Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
(повна назва)

Кафедра Автоматизації проектування обчислювальної техніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Підсистема біометричного розпізнавання особистостей в системі безпеки кіберуніверситету

(тема)

Виконав:

студент II курсу, групи СКСм-22-1
Ситнік Н.О.
(прізвище, ініціали)

Спеціальність
123 – Комп'ютерна інженерія
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Спеціалізовані комп'ютерні системи
(повна назва освітньої програми)

Керівник: доц. Ларченко Л.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри АПОТ _____ Чумаченко С.В.
(підпис) (прізвище, ініціали)

2023 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління

Кафедра Автоматизації проектування обчислювальної техніки

Рівень вищої освіти другий (магістерський)

Спеціальність 123 – Комп'ютерна інженерія
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Спеціалізовані комп'ютерні системи
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

"____" _____ 20___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Ситніку Нікіті Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Підсистема біометричного розпізнавання особистостей в системі безпеки кіберуніверситету

затверджена наказом по університету від "_03_"__11__2023 р. № _____1282 Ст_____

2. Термін подання студентом роботи до екзаменаційної комісії 22.01.2024

3. Вихідні дані до роботи

Мікроконтролер ESP8266

Сканер відбитків пальців AS608

Мова програмування C++

Мови програмування PHP, HTML, CSS

Середовища розробки Arduino IDE та Visual Studio Code

4. Перелік питань, що потрібно опрацювати в роботі

1 Аналіз предметної області та постановка завдання

2 Аналіз сучасних біометричних систем

3 Аналіз біометричного розпізнавання в освітній галузі

4. Розробка апаратної реалізації проекту

5. Розробка програмної реалізації проекту

6. Тестування розробленої системи

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 16 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	Дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання	01.09.2023 - 02.09.2023	
2	Аналіз проблемної галузі, постановка завдання	03.09.2023 - 10.09.2023	
3	Вибір інструментальних засобів та розробка структурної схеми проекту	11.09.2023 - 18.09.2023	
4	Розробка програми для мікроконтролера	19.09.2023 - 10.10.2023	
5	Програмна реалізація веб-сторінки	11.11.2023 – 25.11.2023	
6	Тестування розробленої системи	26.11.2023 - 03.12.2023	
7	Оформлення пояснювальної записки	04.12.2023 - 18.12.2023	
8	Оформлення графічного матеріалу	19.12.2023 - 25.12.2023	
9	Перевірка виконаного проекту керівником	26.12.2023 -05.01.2024	
10	Захист роботи	22.01.2024	

Дата видачі завдання 01 вересня 2023 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Ларченко Л.В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи містить: 60 с., 2 табл., 17 рис., 9 джерел посилання.

СИСТЕМА РОЗПІЗНАВАННЯ, БІОМЕТРИЧНА СИСТЕМА, ВІДБИТКИ ПАЛЬЦІВ, СКАНЕР ВІДБИТКІВ, МІКРОКОНТРОЛЕР ESP, СКАНЕР, БАЗА ДАНИХ, WIFI, ПІН

Метою кваліфікаційної роботи є розробка біометричної підсистеми розпізнавання особистостей в системі безпеки кіберуніверситету.

У ході кваліфікаційної роботи було проаналізовано сучасні біометричні системи, розроблено біометричну систему розпізнавання особистостей на основі мікроконтролера, сканера відбитків пальців та бази даних. У ході виконання роботи були обрані компоненти системи з необхідними технічними характеристиками та функціями, створено структурну та функціональну схеми системи. Було розглянуто принцип роботи та процес створення системи розпізнавання, реалізовано необхідні режими роботи системи.

Розроблено програму для мікроконтролера ESP8266 за допомогою мов C/C++ та веб-сторінку мовами PHP, HTML та CSS для відображення та керування базою даних.

Програма може бути використана у системах з обмеженим доступом, таких, як кіберуніверситет, для відстеження відвідуваності та запобігання проникненню сторонніх осіб.

ABSTRACT

The explanatory note contains 60 pp., 2 tables, 17 figures, 9 sources.

RECOGNITION SYSTEM, BIOMETRIC SYSTEM, FINGERPRINT, FINGERPRINT SCANNER, ESP MICROCONTROLLER, SCANNER, DATABASE, WIFI, PIN

The purpose of the qualification work is the development of a biometric subsystem for recognizing individuals in the security system of a cyber university.

In the course of the qualification work, modern biometric systems were analysed, a biometric system for personal recognition was developed based on a microcontroller, a fingerprint scanner and a database. During the execution of the work, the system components with the necessary technical characteristics and functions were selected, and the structural and functional diagrams of the system were created. The principle of operation and the process of creating a recognition system were considered, the necessary modes of operation of the system were implemented.

Developed an ESP8266 microcontroller program using C/C++ languages and a web page using PHP, HTML and CSS languages to display and manage the database.

The program can be used on systems with shared access, such as a cyber university, to increase accessibility and prevent penetration by third parties.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

GND – GROUND – точка нульового потенціалу мікросхеми

UART – Universal Asynchronous Receiver-Transmitter – вузол обчислювальних пристроїв, призначений для організації зв'язку з іншими цифровими пристроями

RX – Receiver – один з каналів зв’язку, що забезпечує прийом даних, які передаються

TX – Transmitter – один з каналів зв’язку, який є джерелом передачі даних

WEB – World Wide Web – всесвітня павутина

SoC – System on chip – електронна схема, яка вміщує функціональні складові цілого пристрою, тобто мікроконтролер, блок пам’яті, стандартні інтерфейси, на одній мікросхемі

SDK – Software Development Kit – набір із засобів розробки, утиліт і документації, який дає програмістам змогу створювати прикладні програми за визначеною технологією або для певної платформи

TTL – Transistor-transistor logic – транзистор-транзисторна логіка, перша широко поширена технологія виготовлення напівпровідникових інтегральних схем

ВСТУП

Технології забезпечують швидкість і зручність для людей і стають важливим інструментом в освітньому процесі.

У моделі змішаного навчання, де навчання віч-на-віч поєднується з технологіями, студенти та викладачі, а також інші зацікавлені сторони використовують комп'ютери та Інтернет для спілкування та співпраці. Роль Інтернету надає можливості для аналізу електронних дій, що виконуються для фіксації закономірностей, тенденцій і розвитку.

При змішаному навчанні основним способом взаємодії між студентами та викладачем є електронні онлайн-системи, тоді як особиста взаємодія є вторинним способом взаємодії. Це може створити багато проблем для ідентифікації особистостей через недостатній контакт між студентами та викладачем.

Біометричні технології є основою широкого спектру високонадійних рішень для ідентифікації та персональної перевірки. Приклади фізіологічних характеристик включають зображення рук або пальців, риси обличчя та розпізнавання райдужної оболонки ока. Поведінкові характеристики – це риси, які визначаються або набуваються динамічною перевіркою підпису, перевіркою мовця та динамікою натискання клавіш є прикладами поведінкових особливостей.

Метою кваліфікаційної роботи є розробка біометричної системи розпізнавання особистостей в системі кіберуніверситету на основі мікроконтролера та сенсора відбитків пальців.

1 СУЧАСНІ БІОМЕТРИЧНІ СИСТЕМИ

У розділі розглянуто поняття біометричної системи, основні відомості про біометричні системи, переваги та недоліки біометричних систем, проаналізовано методи визначення ідентичності особистостей, визначено мету та постановку завдання.

1.1 Біометрична система безпеки, як складова системи безпеки кіберуніверситету

Контроль доступу – це те, що дозволяє авторизованим особам отримати доступ всередину університету та запобігає несанкціонованим особам. Це одна з найважливіших частин безпеки кіберуніверситету.

Університети зазвичай розташовані на величезній території з кількома входами у будівлю. Це створює проблему безпеки та ускладнює керування доступом, особливо за допомогою традиційних дверей і замків. Саме тому біометрична система розпізнавання особистостей є одним із найкращих вирішень цієї проблеми.

Біометрична система, по суті, є системою розпізнавання образів, яка розпізнає особу на основі вектора ознак, отриманого з певної фізіологічної чи поведінкової характеристики, якою володіє особа. Залежно від контексту застосування, біометрична система працює в одному з двох режимів: верифікації або ідентифікації.

У режимі верифікації система перевіряє особу людини шляхом порівняння отриманих біометричних характеристик з біометричним шаблоном особи, який попередньо зберігається в базі даних системи. У такій системі особа, яка бажає бути розпізнаною, заявляє про себе за допомогою персонального ідентифікаційного номера (PIN-коду), імені для входу, смарт-

картки тощо, і система проводить індивідуальний запит – порівняння, щоб визначити, чи твердження правдиве.

Перевірка особи використовується для позитивного розпізнавання, де мета полягає в тому, щоб запобігти використанню декількома людьми ознак однієї особи[1].

У режимі ідентифікації система розпізнає особу, шукаючи відповідність шаблону у всій базі даних. Система проводить порівняння «один-до-багатьох», щоб встановити особу людини (якщо суб'єкт не зареєстровано в базі даних системи – йому буде відмовлено у доступі)[1].

Ідентифікація дає відповідь на питання: «Хто ця людина?», та є критично важливим компонентом програм негативного розпізнавання, у яких система встановлює, чи є особа тим, ким вона, прямо чи явно, демонструє себе. Мета негативного розпізнавання полягає в тому, щоб одна особа не використовувала кілька ідентичностей.

На рис. 1.1 зображено блок-схеми системи верифікації та системи ідентифікації, які виконують завдання реєстрації користувачів. Модуль реєстрації реєструє осіб у базі даних біометричної системи.

Ідентифікація може бути використана для позитивного розпізнавання, що робить це більш зручним, оскільки користувач не зобов'язаний заявляти про свою особу.

Хоча традиційні методи розпізнавання особи, такі як паролі, PIN-коди, ключі та токени, працюють як позитивне розпізнавання, лише біометричні дані можна використовувати для негативного розпізнавання.

На етапі реєстрації біометричний зчитувач (наприклад, сенсор відбитків пальців) спочатку сканує біометричні характеристики особи, щоб створити їх цифрове представлення.

Система виконує перевірку якості, щоб переконатися, що послідовні етапи можуть надійно обробити отриманий зразок з метою полегшення зіставлення, екстрактор ознак обробляє вхідний зразок для створення компактного, але виразного представлення, яке називається шаблоном.

Залежно від програми біометрична система може зберігати шаблон у своїй центральній базі даних або записувати його на смарт-картку, видану особі.

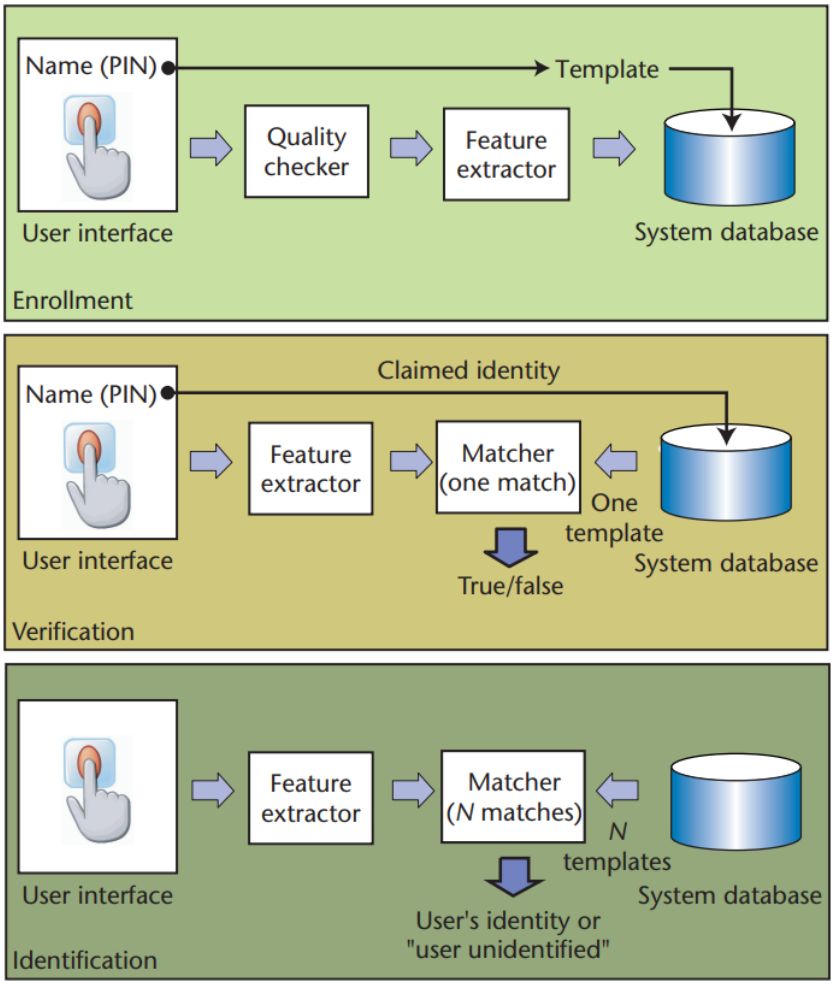


Рисунок 1.1 – Блок-схеми завдань реєстрації, верифікації та ідентифікації

У різних програмах використовується декілька біометричних характеристик. Кожен біометричний показник має свої сильні та слабкі сторони, і вибір залежить від програми.

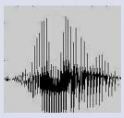
Жоден біометричний показник не може ефективно відповідати вимогам усіх додатків – жоден не є «оптимальним». Залежно від режиму роботи програми та властивостей біометричних характеристик здійснюється підбір певних біометричних даних[2]. Наприклад, як відбиток пальця, так і райдужна оболонка ока є більш точними, ніж метод голосу. Однак у програмі

телебанкінгу голосова техніка може бути кращою, оскільки банк може легко інтегрувати її в існуючу телефонну систему.

У таблиці 1.1 коротко порівнюються п’ять біометричних методів за сімома факторами.

Таблиця 1.1 – Порівняння п’яти біометричних методів за сімома факторами

Table 1. Comparison of several biometric technologies (assessments based on authors' perceptions).

BIOMETRIC	FINGERPRINT	FACE	HAND GEOMETRY	IRIS	VOICE
					
Barriers to universality	Worn ridges; hand or finger impairment	None	Hand impairment	Visual impairment	Speech impairment
Distinctiveness	High	Low	Medium	High	Low
Permanence	High	Medium	Medium	High	Low
Collectibility	Medium	High	High	Medium	Medium
Performance	High	Low	Medium	High	Low
Acceptability	Medium	High	Medium	Low	High
Potential for circumvention	Low	High	Medium	Low	High

1.2 Переваги та недоліки біометричного розпізнавання

Біометрична автентифікація та її використання в сучасних технічних і цифрових програмах має ряд переваг:

- високий рівень безпеки та впевненості – біометрична ідентифікація дає відповіді на питання «щось є у людини» та допомагає підтвердити особу;
- взаємодія з користувачем – зручно та швидко;
- не підлягає передачі – кожен має доступ до унікального набору біометричних даних;
- захист від підробки – біометричні дані важко підробити чи вкрасти;
- високий рівень безпеки та надійності.

Біометрія надає підвищений рівень впевненості в тому, що людина реальна, перевіряючи одну або декілька ознак в реальному світі, чим ідентифікує особу людини. Більшість паролів і PIN-кодів користувачів, особиста ідентифікаційна інформація, ймовірно, були скомпрометовані через витік даних.

Впровадження біометричної автентифікації в процес створює блокпост для шахраїв, який може обійти лише справжній авторизований користувач – хоча шахрай може знати, що людина використовує ім’я свого собаки та деякі щасливі числа для більшості своїх онлайн-акаунтів, він не може використовувати свій відбиток пальця, щоб розблокувати обліковий запис. Крім того, біометричну безпеку можуть забезпечити лише живі люди – на даний момент роботові було б важко пройти сканування райдужної оболонки ока.

Для біометричної автентифікації під час авторизації потрібно ввести дані. Передати чи поділитися фізичною біометрією в цифровому вигляді неможливо, єдиний спосіб використовувати більшість систем біометричної автентифікації – це фізична програма.

Такі біометричні дані, як візерунки обличчя, відбитки пальців, сканування райдужної оболонки ока та інші, майже неможливо відтворити за допомогою сучасних технологій.

Імовірність того, що відбиток пальця користувача точно збігатиметься з чиїмось іншим, становить один із 64 мільярдів. Іншими словами, у користувача є більше шансів виграти в лотерею, ніж мати той самий відбиток пальця, що й хакер, який намагається проникнути у обліковий запис користувача, захищений біометричними даними.

Незважаючи на підвищення безпеки, ефективності та зручності, біометрична автентифікація та її використання в сучасних технічних і цифрових програмах також має недоліки:

- витрати – потрібні значні інвестиції в біометрику для забезпечення безпеки;

- порушення даних – біометричні бази даних все ще можуть бути зламані;
- відстеження та дані. Біометричні пристрої, такі як системи розпізнавання обличчя, можуть обмежити конфіденційність користувачів;
- зміщення – машинне навчання та алгоритми мають бути дуже розвиненими, щоб мінімізувати біометричне демографічне зміщення;
- помилкові спрацьовування та неточність – хибні відмови у доступі або навпаки хибні дозволи доступу все ще можуть виникати, перешкоджаючи вибраним користувачам отримати доступ до систем.

Компанії та уряди, які збирають і зберігають особисті дані користувачів, знаходяться під постійною загрозою з боку хакерів. Оскільки біометричні дані незамінні, організаціям необхідно обробляти конфіденційні біометричні дані з підвищеною безпекою та обережністю – це дорого та технічно складно. Якщо пароль або PIN-код зламано, завжди є можливість змінити його. Те саме не можна сказати про фізіологічні або поведінкові біометричні дані людини.

Більшість поширених методів біометричної автентифікації покладаються на часткову інформацію для автентифікації особи користувача. Наприклад, мобільний біометричний пристрій сканує весь відбиток пальця на етапі реєстрації та перетворює його на дані. Однак майбутня біометрична автентифікація відбитків пальців використовуватиме лише частини відбитків для перевірки особи, тому це буде швидше.

Дослідницька група з Університету Нью-Йорка створила платформу штучного інтелекту, яка змогла шахрайським шляхом зламати автентифікацію за відбитками пальців із показником успішності 20%, порівнюючи схожість часткових відбитків із повними біометричними даними.

1.3 Мета та постановка завдання

Метою кваліфікаційної роботи є розробка біометричної системи розпізнавання особистостей на основі мікроконтролера, сканера відбитків пальців та бази даних.

Згідно з поставленою метою визначено наступні завдання:

- аналіз сучасних біометричних систем;
- аналіз та вибір мікроконтролеру і компонентів, що входять складовими до біометричної системи розпізнавання особистостей;
- розробка структурної схеми проекту;
- розробка функціональної схеми проекту;
- аналіз, вибір мов програмування та середовища розробки мікроконтролера;
- аналіз, вибір мов програмування та середовища розробки WEB-сторінки;
- розробка програмного забезпечення мікроконтролера;
- розробка програмного забезпечення WEB-сторінки;
- створення прототипу WEB-сторінки;
- тестування розробленої біометричної системи розпізнавання особистостей.

2 БІОМЕТРИЧНЕ РОЗПІЗНАВАННЯ ОСОБИСТОСТЕЙ В ОСВІТНІЙ ГАЛУЗІ

У другому розділі розглянуто використання біометричного розпізнавання у різних галузях, переваги та недоліки використання біометричного розпізнавання у освітній сфері.

2.1 Використання біометричного розпізнавання у різних галузях

Історично склалося так, що застосунки, що використовують біометричні дані, ініціювали органи влади для контролю доступу військових, злочинної чи цивільної ідентифікації відповідно до жорстко регульованої правової та технічної бази.

Сьогодні сектори, включаючи освітню сферу, банківську справу, роздрібну торгівлю та мобільну комерцію демонструють велику зацікавленість до переваг біометрії.

Найважливіше те, що за останні роки обізнаність і визнання підвищилися, оскільки мільйони користувачів смартфонів розблоковують свої телефони за допомогою відбитка пальця або обличчя. Біометричні системи потрібні там, де ідентифікація та автентифікація є критичними.

Найбільш типовими галузями використання біометричних технологій є:

- правоохоронні органи та громадська безпека (ідентифікація злочинців/підозрюваних);
- військові (ідентифікація противника/союзника);
- прикордонний, дорожній та міграційний контроль (ідентифікація мандрівника/мігранта/пасажира);
- цивільна ідентифікація (ідентифікація громадянина, резидента, або виборця);

- охорона здоров’я та субсидії (ідентифікація пацієнта, бенефіціара або медичного працівника);
- фізичний і логічний доступ (ідентифікація власника, користувача, працівника/підрядника або партнера);
- комерційні програми (ідентифікація споживача/клієнта).

Біометрія для правоохоронних органів стосується використання біометричних систем, які підтримують функціонування системи безпеки правоохоронних закладів .

До цієї категорії можна віднести засоби ідентифікації відбитків пальців, та долонь злочинців. Вони зберігають, шукають і отримують зображення відбитків пальців і записи у справі.

Сьогодні автоматизовані системи біометричної ідентифікації (ABIS) можуть створювати та зберігати біометричну інформацію, яка відповідає біометричним шаблонам для обличчя за допомогою, так званих, систем Mugshot, пальця та райдужної оболонки ока.

Розпізнавання обличчя в реальному часі – можливість ідентифікації обличчя в режимі реального часу або після події. Розпізнавання також набуває інтересу для громадської безпеки у містах, аеропортах, на кордонах або в інших чутливих місцях, таких як стадіони[2].

Багато чого невідомо про те, як оборонні відомства в усьому світі використовують біометричні дані. Справа в тому, що інформацію важко отримати та поширити, оскільки вона не є публічною. Армії багатьох розвинених країн збирають дані обличчя, райдужну оболонку очей, відбитки пальців і дані ДНК у біометричну систему ідентифікації.

Наприклад, біометрична програма Сполучених Штатів Америки почала діяти в 2004 році і спочатку збирала відбитки пальців. Агентство оборонної криміналістики та біометрії (DFBA) керує системою, відомою як Автоматизована біометрична інформаційна система Міністерства оборони. За даними OneZero 7,4 мільйона ідентифікацій у базі даних переважна більшість походить від військових операцій в Іраку та Афганістані.

Електронний паспорт – звичний біометричний проїзний документ. Друге покоління таких документів, відоме як біометричні паспорти, містить два збережених відбитки пальців і паспортну фотографію.

Біометрія надає незаперечні докази зв'язку між паспортом і його власником. Біометрична автентифікація здійснюється шляхом порівняння обличчя/відбитків пальців, побачених/прочитаних на кордоні, з обличчям/відбитками пальців у мікроконтролері паспорта. Якщо обидва біометричні дані збігаються, автентифікація підтверджується.

Ідентифікація, у разі необхідності, проводиться за допомогою біографічних даних у чіпі та роздруковується. Крім того, багато країн створили біометричні інфраструктури для контролю міграційних потоків на свою територію та зі своєї території. Сканери відбитків пальців і камери на прикордонних постах фіксують інформацію, яка допомагає точніше ідентифікувати мандрівників, які в'їжджають до країни.

Національні посвідчення особи широко поширені в країнах Європи та Близького Сходу або Африки для ідентифікації і програми медичного страхування. Ці біометричні ідентифікаційні картки та відбитки пальців використовуються для підтвердження особи власника перед зверненням до державних послуг або медичної допомоги.

Лікарні, аптеки та клініки використовують картки медичного страхування для перевірки прав на соціальне забезпечення, захищаючи при цьому конфіденційність персональних даних.

Бази даних AFIS (Автоматизована система ідентифікації за відбитками пальців), часто пов'язані з базою даних реєстру цивільного стану, гарантують ідентичність громадян та унікальність для решти населення надійним, швидким та автоматизованим способом. Вони можуть поєднувати цифрові відбитки пальців, фотографії та сканування райдужної оболонки для більшої надійності[2].

В якості прикладу можна назвати індійську систему біометричної реєстрації Aadhaar – це найпоширеніша система біометричної ідентифікації

та автентифікації в Індії. Номер Aadhaar – це 12-значний унікальний ідентифікаційний номер, який надається всім жителям Індії. Номер базується на їх біографічних і біометричних даних: фотографія, десять відбитків пальців, два скани райдужної оболонки ока. Aadhaar надає ідентичність кожному індійцю, що робить багато послуг доступнішими для людей. Це зменшило: корупцію, вартість надання комунальних послуг.

2.2 Можливості, переваги та недоліки застосування біометричного розпізнавання в освітній сфері

Біометрична технологія ідентифікує особу на основі фізичних характеристик, таких як відбитки пальців, райдужна оболонка очей, структура обличчя, голос і навіть жести. Серед усіх інших біометричних даних розпізнавання відбитків пальців є найпростішим у реалізації та найбільш економічно ефективним методом контролю доступу. Завдяки своїм можливостям і додаткам біометрична технологія стала корисним інструментом для викладачів, які прагнуть забезпечити академічну доброчесність.

Компанії та підприємства впроваджують біометричні системи не лише для ідентифікації своїх співробітників, але й для безпеки, контролю доступу, керування персональними пристроями та відстеження часу.

Біометрія привнесла великі зміни в освітньому секторі, оскільки цей сектор розглядається як крок до сталого розвитку, який сприяє економічному зростанню. Біометрична технологія з часом змінила спосіб роботи системи ідентифікації для учнів дошкільних закладів до шкіл вищого стандарту, коледжів, навчальних закладів і професійних курсів. Багато шкіл уже застосували біометричні дані для різних цілей, включаючи системи відвідуваності, подачу їжі в кафетерії, оцінку контролю та безпеки, щоб оптимізувати повсякденну роботу. Названа система успішно працює в країні Індія.

Існують різні програми та заходи, які виконуються одночасно в школі, наприклад, відвідування, доступ, транспорт, розподіл тощо. Ведення записів про участь учнів у різних програмах та заходах є важливим завданням. Безпека стала невід'ємною частиною діяльності школи.

Біометричні технології в основному використовуються в школах у системах керування ідентифікацією, відвідуванням уроків, електронним оцінюванням, безпекою та аналітикою навчання. Завдяки різноманітності додатків використання біометричної ідентифікації в навчальних закладах обов'язково матиме багато переваг.

Точна відвідуваність. Методи біометричної ідентифікації дає викладачам точне уявлення про присутність студентів у аудиторії. Це сприяє щирості та дисциплінованості серед студентів, оскільки вони не можуть обійти біометричний ідентифікаційний пристрій і повинні бути фізично присутні, щоб відвідування було зараховане.

Безпека студентів. Співробітники можуть бути впевнені, що студенти перебувають в університеті, оскільки біометрична перевірка не дає місця для фальшивої відмітки відвідуваності. Більше того, використання біометричної ідентифікації для відвідувачів може гарантувати, що лише знайомі відвідувачі матимуть доступ до студентів.

Ефективні операції – біометричні дані спрощують багато адміністративних процесів, які відбуваються в життєвому циклі студента. Ці процеси забирають багато часу викладачів, тому біометрична ідентифікація може допомогти забезпечити швидкий і легкий контроль доступу.

Перевірка екзаменаційних аудиторій. Поширені порушення під час іспитів – використання підроблених посвідчень особи та видавання себе за інших студентів. Біометрична ідентифікація та перевірка на вході в екзаменаційну аудиторію можуть запобігти таким порушенням і підтримати довіру до іспитів.

Витрати – впровадження біометричних систем відвідування вимагає значних інвестицій. Налагодження такої передової технології та навчання управлінню системами може бути виснажливим і вимагає великої відданості.

Інфраструктура. Біометрична ідентифікація потребує біометричних зчитувачів та розумного програмного забезпечення. Повинна існувати відповідна інфраструктура для встановлення біометричних зчитувачів біля входу та поза класними кімнатами.

Не можна заперечувати, що біометрична технологія є проривом з точки зору безпеки. Запровадження нових технологій може бути порівняно дорогим, але в довгостроковій перспективі має більше переваг.

3 АПАРАТНА РЕАЛІЗАЦІЯ ПРОЕКТУ

У розділі описано обрані компоненти для системи біометричного розпізнавання, їх технічні характеристики, розроблено структурну та функціональну схеми проекту.

3.1 Технічні характеристики компонентів системи біометричного розпізнавання

Для розробки прототипу системи біометричного розпізнавання особистостей було використано наступні компоненти:

- мікроконтролер ESP8266 (NodeMCU);
- оптичний сканер відбитків пальців AS608.

3.1.1 Мікроконтролер ESP8266

NodeMCU (Node MicroController Unit) – це середовище розробки програмного та апаратного забезпечення з відкритим кодом, створене на основі недорогої системи на кристалі (SoC) під назвою ESP8266[3].

ESP8266, розроблений і виготовлений Espressif Systems, містить ключові елементи комп’ютера: центральний процесор, оперативну пам’ять, мережу (Wi-Fi) і навіть сучасну операційну систему та SDK, що робить його чудовим вибором для будь-яких проектів Інтернету речей (IoT)[3].

Зовнішній вигляд та опис пінів ESP8266 (NodeMCU) приведено на рис. 3.1.

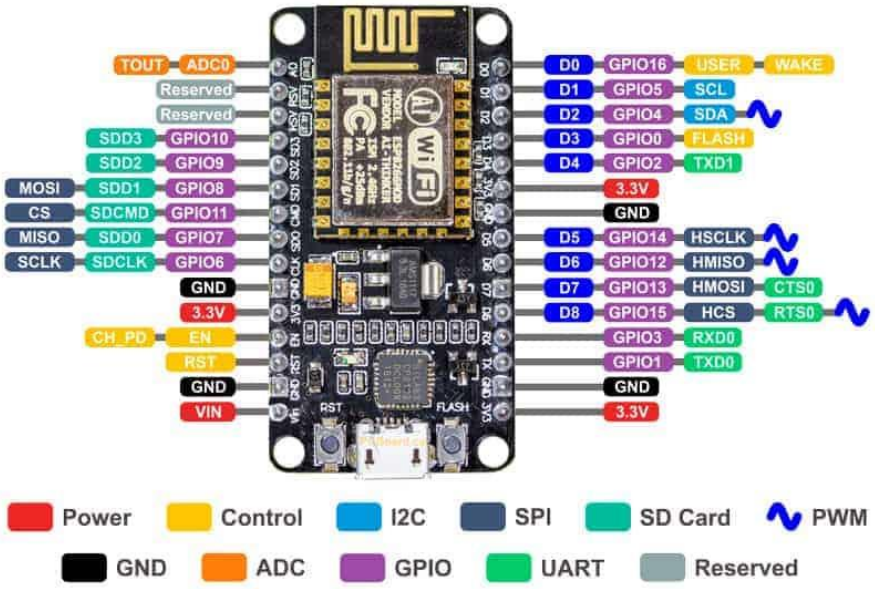


Рисунок 3.1 – Зовнішній вигляд та опис пінів ESP8266 (NodeMCU)

Технічні характеристики мікроконтролера ESP8266 (NodeMCU) наведено у таблиці 3.1.

Таблиця 3.1 – Технічні характеристики мікроконтролера ESP8266 (NodeMCU)

Categories	Items	Parameters
Wi-Fi	Certification	Wi-Fi Alliance
	Protocols	802.11 b/g/n (HT20)
	Frequency Range	2.4 GHz ~ 2.5 GHz (2400 MHz ~ 2483.5 MHz)
	TX Power	802.11 b: +20 dBm
		802.11 g: +17 dBm
		802.11 n: +14 dBm
	Rx Sensitivity	802.11 b: –91 dbm (11 Mbps)
		802.11 g: –75 dbm (54 Mbps)
		802.11 n: –72 dbm (MCS7)
	Antenna	PCB Trace, External, IPEX Connector, Ceramic Chip
Hardware	CPU	Tensilica L106 32-bit processor
	Peripheral Interface	UART/SDIO/SPI/I2C/I2S/IR Remote Control
		GPIO/ADC/PWM/LED Light & Button
	Operating Voltage	2.5 V ~ 3.6 V
	Operating Current	Average value: 80 mA
	Operating Temperature Range	–40 °C ~ 125 °C
	Package Size	QFN32-pin (5 mm x 5 mm)
	External Interface	-
Software	Wi-Fi Mode	Station/SoftAP/SoftAP+Station
	Security	WPA/WPA2
	Encryption	WEP/TKIP/AES
	Firmware Upgrade	UART Download / OTA (via network)
	Software Development	Supports Cloud Server Development / Firmware and SDK for fast on-chip programming
	Network Protocols	IPv4, TCP/UDP/HTTP
	User Configuration	AT Instruction Set, Cloud Server, Android/iOS App

3.1.2 Оптичний сканер відбитків пальців AS608

AS608 – це модуль сканера та зчитувача відбитків пальців. Модуль має можливість реєстрації відбитків пальців, обробки зображення, перевірки збігу відбитків та багато іншого. Він обробляє дані та надсилає оброблені дані на мікроконтролер через послідовний порт. Пристрій використовує чіп DSP, який виконує візуалізацію зображень, пошук функцій, обчислення та пошук[4].

Модуль має вбудовану флеш-пам’ять, яка зберігає дані відбитків пальців і реєструє нові – можна зберегти до 162 відбитків пальців. Він взаємодіє з контролером або будь-якою іншою системою з послідовним TTL і надсилає пакети даних для фотографування, виявляє друк, хешує та виконує пошук. Пристрій має червону та зелену світлодіодну індикацію неправильного та правильного друку[4].

Технічні характеристики оптичного сканера відбитків пальців AS608:

- діапазон напруги живлення: від 3,6 В до 6 В;
- максимальний робочий струм: 120мА;
- піковий струм: 150 мА;
- максимальний час друку зображення: 1 с;
- рівень помилкових прийомів (FAR): <0,001%;
- частота помилкових відхилень (FRR): <1,0%;
- інтерфейс: послідовний UART або TTL;
- ємність пам'яті: 162 відбитків пальців;
- файл підпису: 256 байт;
- файл шаблону: 512 байт;
- швидкість передачі за замовчуванням: 57600;
- площа вікна: 14мм х 18мм;
- робоча температура: від -20 °C до 50 °C;
- робоча вологість: 40% - 85%.

Зовнішній вигляд сенсора AS608 приведено на рис. 3.2.

Рисунок 3.2 – Зовнішній вигляд сенсора AS608

3.2 Структурна та функціональна схеми з’єднання

У ході розробки курсового проекту було реалізовано та зображено структурну та функціональну схеми.

Структурна схема (рис. 3.3) демонструє способи зв’язку між компонентами, а функціональна демонструє з’єднання на фізичному рівні. Компонентами структури є сканер відбитків пальців, мікроконтролер ESP8266, база даних, яка містить дві таблиці з назвами “users” та “users_logs” у яких зберігаються дані про зареєстрованих користувачів та про відвідування цих зареєстрованих користувачів відповідно, та web-сторінка, яка відображає дані з таблиць “users” та “users_logs”.

Після реєстрації користувачів у системі, сканер AS608 працює у режимі сканування відбитків на вхід або вихід користувачів з університету.

Якщо відбиток пальця користувача є у системі, то при його першому скануванні система записує до бази даних, що користувач увійшов у будівлю. При повторному скануванні відбитка пальця цього ж користувача система записує до бази даних, що цей користувач залишив університет.

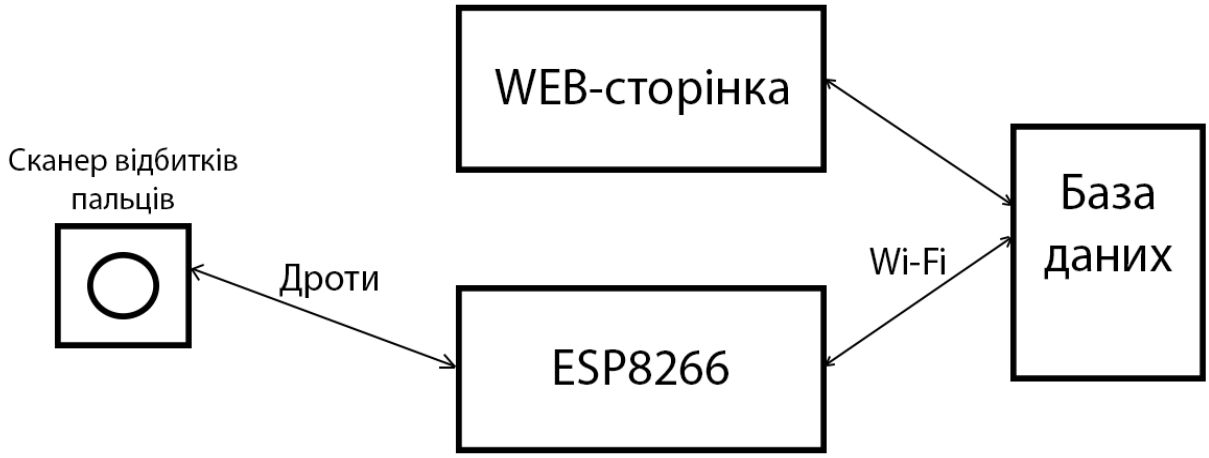


Рисунок 3.3 – Структурна схема проекту

3.3 Схеми з`єднання компонентів

Розглянемо схему підключення складових проекту більш детально.

Для підключення сенсора відбитків пальців AS608 потрібно чотири дроти. Для коректного підключення сканера до мікроконтролера необхідно знати призначення пінів на сенсорі (рис. 3.4).

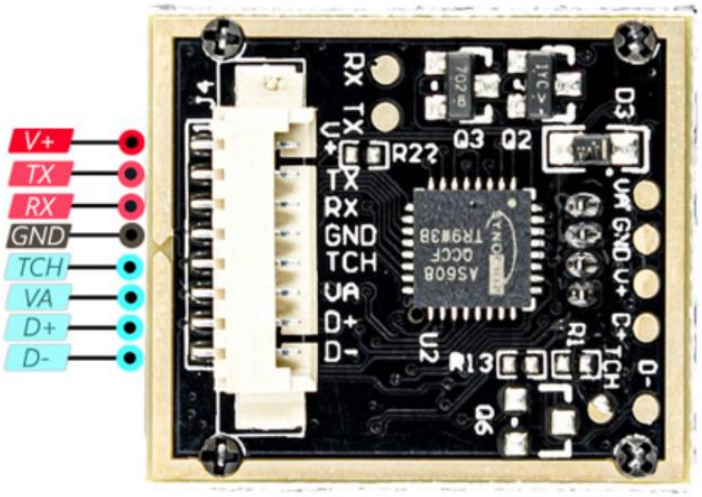


Рисунок 3.4 – Піни підключення сканера відбитку пальця

Підключаємо піни V+ та GND сенсору AS608 до пінів 3V та GND на мікроконтролері, вони відповідають за живлення сенсору відбитків пальців від мікроконтролера, який, у свою чергу живиться від USB-порту комп’ютера.

Далі піни TX та RX на сенсорі підключаємо до пінів D5 та D6 відповідно, вони відповідають за обмін даними між сенсором AS608 та мікроконтролером, який, у свою чергу, має можливість обмінюватися даними з базою даних та надсилати до неї запити за допомогою Wi-Fi мережі.

З’єднання бази даних з мікроконтролером здійснюється з використанням радіочастоти.

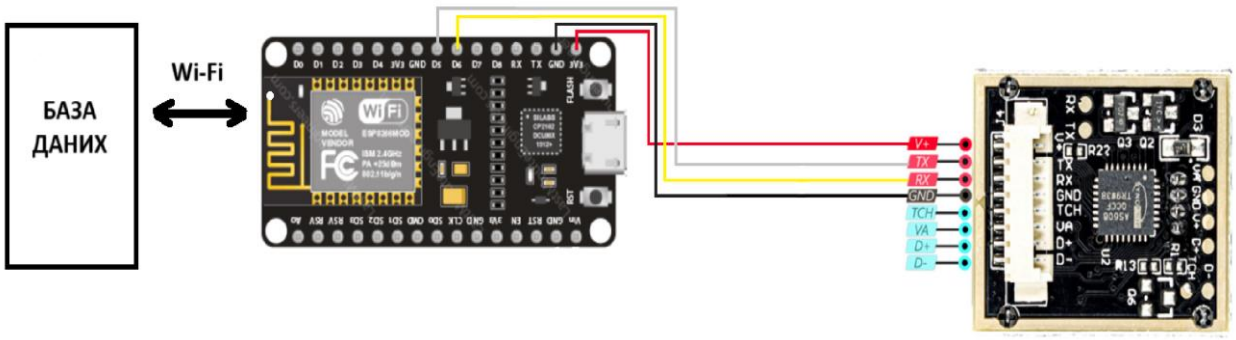


Рисунок 3.5 – Функціональна схема проекту

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОЕКТУ

У розділі приведено обґрунтування вибору мов програмування, опис середовища розробки програми та опис розробленого коду зі скріншотами роботи програми.

4.1 Вибір мови програмування та середовища розробки для мікроконтролера ESP8266(NodeMCU)

Для написання коду для мікроконтролера ESP8266(NodeMCU) існує декілька основних середовищ програмування:

– MicroPython;
– Arduino IDE;
– NodeMCU.

4.1.1 Середовище розробки MicroPython

MicroPython є однією з платформ програмування для мікроконтролерів ESP8266. Для цього використовується мова програмування Python 3, яка включає невеликий піднабір стандартної бібліотеки python і оптимізована для роботи на мікроконтролерах ESP.

Плата ESP8266 – це мікроконтролер з підтримкою Wi-Fi, який запускає MicroPython на чистому металі, що дає вам низькорівневу операційну систему Python, яку можна використовувати для керування всіма видами електронних проектів[5].

MicroPython підтримує такі розширені функції, як інтерактивна підказка, закриття, генератори, обробка винятків і багато іншого.

Дана операційна система працюватиме лише в межах 256 КБ кодового простору та 16 КБ оперативної пам’яті.

4.1.2 Середовище розробки Arduino IDE

Arduino IDE являє собою інтегроване середовище розробки, що включає текстовий редактор, призначений для написання коду, область повідомлень, текстову консоль, панель інструментів із кнопками для загальних функцій і набір меню. Arduino IDE є потужним і простим у програмуванні. Arduino поставляється з власним програмним забезпеченням, це програмне забезпечення працює з операційними системами Windows, Linux і Mac. Перевагою середовища розробки Arduino є можливість використання мов програмування C/C++.

Проект Arduino IDE розпочався в 2003 році як програма для студентів Інституту дизайну взаємодії Ivrea в Івреа, Італія, з метою надати новачкам і професіоналам недорогий і простий спосіб створення пристроїв, які взаємодіють із середовищем за допомогою датчиків і приводів. Типовими прикладами таких пристроїв, призначених для любителів-початківців, є прості роботи, термостати та детектори руху.

4.1.3 Середовище розробки NodeMCU

NodeMCU – це мікропрограмне забезпечення на базі Lua для ESP32 і ESP8266 Wi-Fi SOC від Espressif і використовує файлову систему SPIFFS на основі флеш-пам’яті. Спочатку прошивка була розроблена як супровідний проект до популярних модулів розробки NodeMCU на базі ESP8266, але тепер проект підтримується спільнотою, і мікропрограму можна запускати на будь-якому модулі ESP[6].

4.1.4 Обґрунтування вибору мови програмування для мікроконтролера

Для проекту було обрано середовище розробки Arduino IDE та мови програмування C/C++. Такий вибір було зроблено, опираючись на попередній досвід роботи з мовами програмування C/C++. Також порівнюючи вищеописані мови програмування було зроблено висновок, що

мови C/C++ забезпечують невеликий розмір проекту, який буде завантажений у мікроконтролер, та більшу швидкість роботи програми. Також Arduino IDE має більш інтуїтивно зрозумілий інтерфейс, гнучке та просте підключення бібліотек та просте налаштування проекту.

4.2 Вибір мови програмування та середовища розробки для WEB-сторінки

Для написання коду для WEB-сторінок існує декілька основних середовищ програмування:

– Visual Studio Code;
– IntelliJ IDEA;
– Eclipse.

4.2.1 Visual Studio Code

Visual Studio Code є одним з найпопулярніших та найбільш широко використовуваних середовищ розробки на сьогодні. Це безкоштовне інтегроване середовище розробки, яке підтримує багато мов програмування, включаючи JavaScript, TypeScript, Python та багато інших. Він має розширення, що дозволяють розширити його функціональність, що забезпечує широкі можливості для розробки.

4.2.2 IntelliJ IDEA

IntelliJ IDEA є іншим дуже популярним середовищем розробки, особливо для Java-розробки. Воно підтримує багато мов програмування, включаючи Java, Kotlin, Groovy, Scala та багато інших. IntelliJ IDEA має потужну систему рефакторингу, що дозволяє легко змінювати код та покращувати його.

4.2.3 Eclipse

Eclipse є одним з найбільш популярних середовищ розробки для Java-розробки. Воно підтримує багато мов програмування, таких як C++, Python, PHP та інших. Eclipse має широкий набір інструментів для аналізу коду та підтримує системи контролю версій, такі як Git.

4.2.4 Обґрунтування вибору мови програмування для WEB-сторінки

Для проекту було обрано середовище розробки Visual Studio Code та мови програмування PHP та HTML/CSS. Такий вибір було зроблено, опираючись на попередній досвід роботи з мовою програмування PHP.

HTML є стандартизованою мовою гіпертекстової розмітки документів для перегляду веб-сторінок у браузері.

CSS – формальна мова декорування та опису зовнішнього вигляду документа (веб-сторінки), написаного з використанням мови розмітки (HTML)[7].

Таким чином HTML/CSS – це два основних компоненти, які необхідні для створення веб-сторінок.

4.3 Налаштування середовища розробки Arduino IDE та драйверів

Мікроконтролер ESP8266 має вбудований програматор, тому для коректної роботи з мікроконтролером необхідно інсталювати драйвер CH340 на комп’ютер, з якого буде виконуватись прошивка мікроконтролера. Завдяки вбудованому програматору для підключення комп’ютера до ESP8266 неохідний лише кабель з роз’ємами USB та micro-USB, але деякі кабелі micro-USB використовуються лише для живлення, без можливості передачі даних, тому необхідно завчасно перевірити спроможність кабелю передавати дані, інакше буде неможливо завантажити код до мікроконтролера.

Для початку роботи з Arduino IDE необхідно запустити середовище

розробки і вказати відповідний json-пакет, який необхідний для коректного налаштування плати розробки. Для цього після запуску середовища необхідно відкрити пункт меню Файл/Налаштування і вставити посилання (https://arduino.esp8266.com/stable/package_esp8266com_index.json) у поле “Додаткові посилання для менеджера плат:” і натиснути кнопку “ОК”.

Після цього у пункті Інструменти/Плата/Менеджер плат обрати “esp32” та встановити останню версію пакету. Надалі у панелі Інструменти/Плата/ESP8266 Плати обираємо “NodeMCU 1.0 (ESP-12E)”. Після цих дій можна починати написання коду та роботу з мікроконтролером.

4.4 Підключення необхідних бібліотек в Arduino IDE

До початку написання коду та роботи з проектом необхідно визначити основні бібліотеки для роботи з мікроконтролером: ESP8266WiFi, SoftwareSerial, ESP8266HTTPClient. Ці бібліотеки встановлюються автоматично разом з завантаженням пакету “esp8266” у менеджері плат. Далі була підключена бібліотека Adafruit_Fingerprint, яка для середовища розробки є сторонньою, тому для її підключення потрібно скористатися пунктами меню Інструменти/Керувати бібліотеками, після чого у відкритому вікні менеджера бібліотек, увівши назву відповідної бібліотеки, є можливість завантажити будь-яку доступну версію цієї бібліотеки. Або, якщо на комп’ютері вже є у наявності потрібна бібліотека у форматі ZIP архіва, то можна відкривши у меню пункт Скетч/Підключити бібліотеку/Додати .ZIP бібліотеку та обравши шлях до неї на локальному пристрої.

Для використання бібліотеки у написанні коду мовами C/C++ їх необхідно підключити у файлі основного коду “fingerprint_system.ino” за допомогою директиви препроцесора #include:

```
#include < ESP8266WiFi.h>
```

```
#include < SoftwareSerial.h>
#include < ESP8266HTTPClient.h>
#include " Adafruit_Fingerprint.h"
```

ESP8266WiFi.h – надає широкий набір методів (функцій) і властивостей C++ для налаштування та роботи модуля ESP8266 у режимі станції та/або програмної точки доступу.

Бібліотека SoftwareSerial.h забезпечує послідовний зв'язок із цифровим контактом, відмінним від послідовного порту. Можна мати кілька програмних послідовних портів зі швидкістю до 115200 біт/с.

WiFiClientSecure.h – Бібліотека для легкого виконання запитів HTTP GET, POST і PUT до веб-сервера.

Працює з будь-яким класом, отриманим від клієнта, тому перемикання між Ethernet, WiFi і GSMClient вимагає мінімальних змін коду.

Adafruit_Fingerprint.h – ця бібліотека дозволяє використовувати датчик відбитків пальців Adafruit на будь-якому UART для отримання, зберігання, отримання та запиту відбитків пальців та підходить для додавання біологічної безпеки до наступної збірки.

4.5 Написання програмного коду мовами C/C++

Код мікроконтролера написаний у одному файлі вихідного коду “fingerprint_system.ino” (Додаток Б).

Після підключення усіх необхідних бібліотек директивою #include, оголошуються макроси Finger_Rx та Finger_Tx, які за допомогою директиви препроцесора замінює у подальшому коді ім’я макросу на його значення. Цю директиву використовують як заміну константним оголошенням змінних, бо макрос, за рахунок того, що просто замінює ім’я на значення, не займає пам’яті контролера. Значення цих макросів використовуються як номери пінів, через які мікроконтролер та сканер відбитків пальців будуть передавати дані між собою:

```
#define Finger_Rx 14
#define Finger_Tx 12
```

Після цього створюється екземпляр класу SoftwareSerial з назвою mySerial та конструктором у який, як параметри, передаються раніше створені макроси Finger_Rx та Finger_Tx. Рядком нижче створюється екземпляр класу Adafruit_Fingerprint з назвою finger і екземпляром mySerial, який передається як параметр:

```
SoftwareSerial mySerial(Finger_Rx, Finger_Tx);
Adafruit_Fingerprint finger =
Adafruit_Fingerprint(&mySerial);
```

Надалі оголошуються константні покажчики на строки ssid та password, які зберігають назву WiFi мережі та пароль до неї відповідно:

```
const char* ssid = " UPC2820077";
const char* password = " rrxrmVyj4fh6";
```

Наступними створюються строкові змінні postData, link та apiKeyValue, що містять дані, які будуть передаватися на сервер, посилання, по якому ці дані будуть передаватися та ключ доступу, за яким ці дані будуть захищені. Нижче оголошені змінні FingerID та id, які зберігають унікальні ідентифікатори відбитків пальців для реєстрації у системі та для відвідування відповідно:

```
String postData;
String link =
"http://192.168.0.38/biometric_data/post_data.php";
String apiKeyValue = "tPmAT5Ab3j7F9";
int FingerID = 0;
uint8_t id;
```

Функція setup – це стандартна функція Arduino IDE. Вона призначена для початкового налаштування контролера та ініціалізації необхідних

об’єктів. У ній відбувається виклик функції connectToWiFi(), яка здійснює налаштування та підключення до Wi-Fi мережі (лістинг 4.1).

Лістинг 4.1 – Код функції connectToWiFi()

```
void connectToWiFi()
{
  WiFi.mode(WIFI_OFF);
  delay(1000);
  WiFi.mode(WIFI_STA);
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("Connected");

  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());
}
```

Також у функції setup() перевіряється коректність підключення сенсору відбитка пальця та додавання нових відбитків(лістинг 4.2).

Лістинг 4.2 – Код функції setup()

```
void setup()
{
  Serial.begin(115200);
  delay(2000);
  connectToWiFi();
  finger.begin(57600);
  Serial.println("\n\nAdafruit finger detect test");
  if (finger.verifyPassword()) {
    Serial.println("Found fingerprint sensor!");
  } else {
    Serial.println("Did not find fingerprint sensor :(");
    while (1) { delay(1); }
  }
  for(int i = 0; i < 3; ++i){
```

```
    Enroll();
    delay(1000);
  }
}
```

Додавання нових відбитків відбувається за допомогою функції Enroll(), фрагмент якої представлено далі:

```
  while(Serial.available() == 0){}
  id = Serial.parseInt();
  if (id == 0) {
    Serial.println("id cannot be 0");
    return;
  }
  Serial.print("Enrolling ID #");
  Serial.println(id); Serial.println();
getFingerprintEnroll();
```

Ця функція приймає номер id, який користувач вводить з клавіатури та викликає функцію getFingerprintEnroll(Додаток Б).

Функція getFingerprintEnroll() сканує відбиток пальця та зберігає його до пам’яті сканера (лістинг 4.3).

Лістинг 4.3 – Фрагмент коду функції getFingerprintEnroll ()

```
Serial.print("Creating model for #");  Serial.println(id);

  p = finger.createModel();
  if (p == FINGERPRINT_OK) {
    Serial.println("Prints matched!");
  } else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    Serial.println("Communication error");
    return p;
  } else if (p == FINGERPRINT_ENROLLMISMATCH) {
    Serial.println("Fingerprints did not match");
    return p;
  } else {
    Serial.println("Unknown error");
    return p;
  }
  Serial.print("ID "); Serial.println(id);
  p = finger.storeModel(id);
  if (p == FINGERPRINT_OK) {
    Serial.println("Stored!");
```

```
    confirmAdding();
    return p;
  }
  else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    Serial.println("Communication error");
    return p;
  }
  else if (p == FINGERPRINT_BADLOCATION) {
    Serial.println("Could not store in that location");
    return p;
  }
  else if (p == FINGERPRINT_FLASHERR) {
    Serial.println("Error writing to flash");
    return p;
  }
  else {
    Serial.println("Unknown error");
    return p;
}
```

Після збереження відбувається виклик функції confirmAdding(), яка з'єднується з базою даних, та передає захисний ключ, id щойно відсканованого відбитка, ім'я та стать, які користувач ввів за допомогою клавіатури (лістинг 4.4).

Лістинг 4.4 – Коду функції confirmAdding ()

```
void confirmAdding()
{
  WiFiClient client;
  HTTPClient http;
  Serial.println("Enter your name: ");
  while(Serial.available() == 0){}
  String name = Serial.readString();
  Serial.print(name); Serial.println();
  Serial.println("Enter your gender: ");
  while(Serial.available() == 0){}
  String gender = Serial.readString();
  Serial.print(gender); Serial.println();
  postData = "api_key=" + apiKeyValue + "&enroll_finger_id="
+ String(id)+ "&name=" + name + "&gender=" + gender + "";
  http.begin(client,link);
  http.addHeader("Content-Type", "application/x-www-form-
urlencoded");
  int httpCode = http.POST(postData);
  String payload = http.getString();
  delay(1000);
```

```
  Serial.println(payload);
  http.end();
}
```

Після цього викликається основна функція loop().

Функція loop являє собою нескінченний цикл, який забезпечує безперервну роботу мікроконтролера і у якому відбувається весь функціонал коду. Спочатку відбувається перевірка підключення до мережі WiFi, далі відбувається виклик функції getFingerprintID(Додаток Б), яка сканує відбиток пальця та шукає його у пам’яті сканера, якщо відбиток знайдено – id знайденого відбитка записується у змінну FingerID (лістинг 4.5).

Лістинг 4.5 – Фрагмент коду функції getFingerprintID()

```
    p = finger.fingerFastSearch();
      if (p == FINGERPRINT_OK) {

        Serial.println("Found a print match!");
      } else if (p == FINGERPRINT_PACKETRECIEVEERR) {
        Serial.println("Communication error");
        return -2;
      } else if (p == FINGERPRINT_NOTFOUND) {
        Serial.println("Did not find a match");
        return -1;
      } else {
        Serial.println("Unknown error");
        return -2;
      }
      // found a match

      Serial.print("Found ID #");
    Serial.print(finger.fingerID);

      Serial.print(" with confidence of ");
    Serial.println(finger.confidence);

  return finger.fingerID;
```

Далі, якщо змінна FingerID містить число більше нуля, викликається функція SendFingerprintID і параметром передається змінна FingerID.

Функція SendFingerprintID надсилає запит, який містить захисний ключ

та id відбитку пальця, який був переданий у функцію як параметр (лістинг 4.6). У залежності від відповіді, яку функція отримає від бази даних, на екран виведеться повідомлення про вхід або вихід користувача.

Лістинг 4.6 – Код функції SendFingerprintID()

```
void SendFingerprintID(int finger)
{
  WiFiClient client;
  HTTPClient http;
  postData = "api_key=" + apiKeyValue + "&log_finger_id=" +
String(finger);
  http.begin(client,link);
  http.addHeader("Content-Type", "application/x-www-form-
urlencoded");
  int httpCode = http.POST(postData);
  String payload = http.getString();
  Serial.println(payload);
  Serial.println(postData);

  if (payload.substring(0, 5) == "login") {
    String user_name = payload.substring(5);
    Serial.print("User "); Serial.print(user_name);
Serial.print(" with fingerprint ID = ");
    Serial.print(finger); Serial.println(" login!");
  }
  else if (payload.substring(0, 6) == "logout") {
    String user_name = payload.substring(6);
    Serial.print("User "); Serial.print(user_name);
Serial.print(" with fingerprint ID = ");
    Serial.print(finger); Serial.println(" logout!");
  }
  delay(1000);

  postData = "";
  http.end();
}
```

Останньою виконується функція ChecktoDeleteID. Ця функція надсилає запит до бази даних для перевірки необхідності видалення відбитку пальця з пам’яті сканера (лістинг 4.7). Ця функція синхронізує дані сканера відбитків та бази даних.

Лістинг 4.7 – Код функції ChecktoDeleteID

```
void ChecktoDeleteID()
{
  WiFiClient client;
  HTTPClient http;
  postData = "DeleteID=check";
  http.begin(client,link);
  http.addHeader("Content-Type",    "application/x-www-form-
urlencoded");    //Specify content-type header
  int httpCode = http.POST(postData);
  String payload = http.getString();

  if (payload.substring(0, 6) == "del-id") {
    String del_id = payload.substring(6);
    deleteFingerprint( del_id.toInt() );
  }
  http.end();
}
```

4.6 Написання програмного коду WEB-сторінки та бази даних мовами PHP, HTML/CSS

Код WEB-сторінки розподілений на чотири файли: файл відображення сторінки з усіма зареєстрованими користувачами “index.php”, файл відображення сторінки з відвідуванням користувачів “users_log.php”, файл обробки запитів від мікроконтролера esp8266 “post_data.php”, та файл стилів компонентів сторінки “style.css”.

4.6.1 Створення бази даних та таблиць

Для створення веб-серверу та бази даних у проекті було використано дистрибутив XAMPP.

XAMPP – безкоштовний кросплатформовий дистрибутив для складання локального веб-сервера, містить Apache, MariaDB, мову програмування Perl, інтерпретатор скриптів PHP та додаткові бібліотеки та має відкритий вихідний код, простий у встановленні та використанні[8, 9].

Інтерфейс додатку XAMPP представлено на рис. 4.1.

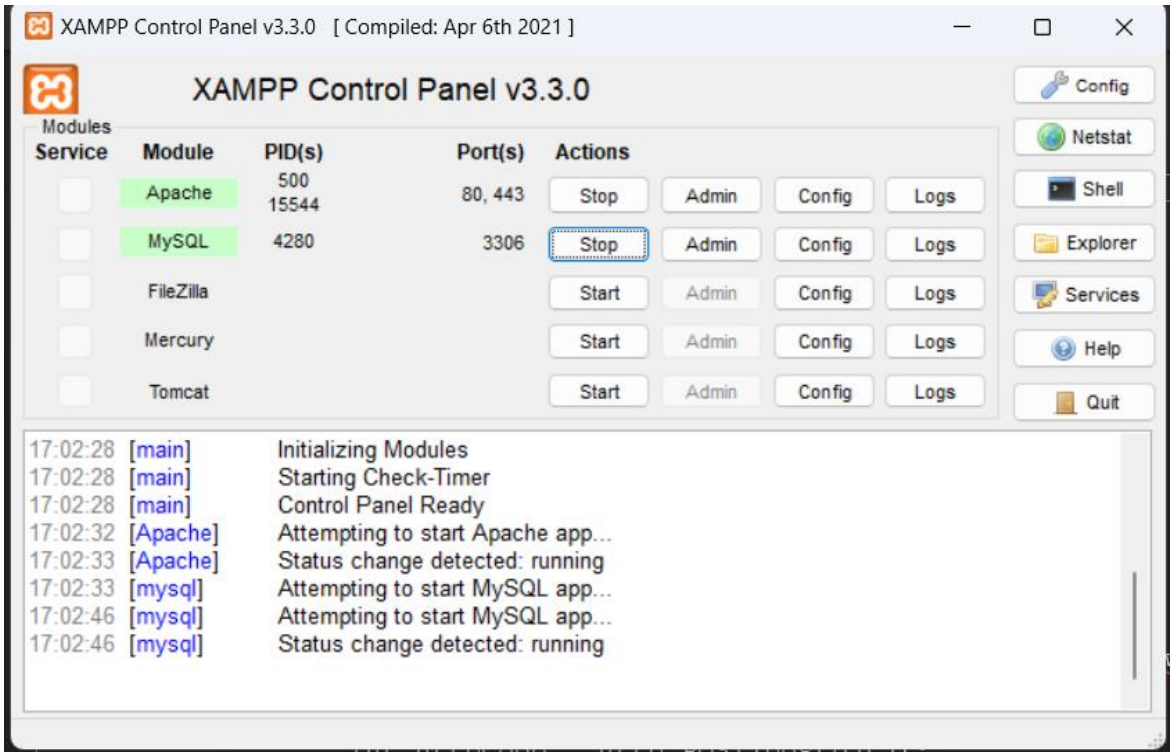


Рисунок 4.1 – Інтерфейс додатку XAMPP

Спочатку, для того щоб увімкнути веб-сервер, необхідно запустити додаток XAMPP Control Panel та запустити модулі Apache та MySQL.

Після цього, для того щоб перейти до інтерфейсу взаємодії з базою даних, необхідно натиснути кнопку Admin у модулі MySQL.

Далі у браузері відкривається phpMyAdmin – веб-сторінка взаємодії з базами даних.

Веб-сторінка phpMyAdmin представлена на рисунку 4.2.

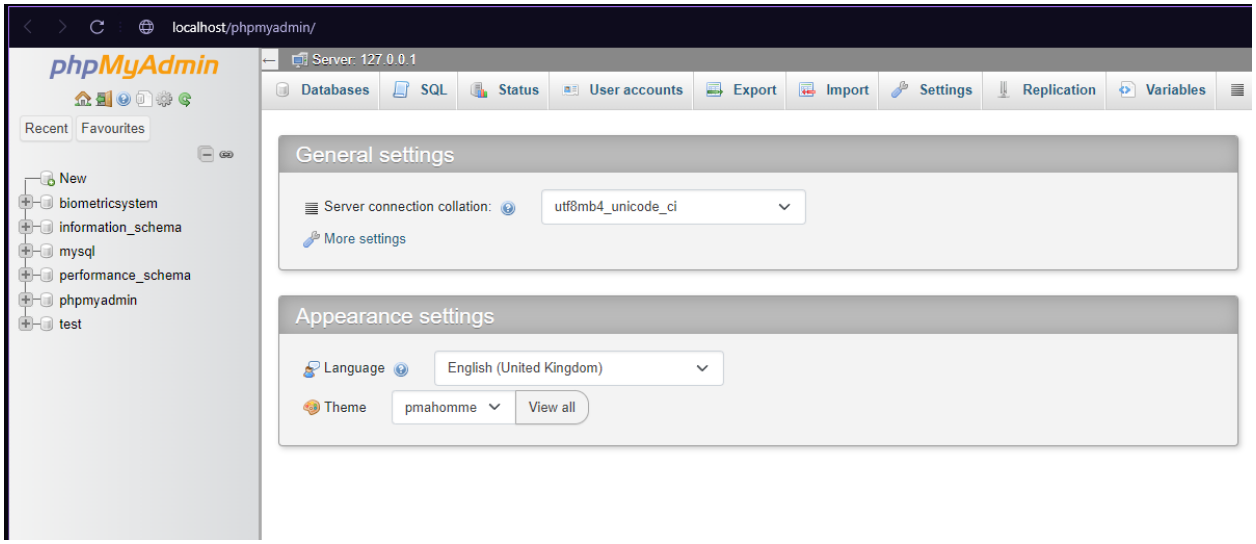


Рисунок 4.2 – Веб-сторінка phpMyAdmin

Після цього створюється база даних biometricsystem за допомогою команди CREATE DATABASE:

```
CREATE DATABASE biometricsystem;
```

Наступними, у базі даних biometricsystem, за допомогою команди CREATE TABLE створюються таблиці users та users_logs:

Лістинг 4.8 – Створення таблиць users та users_logs

```
CREATE TABLE `users` (
        `id` int(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,
        `username` varchar(100) NOT NULL,
        `gender` varchar(10) NOT NULL,
        `fingerprint_id` int(11) NOT NULL,
        `del_fingerid` tinyint(1) NOT NULL DEFAULT '0'
);

CREATE TABLE `users_logs` (
        `id` int(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,
        `username` varchar(100) NOT NULL,
        `fingerprint_id` int(11) NOT NULL,
        `checkindate` date NOT NULL,
         `timein` time NOT NULL,
         `timeout` time NOT NULL);
```

На цьому етапі база даних готова для роботи з нею.

4.6.2 Файл коду сторінки зареєстрованих користувачів “index.php”

У файлі “index.php”(Додаток Б) у тезі <head> задається частота оновлення контенту сторінки на посилання на файл стилів “style.css”:

```
<head>
    <meta http-equiv="refresh" content="5" >
    <link rel="stylesheet"  href="css/style.css">
</head>
```

Після цього у тезі <body> створюються кнопки для навігації між сторінками. Ці кнопки містять посиланнями на веб-сторінки “index.php” та “users_logs.php”:

```
<body>
   <a href="index.php" class="button"> Users</a>
     <a   href="users_log.php"   class="button">   Users
Log</a>
   <h1>Registered Users</h1>
```

Далі, всередині тегу <body>, створюється тег з кодом на мові програмування “PHP” та оголошуються чотири змінні, які містять ім’я сервера, ім’я користувача, пароль та ім’я бази даних:

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "biometricsystem";
```

Наступним встановлюється спроба з’єднання з базою даних і перевіряється чи спроба була вдалою:

```
$conn   =   new   mysqli($servername,   $username,   $password,
$dbname);
```

```
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
```

Після цього перевіряється отримання сторінкою GET запиту з повідомленням “del-id”. Якщо повідомлення отримано, створюється sql-запит для видалення запису з id, яке містилося у повідомленні, з таблиці users:

```
if (isset($_GET['del_id'])) {
    $del_id = $_GET['del_id'];
       $sql="UPDATE users SET username='', gender='',
del_fingerid=1 WHERE fingerprint_id=$del_id";
    if ($conn->query($sql) === TRUE) {
    }
    else {
        echo "Error: " . $sql . "<br>" . $conn->error;
    }
}
```

Після цього створюється таблиця для відображення даних з таблиці users. Код відображення даних з таблиці users приведений у лістингу 4.9.

Лістинг 4.9 – Код відображення даних з таблиці users

```
$sql = "SELECT * FROM users ORDER BY id ASC";
echo '<table cellspacing="5" cellpadding="5">
      <tr>
        <th>ID</th>
        <th>Name</th>
        <th>Gender</th>
        <th>Finger ID</th>
        <th>Operation</th>
      </tr>';
if ($result = $conn->query($sql)) {
    while ($row = $result->fetch_assoc()) {
        $row_id = $row["id"];
        $row_username = $row["username"];
        $row_gender = $row["gender"];
        $row_fingerprint_id = $row["fingerprint_id"];
        echo "<tr>
                <td>$row_id</td>
                <td>$row_username</td>
                <td>$row_gender</td>
```

```
                <td>$row_fingerprint_id</td>
                <td>
          <a href='index.php?del_id=".$row_fingerprint_id."'
                    class='delBtn'> Delete</a>
                </td>
              </tr>";
    }
    $result->free();
}
$conn->close();
?>
```

4.6.3 Файл коду сторінки зареєстрованих користувачів “users_logs.php”

У файлі “users_logs.php”(Додаток Б) задається частота оновлення контенту сторінки, кнопки для навігації між сторінками, чотири змінні, які містять ім’я сервера, ім’я користувача, пароль, ім’я бази даних та встановлюється спроба з’єднання з базою даних так само, як у файлі “index.php”.

Після цього створюється таблиця, яка відображає дані з таблиці users_logs. Код відображення даних з таблиці users_logs приведено у лістингу 4.10.

Лістинг 4.10 – Код відображення даних з таблиці users_logs

```
$sql = "SELECT * FROM users_logs ORDER BY id DESC";
echo '<table cellspacing="5" cellpadding="5">
      <tr>
        <th>ID</th>
        <th>Name</th>
        <th>Finger ID</th>
        <th>Date</th>
        <th>TIME IN</th>
        <th>TIME OUT</th>
      </tr>';

if ($result = $conn->query($sql)) {
    while ($row = $result->fetch_assoc()) {
        $row_id = $row["id"];
        $row_username = $row["username"];
        $row_fingerprint_id = $row["fingerprint_id"];
        $row_checkindate = $row["checkindate"];
        $row_time_in = $row["timein"];
        $row_time_out = $row["timeout"];
```

```
echo '<tr>
                <td>' . $row_id . '</td>
                <td>' . $row_username . '</td>
                <td>' . $row_fingerprint_id . '</td>
                <td>' . $row_checkindate . '</td>
                <td>' . $row_time_in . '</td>
                <td>' . $row_time_out . '</td>

              </tr>';
    }

    $result->free();

}
$conn->close();
```

4.6.4 Файл коду “post_data.php”

У файлі “post_data.php”(Додаток Б) спочатку оголошуються змінні для з’єднання з базою даних та змінні, які попередньо оголошені та ініціалізовані для подальшого використання у коді:

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "biometricsystem";

$api_key_value = "tPmAT5Ab3j7F9";

$api_key= $name = $gender = $enroll_finger_id =
$log_finger_id = "";
```

Далі перевіряється наявність на веб-сервері прийнятих запитів методом POST з повідомленням “enroll_finger_id”, який відповідає за внесення нового користувача до бази даних, і у разі наявності такого запиту перевіряється захисний ключ. У разі співпадіння ключа у повідомленні та ключа веб-сервера – усі передані дані записуються до відповідних змінних.

Після цього створюється з’єднання з веб-сервером та базою даних і створюється sql-запит для внесення нового користувача до таблиці users. Далі цей запит надсилається до веб-серверу і, у разі успіху, виводиться

повідомлення про успішно створений новий запис у таблиці. Код обробки

POST-запиту з повідомленням “enroll_finger_id” наведено у лістингу 4.11.

Лістинг 4.11 – Код обробки POST-запиту з повідомленням “enroll_finger_id”

```
if (isset($_POST['enroll_finger_id'])) {

    $api_key = test_input($_POST["api_key"]);
    if($api_key == $api_key_value) {
        $name = test_input($_POST["name"]);
        $gender = test_input($_POST["gender"]);
                                    $enroll_finger_id       =
test_input($_POST["enroll_finger_id"]);

             $conn   =   new   mysqli($servername,   $username,
$password, $dbname);
        if ($conn->connect_error) {
                    die("Connection   failed:   "   .   $conn-
>connect_error);
        }

           $sql   =   "INSERT   INTO   users   (username,   gender,
fingerprint_id)
         VALUES ('" . $name . "', '" . $gender . "', '" .
$enroll_finger_id . "')";
        if ($conn->query($sql) === TRUE) {
            echo "New record created successfully";
        }
        else {
            echo "Error: " . $sql . "<br>" . $conn->error;
        }

        $conn->close();
    }
    else {
        echo "Wrong API Key provided.";
    }

}
```

Після цього перевіряється наявність на веб-сервері прийнятих запитів методом POST з повідомленням “log_finger_id”, який відповідає за перевірку відвідування користувача, тобто за точний час входу або виходу з аудиторії, і у разі наявності такого запиту перевіряється захисний ключ. У разі

співпадіння ключа у повідомленні та ключа веб-сервера – переданий у повідомленні id користувача записується до змінної та створюється з'єднання з сервером.

Далі до веб-серверу надсилається запит для перевірки наявності у таблиці users користувача з id, прийнятому з повідомлення від сенсора. У разі наявності користувача з таким id його ім'я записується до окремої змінної.

Наступним створюється і надсилається на сервер sql-запит для перевірки наявності у таблиці users_logs користувача зі збереженим у окремій змінній ім'ям.

Якщо такий користувач існує, то це означає, що він вже знаходиться у аудиторії і зараз знову скористався сканером відбитків пальців для виходу з аудиторії, тому у відповідному записі таблиці users_logs оновлюється лише час виходу з аудиторії цього користувача.

У іншому випадку, якщо користувача з таким ім'ям у таблиці не існує, то це означає, що користувач заходить до аудиторії і у таблиці створюється новий запис з ім'ям користувача, який відповідає переданому у повідомленні id та записується час входу. Фрагмент коду обробки POST-запиту з повідомленням "log_finger_id" приведений у лістингу 4.12.

Лістинг 4.12 – Фрагмент коду обробки POST-запиту з повідомленням "log_finger_id"

```
if (!empty($row['username'])){
                $name = test_input($row['username']);
                    $sql = "SELECT * FROM users_logs WHERE
fingerprint_id=$log_finger_id AND checkindate=CURDATE() AND
timeout=''";
                $result = mysqli_query($conn, $sql);
                //Login
                if (!$row = mysqli_fetch_assoc($result)){
                            $sql = "INSERT INTO users_logs
(username, fingerprint_id, checkindate, timein, timeout)
VALUES (?, ?, CURDATE(), CURTIME(), ?)";
                    $result = mysqli_stmt_init($conn);
                    if (!mysqli_stmt_prepare($result, $sql))
{
```

```
                        echo "SQL_Error_Select_login1";
                        exit();
                    }
                    else{
                        $timeout = "";
                             mysqli_stmt_bind_param($result,
"sis", $name, $log_finger_id, $timeout);
                        mysqli_stmt_execute($result);
                        echo "login".$name;
                        exit();
                    }
                }

                //Logout
                else {
                                $sql="UPDATE  users_logs  SET
timeout=CURTIME()      WHERE      fingerprint_id=?      AND
checkindate=CURDATE()";
                    $result = mysqli_stmt_init($conn);
                    if (!mysqli_stmt_prepare($result, $sql))
{
                        echo "SQL_Error_insert_logout1";
                        exit();
                    }
                    else{
                        mysqli_stmt_bind_param($result, "i",
$log_finger_id);
                        mysqli_stmt_execute($result);

                        echo "logout".$name;
                        exit();
                    }
                }
            }
```

Після цього перевіряється наявність на веб-сервері прийнятих запитів методом POST з повідомленням “DeleteID”, який відповідає видалення користувача з бази даних. Якщо запит містить повідомлення з текстом “check”, то створюється з’єднання з веб-сервером та відправляється sql-запит для отримання з таблиці users усіх користувачів з полем del_fingerid=1.

Далі id цих користувачів відправляється на контролер esp8266 для видалення запису з цим id з пам’яті сенсора відбитків пальців.

Після цього відправляється sql-запит для видалення користувачів з цим id з таблиці users. Фрагмент коду обробки POST-запиту з повідомленням “DeleteID” приведено у лістингу 4.13.

Лістинг 4.13 – Фрагмент коду обробки POST-запиту з повідомленням “DeleteID”

```
if (isset($_POST['DeleteID'])) {
    if ($_POST['DeleteID'] == "check") {
            $conn = new mysqli($servername, $username, $password, $dbname);

        if ($conn->connect_error) {
                    die("Connection failed: " . $conn->connect_error);
        }
          $sql = "SELECT fingerprint_id FROM users WHERE del_fingerid=1";
        $result = mysqli_stmt_init($conn);

        if (!mysqli_stmt_prepare($result, $sql)) {
            echo "SQL_Error_Select";
            exit();
        }
        else{
            mysqli_stmt_execute($result);
            $resultl = mysqli_stmt_get_result($result);
            if ($row = mysqli_fetch_assoc($resultl)) {
                echo "del-id".$row['fingerprint_id'];
                        $sql = "DELETE FROM users WHERE del_fingerid=1";
                $result = mysqli_stmt_init($conn);
                if (!mysqli_stmt_prepare($result, $sql)) {
                    echo "SQL_Error_delete";
                    exit();
                }
                else{
                    mysqli_stmt_execute($result);
                    exit();
                }
            }
            else{
                echo "nothing";
                exit();
            }
        }
    }
    else{
        exit();
    }
}
```

4.6.5 Файл коду стилів “style.css”

У файлі “style.css”(Додаток Б) описані стилі та кольори фону, кнопок, заголовків, таблиць, заголовків таблиць, які розміщені на веб-сторінках. Усі кольори підібрані для зручності та читаності веб-сторінок.

Лістинг 4.14 – Фрагмент коду опису стилів веб-сторінок

```
body {
    background: #2d545e;
    box-sizing: border-box;
    color: #000;
    font-size: 1.8rem;
    letter-spacing: -0.015em;
    text-align: center;
}

.button {
    background-color: #666;
    border: none;
    color: white;
    padding: 15px 32px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
    font-size: 18px;
    border-collapse: separate;
    border: 1px solid #000;
}
```

4.7 Демонстрація працездатності проекту

Для початку демонстрації необхідно підключити USB-кабель до мікроконтролера та одного з доступних USB-портів на комп’ютері.

Далі відкривши у пункті меню Інструменти/Монітор порту можна відстежувати дії з мікроконтролером через цей порт.

Скомпілювавши проект у Arduino IDE починається завантаження коду до мікроконтролера. Завантаження коду у пам’ять мікроконтролера показано на рис. 4.3.

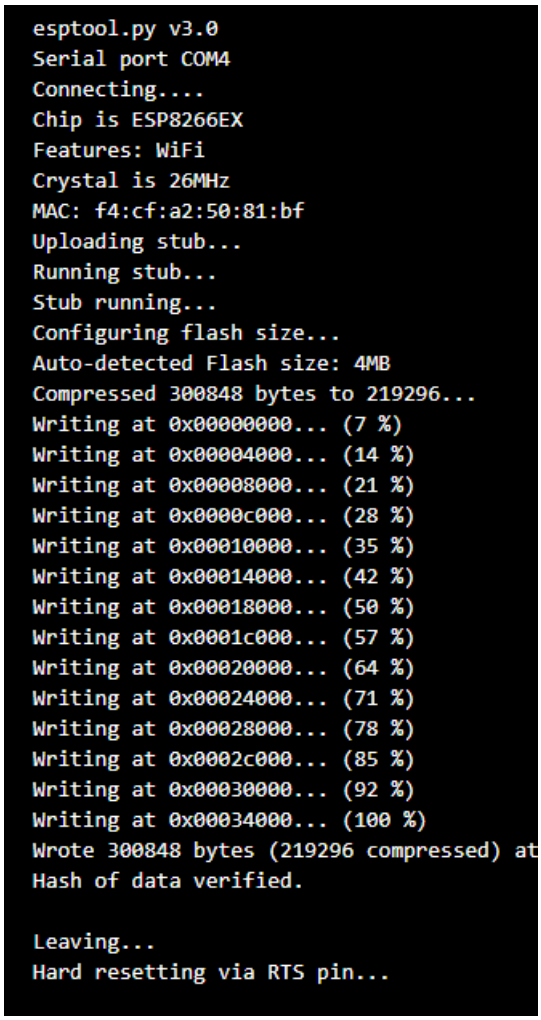


Рисунок 4.3 – Завантаження коду у пам’ять мікроконтролера

Спочатку відбувається спроба підключення до мережі Wi-Fi. При успішному з’єднанні з мережею виводиться IP-адреса (рис. 4.4).

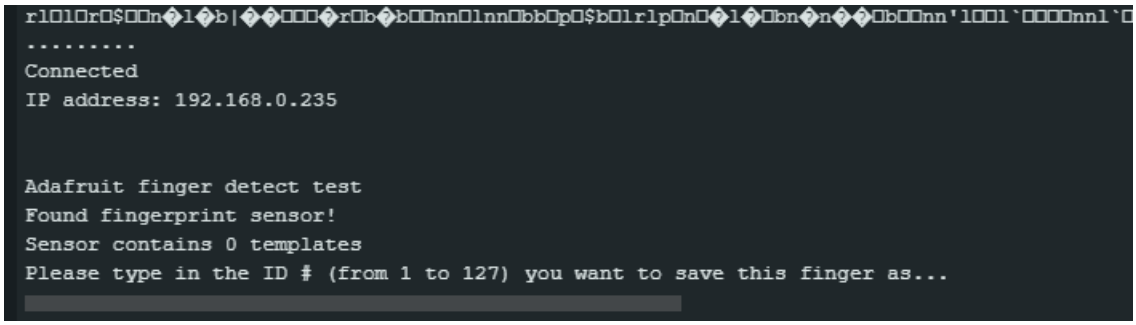


Рисунок 4.4 – З’єднання з мережею Wi-Fi

Далі необхідно ввести id за яким у пам’яті сенсора відбитків пальців буде збережено відбиток пальця і просканувати відбиток, цю процедуру необхідно повторити для 3 користувачів. Збереження відбитків пальців користувачів у пам’яті сенсора приведено на рис. 4.5.

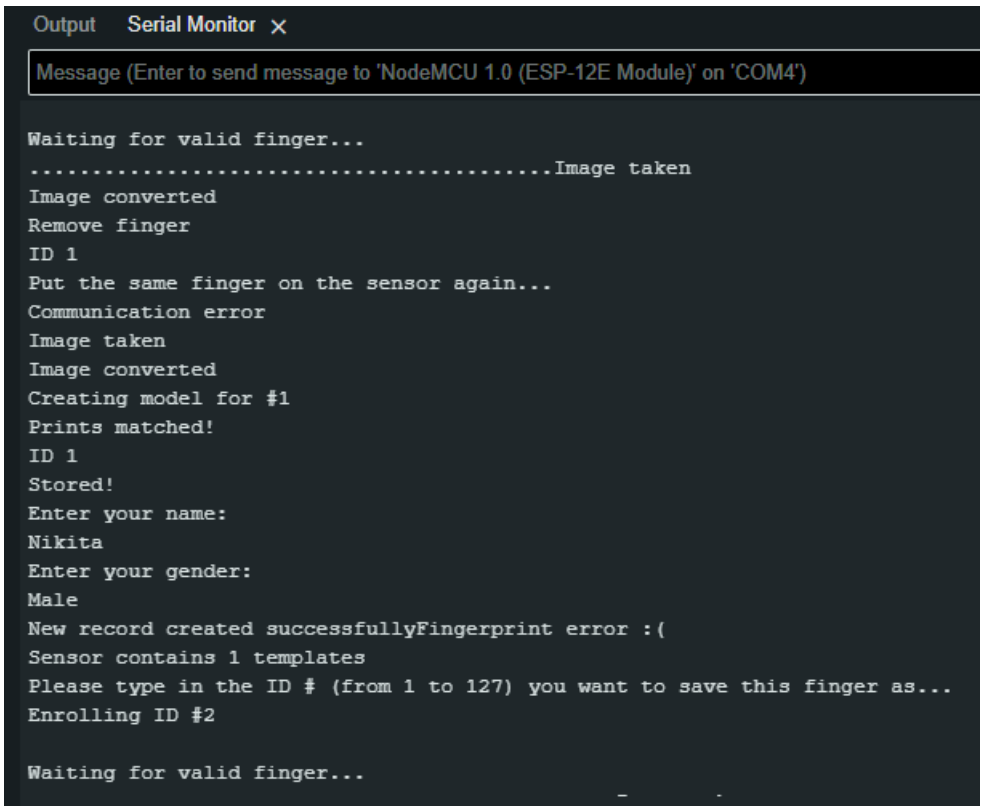


Рисунок 4.5 – Збереження відбитків пальців користувачів у пам’яті сенсора

Введені дані також одразу надсилаються до бази даних і відображаються на веб-сторінці браузера. Веб-сторінка з зареєстрованими користувачами показана на рис. 4.6.

Рисунок 4.6 – Веб-сторінка з зареєстрованими користувачами

Після цього сканер переходить у режим сканування відбитків для входу або виходу з аудиторії. При першому прикладанні пальця система вважає, що зареєстрований користувач увійшов до аудиторії і його дані заносяться до бази даних і відображаються на веб-сторінці(рис. 4.7).

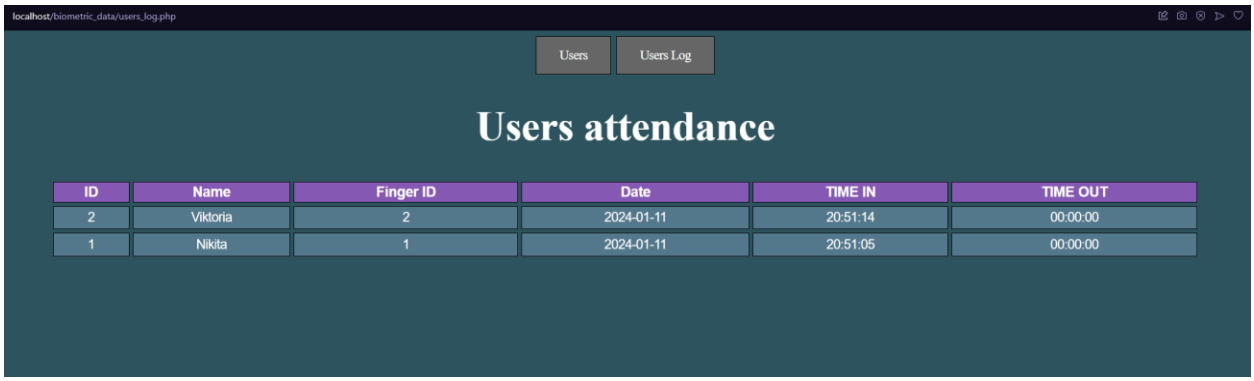


Рисунок 4.7 – Вхід зареєстрованого користувача до аудиторії

При повторному прикладанні пальця до сканеру система вважає, що користувач виходить з аудиторії і заносить час виходу до бази даних(рис. 4.8).

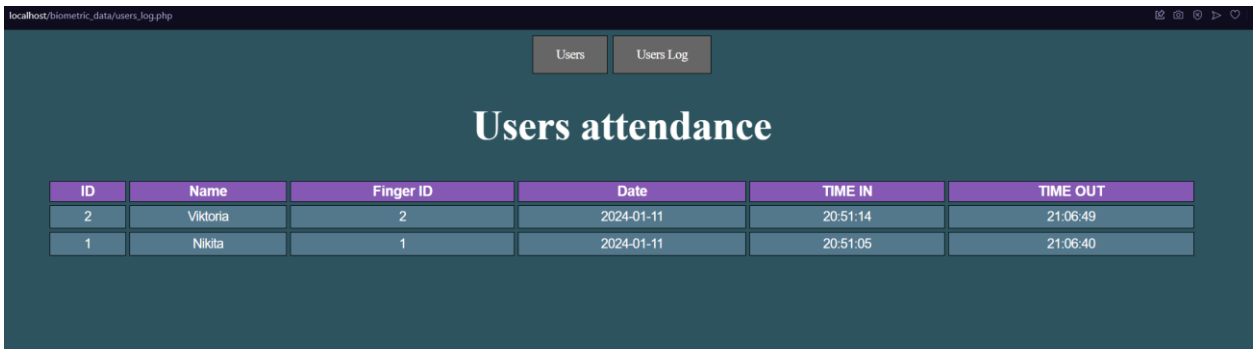


Рисунок 4.8 – Вихід зареєстрованого користувача з аудиторії

Якщо після виходу користувач знову прикладе палець до сканера, то система буде вважати, що цей користувач знова зайшов до аудиторії і створить новий запис у базі даних(рис. 4.9).

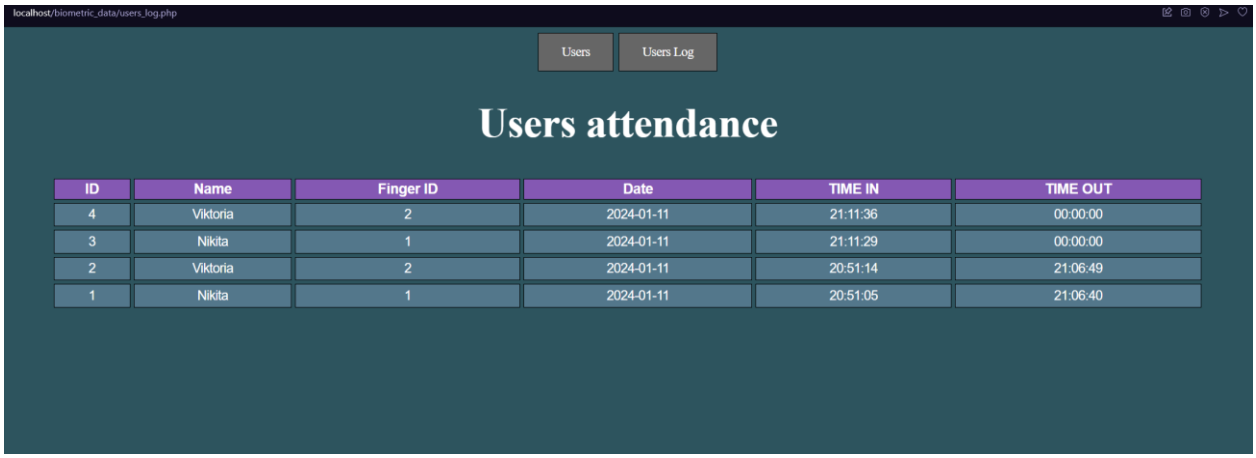


Рисунок 4.9 – Повторний вхід користувача до системи

Також система має можливість видалення зареєстрованого користувача. Користувач видаляється з бази даних і з пам’яті сенсора відбитків пальців(рис. 4.10 і 4.11).

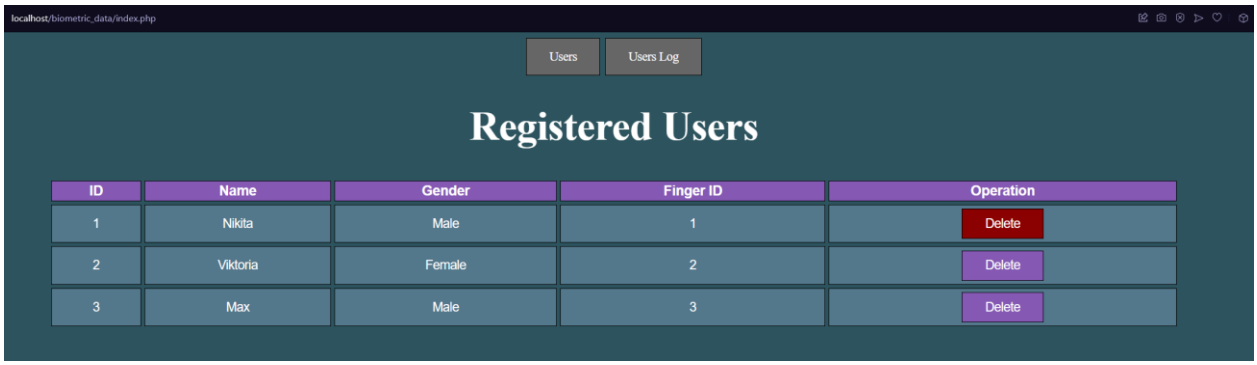


Рисунок 4.10 – Видалення користувача з бази даних

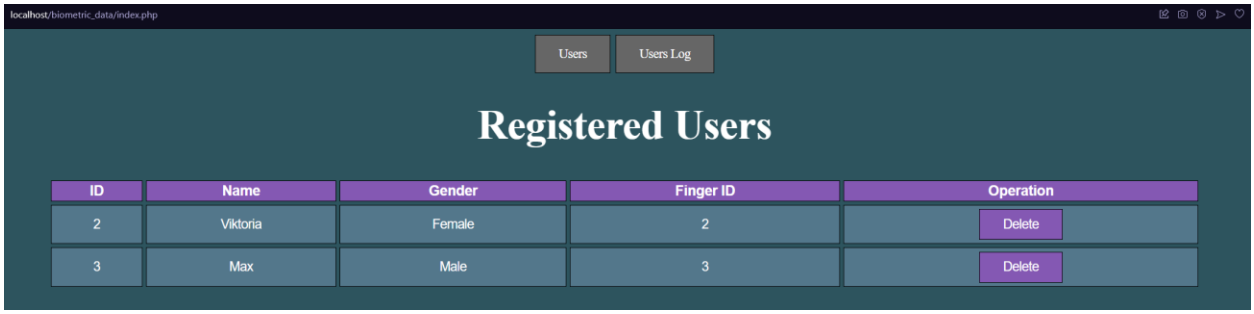


Рисунок 4.11 – Відображення бази даних після внесених змін

ВИСНОВКИ

У ході кваліфікаційної роботи було розроблено біометричну підсистему розпізнавання особистостей (студентів) в системі кіберуніверситету на основі мікроконтролера, сенсора відбитків пальців та бази даних, де мікроконтролер працює у автоматичному режимі, при цьому база даних зберігає усі зареєстровані відбитки пальців студентів. Для реалізації біометричної системи розпізнавання був здійснений вибір усіх необхідних компонентів системи, яка включає у себе мікроконтролер ESP8266(NodeMCU), сенсор відбитків пальців AS608 та базу даних, з необхідними технічними характеристиками та функціями, створено структурну та функціональну схеми системи.

Основою системи розпізнавання став мікроконтролер ESP8266 (NodeMCU), що має усі характеристики, що необхідні для створення системи біометричного розпізнавання – Wi-Fi модуль, який дозволяє мати бездротовий зв’язок з базою даних через мережу Інтернет, сам мікроконтролер, який має невисоку вартість, і завдяки цьому, зменшує вартість усієї системи в цілому, плюсом є те, що він має знижене споживання енергії.

У програмній частині проекту реалізовано взаємодію між сканером відбитків пальців, мікроконтролером, базою даних та веб-сторінкою. При виконанні програмної частини проекту було обрано середовище розробки Arduino IDE та мову програмування C++ для мікроконтролера і середовище розробки Visual Studio Code та мови програмування PHP? HTML та CSS для веб-сторінки відповідно.

Прототип може бути удосконалений, зокрема, за рахунок низького енергоспоживання мікроконтролера ESP8266 (NodeMCU), для живлення усієї системи може бути використано акумулятор, і внаслідок цього, виникає можливість поєднання всієї системи у компактний корпус. Також створена

система біометричного розпізнавання може бути вдосконалена за рахунок додавання нового функціоналу веб-сторінки та покращення вже існуючих функцій. У подальшому вдосконаленні прототипу, для підвищення захисту університету, у систему може бути додано додатковий засіб ідентифікації студентів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Раві Дас Biometric Technology: Authentication, Biocryptography, and Cloud-Based Architecture [Текст] : пер. з англ. – Рутлед, 2020 – ISBN 9780367670078.

2. Карм Вір Арья, Робін Сінгх Бхадорія The Biometric Computing : Recognition and Registration [Текст] : пер. з англ. – Чапман & Холл 2019 – ISBN 9780815393641.

3. ESP8266 (NodeMCU) [Електронний ресурс]. – Режим доступу: https://www.espressif.com/sites/default/files/documentation/0a esp8266ex_datasheet_en.pdf

4. Модуль AS608 [Електронний ресурс]. – Режим доступу: https://rajguruelectronics.com/Product/254/AS608%20Fingerprint%20reader%20sensor%20module.pdf

5. Аніта Гелот, Раджеш Сінгх, Правін Кумар Малік, Лові Радж Гупта, Бхупендра Сінгх Internet of Things with 8051 and ESP8266 [Текст] : пер. з англ. – CPC Пресс; 1-е видання, 2020. – ISBN-10 : 0367534789.

6. Камерон Н. Electronics Projects with the ESP8266 and ESP32: Building Web Pages, Applications, and WiFi Enabled Devices [Текст] : пер. з англ. – Апресс; 1-е видання, 2021. – ISBN-10 : 1484263359.

7. Бен Фрейн Responsive Web Design with HTML5 and CSS3 - Second Edition: Build responsive and future-proof websites [Текст] : пер. з англ. – Пакт Паблішінг, 2017. – ISBN 1784398934.

8. Кемерон Арчі Mastering PHP: A Comprehensive Guide to Web Development [Текст] : пер. з англ. – Опубліковано незалежно, 2023. – ISBN 979-8865139201.

9. Ситнік Н.О. Система біометричного розпізнавання студентів у системі безпеки кіберуніверситету / Н.О. Ситнік // Матеріали тез 27-го міжнародного молодіжного форуму «Радіоелектроніка та молодь у XXI столітті» - м. Харків, травень 2023. - С. 43-44.

Відомість кваліфікаційної роботи

«Підсистема біометричного розпізнавання особистостей
в системі безпеки кіберуніверситету»

	Прізвище та ініціали відповідальної особи	Підпис	Дата
Роботу виконав студент групи СКСм-22-1 Структура кваліфікаційної роботи: – пояснювальна записка 60 с.; – графічний матеріал _16_ арк..	Ситнік Н.О.		03.01.24
Керівник роботи	Ларченко Л.В.		03.01.24
Перевірка на плагіат здійснена. Оригінальність авторського тексту складає _86_ %	Литвинова Є.І.		05.01.24
Нормоконтроль проведено :	Ларченко Л.В.		10.01.24