнта чи лікаря (рис. 3.5).

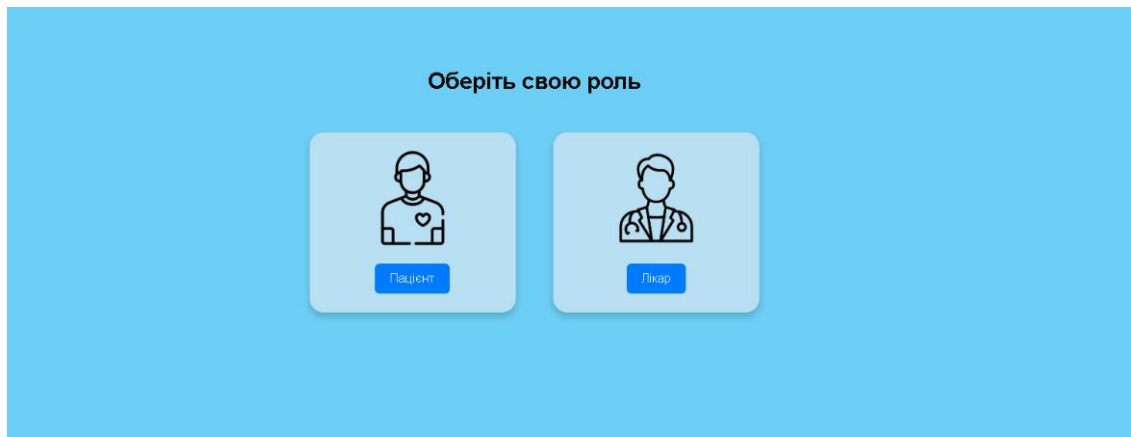


Рисунок 3.5 – Обрання ролі при реєстрації

Після обрання певної ролі користувача перенаправить на форму реєстрації, яка є однаковою як для пацієнта так і для лікаря (рис. 3.6).

The image shows a white registration form titled 'Реєстрація як Лікар' centered on a light blue background. The form contains the following fields from top to bottom: 'Логін' (Login), 'Електронна пошта' (Email), 'Ім'я' (Name), 'Прізвище' (Surname), 'Дата народження' (Date of Birth) with separate dropdowns for day, month, and year, 'Адреса' (Address), 'Місто' (City), 'Країна' (Country), 'Поштовий індекс' (Postal Index), 'Пароль' (Password), and 'Підтвердження паролю' (Password Confirmation). At the bottom of the form is a blue button with the text 'Зареєструватися'.

Рисунок 3.6 – Форма реєстрації

При реєстрації потрібно враховувати певні аспекти: усі поля повинні бути заповнені; не повинні збігатись логін або пошта між користувачами. Якщо якась з цих умов не виконається, то буде показано помилку (рис. 3.7).

Реєстрація як Лікар

Логін

Електронна пошта Заповніть це поле

Ім'я

Прізвище

Дата народження

1

Січень

1900

Адреса

Місто

Країна

Поштовий індекс

Пароль

Підтвердження паролю

Зареєструватися

Рисунок 3.7 – Помилка про необхідність заповнити поле

Як тільки всі умови будуть виконанні та користувач зареєструється, його перенаправить назад на головну сторінку, де потрібно пройти аутентифікацію натиснувши на Увійти. Тут йому потрібно заповнити форму логіну та ввести дані які він вказав при реєстрації (рис. 3.8).

Вхід

Логін

Пароль

Увійти

Рисунок 3.8 – Форма входу на сайт

3.3.2. Вхід як 'пацієнт'

Після успішної аутентифікації пацієнт потрапляє на сторінку свого профілю з інформацією яку він вказав при реєстрації, де він у разі потреби може оновити дані (рис. 3.9). Редагування профілю є зручним, адже значно спрощує процес оформлення власної сторінки та введення правильних

відомостей. Коли пацієнт закінчить вносити зміни, потрібно буде натиснути кнопку «Оновити інформацію», після чого всі внесені виправлення будуть збережені.

Профіль

Логін: artem2025

Поштова адреса: artem2025@gmail.com

Ім'я: Артем

Прізвище: Прокопенко

Дата народження: dd.mm.yyyy

Адреса: пров. Декоратив

Місто: Хмельницький

Країна: Україна

Поштовий індекс: 29001

[Оновити інформацію](#)

Рисунок 3.9 – Профіль пацієнта

У боковому меню при натисканні на «Здоров'я» пацієнт може вказати важливі характеристики, такі як: стать, зріст, вага та тип діабету (рис. 3.10). Пацієнт може в будь-який момент редагувати інформацію про свої характеристики для того щоб змінювати її на актуальну. Для цього потрібно вписати актуальні дані в натиснути на «Зберегти».

Картка здоров'я

Вік: 40 років

Стать: Чоловіча

Тип діабету: Цукровий діабет 1 типу

Ріст (см): 176

Вага (кг): 55,0

[Зберегти зміни](#)

Рисунок 3.10 – Картка здоров'я пацієнта

Пацієнт має можливість вести щоденник діабетика. При натисканні на «Щоденник» у боковому меню пацієнта перенаправляє на сторінку щоденника (рис. 3.11), де пацієнт може створювати записи за необхідний пацієнту день. Це особливо корисно, якщо пацієнт пропустив день запису.

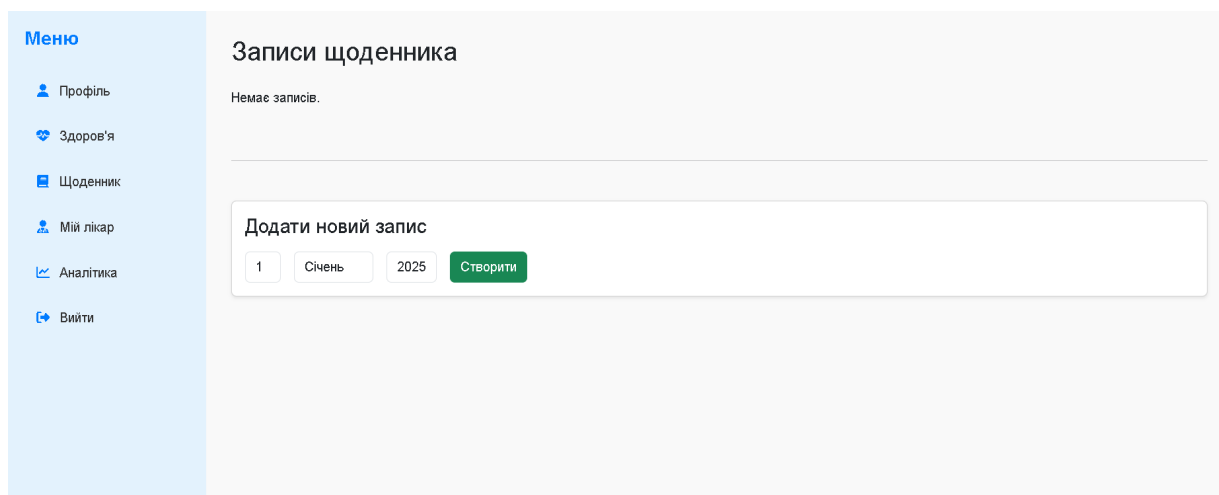


Рисунок 3.11 – Сторінка щоденника

Для того щоб створити перший запис потрібно вказати дату запису і натиснути на кнопку «Створити», після чого з'явиться перший запис (рис. 3.12).

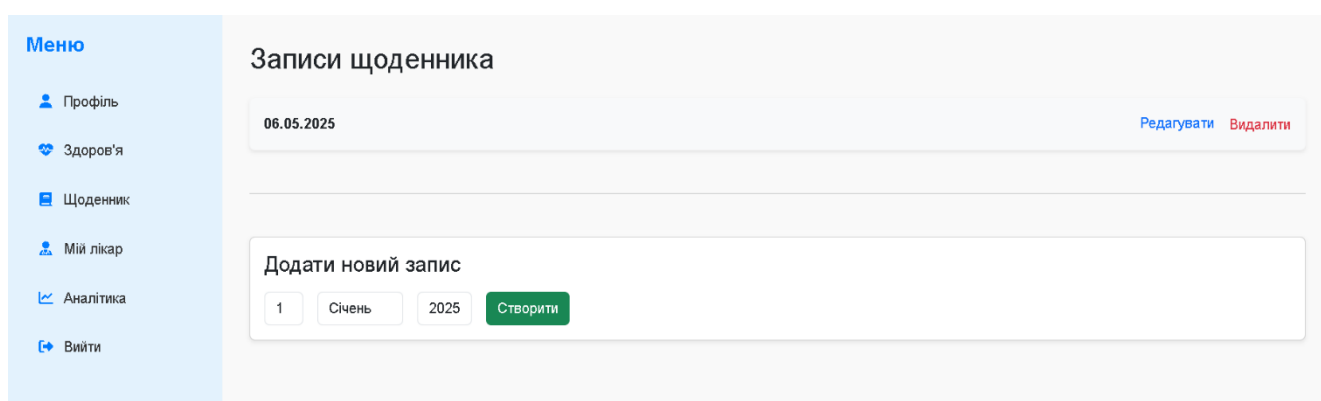


Рисунок 3.12 – Сторінка щоденника зі створеним записом

Щоб вносити заміри за день, потрібно натиснути на надпис «Редагувати» що на записі. Далі пацієнта перекидає на сторінку з замірами за

день (рис. 3.13), де є можливість створити новий замір, заповнити нотатку або додати коментар до запису.

Меню

- Профіль
- Здоров'я
- Щоденник
- Мій лікар
- Аналітика
- Вийти

Запис за 06.05.2025

Додати ще замір

Нотатка:

Зберегти Назад

Коментарі до запису

Коментарів поки що немає.

Залишити коментар...

Додати коментар

Рисунок 3.13 – Сторінка з замірами за день

Він може створити новий замір «Додати замір», де відкривається форма з полями період (до сніданку/після сніданку тощо), час заміру, рівень глюкози в крові, кількість інсуліну (якщо він був введений), опис прийому їжі (якщо був), фізичну активність (якщо була) (рис. 3.14). Усі поля заповнювати не обов'язково.

Меню

- Профіль
- Здоров'я
- Щоденник
- Мій лікар
- Аналітика
- Вийти

Запис за 06.05.2025

Період доби:

Час:

Рівень глюкози (ммоль/л):

Доза інсуліну (од.):

Опис прийому їжі:

Фізична активність:

[Додати ще замір](#)

Рисунок 3.14 – Форма додавання нового заміру

Після того, як пацієнт заповнив форму додавання нового заміру йому потрібно натиснути на «Зберегти», щоб замір відображався на сторінці замірів за день (рис. 3.15). Пацієнт також зможе редагувати заміри у разі необхідності.

Меню

- Профіль
- Здоров'я
- Щоденник
- Мій лікар
- Аналітика
- Вийти

Запис за 06.05.2025

Період доби (Період спостереження)	Час	Рівень глюкози (ммоль/л)	Доза інсуліну (од.)

Опис прийому їжі:

Фізична активність:

[Додати ще замір](#)

Немає записів

Коментарі до запису

Коментарі щодо даного запису:

[Додати коментар](#)

Рисунок 3.15 – Сторінка з наявними замірами

Для того щоб пацієнт комунікував з лікарем, йому потрібно обрати його на сторінці «Мій лікар», яка відкривається з бокового меню. На цій сторінці пацієнту відкривається список з усіх лікарів (рис. 3.16). Він може переглянути інформацію про кожного лікаря, обрати лікаря і надіслати запит на співпрацю.

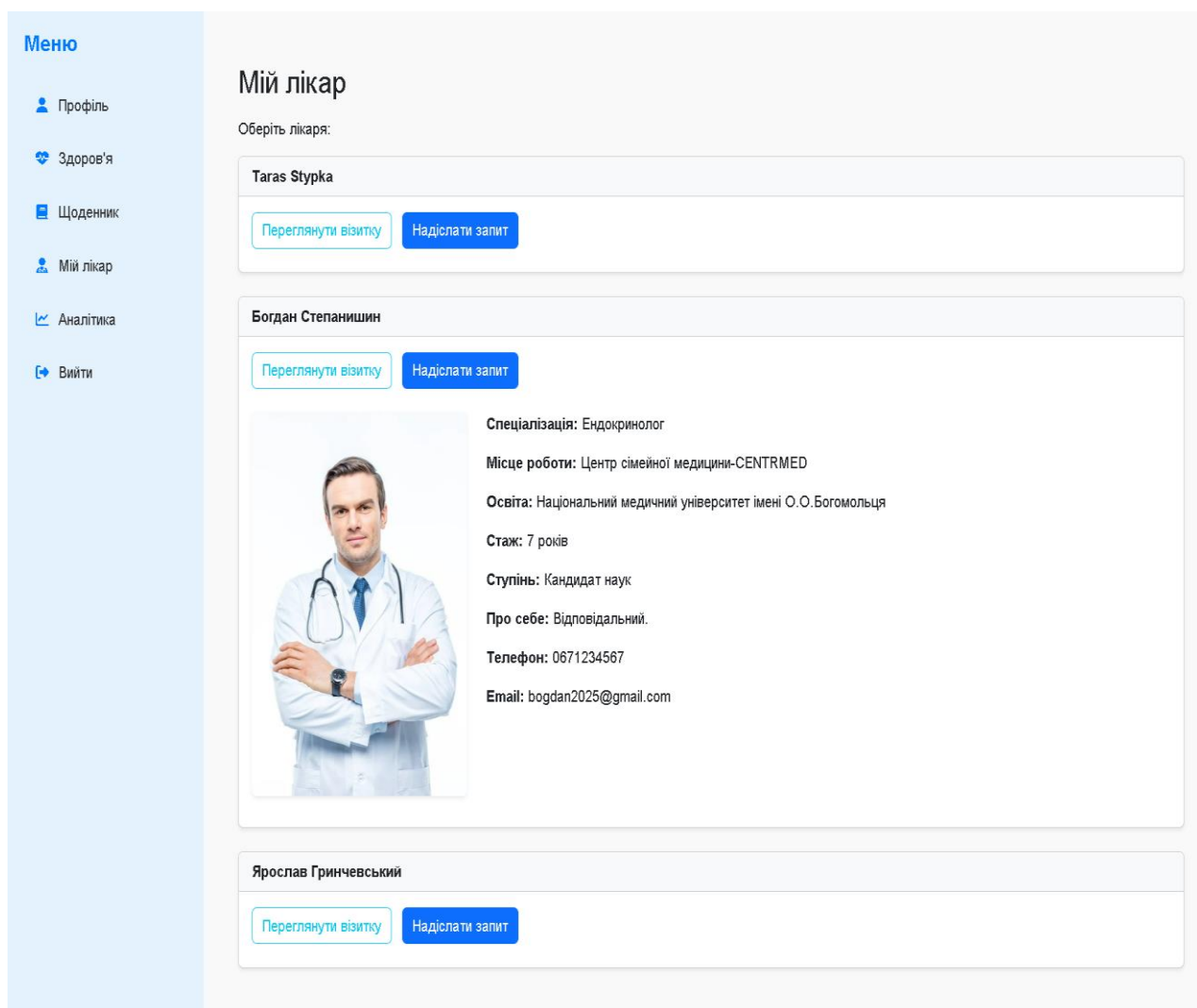


Рисунок 3.16 – Список доступних лікарів

Після того як пацієнт натисни на «Надіслати запит», на сторінці відобразиться інформація про обраного лікаря, статус запиту і можливість скасувати запит у разі вибору не того лікаря (рис. 3.17). Якщо пацієнт відмінить запит, то він зможе надалі обирати лікаря.

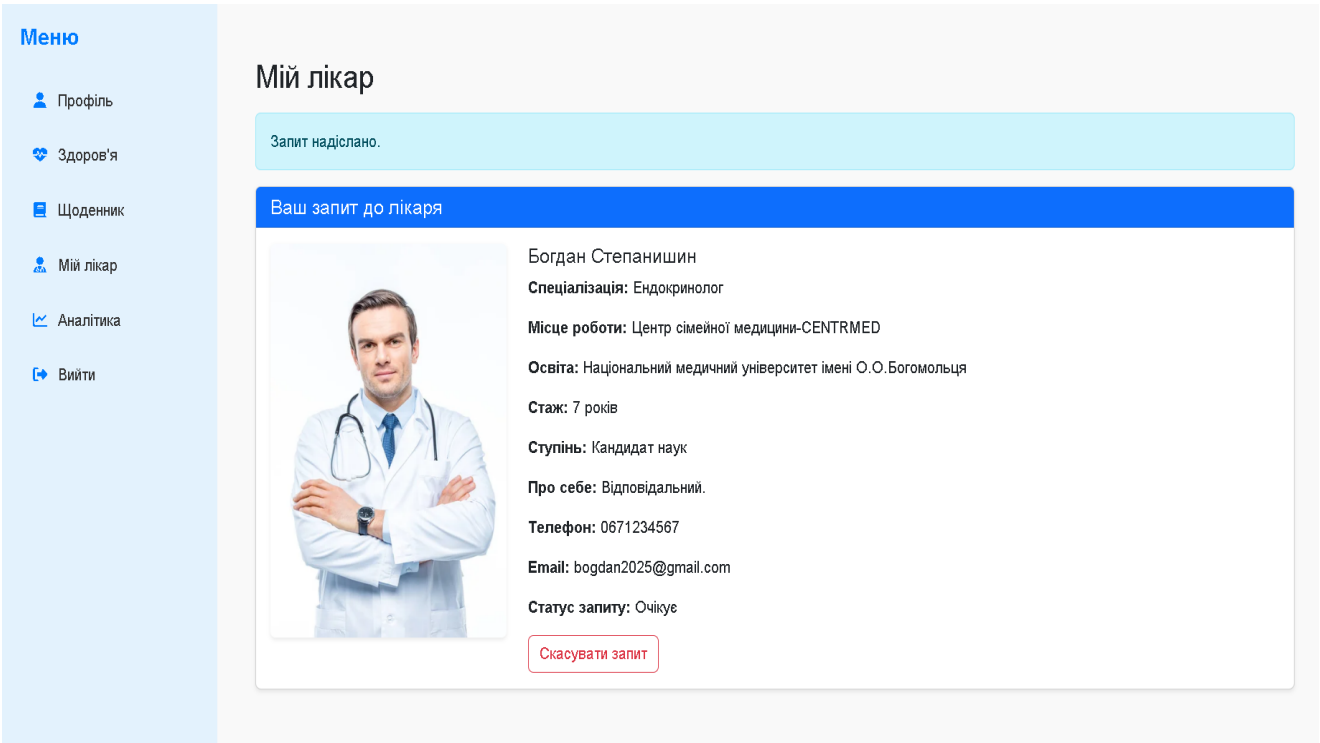


Рисунок 3.17 – Статус запита лікаря

Після того як лікар прийме запит – статус зміниться на «Прийнято» та лікар зможе переглядати та коментувати записи пацієнта.

Також пацієнт може переглянути аналітику щодо своїх записів натиснувши на «Аналітика» на боковому меню. Йому доступно три графіка, які допоможуть краще слідкувати за своїм здоров'ям:

- тренд зміни рівня глюкози за останні 30 днів (рис. 3.18). Завдяки цьому графіку пацієнт зможе вчасно помітити, якщо рівень глюкози стабільно зростає чи зменшується;

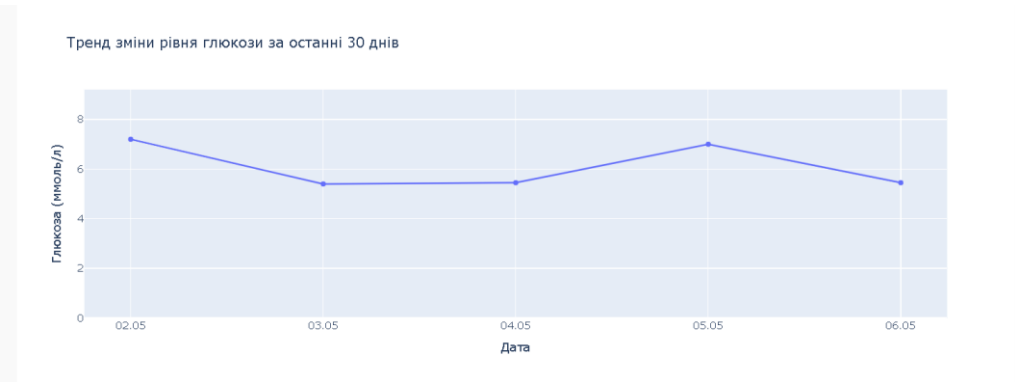


Рисунок 3.18 – Графік зміни глюкози

- порівняння рівня глюкози після інсуліну (рис. 3.19). За допомогою цього графіка пацієнт побачить, чи є зміни в рівні глюкози після інсуліну;

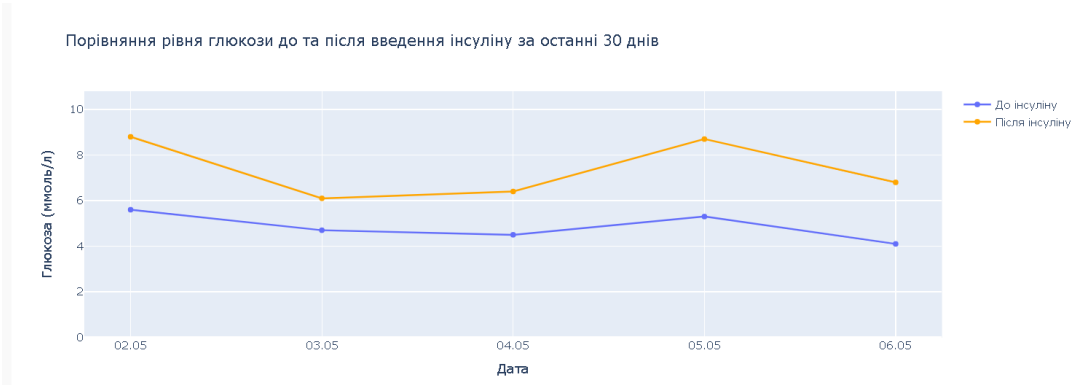


Рисунок 3.19 – Графік порівняння рівня глюкози да та після введення інсуліна

- визначення рівня глюкози після їжі (рис. 3.20). Аналіз рівня глюкози після їжі дасть змогу зрозуміти пацієнту, які продукти викликають найбільше підвищення рівня глюкози.



Рисунок 3.20 – Графік середнього рівня глюкози після їжі

3.3.3. Вхід як «лікар»

Щоб лікарю почати роботу, йому, як і пацієнтові, потрібно пройти авторизацію для входу у систему. Першою сторінкою лікаря є його профіль з

інформацією, яку він надав при реєстрації (рис. 3.21). Тут він може при необхідності редагувати свої дані.

Меню

- Профіль
- Візитка
- Мої пацієнти
- Залити пацієнтів
- Вийти

Профіль

Логін: bogdan2025

Поштова адреса: bogdan2025@gmail.com

Ім'я: Богдан

Прізвище: Степанишин

Дата народження: dd.mm.yyyy

Адреса: Вулична

Місто: Київ

Країна: Україна

Поштовий індекс: 01001

[Оновити інформацію](#)

Рисунок 3.21 – Профіль лікаря

У вкладці «Візитка» бокового меню лікар може вказати більш детальну інформацію про себе, щоб пацієнт міг детальніше ознайомитись з лікарем перед співпрацею. На сторінці візитки лікар може вносити інформацію про свою освіту, місце роботи, свою спеціальність, стаж роботи, науковий ступінь, детальнішу інформацію про себе, номер телефону, робочу пошту та має можливість завантажувати своє фото (рис. 3.22). При необхідності, лікар може редагувати цю інформацію та завантажити нове фото.

Меню

Профіль

Візитка

Мої пацієнти

Запити пацієнтів

Вийти

Візитка

Фото:

Выберите файл

Файл не выбран

Спеціалізація:

Ендокринолог

Місце роботи:

Центр сімейної медицини-CENTRMED

Освіта:

Національний медичний університет імені О.О.Богомольця

Стаж (років):

7

Категорія / Науковий ступінь:

Кандидат наук

Про себе:

Відповідальний.

Телефон:

0671234567

Email для пацієнтів:

bogdan2025@gmail.com

Оновити дані

Рисунок 3.22 – Візитка лікаря

Щоб почати співпрацювати з пацієнтами, потрібно прийняти їхні запити у вкладці «Запити пацієнтів». Після натискання лікаря перекидає на сторінку з списком пацієнтів (рис. 3.23). Він може переглянути інформацію про пацієнтів, прийняти або відхилити запит пацієнта.

Меню

Профіль

Візитка

Мої пацієнти

Запити пацієнтів

Вийти

Запити пацієнтів

Марина Шевченко

Переглянути картку пацієнта

Прийняти запит

Відхилити запит

Стать:

Жіноча

Тип діабету:

Цукровий діабет 2 типу

Ріст:

162 см

Вага:

48,0 кг

Мирослав Бондаренко

Переглянути картку пацієнта

Прийняти запит

Відхилити запит

Рисунок. 3.23 – Запити пацієнтів

Далі у вкладці «Мої пацієнти» лікар може переглянути список своїх пацієнтів (рис. 3.24).

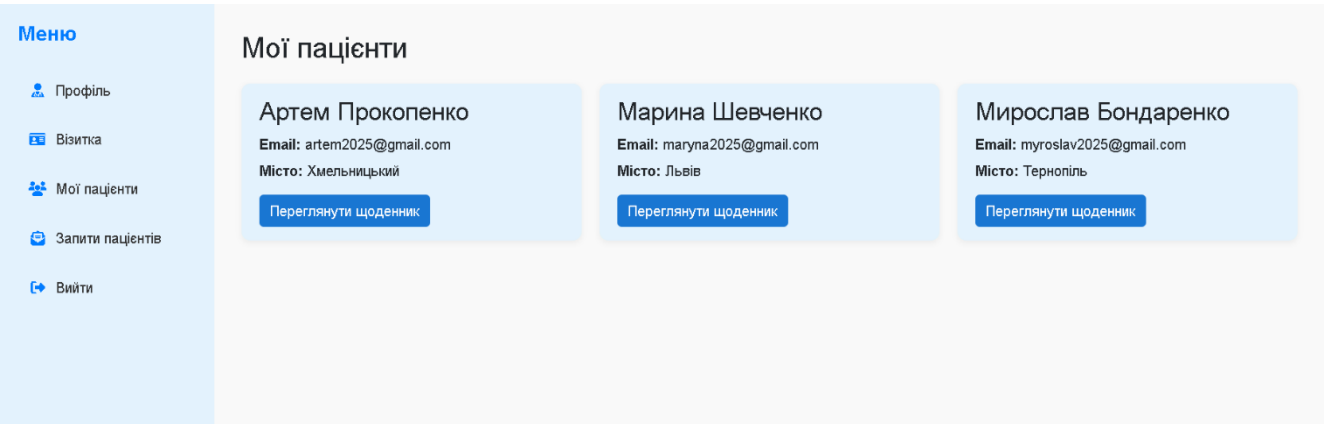


Рисунок 3.24 – Список пацієнтів лікаря

Він може переглядати щоденники кожного пацієнта натиснувши на «Переглянути щоденник». Після цього йому відкриється список з щоденними записами цього пацієнта (рис. 3.25).

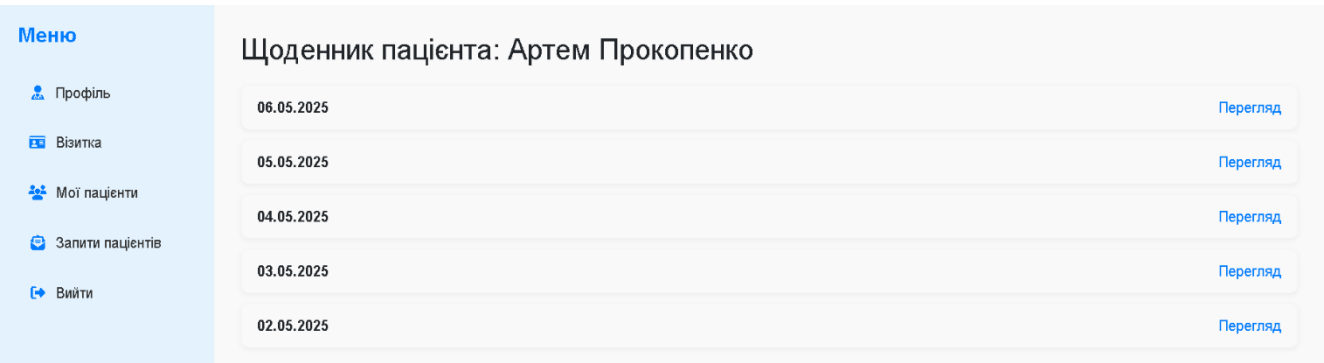


Рисунок 3.25 – Список щоденних записів пацієнта

Лікар може переглядати кожен запис пацієнта. Після того як лікар натискає «Перегляд» він може переглядати заміри пацієнта за обраний день та додавати коментарі за необхідності (рис. 3.26).

Меню

- Профіль
- Візитка
- Мої пацієнти
- Запити пацієнтів
- Вийти

Заміри пацієнта Артем Прокопенко за 06.05.2025

Період: Перед сніданком	Час: 07:12	Глюкоза: 4,10 ммоль/л	Інсулін: 0 од.
Опис: -		Активність: -	

Період: Після сніданку	Час: 09:40	Глюкоза: 6,80 ммоль/л	Інсулін: 1.2 од.
Опис: -		Активність: -	

Коментарі до запису

Богдан Степанишин

06.05.2025 14:53

Все добре. Так тримати!

Артем Прокопенко

06.05.2025 14:54

Дякую. Буду продовжувати.

Додати коментар

Назад до щоденника

Рисунок 3.26 – Заміри пацієнта за обраний день

3.4. Розрахунок економічного ефекту

За темою дипломної роботи виконується функціонально–вартісний аналіз (ФВА) процесу розробки веб–застосунку для моніторингу стану здоров’я інсулінозалежних пацієнтів. Вибір ФВА зумовлений необхідністю обґрунтування техніко–економічної доцільності обраних рішень, визначення ефективності окремих функціональних компонентів системи та оптимального використання ресурсів.

Система повинна складатися з декількох основних підсистем: підсистеми збору даних, підсистеми візуалізації даних та підсистеми користувацького управління. Кожна з них відіграє критичну роль у загальному функціонуванні системи, тому вибір інструментів реалізації для кожної підсистеми має бути технічно і економічно обґрунтованим.

Головною функцією майбутнього веб–застосунку є забезпечення моніторингу стану здоров’я інсулінозалежних пацієнтів, включаючи зручне введення медичних показників, їх збереження та візуалізацію, а також доступ

лікаря до цих даних. Для реалізації цієї мети система повинна відповідати функціональним, технічним та економічним вимогам.

На основі функціонально–вартісного аналізу виділено 4 основні функції веб–застосунку, кожна з яких має декілька можливих варіантів реалізації. Морфологічна карта відображає варіанти виконання функцій (рис. 3.27).

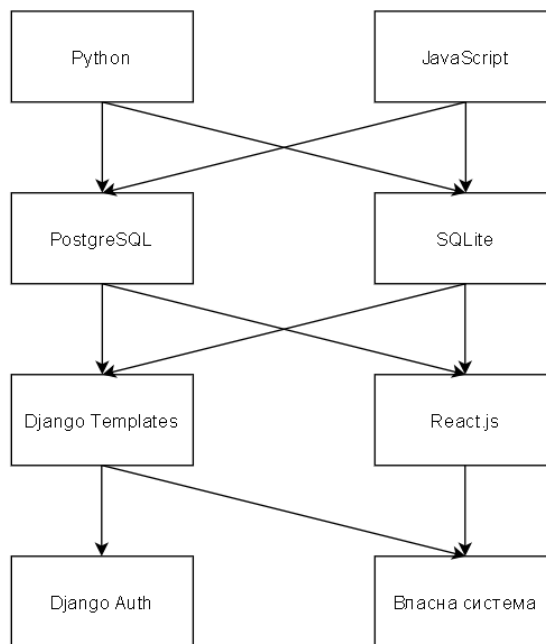


Рис 3.27 – Морфологічна карта

Проведено економічний аналіз трудомісткості, витрат і визначено собівартість проекту. У результаті обґрунтовано вибір найефективнішого варіанту реалізації програмного продукту який має вищий техніко–економічний показник:

- Мова програмування Python
- База даних SQLite
- Інтерфейс Django Templates
- Автентифікація Django

Вартість реалізації веб–застосунку буде коштувати 479 528,438 грн.

Висновок до розділу 3

У цьому розділі розглянуто принципи роботи веб–застосунку. Розроблено структуровану модель бази даних, яка забезпечує цілісність, узгодженість та ефективне зберігання даних у застосунку. Спроектowana логіка веб–застосунку реалізує визначені функціональні вимоги, забезпечує взаємодію між користувачем та серверною частиною, а також гарантує надійну обробку й передачу даних

Було проаналізовано роботу веб–застосунку, проведено детальний розбір кожного складника меню, орієнтований на типи користувачів: пацієнт та лікар. В процесі роботи основний акцент було зроблено на розробку простого та інтуїтивно зрозумілого інтерфейсу, що полегшить навігацію та сприятиме підвищенню ефективності праці.

Зроблено зручну реєстрацію з вибором ролі та авторизацію. Для пацієнта створено комфортний інтерфейс для доступу до персональної медичної інформації, щоденника діабетика та аналітики. Лікарям було надано можливість розпоряджатися пацієнтами, переглядати їхні медичні записи та коментувати записи.

Проведено функціонально–вартісний аналіз розробки веб–застосунку. Розглянуто та обґрунтовано вибір основних функціональних компонентів системи на основі технічних, функціональних і економічних критеріїв.

Також зроблено економічний аналіз трудомісткості, витрат і визначено собівартість проєкту. У результаті обґрунтовано вибір найефективнішого варіанту реалізації програмного продукту, який має вищий техніко–економічний показник.

РОЗДІЛ 4

УЗАГАЛЬНЕННЯ РЕЗУЛЬТАТІВ РОБОТИ

У даному розділі здійснено аналіз досягнутих результатів виконання дипломної роботи. Розглянуто особливості створеного веб–застосунку моніторингу стану здоров'я інсулінозалежних пацієнтів, його відмінності від аналогічних рішень, а також подальші шляхи розвитку програмного продукту.

4.1. Порівняння веб–застосунку з аналогами

Для оцінки ефективності розробленого веб-застосунку було проведено порівняльний аналіз (табл. 4.1) із вже розглянутими рішеннями, призначеними для моніторингу стану здоров'я пацієнтів із цукровим діабетом, а саме: Glooko, mySugr, Tidepool та BloodSugarTracker.

Таблиця 4.1

Порівняння аналогів та їх функцій

Застосунок / Функції	Мій застосунок	Glooko	mySugr	Tidepool	Blood Sugar Tracker
Безкоштовний доступ	Так	Ні	Ні	Так	Так
Для лікаря та пацієнта	Так	Так	Так	Так	Ні
Графіки та звіти	Так	Так	Так	Так	Так
Внесення даних вручну	Так	Так	Так	Так	Так
Фіксація ін'єкцій інсуліну	Так	Так	Так	Так	Так

Фіксація прийомів їжі	Так	Так	Так	Так	Так
-----------------------	-----	-----	-----	-----	-----

Продовження таблиці 4.1

Застосунок / Функції	Мій застосунок	Glooko	mySugr	Tidepool	Blood Sugar Tracker
Фіксація фізичної активності	Так	Так	Так	Так	Так
Коментарі від лікаря	Так	Так	Ні	Так	Ні

Розроблений застосунок найбільше схожий із застосунком Tidepool, адже обидва рішення підтримують роботу як для пацієнта, так і для лікаря, дозволяють вносити дані вручну, створюють графіки, а також мають функцію коментування з боку лікаря. Проте розроблений застосунок відрізняється простотою та доступністю, не потребує синхронізації з медичними пристроями, що робить його зручнішим для користувачів без спеціального обладнання.

Застосунок Glooko надає широкий функціонал із синхронізацією з пристроями та хмарним зберіганням даних, але частина його можливостей доступна лише на платній основі. Аналогічна ситуація і з mySugr, де багато важливих функцій, таких як прогнозування рівня глюкози або персоналізовані звіти, доступні у преміум-версії. Натомість розроблений веб-застосунок не потребує платної підписки і надає користувачам необхідні функції (ведення записів рівня глюкози, фіксацію прийомів їжі, ін'єкцій інсуліну та фізичної активності, можливість отримання коментарів від лікаря) для щоденного моніторингу безкоштовно.

Щодо Blood Sugar Tracker, то аналог має зручний та зрозумілий інтерфейс, дозволяє вручну вводити медичні показники, але він не підтримує взаємодію з лікарями, що є його великим недоліком.

Таким чином, веб–застосунок поєднує в собі ключові переваги сучасних аналогів, але вирізняється відкритістю, функціональністю та орієнтацією на взаємодію між лікарем і пацієнтом.

4.2. Використання веб–застосунку

Розроблений веб-застосунок може бути корисним як для пацієнтів з інсулінозалежним діабетом, так і для лікарів, які закріплені за ними. Його головними перевагами є простота у використанні та доступність. Для роботи не потрібно спеціальне обладнання, достатньо комп'ютера або телефону з доступом до інтернету.

Застосунок призначений для ведення щоденник стану здоров'я, тобто фіксувати рівень глюкози, ін'єкції інсуліну, прийоми їжі та фізичну активність. Усе це можна переглядати у вигляді графіків, а лікар має змогу залишати коментарі до записів. Все це дає змогу підтримувати зв'язок між пацієнтом і лікарем навіть на відстані, що особливо актуально для людей, які живуть у сільській місцевості або не можуть часто відвідувати медичний заклад.

Завдяки зручному інтерфейсу та базовому функціоналу, застосунок підходить як для новачків, так і для більш досвідчених користувачів інтернету. Розробку також можна застосовувати як навчальний інструмент, а саме показувати новим пацієнтам, як правильно вести щоденник і аналізувати свої дані для кращого розуміння свого стану здоров'я та вчасно реагувати на зміни.

4.3. Подальший розвиток веб–застосунку

Незважаючи на те, що розроблений веб-застосунок уже реалізує базову функціональність, необхідну для щоденного контролю стану інсулінозалежних пацієнтів, ще є багато напрямків, у яких застосунок може

бути покращений або функціонал якого, може бути розширений. Серед основних таких напрямків:

- аналіз даних з використанням штучного інтелекту чи машинного навчання для виявлення небезпечних змін, прогнозування гіпоглікемії чи гіперглікемії та формування персональних порад;
- створення чату для більш ефективної комунікації між лікарем та пацієнтом;
- повідомлення про виявлення небезпечних показників та нагадування про внесення замірів за день;
- формування звітів для лікаря та пацієнта з можливістю завантаження на пристрій;
- інтеграція з медичними пристроями (інсулінові помпи) для автоматичного збору показників та більш ефективного моніторингу здоров'я.

Висновок до розділу 4

На основі проведеного аналізу визначено ключові вимоги до веб-застосунку, який розробляється в межах даної роботи. Він має бути безкоштовним, простим у використанні, підтримувати введення даних вручну, створювати графіки, а також забезпечувати можливість коментування записів лікарем. Таким чином, новий застосунок поєднує переваги існуючих аналогів, одночасно усуваючи їхні ключові недоліки.

Розглянуто подальше покращення веб-застосунку, яке передбачає додавання нових функцій, підтримку роботи з медичними пристроями, більш глибокий аналіз показників тощо. Враховуючи досягнуті результати, можна стверджувати, що веб-застосунок має потенціал для впровадження в реальну практику в галузі охорони здоров'я.

ЗАГАЛЬНІ ВИСНОВКИ

Результати роботи

У результаті проробленої роботи було здійснено повний цикл розробки веб–застосунку моніторингу стану здоров'я інсулінозалежних пацієнтів. На етапі аналізу предметної області вивчено основні аспекти інсулінозалежного діабету, розглянуто сучасні методи лікування та надано оцінку існуючим програмним рішенням. Це дозволило сформулювати чіткі вимоги до майбутнього продукту, з урахуванням потреб користувачів і недоліків існуючих застосунків.

Обґрунтовано доцільність використання Python та фреймворку Django, що дозволило створити зручну архітектуру проєкту, реалізувати безпечну роботу з медичними даними та забезпечити стабільну взаємодію з базою даних.

Створено структуру бази даних для збереження показників пацієнтів, розроблено інтерфейси для двох типів користувачів – лікаря та пацієнта. Забезпечено можливість введення, перегляду та коментування медичних даних. Розраховано функціонально–вартісний аналіз створення веб–застосунку моніторингу стану здоров'я інсулінозалежних пацієнтів. Зроблено порівняння вартості розробки та коефіцієнт техніко–економічного рівня. Це допомогло переконатися у правильності вибору необхідних технологій.

Висновки

Було досягнуто мети та вирішенні поставленні завдання. Проведено глибокий аналіз предметної області, визначено основні вимоги до функціоналу майбутнього застосунку. Обґрунтовано вибір технологій, що дозволили створити стабільну, безпечну та зручну у користуванні систему. Реалізовано основні функції, які забезпечують ефективну роботу з медичними даними як для пацієнтів, так і для лікарів. Надалі веб–застосунок

можна покращити додавши автоматичні нагадування, чат між лікарем і пацієнтом, прогнозування показників за допомогою машинного навчання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1) A. Holovaty, J. Kaplan–Moss. *The Django Book: Web development with Django framework*. URL: <http://slav0nic.org.ua/static/books/python/The%20Definitive%20Guide%20to%20Django%20-%20Apress.pdf>
- 2) ADA Professional Practice Committee. Diabetes Technology: Standards of Care in Diabetes–2023. *Diabetes Care*. 2022 Dec;46(Suppl 1):S111–S127. doi:10.2337/dc23–S007. URL: https://diabetesjournals.org/care/article/46/Supplement_1/S111/148041/7-Diabetes-Technology-Standards-of-Care-in
- 3) ADA Professional Practice Committee. Diabetes Technology: Standards of Care in Diabetes–2024. *Diabetes Care*. 2023 Dec;47(Suppl 1):S126–S144. doi:10.2337/dc24–S007. URL: https://diabetesjournals.org/care/article/47/Supplement_1/S126/153939/7-Diabetes-Technology-Standards-of-Care-in
- 4) ADA Professional Practice Committee. Glycemic Targets: Standards of Care in Diabetes–2023. *Diabetes Care*. 2022 Dec;46(Suppl 1):S97–S110. doi:10.2337/dc23–S006. URL: https://diabetesjournals.org/care/article/46/Supplement_1/S97/148053/6-Glycemic-Targets-Standards-of-Care-in-Diabetes
- 5) Authentication | Django documentation. *Django Project*. URL: <https://docs.djangoproject.com/en/5.1/topics/auth/default/>
- 6) Continuous Glucose Monitoring (CGM). *International Diabetes Federation*. URL: <https://idf.org/about-diabetes/continuous-glucose-monitoring/>.
- 7) CSRF | Django documentation. *Django Project*. URL: <https://docs.djangoproject.com/en/5.2/ref/csrf/>
- 8) Diabetes Technology : Standarts of Care in Diabetes – 2024. *Diabetes Care*. URL:

- https://diabetesjournals.org/care/article/47/Supplement_1/S126/153939/7-Diabetes-Technology-Standards-of-Care-in
- 9) Diabetes Technology: Standards of Care in Diabetes–2024. *Diabetes Care*. URL: https://diabetesjournals.org/care/article/47/Supplement_1/S126/153939/7-Diabetes-Technology-Standards-of-Care-in
 - 10) Diabetes. *World Health Organization (WHO)*. URL: <https://www.who.int/news-room/fact-sheets/detail/diabetes>.
 - 11) Django. *Django Project*. URL: <https://www.djangoproject.com>.
 - 12) Django. *JetBrains*. URL: <https://www.jetbrains.com/guide/tags/django/>.
 - 13) django-widget-tweaks. *GitHub*. URL: <https://github.com/jazzband/django-widget-tweaks>
 - 14) Eating Well & Managing Diabetes | ADA. *Diabetes Research, Education, Advocacy | ADA*. URL: <https://diabetes.org/food-nutrition/eating-healthy>.
 - 15) FayeRiley. 100 years of insulin. *Diabetes UK*. URL: <https://www.diabetes.org.uk/our-research/about-our-research/our-impact/discovery-of-insulin#:~:text=Insulin%20was%20discovered%20by%20Sir,than%20a%20year%20or%20two>.
 - 16) GeeksforGeeks. Django Project MVT Structure – GeeksforGeeks. *GeeksforGeeks*. URL: <https://www.geeksforgeeks.org/django-project-mvt-structure/>.
 - 17) Healthy diet. *World Health Organization (WHO)*. URL: <https://www.who.int/news-room/fact-sheets/detail/healthy-diet>.
 - 18) HTML Styles – CSS. *W3Schools*. URL: https://www.w3schools.com/html/html_css.asp
 - 19) JetBrains. PyCharm Features – JetBrains Python IDE. *JetBrains*. URL: <https://www.jetbrains.com/pycharm/features/>.

- 20) LibreTexts. 15.11В: Типи клітин підшлункової залози. *LibreTexts – Ukrayinska*. URL: [https://ukrayinska.libretexts.org/Медицина/Анатомія_і_фізіологія/Книга:_Анатомія_і_фізіологія_\(Boundless\)/15:_Ендокринна_система/15.11:_Підшлункова_залоза/15.11В:_Типи_клітин_підшлункової_залози](https://ukrayinska.libretexts.org/Медицина/Анатомія_і_фізіологія/Книга:_Анатомія_і_фізіологія_(Boundless)/15:_Ендокринна_система/15.11:_Підшлункова_залоза/15.11В:_Типи_клітин_підшлункової_залози).
- 21) Models | Django documentation. *Django Project*. URL: <https://docs.djangoproject.com/en/5.2/topics/db/models/>
- 22) Models and databases | Django documentation. *Django Project*. URL: <https://docs.djangoproject.com/en/stable/topics/db/>.
- 23) Our Community. *Python*. URL: <https://www.python.org/community/>
- 24) Pharmacologic Approaches to Glycemic Treatment: Standards of Care in Diabetes–2024 – PMC. *PMC Home*. URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC10725810/#s1>.
- 25) Pillow. *Pillow*. URL: <https://pillow.readthedocs.io/en/stable/>
- 26) Plotly. *Data Apps for Production | Plotly*. URL: <https://plotly.com/python/>
- 27) Quick start guide | PyCharm. *PyCharm*. URL: <https://www.jetbrains.com/help/pycharm/quick-start-guide.html>.
- 28) SQL–injection | Django documentation. *Django Project*. URL: <https://docs.djangoproject.com/en/5.2/topics/security/#sql-injection-protection>
- 29) SQLite | SQLite documentation. *SQLite*. URL: <https://sqlite.org/docs.html>
- 30) Tauschmann M. et al. Diabetes Technologies: Glucose Monitoring. *Horm Res Paediatr*. 2025. doi:10.1159/000543156.
- 31) Templates | Django documentation. *Django Project*. URL: <https://docs.djangoproject.com/en/5.2/topics/templates/>.
- 32) The Python Standard Library. *Python documentation*. URL: <https://docs.python.org/3/library/index.html>

- 33) UK D. Glycaemic index and diabetes. *Diabetes UK*. URL: <https://www.diabetes.org.uk/living-with-diabetes/eating/carbohydrates-and-diabetes/glycaemic-index-and-diabetes>.
- 34) Ukraine – IDF europe site. *IDF Europe Site*. URL: <https://idf.org/europe/our-network/our-members/ukraine/>
- 35) Web standarts. *W3C*. URL: <https://www.w3.org/standards/>
- 36) What is Python? Executive Summary. *Python*. URL: <https://www.python.org/doc/essays/blurb/>
- 37) WHO guidelines on physical activity and sedentary behaviour. *World Health Organization (WHO)*. URL: <https://www.who.int/publications/i/item/9789240015128>.
- 38) Worldwide trends in diabetes prevalence and treatment from 1990 to 2022: a pooled analysis of 1108 population-representative studies with 141 million participants. *The Lancet*. URL: [https://www.thelancet.com/journals/lancet/article/PIIS0140-6736\(24\)02317-1/fulltext#app-1](https://www.thelancet.com/journals/lancet/article/PIIS0140-6736(24)02317-1/fulltext#app-1)
- 39) XSS | Django documentation. *Django Project*. URL: <https://docs.djangoproject.com/en/5.2/topics/security/#cross-site-scripting-xss-protection>
- 40) Все (і навіть більше) про способи введення інсуліну. *Все про діабет*. URL: <https://diabet.org.ua/vse-i-navit-bilshe-pro-sposoby-vvedennya-insulinu/>
- 41) Державний експертний центр Міністерства охорони здоров'я України. URL: https://www.dec.gov.ua/wp-content/uploads/2023/02/413_28022023_dod.pdf.
- 42) Діабет в Україні. Центр екстреної медичної допомоги та медицини катастроф міста Києва. URL: <http://ssmp.health.kiev.ua/index.php/porady-likaria/550-diabet-v-ukrajini>
- 43) Діабет та інсулін. Що потрібно знати пацієнту. *Здоров'я України | Інформація для спеціалістів охорони здоров'я – Health-ua*. URL:

- <https://health-ua.com/article/36342-dabet-ta-nsuln-sho-potrбно-znati-patcentu>.
- 44) Інсулінова помпа – ефективний і безпечний спосіб лікування цукрового діабету 1 типу. *МЕД ЕК СЕРВІС*. URL: <https://mes.com.ua/ua/a412849-ndtronko-insulinovaya-pompa.html>
- 45) Учасники проєктів Вікімедіа. Історія діабетології – Вікіпедія. *Вікіпедія*. URL: https://uk.wikipedia.org/wiki/Історія_діабетології.
- 46) Цукровий діабет – небезпечна хвороба, підступність якої у тривалому безсимптомному перебігу – Валентина Гінзбург. *Офіційний портал КМДА – Головна*. URL: https://kyivcity.gov.ua/news/tsukroviy_diabet_nebezpechna_khvoroba_pidstupnist_yako_u_trivalomu_bezsimptomnomu_perebigu_valentina_ginzburg/
- 47) Цукровий діабет 2-го типу. *DANN – Diabetologiczna Akademia Novo Nordisk*. URL: <https://www.dann.edu.pl/content/dam/hcpexperience/pl/dann-edu-pl/pl/ua-materialy-edukacyjne-do-pobrania/diabetes%20T2%20%202022-03-16.pdf.coredownload.inline.pdf>
- 48) Цукровий діабет. *Empendium*. URL: <https://empendium.com/ua/chapter/B27.II.13.1>
- 49) Цукровий діабет: як запобігти і як з ним жити | Центр громадського здоров'я. *Центр громадського здоров'я України | МОЗ*. URL: <https://phc.org.ua/news/cukroviy-diabet-yak-zapobigti-i-yak-z-nim-zhiti>.
- 50) Цукровий діабет: як запобігти і як з ним жити | Центр громадського здоров'я. *Центр громадського здоров'я України | МОЗ*. URL: <https://phc.org.ua/news/cukroviy-diabet-yak-zapobigti-i-yak-z-nim-zhiti>.