НІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач випускової кафедри
_____ Аліна САВЧЕНКО
«___»_____2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВР

ЗА ОСВІТНЬО-ПРОФЕСІЙНОЮ ПРОГРАМОЮ

«ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ПРОЕКТУВАННЯ»

Тема: «Система аутентифікації на сайті в мобільному браузері на основі технології NFC»

Виконавець: Артем ШВЕЦЬ

Керівник: к.т.н. Олег ЗУДОВ

Нормоконтролер: к.т.н., доцент Олена ТОЛСТІКОВА

КИЇВ 2023

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет *комп'ютерних наук та технологій*
Кафедра *комп'ютерних інформаційних технологій*
Спеціальність *122 «Комп'ютерні науки»*
Освітньо-професійна програма *«Інформаційні технології проектування»*

ЗАТВЕРДЖУЮ:
завідувач кафедри КІТ
_____Аліна САВЧЕНКО
(підпис)
«___»_____2023 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи
Швеця Артема Сергійовича
(ПІБ випускника)

1. Тема роботи: «Система аутентифікації на сайті в мобільному браузері на основі технології NFC» затверджена наказом ректора № 623/ст від 01.05.2023р.

2. Термін виконання роботи: з 15 травня 2023 року по 25 червня 2023 року.

3. Вихідні дані до роботи: Сайт з можливістю автентифікації за допомогою NFC зроблений за допомогою JavaScript, PHP, HTML.

4. Зміст пояснювальної записки: 1. Огляд та аналіз предметної області 2. Програмне забезпечення для програмування NFC 3. Створення сайту.

5. Перелік обов'язкового ілюстративного матеріалу: 1.Основні принципи технології NFC 2. Історія створення та використання NFC 3. Аналоги проекту 4.Вибір програмного забезпечення для програмування NFC на сайті 5. Особливості програмування nfc за допомогою javascript 6. Опис файлової структури та принципу роботи 7. Створення головної Html сторінкі з аунтифікацією на сайті за допомогою NFC 8. Огляд роботи системи аунтифікації.

6. Календарний план-графік

№ з/п	Завдання	Термін виконання	Підпис керівника
1	Огляд та аналіз предметної області. Написання 1 розділу, представлення керівнику.	15.05.2023-20.05.2023	
2	Вибір та опис використаних технологій. Написання 2 розділу, представлення керівнику.	21.05.2023-27.05.2023	
3	Написання 3 розділу, представлення керівнику.	28.05.2023-04.06.2023	
4	Загальне редагування та друк пояснювальної записки.	05.06.2023-08.03.2023	
5	Проходження нормоконтролю, перепліт пояснювальної записки.	09.06.2023-15.06.2023	
6	Розробка тексту доповіді. Оформлення графічного матеріалу для презентації	16.06.2023-18.06.2023	

7. Дата видачі завдання 15.05.2023р.

Керівник кваліфікаційної роботи _____ Олег ЗУДОВ
(підпис керівника)

Завдання прийняв до виконання _____ Артем ШВЕЦЬ
(підпис випускника)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи на тему: «Система аутентифікації на сайті в мобільному браузері на основі технології NFC» містить: 45 сторінок, 24 рисунки, 14 інформаційних джерел.

Об’єкт дослідження – технології аутентифікації.

Предмет дослідження – система аутентифікації на сайті в мобільному браузері на основі технології NFC.

Мета кваліфікаційної роботи – розробити сайт в якому є можливість аунтифікації за допомогою NFC.

Методи дослідження – логічний, синтезу, аналізу, порівняльний, обробка літературних джерел та моделювання.

Результат кваліфікаційної роботи рекомендується використовувати як основу для реалізації аунтифікаціїї через NFC. Для розробки данного сайту було використано мови програмування JavaScript, PHP, HTML та мова стилів CSS які підходять для данного рішення проблеми найефективніше.

КЛЮЧОВІ СЛОВА HTML, JAVASCRIPT, NFC, JAVA, PHP, PYTHON, FLASK, DJANGO, PG ADMIN, SQL, JSON, RFID.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

NFC	−	Near-Field Communication
RFID	−	Radio Frequency IDentification
FLASK	−	Фреймворк для Python
DJANGO	−	Фреймворк для Python
PG ADMIN	−	Застосунок для створення баз данних
PHP	−	Hypertext Preprocessor

ВСТУП

У сучасному світі інтернет відіграє важливу роль у житті людей. За допомогою мережі можна отримати доступ до різноманітної інформації, здійснювати онлайн-покупки та багато іншого. Проте, разом з популярністю інтернету зростає і кількість людей що бажають дістати особисту інформацію з різніх сайтів. У зв’язку з цим стає все більш актуальною проблема безпеки входу на сайти. Щоб забезпечити безпеку входу на сайт використовують різні методи аунтифікації. . Тому існує потреба в розробці інструментів, які б дозволяли захистити свої данні від втрати їх в інтернеті.

У даній роботі розглядається інструмент аунтифікації на сайті задопомогою NFC чипу.

Метою дослідження є розробка програми, яка би дозволяла підвищувати як безпечність аунтифікації на сайті так і зручність. Однією з найбільших проблем при вході на сайти є збір та використання користувальницьких даних зловмисниками. Ця проблема стає ще більш актуальною, коли користувачі використовують слабкі паролі, які можуть бути легко підібраними. Нажаль більшісті Користувачам часто потрібно запам’ятовувати довгі паролі для кожного сайту, на який вони зайшли. Це може бути незручним та важким завданням, якщо потрібно запам’ятати більше одного пароля, особливо для доволі старших людей. Також у деякіх користувачив можуть виникнути проблеми з використанням двофакторної аутентифікації та введення коду з SMS повідомлення чи введення цього коду через спеціальний додаток.

Об’єктом дослідження кваліфікаційної робити є аунтифікація на сайті за допомогою NFC.

Наукова новизна авторизації в мобільному браузері за допомогою NFC заключається в технології яка пропонує зміну традиційного підходу до авторизації в мобільних браузерах, шляхом використання технології безконтактного зв'язку NFC.

NFC - це короткодіюча радіочастотна технологія, яка дозволяє обмінюватися даними між двома пристроями, які знаходяться близько один до одного. Ця технологія зазвичай використовується для безконтактної оплати, передачі інформації, обміну контактами та інших додатків. В данному випадку якщо викростовувати її авторизації в мобільному браузері то це викристання NFC надає впевненість в безпеці як додатковий крок захисту інформаціїї так і зручність її використання. Замість введення пароля або використання відбитку пальця, користувач може просто наблизити свій сумісний пристрій до телефону щоб моментально здійснити вхід до сайту з потрібною інформацією.

Наукова новизна полягає в тому, що дана технологія розширює можливості авторизації, забезпечуючи безпеку та швидкість процесу. Вона дозволяє запам'ятовувати та автоматично вводити дані для авторизації, що економить час користувача. Крім того, вона є безпечнішою, оскільки дані передаються безпосередньо між пристроями, що мінімізує ризик підбору паролів або використання недостовірних аутентифікаційних методів.

Ця технологія може знайти застосування в різних сферах, як в фінансових послугах, електронній комерції, соціальних мережах так і в звичайному повсяк денному житті а також багато інших сферах. Вона відкриває нові перспективи для забезпечення безпеки, швидкості та зручності в мобільних браузерах, забезпечуючи велику кількість можливостей для подальшого розвитку цифрових технологій та покращення користувацького досвіду.

РОЗДІЛ 1
ОГЛЯД ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Основні принципи технології NFC

Технологія NFC (Near Field Communication) є бездротовим зв'язком на короткій відстані, що дозволяє обмінюватися даними між двома пристроями, що знаходяться в непосредственній близькості один від одного (зазвичай відстань не більше 4 см). Вона базується на радіочастотній ідентифікації (RFID) та забезпечує швидку та безпечну передачу інформації.

NFC використовує частоту 13,56 МГц і використовує два основні режими роботи: активний та пасивний. У режимі активного пристрою, наприклад смартфона, він може відправляти та приймати дані. У режимі пасивного пристрою, наприклад безконтактної картки, він може тільки передавати дані, але не може їх приймати.

Основна ідея NFC полягає в тому, що два пристрої з NFC можуть встановлювати безпровідний зв'язок один з одним шляхом наближення або торкання їхніх антен. Це створює миттєвий зв'язок, який дозволяє обмінюватись даними без необхідності додаткових дій користувача, таких як налаштування підключення або пошук пристроїв. основні переваги NFC завдяки яким він почав набирати популярність в багатьох сферах нашого життя:

Простота використання NFC не потребує складних налаштувань або підключень. Достатньо просто наблизити два пристрої один до одного, і вони зможуть обмінюватись даними. Завдяки цьому пошвидшилась перевірка білетів, документів та багатьох інших документних операцій. Завдяки швидкості передачи данних NFC може передавати дані дуже

швидко, зазвичай в межах кількох мегабайт на секунду. Це дозволяє швидко обмінюватись інформацією, такою як контакти, фотографії або файли. Технологія NFC використовує протоколи шифрування для забезпечення безпеки передачі даних. Це робить його безпечним для використання в різних сферах, включаючи фінансові транзакції та автентифікацію. Тому використання безконтактних міток NFC дозволяє безпечно виконувати ниски певних задач з цінними документами без ризику втрати їх.

Одна з основних переваг NFC це сумісність з багатьма типами пристроїв, включаючи смартфони, планшети, ноутбуки та спеціальні пристрої. Це дозволяє йому застосовуватись в різних сферах, таких як платежі, доступ до приміщень, контроль запасів та в багатьох інших сферах.

Також для З'єднання пристроїв за допомогою NFC можна використовувати пристрої двух типів пасивний та активний.

У активному режимі два пристрої виконують ролі "активного" пристрою, тобто обидва можуть ініціювати та передавати дані. Передача даних відбувається, коли обидва пристрої знаходяться на відстані один від одного, не більше кількох сантиметрів. Зазвичай активний режим використовується для взаємодії між двома сумісними пристроями, такими як обмін повідомленнями, передача файлів, підключення до бездротових навушників.

У пасивному режимі один пристрій має роль "пасивного" пристрою, який передає дані. Наприклад NFC- мітка.

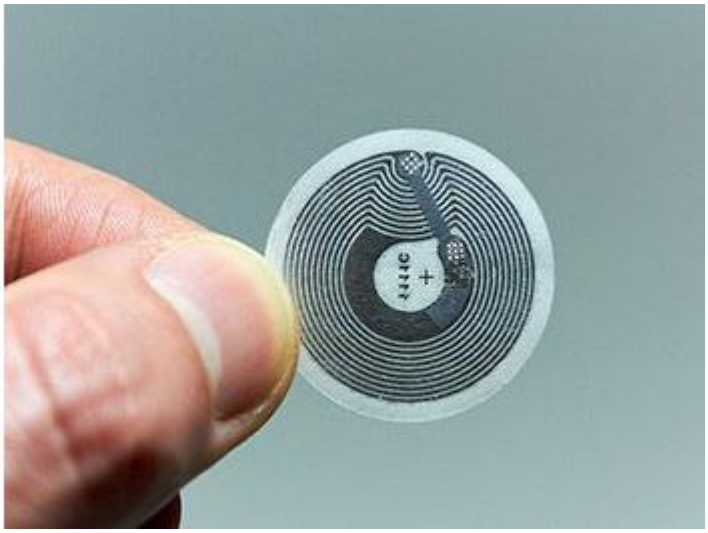

Рис. 1.1. Приклад NFC мітки

Для з’єднання пристроїв з міткою потрібен активний пристрій (смартфон аобо считувач). У цьому режимі передача даних здійснюється, коли активний пристрій наближається до пасивного пристрою на відстань не більше кількох сантиметрів. Пасивний пристрій починає передавати дані, які можуть бути зчитані активним пристроєм. Наприклад, це може бути ідентифікатор, URL-адреса, текстова інформація тощо. Пасивний режим з'єднання застосовується в багатьох сферах, таких як безконтактні платежі, доступ до будівель або транспорту, використання міток для ідентифікації товарів і т.д. [1-4]

1.2. Історія створення та використання NFC

Історія створення та використання технології безконтактного зв'язку NFC (Near Field Communication) налічує багато років і проходить через кілька етапів розвитку, покращень та впроваджень. Існує 5 основниї єтапів створення NFC.

Етап 1: Відкриття технології RFID Технологія безконтактного зв'язку NFC виникла на основі технології RFID, яка дозволяє ідентифікувати об'єкти за допомогою радіочастотного зв'язку. Наприклад, багажні квитки в аеропорту можуть містити RFID-мітки, які дозволяють автоматично розпізнавати та відстежувати багаж під час його перевезення. Компанії в

галузі логістики та транспорту використовують технологію RFID для ефективного відстеження та управління запасами. Наприклад, великі торгові мережі можуть використовувати RFID-мітки для швидкого сканування та інвентаризації товарів на полицях магазинів.

Етап 2: Створення NFC. Технологія NFC була розроблена в результаті спільних зусиль компаній, таких як Philips, Sony та Nokia. NFC використовує частоту 13,56 МГц, що забезпечує сумісність з вже існуючими мітками RFID. За допомогою технології NFC можна здійснювати безконтактні платежі. Наприклад, сучасні мобільні платіжні системи, такі як Apple Pay або Google Pay, використовують NFC для проведення операцій. Користувач просто наближає свій сумісний смартфон до безконтактного терміналу платежів, що дозволяє зручно та безпечно здійснювати покупки.

Етап 3: Перші використання NFC У Японії у 2004 році було запущено систему безконтактних платежів FeliCa, яка використовувала NFC. Вона широко застосовувалася для оплати проїзду в громадському транспорті, покупки в автоматах та інших сервісів. Великі міста, наприклад, Токіо, використовують безконтактні картки з NFC для оплати проїзду в метро. Пасажирі просто прикладають свою картку до читача біля турнікету, що дозволяє швидко та зручно пройти через контроль.

Етап 4: Розширення застосувань NFC. З плином часу NFC стала використовуватися в багатьох сферах. Вона може використовуватися для передачі файлів, обміну контактами, автоматичного налаштування пристроїв Bluetooth та Wi-Fi, аутентифікації та інших завдань. Компанії можуть використовувати технологію NFC для безпечного та зручного доступу до приміщень. Наприклад, працівники можуть мати картки з NFC, які дозволяють їм просто наблизити картку до зчитувача біля дверей для автоматичного розблокування.

Етап 5: Впровадження NFC в мобільних пристроях. З появою смартфонів з підтримкою NFC, технологія отримала ще більшу популярність. Виробники, такі як Apple, Samsung, Huawei, включають NFC в свої пристрої,

що дозволяє користувачам використовувати безконтактні платежі, перевірку балансу, прохід на події, використання віртуальних квитків та інших функцій. За допомогою NFC користувачі можуть швидко і легко передавати файли, фотографії або контакти. Достатньо просто наблизити два сумісних смартфона один до одного, і вони автоматично встановлять з'єднання та передадуть вказану інформацію.

Рис. 1.2. Приклад застосування NFC для безконтакнтних платежів

Це лише кілька різноманітних застосувань технології NFC. Завдяки безпечному та зручному безконтактному зв'язку, NFC стає все більш популярною технологією у різних сферах, спрощуючи наші щоденні завдання та полегшуючи обмін інформацією. [14]

1.3. Аналоги проекту

Дуже важливо дослідити та порівняти аналоги цього підходу зі звичайною авторизацією на сайтах. Щоб зрозуміти різницю нового підходу за допомогою NFC з звичайною аунтифікацією.

Звичайна авторизація на сайтах включає введення ідентифікаційних даних (логіну та пароля) для доступу до облікового запису користувача. З всього можливо виділити такі плюси в цьому підході:

– Широко поширений метод авторизації, який підтримується майже всіма веб-додатками та браузерами.

– Зручність для користувачів, оскільки вони можуть використовувати одні й ті ж ідентифікаційні дані для доступу до різних сайтів.

– Забезпечує високий рівень безпеки, якщо користувач використовує міцний пароль та дотримується належних заходів безпеки.

Але все ж таки існують декілька мінусів повязаних з безпекою інформаціїї користувача.

Великий ризик підбору пароля або витоку ідентифікаційних даних, особливо якщо користувач використовує слабкий пароль або виконує авторизацію на ненадійних пристроях або мережах. Існує також потреба в регулярній зміні паролів та підтриманні їх унікальності для різних облікових записів. Та основна проблема для більшості користувачів це вимога запам'ятовувати та вводити ідентифікаційні дані на кожному вході на сайті.

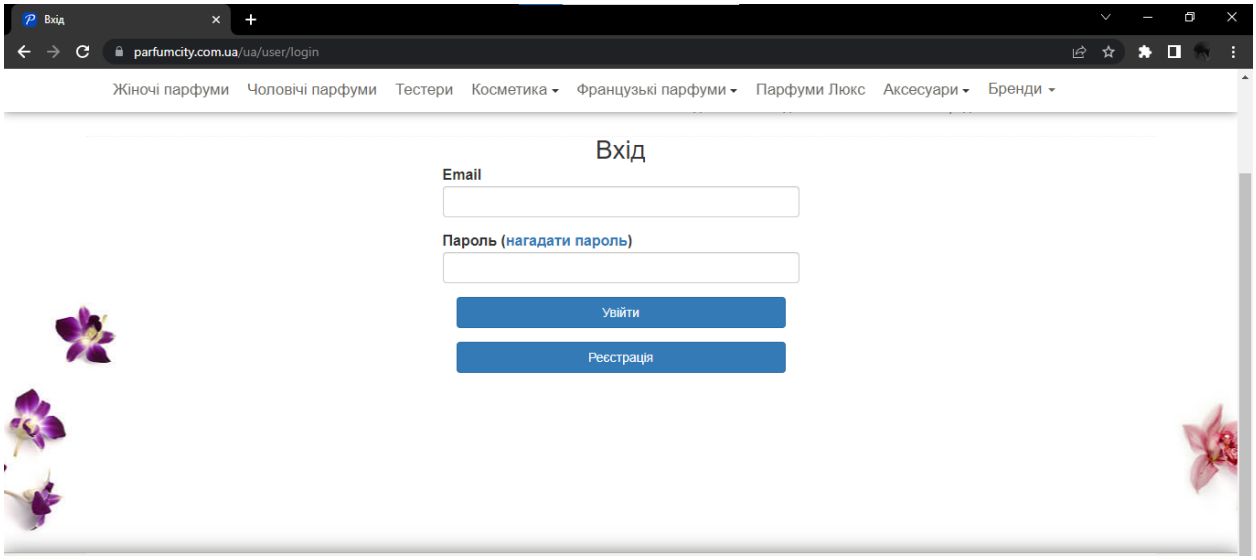

Рис. 1.3. Приклад звичайної авторизації на сайті

Розглянемо авторизацію за допомогою NFC у мобільному браузері. Вона використовує безконтактну технологію NFC для обміну ідентифікаційними даними між мобільним пристроєм та веб-додатком. Плюсами цього підходу є:

− Зручність та швидкість авторизації, оскільки користувачу потрібно лише прикласти мобільний пристрій до зчитувача NFC для передачі ідентифікаційних даних.

− Високий рівень безпеки, оскільки ідентифікаційні дані зберігаються на безпечному елементі (наприклад, Secure Element) у мобільному пристрої.

− Зменшення ризику підбору пароля або витоку ідентифікаційних даних, оскільки користувачу не потрібно вводити ідентифікаційні дані на кожному вході.

Але зважаючи на не дуже велику популярність серед мобільних сайтів може виникнути проблема з обмеженою підтримкою NFC на деяких старих або дешевих мобільних пристроях. Потреба в наявності зчитувача NFC у веб-додатку або серверній інфраструктурі для обробки ідентифікаційних даних. Та вимога до користувача мати мобільний пристрій з підтримкою NFC та використовувати його для авторизації.

Зважаючи на всі оглянуті плюси на та мінуси звичайної авторизації та авторизації за допомогою NFC, можна сказати що порівнянно з звичайним методов аунтифікація за допомогою NFC більш швидша, безпечніша, зручніша. Хоч і має декілька мінусів стосовно відносно не великої популярності в мобільних сайтах та вимог стосовно пристроя користувача.

ВИСНОВКИ ДО РОЗДІЛУ 1

Технологія безконтактного зв'язку NFC пройшла довгий шлях розвитку, починаючи з використання технології RFID і до її широкого впровадження в різних сферах.

Висновуючи з дослідження технології NFC, можна сказати, що ця бездротова зв'язкова технологія, на радіочастотній ідентифікації, виявляється надзвичайно корисною та популярною у різних сферах нашого життя.

NFC дозволяє швидку та безпечну передачу інформації між двома пристроями, що знаходяться в непосредственній близькості один від одного. Ця технологія не потребує складних налаштувань або підключень, що робить її простою у використанні. Завдяки швидкості передачі данних, NFC забезпечує швидкий обмін інформацією, а протоколи шифрування гарантують безпеку передачі даних.

Історія розвитку технології NFC свідчить про поступове удосконалення та розширення її застосувань. Вона виникла на основі технології RFID і пройшла крізь кілька етапів, починаючи з відкриття технології RFID і закінчуючи впровадженням NFC в мобільних пристроях. Сьогодні NFC використовується в безконтактних платежах, доступі до приміщень, обміні даними між пристроями та багатьох інших сферах.

Загалом, технологія NFC стає необхідним інструментом у нашому сучасному світі, полегшуючи багато щоденних процесів і сприяючи зручності та безпеці в обміні інформацією. З його допомогою ми можемо швидко та легко здійснювати покупки, передавати файли, обмінюватися контактами та виконувати інші завдання, що робить наше життя більш ефективним та зручним.

РОЗДІЛ 2

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ПРОГРАМУВАННЯ NFC

2.1. Вибір програмного забезпечення для програмування NFC на сайті

Для створення веб-сайту з функцією NFC можна використовувати різні мови програмування та технології. Кожна з них має свої переваги, ось декілька з них які можливо використати в програмуванні сайту з функцією NFC

− **JavaScript**: JavaScript є основною мовою програмування для розробки веб-сайтів. JavaScript можна використовувати для взаємодії з NFC-пристроями, використовуючи стандартний Web NFC API або сторонні бібліотеки, такі як NFC.js.

− **HTML5**: HTML5 має підтримку для роботи з NFC. Можливо використовувати спеціальні атрибути та API HTML5 для зчитування та запису даних на NFC-тегах.

− **PHP**: PHP можливо використати для обробки запитів зі сторони сервера та обробки всіх данних з NFC-тегів та взаємодію з базою даних.

− **Python**: Python також може використовуватись для створення веб-сайтів з функцією NFC. За допомогою Python-бібліотеки, такі як nfcpy, використовуються для зчитування та запису даних на NFC-тегах.

− **Java**: Для роботи з Java-технологій, треба використовувати бібліотеки, такі як javax.smartcardio, для роботи з NFC-пристроями.

HTML5 є універсальним стандартним пристроєм який підтримується більшістю сучасних мобільних браузерів. Це означає, що веб-додаток,

розроблений з використанням HTML5, може бути запущений на багатьох різних пристроях без необхідності в спеціалізованих додатках.

Завдяки простоті розробки та простому і зрозумілу синтаксис та велику кількість ресурсів та документації, що полегшує розробку аутентифікації за допомогою NFC. Він також дозволяє використовувати існуючі веб-технології, такі як JavaScript, для створення динамічних інтерфейсів та обробки даних. А також стандартізація та підтримка багатьох пристроїв, розгортання аутентифікації за допомогою NFC на мобільних пристроях може бути швидким та ефективним процесом при використанні HTML5.

Але HTML5 API для NFC має обмежену функціональність порівняно зі спеціалізованими мобільними платформами або нативними додатками, це може обмежити можливості реалізації аутентифікації.

Розглянемо що таке PHP. PHP являє собою мову програмування загального призначення, розробленою спеціально для веб-розробки. Вона широко використовується для створення динамічних веб-сайтів і взаємодії з базами даних. PHP є однією з найпопулярніших мов програмування для веб-розробки, що дозволяє розробникам ефективно створювати веб-застосунки різного роду, від невеликих сайтів до складних веб-порталів.

Основні особливості Php

− Синтаксис: PHP має синтаксис, подібний до синтаксису C, що робить його досить простим для вивчення і розуміння, особливо для тих, хто вже знайомий з C, C++, Java або іншими подібними мовами.

− Вбудована підтримка веб-розробки: PHP надає вбудовану підтримку для веб-розробки, що дозволяє легко включати PHP-код безпосередньо в HTML-сторінки. Це робить можливим динамічну генерацію вмісту сторінок на основі даних з бази даних, обробку форм, керування сесіями користувачів та багато іншого.

− Підтримка різних баз даних: PHP має вбудовану підтримку для багатьох популярних систем управління базами даних (СУБД), таких як MySQL, PostgreSQL, SQLite, Oracle і багатьох інших. Це дає розробникам

можливість зручно працювати з даними в базі даних, виконувати запити, додавати, редагувати та видаляти дані.

− Розширюваність: PHP має велику кількість розширень (extensions) і бібліотек, які дозволяють розробникам розширювати можливості мови і використовувати готові рішення для різних завдань. Наприклад, існують розширення для роботи з графікою, генерації PDF-файлів, роботи з API соціальних мереж та багато іншого.

− Підтримка ООП: PHP підтримує об'єктно-орієнтоване програмування (ООП), що дозволяє розробникам створювати класи, об'єкти, спадкування, поліморфізм та інші концепції ООП. Це спрощує розробку складних програм та поліпшує їх структуру і повторне використання коду.

− Платформонезалежність: PHP підтримує багато операційних систем, таких як Windows, Linux, macOS і інші, що дозволяє розгортати PHP-застосунки на різних серверах. Крім того, він сумісний з багатьма веб-серверами, такими як Apache, Nginx і Microsoft IIS.

− Активна спільнота: PHP має велику активну спільноту розробників, що створює велику кількість ресурсів, документації, форумів, блогів та інших джерел, де можна знайти допомогу, поради та готові рішення для різних задач.

Загалом, PHP є потужним і гнучким інструментом для розробки веб-додатків. Він надає широкі можливості для взаємодії з веб-серверами та базами даних, розробки функціональних інтерфейсів, обробки форм, керування користувачами та багато іншого.

Використання PHP для створення серверної частини аунтифікації має особливості а саме:

PHP-розробка серверної частини Для обробки та перевірки даних, що передаються через NFC, потрібно реалізувати серверну частину за допомогою PHP. PHP дозволяє створити сервер, який може отримувати дані від мобільного пристрою, валідувати їх та зберігати аутентифікаційні дані у базі даних. Для зберігання аутентифікаційних даних, таких як

ідентифікатори NFC, користувальницькі дані тощо, можна використовувати базу даних MySQL або іншу сумісну з PHP. PHP забезпечує можливість зчитування та запису даних у базі даних, що дозволяє зберігати та перевіряти інформацію для аутентифікації. Основна увага у створенні бази данних повинна бути приділена безпеці в процесі розробки аутентифікації з використанням NFC. Потрібно ретельно перевіряти та валідувати дані, що надходять з мобільного пристрою, і забезпечити захист від можливих атак, таких як перехоплення або модифікація даних. Загалом, аутентифікація на сайті у мобільному браузері за допомогою NFC з використанням PHP може бути зручним та безпечним методом, але потребує уваги до безпеки та реалізації серверної логіки а також для обробки даних аутентифікації та перевірки їх коректності потрібна реалізація серверної логіки з використанням PHP, що може вимагати додаткової роботи і ресурсів.

Python можна застосувати для серверної розробки бо ця мова програмуваня має дуже велику кількість бібліотек та frameworks для веб-розробки а саме для створення серверної части можна викростати фреймворки такі як Flask або Django

Flask є мінімалістичним фреймворком, який надає основні функціональні можливості для розробки веб-додатків. Він має просту структуру та інтуїтивний синтаксис, що робить його дуже легким у використанні та вивченні. Він дає велику свободу у виборі і налаштуванні компонентів додатку. В ньому можливо вибрати бібліотеки, шаблонізатори, бази даних та інші компоненти самостійно.

Flask добре підходить для невеликих та середніх проектів. Він дозволяє швидко розпочати розробку та має гнучкі можливості розширення для забезпечення масштабованості проекту при потребі.

Django пропонує повний стек інструментів та функціональність, що допомагають швидко створювати повнофункціональні веб-додатки. Він має вбудовану адміністративну панель, механізм маршрутизації, ORM та багато іншого. Також має вбудовану підтримку для багатьох загальноприйнятих

функцій, таких як аутентифікація користувачів, керування сеансами, робота з базами даних, кешування, адміністративний інтерфейс тощо. Це значно спрощує розробку проекту і зменшує кількість коду, який потрібно написати.

Django підходить для проектів будь-якого розміру, від невеликих до складних та великих. Він надає інструменти для розробки високонавантажених систем та масштабованості, такі як горизонтальне масштабування, кешування, розподілена обробка завдань та інші.

Крім серверної частини Python надає бібліотеки, які дозволяють зчитувати та записувати дані з NFC-чипів. Наприклад, бібліотека nfcpy для взаємодії з NFC-рідерами та читання/запису даних на NFC-теги. API такжож можна зробити використовую чи Python і веб-фреймворків можна створити веб-сервер, який буде прослуховувати запити від мобільного браузера та здійснювати аутентифікацію. До цього можливо додати API для обміну даними з мобільним пристроєм через NFC.

Стосовно безпеки Python надає велику кількість потужних бібліотек для роботи з шифруванням, хешуванням паролів та іншими методами безпеки. Завдяки яким можливо використати ці бібліотеки для забезпечення безпеки аутентифікації та зберігання користувацьких даних.

В цілому Python доволі гнучка мова програмування з високим рівнем абстракції, що дозволяє швидко розробляти та тестувати рішення. З можливістю легко взаємодіяти з NFC-чипами, створювати веб-сервери та реалізовувати безпеку. Його Велика кількість бібліотек для будь-яких потреб можна використати для роботи з NFC, веб-розробку та безпеку. Основним безперечним плюсом Python являє собою мультиплатформенність що означає, що код можна запускати на різних операційних системах, включаючи Windows, macOS і Linux. Це робить його гнучким варіантом для розробки аутентифікації на різних пристроях та платформах. Не зважаючи на всі плюси, стосовно бібліотек з роботою NFC підтримка може бути обмеженою порівняно з іншими мовами програмування.

В даній роботі для створення сайту з функцією авторизації через NFC будемо використовувати JavaScript для програмування NFC на веб-сайті через певні переваги а саме:

− **Кросплатформеність:** JavaScript є мовою, яка підтримується всіма сучасними браузерами. Це означає, що ви можете написати код один раз і запустити його на будь-якому пристрої або браузері, що підтримує NFC.

− **Веб-орієнтованість**: JavaScript зазвичай використовується для розробки веб-додатків, тому використання його для програмування NFC на веб-сайті дозволяє забезпечити єдиний інтерфейс для користувачів, незалежно від пристрою або операційної системи.

− **Багатофункціональність**: JavaScript має широкі можливості і надає доступ до різноманітних API, які дозволяють взаємодіяти з різними пристроями та периферійними пристроями, включаючи NFC. Ви можете зчитувати, записувати та обробляти дані з NFC-тегів за допомогою різних методів та функцій, які надаються JavaScript-бібліотеками або стандартними API, такими як Web NFC API.

− **Розширюваність**: JavaScript має велику спільноту розробників та багато сторонніх бібліотек і фреймворків, які допомагають вам виконувати складні завдання, включаючи роботу з NFC. Ви можете використовувати ці інструменти для швидкого розроблення функціональності NFC на своєму веб-сайті.

Загалом, використання JavaScript для програмування NFC на веб-сайті є гнучким і широко підтримуваним підходом, який дозволяє вам створити потужну функціональність NFC на вашому веб-сайті для взаємодії з NFC-пристроями. [5-12]

2.2. Особливості програмування NFC за допомогою JavaScript

Щоб почати використовувати функції JavaScript для авторизації на сайті спочатку потрібно зрозуміти основні кроки а саме:

Перевірка підтримки NFC у браузері. Щоб перевіри чи підтримує браузер считування та обробку NFC міток за допомогою JavaScript потрібно використати наступний код :

```
if (`NDEFReader` in window) {
   console.log('Браузер підтримує NFC.');
} else {
   console.log('Браузер не підтримує NFC.');
}
```

Рис. 2.1. Перевірка підтримки браузером NFC

Цей код буде вказувати в консолі розробника чи підтримує браузер “WEB NFC API”

Далі для того щоб почати зчитування даних з NFC-мітки потрібно використовувати методи API, які надаються Web NFC API або іншими бібліотеками за бажанням. Приклад зображено на (Рис. 2.2.)

```
  const reader = new NDEFReader();
reader.scan().then(() => {
  reader.addEventListener('reading', event => {
    const message = event.message;


  });
});
```

Рис. 2.2. Метод зчитування данних.

В цьому прикладі код отримує повідомлення з NFC-тегу а також можна вписати певні команди для подальшого оброблення отриманих данних яке може включати перевірку та валідацію даних, порівняння зі збереженими даними користувача або виконання інших необхідних перевірок. Наприклад, перевірка, чи існує в базі даних запис, що відповідає зчитаному значенню NFC-тега.

Після обробки даних потрібно передати їх на сервер для подальшої авторизації та обробки. Використовуючи методи API, такі як Fetch або XMLHttpRequest, для відправлення даних на сервер. Наприклад:

```
fetch('/auth', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json'
    },
    body: JSON.stringify(data)
  })
  .then(response => {

  })
```

Рис. 2.3. Метод відправлення данних на сервер.

В випадках коли за будь яких причин не вдалося отримати відповідь від бази даннити та сталися помилка потрібно використати команду

.catch(error => {}) яка повідомить про помилку та вивиде це на екран На сервері потрібно перевірити передані дані, виконати авторизацію та надати доступ до відповідних ресурсів на веб-сайті. Відповідь сервера може містити додаткову інформацію про статус авторизації та доступ до ресурсів, яку потрібно обробити на клієнтській стороні. [13]

2.3. Опис файлової структури та принципу роботи

Перш за все, структура поділяється на 2 частини клієнтську та серверну. Клієнтська частина являє собою веб сторінку з функціями аунтифікаії на сайті реалізовану за допомогою JavaScript. Задопомогою цієї сторінки користувас може завантажити свій NFC код до сайту для регістрації та аунтифікації.

Серверна частина відповідає за пошук інформаціїї стосовно введеного NFC коду. Після того як користувач ввів свій код інформаія відправляється на сервер для подальшої обробки а саме якщо код був зарегестрований відправляє данні для автоматичної аунтифікації на сайті, в випадку коли інформація не знаходилась в базі данних відправляє до користувача повідомленя що данний NFC кода не має в базі данний та пропонує зарегеструвати його як нового користувача, після регестрації новий код буде внесено до бази данний для подальшого використання цього коду для аунтифікації

Поділ на клієнтську та серверну частини для аутентифікації на сайті має ниску переваг а саме:

Безпечність відправки данних. Клієнтська частина відповідає за отримання та обробку даних з NFC-тегу, виконання локальної перевірки та передачу оброблених даних на сервер. Тим самим, конфіденційні дані, такі як ідентифікатор NFC або користувальницькі дані, не відправляються напряму на сервер, що зменшує ризик їх перехоплення.

Швидкість обробки данних. Клієнтська частина може виконувати швидкі операції на мобільному пристрої без необхідності звертатися до сервера. Це полегшує обробку та валідацію даних, а також забезпечує користувачеві швидку відповідь від системи.

Перевірка та обробка даних з NFC-тегу що відбувається на клієнтському пристрої, дозволяє зменшити навантаження на сервер. Сервер виконує лише фінальну перевірку та авторизацію, що покращує масштабованість та продуктивність системи.

Гнучкість виористання. оділ на клієнтську та серверну частини дозволяє використовувати різні технології та мови програмування на кожній з частин. Це дозволяє використовувати найкращі інструменти для кожної з частин та забезпечує гнучкість у виборі технологій.

Стосовно клієнської частини сайту, візуальну частину сайти виконує файли стилістичних таблиць типу CSS вони вказують стилі для відображення інтерфейса користувача.

CSS є мовою розмітки стилів, яка використовується для візуального оформлення веб-сторінок і додавання стилів до HTML-документів. Вона визначає, як елементи HTML повинні відображатися на екрані, включаючи кольори, шрифти, розташування, розміри, відступи, фонові зображення та інші властивості.

Основні особливості CSS це розділення опису вигляду сторінки від основного контенту. Тобто це створення окремого файлу CSS і підключити його до HTML-документу, щоб застосувати стилі до всіх елементів на сторінці без необхідності змінювати сам HTML-код. Це полегшує підтримку і зміну стилю веб-сайту.

Каскадність CSS дозволяє визначати пріоритети стилів. Де можливо встановлювати загальні стилі для елементів, а потім перевизначати їх для конкретних елементів або класів. Це дозволяє зручно керувати виглядом окремих елементів і забезпечує більшу гнучкість у налаштуванні стилів.

Селектори в CSS використовують для вибору елементів, до яких будуть застосовані певні стилі. Селектори можуть бути базовими (наприклад, назви тегів) або деталізованими (за атрибутами, класами, ідентифікаторами тощо). Це дозволяє точно визначати, які елементи повинні бути стилізовані.

CSS надає широкий спектр властивостей, які можна використовувати для задання стилів елементів. Деякі з них включають кольори, шрифти, розміри, відступи, рамки, фонові зображення, анімацію, трансформації та багато іншого. Значення властивостей можуть бути задані в різних форматах, наприклад, кольори можуть бути вказані в HEX-коді, RGB-значенні або назві кольору.

Також в CSS використовується ієрархічна структура для організації стилів. Завдяки чому можливо задати стилі для окремих елементів, а також вкладати їх в різні контейнери і групи. Це дозволяє створювати складні макети і контролювати зовнішній вигляд елементів на основі їх розташування у структурі документа.

CSS надає можливості для створення респонсивного дизайну, що адаптується до різних розмірів екранів. За допомогою медіа-запитів можна застосовувати різні стилі до пристроїв з різними розмірами екранів, що забезпечує оптимальне відображення веб-сторінок на різних пристроях.

Завдяки цьому найкращим вибором буде використання CSS для візуального оформлення веб-сторінок. Він дозволяє розділити стиль від контенту, забезпечує гнучкість і легкість зміни стилю, а також дозволяє створювати привабливі та професійні веб-сайти з візуально привабливим інтерфейсом.

Файлова структуру проекту складається з наступних елементів:

− Html файл - Це основний HTML-файл, який буде відображатись при завантаженні веб-сайту. У ньому будуть застосовуватися необхідні теги та скрипти для роботи з NFC та аутентифікацією.

− Css файл – Файл завдяки якому буде налаштована візуальна частина веб-сторінки з особливими правилами для різних елементів.

− Js файл - Цей файл JavaScript використовуватиметься для реалізації логіки NFC та аутентифікації. В ньому знаходиться код для взаємодії з NFC-рідером, отримання даних з міток NFC, аутентифікації та взаємодії з сервером. Також цей файл можна включити як модуль до Html файлу.

− php файл це сервер який потрібен для створееня скриптів які будуть аунтифікувати користувача на сайті.

Принцип роботи аутентифікації на веб-сайті в мобільному браузері за допомогою NFC:

Підготовка сторінки: В HTML-файлі потрібно підключити необхідні бібліотеки та скрипти, які дозволять взаємодіяти з NFC-рідером. Також включаємо кнопку або будь-який інший елемент, який можна натиснути, щоб розпочати процес аутентифікації.

Натискання кнопки: Користувач натискає кнопку на сторінці, що починає процес аутентифікації.

Зчитування NFC: За допомогою JavaScript (у файлі .js) створюємо обробник подій для кнопки, який викликає функцію для зчитування NFC. Ця функція запускає NFC-рідер на мобільному пристрої та очікує моменту, коли мітка NFC буде зближена з рідером.

Отримання даних NFC: Коли мітка NFC зближена з рідером, скрипт отримує дані з мітки. Ці дані можуть містити ідентифікатор або будь-яку іншу інформацію, яка буде використовуватися для аутентифікації.

Відправка на сервер: З отриманими даними NFC скрипт може виконати запит до сервера для аутентифікації. Серверна сторона перевіряє отримані дані та надіслати відповідь, що підтверджує успішну аутентифікацію або невдачу.

Результат аутентифікації: Залежно від результату аутентифікації, скрипт відображає відповідні повідомлення або виконати інші дії на сторінці.

ВИСНОВКИ ДО РОЗДІЛУ 2

Ретельно дослідивши всі необхідні елементи для створення сайту аунтифікації за допомогою NFC можна обрати якими інструментами буде краще оперувати для досягнення мети. Використовуючи JavaScript в поєднанні з PHP та Html можливо зробити функціональний сайт який буде раціонально розподіляти навантаження на сервера, виконуючи більшість операцій на самому пристрої без додаткових обробіток данних на сервері та базі данних. Також використовуючи стилі CSS можливо вдосконалити візуальну клієнську частину сайту шляхом додавання фонів, побудовою вікон та багато іншого.

РОЗДІЛ 3
СТВОРЕННЯ САЙТУ

3.1. Створення головної Html сторінкі з аунтифікацією на сайті за допомогою NFC

Для початку В <head>-частині сторінки встановлюємо налаштування метатегів, які визначають кодування символів, сумісність зі старими версіями Internet Explorer та налаштування масштабування веб-сторінки для пристроїв з різною шириною екрану. Також підключається зовнішній файл стилів style.css.

```
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login</title>
    <link rel="stylesheet" href="css/style.css">
</head>
```

Рис. 3.1. Підключення стилів CSS.

У частині коду <body>- розміщяємо контейнер з класом auth-wrapper, який містить форму завдяки якій буде відбуватися авторизації на сайті.

Форма auth-wrapper має клас auth і містить два <input>-поля: одне для введення логіну (з атрибутом id="login") та інше для введення пароля (з атрибутом id="password").

```
<div class="auth-wrapper">
    <form class="auth">
        <input class="auth__inp" id="login" type="text" name="login" placeholder="Login">
        <input class="auth__inp" id="password" type="password" name="pass" placeholder="Password">
```

Рис. 3.2. Форма аунтифікації

Додаємо в внутрішню частину з класом auth__btn-box що містить дві кнопки: кнопку для входу (auth__login-btn) і кнопку для реєстрації (auth__signup-btn). Присвоюємо кнопкам атрибут onclick, який буде вказувати на виклик відповідних функцій loginUser() і signupUser() при їх натисканні.

```
<div class="auth__btn-box">
    <button class="auth__btn auth__login-btn" type="button" onclick="loginUser()">Log In</button>
    <button class="auth__btn auth__signup-btn" type="button" onclick="signupUser()">Sign Up</button>
</div>
```

Рис. 3.3. Створення кнопок.

Далі додаємо ще один внутрішній з класом auth__nfc-box який буде містити текст "Or" і кнопку для входу за допомогою NFC (auth__nfc-btn). Цій кнопці присвоюємо атрибут onclick, який буде вказувати на виклик функції readNFC() при натисканні.

```
<div class="auth__nfc-box">
    <p>Or</p>
    <button class="auth__btn auth__nfc-btn" onclick="readNFC()">Log In with NFC</button>
</div>
```

Рис. 3.4. Виклик функції readNFC.

Використовуючи команду

<script src="file_name.js"></script>

Ми будемо ссилатися на файл JavaScript для обробки введених данний при аунтифікації.

Для файлу JavaScript створюємо об'єкт nfcOptions з опцією mediaType, яка буде вказувати на тип даних, очікуваних з NFC-тегу. Будемо використовувати данні типу text/utf-8.

```
const readNFC = function() {
    const nfcOptions = { mediaType: "text/utf-8" };
```

Рис. 3.5. Створення об'єкту nfcOptions.

Для надання методів сканування та читання даниз з NFC-тегу створемо екземпляр NDEFReader.

```
const ndef = new NDEFReader();
```

Рис. 3.6. Створення NDEFReader.

Додаєм прослуховувач подій reading, який викликається, коли дані з NFC-тегу успішно зчитуються. Він містить функцію, яка відправляє отримані дані на сервер для авторизації. Данними які будуть считуватися є серійний номер (serialNumber) зчитаного NFC-тегу, він буде використовуватися як логін для авторизації.

```
ndef.addEventListener("reading", ({ serialNumber }) => {

    fetch("http://mysite/auth.php", {
        method: "POST",
        body: JSON.stringify({
            login: serialNumber,
            pass: "",
        }),
        headers: { "Content-type": "application/json; charset=UTF-8" },
    })
        .then((response) => response.json())
        .then((json) => console.log(json));
});
```

Рис. 3.7. Додавання прослуховування подій.

Сканування NFC буде виконуватися за допомогою методу ndef.scan(nfcOptions). Якщо сканування успішне, виводиться повідомлення "Scan started successfully." у консолі. Якщо під час сканування сталася помилка, виводиться повідомлення про помилку у консолі.

```
    ndef.scan(nfcOptions)
        .then(() => {
            console.log("Scan started successfully.");

        })
        .catch((error) => {
            console.log(`Error: ${error}`);

        });
};
```

Рис. 3.8. Створення виводу повідомлення у консоль.

Після аунтифікації коритувача перенесе на головну сторінку сайту з деякою інформацією яка має такий вигляд:

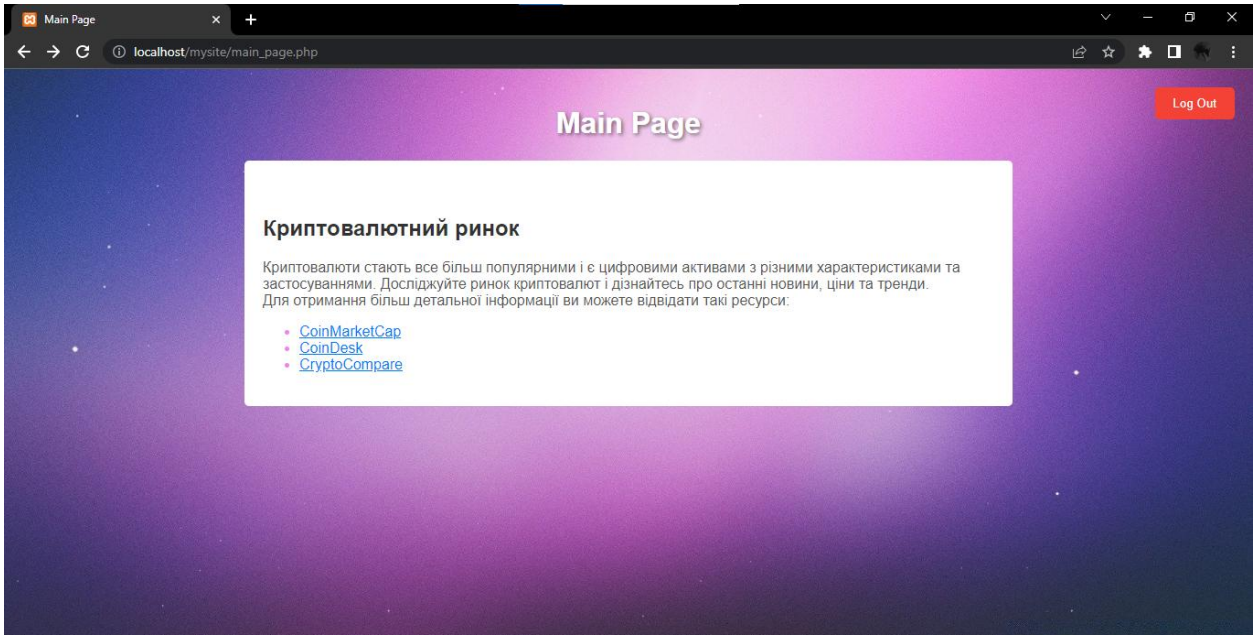

Рис. 3.9. Основна частина сайту.

Для повернення на сторінку аунтифікації треба натиснути кнопку Logg out

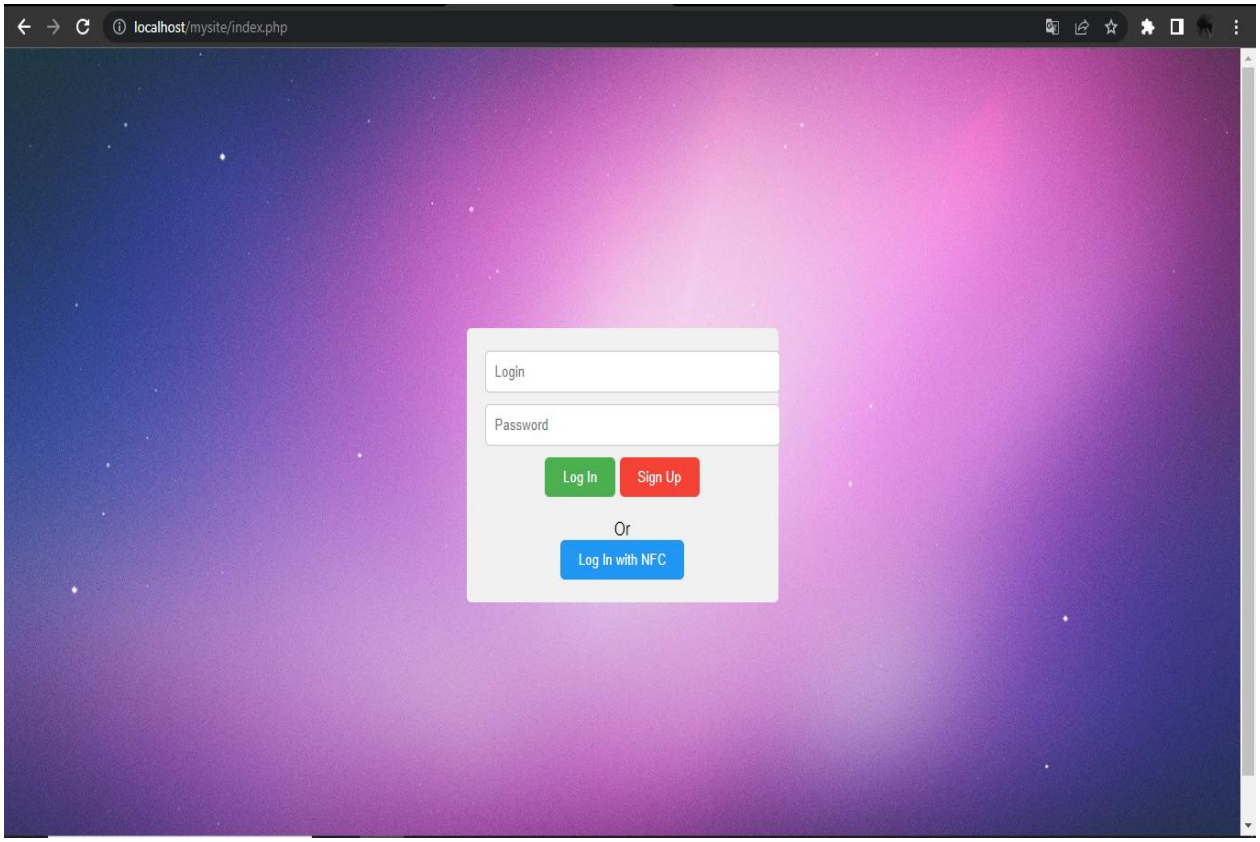

Рис. 3.10. Меню аунтифікації.

3.2. Створення бази данних для сайту

При скануванні NFC-тегу, використовується отриманий серійний номер (serialNumber) з NFC-тегу як ідентифікатор користувача. JavaScript-код може відправляти цей серійний номер на сервер, який потім може перевірити, чи існує користувач з таким серійним номером в базі даних. База данних була створення задопомогою PG4 admin та має такий вигляд на мові програмування SQL

```sql
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";


CREATE TABLE `users` (
  `id` int(11) NOT NULL,
  `login` varchar(32) DEFAULT NULL,
  `password` varchar(32) DEFAULT NULL,
  `nfc` varchar(255) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;


INSERT INTO `users` (`id`, `login`, `password`, `nfc`) VALUES
(1, 'admin', 'admintest', NULL);


ALTER TABLE `users`
  ADD PRIMARY KEY (`id`),
  ADD UNIQUE KEY `login` (`login`);


ALTER TABLE `users`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;
COMMIT;
```

Рис. 3.11. База данних.

Цей код Виконує налаштування SQL-режиму:

− встановлює режим NO_AUTO_VALUE_ON_ZERO, який забороняє автоматичне присвоєння значення 0 стовпцям з автоінкрементом.

− Запускає транзакції і встановлює часовий пояс.

− Створює таблицю users з наступними стовпцями: id, login, password і nfc. Типом таблиці є InnoDB, а набір символів буде встановлей - utf8.

− Вставляє один рядок даних в таблицю users. Цей рядок має значення id рівне 1, login рівне 'admin', password рівне 'admintest', а nfc має значення NULL.

− Встановлює індекси для таблиці users. Первинний ключ (id) і унікальний ключ (login).

− Встановлює автоінкремент для стовпця id в таблиці users. Значення AUTO_INCREMENT=2 говорить про те, що наступне автоматично згенероване значення буде 2.

3.3. Створення серверної частини сайту

Для створення серверної части будемо використовувати скрипт PHP. Він буде відповідати за обробку запиту на аутентифікацію користувача з використанням введеного логіну та пароля. Цей код буде виконувати запит до бази даних, порівнює логін та пароль з таблиці users зі значеннями, отриманими з POST-запиту, і повертати результат у форматі JSON. Залежно від результату запиту, може бути повернено порожній масив або масив об'єктів з даними про користувача.

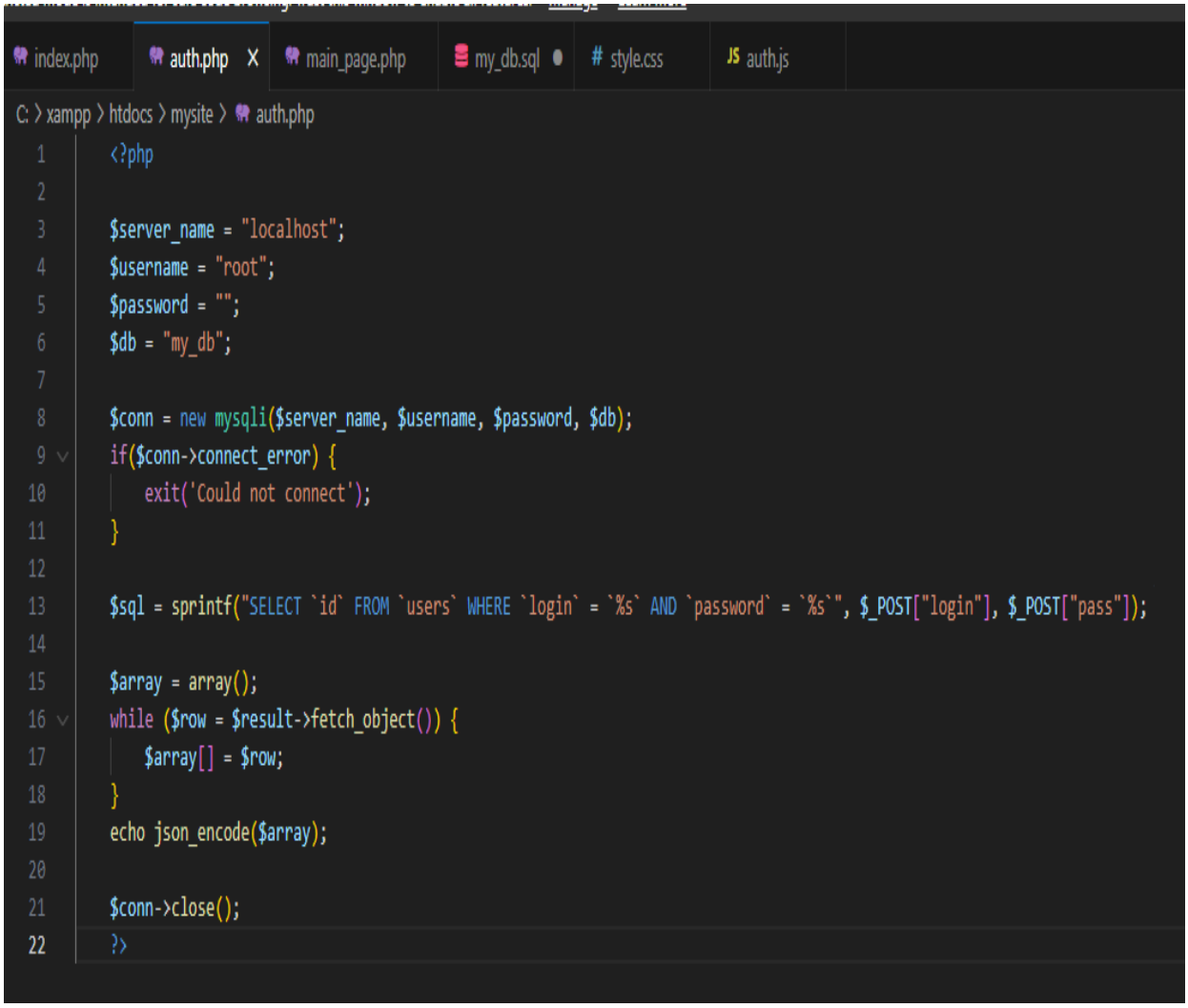

Рис. 3.12. Серверна частина.

В цьому коді на малюнку Визначаються змінні для налаштування підключення до бази даних: $server_name (ім'я сервера), $username (ім'я користувача бази даних), $password (пароль користувача бази даних) і $db (ім'я бази даних).

Виконується підключення до бази даних з використанням зазначених змінних. Якщо підключення не вдається (наприклад, неправильні дані для підключення), виконується вихід з програми з повідомленням "Could not connect".

Згенерований SQL-запит для вибірки даних з таблиці users, де логін (login) та пароль (password) повинні відповідати значенням, отриманим з POST-запиту. Зауважте, що в коді є помилка - символи %s в запиті мають бути оточені одинарними лапками замість зворотних лапок.

Ініціалізується порожній масив $array, в який будуть додаватися результати запиту. Запускається цикл while, який проходить через кожний рядок результату запиту ($result) і додає його в масив $array у вигляді об'єкта.

За допомогою функції json_encode масив $array перетворюється в рядок JSON і виводиться за допомогою функції echo. Закривається з'єднання з базою даних за допомогою методу close().

3.4. Огляд роботи системи аунтифікації

Розглянемо роботу вікна аунтифікації

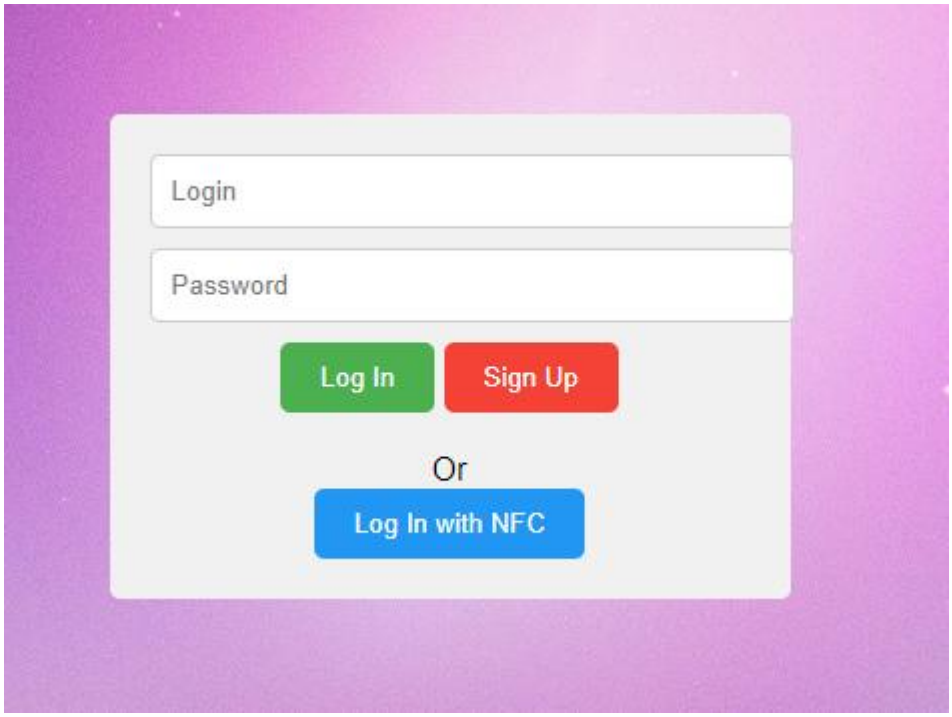

Рис. 3.13. Вікно аунтифікації.

Для початку розглянемо кнопку «Sing up» при натисканні на неї в нас буде відкриватися вікно регістрації.

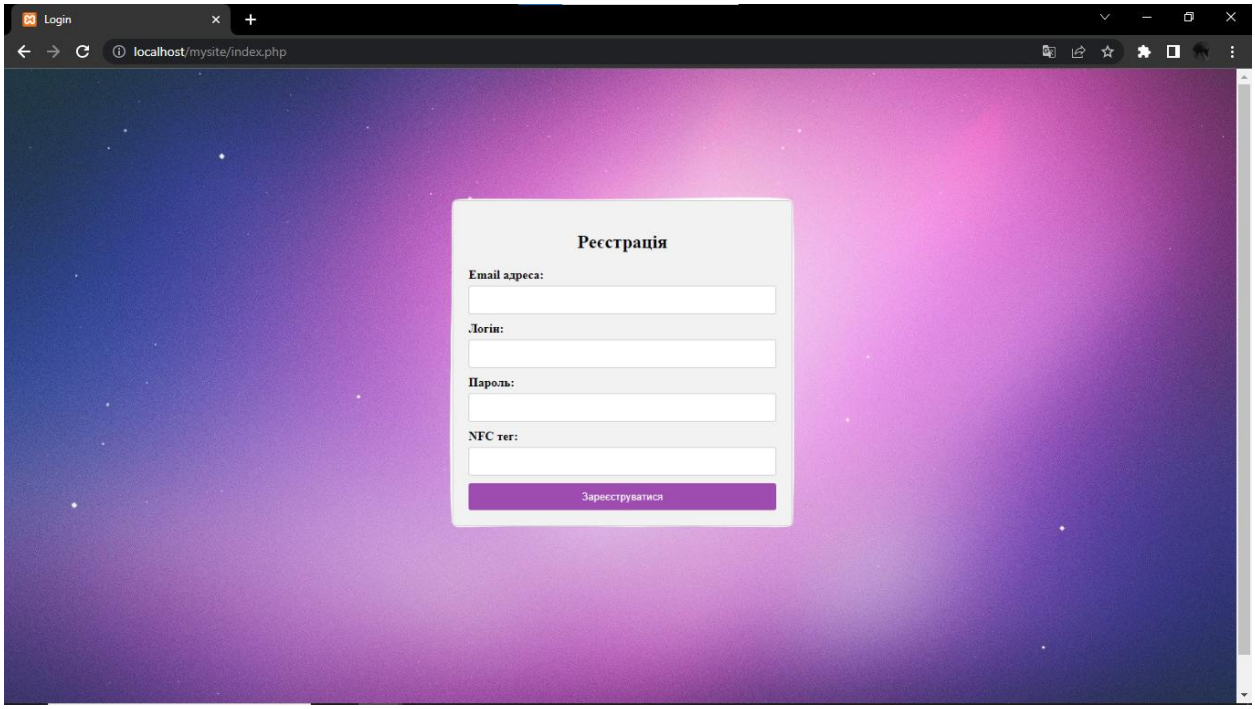

Рис. 3.14. Вікно регистрації.

Після регістрації нас перекине знову до початкового вікна і після введення наших данних перейдемо до основної сторінки.

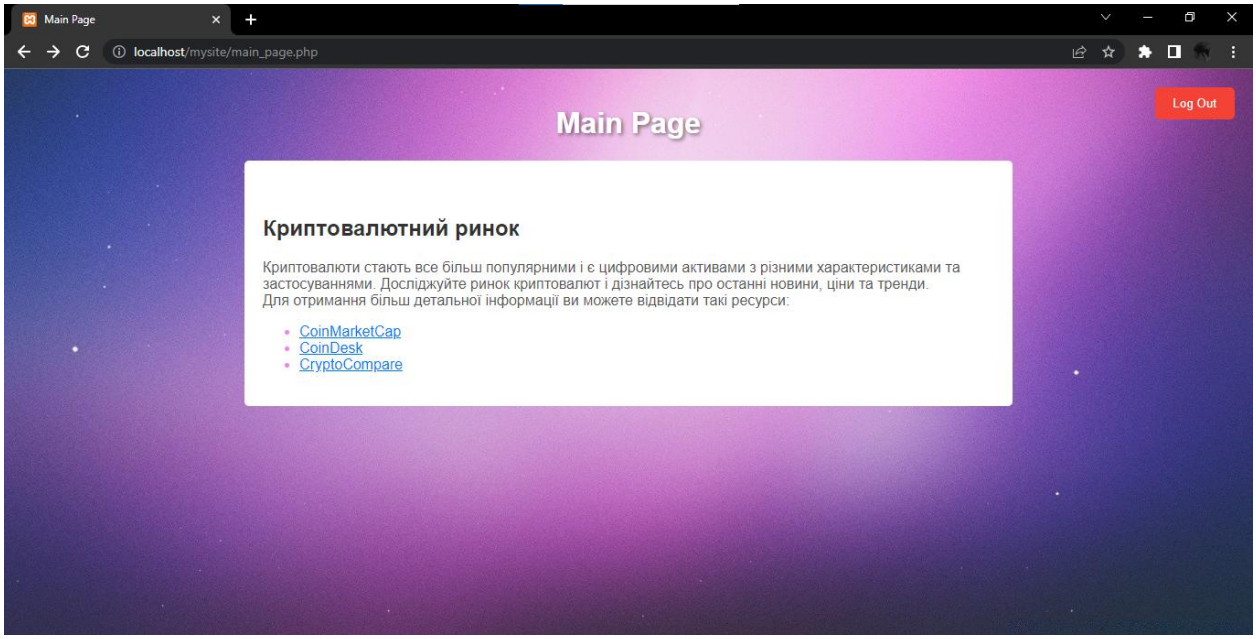

Рис. 3.15. Головна сторінка.

Демонстрація декількох посилань, які відкриють сайти з інформацією стосовно крипто валют яка оновлюється в реальному часі представлена на

рис. 3.16.

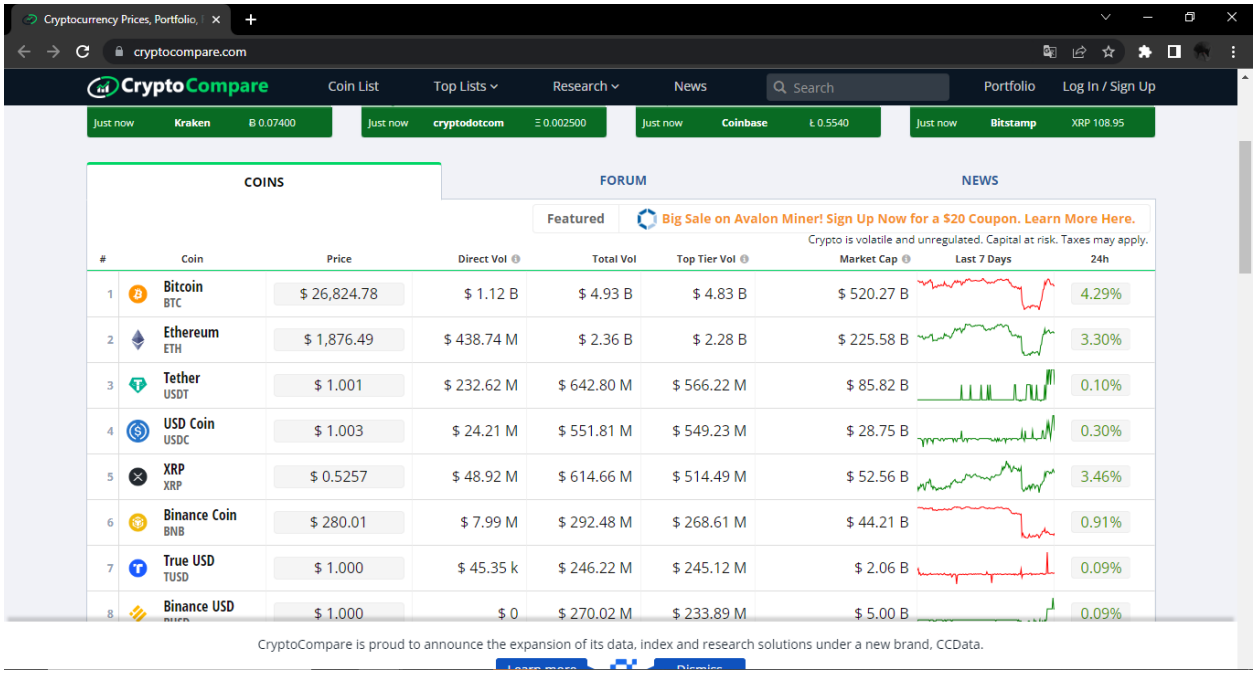

Рис. 3.16. Посилання на головній сторінці.

Натиснувши кнопку на основному сайті «Log Out» відкриеться вікно аунтифікації. Якщо натиснути кнопку «Log in with NFC» в фоновому режими браузер почне зчинування NFC тегу для подальшого вводу його в поля. Після декількох не вдалих спроб аунтифікування на початковомувікні виступить кнопка «ForgotPassword?»

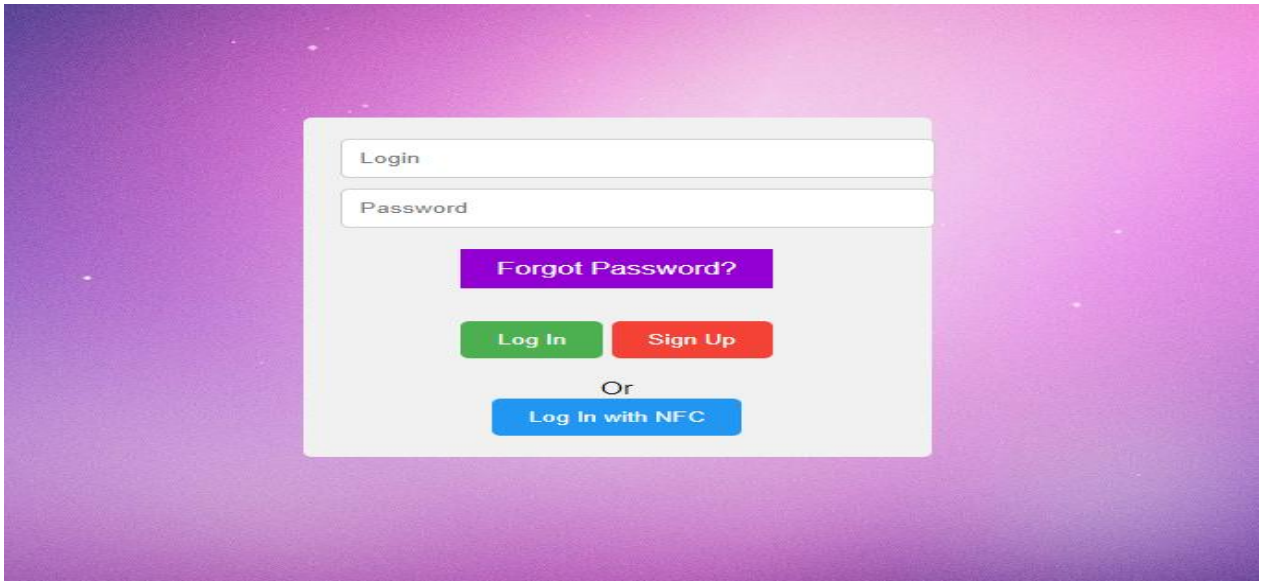

Рис. 3.17. Кнопка «ForgotPassword?».

При натискані на яку відкриеться діалогове вікно для відправки нового пароля на вже зарегестровану почту

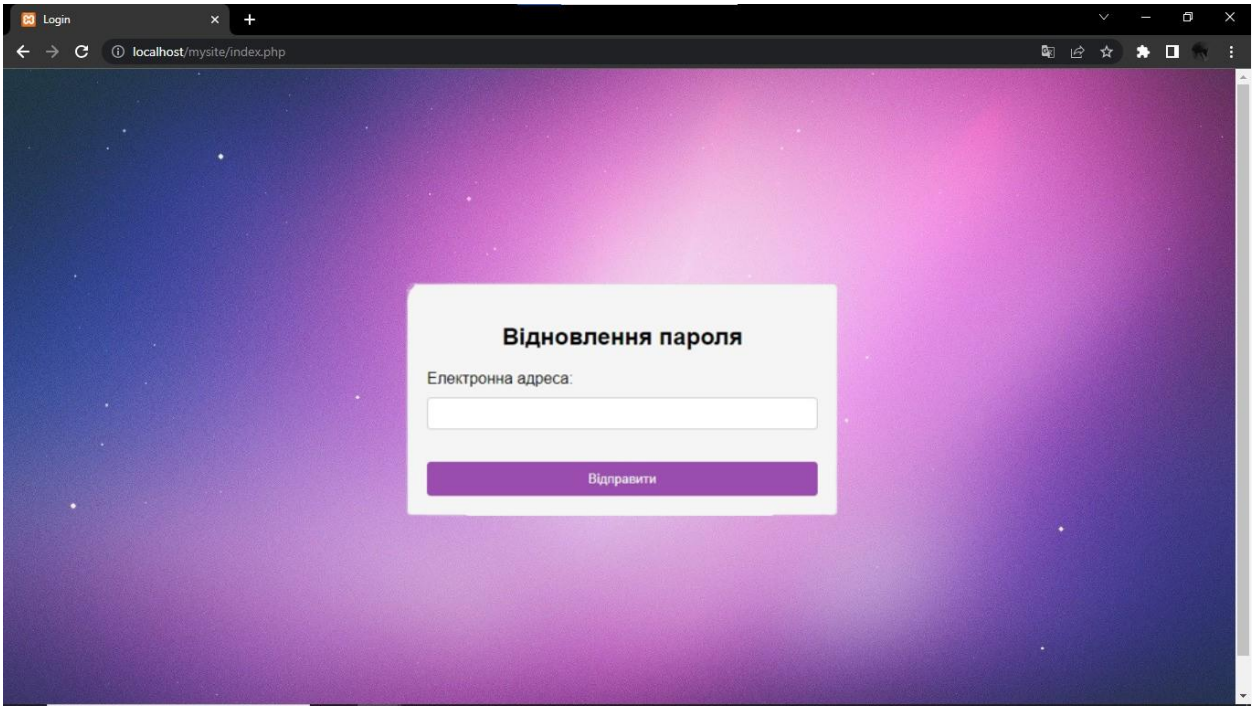

Рис. 3.18. Вікно відновлення паролю.

Вся данна робота може використовуватися як макет для створення Сайту з функцією ауетифікації за допомогою NFC.

ВИСНОВКИ ДО РОЗДІЛУ 3

В цьому розділі було створенно HTML сторінку з аутентифікацією на сайті за допомогою NFC. Також було наведено кроки для створення необхідних елементів на сторінці та налаштування серверної частини.
Було описано структуру HTML сторінки з формою для авторизації.

Використування метатегів для налаштування кодування символів та сумісності з різними браузерами. Підключення зовнішніх файлів стилів. На сторінці було розміщєнно контейнер з формою, яка містить поля для введення логіну та пароля. Також додано кнопки для входу та реєстрації, а також кнопка для входу за допомогою NFC. Для обробки введених даних при аутентифікації використовувався файл JavaScript.

Також було розглянено опис створення бази даних для зберігання інформації про користувачів. Використування SQL для створення таблиці зі стовпцями для ідентифікатора, логіну, пароля та NFC-даних. Також встановлення індекса та автоінкременту для таблиці.

Створення серверної частини сайту з використанням PHP. Код відповідає за обробку запиту на аутентифікацію користувача. Він підключається до бази даних і виконує запит, щоб перевірити введені логін та пароль. Результат запиту повертається у форматі JSON. Залежно від результату запиту, можуть бути повернуті порожній масив або дані про користувача.

В цілому, використовуючи поєднання CSS Html JavaScript PHP та SQL ми можемо реалізувати створення сайту в мобільному браузері в якому буде можливість аунтифікуватися та зарегеструватися за допомогою NFC-коду.

ВИСНОВКИ

Розглядаючи Технологія NFC (Near Field Communication) в кваліфікаційній роботі було проаналізовано що вона пройшла великий шлях розвитку, від використання RFID до широкого застосування в різних галузях. Вона є бездротовою зв'язковою технологією на основі радіочастотної ідентифікації, яка виявилась надзвичайно корисною і популярною в нашому повсякденному житті.

NFC дозволяє швидку і безпечну передачу інформації між двома пристроями, які знаходяться в непосредственній близькості один від одного. Ця технологія є простою у використанні, не потребує складних налаштувань або підключень. Завдяки високій швидкості передачі даних, NFC забезпечує швидкий обмін інформацією, а протоколи шифрування гарантують безпеку передачі даних.

Історія розвитку технології NFC свідчить про поступове вдосконалення і розширення її застосувань. Вона виникла на основі технології RFID і пройшла кілька етапів розвитку, починаючи зі створення технології RFID і закінчуючи впровадженням NFC в мобільних пристроях. Сьогодні NFC використовується в безконтактних платежах, доступі до приміщень, обміні даними між пристроями та в багатьох інших сферах.

В цілому, технологія NFC стає необхідним інструментом у нашому сучасному світі, полегшуючи багато щоденних процесів і сприяючи зручності та безпеці в обміні інформацією. За допомогою NFC ми можемо швидко та легко здійснювати покупки, передавати файли, обмінюватися контактами та виконувати інші завдання, що робить наше життя більш ефективним та зручним.

Для створення сайту з аутентифікацією за допомогою NFC було використовувано комбінацію JavaScript, PHP, HTML та CSS. JavaScript і PHP які забезпечили функціональність сайту та розподіл навантаження на сервер і пристрій користувача. HTML та CSS використовуються для створення структури та візуального оформлення сторінки.

На сторінці була створена форма для авторизації, використовуючи HTML. Вона може містити поля для введення логіну та пароля, а також кнопки для входу та реєстрації. Додатково була реалізована кнопка для входу за допомогою NFC.

Для обробки введених даних при аутентифікації було використувано JavaScript. Він взаємодіє з серверною частиною, написаною на PHP, шляхом відправки запиту на перевірку логіну та пароля. Результат запиту повертається у форматі JSON і відображений на сторінці.

Для зберігання інформації про користувачів була спроектована базу даних і використувана таблиця SQL з необхідними стовпцями, такими як ідентифікатор, логін, пароль та NFC-дані. PHP може здійснювати підключення до бази даних та виконувати запити для перевірки інформації при аутентифікації.

Використовуючи комбінацію CSS, HTML, JavaScript, PHP та SQL, була реалізовано створення сайту з можливістю аутентифікації за допомогою NFC. Цей сайт може працювати в мобільному браузері та надавати можливість користувачам зареєструватися або увійти за допомогою NFC-коду.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. What is NFC and how does it work автор: Calvin Wankhede [наукова стаття] режим доступу: https://www.androidauthority.com/what-is-nfc-270730/
2. Функція NFC в смартфоні: навіщо потрібна і як її використовувати [наукова стаття] режим доступу: https://hotline.ua/guides/funkcya-nfc-v-smartfon-navscho-potrbna--yak--vikoristovuvati/
3. Що таке NFC у телефоні автор: Stanislav Shulga [наукова стаття] режим доступу:https://blog.portmone.com.ua/uk/posts/information-posts-uk/shho-take-nfc-u-telefoni
4. Near Field Communication [наукова стаття] режим доступу: https://uk.wikipedia.org/wiki/Near-field_communication
5. Що таке JavaScript [наукова стаття] режим доступу: https://dou.ua/lenta/articles/how-to-learn-javascript/
6. Advantages and Disadvantages of HTML5 автор: Pranjali Nandan [наукова стаття] режим доступу: https://www.codementor.io/@pj0613/advantages-and-disadvantages-of-html5-1rbr08003f\
7. HTML Commands автор: Priya Pedamkar [наукова стаття] режим доступу: https://www.educba.com/html-commands/
8. HTML5 автор: Robert Sheldon [наукова стаття] режим доступу: https://www.techtarget.com/whatis/definition/HTML5
9. Що таке php? [наукова стаття] режим доступу: https://freehost.com.ua/ukr/faq/wiki/chto-takoe-php/
10. Переваги та недоліки Java [наукова стаття] режим доступу: https://from-ua.info/perevahy-ta-nedoliky-java/
11. The Joy of PHP Programming автор: Alan Forbes. Видавництво: Leanpub 2020.-128 с. [наукова робота] режим доступу: https://kupdf.net/download/the-joy-of-php-alan-forbes_58ebadaddc0d60cb15da9816_pdf

12. Effective Java автор: Joshua Bloch Видавництво: Addison-Wesley 2017.-219 с. [наукова робота] режим доступу: https://org2.knuba.edu.ua/pluginfile.php/58037/mod_resource/content/1/Dzhoshua_Blokh_Java_Effektivnoe_programmirovanie.pdf

13. Head First JavaScript Programming: A Brain-Friendly Guide автори: Elizabeth Freeman and Eric Freeman Видавництво: O`reilly Media 2014.-702 с. [наукова робота] режим доступу: https://www.academia.edu/36424852/OREILLY_Head_First_JavaScript_Programming_A_Brain_Friendly_Guide_A_learners_guide_to

14. Near-field communication [наукова стаття] режим доступу: https://en.wikipedia.org/wiki/Near-field_communication