Міністерство освіти і науки України

Луцький національний технічний університет
(повне найменування закладу вищої освіти)

Факультет комп'ютерних та інформаційних технологій
(повне найменування факультету)

Кафедра комп'ютерної інженерії та охоронних систем
(повне найменування кафедри)

КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»

СИСТЕМА АУНТИФІКАЦІЇ КОРИСТУВАЧІВ З ВИКОРИСТАННЯМ ВЕБ-ІНТЕРФЕЙСУ НА БАЗІ RASPBERRY PI

USER AUTHENTICATION SYSTEM USING A WEB INTERFACE BASED ON RASPBERRY PI

спеціальність 123 Комп'ютерна інженерія
(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія
(назва освітньої програми)

Виконав: здобувач вищої освіти
групи КІ-42
Вільний Дмитро Андрійович

(підпис)

Керівник:
асистент
Кулакевич Олег Русланович

(підпис)

Кваліфікаційну роботу
допущено до захисту
«____» червня 2026 р.
Гарант освітньої програми:
к.т.н., доцент
Лавренчук Світлана Василівна

(підпис)

Луцьк – 2026 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет *комп'ютерних та інформаційних технологій*
Кафедра *комп'ютерної інженерії та безпеки*
Ступінь вищої освіти: *бакалавр*
Галузь знань: *12 Інформаційні технології*
Спеціальність: *123 Комп'ютерна інженерія*
Освітня програма: *«Комп'ютерна інженерія»*

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ доц. Т. Терлецький
« 23 » 12 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Вільному Дмитру Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи *СИСТЕМА АУНТИФІКАЦІЇ КОРИСТУВАЧІВ З ВИКОРИСТАННЯМ ВЕБ-ІНТЕРФЕЙСУ НА БАЗІ RASPBERRY PI*

Керівник роботи *асистент Кулакевич Олег Русланович*

затверджені наказом закладу вищої освіти від «20» грудня 2025 року № 536/01-02
2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 28.05.2026 р.

3. Вихідні дані до роботи *Джерелом розробки є науково-технічна література та публікації в періодичних виданнях з даного питання, опубліковані зарубіжні та вітчизняні роботи в даній області, різні інтернет-ресурси технічного спрямування*

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):
Вступ
Аналіз предметної області та існуючих рішень
Розробка апаратної частини мобільної платформи
Реалізація та системи аутентифікації на базі raspberry pi
Висновки

5. Перелік графічного (ілюстративного) матеріалу:
Архітектура системи аутентифікації користувачів
Інтерфейс вебзастосунку системи
Процес налаштування Raspberry P;
Реалізація NFC-аутентифікації користувачів
Організація API-взаємодії між Raspberry Pi та сервером
Журнал подій та керування користувачами

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз предметної області та існуючих рішень	*Кулакевич О.Р., асистент*		
Розробка апаратної частини мобільної платформи	*Кулакевич О.Р., асистент*		
Реалізація та системи аутентифікації на базі raspberry pi	*Кулакевич О.Р., асистент*		
Нормоконтроль	*Багнюк Н. В., доцент*		
Гарант ОП	*Лавренчук С. В., доцент*		
Показник запозичень тексту	***___%***		
Академічна доброчесність	*Міскевич О. І., ст. викладач*		

7. Дата видачі завдання 23.12.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	*Огляд літератури із досліджуваної проблеми, аналіз предметної області та наявних рішень*	до 10.02.2026 р.	
2.	*Аналіз предметної області та існуючих рішень*	до 02.03.2026 р.	
3.	*Розробка апаратної частини мобільної платформи*	до 02.04.2026 р.	
4.	*Реалізація та системи аутентифікації на базі raspberry pi та формування додатків*	до 10.04.2026 р.	
5.	*Представлення остаточного варіанту кваліфікаційної роботи керівникові*	до 01.05.2026 р.	
6.	*Нормоконтроль*	до 23.05.2026 р.	
7.	*Інструментальна перевірка на академічний плагіат*	до 26.05.2026 р.	
8.	*Здача кваліфікаційної роботи та всіх супровідних документів на кафедру*	до 28.06.2026 р.	

Здобувач вищої освіти _____ Вільний Д.А.
(підпис) (прізвище, ініціали)

Керівник кваліфікаційної роботи _____ Кулакевич О.Р.
(підпис) (прізвище, ініціали)

АНОТАЦІЯ

Вільний Д.А. Система аутентифікації користувачів з використанням веб-інтерфейсу на базі Raspberry Pi. Рукопис.

Кваліфікаційна робота бакалавра ОП «Комп'ютерна інженерія» спеціальності 123 Комп'ютерна інженерія. Луцький національний технічний університет. Луцьк, 2026.

Кваліфікаційна робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків.

У першому розділі виконано аналіз сучасних методів аутентифікації користувачів, розглянуто особливості апаратної та програмної аутентифікації, принципи роботи NFC-технологій та сучасних систем контролю доступу. Проведено аналіз апаратної платформи Raspberry Pi, NFC-зчитувачів та засобів безконтактної ідентифікації користувачів. Обґрунтовано вибір апаратних компонентів для реалізації системи аутентифікації.

У другому розділі обґрунтовано вибір програмних та апаратних засобів реалізації системи. Розглянуто процес встановлення та налаштування операційної системи Raspberry Pi OS, серверного середовища Apache, PHP, MySQL та фреймворку Laravel. Розроблено архітектуру системи, механізм взаємодії NFC-зчитувача із серверною частиною через REST API, а також структуру бази даних і адміністративного веб-інтерфейсу.

У третьому розділі реалізовано програмно-апаратний комплекс аутентифікації користувачів на базі Raspberry Pi 3 Model B та ACS ACR1252U III USB. Реалізовано механізм генерації та перевірки API-токенів, систему журналювання дій користувачів, адміністративну панель керування та веб-інтерфейс системи. Проведено тестування роботи NFC-аутентифікації, API-взаємодії та функціональних можливостей системи.

Ключові слова: Raspberry Pi, NFC, аутентифікація, веб-інтерфейс, Laravel, API, токен, REST API, система контролю доступу, PHP.

ANNOTATION

Vilniy D. User authentication system using a web interface based on Raspberry Pi. Manuscript.

Qualification work of the bachelor of the specialty "Computer Engineering" of the specialty 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2026.

The qualification work consists of an introduction, three sections, conclusions, a list of used sources and appendices.

The first section analyzes modern methods of user authentication, considers the features of hardware and software authentication, the principles of NFC technologies and modern access control systems. The Raspberry Pi hardware platform, NFC readers and contactless user identification tools are analyzed. The choice of hardware components for implementing the authentication system is justified.

The second section justifies the choice of software and hardware tools for implementing the system. The process of installing and configuring the Raspberry Pi OS operating system, the Apache server environment, PHP, MySQL and the Laravel framework is considered. The system architecture, the mechanism for interaction of the NFC reader with the server part via REST API, as well as the structure of the database and the administrative web interface have been developed.

In the third section, a hardware and software complex for user authentication based on Raspberry Pi 3 Model B and ACS ACR1252U III USB has been implemented. The mechanism for generating and verifying API tokens, a user action logging system, an administrative control panel and a web interface of the system have been implemented. The operation of NFC authentication, API interaction and system functionality have been tested.

Keywords: Raspberry Pi, NFC, authentication, web interface, Laravel, API, token, REST API, access control system, PHP.

ЗМІСТ

ВСТУП

У сучасних інформаційних системах питання безпеки та контролю доступу користувачів є одним із ключових напрямів розвитку. Зростання кількості цифрових сервісів, автоматизованих систем управління та IoT-рішень вимагає впровадження надійних механізмів аутентифікації. Особливу актуальність набувають системи, що поєднують апаратні засоби і веб-технології, оскільки вони забезпечують зручність використання, централізоване управління та підвищений рівень безпеки. Використання одноплатних комп'ютерів, зокрема Raspberry Pi, дозволяє створювати компактні та ефективні рішення для ідентифікації користувачів за допомогою карток доступу.

Метою роботи є розробка системи аутентифікації користувачів з використанням веб-інтерфейсу на базі Raspberry Pi та карткової технології ідентифікації.

Об'єкт дослідження – системи аутентифікації користувачів у комп'ютерних та вбудованих інформаційних системах.

Предмет дослідження – система аутентифікації користувачів з використанням веб-інтерфейсу та карткової технології на базі одноплатного комп'ютера Raspberry Pi.

Завдання, які необхідно виконати:

– реалізувати систему зчитування даних з карт користувачів та їх обробку на базі Raspberry Pi;

– розробити веб-інтерфейс для керування користувачами та перегляду результатів аутентифікації;

– дослідити принципи роботи RFID/NFC технологій та їх інтеграцію з веб-системами;

– візуалізувати структуру та архітектуру розробленої системи аутентифікації;

– спроектувати базу даних для зберігання інформації про користувачів та їх картки доступу;

– запропонувати методи підвищення безпеки та надійності роботи системи.

Реалізація системи аутентифікації користувачів на базі Raspberry Pi дозволяє об’єднати апаратні та програмні компоненти в єдине інтегроване рішення, яке забезпечує швидку та надійну ідентифікацію користувачів. Використання карткової технології значно спрощує процес доступу, зменшує час перевірки та підвищує зручність експлуатації системи.

Запропонована система може бути застосована у різних сферах, зокрема в навчальних закладах, офісах, лабораторіях та інших об’єктах, де необхідний контроль доступу. Веб-інтерфейс забезпечує можливість віддаленого керування користувачами, моніторингу подій та ведення журналу аутентифікації.

Таким чином, розробка даної системи є актуальною та практично значущою, оскільки поєднує сучасні технології ідентифікації, веб-розробки та вбудованих систем, що дозволяє створити ефективне та масштабоване рішення для контролю доступу користувачів.

РОЗДІЛ 1
АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ РІШЕНЬ

1.1 Поняття та принципи аутентифікації користувачів

Аутентифікація користувачів є одним із ключових механізмів забезпечення інформаційної безпеки сучасних інформаційних систем. Вона призначена для підтвердження заявленої особою ідентичності перед наданням доступу до ресурсів системи.

«Аутентифікація – це процес перевірки облікових даних користувача (зазвичай імені та пароля). Іншими словами введені облікові дані звіряються з даними, що зберігаються в базі даних (користувача перевіряють за допомогою пароля, лист перевіряють по електронного підпису)» [1]. Основною метою аутентифікації є запобігання несанкціонованому доступу, захист конфіденційної інформації та забезпечення цілісності даних.

Щоб пройти автентифікацію, людям потрібно ввести імена користувачів, паролі або PIN-коди, просканувати обличчя чи відбитки пальців тощо. «Захист ідентичностей забезпечується за рахунок того, що жоден із цих способів автентифікації не зберігається в базі даних служби. Паролі гешуються (не шифруються), і геші зберігаються в базі даних. Коли користувач намагається ввійти за допомогою пароля, введений пароль повторно гешується, а потім порівнюється з тим, що зберігається в базі даних. Якщо два геші збігаються, користувач отримує доступ» [2]. У випадку використання сканування відбитків пальців або розпізнавання обличчя отримані біометричні дані підлягають кодуванню, шифруванню та зберіганню на пристрої.

Основними принципами створення систем аутентифікації є надійність, висока швидкість роботи, можливість масштабування та захищеність від атак. Така система має протидіяти підбору паролів, атакам «людина посередині», SQL-ін’єкціям та іншим поширеним загрозам інформаційної безпеки мережі.

1.2 Методи аутентифікації

1.2.1 Парольна аутентифікація

«Ще не дуже давно парольна ідентифікація (рис. 1.1) була чи ледве не єдиним способом визначення особистості користувача. Справа в тому, що парольна ідентифікація найбільш проста як у реалізації, так й у використанні» [3].

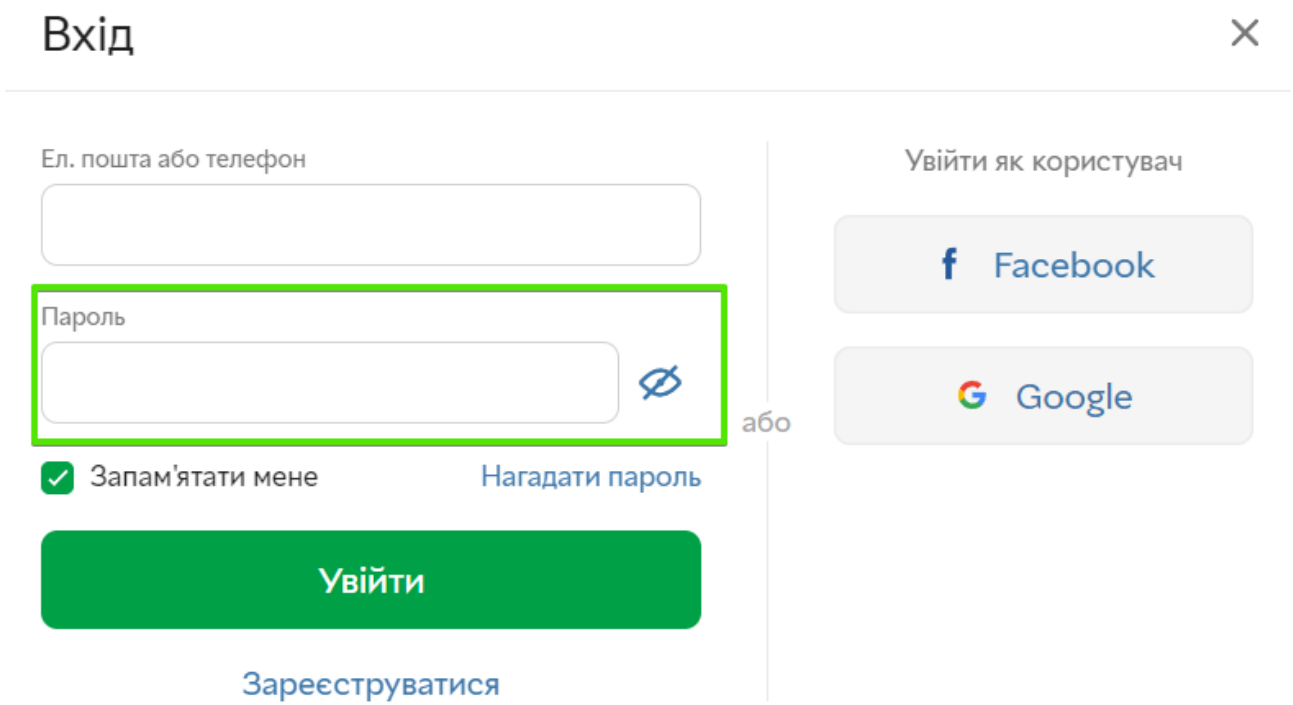

Рисунок 1.1 – Парольна аутентифікація [4]

Парольна аутентифікація є одним із найпоширеніших способів захисту інформаційних систем. Вона ґрунтується на застосуванні секретного набору символів, який відомий лише користувачу та системі. Для підвищення рівня безпеки паролі зберігаються у вигляді хеш-значень із використанням криптографічних алгоритмів.

Для забезпечення достатнього рівня безпеки пароль повинен містити щонайменше 6-8 символів, не бути простим для запам’ятовування та змінюватися не рідше одного разу на 90 днів.

Чим довший пароль, тим складніше його підібрати або обчислити зловмиснику. Збільшення довжини пароля навіть на один символ значно підвищує стійкість до атак.

1.2.2 Двофакторна аутентифікація

Двофакторна аутентифікація (рис. 1.2) базується на застосуванні двох різних способів підтвердження особи. Зазвичай вона поєднує пароль та одноразовий код підтвердження, який може надсилатися через SMS або генеруватися за допомогою спеціального програмного забезпечення. Такий метод значно зменшує ймовірність несанкціонованого доступу.

Рисунок 1.2 – Двофакторна аутентифікація [5]

«У двофакторної авторизації є ще й додаткові переваги. У разі несанкціонованої спроби входу в обліковий запис, ви отримаєте повідомлення, і тут же зможете змінити пароль, щоб надалі не турбуватися про те, що хтось заволодів вашими персональними даними» [5].

Для реалізації двофакторної аутентифікації можуть застосовуватися такі способи підтвердження:

– телефонний дзвінок для підтвердження особи;

– код, надісланий на електронну пошту;

– використання телефону для отримання одноразового SMS-коду.

1.2.3 Біометрична аутентифікація

Біометрія – це технологія розпізнавання особи за її фізичними характеристиками (наприклад, обличчя, відбитки пальців, сітківка ока) або поведінковими особливостями (такими як голос чи особливості ходи). Отже, «біометрична аутентифікація (рис. 1.3) – це процес організації безпеки, який ґрунтується на технології розпізнавання персони за її біологічними характеристиками. Їхня унікальність дає можливість підтвердити, чи є людина саме тим користувачем, за якого себе видає» [6].

Рисунок 1.3 – Біометрична аутентифікація [6]

Це є найпоширенішим типом біометричних систем. Існує три основні види такого пристрою:

– перетворення відбитка пальця в цифровий код за допомогою оптичного сенсора;

– обробка відбитка за допомогою лінійного теплового датчика;

– зчитування відбитка з використанням ємнісного датчика.

«Різниця полягає лише в тому, який тип взаємодії буде зі сканером. Людина може просто прикласти палець (так працює оптичний та місткісний) або провести їм по сенсору (це принцип теплового пристрою)» [6].

1.2.4 Апаратна аутентифікація

Апаратна аутентифікація (рис. 1.4) – це спосіб захисту, який передбачає використання фізичних пристроїв, для підтвердження особи користувача. Вона забезпечує вищий рівень безпеки порівняно з паролями, оскільки засіб підтвердження фізично знаходиться у власності користувача.

Рисунок 1.4 – Апаратна аутентифікація [7]

До основних різновидів апаратної аутентифікації належать:

– аутентифікація за допомогою смарт-карт – застосування спеціальних карт із вбудованим мікропроцесором, який зберігає криптографічні ключі та інші дані користувача;

– використання USB-токенів – підключення апаратних пристроїв через USB-інтерфейс, які містять захищене сховище для ідентифікаційної інформації;

– аутентифікація через апаратні модулі безпеки – застосування спеціалізованих криптографічних пристроїв для збереження ключів і виконання криптографічних операцій.

«Головним достоїнством застосування апаратної ідентифікації є досить висока надійність. У пам’яті токенів можуть зберігатися ключі, підібрати які хакерам не вдасться. Крім того, у них реалізовано чимало різних захисних механізмів» [3].

1.3 Аналіз сучасних систем аутентифікації

Аутентифікація користувача є основним елементом системи інформаційної безпеки, оскільки вона дозволяє встановити справжність суб’єкта доступу до інформаційних ресурсів. Завдяки механізмам автентифікації забезпечується регулювання прав доступу до системи, а також створюються умови для надійного захисту конфіденційної інформації від несанкціонованого використання.

Сучасні системи аутентифікації активно застосовуються у веб-додатках, корпоративних інформаційних системах та IoT-пристроях. Основною тенденцією розвитку є підвищення рівня безпеки при збереженні зручності використання.

У зв’язку зі зростанням кількості кіберзагроз та поширенням цифрових сервісів виникає необхідність у впровадженні більш надійних механізмів перевірки особи користувача. Традиційні методи аутентифікації, зокрема використання лише пароля, поступово втрачають ефективність через ризики підбору паролів, фішингових атак та витоку облікових даних. Саме тому сучасні системи безпеки все частіше використовують комбіновані підходи, які поєднують декілька факторів аутентифікації.

Одним із найбільш поширених підходів є багатофакторна аутентифікація (табл. 1.1), що передбачає використання декількох незалежних факторів підтвердження особи користувача.

Таблиця 1.1 – Фактори аутентифікації

Знання	Володіння	Біометрія

пароль, PIN-код	смартфон, токен, смарт-карта	відбиток пальця, обличчя

Поєднання цих факторів дозволяє значно підвищити рівень захисту інформаційних систем від несанкціонованого доступу.

Особливу роль у розвитку сучасних систем аутентифікації відіграє використання апаратних засобів безпеки. Апаратні токени, смарт-карти та криптографічні модулі забезпечують безпечне зберігання ключів та виконання криптографічних операцій без їхнього розкриття програмному середовищу. Це значно знижує ризик компрометації даних навіть у випадку зараження системи шкідливим програмним забезпеченням.

Також поширюється використання систем єдиного входу (Single Sign-On), які дозволяють користувачу отримувати доступ до декількох сервісів після одноразової аутентифікації. «Single Sign-On (рис. 1.5) – технологія, що дозволяє користувачеві переходити з одного онлайн-сервісу до іншого, без повторної автентифікації, за єдиним ID» [8].

Використання технології Single Sign-On дозволяє спростити процес взаємодії користувача з інформаційними системами та підвищити зручність роботи із цифровими сервісами. Крім цього, централізований механізм аутентифікації забезпечує ефективніше управління доступом користувачів та знижує ймовірність використання слабких або повторюваних паролів.

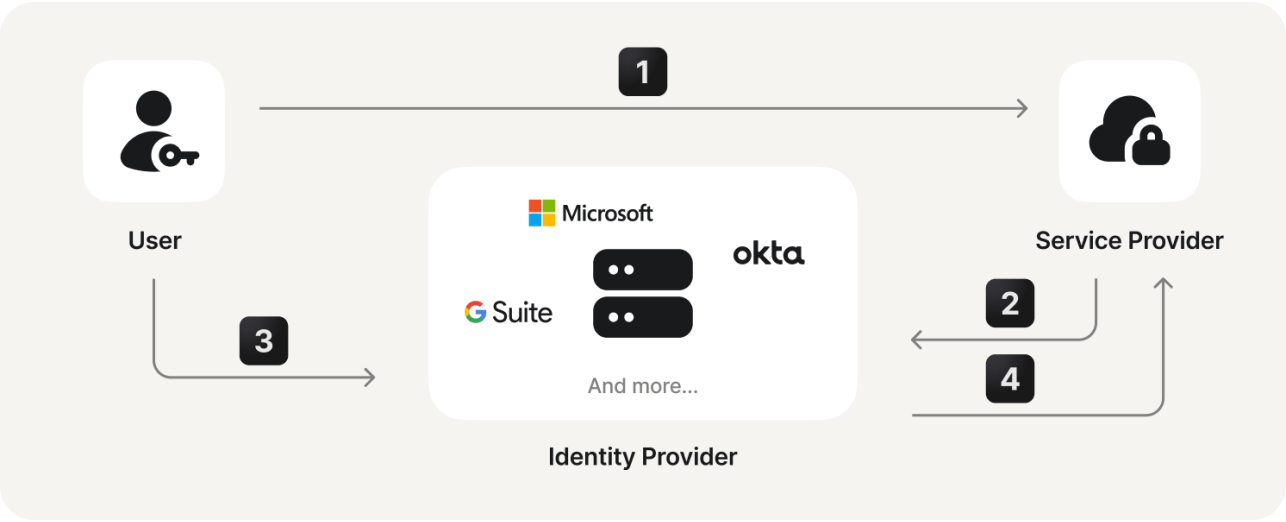

Рисунок 1.5 – Single Sign On [9]

Крім того, з розвитком технологій Інтернету речей зростає потреба у впровадженні надійних механізмів аутентифікації для вбудованих пристроїв. Такі пристрої часто мають обмежені обчислювальні ресурси, тому системи аутентифікації повинні бути не лише безпечними, але й оптимізованими з точки зору продуктивності та енергоспоживання.

Таким чином, сучасні системи аутентифікації спрямовані на поєднання високого рівня безпеки, зручності використання та адаптивності до різних типів інформаційних систем і пристроїв. Це дозволяє ефективно протидіяти сучасним кіберзагрозам та забезпечувати надійний захист інформаційних ресурсів.

1.4 Аналіз апаратної платформи Raspberry Pi

Одноплатний комп’ютер Raspberry Pi (рис. 1.6) є сучасною апаратною платформою, яка широко застосовується для створення вбудованих та мережевих систем, зокрема у сфері Інтернету речей та систем інформаційної безпеки. Завдяки компактним розмірам, низькому енергоспоживанню та достатній обчислювальній потужності, дана платформа є доцільним вибором для реалізації системи аутентифікації користувачів з використанням веб-інтерфейсу.

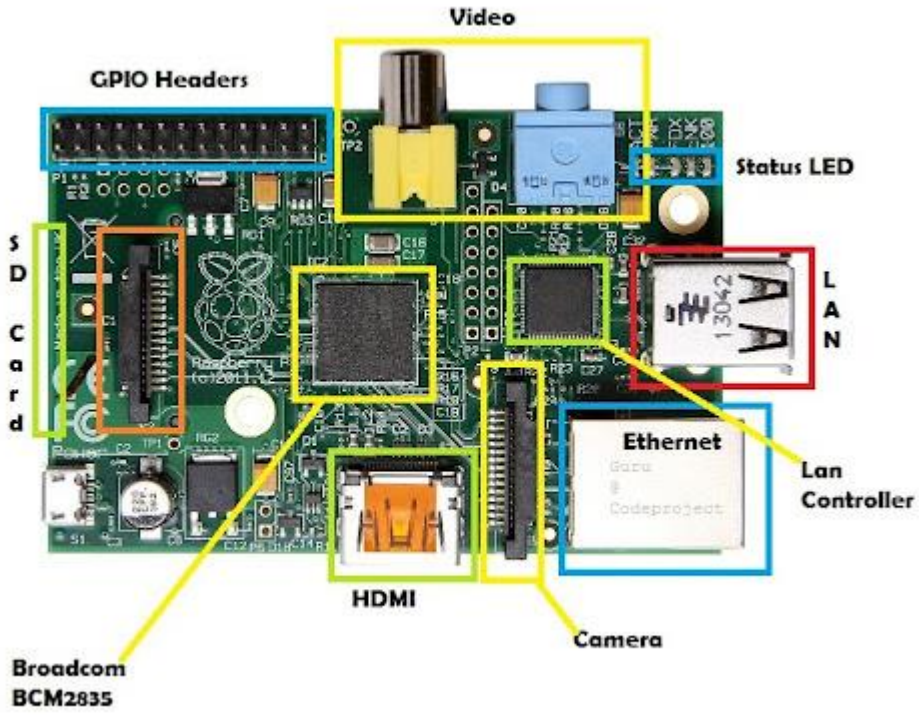

Рисунок 1.6 – Одноплатний комп’ютер Raspberry Pi [10]

Raspberry Pi являє собою компактний комп’ютер, що поєднує в собі процесор, оперативну пам’ять, мережеві модулі та інтерфейси введення-виведення. «По суті це дуже дешевий комп’ютер, що працює під управлінням Linux, але він також має набір виводів GPIO (введення/виведення загального призначення), що дають змогу керувати електронними компонентами для фізичних обчислень і досліджувати Інтернет речей (IoT). На платі Raspberry Pi розташовані процесор, пам’ять, порти LAN, USB і micro HDMI, а також слот для карти пам’яті micro SD» [11].

«Ініціатором проекту Raspberry Pi є британський благодійний фонд Raspberry Pi Foundation. Комп’ютер планувався як пристрій для навчання дітей програмуванню, однак здобув популярність і в інших сферах – зокрема, на його основі роблять домашні медіацентри» [12].

Найбільш поширеною моделлю для практичних задач є Raspberry Pi 4, яка має достатню продуктивність для обробки запитів у реальному часі. В таблиці 1.2 наведемо технічні характеристики Raspberry Pi.

Таблиця 1.2 – Технічні характеристики Raspberry Pi

Особливість	Специфікація
Розмір	Розміром з кредитну картку
Процесор	ARM
Оперативна пам’ять	512 MB
Порт Ethernet	Так
Порти USB	Два
Вихідний відеосигнал	HDMI та композитний RCA
Звуковий вихід	3,5 мм роз’єм
Піни GPIO	26-контактний роз’єм з відстанню 0,1 дюйма
Вимоги до операційної системи	SD-карта з ОС (не входить до комплекту)

Платформа підтримує операційні системи на базі Linux, що дозволяє використовувати сучасні інструменти для розробки веб-додатків та серверних систем (табл. 1.3). Це забезпечує можливість створення повноцінного веб-сервера без необхідності використання дорогого обладнання.

Таблиця 1.3 – Програмні засоби, що можуть використовуватися на Raspberry Pi

Категорія	Приклади
Операційна система	Raspberry Pi OS, Ubuntu
Веб-сервер	Nginx, Apache
Мова програмування	Python, JavaScript (Node.js), PHP
База даних	MySQL, PostgreSQL, SQLite

Однією з ключових особливостей Raspberry Pi є наявність GPIO-портів, які дозволяють підключати зовнішні пристрої (табл. 1.4). Це особливо важливо для реалізації апаратної аутентифікації, оскільки дає змогу інтегрувати різні сенсори та модулі доступу.

Таблиця 1.4 – Апаратні модулі для системи аутентифікації

Тип модуля	Призначення
RFID-зчитувач	Безконтактна ідентифікація користувача
Сканер відбитків	Біометрична аутентифікація
Камера	Розпізнавання обличчя
Клавіатура / keypad	Введення PIN-коду
USB-токен	Апаратна ідентифікація

Платформа Raspberry Pi має ряд переваг, зокрема низьку вартість, компактність та енергоефективність, що робить її зручною для використання у вбудованих системах. Підтримка операційних систем на базі Linux забезпечує широкі можливості для розробки веб-додатків, а наявність GPIO-інтерфейсів дозволяє підключати різноманітні апаратні модулі для реалізації аутентифікації.

Разом із тим, платформа має і певні недоліки. До них належать обмежена продуктивність у порівнянні з серверними рішеннями, залежність від microSD як носія даних та обмежені апаратні ресурси. Проте для систем середнього рівня складності ці обмеження не є критичними.

Отже, використання Raspberry Pi як апаратної основи є обґрунтованим вибором, що забезпечує оптимальне поєднання вартості, функціональності та ефективності.

РОЗДІЛ 2
РОЗРОБКА АПАРАТНОЇ ЧАСТИНИ МОБІЛЬНОЇ ПЛАТФОРМИ

2.1 Вибір та обґрунтування апаратних компонентів

Апаратна частина розроблюваної системи аутентифікації включає одноплатний комп’ютер, пристрій зчитування ідентифікаційних даних та носій ідентифікації користувача. У якості базових компонентів обрано платформу Raspberry Pi 3, NFC-зчитувач ACS ACR1252U III USB та NFC-картку стандарту MIFARE. Така конфігурація забезпечує реалізацію сучасної безконтактної системи доступу з високим рівнем безпеки та зручністю використання.

2.1.1 Мікрокомп’ютер Raspberry Pi 3 Model B

Raspberry Pi 3 Model B (рис. 2.1) – це компактний одноплатний мінікомп’ютер, габарити якого лише трохи більші за кредитну картку. Дана модель відзначається високою продуктивністю та розглядається виробником як універсальне рішення для реалізації проєктів різного рівня складності – від простих до складних. Він забезпечує виконання серверної логіки, обробку даних аутентифікації та взаємодію з периферійними пристроями.

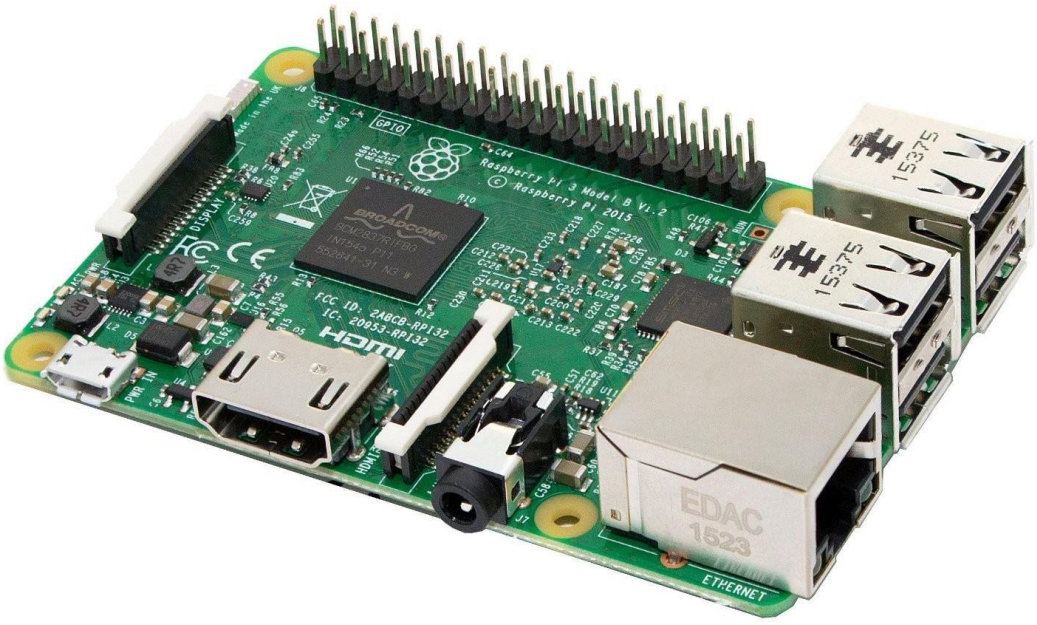

Рисунок 2.1 – Мікрокомп’ютер Raspberry Pi 3 Model B [13]

«До цього комп’ютера можна підключити величезну кількість зовнішніх пристроїв, завдяки чому на його базі можна побудувати як проект в області робототехніки, так і повноцінний компактний комп’ютер. Просто додайте клавіатуру, мишу, дисплей, блок живлення, карту micro SD із встановленим дистрибутивом Linux, і у вас буде повноцінний комп’ютер, який зможе запускати додатки» [14].

Raspberry Pi 3 Model B має такі основні інтерфейси:

– процесор: Broadcom BCM2837 (SoC) на базі ARM Cortex-A53;

– графічний процесор: двоядерний VideoCore IV з тактовою частотою 400 МГц;

– графічні можливості: підтримка OpenGL ES 2.0, апаратне прискорення OpenVG, декодування відео у форматі H.264 (1080p30, high-profile);

– оперативна пам’ять: 1 ГБ LPDDR2;

– носій даних: слот для карт пам’яті MicroSD;

– підтримувані операційні системи: NOOBS, Raspbian, LibreELEC, OpenELEC, OSMC, Pinet, RISC OS, Snappy Ubuntu Core, Ubuntu Mate, Windows IoT Core, Xbian та інші;

– інтерфейси та роз’єми: 40-контактний GPIO, 15-контактний інтерфейс MIPI CSI для камери, HDMI (версії 1.3 і 1.4), інтерфейс дисплея DSI, чотири порти USB 2.0, RCA (PAL/NTSC), аудіовихід 3,5 мм;

– відеовихід: HDMI;

– мережеві можливості: Ethernet 10/100Base-T через роз’єм RJ-45.

Одноплатний комп’ютер Raspberry Pi 3 Model B є універсальною обчислювальною платформою, яка поєднує компактність, енергоефективність та достатню продуктивність для реалізації вбудованих і мережевих систем. Його архітектура базується на системі-на-кристалі Broadcom BCM2837, що включає чотириядерний процесор ARM Cortex-A53 із тактовою частотою 1,2 ГГц. Така конфігурація забезпечує стабільну роботу прикладних програм, обробку запитів користувачів та виконання серверних задач невеликого масштабу.

Вбудовані модулі бездротового зв’язку Wi-Fi стандарту 802.11 b/g/n та Bluetooth 4.1 значно розширюють функціональні можливості пристрою. Завдяки цьому забезпечується можливість підключення до локальних мереж, організації віддаленого доступу до системи, а також взаємодії з мобільними пристроями або іншими IoT-компонентами без використання додаткових адаптерів.

Живлення пристрою здійснюється через роз’єм micro-USB із рекомендованими параметрами 5 В та 2,5 А. Енергоспоживання плати є відносно низьким і залежить від навантаження, що дозволяє використовувати її у системах із обмеженими енергетичними ресурсами або в автономному режимі. Така характеристика є важливою для систем контролю доступу, які повинні працювати безперервно.

Програмна складова Raspberry Pi 3 Model B базується на підтримці широкого спектра операційних систем, серед яких найбільш популярною є Raspberry Pi OS. Крім того, можливе використання інших дистрибутивів Linux, що дозволяє застосовувати сучасні засоби розробки програмного забезпечення, зокрема мови програмування Python, JavaScript (Node.js), а також бібліотеки для роботи з мережами та криптографією. Це створює сприятливі умови для реалізації веб-інтерфейсу та серверної логіки системи аутентифікації.

Важливою особливістю плати є наявність універсального інтерфейсу GPIO, який забезпечує пряме підключення зовнішніх модулів. Це дозволяє інтегрувати додаткові пристрої, такі як зчитувачі карт, сенсори або індикатори, що значно розширює функціональні можливості системи. У контексті даної роботи це відкриває можливість підключення NFC-зчитувача та реалізації апаратної складової аутентифікації.

Таким чином, Raspberry Pi 3 Model B є збалансованим рішенням для реалізації системи аутентифікації користувачів. Поєднання достатньої продуктивності, підтримки мережевих технологій, можливості інтеграції з апаратними модулями та низького енергоспоживання робить дану платформу оптимальним вибором для поставлених завдань.

2.1.2 NFC-зчитувач ACS

NFC-зчитувач ACS ACR1252U III USB (рис. 2.2) призначений для безконтактного зчитування та запису даних із NFC-карт і пристроїв. Він працює на частоті 13,56 МГц та підтримує сучасні стандарти NFC і RFID.

Рисунок 2.2 – NFC-зчитувач ACS ACR1252U III USB [15]

«Він має слот SAM (модуль безпечного доступу), який можна використовувати разом з карткою SAM для диверсифікації ключів та взаємної автентифікації, забезпечуючи високий рівень безпеки безконтактних транзакцій. Також підтримується оновлення прошивки після розгортання, що усуває необхідність додаткової модифікації обладнання» [16].

В таблиці 2.1 наведемо технічні характеристики NFC зчитувача ACR 1252 U USB III

Таблиця 2.1 – Характеристики NFC зчитувача ACR 1252 U USB III

Параметр	Характеристика
Інтерфейс	ISO14443 тип A та B, Mifare, FeliCa, 4 типи NFC міток (ISO/IEC18092)
Операційна система	Windows®, Linux®, Mac OS®, Android™ 3.1 та вище
Сертифікати/відповідність	ISO 18092, ISO 14443, ISO 7816, NFC Forum, FeliCa Performance Certification, PC/SC, CCID, LASCOM, EN 60950 / IEC 60950, CE, FCC, VCCI, MIC, KC, RoHS 2, USB Full Speed, Microsoft®
Дальність зчитування	до 50 мм (залежно від типу мітки)
Розміри	98 мм x 65 мм x 12,8 мм
Вага	81 г
Комунікаційний інтерфейс	USB 2.0 Full Speed
Робоча частота	13,56 МГц
Напруга живлення	5 В (регульований)
Струм споживання	200 мА (максимальний)
Робоча температура	від 0° С до +50° С
Індикація	Звукова та світлова індикація

Основні переваги NFC-зчитувача ACR1252U USB III:

– наявність комунікаційного інтерфейсу USB 2.0 Full Speed;

– відповідність стандарту CCID;

– можливість оновлення вбудованого програмного забезпечення через USB;

– швидкість операцій читання та запису до 424 кбіт/с;

– інтегрована антена для безконтактної роботи з картами та мітками на відстані до 50 мм (залежно від їх типу);

– підтримка смарт-карт і міток стандартів ISO14443 A і B, MIFARE, FeliCa, ISO/IEC 18092 (усі чотири типи NFC-міток);

– сумісність із MIFARE (UID довжиною 7 байт), а також MIFARE Plus і MIFARE DESFire;

– наявність вбудованого механізму антиколізії;

– інтегрований слот SAM для карт відповідно до стандарту ISO7816 (клас A);

– підтримка NFC-режимів: читання/запис, піринговий режим і емуляція картки;

– сумісність з інтерфейсами PC/SC та CT-API;

– наявність двох програмованих світлодіодів і програмованого звукового сигналізатора.

Вибір зчитувача ACS ACR1252U III USB обумовлений його високою сумісністю з сучасними стандартами NFC, що забезпечує універсальність у використанні. Підтримка протоколів PC/SC дозволяє легко інтегрувати пристрій у програмне середовище Raspberry Pi без необхідності розробки низькорівневих драйверів.

Наявність апаратних механізмів безпеки, таких як SAM-модуль, підвищує рівень захисту системи аутентифікації, що є критично важливим для обробки персональних даних користувачів. Крім того, простота підключення та стабільність роботи роблять цей зчитувач оптимальним рішенням для розроблюваної системи.

Таким чином, використання ACR1252U III USB дозволяє забезпечити надійну, швидку та безпечну взаємодію з NFC-картами, що є ключовим елементом реалізації апаратної аутентифікації у даній роботі.

2.1.3 NFC-картка

NFC-картка (рис. 2.3) є носієм ідентифікаційної інформації користувача. Вона працює за стандартом ISO/IEC 14443 і використовується для безконтактної передачі даних.

Рисунок 2.3 – NFC-карта RFID 13.56 МГц [17]

«Картки частоти 13,56 МГц виконані по ISO14443A популярні завдяки наявності шифрування і можливості запису додаткової інформації у пам’ять. Кожна картка має свій унікальний UID-номер. На відміну від EM-Marine, картку Mifare практично неможливо скопіювати, що є ключовою перевагою для систем контролю доступу. Дана картка допускає нанесення текстового і графічного зображення на поверхню за технологією термодрук, офсетний друк або шовкографія» [18].

Окрім UID, картка оснащена пам’яттю обсягом 1 КБ, яка може використовуватися для зберігання користувацьких даних. Зокрема, це можуть бути ПІБ користувача, номер облікового запису, баланс, термін дії картки або навіть шаблони біометричної інформації.

Наведемо технічні характеристики NFC-карти RFID в таблиці 2.2

Таблиця 2.2 – Характеристики NFC-карти RFID

Параметр	Характеристика
Чіп	Philips Mifare 1 S50
Ємність	8Kbit
Габарити розмір	30x40x3 мм

Продовження таблиці 2.2

Робоча частота	13,56 MHz
Швидкість обміну	106K Boud
Дистанція читання/запис	2,5 ~ 10 см
Час читання/запис	1 ~ 2 мс
Робоча температура	-20-85 C
Кількість циклів перезапису	>100,000

NFC-картки обрані як засіб ідентифікації через їх широке використання у системах контролю доступу, швидкість роботи та зручність для користувача. Вони легко інтегруються з обраним NFC-зчитувачем і забезпечують достатній рівень безпеки для даної системи.

2.2 Архітектура системи аутентифікації

Архітектура розроблюваної системи аутентифікації користувачів побудована за клієнт-серверним (рис. 2.4) принципом із використанням апаратного модуля ідентифікації. Основною метою такої архітектури є забезпечення надійної, масштабованої та зручної взаємодії між користувачем, апаратною частиною та серверною логікою.

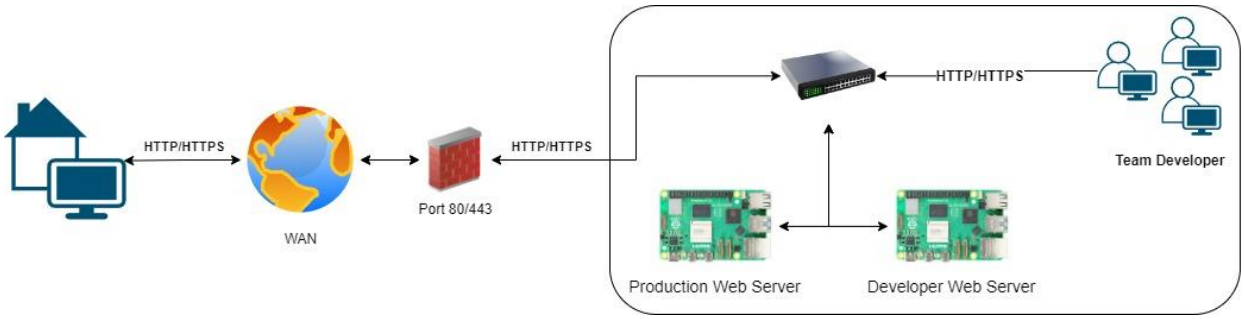

Рисунок 2.4 – Приклад веб-серверу Raspberry Pi [19]

Система складається з трьох основних рівнів: апаратного, серверного та клієнтського. Кожен із рівнів виконує окремі функції, що дозволяє розділити

обробку даних, підвищити ефективність роботи та спростити подальше розширення системи.

Апаратний рівень включає одноплатний комп’ютер Raspberry Pi 3 Model B та NFC-зчитувач ACS ACR1252U III USB, які разом забезпечують процес ідентифікації користувача. NFC-картка виступає носієм унікального ідентифікатора.

Після піднесення картки до зчитувача відбувається зчитування її UID або інших даних, які передаються до Raspberry Pi через USB-інтерфейс. Далі ці дані обробляються серверною частиною системи.

Серверний рівень реалізується безпосередньо на Raspberry Pi та відповідає за обробку запитів, аутентифікацію користувачів і взаємодію з базою даних. На цьому рівні функціонують:

– веб-сервер;

– база даних користувачів;

– модуль обробки NFC-даних;

– модуль безпеки (хешування, перевірка доступу).

Після отримання даних із NFC-зчитувача сервер виконує перевірку ідентифікатора користувача у базі даних. У разі успішної аутентифікації система надає доступ або виконує відповідну дію.

Клієнтський рівень представлений веб-інтерфейсом, який забезпечує взаємодію користувача з системою. Через браузер користувач може:

– реєструвати нові картки;

– переглядати список користувачів;

– керувати правами доступу;

– переглядати журнал подій.

Веб-інтерфейс взаємодіє із сервером через HTTP/HTTPS-запити, що забезпечує зручність використання та незалежність від конкретного пристрою.

Процес аутентифікації у системі відбувається у такій послідовності:

– користувач підносить NFC-картку до зчитувача;

– зчитувач отримує ідентифікатор картки та передає його до Raspberry Pi;

– серверна частина обробляє отримані дані;

– виконується пошук відповідного запису у базі даних;

– у разі збігу користувач вважається автентифікованим;

– результат відображається у веб-інтерфейсі або виконується відповідна дія (наприклад, відкриття доступу).

Розроблена архітектура системи аутентифікації забезпечує ефективну взаємодію між апаратними та програмними компонентами. Вона поєднує можливості безконтактної ідентифікації з гнучкістю веб-технологій, що дозволяє створити надійну та зручну систему контролю доступу.

2.3 Вибір та обґрунтування програмної складової системи

Одним із важливих етапів розробки системи аутентифікації користувачів є вибір програмних засобів, які забезпечують реалізацію функціональних можливостей системи, взаємодію між її компонентами, обробку інформації та управління доступом користувачів. Вибір програмної складової здійснювався з урахуванням вимог до продуктивності, безпеки, масштабованості, стабільності роботи та можливості інтеграції з апаратними компонентами системи.

Для реалізації серверної частини системи було обрано мову програмування PHP та фреймворк Laravel. «Використання PHP обумовлено його широким застосуванням у сфері веб-розробки, високою продуктивністю при обробці серверних запитів та можливістю ефективної інтеграції з реляційними базами даних. Крім того, PHP має велику кількість бібліотек та засобів для реалізації механізмів аутентифікації, роботи з API та управління користувацькими сесіями» [20].

Фреймворк Laravel було обрано як основу серверної логіки системи завдяки його сучасній архітектурі, високому рівню структурованості коду та наявності вбудованих механізмів для реалізації веб-застосунків. Laravel реалізує архітектурний шаблон MVC (Model-View-Controller). «MVC – це архітектурний шаблон, який використовується під час проєктування та розробки програмного

забезпечення. Цей шаблон передбачає поділ системи на три взаємопов’язані частини: модель даних, вигляд (інтерфейс користувача) та модуль керування» [21]. Такий підхід дозволяє спростити процес розробки, підвищити зручність супроводу програмного забезпечення та забезпечити його подальшу масштабованість.

Для забезпечення роботи веб-застосунку та обробки HTTP-запитів у системі використовується веб-сервер Apache HTTP Server. «Даний сервер призначений для передачі веб-вмісту, такого як HTML-сторінки, зображення та мультимедіа, через Інтернет до браузерів користувачів. Він робить це, отримуючи запити клієнта (зазвичай через браузер) і надаючи відповіді (веб-сторінки) назад клієнту. Він використовує протоколи HTTP і HTTPS для забезпечення безперебійного зв’язку між сервером і клієнтом» [22]. Використання Apache обумовлено його стабільністю, широкими можливостями конфігурації та високою сумісністю із PHP та Laravel.

Для управління залежностями та зовнішніми бібліотеками у проєкті використовується Composer. Застосування даного інструмента дозволяє автоматизувати встановлення програмних пакетів, керувати їх версіями та забезпечувати швидке розгортання програмного середовища. Використання Composer також спрощує процес оновлення компонентів системи та підтримки її актуального стану.

Для організації зберігання інформації про користувачів, NFC-картки, права доступу та журнал подій використовується система керування базами даних MySQL. Використання MySQL забезпечує ефективне зберігання та швидку обробку даних, а також підтримку складних SQL-запитів. Реляційна модель бази даних дозволяє забезпечити цілісність інформації та логічний зв’язок між таблицями системи.

Обмін даними між апаратною частиною системи та серверною логікою реалізовано за допомогою REST API. Використання API забезпечує передачу ідентифікаційних даних, отриманих від NFC-зчитувача, у вигляді HTTP-запитів.

Такий підхід дозволяє розділити апаратну та програмну складові системи, що підвищує гнучкість архітектури та спрощує її подальшу модернізацію.

Для реалізації адміністративного інтерфейсу системи використовується «Filament – панель адміністрування на основі стеку для застосунків Laravel. Використання даного інструмента для більш спрощеного, інтуїтивного та зручного для розробників способу створення панелей адміністративів і форм, використовуючи потужність сучасних веб-технологій» [23]. Це скорочує час розробки адміністративної частини та спрощує процес управління системою.

Таким чином, обраний комплекс програмних засобів забезпечує повноцінну реалізацію системи аутентифікації користувачів, включаючи обробку запитів, управління даними, взаємодію з апаратними компонентами та адміністрування системи. Використання сучасних програмних технологій забезпечує високий рівень продуктивності, безпеки та масштабованості, що є важливими характеристиками для ефективного функціонування системи.

РОЗДІЛ 3
РЕАЛІЗАЦІЯ ТА СИСТЕМИ АУТЕНТИФІКАЦІЇ НА БАЗІ RASPBERRY PI

3.1 Встановлення та налаштування операційної системи і програмного середовища

Для реалізації системи аутентифікації користувачів першочергово є підготовка програмно-апаратного середовища, що включає встановлення операційної системи на платформу Raspberry Pi, налаштування базових параметрів системи та підготовку серверного середовища для розгортання веб-застосунку.

Як апаратну платформу для реалізації системи було використано Raspberry Pi 3 Model B, який виконує роль центрального вузла обробки даних. Для забезпечення стабільної роботи системи на пристрій було встановлено операційну систему Raspberry Pi OS, яка є офіційною операційною системою для пристроїв Raspberry Pi та базується на Linux.

Встановлення операційної системи здійснювалося за допомогою програмного забезпечення «Raspberry Pi Imager – це швидкий та простий спосіб встановити ОС Raspberry Pi та інші операційні системи на карту microSD, готову до використання з Raspberry Pi» [24]. Даний інструмент значно спрощує процес інсталяції та дозволяє виконати попереднє налаштування системи ще до першого запуску.

На першому етапі у програмі Raspberry Pi Imager було обрано модель пристрою Raspberry Pi 3 (рис. 3.1), після чого виконано вибір операційної системи та накопичувача для запису образу.

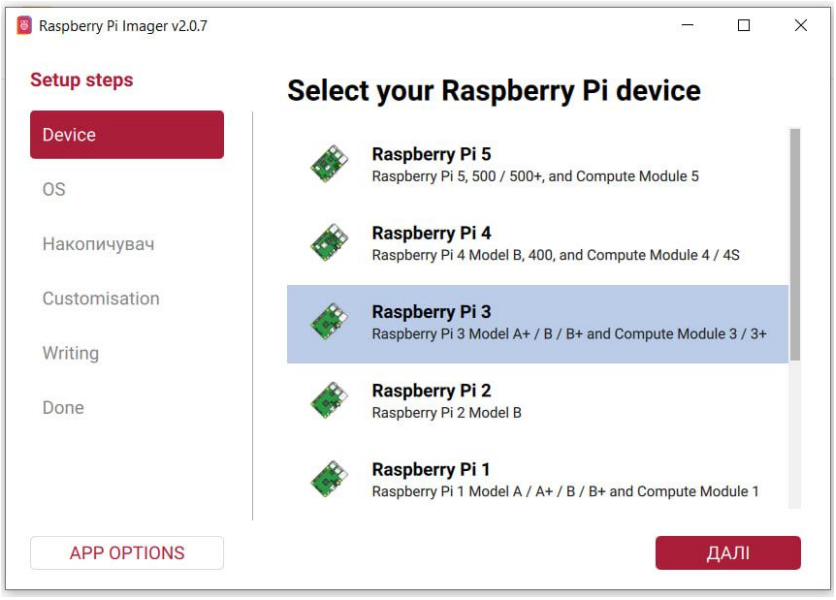

Рисунок 3.1 – Вибір моделі пристрою у Raspberry Pi Imager

Перед записом образу операційної системи на карту пам’яті необхідно виконати її попередню підготовку. Для цього здійснюється форматування microSD-карти, яка використовується як основний носій даних для Raspberry Pi 3 Model B. Форматування дозволяє видалити попередні дані, очистити файлову систему та підготувати накопичувач до коректного запису нового образу операційної системи. На рисунку 3.2 показано етап вибору режиму форматування у програмі Raspberry Pi Imager, де обрано пункт «Видалити».

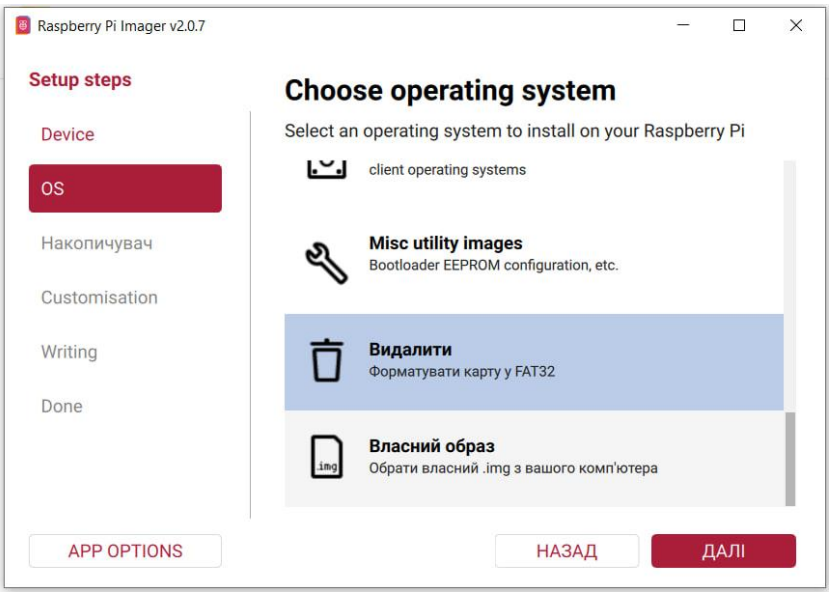

Рисунок 3.2 – Форматування microSD-карти перед записом операційної системи

На рисунку 3.3 представлено вибір накопичувача SDHC Card, який буде використовуватись для форматування та подальшого запису операційної системи.

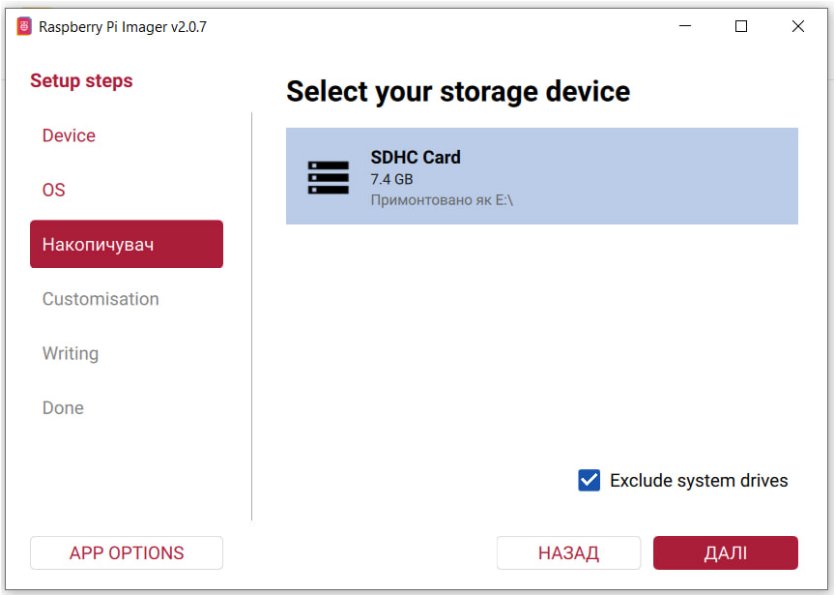

Рисунок 3.3 – Вибір накопичувача для форматування

Після вибору параметрів програма на (рис. 3.4) відображає підсумкову інформацію про модель пристрою, режим форматування та вибраний накопичувач. Після перевірки даних запускається процес форматування карти пам’яті.

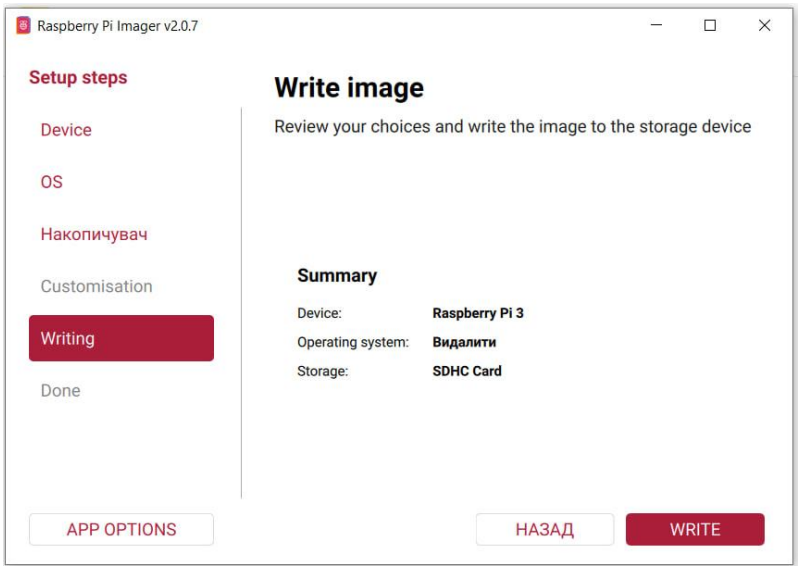

Рисунок 3.4 – Підтвердження параметрів форматування

Після завершення форматування на (рис. 3.5) карта пам’яті готова до запису образу операційної системи. Такий підхід дозволяє уникнути помилок під час

встановлення системи та забезпечує стабільну роботу пристрою після першого запуску.

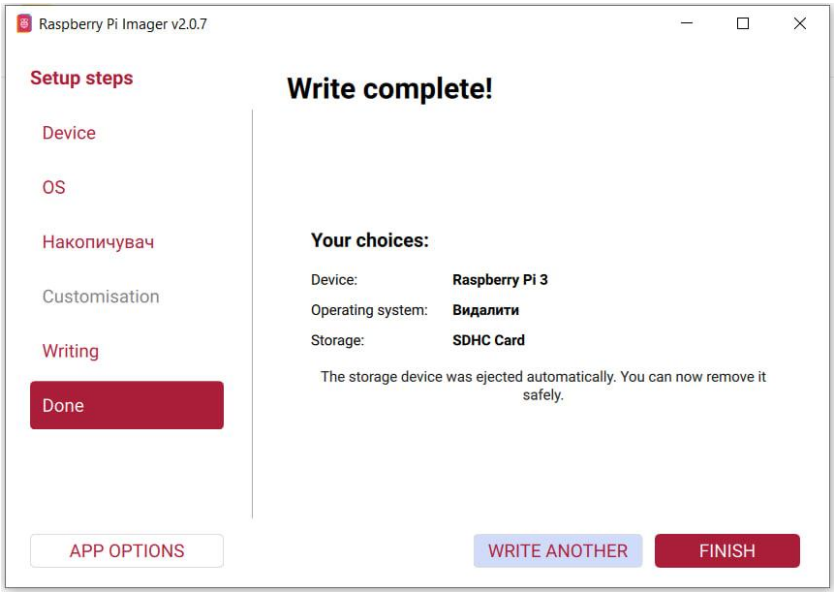

Рисунок 3.5 – Повідомлення про успішне завершення форматування

На другому етапі у програмному середовищі Raspberry Pi Imager здійснюється вибір накопичувача, на який буде виконано запис операційної системи. У якості носія використовується карта пам’яті microSD, яка попередньо була відформатована та підготовлена до встановлення системи. Після вибору накопичувача запускається процес запису образу операційної системи, під час якого всі необхідні системні файли копіюються на карту пам’яті (рисунок 3.6). Даний етап є важливим, оскільки саме на цьому носії зберігатиметься операційна система, системні налаштування та програмне забезпечення, необхідне для подальшої реалізації та функціонування системи аутентифікації користувачів.

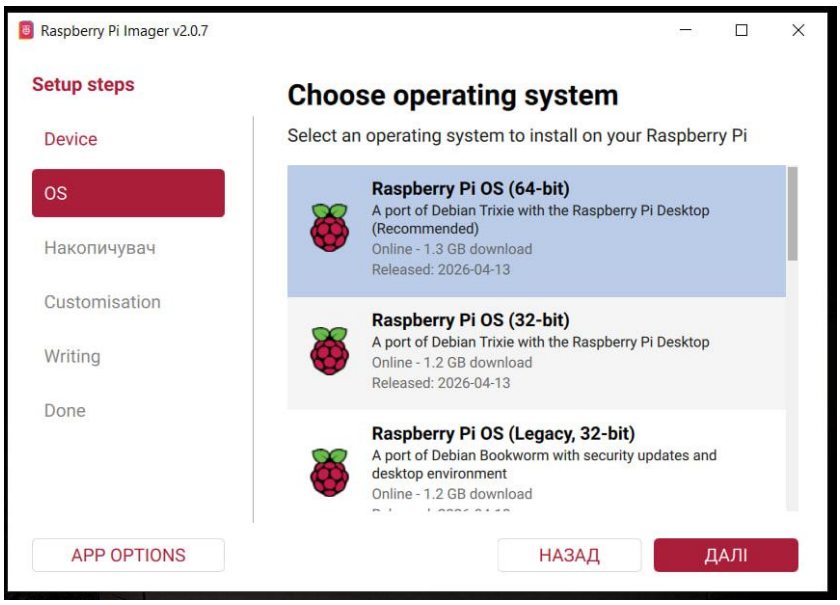

Рисунок 3.6 – Вибір операційної системи у Raspberry Pi Imager

Після завершення процесу запису образу операційної системи карта пам’яті microSD вилучається з комп’ютера та встановлюється у відповідний слот плати Raspberry Pi 3 Model B. Після підключення живлення пристрій автоматично розпочинає завантаження операційної системи з підготовленого носія.

Під час першого запуску системи виконується початкове налаштування основних параметрів, що включає вибір мови інтерфейсу, часової зони, налаштування мережевого підключення та створення облікового запису адміністратора. Даний етап є важливим для забезпечення коректної роботи операційної системи та підготовки пристрою до встановлення серверного програмного забезпечення.

Після завершення базового налаштування було виконано оновлення системних пакетів та програмних компонентів, що дозволило забезпечити актуальність встановленого програмного забезпечення, підвищити стабільність роботи системи та усунути можливі проблеми сумісності.

3.2 Налаштування Raspberry Pi

Підготовка програмного середовища Raspberry Pi розпочинається з оновлення локальної бази пакетів операційної системи. Виконання команди sudo apt update (рис. 3.7) забезпечує синхронізацію інформації з офіційними репозиторіями Debian та Raspberry Pi OS, що дозволяє отримати актуальні версії програмного забезпечення та системних компонентів. Проведення цієї операції є важливим етапом налаштування серверного середовища, оскільки гарантує сумісність пакетів і підвищує стабільність подальшої роботи системи. Після завершення перевірки система відображає перелік доступних оновлень та інформацію про стан пакетного менеджера.

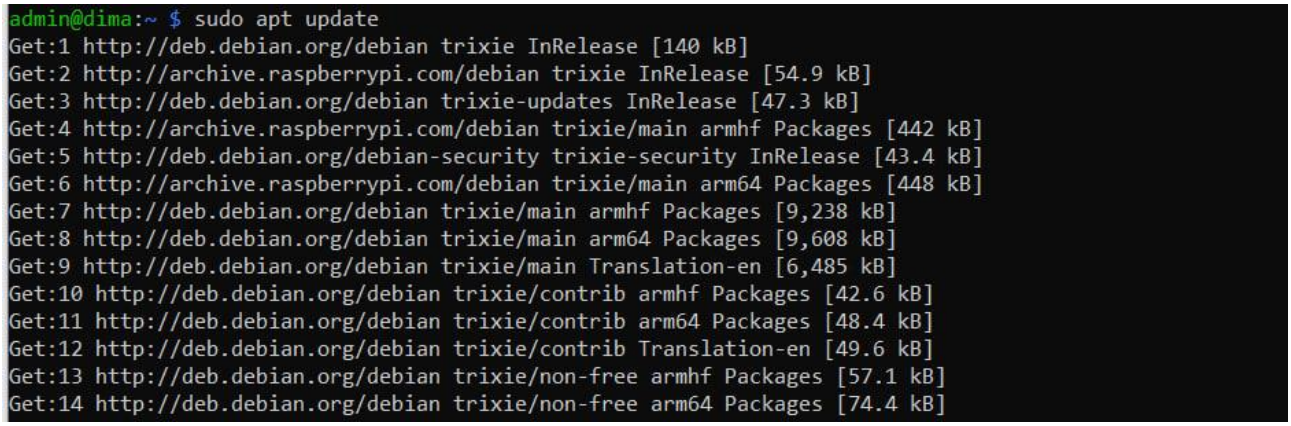

Рисунок 3.7 – Оновлення списку системних пакетів у Raspberry Pi OS.

Наступним кроком налаштування є встановлення вебсервера Apache2 (рис. 3.8), який забезпечує функціонування серверної частини вебзастосунку. Інсталяція виконувалася засобами пакетного менеджера APT із автоматичним підключенням необхідних бібліотек, службових модулів та допоміжних компонентів. Apache2 виконує обробку HTTP-запитів користувачів, організовує передачу вебсторінок клієнтським пристроям і підтримує роботу локального серверного середовища. Використання цього вебсервера є одним із базових рішень під час розгортання веборієнтованих інформаційних систем.

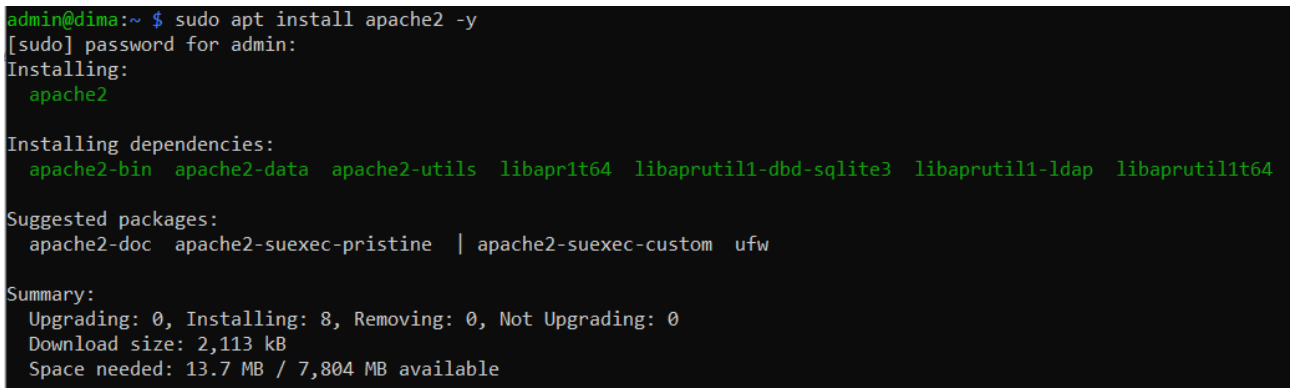

Рисунок 3.8 – Інсталяція вебсервера Apache2 у середовищі Raspberry Pi OS.

Завершальною частиною налаштування серверного середовища є встановлення інтерпретатора PHP, який забезпечує підтримку виконання серверних сценаріїв та формування динамічного вебконтенту. Інсталяція здійснювалася командою sudo apt install php –y (рис. 3.9), що дозволило автоматизувати процес підтвердження встановлення пакетів. Після інтеграції PHP із вебсервером Apache2 система отримує можливість обробки даних користувача, взаємодії з базами даних і реалізації функціональних можливостей вебзастосунку. Така конфігурація формує повноцінне програмне середовище для розробки та експлуатації серверних вебтехнологій.

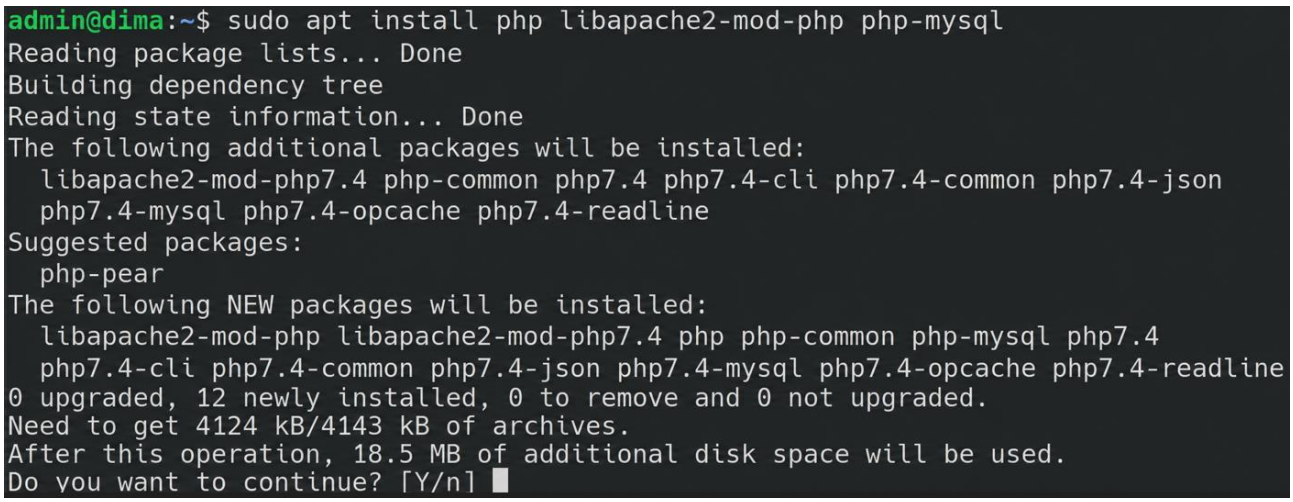

Рисунок 3.9 – Встановлення інтерпретатора PHP у серверному середовищі.

3.3 Реалізація адміністративної панелі та тестування системи

3.3.1 Реалізація функцій адміністративної панелі

Для реалізації механізму керування API-токенами користувачів у адміністративній панелі системи було створено окрему дію, яка дозволяє виконувати повторну генерацію токена доступу. Даний функціонал реалізовано у класі ресурсу користувачів адміністративної панелі на базі Filament.

Метод getHeaderActions() повертає масив дій, що відображаються у верхній частині сторінки керування користувачем. Однією з таких дій є кнопка regenerate_token, призначена для створення нового API-токена користувача. Для підвищення зручності використання елементу інтерфейсу встановлено текстову назву «Перегенерувати токен», графічну іконку та кольорове оформлення типу warning, що візуально виділяє дію серед інших елементів інтерфейсу.

Перед виконанням операції передбачено механізм підтвердження дії за допомогою методу requiresConfirmation(). Це дозволяє уникнути випадкового перегенерування токена та підвищує безпечність роботи адміністративної панелі.

Основна логіка реалізована у функції action(), яка виконується після підтвердження дії користувачем. На початку виконання отримується поточний об’єкт користувача через змінну $this->record. Після цього здійснюється видалення всіх наявних токенів користувача методом tokens()->delete(). Такий підхід забезпечує деактивацію попередніх токенів доступу та унеможливлює їх подальше використання.

Далі виконується генерація нового токена за допомогою методу createToken(‘api’). Створений токен повертається у відкритому вигляді через властивість plainTextToken та зберігається у змінній $token. Після генерації нове значення токена записується у поле token моделі користувача за допомогою методу forceFill() із подальшим збереженням змін у базі даних методом save() (лістинг 3.1).

Лістинг 3.1 – Функція повторної генерації токена доступу

```
protected static string $resource = UserResource::class;

    protected function getHeaderActions(): array
```

```
    {
        return [
            Action::make('regenerate_token')
                ->label('Перегенерувати токен')
                ->icon('heroicon-o-arrow-path')
                ->color('warning')
                ->requiresConfirmation()
                ->action(function (): void {
                    $user = $this->record;
                    $user->tokens()->delete();

                    $token = $user->createToken('api')->plainTextToken;
                    $user->forceFill(['token' => $token])->save();

                    Notification::make()
                        ->success()
                        ->title('Токен перегенеровано')
                        ->body('Новий токен збережено у полі token.')
                        ->send();
                }),
            DeleteAction::make(),
        ];
    }
}
```

кінець лістингу 3.1

Для інформування адміністратора про успішне завершення операції використовується механізм системних повідомлень Notification::make(). Після успішного створення нового токена виводиться повідомлення із заголовком «Токен перегенеровано» та додатковим текстом про успішне збереження нового значення токена. Такий підхід забезпечує зручний контроль виконання адміністративних операцій та покращує взаємодію користувача із системою.

В адміністративній панелі передбачено функцію додавання нових користувачів через веб-інтерфейс. Для реалізації даної можливості використовується стандартна дія CreateAction. Використання даного механізму дозволяє виконувати створення нових записів без необхідності прямого редагування бази даних або програмного коду.

У класі сторінки керування користувачами визначено метод getHeaderActions(), який формує набір дій, доступних у верхній частині сторінки.

У межах даного методу повертається масив із дією CreateAction::make(), що відповідає за створення нового запису користувача (лістинг 3.2).

Лістинг 3.2 – Функція додавання користувачів

```
{
    protected static string $resource = UserResource::class;

    protected function getHeaderActions(): array
    {
        return [
            CreateAction::make(),
        ];
    }
}
```

кінець лістингу 3.2

Після активації даної дії система автоматично відкриває форму створення користувача, у якій адміністратор може ввести необхідні дані, зокрема ім’я користувача, адресу електронної пошти, токен або інші параметри, передбачені структурою моделі. Після підтвердження введених даних виконується перевірка коректності заповнення полів та запис інформації до бази даних.

Використання вбудованого механізму CreateAction дозволяє значно спростити реалізацію CRUD-операцій у системі, забезпечує стандартизований підхід до створення записів та мінімізує обсяг програмного коду. Крім того, інтеграція з механізмами Filament забезпечує автоматичну генерацію елементів інтерфейсу, обробку валідації даних та взаємодію з моделями бази даних.

Для автоматизації процесу створення користувачів у системі реалізовано механізм генерації API-токена безпосередньо під час створення нового запису. Даний підхід дозволяє забезпечити автоматичне формування унікального токена доступу для кожного користувача.

Основна логіка реалізована у методі handleRecordCreation(), який виконується під час створення нового користувача через адміністративну панель системи. Метод приймає масив $data, що містить дані, введені у формі створення користувача.

Спочатку із масиву даних видаляються поля token та token_preview за допомогою функції unset(). Це дозволяє уникнути збереження службових або тимчасових значень, які не повинні безпосередньо передаватися до бази даних під час створення нового запису.

Після обробки вхідних даних виконується створення нового користувача за допомогою методу create(), який належить до моделі системи. Створений об’єкт користувача зберігається у змінній $record, що дозволяє виконувати подальші операції над новоствореним записом.

Далі реалізовано автоматичне створення API-токена через метод createToken(‘api’). Даний механізм формує унікальний токен доступу, який використовується для взаємодії користувача із серверною частиною системи через API. Значення створеного токена отримується через властивість plainTextToken та записується у змінну $token.

Після генерації токена його значення зберігається у полі token моделі користувача за допомогою методу forceFill() із подальшим записом змін до бази даних методом save() (лістинг 3.3).

Лістинг 3.3 – Функція генерації API-токена

```
{
    protected static string $resource = UserResource::class;

    protected function handleRecordCreation(array $data): Model
    {
        unset($data['token'], $data['token_preview']);

        /** @var Model $record */
        $record = static::getModel()::create($data);

        $token = $record->createToken('api')->plainTextToken;
        $record->forceFill(['token' => $token])->save();
        return $record;
    }
}
```

кінець лістингу 3.3

Використання даного підходу забезпечує автоматичне прив’язування токена до відповідного користувача одразу після створення облікового запису.

Завершальним кроком роботи методу є повернення створеного об’єкта користувача через оператор return. Це дозволяє системі продовжити подальшу обробку створеного запису та забезпечує коректну інтеграцію із механізмами адміністративної панелі.

У методі configure() реалізовано налаштування структури форми редагування користувача в адміністративній панелі системи. Метод приймає об’єкт типу Schema, який використовується для побудови елементів інтерфейсу.

Структура форми організована за допомогою секції Section::make(‘Користувач’), яка об’єднує основні елементи керування даними користувача в окремий логічний блок. Для покращення інформативності інтерфейсу використовується опис секції через метод description(), у якому зазначено призначення блоку та його зв’язок із API-токеном системи.

Поле TextInput::make(‘name’) призначене для введення імені користувача. Для нього встановлено текстову мітку «Ім’я», а також визначено обов’язковість заповнення за допомогою методу required(). Коректність введених даних контролюється через метод maxLength(255), який обмежує максимальну довжину значення та запобігає перевищенню допустимого розміру поля у базі даних.

Окреме поле TextInput::make(‘token’) використовується для відображення поточного API-токена користувача. Дане поле має виключно інформаційний характер і не передбачає можливості редагування значення через інтерфейс системи. З цією метою застосовується метод disabled(), який блокує зміну даних користувачем (лістинг 3.4).

Лістинг 3.4 – Функція редагування користувача

```
{
    public static function configure(Schema $schema): Schema
    {
        return $schema
            ->components([
                Section::make('Користувач')
                    ->description('Основні дані користувача та API токен.')
                    ->schema([
                        TextInput::make('name')
                            ->label('Ім’я')
                            ->required()
                            ->maxLength(255),
```

```
                        TextInput::make('token')
                            ->label('Поточний токен')
                            ->formatStateUsing(fn (?string $state): string
=> filled($state)
                                ? Str::limit($state, 36, '...')
                                : 'Токен буде згенеровано автоматично')
                            ->disabled()
                            ->dehydrated(false),
                    ])->columnSpanFull(),
            ]);
    }
}
```

кінець лістингу 3.4

Форматування значення токена виконується за допомогою методу formatStateUsing(). Якщо токен існує, його значення автоматично скорочується за допомогою методу Str::limit() до 36 символів із додаванням символів «...». Це дозволяє забезпечити компактне відображення довгих рядків у межах веб-інтерфейсу. За відсутності токена у полі відображається повідомлення «Токен буде згенеровано автоматично», яке інформує адміністратора про автоматичне створення токена під час реєстрації користувача.

Для виключення поля токена із процесу збереження форми використовується метод dehydrated(false). Такий підхід запобігає передачі відображуваного значення у запитах до сервера та унеможливлює випадковий перезапис токена у базі даних.

3.3.2 Веб-інтерфейс адміністративної панелі

При вході в застосунок користувач отримує доступ до адміністративної панелі системи автентифікації (рис. 3.10), через яку здійснюється взаємодія з основними функціями вебзастосунку. Інтерфейс побудований із використанням бібліотеки Filament, що дозволяє організувати зручну структуру керування користувачами та службовими даними системи.

У лівій частині сторінки розміщене навігаційне меню, за допомогою якого реалізовано перехід між окремими розділами застосунку. Передбачено доступ до інформаційної панелі, журналу дій користувачів та списку облікових записів. Окремо виділено модуль адміністраторів, що використовується для керування користувачами з розширеними правами доступу.

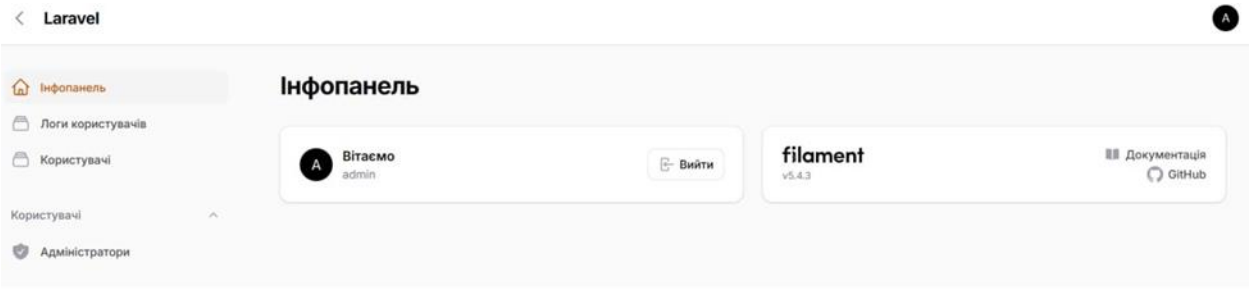

Рисунок 3.10 – Адміністративна панель системи автентифікації

Після успішної авторизації у центральній частині сторінки відображається інформація про поточного користувача та елементи керування сеансом роботи. Для завершення роботи із системою передбачено окрему кнопку виходу, що дозволяє безпечно завершити активну сесію.

Для контролю активності користувачів у системі реалізовано окремий розділ журналу подій, у якому відображається інформація про виконані дії під час роботи із застосунком. Доступ до цього модуля здійснюється через навігаційне меню адміністративної панелі.

У центральній частині сторінки представлено таблицю з переліком зафіксованих подій. Для кожного запису відображається ім’я користувача, тип виконаної дії та час її створення. У даному випадку система фіксує події входу та виходу користувачів, що дозволяє здійснювати базовий контроль активності облікових записів (рис. 3.11).

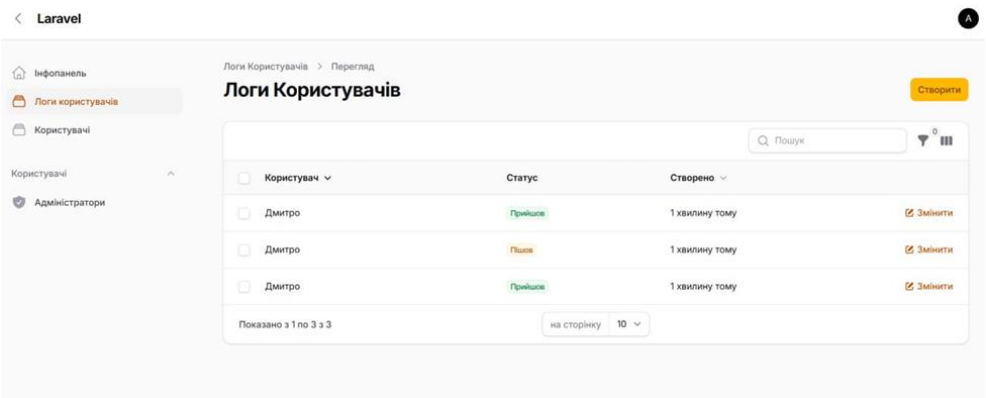

Рисунок 3.11 – Контроль активності облікових записів

Інтерфейс також містить засоби пошуку та фільтрації записів, що спрощує роботу з великою кількістю журналів. Для керування окремими записами передбачено функцію редагування, доступ до якої здійснюється безпосередньо з таблиці.

Використання журналу подій дозволяє підвищити контроль за роботою системи та забезпечує можливість перегляду інформації про дії користувачів у межах вебзастосунку. Також передбачено окремий розділ для керування зареєстрованими користувачами та їх обліковими даними. Доступ до даного модуля здійснюється через адміністративну панель, де відображається перелік користувачів із відповідною службовою інформацією.

У таблиці представлено ім’я користувача, унікальний токен та дату останнього оновлення запису. Використання токенів дозволяє реалізувати механізми додаткової автентифікації та забезпечити взаємодію користувача із системою через захищені запити (рис. 3.12).

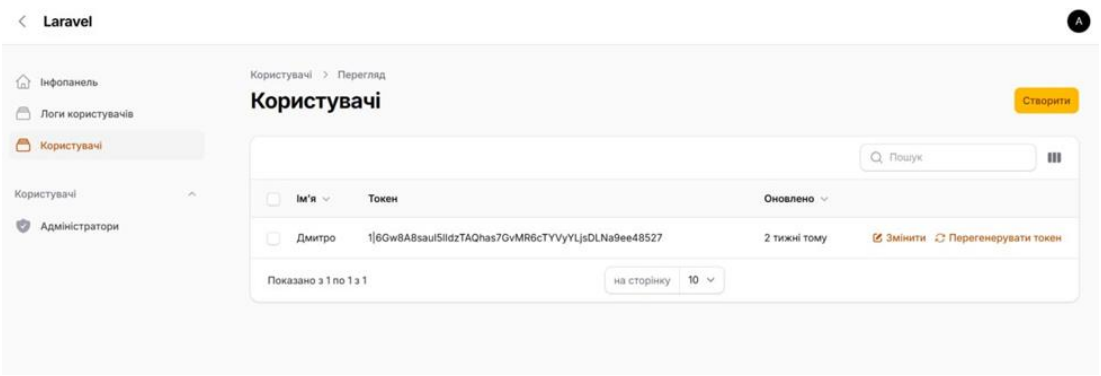

Рисунок 3.12 – Інформація про користувача

Для кожного облікового запису передбачено можливість редагування даних, а також функцію повторної генерації токена. Це дозволяє оновлювати ключі доступу у випадку необхідності та підвищує рівень безпеки системи.

Інтерфейс модуля також містить засоби пошуку та навігації по записах, що спрощує роботу з користувачами та забезпечує швидкий доступ до необхідної інформації.

Для додавання нових облікових записів у системі реалізовано окрему форму створення користувача. Доступ до неї здійснюється через адміністративну панель після вибору відповідної дії у модулі керування користувачами.

У формі передбачено введення основних даних користувача, зокрема імені, яке використовується для подальшої ідентифікації в системі. Окрім цього, інтерфейс містить поле для токена доступу, генерація якого виконується автоматично під час створення запису (рис. 3.13).

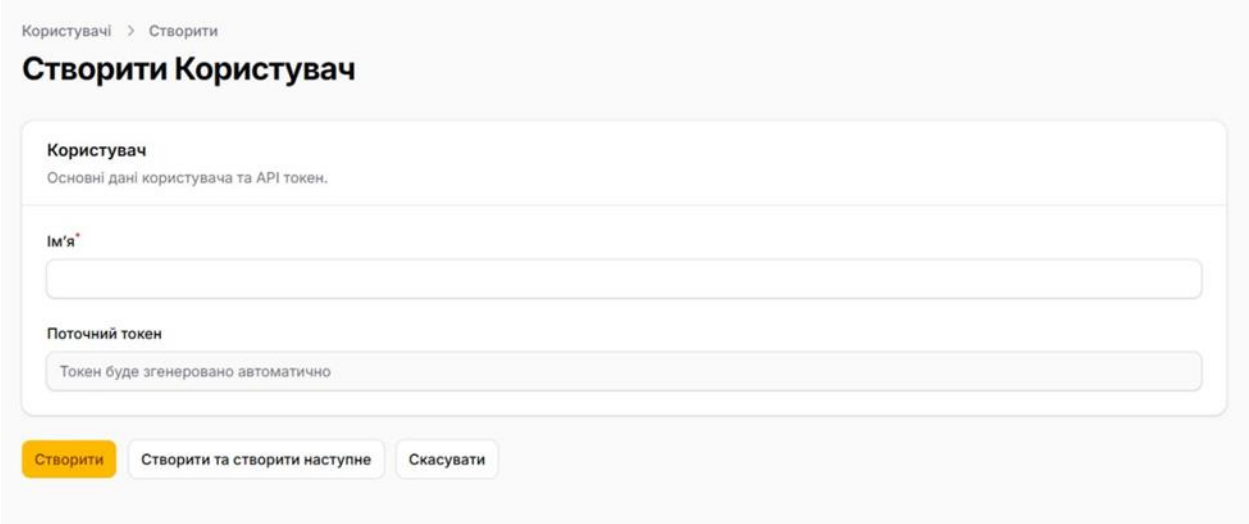

Рисунок 3.13 – Створення користувача

У нижній частині сторінки розміщено елементи керування процесом створення користувача. Передбачено можливість збереження поточного запису, створення нового користувача без повернення до списку, а також скасування виконуваної операції.

Для керування адміністративними обліковими записами у системі передбачено окремий модуль адміністраторів. У даному розділі відображається список користувачів із розширеними правами доступу, їх електронна пошта та статус підтвердження облікового запису (рис. 3.14).

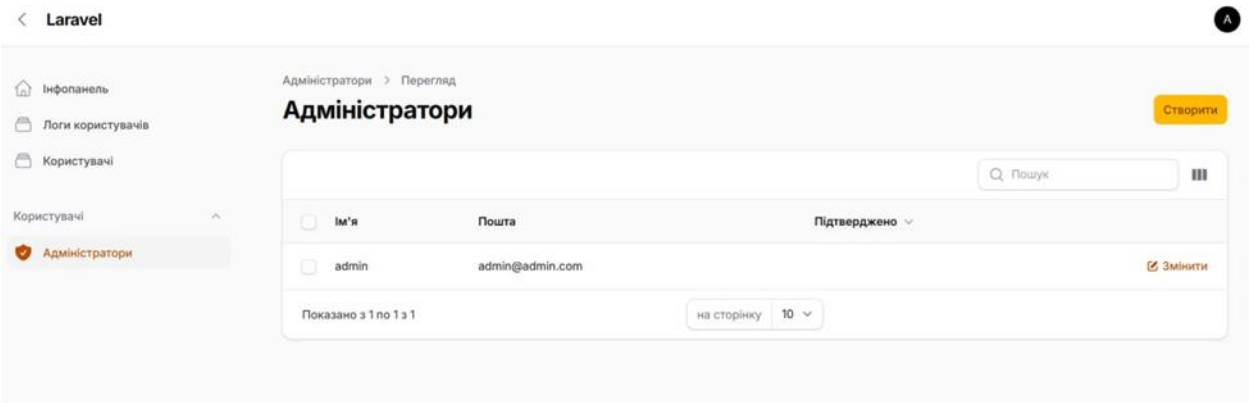

Рисунок 3.14 – Модуль адміністратора

ВИСНОВКИ

У кваліфікаційній роботі розроблено систему аутентифікації користувачів з використанням веб-інтерфейсу на базі Raspberry Pi 3 Model B та карткової технології ідентифікації. Реалізоване рішення поєднує апаратні та програмні компоненти в єдину інтегровану систему, що забезпечує автоматизовану ідентифікацію користувачів, централізоване керування доступом та ведення журналу подій.

У процесі виконання роботи було реалізовано систему зчитування даних із NFC-карток користувачів та їх подальшу обробку на базі Raspberry Pi. Для цього використано NFC-зчитувач ACS ACR1252U III USB, який забезпечує зчитування

унікального ідентифікатора картки та передачу отриманих даних до серверної частини системи через API. Реалізований механізм дозволяє виконувати швидку та автоматизовану ідентифікацію користувачів у режимі реального часу.

У межах роботи було розроблено веб-інтерфейс для керування користувачами та перегляду результатів аутентифікації. Серверну частину системи реалізовано за допомогою Laravel із використанням мови програмування PHP та веб-сервера Apache HTTP Server. Для створення адміністративної панелі використано Filament, що дозволило реалізувати функції додавання користувачів, автоматичної генерації API-токенів, перегляду журналів подій та керування даними системи через веб-інтерфейс.

Під час виконання роботи було досліджено принципи функціонування RFID/NFC-технологій та особливості їх інтеграції з веб-системами. Проведений аналіз дозволив визначити переваги безконтактної ідентифікації, зокрема високу швидкість обробки даних, зручність використання та можливість інтеграції із сучасними інформаційними системами. Також було розглянуто принципи взаємодії NFC-пристроїв із серверною частиною через REST API.

У роботі виконано візуалізацію структури та архітектури системи аутентифікації користувачів. Було сформовано структурні схеми взаємодії апаратних і програмних компонентів, що дозволило відобразити процес обробки даних від моменту зчитування NFC-картки до перевірки користувача у базі даних та відображення результату у веб-інтерфейсі. Побудована архітектура забезпечує модульність системи та можливість її подальшого масштабування.

Для забезпечення зберігання та обробки інформації про користувачів і картки доступу було спроєктовано базу даних на основі MySQL. Структура бази даних забезпечує збереження даних користувачів, API-токенів та журналу аутентифікації, а також підтримує ефективну взаємодію між серверною логікою та веб-інтерфейсом системи.

У межах роботи запропоновано методи підвищення безпеки та надійності функціонування системи. Реалізовано механізм API-аутентифікації з використанням токенів доступу, автоматичне генерування токенів для користувачів, розмежування доступу до адміністративної панелі та ведення

журналу подій. Додатково використано механізми валідації даних та підтвердження адміністративних дій, що дозволяє знизити ризик несанкціонованого доступу та підвищити рівень захисту інформації.

Отримані результати підтверджують доцільність використання одноплатних комп’ютерів Raspberry Pi та NFC-технологій для побудови сучасних систем контролю доступу. Розроблена система характеризується гнучкістю, масштабованістю та можливістю практичного застосування у навчальних закладах, офісних приміщеннях, лабораторіях та інших об’єктах, де необхідно забезпечити ефективний контроль доступу користувачів.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ідентифікація та аутентифікація користувачів. URL: https://sites.google.com/view/dcptoinformat/ (дата звернення: 23.02.2026).

2. Що таке автентифікація? URL: https://www.microsoft.com/uk-ua/security/business/security-101/what-is-authentication (дата звернення: 23.02.2026).

3. Аутентифікація користувачів. URL: https://e-tk.lntu.edu.ua/mod/resource/view.php?id=3261 (дата звернення: 23.02.2026).

4. Парольна аутентифікація. URL: https://training.qatestlab.com/wp-content/uploads/2023/06/2.png (дата звернення: 23.02.2026).

5. Навіщо і як використовувати двофакторну аутентифікацію?. URL: https://root-nation.com/ua/articles-ua/internet-ua/ua-2f-authentification-howto/ (дата звернення: 23.02.2026).

6. Види біометричної аутентифікації. URL: https://nixj.ua/vidi-bometrichno-autentifkac (дата звернення: 23.02.2026).

7. Апаратна аутентифікація. URL: https://upload.wikimedia.org/wikipedia/commons/7/7b/EToken_6_models.jpg (дата звернення: 23.02.2026).

8. Single sign-on. URL: https://uk.wikipedia.org/wiki/Single_sign-on (date of access: 12.03.2026).

9. Технологія Single Sign On. URL: https://sb.nordcdn.com/transform/ed33728d-2a9b-4359-a3e5-c836d6b9fcf8/2-inner-asset-sso-76x340?format=webp&quality=80&io=transform:fill,width:1536 (дата звернення: 12.03.2026).

10. Одноплатний комп’ютер Raspberry Pi. URL: https://lh3.googleusercontent.com/proxy/ScIvUZHBvgk-WxTFGIGJQI7n2SsloaTgtVFFfvRJygTrmIsc6hQ5hg07uEcz59PascEzpXo3jcym54i-Qj2R0I8o5wEv7P4WAO6RjzxskRsPhKHzCjNKhg (дата звернення: 12.03.2026).

11. Що таке мікрокомп’ютери?. URL: https://e-server.com.ua/uk/poradi/shho-take-mikrokompiuteri-rozpovidajemo-na-prikladi-raspberry-

pi?srsltid=AfmBOor_hCHrmAiF2-FfAWU2QAYXeo9Cpl9pL9PxzEm1wz0dHuuT6Zjo (дата звернення: 22.03.2026).

12. Raspberry Pi. URL: https://uk.wikipedia.org/wiki/Raspberry_Pi (date access: 22.03.2026).

13. Raspberry Pi 3 Model B. URL: https://miniboard.com.ua/3033/272.jpg (date of access: 22.03.2026).

14. Мікрокомп’ютер Raspberry Pi 3 Model B. URL: https://evo.net.ua/raspberry-pi-3-model-b/770/ (дата звернення: 22.03.2026).

15. NFC-зчитувач ACS. URL: https://idcard.com.ua/media/catalog/product/cache/74c1057f7991b4edb2bc7bdaa94de933/n/f/nfc_acr1252_iii_usb.jpg (дата звернення: 22.03.2026).

16. USB NFC Reader 3. URL: https://www.acs.com.hk/en/products/342/acr1252u-usb-nfc-reader-iii-nfc-forum-certified-reader/?utm_source=chatgpt.com (date of access: 22.03.2026).

17. NFC-карта RFID. URL: https://www.rfidfuture.com/wp-content/uploads/2021/03/13.56Mhz-rfid-cards.jpg (дата звернення: 22.03.2026).

18. Бесконтактна картка Mifare Classic 1K ISO 13.56 МГц. URL: https://zkstore.com.ua/ua/p1012378992-beskontaktnaya-karta-mifare.html?srsltid=AfmBOorWRITs2TufcAvaEecEg9oYNgYNqaaLkArOrMw_eiuu0N1FPuTj (дата звернення: 22.03.2026).

19. Приклад веб-серверу Raspberry Pi. URL: https://cdn.shopify.com/s/files/1/0474/7729/3217/files/5_bc110acc-1da5-4e8a-ad21-444fcc37387b.png?v=1721617769 (дата звернення: 22.03.2026).

20. Що таке PHP та що з цим можна робити?. URL: https://www.php.net/manual/uk/introduction.php (дата звернення: 22.03.2026).

21. Модель-вид-контролер. URL: https://uk.wikipedia.org/wiki/Модель-вид-контролер (дата звернення: 22.03.2026).

22. Що таке Apache і як він впливає на розробку веб-сайтів?. URL: https://alexhost.com/uk/faq/what-is-apache-and-what-does-it-do-for-website-development/ (дата звернення: 22.03.2026).

23. Laravel Filament: революціонізація розробки адміністративних панелей у Laravel. URL: https://ua.linkedin.com/pulse/laravel-filament-revolutionizing-admin-panel-neelesh-chakraborty-cuhxf?tl=uk (date of access: 22.03.2026).

24. Raspberry Pi Imager. URL: https://www.raspberrypi.com/software/ (date of access: 05.04.2026).