НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА ШТУЧНОГО ІНТЕЛЕКТУ

На правах рукопису
УДК 004.852

До захисту допущено
В. о. зав. кафедри ШІ
О. І. Чумаченко
«___» _____ 2022 р.

# Магістерська дисертація

на здобуття ступеня магістра за спеціальністю 122 «Комп'ютерні науки»

на тему: «Методи багатофакторної автентифікації до веб додатків за допомогою штучного інтелекту та технології блокчейн»

Виконав:
студент 2 курсу, групи КІ-11мп
Славінський Всеволод Олександрович _____

Керівник старший викладач
кафедри штучного інтелекту
КПІ ім. Ігоря Сікорського,
доктор філософії Гуськова В. Г. _____

Рецензент: зав. кафедри СП,
д.т.н. проф. Мухін Вадим Євгенович _____

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів
без відповідних посилань

Студент (*підпис*): _____

Київ
2022

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА ШТУЧНОГО ІНТЕЛЕКТУ

Рівень вищої освіти — другий (магістерський)
Спеціальність (ОПП) — 122 «Комп'ютерні науки»

ЗАТВЕРДЖУЮ
В. о. завідувача кафедри ШІ
О. І. Чумаченко
«___» _____ 2022 р.

**ЗАВДАННЯ**

на магістерську дисертацію студенту Славінському Всеволоду Олександровичу

1. Тема дисертації: «Методи багатофакторної автентифікації до веб додатків за допомогою штучного інтелекту та технології блокчейн», науковий керівник дисертації Гуськова Віра Генадіївна, старший викладач кафедри штучного інтелекту КПІ ім. Ігоря Сікорського, доктор філософії, затверджені наказом по університету від «03» листопада 2022 р. № 4046-с

2. Термін подання студентом дисертації: 12.12.2022 р.

3.Об'єкт дослідження: мультифакторна авторизація та автентифікація в мережі Web3.0

4. Предмет дослідження: нейронні мережі для розпізнавання облич та блокчейн для роботи з розподіленими ідентифікаторами

5. Перелік завдань, які потрібно зробити:

1) здійснити огляд технічної літератури за темою роботи;

2) дослідити актуальність обраної теми;

3) ознайомитись із існуючими методами автентифікації

4) здійснити порівняльний аналіз наявних методів, виявити їх переваги та недоліки;

5) розробити та реалізувати систему, що вирішує задачу розпізнавання облич;

6) розробити та реалізувати систему, що вирішує задачу автентифікації через блокчейн;

7) розробити концептуальні висновки;

8) підготувати ілюстративний матеріал;

9) оформити пояснювальну записку.

6. Перелік графічного матеріалу.

**7.** Дата видачі завдання: 1 вересня 2022 р.

Календарний план

| № з/п | Назва етапів виконання магістерської дисертації | Термін виконання етапів роботи | Примітка |
|---|---|---|---|
| 1. | Вивчення літератури за темою роботи. | 01.09.2022 – 14.09.2022 | Виконано |
| 2. | Підготовка першого розділу. | 14.09.2022 – 01.10.2022 | Виконано |
| 3. | Підготовка другого розділу. | 01.10.2022 – 10.10.2022 | Виконано |
| 4. | Розробка програмного продукту. | 10.10.2022 – 01.11.2022 | Виконано |
| 5. | Підготовка третього розділу | 01.11.2022 – 18.11.2022 | Виконано |
| 6. | Підготовка частини стартап-проєкту | 18.11.2022 – 21.11.2022 | Виконано |
| 9. | Концептуальні висновки. Перспективи розвитку отриманих рішень | 21.11.2022 – 25.11.2022 | Виконано |
| 10. | Оформлення пояснювальної записки | 25.11.2022 – 12.12.2022 | Виконано |

Студент Славінський Всеволод

Науковий керівник дисертації Гуськова Віра

# РЕФЕРАТ

Магістерська дисертація: 124 с., 22 табл., 24 рис., 23 джерел, 1 додаток.

АВТЕНТИФІКАЦІЯ, МУЛЬТИФАКТОРНА АВТЕНТИФІКАЦІЯ, БЛОКЧЕЙН, WEB3.0, РОЗПІЗНАВАННЯ ОБЛИЧ, ВЕБ ЗАСТОСУНОК, КРИПТОГАМАНЦІ, РОЗПОДІЛЕНІ АВТЕНТИФІКАТОРИ, РОЗПОДІЛЕНИЙ ДОДАТОК

Об'єктом дослідження є мультифакторна авторизація та автентифікація в мережі Web3.0

Предметами дослідження визначено нейронні мережі для розпізнавання облич та блокчейн для роботи з розподіленими ідентифікаторами

Мета дослідження полягає у аналізі алгоритмів видачі рекомендацій, що базуються на моделях матричної факторизації, а також алгоритмів, що використовують аппарат мереж глибокого навчання.

У роботі розглянуто альтернативні методи авторизації, такі як авторизація через біометрію, а також новітній, автентифікація через блокчейн.

Важливість роботи полягає у створенні прикладу використання мультифакторної автентифікації на основі нейронних мереж і блокчейну в мережі Web3.0. Отримані результати надалі можна буде розвинути в самостійну бібліотеку, яку зможуть використовувати незалежні розробники для розроблення своїх додатків в розподіленому інтернеті. Дана робота допоможе зробити систему безпечнішою та зручнішою для користувачів, а також дасть змогу перестати зберігати свої дані у великих корпорацій, таких як Google або Facebook.

Результатом дослідження є порівняння існуючих методів авторизації та автентифікації, а також перспективи автентифікації за допомогою блокчейну та біометрії у вебі 3.0.

# ABSTRACT

Master thesis: 124 p., 22 tab., 24 fig., 23 references, 1 appendix.

AUTHENTICATION, MULTIFACTOR AUTHENTICATION, BLOCKCHAIN, WEB3.0, FACE RECOGNITION, WEB APPLICATION, CRYPTO WALLETS, DISTRIBUTED AUTHENTICATORS, DISTRIBUTED APPLICATION

The object of research is multifactor authorization and authentication in the Web3.0 network

The subjects of the study are neural networks for face recognition and blockchain for working with distributed identifiers

The purpose of the study is to analyze recommendation algorithms based on matrix factorization models, as well as algorithms using deep learning networks.

The work considers alternative methods of authorization, such as authorization through biometrics, as well as the latest, authentication through the blockchain.

The importance of the work is to create an example of using multifactor authentication based on neural networks and blockchain in the Web3.0 network. The results obtained can be further developed into an independent library that can be used by independent developers to develop their applications in the distributed Internet. This work will help to make the system safer and more convenient for users and will allow to stop storing their data in large corporations such as Google or Facebook.

The result of the study is a comparison of existing methods of authorization and authentication, as well as the prospects for authentication using blockchain and biometrics in Web 3.0.

ЗМІСТ

# ВСТУП

У сучасному інтернеті користувачам постійно доводиться мати справу з проблемами автентифікації та авторизації, оскільки практично всі веб-сайти вимагають підтвердження особи. Однак поточні методи авторизації недосконалі, оскільки вони покладаються на застарілі методики, як-от ідентифікація через e-mail і пароль, які дуже легко зламати або ненавмисно передати зловмисником.

У цій роботі ми розглянемо більш захищений варіант автентифікації – мультифакторну автентифікації, а також альтернативні методи авторизації та автентифікації, такі як автентифікація через біометрію за допомогою обличчя завдяки нейронним мережам, а також новітній підхід до автентифікації користувачів, що буде широко розповсюджений у новітньому інтернеті, Web3.0, автентифікація через блокчейн.

Важливість роботи полягає у створенні прикладу використання мультифакторної автентифікації на основі нейронних мереж і блокчейну в мережі Web3.0. Отримані результати надалі можна буде розвинути в самостійну бібліотеку, яку зможуть використовувати незалежні розробники для розроблення своїх додатків в розподіленому інтернеті. Дана робота допоможе зробити систему безпечнішою та зручнішою для користувачів, а також дасть змогу перестати зберігати свої дані у великих корпорацій, таких як Google або Facebook.

Об'єктом дослідження є мультифакторна авторизація та автентифікація в мережі Web3.0

Предметами дослідження визначено нейронні мережі для розпізнавання облич та блокчейн для роботи з розподіленими ідентифікаторами

Пояснювальна записка складається з чотирьох розділів. У першому розділі проводиться опис основних понять роботи, уточнення її актуальності, а також проводиться огляд об’єкта дослідження. Другий розділ складається з

трьох великих частин, в ньому по-перше більш детально розглянуто як працює мультифакторна автентифікація, її види та особливості. Після цього розглядається принципи та математичні методи розпізнавання облич та наприкінці багато уваги було зосереджено на децентралізованій автентифікації, що являє собою головну та найскладнішу частину роботи. Також коротко розглянуто останні стандарти децентралізованого ідентифікатора (DID), а також його цілі, переваги та недоліки. У третій частині розглянуто практичну частину роботи, а саме реалізацію авторизації та автентифікації за допомогою блокчейну та нейронних мереж. Практична частина ділиться на два великі частини роботи, а саме нейромережа для розпізнавання облич і реалізація розподіленої ідентифікації з використанням криптогаманців. Четвертий розділ присвячено формулюванню стартап-проекту.

# РОЗДІЛ 1. ОСНОВНІ ПОЛОЖЕННЯ

## 1.1 Актуальність роботи

Web3 пропонує величезний потенціал для того, щоб зробити Інтернет цінним, захоплюючим, відкритим і доступним для всіх, щоб кожен міг вчитися і розвиватися в ньому. Однак автентифікація в Web3 має важливе значення. Web3-додатки, швидше за все, не будуть працювати, якщо вони не матимуть можливості входу користувачів в блокчейн. Таким чином, для успішної реалізації web3-додатків або додатків нового покоління з використанням блокчейну, вони повинні мати можливість використовувати web3-автентифікацію.[6]

Кількість сфер застосування блокчейн-технологій у бізнесі продовжує зростати. Багато хто використовує приватні дозволені блокчейни, якими користуються члени галузі. На відміну від публічних блокчейнів, таких як Ethereum, Solana та інші, ці приватні блокчейни працюють лише за запрошеннями. Цей елемент створює рівень автентифікації та авторизації між користувачем та даними блокчейну.[15]

Без ідентичності Web3 людям було б складно підтверджувати свої повноваження, що виявилося б проблематичним у нинішній екосистемі Web3. Таким чином, ідентичності Web3 відіграють важливу роль на наступному етапі розвитку Інтернету. На жаль, концепція ідентичності Web3 для всіх користувачів була проігнорована в порівнянні з більш відомими ідеями, такими як DeFi, DAO, NFT тощо. Однак, гнучкий, спільний і стійкий рівень ідентичності має важливе значення для екосистеми децентралізованого Інтернету. Отже, що саме являє собою ідентичність Web3?

Ідентифікація на основі Web3 надає користувачам повний контроль для безпосереднього управління їхньою інформацією. Крім того, ідентичність Web3 - це, як правило, адреса в Інтернеті, яка використовується для пошуку інформації та даних, пов'язаних з конкретною особою. Ця інформація містить

відомості, що дозволяють використовувати такі випадки використання, як шифрування даних, зв'язок, механізм входу в систему тощо. Ці дані захищені криптографічними доказами, такими як цифрові підписи.

Підсумовуючи, можна сказати, що ідентифікація через Web3 - це, по суті, безпечна та захищена цифрова ідентичність для фізичних осіб, яка надає їм більше контролю над їхніми особистими даними та інформацією. Це не зовсім відрізняється від традиційних ідентичностей, які обговорюються в Web2 або навіть Web1. Однак, основна відмінність полягає в тому, що стосується права власності та інтероперабельності.

## 1.2 Основні поняття предметної області

### 1.2.1 Автентифікація

Автентифікація - це акт підтвердження того, що користувачі є тими, за кого вони себе видають. Це перший крок у будь-якому процесі забезпечення безпеки.

Метою автентифікації є перевірка того, що хтось або щось є тим, за кого вони себе видають. Існує багато форм автентифікації. Наприклад, у світі мистецтва існують процеси та інституції, які підтверджують, що картина або скульптура є роботою конкретного художника. Так само уряди використовують різні методи автентифікації, щоб захистити свою валюту від підробки. Як правило, автентифікація захищає предмети, що мають цінність, а в інформаційну епоху - системи та дані.

Приклад: Замок на дверях надає доступ лише тому, хто має правильний ключ, так само, як система надає доступ лише тим користувачам, які мають правильні облікові дані. [7]

### 1.2.2 Авторизація

Авторизація в системній безпеці - це процес надання користувачеві дозволу на доступ до певного ресурсу або функції. Цей термін часто використовується як взаємозамінний з контролем доступу або привілеями клієнта.

Надання комусь дозволу на завантаження певного файлу на сервері або надання окремим користувачам адміністративного доступу до програми є хорошими прикладами авторизації.

Приклад: Перебуваючи в будинку, особа має дозвіл на доступ до кухні та відкриття шафи, в якій зберігається корм для домашніх тварин. Особа може не мати дозволу на вхід до спальні для короткого сну. [7]

### 1.2.3 Біометрична автентифікація.

Біометрична автентифікація - це процес забезпечення безпеки, який спирається на унікальні біологічні характеристики осіб для підтвердження того, що вони є тими, за кого себе видають. Системи біометричної автентифікації порівнюють фізичні або поведінкові ознаки зі збереженими, підтвердженими, автентичними даними в базі даних. Якщо обидва зразки біометричних даних збігаються, автентифікація підтверджується. Зазвичай біометрична автентифікація використовується для управління доступом до фізичних і цифрових ресурсів, таких як будівлі, приміщення і обчислювальні пристрої.

Біометрична ідентифікація використовує біометричні дані, такі як відбитки пальців або сканування обличчя, для ідентифікації особи, тоді як біометрична автентифікація - це використання біометричних даних для підтвердження того, що люди є тими, за кого вони себе видають. [5]

### 1.2.4 Блокчейн

Блокчейн - це загальний, незмінний реєстр, який полегшує процес запису транзакцій та відстеження активів у бізнес-мережі. Актив може бути матеріальним (будинок, автомобіль, готівка, земля) або нематеріальним (інтелектуальна власність, патенти, авторські права, брендинг). Практично все, що має цінність, можна відстежувати і торгувати в мережі блокчейн, знижуючи ризик і скорочуючи витрати для всіх учасників. [9]

Ключові елементи блокчейну:

- **Технологія розподіленого реєстру**. Всі учасники мережі мають доступ до розподіленого реєстру та його незмінного запису транзакцій. За допомогою цього розподіленого реєстру транзакції реєструються лише один раз, усуваючи дублювання зусиль, характерне для традиційних бізнес-мереж.
- **Незмінні записи**. Жоден учасник не може змінити або підробити транзакцію після того, як вона була записана в загальний реєстр. Якщо запис транзакції містить помилку, необхідно додати нову транзакцію, щоб виправити помилку, і тоді обидві транзакції стають видимими.
- **Смарт-контракти**. Для прискорення транзакцій в блокчейні зберігається набір правил, який називається смарт-контрактом і виконується автоматично. Смарт-контракт може визначати умови

переказу корпоративних облігацій, включати умови оплати туристичної страховки та багато іншого.

Блокчейн зберігає дані у вигляді блоків, які потім пов'язуються між собою за допомогою криптографії.

Коли надходять нові дані, вони заносяться в новий блок. Після того, як блок заповнюється даними, він приєднується до попереднього блоку, що робить дані з'єднаними в хронологічному порядку.

У блокчейні можуть зберігатися різні типи інформації, але найпоширенішим його використанням до цього часу було використання в якості реєстру для транзакцій.

У випадку з біткоїном, блокчейн використовується децентралізовано, так що жодна особа або група осіб не має контролю - скоріше, всі користувачі колективно зберігають контроль.

Децентралізовані блокчейни є незмінними, що означає, що введені дані є незворотними. Для Біткоїна це означає, що транзакції постійно записуються і доступні для перегляду будь-кому. [10]

### 1.2.5 Ідентифікатор

Коли користувач (або інша особа) заявляє про свою особу, це називається ідентифікацією. Для ідентифікації може використовуватися ім'я користувача, ідентифікатор процесу, смарт-карта або будь-що інше, що може однозначно ідентифікувати предмет або особу. Системи безпеки використовують цей метод ідентифікації, щоб визначити, чи має особа дозвіл на доступ до об'єкта.

До поширених ідентифікаторів відносяться

– Ім'я

- Номер соціального страхування/податковий ідентифікаційний номер
- Номер мобільного телефону
- Дата та місце народження
- Цифрові ідентифікаційні дані, наприклад, адреси електронної пошти, імена користувачів, аватарки

Ідентифікація особи - це процес ототожнення конкретної особи з конкретною ідентифікацією. Вона вважається важливим процесом, оскільки вирішує певні питання щодо особи, такі як: "Чи є ця особа тією, за кого вона себе видає?", "Чи була ця особа тут раніше?", або "Чи слід надавати цій особі доступ до нашої системи?". [11]

Ідентифікація вигідна для організацій, оскільки вона:

- Може бути легко інтегрована в різні системи
- Недорога для обробки та розвитку
- Служить стримуючим фактором для самозванців

### 1.2.6 Криптовалюта та криптогаманці.

Криптовалюта - це цифрова або віртуальна валюта, яка захищена криптографією, що робить майже неможливим її підробку або подвійне використання. Багато криптовалют є децентралізованими мережами, заснованими на технології блокчейн - розподіленому реєстрі, який забезпечується розрізненою мережею комп'ютерів.

Визначальною особливістю криптовалют є те, що вони, як правило, не випускаються жодним центральним органом влади, що робить їх теоретично непідвладними державному втручанню або маніпуляціям.[16]

Криптовалютні гаманці зберігають публічні та приватні ключі користувачів, надаючи простий у використанні інтерфейс для управління криптовалютними балансами. Вони також підтримують перекази криптовалюти через блокчейн. Деякі гаманці навіть дозволяють користувачам виконувати певні дії зі своїми криптовалютними активами, такі як купівля та продаж або взаємодія з децентралізованими додатками (dapps).

Криптовалютний гаманець складається з трьох компонентів:

**Публічний ключ** - це адреса, на яку можна відправити та отримати транзакцію. Це як адреса електронної пошти.

**Приватний ключ** - цей ключ повинен зберігатися в таємниці. Цей ключ дозволяє користувачам отримувати доступ до грошей і може бути використаний для нових транзакцій. Він використовується як підпис користувача в транзакції.

**Початкова фраза** - це фраза, яка використовується для генерації багатьох приватних ключів. Вона служить в якості кореневого коду гаманця, що дає доступ до всіх ключів і адрес. Також може використовуватися для генерації нових приватних ключів.[6]

Гаманці Web3 можуть використовуватися для автентифікації транзакцій Web3. Одними з найбільш надійних рішень є MetaMask, WalletConnect, Web3Auth і Formatic. Кожен варіант пропонує відмінний користувальницький досвід. MetaMask, WalletConnect і Formatic ідеально підходять для користувачів криптовалют, в той час як Web3Auth і Formatic більш доступні для всіх користувачів. MetaMask є, мабуть, найкращим рішенням для онлайн-користувачів, в той час як WalletConnect є обов'язковим для мобільних користувачів. Ми будемо розглядати MetaMask як рішення для гаманця, тому необхідно знати огляд MetaMask, перш ніж описувати технічні кроки web3-автентифікації через нього.

MetaMask - це розширення для браузера, яке виконує функції додатку та криптогаманця. Це шлюз між блокчейном і розширеннями для браузерів. MetaMask можна завантажити як розширення для браузера і встановити.

MetaMask дозволяє користувачеві керувати приватним ключем, який контролює адресу Ethereum і полегшує транзакції з блокчейн-додатками. За замовчуванням MetaMask включає мережу Ethereum та найпопулярніші тестові мережі Ethereum. Це дозволяє легко додавати EVM-сумісні мережі. Вхід за допомогою MetaMask швидкий і зручний, тому він є кращим вибором для автентифікації в web3. Крім того, MetaMask значно спрощує створення dApps.

Автентифікація за допомогою гаманця посилює безпеку додатків, усуваючи ризиковані практики управління паролями і знижує накладні витрати на управління паролями в БД. Коли ми підключаємося до гаманця, ми отримуємо публічний ключ/публічну адресу/адресу гаманця, які ми можемо використовувати для відображення та управління даними користувача.

### 1.2.7 Мультифакторна автентифікація

Багатофакторна автентифікація (MFA) - це метод автентифікації, який вимагає від користувача надання двох або більше факторів перевірки для отримання доступу до ресурсу, такого як додаток, онлайн-акаунт або VPN. MFA є основним компонентом ефективної політики управління ідентифікацією та доступом (IAM). Замість того, щоб просто запитувати ім'я користувача та пароль, MFA вимагає один або більше додаткових факторів перевірки, що зменшує ймовірність успішної кібератаки.

MFA працює, вимагаючи додаткової верифікаційної інформації (факторів). Одним з найпоширеніших факторів MFA, з якими стикаються користувачі, є одноразові паролі (OTP). OTP - це 4-8-значні коди, які ви часто отримуєте електронною поштою, SMS або через мобільний додаток. За допомогою OTP новий код генерується періодично або кожного разу, коли

подається запит на автентифікацію. Код генерується на основі початкового значення, яке присвоюється користувачеві при першій реєстрації, та деяких інших факторів, які можуть бути просто лічильником, який збільшується, або значенням часу. [13]

### 1.2.8 DAPP

Децентралізований додаток - або dapp - це цифровий додаток, який можна знайти на будь-якому смартфоні або ноутбуці, з додатковою функцією використання технології блокчейн, щоб уберегти дані користувачів від рук організацій, які стоять за ним. Так само, як криптовалюта є децентралізованими грошима, додатки є децентралізованими додатками.

Блокчейн зберігає копії свого розширюваного стеку даних на великій кількості комп'ютерів-учасників, відомих як "вузли", одночасно. Ці комп'ютери належать користувачам, а не творцям додатку. Повне пояснення того, як працює технологія блокчейн, можна знайти тут.

Dapps настільки ж різноманітні, як і звичайні додатки: вони можуть надавати соціальні мережі, ігри, розваги, інструменти для підвищення продуктивності і так далі. Багато з них розроблені як інструменти, що допомагають споживачам отримати доступ до децентралізованих фінансових послуг, або DeFi. Ця остання функція настільки поширена, що в офіційному документі мережі Ethereum даппи поділяються на "фінансові", "напівфінансові" та "інші". [8]

### 1.2.9 DID (Decentralized Identifier)

Традиційні ідентифікатори, такі як ваше юридичне ім'я або адреса електронної пошти, покладаються на третіх осіб - уряди та провайдерів електронної пошти. Децентралізовані ідентифікатори (DID) відрізняються від них - вони не видаються, не управляються і не контролюються жодною центральною організацією.

Децентралізовані ідентифікатори видаються, зберігаються і контролюються фізичними особами. Акаунт Ethereum є прикладом децентралізованого ідентифікатора. Ви можете створювати скільки завгодно облікових записів без дозволу будь-кого і без необхідності зберігати їх в центральному реєстрі.

Децентралізовані ідентифікатори зберігаються в розподілених книгах (блокчейнах) або однорангових мережах. Це робить DID глобально унікальними, вирішуваними з високою доступністю та криптографічно перевіреними. Децентралізований ідентифікатор може бути пов'язаний з різними суб'єктами, включаючи людей, організації або державні установи. [12]

### 1.2.10Web3.0

Щоб зрозуміти що таке Web3.0 потрібно спочатку подивитися в історію інтернету та розібратися якими були Web1.0 та Web2.0.

Бернерс-Лі став піонером раннього розвитку Інтернету в 1990 році, коли він працював комп'ютерним науковцем в європейському дослідницькому центрі ЦЕРН. До жовтня 1990 року Бернерс-Лі написав три фундаментальні

технології, які стали основою Інтернету, включаючи найперший редактор/браузер веб-сторінок:

- HTML: HyperText Markup Language, мова розмітки або форматування Інтернету
- URI або URL: Уніфікований ідентифікатор ресурсу або локатор, унікальна адреса, що використовується для ідентифікації кожного ресурсу в Інтернеті
- HTTP: протокол передачі гіпертексту, який дозволяє отримувати пов'язані ресурси з усього Інтернету

Це і був Веб 1.0, це була епоха статичних веб-сторінок, що завантажувалися з серверів - далеко не той витончений контент, який сьогодні сприймається як належне.

Веб 2.0 (зараз ми тут) означає зміну парадигми використання Інтернету. За останні 15-20 років звичайні веб-сторінки Веб 1.0 були повністю замінені інтерактивністю, соціальними зв'язками та контентом, створеним користувачами, Веб 2.0. Веб 2.0 робить можливим перегляд користувацького контенту мільйонами людей по всьому світу практично в одну мить; це безпрецедентне охоплення призвело до вибуху цього типу контенту в останні роки.

Веб 3.0 представляє наступну ітерацію або фазу еволюції Інтернету і потенційно може бути настільки ж революційним і представляти таку ж велику зміну парадигми, як і Веб 2.0. Веб 3.0 побудований на ключових концепціях децентралізації, відкритості та більшої корисності для користувачів. Бернерс-Лі виклав деякі з цих ключових концепцій ще в 1990-х роках, як зазначено нижче:

**Децентралізація**: Це основний принцип Веб 3.0. У Веб 2.0 комп'ютери використовують HTTP у вигляді унікальних веб-адрес для пошуку інформації, яка зберігається у фіксованому місці, як правило, на одному сервері. У Веб 3.0, оскільки інформація буде знаходитися на основі її змісту, вона може зберігатися в декількох місцях одночасно і, отже, бути децентралізованою. Це

дозволило б розбити величезні бази даних, які зараз утримуються такими інтернет-гігантами, як Meta і Google, і надало б користувачам більший контроль над ними.

За допомогою Веб 3.0 дані, що генеруються розрізненими і дедалі потужнішими обчислювальними ресурсами, в тому числі мобільними телефонами, настільними комп'ютерами, приладами, транспортними засобами і датчиками, будуть продаватись користувачами через децентралізовані мережі передачі даних, що гарантуватиме збереження за користувачами контролю над правом власності.

**Без довіри і без дозволу**: На додаток до децентралізації та програмного забезпечення з відкритим вихідним кодом, Веб 3.0 також буде без довіри (тобто мережа дозволить учасникам взаємодіяти безпосередньо, не проходячи через довіреного посередника) та без дозволу (що означає, що будь-хто може брати участь без дозволу від керівного органу). Як наслідок, додатки Веб 3.0 будуть працювати на блокчейнах або децентралізованих однорангових мережах, або їх комбінації - такі децентралізовані додатки називаються dApps.

**Штучний інтелект (ШІ) і машинне навчання**: У Веб 3.0 комп'ютери зможуть розуміти інформацію так само, як і люди, за допомогою технологій, заснованих на концепціях Семантичного Інтернету і обробці природної мови. Веб 3.0 також використовуватиме машинне навчання, яке є галуззю штучного інтелекту (ШІ), що використовує дані та алгоритми для імітації того, як навчається людина, поступово підвищуючи його точність. Ці можливості дозволять комп'ютерам давати більш швидкі і релевантні результати в багатьох сферах, таких як розробка ліків і нових матеріалів, на відміну від простої таргетованої реклами, яка становить основну частину нинішніх зусиль.

**Зв'язність і повсюдність**: З Веб 3.0 інформація і контент стають більш пов'язаними і повсюдними, доступ до них здійснюється за допомогою різних додатків і все більшої кількості повсякденних пристроїв, підключених до Інтернету, одним з прикладів яких є Інтернет речей. [4]

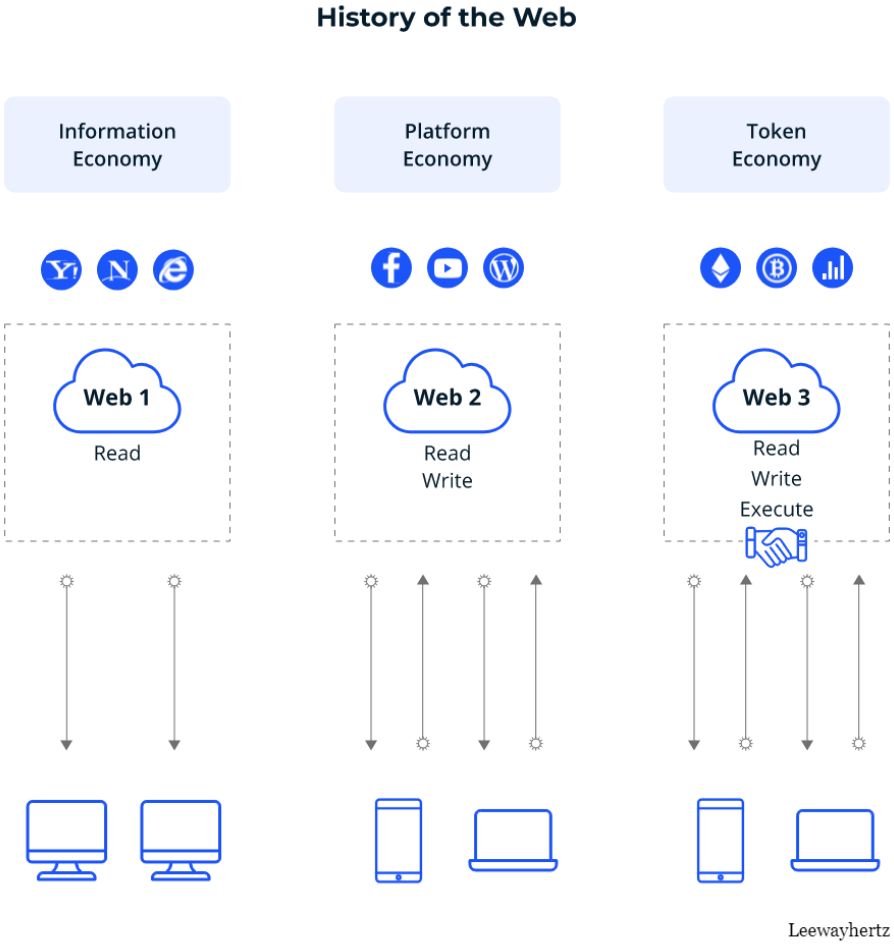


Рисунок 1.1. Різниця між поколіннями Веба [6]

## 1.3 Аналіз існуючих методів автентифікації та авторизації

Для ідентифікації особи може використовуватися документ, що посвідчує особу, наприклад, посвідчення особи (також відоме як IC, ID-картка, картка громадянина), або паспортна картка (якщо вона має невеликий формат, як звичайна кредитна картка). Деякі країни також видають офіційні документи, що посвідчують особу, такі як національні посвідчення особи, які можуть бути обов'язковими або необов'язковими, в той час як інші можуть

покладатися на регіональні посвідчення особи або неофіційні документи для підтвердження особи.

Існують і деякі інші прийнятні форми посвідчення особи:

**Щось, що особа знає (SYK):** пароль, PIN-код, дівоче прізвище матері або кодова комбінація замка. Ідентифікація особи за допомогою чогось, що вона вже знає, є, ймовірно, найпростішим варіантом, але одним з найменш безпечних.

**Щось, що є у особи (SYH):** ключ, свайп-карта, картка доступу або бейдж - все це приклади предметів, якими може володіти особа. Цей метод зазвичай використовується для отримання доступу до таких об'єктів, як банки та офіси, але він також може бути використаний для отримання доступу до конфіденційних місць або перевірки системних облікових даних. Це також простий варіант, але ці предмети легко вкрасти.

**Щось, чим є людина(SYA)**: біометричні дані особи є унікальними і не можуть бути втрачені або викрадені. Використання біометричних даних для ідентифікації особи є найбільш точним і безпечним варіантом.

Кіберзлочинці постійно вдосконалюють свої системні атаки. Як наслідок, команди безпеки мають справу з безліччю постійно мінливих проблем з автентифікацією. Ось чому компанії починають розгортати більш складні плани, які включають автентифікацію. Деякі з найпоширеніших методів автентифікації, що використовуються для захисту сучасних систем, включають в себе:

**Face Recognition**. Автентифікація за обличчям - це технологія, яка дозволяє людям отримувати доступ до онлайн-сервісів, фізичних параметрів та інших ресурсів, використовуючи зображення свого обличчя.

Автентифікація за обличчям, яка також називається розпізнаванням обличчя, спирається на вбудовану технологію зондування мобільних та інших пристроїв. На відміну від інших рішень ідентифікації, таких як паролі, верифікація за допомогою електронної пошти, селфі або зображень, або ідентифікація за відбитками пальців, біометричне розпізнавання обличчя

використовує унікальні математичні та динамічні шаблони, які працюють як сканер обличчя, що робить цю систему однією з найбезпечніших та найефективніших.

**Fingerprint Recognition**. Автентифікація за відбитками пальців - це акт перевірки особи на основі одного або декількох відбитків пальців. Ця концепція використовується протягом десятиліть у різних сферах, включаючи цифрову ідентифікацію, кримінальне правосуддя, фінансові послуги та охорону кордонів.

**FIDO 2. Fast Identity Online (FIDO)** - це відкритий стандарт для безпарольної автентифікації. FIDO дозволяє користувачам і організаціям використовувати цей стандарт для входу на свої ресурси без імені користувача або пароля за допомогою зовнішнього ключа безпеки або ключа платформи, вбудованого в пристрій.

Ці ключі безпеки FIDO2, як правило, є USB-пристроями, але можуть також використовувати Bluetooth або NFC. Завдяки апаратному пристрою, який обробляє автентифікацію, безпека облікового запису підвищується, оскільки немає пароля, який можна було б розкрити або вгадати.

**SMS OTP**. Це зручний метод, який не вимагає від користувачів встановлення будь-яких додатків. Замість цього, для автентифікації на зареєстрований телефон користувача надсилається одноразовий пароль за допомогою SMS, який використовується для автентифікації.

Загальні проблеми:

- Проблеми з користувацьким інтерфейсом - OTP часто мають обмеження за часом, а обмежений прийом мобільних операторів може спричинити проблеми для користувачів у віддалених районах.
- Вразливість до шкідливого програмного забезпечення, атак SS7 та підміни SIM-карт.

**OATH OTP (Soft Tokens).** Це програмне забезпечення, яке може бути вбудоване в мобільні додатки і використовує криптографічні операції для

автентифікації користувача та пристрою. Ці рішення зазвичай забезпечують більш плавний користувальницький інтерфейс; немає необхідності перемикатися між додатками або покладатися на апаратний пристрій. З точки зору безпеки, вони мають значні переваги, оскільки SDK з м'якими токенами підтримують просунуту криптографію, наприклад, цифрові підписи.

Загальні проблеми:

- Поганий користувальницький досвід (UX) - користувачі повинні постійно перемикатися між додатками для підтвердження особи/транзакції; користувач втрачає доступ з кожною зміною/втратою/оновленням свого смартфона; немає безпечних варіантів резервного копіювання

**Blockchain**. Блокчейн-автентифікація відноситься до системи, розробленої для підвищення безпеки користувачів та перевірки особи користувача і дозволяє користувачам підключатися до ресурсів, заснованих на технологіях цифрової валюти, транзакцій, криптовалют тощо.

Вона використовує технологію розподіленого реєстру блокчейн та методи автентифікації для підвищення конфіденційності та безпеки систем автентифікації. Вся мережа, заснована на блокчейні, здатна мати власну цілісність даних.

Персональна інформація, яка використовується для перевірки особи користувача, зберігається в хеші блоку, наприклад, ім'я користувача або пароль. Це допоможе досягти суверенної ідентичності.

**Passwords**. Паролі є найпоширенішим методом автентифікації. Паролі можуть бути у вигляді рядка букв, цифр або спеціальних символів. Для захисту необхідно створювати надійні паролі, які включають комбінацію всіх можливих варіантів.

Однак паролі схильні до фішингових атак, що послаблює їх ефективність. Середньостатистична людина має близько 25 різних облікових записів в Інтернеті, але тільки 54% користувачів використовують різні паролі для своїх облікових записів.

Справа в тому, що існує дуже багато паролів, які потрібно пам'ятати. Як наслідок, багато людей обирають зручність, а не безпеку. Більшість людей використовують прості паролі замість того, щоб створювати надійні паролі, тому що їх легше запам'ятати.

**Secret Questions**. Безпечні питання є поширеним методом автентифікації особи, з яким ви, ймовірно, стикалися раніше. Створюючи обліковий запис або реєструючись на послугу в Інтернеті, користувачі конфіденційно передають відповіді на секретні запитання провайдеру.

Як правило, ці питання безпеки та відповіді на них використовуються для самообслуговування відновлення пароля - введення правильної відповіді перевіряє користувача та дозволяє йому скинути пароль - хоча ви також можете використовувати питання безпеки як додатковий фактор автентифікації для входу в систему.

**Personal Info**. Персональна автентифікаційна інформація - ПІН-код або будь-який інший пароль чи інформація, яку користувачі створюють або приймають для використання з метою автентифікації своєї особи у додаток. Інші приклади Персональної автентифікаційної інформації можуть включати ім’я, дату народження, почтовий індекс, місце проживання, які можуть використовуватися або вимагатися для здійснення Інтернет-транзакцій або інших операцій.

**Email OTP**. Метод Email OTP дозволяє авторизуватися за допомогою одноразового пароля (OTP), який надсилається на зареєстровану адресу електронної пошти. При спробі авторизації на будь-якому сервісі сервер надсилає OTP на зареєстровану електронну адресу користувача.

Щоб використовувати функцію Email OTP, необхідно спочатку зареєструвати альтернативний ідентифікатор електронної пошти. Реєстрація альтернативного ідентифікатора електронної пошти необхідна для того, щоб OTP могло бути надіслано на цей ідентифікатор, оскільки ви не зможете отримати доступ до основного ідентифікатора електронної пошти, якщо

обліковий запис буде заблоковано або якщо ви забудете пароль до облікового запису.

## 1.4 Результати дослідження

Кожен метод автентифікації користувача, як правило, може бути оцінений за допомогою наступних трьох ключових параметрів:

**Зручність**: Наскільки природно і без проблем для кінцевого користувача використовувати цю автентифікацію?

**Безпека**: наскільки складно зловмиснику обдурити автентифікацію?

**Розгортання**: Наскільки легко розгортати для всіх користувачів на різних платформах, пристроях, географічних регіонах тощо? Важливим фактором є те, чи є сценарій використання B2C або B2E, тобто чи є кінцевий користувач споживачем або працівником.

Тож для дослідження візьмемо описані вище методи автентифікації і проаналізуємо їх за обраними факторами та наведемо табличний огляд деяких з найпопулярніших методів автентифікації з оцінками за кожним з трьох параметрів. Зверніть увагу, що ці бали є простими відносними показниками, які дають загальне уявлення про відносні сильні та слабкі сторони кожного методу.

Таблиця 1.1 Порівняння існуючих методів автентифікації

| **Методи автентифікації** | **Вид** | **Зручність** | **Безпека** | **Розгортання** | **Коментарі** |
|---|---|---|---|---|---|
| Face Recognition | SYA | 9 | 7 | 2 | Загальне розпізнавання облич - непослідовна безпека, сильно залежить від апаратного та програмного забезпечення пристрою.<br><br>Наприклад, безпека розпізнавання облич на Android значною мірою залежить від марки пристрою. |
| Fingerprint Recognition | SYA | 7 | 7 | 4 | Загалом безпечніший і більш розповсюджений, ніж розпізнавання обличчя, і з кожним роком стає все більш розповсюдженим. Практично всі смартфони мають досить пристойний сканер відбитків пальців. |
| FIDO 2 | SYH | 6 | 8 | 3 | Використовує біометричні автентифікатори і намагається зробити їх сумісними на різних пристроях, працюючи з усіма гравцями галузі, але має деякий час, щоб дозріти в плані підтримки. |
| SMS OTP | SYH | 3 | 5 | 8 | Не дуже безпечний, оскільки схильний до атак на підміну SIM-карт, але активно використовується як другий |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | фактор у сценаріях B2C, оскільки кожен користувач має смартфон. |
| OATH OTP (Soft Tokens) | SYH | 3 | 8 | 5 | Досить безпечний, але потребує програми-автентифікатора, і користувач повинен знати, як її налаштувати. Це може бути корисно, якщо кінцевий користувач технічно підкований, як правило, у сценаріях B2E. |
| Blockchain | SYH | 8 | 10 | 5 | Дуже безпечний спосіб автентифікації, і в той же час досить зручний оскільки не вимагає носіння спеціальних пристроїв. Досить мати доступ до гаманця з будь-якого пристрою будь то персональний комп'ютер або телефон. Також легко вбудовується в додатки оскільки спочатку був розроблений в середовищі web і має хорошу інтеграцію з будь-якими веб додатками. |
| Passwords | SYK | 4 | 3 | 6 | Найбільш розповсюджений через історію та можливість розгортання має добре відомі проблеми з безпекою. Всі фахівці з безпеки хочуть, щоб вона зникла якомога швидше. |
| Secret Questions | SYK | 5 | 2 | 7 | Можливо, це гірше, ніж паролі, особливо в епоху соціальних мереж, оскільки багато |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | інформації про користувачів є загальнодоступною. |
| Personal Info | SYK | 6 | 0 | 8 | Запитувати особисту інформацію, таку як дата народження, є жахливим методом в епоху соціальних мереж. Він навіть не повинен бути вказаний як метод автентифікації, але я бачив, що деякі сервіси використовують його. |
| Email OTP | SYK | 4 | 5 | 7 | Гідний метод для сценаріїв B2C, таких як SMS, і не залежить від іншого пристрою, але в значній мірі залежить від ефективності та безпеки електронної пошти користувача. |

Якщо зобразити ці оцінки на бульбашковій діаграмі, де розмір бульбашки відображає розгортання, безпеку по осі Y та зручність використання по осі X, то вона виглядатиме приблизно так:

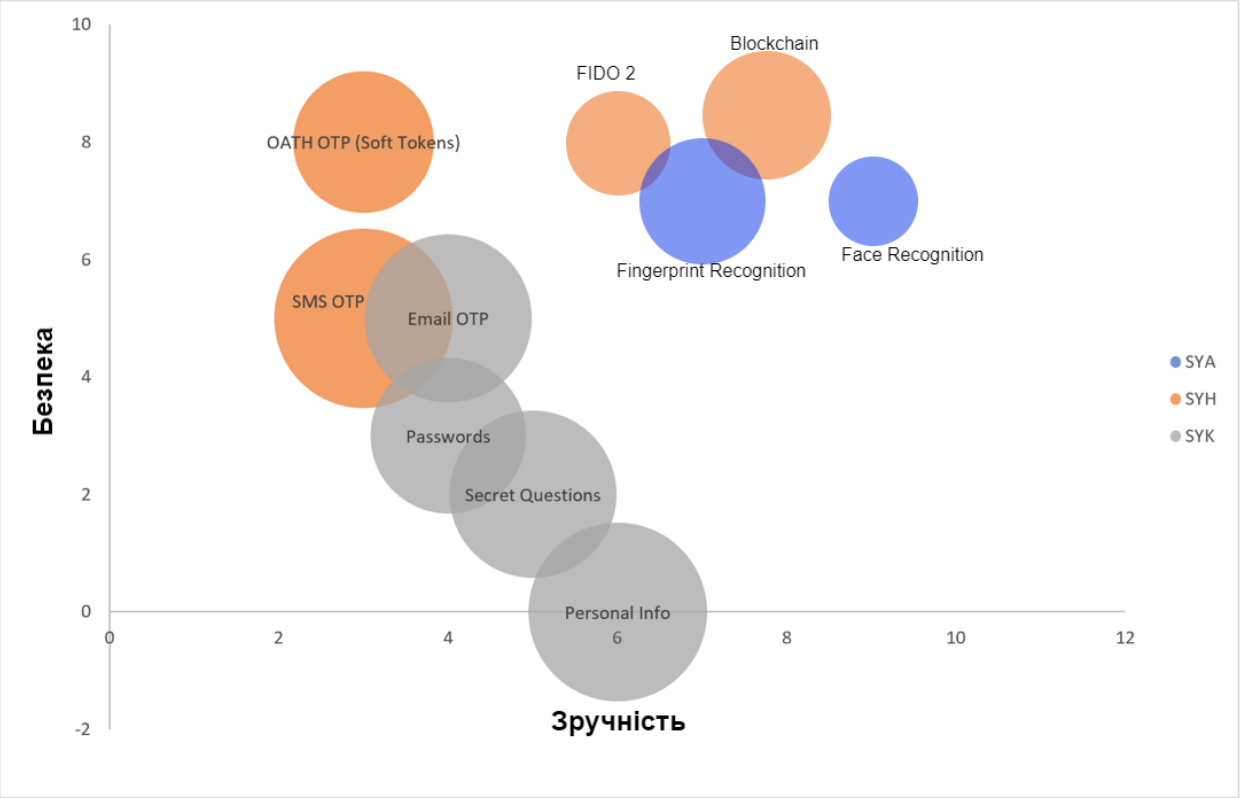


Рисунок 1.2 Порівняння методів автентифікації користувачів за 3 параметрами - безпека, зручність та розгортання. Розмір бульбашки вказує на можливість розгортання цього методу автентифікації

## 1.5 Висновки до порівняння

Дивлячись на отримані результати можна легко побачити, що в більшості ситуацій єдиний метод автентифікації користувача не може бути панацеєю. Саме для цього на допомогу приходить мультифакторна автентифікація, вона може допомогти убезпечити вхід користувача, використовуючи другий фактор, бажано іншої категорії. Як ми бачимо з результатів, фактор знання найслабший у безпеці та зручності, оскільки вимагають постійного введення даних і їх запам'ятовування.

Якщо ж говорити про фактор того, хто ти є, то найпростіше використовувати розпізнавання облич, оскільки воно вимагає найменше дій

від користувача і є більш унікальним, оскільки відбиток пальця простіше підробити і складніше розпізнати.

І нарешті останній фактор - це те що людина має. З них найкращі показники у блокчейна, оскільки він є найнадійнішим, оскільки блокчейн гарантує, що дані користувача не будуть змінені, оскільки вони вже записані в блокчейн, а також немає необхідності мати спеціальний пристрій, достатньо мати доступ до свого криптогаманця.

## 1.6 Висновки до розділу 1

У цьому розділі було пояснено всі основні поняття цієї роботи, починаючи від досить поширених понять, таких як авторизація та автентифікація, закінчуючи складними та специфічними такими як розподілений застосунок і розподілені ідентифікатори.

Також детально було розказано про історію вебу починаючи від старого інтернету, після обговорили чим відрізняється поточний інтернет і що буде далі, а також які основні чинники визначають новий інтернет (веб 3.0). Окрім цього, в роботі було описано актуальність роботи, ми зрозуміли, що автентифікація web 3.0 - дуже актуальна проблема й одного рішення для неї на поточний момент немає, тому робота є дуже актуальною.

Також розглянули варіації роботи автентифікації, зрозуміли, що існує безліч різних методів автентифікації та авторизації. Обговорили, що є три різні способи автентифікації: те, що людина знає, те, що у людини є, те, ким людина є. Крім цього розглянули варіації мультифакторної автентифікації та пройшлися базовими поняттями біометричної мультифакторної автентифікації.

# РОЗДІЛ 2. ТЕОРЕТИЧНІ ВІДОМОСТІ

## 2.1 Біометрична автентифікація

### 2.1.1 Загальна інформація

Думайте про біометрію з двох частин: "Біо" як "біологія". Біологія - це наукове вивчення життя і живих організмів. "Метрика" - це не просто інструмент, який використовується у світі (за винятком США) для вимірювання відстані між місцями; метрика - це заснована на правилах система вимірювання даних, яка часто використовується для порівняльних або відстежувальних цілей.

Біологія в значній мірі є якісною, а метрика - кількісною. Як дві речі, які здаються непоєднуваними, можуть об'єднатися для того, щоб забезпечити додаток для автентифікації, який створює безпеку і захист в цифровому світі, долаючи розрив між розривом з реальністю? Багато експертів сьогодні стверджують, що оскільки біометричні ідентифікатори унікальні для кожної людини, біометрична ідентифікація в кінцевому підсумку є більш безпечною, ніж традиційні паролі, двофакторна автентифікація і відповіді, засновані на знаннях.

Ми відповімо на деякі загальні питання про те, що таке біометрія, як базова система біометричного розпізнавання працює з ідентифікацією особи, обговоримо сучасні рішення біометричної ідентифікації та типи скринінгу. Ми також обговоримо переваги та недоліки біометрії. [17]

Види біометрії та їх показники: фізіологічні та поведінкові

Якщо ви коли-небудь розблоковували свій мобільний пристрій пальцем, сканували обличчя, щоб дізнатися, скільки грошей у додатку вашого банку, або кричали "Агов, Алекса", щоб дізнатися, скільки часу варити яйце - вітаємо! Ви використали свої біометричні дані. Біометричні дані (в тому числі і ті, що

були використані у вищезгаданому прикладі) відносяться до однієї з двох категорій: фізіологічні та поведінкові.

Відбиток пальця людини - найпоширеніший біометричний показник, який сьогодні використовується у світі для ідентифікації особи - відноситься до категорії "фізіологічних" біометричних показників - специфічний фізичний візерунок на тілі людини. Сканування обличчя однієї і тієї ж особи, або розпізнавання обличчя, також є фізіологічною біометричною ознакою, але може бути сегментоване для відображення інших фізіологічних біометричних датчиків, таких як форма вух, ширина очей на відстані один від одного, форма і довжина носа, тип волосся та інші. Фізіологічні біометричні дані аналізуються за допомогою таких речей, як розпізнавання обличчя і зчитування відбитків пальців - елементів, які є досить поширеними на мобільних пристроях, таких як смартфони, ноутбуки і планшети.

Голос людини є "поведінковим" біометричним показником - специфічні шаблони, які пов'язані з діями людини. Фізичний відбиток пальця може бути знятий з пристрою, але те, як ви використовуєте цей пристрій, може бути виміряно для створення профілю. Хоча існують певні перетини з фізичними ознаками, поведінкові біометричні показники все частіше використовуються в цифрових додатках та в Інтернеті для відстеження та визначення особи на основі набору шаблонів, створених на основі того, як вона себе поводить. Наприклад, більшість сучасних компаній, які мають цифрову платформу, розглядають поведінкові характеристики, такі як прокрутка веб-сторінки за допомогою миші, проведення пальцем по веб-сторінці, що вказує на мобільний перегляд, або кліки проти сильних натискань, як один з методів біометричного розпізнавання, який може допомогти побудувати профіль особистості людини.

Фізіологічні - форма тіла

1. Відбиток пальця - гребені на пальці
2. Геометрія руки - відстань пальців один від одного, довжина пальців і т.д.

3. Відбиток долоні - лінії руки на долоні та товщина/ширина долоні
4. ДНК - аналіз генетичної послідовності
5. Кров - група крові
6. Виміри обличчя - включаючи геометрію вух, носа, розмір і форму голови, відстань між очима, колір волосся і т.д.
7. Райдужка і сітківка - колір і форма очей
8. Вени - малюнок вен в області очей, рук,
9. Серцебиття та ЕКГ

Поведінкові - патерни, виявлені в поведінці людини:

1. Ритм набору тексту і динаміка натискання клавіш
2. Хода при ходьбі
3. Голос і мовленнєві інтонації
4. Жестикуляція
5. Веб-навігація - прокрутка та свайп
6. Розпізнавання письмового тексту, наприклад, підпису або шрифту
7. Геолокація та IP-адреси
8. Купівельні звички
9. Використання пристрою
10. Історія браузера та файли cookie

### 2.1.2 Переваги та недоліки

#### 2.1.2.1 Переваги

Часто біометрія відтворює чинні паролі — популярний метод розблокування смартфонів та доступу до мобільних додатків. Однак пароль не видаляється з процесу автентифікації, та біометричні дані використовуються

в якості ярлика. Використання біометрії для автентифікації працює тільки в тому випадку, якщо паролі повною мірою виключені з процесу автентифікації. Це фундаментальний крок до контролю доступу без пароля, оскільки якщо пароль все ще залишається у фоновому режимі, все ще є ризик злому даних.

Щоб отримати всі переваги автентифікації без пароля, біометрія повинна використовуватися для забезпечення безпеки, а не тільки для зручності. Це означає реалізацію політик, які повністю видаляють паролі з процесу автентифікації — користувачеві ніколи не доведеться створювати пароль, вводити комбінацію або скидати її. При реальному розпізнаванні особистості без пароля, єдиний спосіб автентифікації людини — це використання комбінації біометричних даних.

Попри розповіді про те, як зловмисники вводять в оману біометричні зчитувачі, відбити атаку за допомогою спуфінга дуже складно. Проникнути в систему безпеки за допомогою пароля просто — зловмисник може легко набрати вкрадені облікові дані на клавіатурі, але біометричні зображення можна безпосередньо ввести в зчитувач: по-перше, вони повинні бути перетворені в об'єкт. Тому, щоб досягти успіху у використанні вкрадених зображень відбитків пальців, злочинець повинен зробити зліпки пальців людини досить хорошими, щоб обдурити технологію.

Навіть в разі фальсифікації, системи біометричної автентифікації містять додаткові заходи безпеки для захисту від таких атак. Це охоплює технологію "liveness detection" (детектування живого користувача), яка вимагає від людини моргнути або поворушити пальцями, щоб довести, що вона виконує автентифікацію в режимі поточного часу. Також варто відзначити поведінкову біометрію, яка використовує штучний інтелект для обліку взаємодії людей зі своїми смартфонами та іншими пристроями в процесі автентифікації.

Звичайно, такі спуфінгові атаки не вважаються проблемою, якщо зловмисники не можуть отримати біометричні зображення в першу чергу. Це ставить під питання проблему правильного зберігання чутливих біометричних

зображень, що має починатися з шифрування. Далі зображення не повинні зберігатися в єдиному місці, наприклад як на сервері, так і на смартфоні — це означає, що якщо зловмисники проникають в одну базу даних, вони отримають тільки частину біометричного зображення, що зробить його непридатним для використання.

З огляду на ймовірність успіху, зловмисники навряд чи припинять використовувати паролі в якості векторів проникнення. Захист компаній від атак, пов'язаних з вкраденими паролями, вимагає нового підходу до автентифікації, який замінює методи засновані на знаннях, біометрією. Це означає, що дана технологія розглядається як інструмент безпеки, а не як механізм прискорення автентифікації. Коли біометрія дозволить назавжди розпрощатися з паролями, ми станемо на крок ближче до надійно захищених організацій. [3]

#### 2.1.2.2 Недоліки біометричної автентифікації

Незважаючи на підвищену безпеку, ефективність і зручність, біометрична автентифікація та її використання в сучасних технологіях і цифрових додатках також має недоліки:

- Витрати - значні інвестиції в біометрію для забезпечення безпеки
- Порушення даних - біометричні бази даних все ще можуть бути зламані
- Відстеження і дані - біометричні пристрої, такі як системи розпізнавання облич, можуть обмежувати конфіденційність користувачів

- Упередженість - Машинне навчання і алгоритми повинні бути дуже досконалими, щоб мінімізувати біометричну демографічну упередженість
- Помилкові спрацьовування і неточність - помилкові відхилення і помилкові прийняття все ще можуть мати місце, перешкоджаючи доступу обраних користувачів до систем

**Витрати.** Не дивно, що більш досконала система безпеки вимагатиме значних інвестицій і витрат на її впровадження. В опитуванні 2018 року, проведеному компанією Spiceworks, 67 відсотків IT-фахівців назвали вартість "найбільшою причиною відмови від біометричної автентифікації". Перехід на біометричну автентифікацію буде не єдиним, за що компаніям доведеться заплатити: 47% опитаних заявили про необхідність модернізації поточних систем для підтримки переходу на біометричну автентифікацію на своїх пристроях.

**Витоки даних.** Компанії та уряди, які збирають та зберігають персональні дані користувачів, перебувають під постійною загрозою з боку хакерів. Оскільки біометричні дані незамінні, організаціям необхідно ставитися до конфіденційних біометричних даних з підвищеною безпекою і обережністю - це дорого і технічно складно, щоб випереджати розвиток шахрайства. Якщо пароль або пін-код скомпрометовані, завжди є можливість їх змінити. Цього не можна сказати про фізіологічні або поведінкові біометричні дані людини.

**Відстеження та дані.** Оскільки світ збільшує використання біометричних систем автентифікації, таких як технологія розпізнавання обличчя та інші заходи біометричної безпеки, необхідно брати до уваги конфіденційність користувачів. Коли біометричні дані перетворюються в дані і зберігаються, особливо в місцях або країнах, де застосовуються великі заходи спостереження, користувач ризикує залишити постійний цифровий запис, який потенційно може бути відстежений недобросовісними суб'єктами. У багатьох випадках організації та уряди використовували програмне

забезпечення для розпізнавання осіб для відстеження та ідентифікації людей з лякаючою точністю, що значно порушує недоторканність приватного життя. Зі збільшенням спостереження біометричні дані можуть стати постійною цифровою міткою, яка може бути використана для відстеження людини, як з її відома, так і без неї.

**Упередженість.** Мінімізація демографічної упередженості в біометрії при перевірці особи заявників під час цифрового вступу на роботу є складним завданням для постачальників послуг. Погане впровадження технології або навмисне зловживання може призвести до дискримінації та виключення. Без перевіреного, орієнтованого на документи рішення для підтвердження особи, крос-демографічні показники можуть бути ненадійними та обмежувати доступ клієнтів до таких необхідних речей, як кредити та розширюваний спектр цифрових послуг.

**Хибні спрацьовування і неточність.** Більшість поширених методів біометричної автентифікації покладаються на часткову інформацію для підтвердження особи користувача. Наприклад, мобільний біометричний пристрій сканує весь відбиток пальця на етапі реєстрації і перетворює його в дані. Однак у майбутньому біометрична автентифікація за відбитками пальців буде використовувати лише частини відбитків для підтвердження особи, щоб бути швидшою і швидшою. У 2018 році дослідницька група з Нью-Йоркського університету створила платформу штучного інтелекту, яка змогла шахрайським шляхом зламати автентифікацію за відбитками пальців з успішністю 20% шляхом зіставлення схожості часткових відбитків з повними біометричними даними. [17]

### 2.1.3 Математичне обґрунтування /методи

Найпоширенішим типом алгоритму машинного навчання, що використовується для розпізнавання облич, є згорткова нейронна мережа з глибоким навчанням (Convolutional Neural Network, CNN). CNN - це тип штучних нейронних мереж, які добре підходять для задач класифікації зображень.

CNN навчаються витягувати ознаки із зображень і використовувати ці ознаки для класифікації зображень за різними категоріями. Глибина ШНМ важлива для розпізнавання обличчя, оскільки вона дозволяє ШНМ вивчати більш складні риси обличчя.

Наприклад, неглибока CNN може навчитися розпізнавати лише прості риси обличчя, такі як форма носа або положення очей. З іншого боку, глибокий CNN може навчитися ідентифікувати більш складні риси обличчя, такі як текстура шкіри або форма підборіддя. Після того, як ШНМ буде навчена на наборі даних зображень облич, вона може бути використана для ідентифікації облич на нових зображеннях. Цей процес називається розпізнаванням облич.

#### 2.1.3.1 Опис елементів CNN

**Convolutional layer.** Згортковий шар є основним будівельним блоком згорткової мережі, який виконує більшу частину обчислювальної роботи. Основною метою шару згортки є виділення особливостей з вхідних даних, якими є зображення. Згортка зберігає просторовий зв'язок між пікселями, вивчаючи особливості зображення за допомогою невеликих квадратів

вхідного зображення. Вхідне зображення згортається за допомогою набору нейронів, що навчаються. Це призводить до створення карти ознак або карти активації на вихідному зображенні, після чого карти ознак подаються як вхідні дані для наступного шару згортки.

**Pooling Layer.** Шар об'єднання зменшує розмірність кожної карти активації, але зберігає найбільш важливу інформацію. Вхідні зображення розбиваються на набір прямокутників, що не перекриваються. Кожна область зменшується за допомогою нелінійної операції, такої як середнє або максимальне значення. Цей шар досягає кращого узагальнення, швидшої збіжності, стійкий до перекладу та спотворень і зазвичай розміщується між згортковими шарами.

**Relu Layer.** ReLU є нелінійною операцією і включає в себе блоки, що використовують випрямляч. Це поелементна операція, що означає, що вона застосовується до кожного пікселя і відновлює всі від'ємні значення на карті ознак до нуля. Для того, щоб зрозуміти, як працює ReLU, ми припускаємо, що є вхід нейрона, заданий як x, і з цього випрямляч визначається як f(x)=max(0,x) в літературі для нейронних мереж.

**Fully Connected Layer**. Термін "повністю з'єднаний шар" (ПЗШ) означає, що кожен фільтр у попередньому шарі з'єднаний з кожним фільтром у наступному шарі. Вихідні дані шарів згортки, об'єднання та ReLU є втіленням високорівневих характеристик вхідного зображення. Метою застосування FCL є використання цих ознак для класифікації вхідного зображення на різні класи на основі навчального набору даних. FCL розглядається як кінцевий об'єднуючий шар, що подає ознаки на класифікатор, який використовує функцію активації Softmax. Сума вихідних ймовірностей повністю пов'язаного шару дорівнює 1, що забезпечується використанням функції активації Softmax. Функція Softmax бере вектор довільних дійсних оцінок і стискає його до вектора значень від нуля до одиниці, які в сумі дорівнюють одиниці.[20]

### 2.1.4 Приклади реалізації та використання

Розпізнавання облич поділяється на три етапи:

**Вирівнювання та виявлення облич** - Перший крок полягає у виявленні облич на вхідному зображенні. Це можна зробити за допомогою каскадного класифікатора Хаара, який є різновидом алгоритму машинного навчання, що тренується на позитивних і негативних зображеннях. Машина повинна знайти обличчя на зображенні або відео. На сьогоднішній день більшість камер мають вбудовану функцію розпізнавання облич. Розпізнавання облич також використовується Snapchat, Facebook та іншими соціальними мережами, щоб дозволити користувачам додавати ефекти до фотографій і відео, які вони знімають за допомогою своїх додатків.

Проблема в контексті розпізнавання облич полягає в тому, що часто обличчя не спрямоване фронтально до камери. Обличчя, які відвернуті від фокусної точки, виглядають для комп'ютера зовсім інакше. Потрібен алгоритм для нормалізації обличчя, щоб воно відповідало обличчям в базі даних. Одним із способів досягти цього є використання декількох загальних орієнтирів обличчя. Наприклад, нижня частина підборіддя, верхня частина носа, зовнішні частини очей, різні точки навколо очей і рота тощо. Алгоритм машинного навчання потрібно навчити знаходити ці точки на будь-якому обличчі і повертати обличчя до центру.

**Вимірювання та вилучення ознак** - Після того, як обличчя вирівняні та виявлені, наступним кроком є вилучення з них ознак. Тут з'являється згорткова нейронна мережа (CNN). CNN здатна витягувати високорівневі ознаки із зображення, які потім використовуються для ідентифікації облич у базі даних.

**Розпізнавання облич** - останнім кроком є зіставлення витягнутих ознак з обличчями в базі даних. Зазвичай це робиться за допомогою метрики евклідової відстані, яка вимірює схожість між двома векторами.[19]

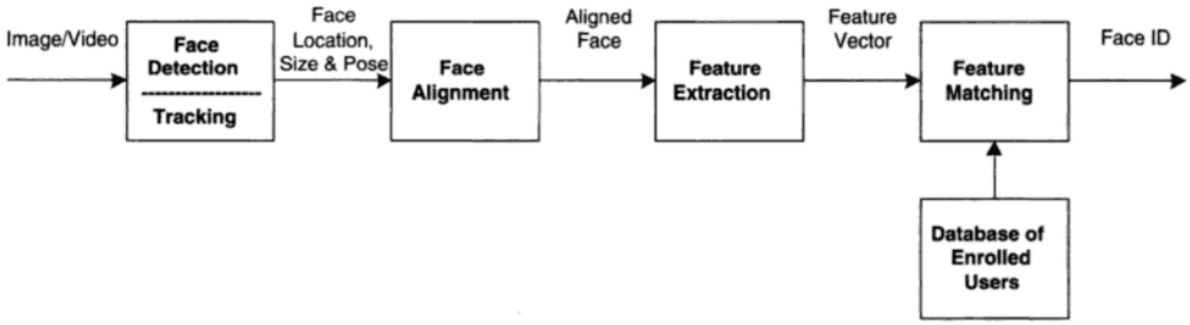


Рисунок 2.1 Процесс розпізнання обличчя [19]

## 2.2 Децентралізована автентифікація

### 2.2.1 Загальна інформація

Децентралізовані ідентифікатори (DID) - це новий тип ідентифікаторів, який дозволяє здійснювати децентралізовану цифрову ідентифікацію, що піддається перевірці. DID відноситься до будь-якого суб'єкта (наприклад, особи, організації, речі, моделі даних, абстрактної сутності тощо), як визначено контролером DID. На відміну від типових федеративних ідентифікаторів, DID були розроблені таким чином, щоб їх можна було відокремити від централізованих реєстрів, постачальників ідентифікаційних даних та центрів сертифікації. Зокрема, хоча інші сторони можуть бути використані для того, щоб допомогти виявити інформацію, пов'язану з DID, дизайн дозволяє контролеру DID довести контроль над ним, не вимагаючи дозволу від будь-якої іншої сторони. DID - це URI, які пов'язують суб'єкт DID

з документом DID, що дозволяє здійснювати довірчі взаємодії, пов'язані з цим суб'єктом.


Рисунок 2.2 Приклад децентралізованого ідентифікатора [1]

Кожен DID-документ може виражати криптографічний матеріал, методи перевірки або послуги, які забезпечують набір механізмів, що дозволяють контролеру DID довести контроль над DID. Послуги забезпечують довірчу взаємодію, пов'язану з суб'єктом DID. DID може надавати засоби для повернення самого об'єкта DID, якщо об'єкт DID є інформаційним ресурсом, таким як модель даних.

Цей документ визначає синтаксис DID, загальну модель даних, основні властивості, серіалізовані представлення, операції DID та пояснення процесу перетворення DID у ресурси, які вони представляють. [1]

EXAMPLE 1: A simple DID document

```
{
  "@context": [
    "https://www.w3.org/ns/did/v1",
    "https://w3id.org/security/suites/ed25519-2020/v1"
  ]
  "id": "did:example:123456789abcdefghi",
  "authentication": [{
    // used to authenticate as did:...fghi
    "id": "did:example:123456789abcdefghi#keys-1",
    "type": "Ed25519VerificationKey2020",
    "controller": "did:example:123456789abcdefghi",
    "publicKeyMultibase": "zH3C2AVvLMv6gmMNam3uVAjZpfkcJCwDwnZn6z3wXmqPV"
  }]
}
```

Рисунок 2.3 Приклад DID-документа [1]

Таке децентралізоване управління ідентифікацією включає підхід до управління ідентифікацією та доступом, який дозволяє людям генерувати, управляти та контролювати свою персональну інформацію (PII) без централізованої третьої сторони, такої як реєстр, постачальник ідентифікаційних даних або орган сертифікації.

Персональні дані, які вважаються приватними та конфіденційними даними, відносяться до сукупності інформації про конкретних осіб, яка прямо чи опосередковано їх ідентифікує. Зазвичай вона поєднує ім'я, вік, адресу, біометричні дані, громадянство, місце роботи, рахунки кредитних карток, кредитну історію тощо. Крім PII, інформація, яка формує децентралізовану цифрову ідентичність, включає в себе дані з електронних пристроїв в Інтернеті, такі як імена користувачів і паролі, історія пошуку, історія покупок та інші.

За допомогою децентралізованої ідентичності користувачі можуть контролювати власну PII і надавати тільки ту інформацію, яка необхідна для перевірки. Децентралізоване управління ідентифікацією підтримує структуру довіри до ідентичності, де користувачі, організації та речі взаємодіють один з одним прозоро та безпечно. [18]

#### 2.2.1.1 Цілі ДІД

- Простота створення – створення, можливо, тисяч DID має бути швидким і "дешевим"
- Децентралізованість – не залежать від централізованих реєстрів, постачальників ідентифікаційних даних, органів влади тощо
- Постійність –одного разу створений, він назавжди закріплюється за суб'єктом
- Можливість відновлення –можна з'ясувати базовий набір інформації про суб'єкта
- Криптографічно перевіряється – існує механізм криптографічного підтвердження ідентичності та права власності

### 2.2.2 Переваги та недоліки

Чотири основні переваги децентралізованого управління ідентифікацією включають контроль, безпеку, конфіденційність і простоту використання. Однак основними проблемами є низький рівень впровадження, відсутність регулювання та інтероперабельності.

Контроль дає власникам ідентичностей і цифрових пристроїв владу над своїми цифровими ідентифікаторами. Оскільки користувачі мають повний контроль і право власності на свої ідентифікаційні дані та облікові дані, вони можуть вирішувати, яку інформацію вони хочуть розкрити, і можуть довести свої претензії, не залежачи від будь-якої іншої сторони.

Безпека зменшує поверхню атаки шляхом зберігання PII. Блокчейн - це зашифрована децентралізована система зберігання, яка є безпечною, гнучкою і непроникною за своєю конструкцією, що знижує ризик отримання зловмисником несанкціонованого доступу для крадіжки або монетизації даних користувача.

Децентралізоване управління ідентифікацією також допомагає організаціям знизити ризики безпеки. Виходячи з того, як організації збирають, обробляють і зберігають дані користувачів, вони підпадають під дію нормативних актів. Організації стикаються з санкціями і штрафами навіть за ненавмисне порушення правил або витік даних. Завдяки децентралізованому управлінню ідентифікаційними даними вони мають можливість збирати та зберігати менше ідентифікаційних даних, спрощуючи свої обов'язки щодо дотримання вимог законодавства та зменшуючи ризики кібератак та неправомірного використання інформації.

Конфіденційність дозволяє організаціям використовувати принцип найменших привілеїв (PoLP) для визначення мінімального або вибіркового доступу до ідентифікаційних даних. PoLP - це термін, пов'язаний з інформаційною безпекою. Він стверджує, що будь-яка людина, гаджет або процес повинні мати лише мінімальні права, необхідні для виконання розглянутого завдання.

І останнє, але не менш важливе: технологія децентралізованої ідентифікації дає користувачам перевагу легкого створення та управління своїми ідентифікаційними даними за допомогою зручних нетермінальних децентралізованих ідентифікаційних додатків та платформ.

Щодо недоліків, то їх є чимало, в першу чергу - адаптація. Уряди та організації все ще намагаються з'ясувати, як розгорнути технологію децентралізованої ідентичності в масштабах, в той час як більшість нетехнологічних користувачів навіть не чули про це явище.

Подолання застарілих систем і правил, а також створення сумісних глобальних стандартів і управління також є важливими питаннями. Хоча

другорядною проблемою залишається вразливість ідентифікаційних даних, яка стосується дублювання, плутанини та неточності в управлінні ідентифікаційними даними. [18]

### 2.2.3 Математичне обґрунтування /методи

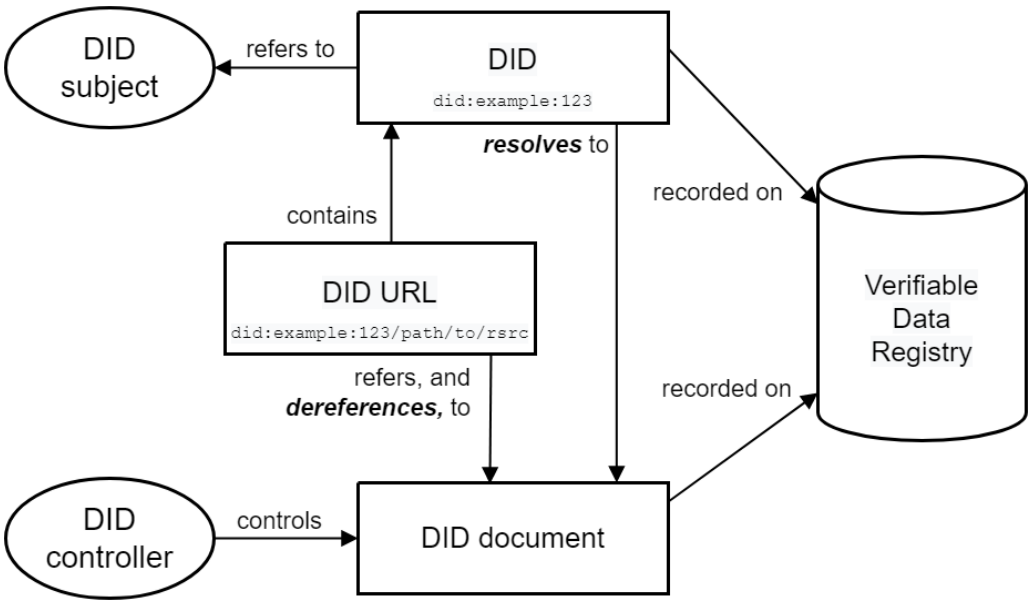


Рисунок 2.4. Огляд архітектури DID та взаємозв'язок основних компонентів [2]

**DIDs and DID URLs**. Децентралізований ідентифікатор, або DID, - це URI, що складається з трьох частин: схеми did:, ідентифікатора методу та унікального, специфічного для методу ідентифікатора, визначеного методом DID. DID можуть бути розпізнані в DID-документи. URL-адреса DID розширює синтаксис базового DID, щоб включити інші стандартні компоненти URI, такі як шлях, запит і фрагмент, для того, щоб знайти певний

ресурс - наприклад, криптографічний відкритий ключ всередині документа DID або ресурс, зовнішній для документа DID.

**DID subjects.** Суб'єкт DID - це, за визначенням, об'єкт, ідентифікований DID. Суб'єктом DID може бути також контролер DID. Суб'єктом DID може бути що завгодно: особа, група, організація, річ або концепція.

**DID controllers**. Контролер DID - це суб'єкт (особа, організація або автономне програмне забезпечення), який має можливість - як визначено методом DID - вносити зміни до документа DID. Ця можливість, як правило, забезпечується контролем набору криптографічних ключів, що використовуються програмним забезпеченням, яке діє від імені контролера, хоча вона також може бути забезпечена за допомогою інших механізмів. Слід зазначити, що DID може мати більше одного контролера, а суб'єктом DID може бути контролер DID або один з них.

**Verifiable data registries**. Для того, щоб можна було перетворити на документи DID, DID, як правило, реєструються в базовій системі або мережі певного типу. Незалежно від конкретної технології, що використовується, будь-яка така система, яка підтримує запис DID та повернення даних, необхідних для створення DID-документів, називається реєстром даних, що перевіряється. Приклади включають розподілені реєстри, децентралізовані файлові системи, бази даних будь-якого типу, однорангові мережі та інші форми довіреного зберігання даних.

**DID documents**. Документи DID містять інформацію, пов'язану з DID. Вони, як правило, виражають методи перевірки, такі як криптографічні відкриті ключі, та послуги, що мають відношення до взаємодії з суб'єктом DID. Загальні властивості, що підтримуються в документі DID, визначені в розділі 5. Основні властивості. DID-документ може бути серіалізований у байтовий потік (див. 6. Представлення).

**DID methods**. Методи DID - це механізм, за допомогою якого певний тип DID та пов'язаний з ним DID-документ створюються, вирішуються, оновлюються та деактивуються.

**DID resolvers and DID resolution**. Розв'язувач DID - це компонент системи, який приймає DID на вході та створює відповідний DID-документ на виході. Цей процес називається перетворенням DID. Етапи обробки конкретного типу DID визначаються відповідною специфікацією методу DID.

**DID URL dereferencers and DID URL dereferencing.** Розв'язувач DID URL-адреси - це компонент системи, який приймає DID URL-адресу на вході і виробляє ресурс на виході. Цей процес називається розыменуванням DID URL-адреси. [1]

### 2.2.4 Приклади реалізації та використання

Основою децентралізованого управління ідентифікацією є використання децентралізованих зашифрованих гаманців на базі блокчейну.

У системі децентралізованої ідентифікації користувачі використовують децентралізовані гаманці - спеціальні додатки, які дозволяють їм створювати свої децентралізовані ідентифікатори, зберігати свої PII та керувати своїми VC - замість того, щоб зберігати ідентифікаційну інформацію на численних веб-сайтах, контрольованих посередниками.

Окрім розподіленої архітектури, ці децентралізовані ідентифікаційні гаманці зашифровані. Паролі для доступу до них замінюються нефішинговими криптографічними ключами, які не представляють жодної вразливості у випадку злому. Децентралізований гаманець генерує пару криптографічних ключів: публічний і приватний. Публічний ключ відрізняє конкретний гаманець, тоді як приватний, який зберігається в гаманці, необхідний під час процесу автентифікації.

Децентралізовані ідентифікаційні гаманці прозоро автентифікують користувачів, але водночас захищають їхні комунікації та дані.

Децентралізовані додатки (DApps) зберігають PII, підтверджені ідентифікаційні дані та інформацію, необхідну для встановлення довіри, підтвердження відповідності або просто завершення транзакції. Ці гаманці допомагають користувачам надавати та відкликати доступ до ідентифікаційної інформації з одного джерела, що робить це швидше та простіше.

Крім того, ця інформація в гаманці підписується кількома довіреними сторонами, що підтверджує її достовірність. Наприклад, цифрові ідентифікатори можуть отримати схвалення від таких емітентів, як університети, роботодавці або державні структури. Використовуючи децентралізований ідентифікаційний гаманець, користувачі можуть пред'являти докази своєї особи будь-якій третій стороні.[18]

Гаманці Web3 можуть використовуватися для автентифікації транзакцій Web3. Одними з найбільш надійних рішень є MetaMask, WalletConnect, Web3Auth і Formatic. Кожен варіант пропонує відмінний користувальницький досвід. MetaMask, WalletConnect і Formatic ідеально підходять для користувачів криптовалют, в той час як Web3Auth і Formatic більш доступні для всіх користувачів. MetaMask є, мабуть, найкращим рішенням для онлайн-користувачів, в той час як WalletConnect є обов'язковим для мобільних користувачів. Ми будемо розглядати MetaMask як рішення для гаманця, тому необхідно знати огляд MetaMask, перш ніж описувати технічні кроки web3-автентифікації через нього.

MetaMask - це розширення для браузера, яке виконує функції додатку та криптогаманця. Це шлюз між блокчейном і розширеннями для браузерів. MetaMask можна завантажити як розширення для браузера і встановити. MetaMask дозволяє користувачеві керувати приватним ключем, який контролює адресу Ethereum і полегшує транзакції з блокчейн-додатками. За замовчуванням MetaMask включає мережу Ethereum та найпопулярніші тестові мережі Ethereum. Це дозволяє легко додавати EVM-сумісні мережі. Вхід за допомогою MetaMask швидкий і зручний, тому він є кращим вибором

для автентифікації в web3. Крім того, MetaMask значно спрощує створення dApps.

Автентифікація за допомогою гаманця посилює безпеку додатків, усуваючи ризиковані практики управління паролями і знижує накладні витрати на управління паролями в БД. Коли ми підключаємося до гаманця, ми отримуємо відкритий ключ/публічну адресу/адресу гаманця, які ми можемо використовувати для відображення та управління даними користувача. [6]

Наступна діаграма послідовності може допомогти краще зрозуміти потік автентифікації

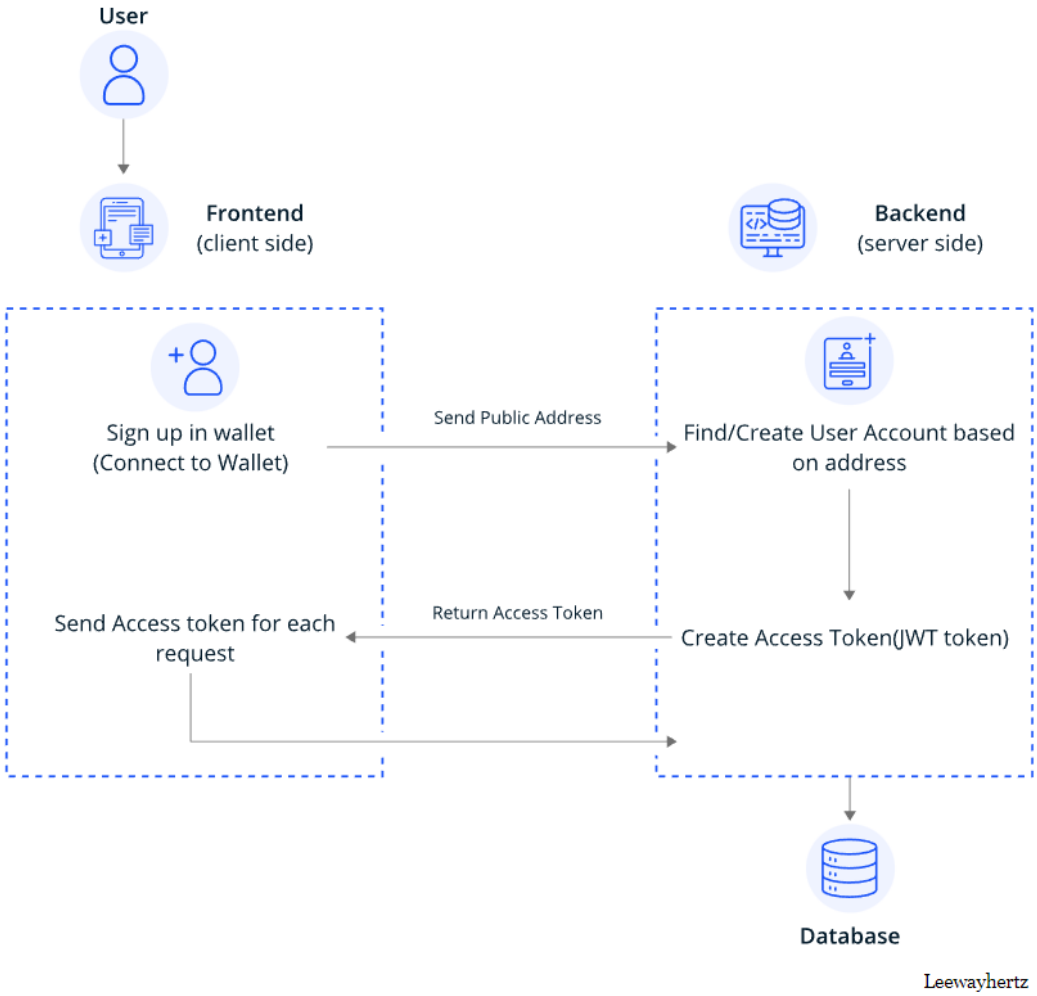


Рисунок 2.5 Процес автентифікації за допомогою криптогаманця [15]

## 2.3 Висновки до розділу 2

Web3-автентифікація важлива при створенні децентралізованих додатків. Хоча концепція автентифікації web3 може здатися складною, за допомогою авторитетних гаманців web3 ми можемо здійснити автентифікацію без особливих труднощів. Багато криптовалютних гаманців зараз доступні у вигляді розширень для браузерів або мобільних додатків. Вони пропонують зручний користувальницький інтерфейс і бувають різних форматів. Вони можуть використовуватися для входу в систему web3, а їх основне призначення - зберігання та управління криптовалютами. Гаманці Web3 дозволяють зберігати, відправляти і отримувати взаємозамінні і не взаємозамінні токени. Ці гаманці дозволяють користувачам отримати доступ до DeFi платформ і NFT ринку. Таким чином, гаманці web3 - це must-have для розробників блокчейну. За допомогою web3-автентифікації користувачі отримують можливість взаємодіяти з іншими автентифікованими користувачами та функціями мережі.

# РОЗДІЛ 3. ПОБУДОВА СИСТЕМИ ДЛЯ ДЕМОНСТРАЦІЇ РОБОТИ МЕТОДІВ МУЛЬТИФАКТОРНОЇ АВТЕНТИФІКАЦІЇ НА ПРИКЛАДІ РОЗПІЗНАВАННЯ ОБЛИЧ ТА РОЗПОДІЛЕНИХ АВТЕНТИФІКАТОРІВ

## 3.1 Архітектура системи та вибір мови програмування і фреймворків

Перед початком побудови архітектури слід врахувати наступні системні вимоги:

- Необхідно, щоб система була доступна з будь-якого пристрою.
- Система не повинна вимагати завантаження будь-яких додаткових модулів або бібліотек для того, щоб функціонувати.
- Різні компоненти не повинні залежати один від одного, щоб їх можна було створювати одночасно.
- Система повинна мати простий користувацький інтерфейс, оскільки основною аудиторією є особи, які не володіють комп'ютерною технікою.
- Інтерфейс програмування системи повинен бути зрозумілим, щоб її можна було легко інтегрувати в іншу систему в майбутньому.
- Кожна з потреб є критично важливою, і ми не можемо дозволити собі нехтувати ними. Отже, для того, щоб система відповідала всім вимогам, було створено веб-додаток.

Веб-додаток - це клієнт-серверний додаток, в якому клієнт використовує браузер для взаємодії з веб-сервером. Логіка роботи веб-додатку розподілена між сервером і клієнтом, дані зберігаються переважно на сервері, а обмін інформацією здійснюється через мережу. Однією з переваг цієї технології є те, що клієнти не залежать від операційної системи користувача, тому веб-додатки є крос-платформними сервісами.

Веб-додаток приймає запит від клієнта і виконує обчислення перед тим, як згенерувати веб-сторінку і відправити її клієнту за протоколом HTTP по мережі.[21]

Особливості цієї моделі полягають у тому, що користувач подає певний запит на сервер, який систематично обробляє його і повертає кінцевий результат клієнту. Однією з ключових переваг сервера є можливість обслуговування декількох клієнтів одночасно. При одночасному надходженні декількох запитів вони ставляться в чергу і виконуються сервером послідовно.

На стороні сервера використовуються наступні параметри:

– зберігання, захист і доступ до даних
– обробка та перевірка клієнтських запитів
– спілкування з клієнтом

На стороні клієнта використовуються наступні параметри:

– сторінка з графічним інтерфейсом користувача, яка є інтерактивною,
– створення запиту до сервера та його подальша передача,
– отримання відповіді від сервера та її обробка [22]

### 3.1.1 Розпізнавання облич

Для розпізнавання облич була використана бібліотека OpenCV. OpenCV - це бібліотека з відкритим вихідним кодом. Вона підтримується різними мовами програмування, такими як R, Python тощо. Вона працює, ймовірно, на більшості платформ, таких як Windows, Linux та macOS.

Переваги OpenCV:

– OpenCV є безкоштовною бібліотекою з відкритим вихідним кодом.
– OpenCV є швидкою, оскільки написана на мові C/C++, порівняно з іншими.
– З меншим об'ємом оперативної пам'яті OpenCV працює краще.

– Він підтримує більшість операційних систем, таких як Windows, Linux та macOS.

OpenCV - це бібліотека обробки відео та зображень, яка використовується для аналізу зображень та відео, таких як розпізнавання облич, зчитування номерних знаків, редагування фотографій, просунутий робототехнічний зір та багато іншого.

### 3.1.2 Розподілені автентифікатори

Для роботи з розподіленими автентифікаторами було використано фреймворк Ceramic network. Ceramic - це децентралізована мережа для композитних даних. Ceramic дозволяє створювати додатки з композитними даними Web3 так само просто, як переглядати ринок моделей даних, підключати їх до свого додатку, зберігати, оновлювати та отримувати дані з цих моделей. Коли різні додатки повторно використовують одні й ті ж моделі даних, їхні дані автоматично стають сумісними. Децентралізуючи бази даних додатків, Ceramic робить дані сумісними і придатними для повторного використання у всіх додатках.

#### 3.1.2.1 Блокчейн провайдер

Якщо ви хочете, щоб ваш додаток мав доступ до блокчейну, вам потрібно використовувати провайдера. Провайдери використовуються замість

того, щоб запускати вузли блокчейну самостійно. У провайдерів є два основних завдання:

– Повідомити вашому додатку, до якого блокчейну підключитися.
– Після підключення виконувати запити і відправляти підписані транзакції, які змінюють стан блокчейну.

Metamask - один з найпопулярніших блокчейн-провайдерів, і саме він буде використовуватись для підключення нашого додатку до блокчейну Ethereum. Простіше кажучи, провайдери автентифікують користувачів для виконання дій в блокчейні.

### 3.1.3 Клієнт

Фронтенд (або клієнтська частина додатку) запускається в браузері користувача. Ця частина написана на мові програмування Javascript. Додаток може складатися виключно з клієнтської частини, або може вимагати збереження даних користувача протягом більш ніж одного сеансу роботи. Це може бути що завгодно - від графічних редакторів до простих іграшок.

Односторінковий додаток (SPA або односторінковий додаток). За допомогою їх взаємодії між бекендом, так і фронтендом можна спроектувати додаток, який працює без перезавантаження сторінки в браузері. У більш простому варіанті переходи між частинами викликають перезавантаження, але будь-які дії всередині розділу - ні. [23]

#### 3.1.3.1 Мова програмування

Оскільки клієнтська частина виконується в браузері, ніяких специфічних налаштувань немає, тому було використано Javascript. TypeScript є варіантом, однак Javascript є більш гнучким і може зробити роботу програми швидшою.

#### 3.1.3.1 Фреймворк

Вибір фреймворку є більш цікавим, оскільки існує безліч можливостей для створення веб-сторінки на Javascript. Найбільш популярними на даний момент є три фреймворки: React, Angular та VueJs. Оскільки React та Angular призначені для масштабних додатків зі складною бізнес-логікою, ми обрали для створення додатку використання VueJs.

Vue - це сучасний фреймворк для створення користувацьких інтерфейсів. Vue, на відміну від інших монолітних фреймворків, побудований з нуля для швидкого застосування. Основна бібліотека займається лише шаром відображення і може бути використана в поєднанні з іншими бібліотеками або існуючими проектами.

### 3.1.4 Сервер

#### 3.1.4.1 Мова програмування

Існує багато мов програмування для створення серверних компонентів веб-додатків. Ми обрали Python для роботи з двох причин. Перша полягає в тому, що моделі розпізнавання облич написані на Python, тому логічно, що з'єднати їх з бекендом на основі Python буде найпростіше. Друга причина - обраний фреймворк. Інші мови програмування потребують великої та складної інфраструктури на стороні сервера, але є фреймворк, який дозволяє швидко розпочати роботу, не вимагаючи багатьох рівнів абстракції.

#### 3.1.4.2 Фреймворк

Flask - це мікрофреймворк на основі Python. Він класифікується як мікрофреймворк, тому що не вимагає спеціальних інструментів або бібліотек. У ньому відсутній рівень абстракції бази даних, валідація форм та інші компоненти, де сторонні бібліотеки забезпечують загальну функціональність. З іншого боку, Flask дозволяє створювати розширення, які можуть додавати функціональність додатку так, ніби вони вбудовані в саму Flask.

## 3.2 Пояснення до основних робочих процесів

Для кращого розуміння системи, пропоную пройтися основними робочими процесами, які використовувалися в цій роботі.

### 3.2.1 Збереження біометричної інформації

Щоб реалізувати вхід за біометричними даними, спочатку необхідно ці дані зберегти й обробити. Тому перший раз, коли користувач заходить у застосунок і авторизується, йому пропонують зберегти його біометричні дані.

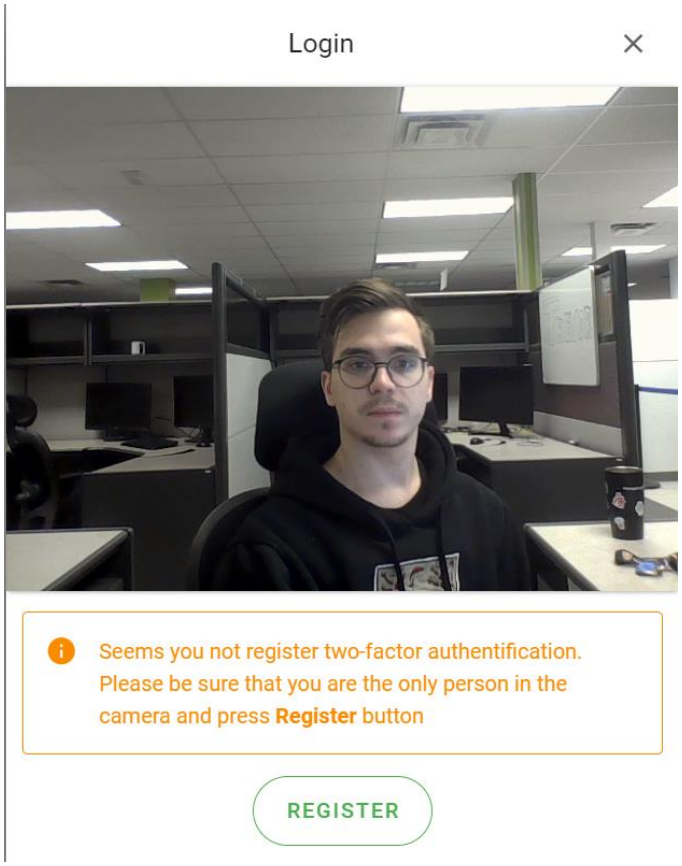


Рисунок 3.1 Реєстрація біометричних даних

На сервер ми передаємо масив картинок, який зберігаємо для подальшої обробки і статистики.

```
def save_faces(userName, images):
    folder = Path(constants.PathToImages + userName)
    print(folder)
    if folder.exists() and folder.is_dir():
            shutil.rmtree(folder)
    os.makedirs(folder)
    for i, image in enumerate(images):
        imgdata = base64.b64decode(image[22:])
        filename = str(i)+'.png'
        with open(os.path.join(folder, filename), "wb+") as fh:
            fh.write(imgdata)
        print("Photo ", i, " added")
```

Рисунок 3.2 Збереження біометричних даних

Після цього ми беремо зображення і зберігаємо числові характеристики особливостей обличчя людини і після цього шифруємо його в спеціальний файл, щоб дані були захищені.

```
def save_encodings(userName):

    imagePath = list(paths.list_images(os.path.join(constants.PathToImages, userName)))
    encoding_file_name = userName + ".pic"
    encodings = []
    for img_path in imagePath:
        img = cv2.imread(img_path)
        rgb_img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

        try:
            img_encoding = face_recognition.face_encodings(rgb_img)[0]
        except:
            print("Face not found!   ", img_path)
            continue
        encodings.append(img_encoding)
    filename = os.path.join(constants.PathToEncodings, encoding_file_name)
    with open(os.path.join(filename), "wb+") as f:
        f.write(pickle.dumps(encodings))
    print("encoding saved")
```

Рисунок 3.3 Збереження масиву ознак облич

### 3.2.2 Розпізнавання обличчя

Після того як людина вже зареєструвалася, наступним кроком буде розпізнавання обличчя. Ми розв'язуємо задачу перевірки, чи є людина, яка хоче увійти в акаунт, дійсно власником акаунта. Для цього ми беремо збережені зображення і за допомогою функції порівняння облич порівнюємо поточне зображення зі збереженим масивом особливостей обличчя власника акаунта.

Якщо ми знайшли збіг, то виводимо, що обличчя знайдено і результат валідний. Однак якщо результат показав, що людина не є власником акаунта, то програма виведе, що результат не валідний. Останній варіант результату роботи програми є випадок, коли на фото не було знайдено жодного обличчя, тоді програма виведе, що обличчя не знайдено.

```
def find_face(userName, frame) -> str:
    current_face_encodings = load_encoding_images(userName)

    frame = base64ToImage(frame)
    small_frame = cv2.resize(frame, (0, 0), fx=frame_resizing, fy=frame_resizing)
    rgb_small_frame = cv2.cvtColor(small_frame, cv2.COLOR_BGR2RGB)
    face_locations = face_recognition.face_locations(rgb_small_frame)
    face_encodings = face_recognition.face_encodings(rgb_small_frame, face_locations)
    for face_encoding in face_encodings:
        # See if the face is a match for the known face(s)
        matches = face_recognition.compare_faces(current_face_encodings, face_encoding)

        if True not in matches:
            return constants.FaceAuthStatus.Invalid

        face_distances = face_recognition.face_distance(current_face_encodings, face_encoding)
        best_match_index = np.argmin(face_distances)
        if matches[best_match_index]:
            return constants.FaceAuthStatus.Valid

    return constants.FaceAuthStatus.FaceNotFound
```

Рисунок 3.4 Функція для розпізнавання обличчя

### 3.2.3 Вхід в систему за допомогою блокчейну

Щоб зайти в систему блокчейна ми використовуємо зовнішнього провайдера Metamask, то це і описано вище. Як ви можете бачити на малюнку, під час входу в блокчейн з'являється спливаюче вікно з інформацією, на якому зазначено джерело, який сайт хоче підключитися до гаманця, номер самого гаманця, а також найважливіше - розподілений ідентифікатор (did).

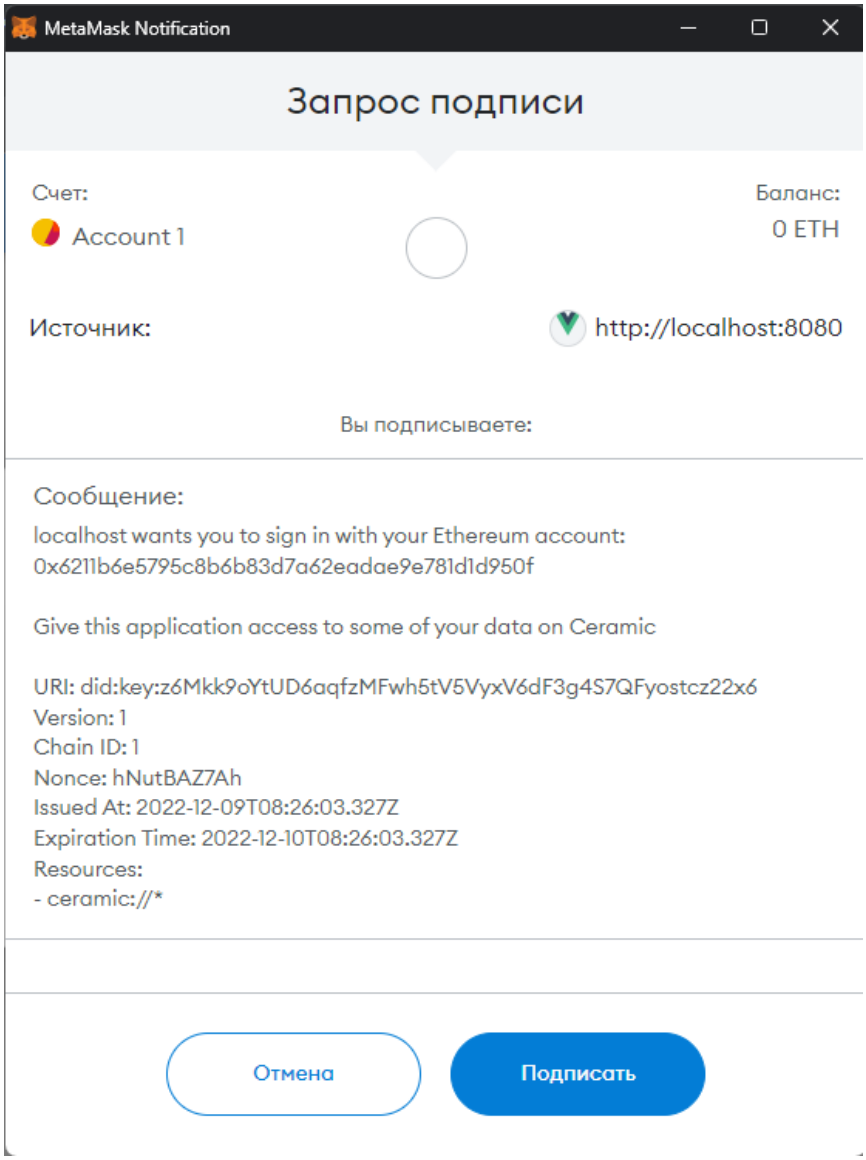


Рисунок 3.5 Окно провайдера блокчейна

Як можна бачити з коду, ми підключаємося до клієнта блокчейна за певною адресою - це адреса блокчейна, де міститься вся інформація, потрібна нам. Також ми отримуємо схеми даних, за якими ми будемо отримувати профіль користувача в розподіленій мережі web 3.0

```
ceramic: new CeramicClient("https://ceramic-clay.3boxlabs.com"),
blockchainProfile: null,
aliases: {
  schemas: {
    basicProfile:
      "ceramic://k3y52l7qbv1frxt706gqfzmq6cbqdkptzk8uudaryhlkf6ly9vx21hqu4r6k1jqio",
  },
  definitions: {
    BasicProfile:
      "kjzl6cwe1jw145cjbeko9kil8g9bxszjhyde21ob8epxuxkaon1izyqsu8wgcic",
  },
```

Рисунок 3.6 Строки підключення до блокчейну

Нижче представлена функція підключення. Вона складається з двох частин: спочатку ми підключаємося до облікового запису за допомогою провайдера, а потім отримуємо профіль із мережі web 3.0, щоб передати його потім далі до нашого застосунку та використовувати.

```
async connect() {
  if (window.ethereum == null) {
    console.error("No injected Ethereum provider found");
  }
  try {
    this.loading = true;
    await this.authenticateWithEthereum(window.ethereum);
    await this.getProfileFromCeramic();
    this.loading = false;
  } catch (error) {
    console.error(error);
    this.loading = false;
  }
},
```

Рисунок 3.7 Функція логіну до блокчейну

```
async authenticateWithEthereum(ethereumProvider) {
  const accounts = await ethereumProvider.request({
    method: "eth_requestAccounts",
  });

  const authProvider = new EthereumAuthProvider(
    ethereumProvider,
    accounts[0]      You, 2 days ago • feat: Create firs
  );
  const session = new DIDSession({ authProvider });
  const did = await session.authorize();
  console.log("my did", did);
  this.ceramic.did = did;
},
```

Рисунок 3.8 Завдяки провайдеру блокчейну, авторизуємося

### 3.2.4 Оновлення профілю у блокчейні

Також було зроблено можливість оновити профіль користувача в 3.0 в інтернеті, для цього ми також використовуємо сховище даних, яке прив'язане до блокчейну, що був описаний раніше, і схеми даних, за якими ми дістаємо профіль користувача.

```
async onSubmit() {
  try {
    this.loading = true;
    //use the DIDDatastore to merge profile data to Ceramic
    await this.datastore.merge("BasicProfile", this.profileToEdit);

    //use the DIDDatastore to get profile data from Ceramic
    this.profileToEdit = await this.datastore.get("BasicProfile");

    this.loading = false;
  } catch (error) {
    this.loading = false;
    console.error(error);
  }
},
```

Рисунок 3.9 Функція оновлення профілю

## 3.3 Приклад роботи веб додатку

Коли користувач вперше заходить у застосунок, він бачить простий інтерфейс - одну велику кнопку під'єднатися, після натискання на яку почнеться робота із застосунком.

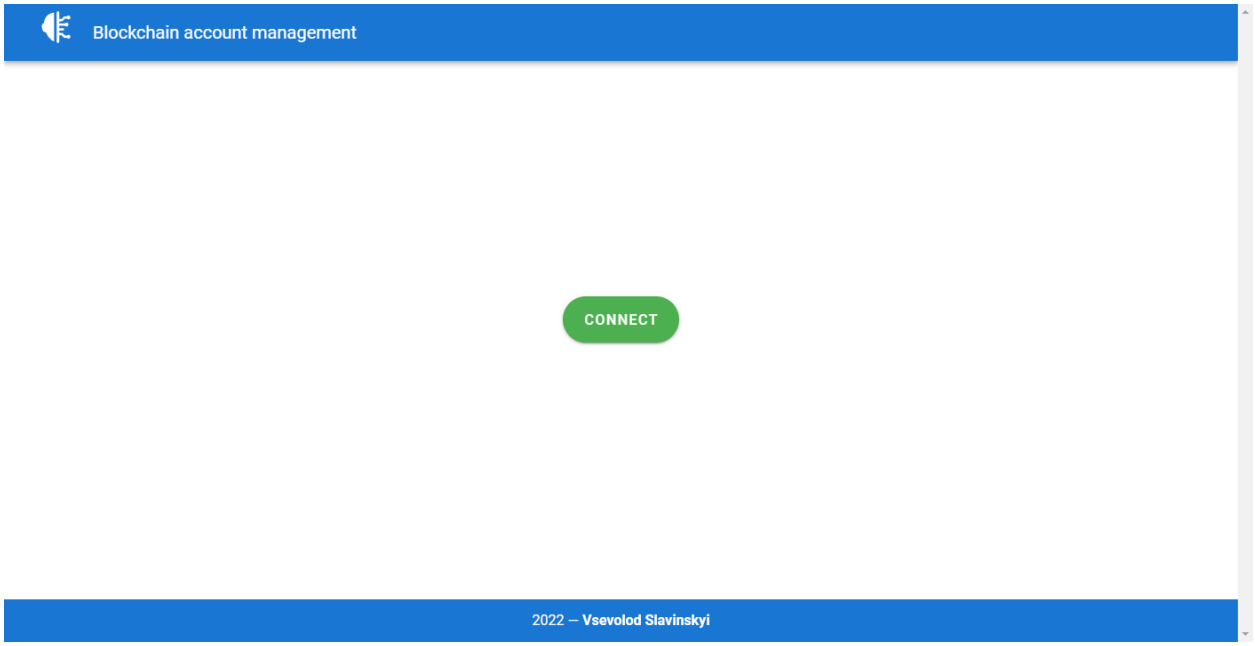


Рисунок 3.Кнопка логін

Щойно кнопку було натиснуто, з'явиться спливаюче вікно з пропозицією увійти за допомогою свого криптогаманця.


Рисунок 3.11 Діалог автентифікації

Відразу після цього після натискання на кнопку з'явиться ще одне спливаюче вікно від провайдера з пропозицією підписати повідомлення про вхід і погодитися з умовами. Це повідомлення може прийти як і на комп'ютер, так і на телефон залежно від того, де встановлено застосунок криптогаманця.

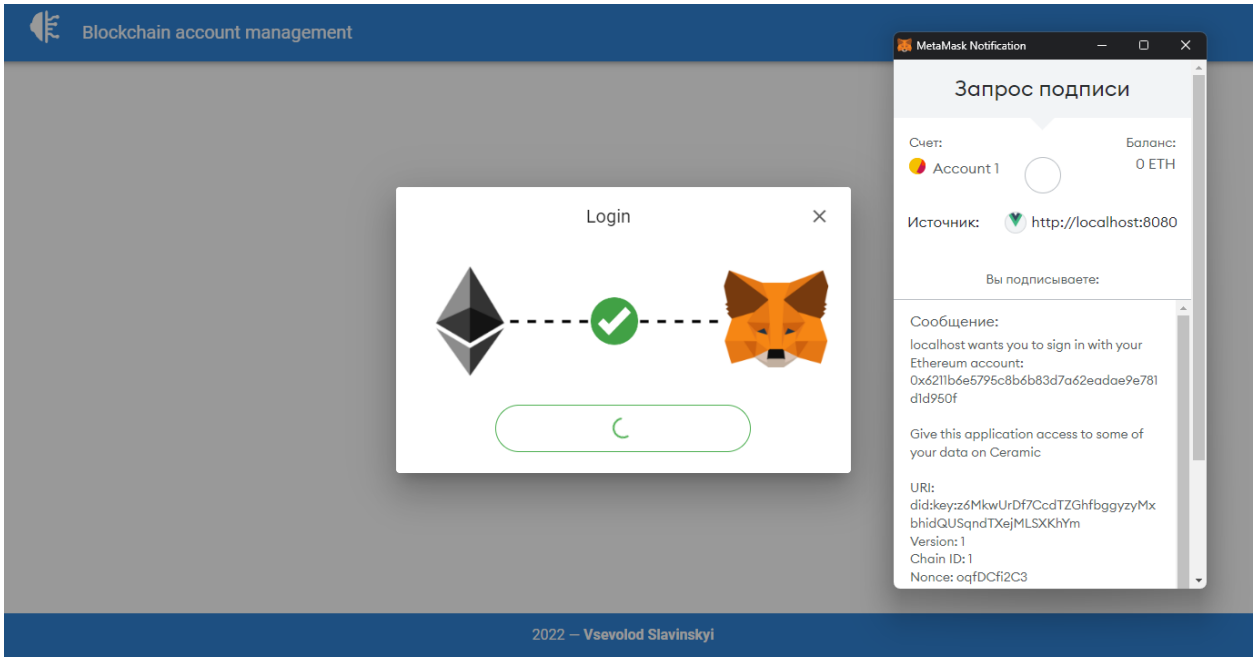


Рисунок 3.12 Окно провайдера блокчейна

Після того як перший етап автентифікації пройдено користувачеві запропонують скористатися другим фактором модифікації - своєю біометрією. Програма намагається знайти користувача в базі даних, якщо ж вона його не знайшла, то спершу запропонує зареєструвати своє обличчя як другий фактор автентифікації.

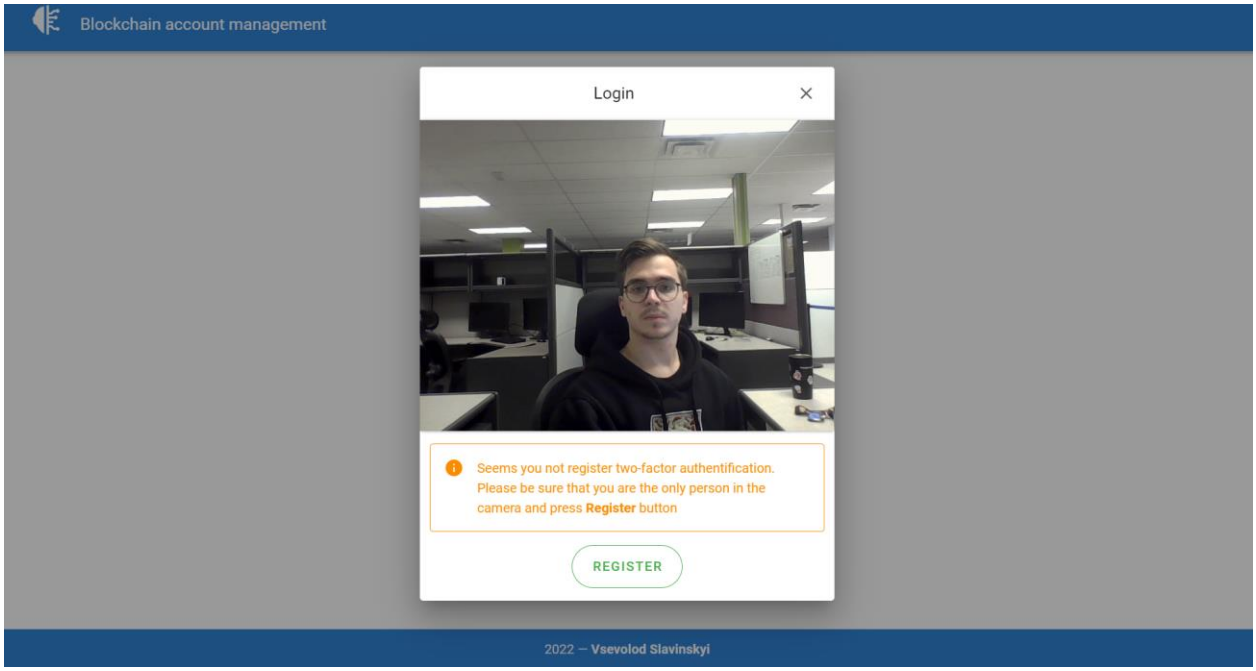


Рисунок 3.13 Початок реєстрації біометричних даних

Щойно користувач почне натискати на кнопку зареєструватися, програма почне робити серію коротких фотографій, щоб потім зберегти їх у базу й обробити для подальшого використання.

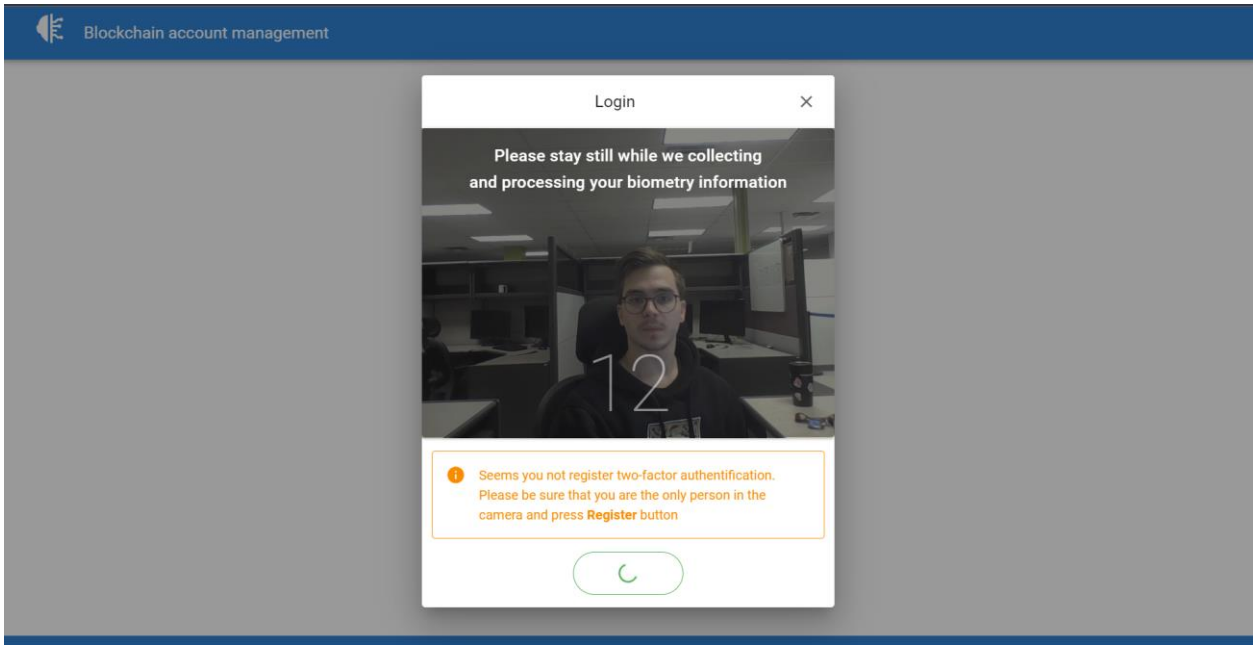


Рисунок 3.14 Процес реєстрації біометричних даних

Щойно користувач зареєструє свої дані, додаток запропонує йому увійти у свій профіль, використовуючи щойно отримані біометричні дані. Таке ж вікно користувач побачить усі наступні рази, коли заходитиме в додаток, оскільки про нього вже буде збережена біометрична інформація. Після натискання на кнопку логін програма зробить фотографію користувача і надішле її на сервер, де проаналізує і зрозуміє, чи є людина на фото володарем гаманця.

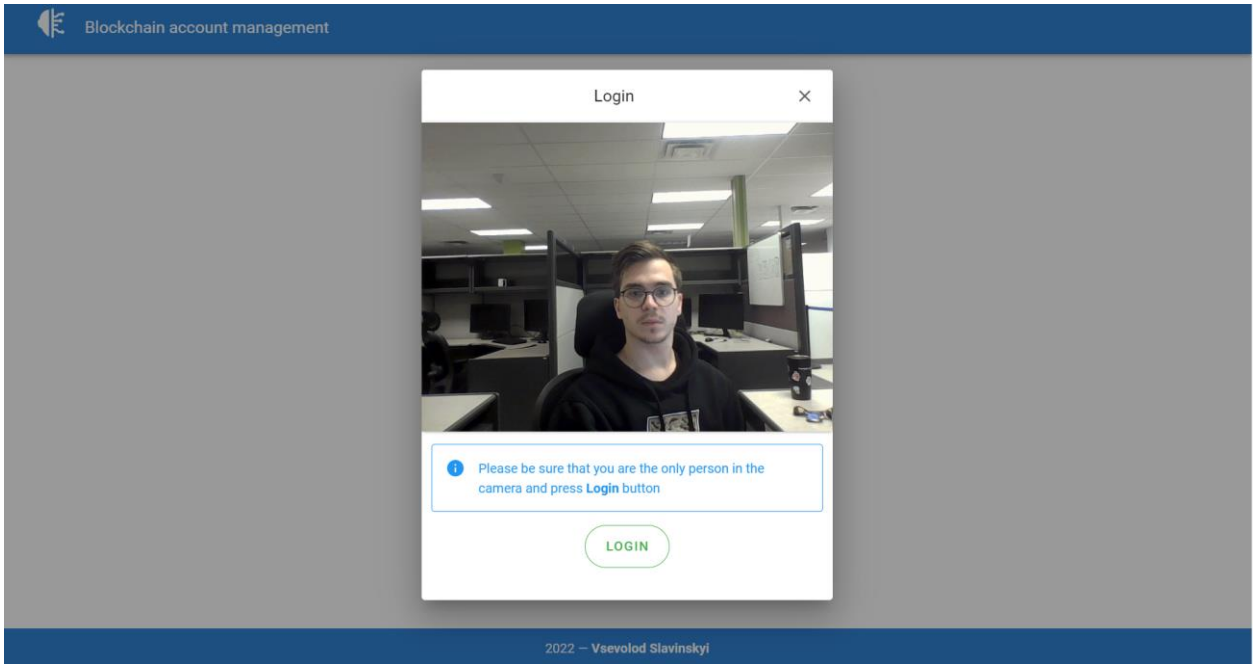


Рисунок 3.15 Процес виконання другого фактору автентифікації

І нарешті після двох чинників автентифікації з'явиться профіль людини, який зберігається в розподіленому інтернеті web 3.0. На екрані ви бачите основні поля, доступні користувачам для перегляду та редагування.

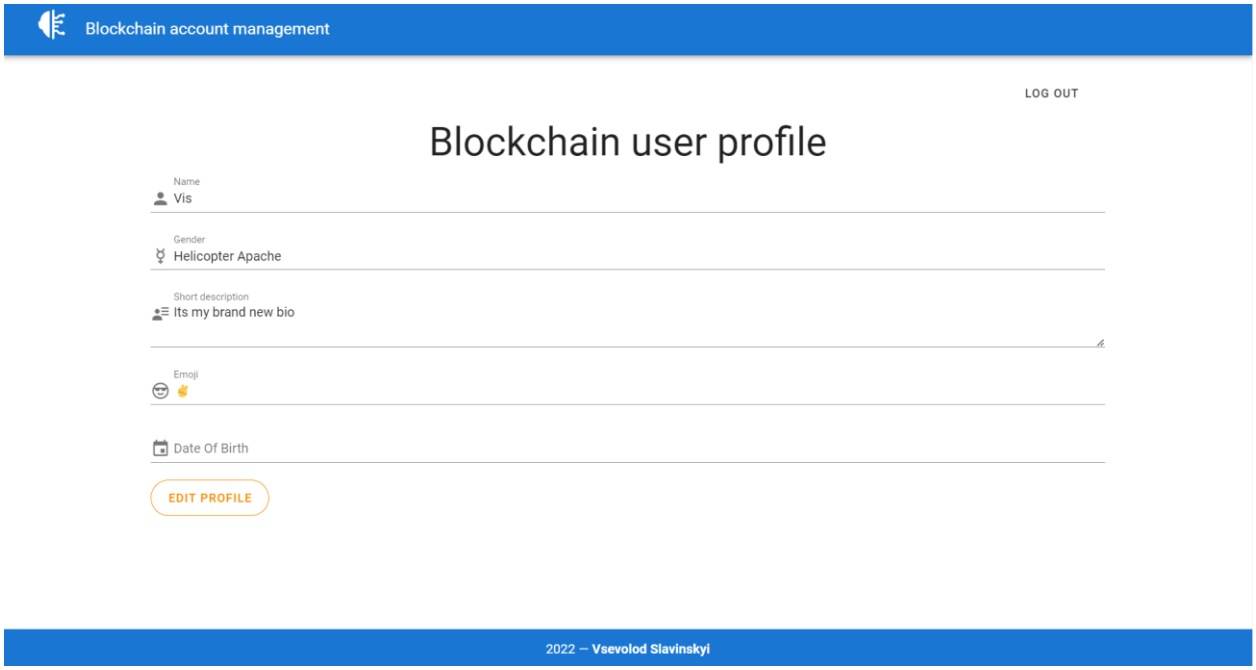


Рисунок 3.16 Профіль користувача, який зберігається у блокчейні

Після натискання на кнопку редагувати профіль усі текстові поля стануть активними, і ви можете оновити інформацію про себе та зберегти її після цього.

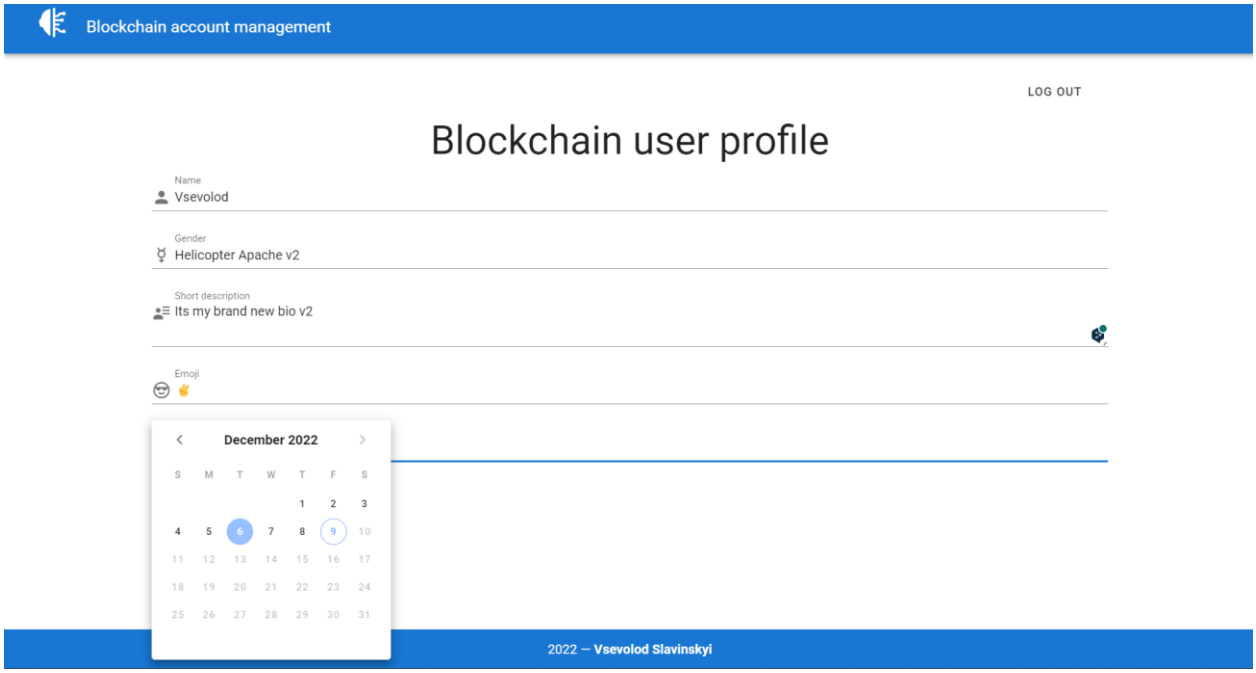


Рисунок 3.17 Редагування профілю

## 3.4 Висновки до розділу 3

В даному розділі було проаналізовано архітектуру практичної складової, обґрунтування обраних технологій виходячи з основних вимог до системи, а також технології, що використовуються на клієнтському, серверному принципі та взаємодію між ними.

Для демонстрації практичної складової було обрано веб-додаток на основі клієнт-серверної архітектури та попередньо навчених моделей. Для розпізнавання облич використано бібліотеку OpenCV, а для роботи з розподіленими автентифікаторами - фреймворк Ceramic Network. Для швидкого застосування було обрано створення клієнтської частини за допомогою фреймворку Vue Js. Фреймворк Flask був обраний в якості основи для серверної складової.

У наступному пункті було показано роботу основних функцій і процедур у веб-застосунку, після цього було описано роботу мультифакторної автентифікації на основі блокчейна і розпізнавання облич.

# РОЗДІЛ 4 РОЗРОБКА СТАРТАП-ПРОЕКТУ

Останні роки довели, що стартапи сильніші, ніж більшість думала. Гнучкий та інноваційний підхід були ключовими, коли світ зіштовхнувся з такими проблемами, як віддалена робота, масштабні зміни в галузі та ринку, а також абсолютно нова реальність.

Здатність стартапів швидко змінюватися та адаптуватися і є ключовою властивістю даного виду бізнесу, не кажучи вже про те, що багато стартапів продовжували рости та розширювати свої команди.

Стартап — це бізнес-структура, що розвивається та працює на основі інновацій, створена для вирішення проблеми шляхом надання нової пропозиції в умовах надзвичайної невизначеності.

Власне, стартап – це бізнес, який:

- швидко росте;
- порушує ринок або галузь (щось нове, що змушує конкурентів удосконалюватись);
- вирішує проблему;
- працює в умовах надзвичайної невизначеності.

Багато підприємців і відомих бізнес-магнатів визначають стартап як культуру та менталітет побудови бізнесу на основі інноваційної ідеї для вирішення критичних проблем.

Одне, що відрізняє стартапи від інших компаній — це зв’язок між їхнім продуктом та його попитом. Стартапи мають продукти, орієнтовані на невикористаний ринок. Підприємці-початківці знають ідеальну стратегію, щоб створити продукт, який хоче ринок, а також охопити й обслуговувати їх усіх. Це викликає швидке зростання.

## 4.1 Опис ідеї проекту

В межах підпункту було проаналізовано і подано у вигляді таблиць:

- зміст ідеї;
- можливі напрямки застосування;
- можливі напрямки застосування;
- основні вигоди, що може отримати користувач товару;
- чим відрізняється від існуючих аналогів та замінників.

Перші три пункти подані у вигляді таблиці (таблиця 4.1) і дають цілісне уявлення про зміст ідеї та можливі базові потенційні ринки, в межах яких потрібно шукати групи потенційних клієнтів.

Таблиця 4.1 — Опис ідеї стартап-проекту

| Зміст ідеї | Напрямки застосування | Вигоди для користувача |
|---|---|---|
| Дана комплексна система дозволяє розвязати проблему видачі контекстуальних рекомендацій. | Видача рекомендацій користувачу | Збільшення прибутку шляхом покращення якості рекомендацій |
| | Генерування динамічних вкладень графу взаємодій | Збільшення об’єму клієнської бази |

Аналіз потенційних техніко-економічних переваг ідеї (чим відрізняється від існуючих аналогів та замінників) порівняно із пропозиціями конкурентів передбачає:

- визначення переліку техніко-економічних властивостей та характеристик ідеї;

- визначення попереднього кола конкурентів (проектів-конкурентів) або товарів-замінників чи товарів-аналогів, що вже існують на ринку, та проводиться збір інформації щодо значень техніко-економічних показників для ідеї власного проекту та проектів-конкурентів відповідно до визначеного вище переліку;
- проводиться порівняльний аналіз показників: для власної ідеї визначаються показники, що мають:
- гірші значення (W, слабкі);
- аналогічні (N, нейтральні) значення;
- кращі значення (S, сильні) (табл. 4.2).

Таблиця 4.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

| № п/п | Технікоекономічні характеристики ідеї | (потенційні) товари/концепції конкурентів | | | W (слабка сторона) | N (нейтральна сторона) | S (сильна сторона) |
|---|---|---|---|---|---|---|---|
| | | Мій проект | Adobe Target | Amazon Personalize | | | |
| 1 | Форма виконання | Надавання послуг | Надавання послуг | Надавання послуг | | + | |
| 2. | Собівартість | Низька | Висока | Висока | | | + |
| 3. | Функціонал | Широкий | Широкий | Широкий | + | | |

Визначений перелік слабких, сильних та нейтральних характеристик та властивостей ідеї потенційного товару є підґрунтям для формування його конкурентоспроможності.

## 4.2 Технологічний аудит ідеї проекту

В межах даного підрозділу було проведено аудит технології, за допомогою якої можна реалізувати ідею проекту (технології створення товару). Визначення технологічної здійсненності ідеї проекту передбачає аналіз таких складових (таблиця 4.3):

1. За якою технологією буде виготовлено товар згідно ідеї проекту?
2. Чи існують такі технології, чи їх потрібно розробити/доробити?
3. Чи доступні такі технології авторам проекту?

Таблиця 4.3 – Технологічна здійсненність ідеї проекту

| № п/п | Ідея проекту | Технології її реалізації | Наявність технологій | Доступність технологій |
|---|---|---|---|---|
| 1 | Створення системи генерування рекомендацій користувачу за даними покупок користувачів | Використання мови програмування Python | Наявна | Доступна |
| | | Використання мови програмування C# | Відсутні | Недоступні |

| | | | | |
|---|---|---|---|---|
| | | Використання мови С++ | Відсутні | Недоступні |
| Обрана технологія реалізації ідеї проекту: мова програмування Python. | | | | |

За результатами аналізу таблиці зроблено висновок щодо можливості технологічної реалізації проекту. Технологічним шляхом реалізації проекту було обрано такі технології, як Python через доступність та безкоштовність.

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів.

Спочатку було проведено аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (таблиця 4.4).

Таблиця 4.4 – Попередня характеристика потенційного ринку стартап-проекту

| № п/п | Показники стану ринку (найменування) | Характеристика |
|---|---|---|
| 1 | Кількість головних гравців, од | 50 |
| 2 | Загальний обсяг продаж, грн/ум.од | 10000 |
| 3 | Динаміка ринку | Зростає |
| 4 | Наявність обмежень для входу | Немає |

| | | |
|---|---|---|
| 5 | Специфічні вимоги до стандартизації та сертифікації | Немає |
| 6 | Середня норма рентабельності в галузі, % | 20% |

За результатами аналізу таблиці 4.4 було зроблено висновок, що ринок є привабливим для входження.

Надалі були визначені потенційні групи клієнтів, їх характеристики та сформовано орієнтовний перелік вимог до товару для кожної групи (табл. 4.5).

Таблиця 4.5 – Характеристика потенційних клієнтів стартап-проекту

| Потреба, що формує ринок | Цільова аудиторія | Відмінності у поведінці різних потенційних цільових груп клієнтів | Вимоги споживачів до товару |
|---|---|---|---|
| Точна та швидка генерація рекомендацій цільовій аудиторії | Власник бізнесу | Велика кількість даних | Простота використання, висока точність |
| Підбір рекомендацій | Кінцевий користувач | Цікавить ростота у використанні, низька ціна підтримки системи | Швидкість створення, низька ціна |

Після визначення потенційних груп клієнтів було проведено аналіз ринкового середовища: складено таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають (табл. 4.6, 4.7).

Таблиця 4.6 – Фактори загроз

| № п/ п | Фактор | Зміст загрози | Можлива реакція компанії |
|---|---|---|---|
| 1 | Конкуренція | Вихід на ринок продуктів з кращими характеристиками | Передбачити додаткові переваги власного програмного продукту (ПП) для того, щоб повідомити про них саме після виходу на ринок конкурентів. Вдосконалення технічних моментів власного продукту. Обрати нову цільову аудиторію і зосередитися на ній: зниження цін. |
| 2 | Зміна потреб користувачів | Користувачам необхідний сервіс з більшим/новим функціоналом. | Розроблення гнучкої архітектури програмного забезпечення для легшого впровадження нового функціоналу |

Таблиця 4.7 – Фактори можливостей

| № п/п | Фактор | Зміст можливості | Можлива реакція компанії |
|---|---|---|---|
| 1 | Гнучкі ціни | Зменшення ціни товару задля збільшення попиту | Введення власних гнучких цін |

| | | | |
|---|---|---|---|
| 2 | Поява нових методів квантування зображення | З’являться нові методи, що будуть швидше, та більш точно квантувати зображення | Покращити ПП додаванням нового функціоналу, розширення можливостей |

Надалі було проведено аналіз пропозиції: визначили загальні риси конкуренції на ринку (таблиця 4.8).

Таблиця 4.8 – Ступеневий аналіз конкуренції на ринку

| Особливості конкурентного середовища | В чому проявляється дана характеристика | Вплив на діяльність підприємства |
|---|---|---|
| 1. Тип конкуренції- чиста | Існує величезна кількість конкурентів на ринку. | Якісно провести рекламу. |
| 2. За рівнем конкурентної боротьби - міжнародний | Компанії-конкуренти з інших країн | Створити основу ПП таким чином, щоб можна було легко переробити даний ПП для використання у галузях інших країн. |
| 3. За галузевою ознакою - міжгалузева | Продукт може використовуватись для різних галузей | Постійне вдосконалення продукту, що не має прив’язки до сфери |
| 4. Конкуренція за видами товарів: - товарно-видова | Конкуренція між видами ПП, їх особливостями. | Створити ПП, враховуючи недозображення конкурентів |

| | | |
|---|---|---|
| 5. За характером конкурентних переваг - нецінова | Вдосконалення технології створення ПП, щоб собівартість була нижчою | Удосконалення моделі. Використання більш дешевих технологій для розробки, ніж використовують конкуренти, але тільки якщо ці технології відповідають необхідним вимогам якості. |
| 6. За інтенсивністю - не марочна | Бренд присутній, але його роль незначна | Реклама, участь у конференціях, семінарах. |

Було проведено аналіз конкуренції у галузі за моделлю М. Портера (табл. 4.9).

Таблиця 4.9 – Аналіз конкуренції в галузі за М. Портером

| Складові аналізу | Прямі конкуренти в галузі | Потенційні конкуренти | Постачальники | Клієнти | Товаризамінники |
|---|---|---|---|---|---|
| | Навести перелік прямих конкурентів | Визначити бар’єри входження в ринок | Визначити фактори сили постачальників | Визначити фактори сили споживачів | Фактори загроз з боку замінників |
| | Amazon Personalize | Наявність вже існуючих рішень | - | Контроль якості продукту | Наявність більш широкого функціоналу, зручнішого інтерфейсу та авторитет |

| | | | | | |
|---|---|---|---|---|---|
| Висновки: | Досить інтенсивна конкурентна боротьба з іншими гравцями | Є можливості виходу на ринок, але є і конкуренти. | - | Клієнти диктують умови роботи на ринку: зручний інтерфейс | Необхідно випускати ПЗ не гірше, ніж у конкурентів та розширяти функціонал. |

За результатами аналізу було зроблено висновок про можливість роботи на ринку з огляду на конкурентну ситуацію.

Цей висновок був врахований при формулюванні переліку факторів конкурентоспроможності у наступному пункті. На основі аналізу конкуренції, проведеного в таблиці, а також із урахуванням характеристик ідеї проекту (табл. 4.2), вимог споживачів до товару (табл. 4.5) та факторів маркетингового середовища (табл. 4.6, 4.7) визначається та обґрунтовується перелік факторів конкурентоспроможності.

Аналіз оформлено у (табл. 4.10).

Таблиця 4.10 – Обґрунтування факторів конкурентоспроможності

| № п/п | Фактор конкурентоспроможності | Обґрунтування |
|---|---|---|
| 1 | Ціна | Один із факторів для вибору продукту клієнтом. |
| 2 | Якість | Один із факторів для вибору продукту клієнтом. |
| 3 | Зручність роботи з програмою | Дозволяє користувачу легко працювати з програмою |

За визначеними факторами конкурентоспроможності (табл. 4.10) проведено аналіз сильних та слабких сторін стартап-проекту (табл. 4.11).

Таблиця 4.11 – Порівняльний аналіз сильних та слабких сторін проекту

| № п/п | Фактор конкурентоспроможності | Бали 1-20 | Рейтинг товарів-конкурентів у порівнянні | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | –3 | –2 | –1 | 0 | +1 | +2 | +3 |
| 1 | Ціна | 15 | | | | | * | | |
| 2 | Якість | 10 | | | * | | | | |
| 3 | Зручність роботи з програмою | 15 | | | | | * | | |

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (табл. 4.12) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (таблиця 4.11).

Перелік ринкових загроз та ринкових можливостей було складено на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Ринкові загрози та ринкові можливості є наслідками впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення.

Таблиця 4.12 – SWOT-аналіз стартап-проекту

| Сильні сторони: | Слабкі сторони: |
|---|---|
| Якість<br>Простота використання<br>Висока швидкодія | Дуже насичений ринок, мала кількість функціоналу, відсутня кросплатформеність. |

| | |
|---|---|
| Можливості:<br>насичення ринку новим підходом до прогнозування; різноманітна клієнтура, вдосконалення системи | Загрози:<br>Конкуренція |

На основі SWOT-аналізу було розроблено альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок.

Визначені альтернативи були проаналізовані з точки зору строків та ймовірності отримання ресурсів (таблиця 4.13).

Таблиця 4.13 – Альтернативи ринкового впровадження стартап-проекту

| № п/п | Альтернатива ринкової поведінки | Ймовірність отримання ресурсів | Строки реалізації |
|---|---|---|---|
| 1 | PR, просування бренду | 50% | 6 місяців |
| 2 | Перехід на безкоштовне розповсюдження | 75% | 3 місяців |
| 3 | Партнерство для об’єднання продукції | 65% | 2 місяці |

Після аналізу було обрано альтернативу №2.

## 4.3 Аналіз ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: було проведено опис цільових груп потенційних споживачів (таблиця 4.14).

Таблиця 4.14 – Вибір цільових груп потенційних споживачів

| № п/п | Опис профілю цільової групи потенційних клієнтів | Готовність споживачів сприйняти продукт | Орієнтовний попит в межах цільової групи | Інтенсивність конкуренції в сегменті | Простота входу у сегмент |
|---|---|---|---|---|---|
| 1 | Підприємці | Висока | Високий | Сильна | Просто |
| 2 | Великі компанії | Середня: велика конкуренція і можливість власних веб-відділів | Високий | Сильна | Складно |
| 3 | Маленькі компанії. | Низька | Низький | Слабка | Середня |
| Які цільові групи обрано:1,2,3 | | | | | |

За результатами аналізу потенційних груп споживачів було обрано цільові групу, для яких буде запропоновано даний товар, та визначено стратегію охоплення ринку - стратегію диференційованого маркетингу (компанія працює з декількома сегментами).

Для роботи в обраних сегментах ринку сформовано базову стратегію розвитку (таблиця 4.15).

Таблиця 4.15 – Визначення базової стратегії розвитку

| № п/п | Обрана альтернатива розвитку проекту | Стратегія охоплення ринку | Ключові конкурентоспроможні позиції відповідно до обраної альтернативи | Базова стратегія розвитку |
|---|---|---|---|---|
| 1 | Постійне оновлення і покращення продукту | Ринкове позиціонування на індивіуальних користувачів | Швидкодія, якість продукту | Концентрований маркетинг |

Наступним кроком обрано стратегію конкурентної поведінки (таблиця 4.16).

Таблиця 4.16 – Визначення базової стратегії конкурентної поведінки

| № п/п | Чи є проект «першопрохідцем» на ринку? | Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів? | Чи буде компанія копіювати основні характеристики товару конкурента, і які? | Стратегія конкурентної поведінки |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| 1 | Ні. | Компанія буде шукати нових споживачів та забирати існуючих у конкурентів | Буде копіювати, удосконалювати та створювати свої унікальні пропозиції | Зайняття конкурентної ніші |

На основі вимог споживачів з обраних сегментів до постачальника (стартап-компанії) та до продукту (табл. 4.5), а також в залежності від обраної базової стратегії розвитку (табл. 4.15) та стратегії конкурентної поведінки (таблиця 4.16) розроблено стратегію позиціонування (таблиця 4.17), що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати торгівельну марку/проект.

Таблиця 4.17 – Визначення стратегії позиціонування

| № п/п | Вимоги до товару цільової аудиторії | Базова стратегія розвитку | Ключові конкурентоспроможні позиції власного стартаппроекту | Вибір асоціацій, які мають сформувати комплексну позицію власного проекту |
|---|---|---|---|---|
| 1 | Легкість розуміння, зручний інтерфейс, надійний, швидкий, точний та достовірний ПП для | Стратегія диференціації | Позиція на основі порівняння фірми з товарами конкурентів; Відмінні | Економія часу; Зручність застосування; Практичність та точність результату |

| | | | | |
|---|---|---|---|---|
| | генерації рекомендацій. | | особливості споживача | |

Результатом виконання підрозділу стала узгоджена система рішень щодо ринкової поведінки стартап-компанії, яка визначає напрями роботи стартап- компанії на ринку.

## 4.4 Розроблення маркетингової програми стартап-проекту

Сформовано маркетингову концепцію товару, який отримає споживач. Для цього підсумовано результати попереднього аналізу конкурентоспроможності товару (таблиця 4.18). Концепція товару – письмовий опис фізичних та інших характеристик товару, які сприймаються споживачем, і набору вигод, які він обіцяє певній групі споживачів.

Таблиця 4.18 – Визначення ключових переваг концепції потенційного товару

| № п/п | Потреба | Вигода, яку пропонує товар | Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити) |
|---|---|---|---|

| | | | |
|---|---|---|---|
| 1 | Швидкість отримання результату | Швидка видача рекомендацій наступної пропозиці | Необхідно покращити швидкість навчання моделі для видачі рекомендацій наступної пропозиці |
| 2 | Зручність застосування | Нативна підтримка мови програмування Python | Розробка зручного прикладного програмного інтерфейсу |
| 3 | Практичність та точність результату | Користувач отримує точні (з малою похибкою розбіжності) результати рекомендацій. | Користувач на виході роботи ПП отримує модель та прогноз, котрі відповідають необхідним показникам варіативності та точності. |

Розроблено трирівневу маркетингову модель товару: уточнюється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання (таблиця 4.19).

1-й рівень. При формуванні задуму товару вирішується питання щодо того, засобом вирішення якої потреби і / або проблеми буде даний товар, яка його основна вигода. Дане питання безпосередньо пов’язаний з формуванням технічного завдання в процесі розробки конструкторської документації на виріб.

2-й рівень. Цей рівень являє рішення того, як буде реалізований товар в реальному/ включає в себе якість, властивості, дизайн, упаковку, ціну.

3-й рівень  Товар з підкріпленням (супроводом) – додаткові послуги та переваги для споживача, що створюються на основі товару за задумом і товару в реальному виконанні (гарантії якості , доставка, умови оплати та ін).

Таблиця 4.19 – Опис трьох рівнів моделі товару

| Рівні товару | Сутність та складові | | |
|---|---|---|---|
| I. Товар за задумом | Генерація палітри кольорів зображення за допомогою інтелектуальних систем | | |
| II. Товар реальному виконанні | Властивості/характеристики | М/Нм | Вр/Тх/Тл/Е/Ор |
| | 1. Індивідуальний підхід.<br>2. Низька ціна.<br>3. Простота у використанні. | 1.Нм<br>2.Нм<br>3.Нм | 1.Технологічна<br>2.Економічна<br>3.Технологічна |
| | Якість: тестування фірмами аудиторами | | |
| | Пакування: відсутнє | | |
| | Марка: ROSO | | |

Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар (остаточне визначення ціни відбувається під час фінансово-економічного аналізу проекту), яке передбачає аналіз ціни на товари-аналоги або товари субститути, а також аналіз рівня доходів цільової групи споживачів (таблиця 4.20). Аналіз проведено експертним методом.

Таблиця 4.20 – Визначення меж встановлення ціни

| № п/п | Рівень цін на товари-замінники | Рівень цін на товари-аналоги | Рівень доходів цільової групи споживачів | Верхня та нижня межі встановлення ціни на товар/послугу |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| 1 | 2000$ | 3700$ | У всіх трьох груп високий рівень доходів | 900$-- |

Наступним кроком є визначення оптимальної системи збуту, в межах якого було прийняте рішення (таблиця 4.21).

Таблиця 4.21 – Формування системи збуту

| № п/п | Специфіка закупівельної поведінки цільових клієнтів | Функції збуту, які має виконувати постачальник товару | Глибина каналу збуту | Оптимальна система збуту |
|---|---|---|---|---|
| 1 | Канал нульового рівня | Продаж | 0(напряму) | Власна |

Останньою складової маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (таблиця 4.22).

Таблиця 4.22 – Концепція маркетингових комунікацій

| № п/ п | Специфіка поведінки цільових клієнтів | Канали комунікацій, якими користуються цільові клієнти | Ключові позиції, обрані для позиціонування | Завдання рекламного повідомлення | Концепція рекламного звернення |
|---|---|---|---|---|---|

| | | | | | |
|---|---|---|---|---|---|
| | Інтеграція API у клієнтській системі | Інтернет | Низька ціна, простота використання, універсальність | Показати переваги рішення над конкурентами, виділити ключові особливості | Створення сайту продукту, розповсюдження інформації про продукт на спеціалізованих ресурсах. |

Було визначено, що придбання продукту буде проводитись через мережу Інтернет або при безпосередньому спілкуванні із представниками компанії. Розповсюдження інформації про продукт буде проводитись виключно через Інтернет, адже аудиторія даного продукту активно користується всесвітньою мережею.

Результатом підрозділу стала ринкова (маркетингова) програма, що включає в себе концепції товару, збуту, просування та попередній аналіз можливостей ціноутворення, спирається на цінності та потреби потенційних клієнтів, конкурентні переваги ідеї, стан та динаміку ринкового середовища, в межах якого впроваджено проект, та відповідну обрану альтернативу ринкової поведінки.

## 4.5 Висновки

В даному розділі проведено підготовчий аналіз для впровадження розробленої системи в якості стартап проекту. Досліджено аналогічні конкурентні системи, встановлено сильні та слабкі сторони системи в

порівнянні з ними. Також було досліджено можливі шляхи розповсюдження продукту та його ймовірну аудиторію, рівень доходів та ймовірну ціну продукту, що розробляється.

Було проведено аналіз потенційних ризиків і можливостей, а також розраховані основні фінансово-економічні показники проекту. Отримані результати кажуть про те, що реалізація проекту є доцільною. Було визначено сильні сторони проекту: зручність у використанні, ціна, якість та широкий функціонал. Серед слабких варто виділити повільне навчання. Варто відмітити можливість реклами продукту на спеціалізованих ресурсах із зазначенням сильних сторін проекту.

# ВИСНОВКИ

У даній роботі було запропоновано створення нової архітектури для мультифакторної автентифікації, що використовує апарат блокчейну для роботи з розподіленими ідентифікаторами, а також розпізнавання облич як другий фактор під час автентифікації.

Під час розв'язання задачі проєктування системи досить важливо розуміти контекст задачі - це умови, які стоять перед такою системою. Ця система має бути дуже безпечною, оскільки надає доступ до персональних даних, а також досить зручною для користувача. Крім цього не варто забувати про зручність і ефективність розгортання системи на інші сервіси. Веб 3.0 аутентифікація має важливе значення оскільки web 3 додатки, швидше за все, не працюватимуть, якщо не буде можливості входу користувачів через блокчейн, отже, для успішної реалізації web 3.0 додатків необхідно мати можливість виробляти web 3.0 аутентифікацію.

Кількість комерційних застосувань технології блокчейн зростає. Багато учасників індустрії використовують приватні дозволені блокчейни. На відміну від публічних блокчейнів, таких як Ethereum, Solana та інші, ці приватні блокчейни доступні виключно за запрошенням. Цей компонент додає рівень автентифікації та авторизації між користувачем та даними блокчейну.

Ідентифікація на основі Web3 дозволяє особам повністю контролювати свою інформацію. Крім того, ідентифікатор Web3 часто є інтернет-адресою, яка використовується для пошуку інформації та даних, що відносяться до однієї особи. Ці дані включають інформацію, яка підтримує такі варіанти використання, як шифрування даних, зв'язок, механізми входу в систему і так далі. Криптографічні докази, такі як цифрові підписи, захищають ці дані.

Підводячи підсумок, ідентифікація Web3 - це, по суті, безпечна і захищена цифрова ідентичність для фізичних осіб, що надає їм більший контроль над своїми персональними даними та інформацією. Це схоже на

традиційні ідентичності, описані в Web2 або навіть Web1. Фундаментальна відмінність полягає в питаннях власності та інтероперабельності.

У цій дисертації було проведено аналіз наявних методів авторизацій за кількома критеріями, як-от зручність для користувача, безпека системи, зручність розгортання на інших проєктах. Також під час аналізів бралася увага на те, до якої категорії автентифікації належить метод, чи то: те, що є в людини, те, чим людина є, те, що людина знає. За допомогою окличного аналізу було визначено, що метод "те що людина знає" є застарілим і незручним для користувача. З методів токен людина є найкраще себе показав метод розпізнавання облич, оскільки він є доволі безпечним та найзручнішим серед тих, що були розглянуті. Також варто зауважити явного лідера в категорії те що у людини є - блокчейн. Цей метод показав себе найбезпечнішим оскільки ґрунтується на блокчейні, що є бій математично підтверджений незмінний незмінний список, а також доволі зручний у розгортанні на інші додатки, оскільки від самого початку був заточений під те, щоб перебувати в розподіленому інтернеті.

У рамках поточного дослідження було розроблено веб-додаток, що демонструє можливості системи мультифакторної автентифікації блокчейна і розпізнавання облич. Працездатність і ефективність цього додатка було перевірено експериментальним шляхом і протестовано на декількох користувачах.

Варто зазначити, що під час розв'язання основної задачі було розв'язано задачу менеджменту профілю користувача в розподіленій мережі web 3.0 на прикладі додатка з безпечною мультифакторною аутентифікацією.

Завдання є актуальним у всіх сферах використання, де є потреба авторизації користувача в систему, тобто розуміння того ким користувач є і чи можна йому довіряти. Обрана архітектура показала себе дуже безпечною, зручною для користувача, а також легко вбудовується в нові системи.

# СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ


1. Decentralized Identifiers [Електронний ресурс] – Режим доступу https://www.w3.org/TR/did-core/
2. Decentralized Identifiers [Електронний ресурс] – Режим доступу https://iherman.github.io/did-talks/talks/2020-Fintech/
3. Чому біометрія дозволить нам назавжди відмовитися від паролів? [Електронний ресурс] – Режим доступу https://worldvision.com.ua/pochemu-biometriya-pozvolit-nam-navsegda-otkazatsya-ot-paroley/
4. Web 3.0 Explained, Plus the History of Web 1.0 and 2.0 [Електронний ресурс] – Режим доступу https://www.investopedia.com/web-20-web-30-5208698
5. What is biometric authentication? [Електронний ресурс] – Режим доступу https://www.techtarget.com/searchsecurity/definition/biometric-authentication
6. HOW DOES AUTHENTICATION/AUTHORIZATION WORK IN WEB3 [Електронний ресурс] – Режим доступу https://www.leewayhertz.com/how-does-authentication-authorization-work-in-web3
7. Authentication vs. Authorization [Електронний ресурс] – Режим доступу https://www.okta.com/identity-101/authentication-vs-authorization
8. What Is a Dapp? Decentralized Apps Explained [Електронний ресурс] – Режим доступу https://www.coindesk.com/learn/what-is-a-dapp-decentralized-apps-explained/
9. What is blockchain technology? [Електронний ресурс] – Режим доступу https://www.ibm.com/topics/what-is-blockchain

10. Blockchain Facts: What Is It, How It Works, and How It Can Be Used [Електронний ресурс] – Режим доступу https://www.investopedia.com/terms/b/blockchain.asp
11. Identification, Authentication, Authorization – What's The Difference [Електронний ресурс] – Режим доступу https://imageware.io/identification-authentication-authorization-difference/
12. Decentralized identity [Електронний ресурс] – Режим доступу https://ethereum.org/en/decentralized-identity/
13. What is Multi-Factor Authentication (MFA) and How Does it Work? [Електронний ресурс] – Режим доступу https://www.onelogin.com/learn/what-is-mfa
14. Two-Factor Authentication (2FA) [Електронний ресурс] – Режим доступу https://duo.com/product/multi-factor-authentication-mfa/two-factor-authentication-2fa
15. Identity and Web3 [Електронний ресурс] – Режим доступу https://auth0.com/blog/identity-and-web3/
16. Cryptocurrency Explained With Pros and Cons for Investment [Електронний ресурс] – Режим доступу https://www.investopedia.com/terms/c/cryptocurrency.asp
17. Advantages and disadvantages of biometrics [Електронний ресурс] – Режим доступу https://www.miteksystems.com/blog/advantages-and-disadvantages-of-biometrics
18. What is decentralized identity in blockchain? [Електронний ресурс] – Режим доступу https://cointelegraph.com/explained/what-is-decentralized-identity-in-blockchain
19. Machine learning and face recognition [Електронний ресурс] – Режим доступу https://www.pxl-vision.com/en/blog/machine-learning-and-how-it-applies-to-facial-recognition-technology

20. Face recognition based on convolutional neural network [Електронний ресурс] – Режим доступу https://ieeexplore.ieee.org/document/8248937
21. Web application [Електронний ресурс] – Режим доступу: https://en.wikipedia.org/wiki/Web_application
22. Основные понятия и особенности клиент-серверной архитектуры [Електронний ресурс] – Режим доступу https://testmatick.com/ru/osnovnye-ponyatiya-i-osobennosti-klient-servernoj-arhitektury/
23. Как работают веб приложения [Електронний ресурс] – Режим доступу https://habr.com/ru/post/450282/

# ДОДАТОК А. ЛІСТИНГ ПРОГРАМНОГО КОДУ

## 1. Файл service.py

```
import cv2
import face_recognition
import os
import numpy as np
from app import constants
import base64
import json
from imutils import paths
import base64
import shutil
from pathlib import Path
import pickle


current_face_encodings = []
current_face_names =[]
frame_resizing = 0.5#1#0.25

def check_existence(id):
    encoding_file_name = id + ".pic"
    filename = os.path.join(constants.PathToEncodings, encoding_file_name)
    my_file = Path(filename)
    return my_file.is_file()

def load_encoding_images(userName):
    encoding_file_name = userName + ".pic"
    filename = os.path.join(constants.PathToEncodings, encoding_file_name)
    current_face_encodings = pickle.loads(open(filename, "rb").read())
    current_face_names = [userName for i in
range(len(current_face_encodings))]
    print("load_encoding_images Encoding images loaded",
len(current_face_encodings))
    return current_face_encodings
```

```
def find_face(userName, frame) -> str:
    current_face_encodings = load_encoding_images(userName)

    frame = base64ToImage(frame)
    small_frame = cv2.resize(frame, (0, 0), fx=frame_resizing,
fy=frame_resizing)
    rgb_small_frame = cv2.cvtColor(small_frame, cv2.COLOR_BGR2RGB)
    face_locations = face_recognition.face_locations(rgb_small_frame)
    face_encodings = face_recognition.face_encodings(rgb_small_frame,
face_locations)
    for face_encoding in face_encodings:
        # See if the face is a match for the known face(s)
        matches = face_recognition.compare_faces(current_face_encodings,
face_encoding)

        if True not in matches:
            return constants.FaceAuthStatus.Invalid

        face_distances =
face_recognition.face_distance(current_face_encodings, face_encoding)
        best_match_index = np.argmin(face_distances)
        if matches[best_match_index]:
            return constants.FaceAuthStatus.Valid

    return constants.FaceAuthStatus.FaceNotFound


def base64ToImage(b64_str):
    im_bytes = base64.b64decode(b64_str)
    im_arr = np.frombuffer(im_bytes, dtype=np.uint8)
    return cv2.imdecode(im_arr, flags=cv2.IMREAD_COLOR)


def save_encodings(userName):

    imagePath =
list(paths.list_images(os.path.join(constants.PathToImages, userName)))
    encoding_file_name = userName + ".pic"
    encodings = []
    for img_path in imagePath:
```

```
            img = cv2.imread(img_path)
            rgb_img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

            try:
                img_encoding = face_recognition.face_encodings(rgb_img)[0]
            except:
                print("Face not found!   ", img_path)
                continue
            encodings.append(img_encoding)
        filename = os.path.join(constants.PathToEncodings, encoding_file_name)
        with open(os.path.join(filename), "wb+") as f:
            f.write(pickle.dumps(encodings))
        print("encoding saved")



def save_faces(userName, images):
    folder = Path(constants.PathToImages + userName)
    print(folder)
    if folder.exists() and folder.is_dir():
            shutil.rmtree(folder)
    os.makedirs(folder)
    for i, image in enumerate(images):
        imgdata = base64.b64decode(image[22:])
        filename = str(i)+'.png'
        with open(os.path.join(folder, filename), "wb+") as fh:
            fh.write(imgdata)
        print("Photo ", i, " added")
```

2. Файл routes.py

```
from flask import render_template, jsonify, request
from app import service
from flask_api import status
import json

from app import app,APP_ROOT

@app.route('/')
def home():
```

```
    return render_template('index.html',title='Home')


@app.route("/load_faces",  methods=['POST'])
def register():
    if request.method == 'POST':
        model = request.get_json()
        service.save_faces(model['name'], model['images'])
        service.save_encodings(model['name'])
        return "Success", 201


@app.route("/check/<id>",  methods=['GET'])
def check(id):
    if request.method == 'GET':
        res = {
            'data' : service.check_existence(id)
            }

        return  jsonify(res)


@app.route("/find_face",  methods=['POST'])
def login():
    if request.method == 'POST':
        model = request.get_json()
        image = model['image']
        userName = model['name']
        (status) = service.find_face(userName, image[22:])
        res = {
            'data' : status
            }
        print(res, type(res))
        return jsonify(res)
```

3. Файл constants.py

```
from pathlib import Path
```

```
class FaceAuthStatus():
    Valid = "Valid"
    Invalid = "Invalid"
    FaceNotFound = "FaceNotFound"

PathToImages = 'data/faces/'

PathToEncodings = 'data/encodings/'
```

4. Файл AuthDialog.vue

```
<!-- eslint-disable vue/no-mutating-props -->
<template>
  <v-dialog
    v-model="dialog"
    eager
    persistent
    max-width="500px"
  >
    <v-card :loading="loading">
      <v-toolbar flat>
        <v-spacer></v-spacer>
        <v-toolbar-title>Login</v-toolbar-title>

        <v-spacer></v-spacer>
        <v-btn
          class="me-1"
          icon
          small
          light
          @click="onClose()"
        >
          <v-icon color="grey darken-2"> mdi-close </v-icon>
        </v-btn>
      </v-toolbar>
      <div v-if="step == 1">
        <blockcain-auth-card @onConnect="walletConnected">
        </blockcain-auth-card>
      </div>
```

```
      <div v-if="step == 2">
        <face-recognition
          @faceRecognized="onFaceRecognized"
          :walletId="walletId"
        >
        </face-recognition>
        <v-snackbar
          top
          right
          text
          :color="snackColor"
          v-model="snackbarShow"
        >
          {{ snackbarText }}

          <template v-slot:action="{ attrs }">
            <v-btn
              class="me-1"
              icon
              small
              light
              v-bind="attrs"
              @click="snackbarShow = false"
            >
              <v-icon color="grey darken-2"> mdi-close </v-icon>
            </v-btn>
          </template>
        </v-snackbar>
      </div>
    </v-card>
  </v-dialog>
</template>

<script>
import FaceRecognitionStatuses from "@/constants/FaceRecognitionStatuses";
import BlockcainAuthCard from "./BlockcainAuthCard.vue";
import FaceRecognition from "./FaceRecognition.vue";

export default {
```

```
name: "AuthDialog",
components: { BlockcainAuthCard, FaceRecognition },

data: () => ({
  step: 1,
  walletId: null,
  loading: false,
  snackbarShow: false,
  snackColor: "primary",
  snackbarText: "",
  dataStore: null,
}),

props: {
  dialog: Boolean,
},

mounted() {},

methods: {
  onClose() {
    this.step = 1;
    this.$emit("close");
  },

  walletConnected(walletId, dataStore) {
    console.log("walletId", walletId);
    this.walletId = walletId;
    this.dataStore = dataStore;
    this.step++;
  },

  onFaceRecognized(recognitionStatus) {
    [this.snackbarText, this.snackColor] =
      this.getSnackbarText(recognitionStatus);
    this.snackbarShow = true;
    if (recognitionStatus == FaceRecognitionStatuses.Valid) {
      this.onClose();
      this.$emit("authFinished", this.walletId, this.dataStore);
```

```
      }
    },

    getSnackbarText(recognitionStatus) {
      console.log(
        "getSnackbarText",
        recognitionStatus,
        FaceRecognitionStatuses.Valid,
        recognitionStatus == FaceRecognitionStatuses.Valid
      );
      switch (recognitionStatus) {
        case FaceRecognitionStatuses.Valid:
          if (this.profile && this.profile.name)
            return [`Welcome back, ${this.profile.name}!`, "success"];
          else return [`Welcome back!`, "success"];
        case FaceRecognitionStatuses.Invalid:
          return ["You are not the right person, please try again!",
"error"];
        case FaceRecognitionStatuses.FaceNotFound:
          return ["Face was not found, please take another photo.",
"warning"];
        default:
          return ["Some error happens, please try again.", "info"];
      }
    },
  },
};
</script>

<style></style>
```

5. Файл BlockcainAuthCard.vue

```
<template>
  <v-card
    class="d-flex flex-column justify-center align-center pb-5"
    flat
    height="250"
  >
    <v-img
      class="ml-n5"
```

```
      width="500"
      contain
      src="@/assets/auth_process.png"
    ></v-img>

    <v-btn
      x-large
      :outlined="!blockchainProfile"
      :loading="loading"
      color="success"
      class="mb-3"
      rounded
      @click="connect()"
    >
      {{ btnText }}
      <v-icon
        v-if="blockchainProfile"
        right
      >
        mdi-check
      </v-icon>
    </v-btn>
  </v-card>
</template>

<script>
import { CeramicClient } from "@ceramicnetwork/http-client";
import { EthereumAuthProvider } from "@ceramicnetwork/blockchain-utils-
linking";
import { DIDDataStore } from "@glazed/did-datastore";
import { DIDSession } from "@glazed/did-session";

export default {
  name: "BlockcainAuthCard",

  data: () => ({
    loading: false,
    ceramic: new CeramicClient("https://ceramic-clay.3boxlabs.com"),
    blockchainProfile: null,
```

```
    aliases: {
      schemas: {
        basicProfile:
          "ceramic://k3y52l7qbv1frxt706gqfzmq6cbqdkptzk8uudaryhlkf6ly9vx21
hqu4r6k1jqio",
      },
      definitions: {
        BasicProfile:
          "kjzl6cwe1jw145cjbeko9kil8g9bxszjhyde21ob8epxuxkaon1izyqsu8wgcic
",
      },
      tiles: {},
    },
    datastore: null,
  }),

  components: {},
  mounted() {
    this.datastore = new DIDDataStore({
      ceramic: this.ceramic,
      model: this.aliases,
    });
  },

  methods: {
    async authenticateWithEthereum(ethereumProvider) {
      const accounts = await ethereumProvider.request({
        method: "eth_requestAccounts",
      });

      const authProvider = new EthereumAuthProvider(
        ethereumProvider,
        accounts[0]
      );
      const session = new DIDSession({ authProvider });
      const did = await session.authorize();
      console.log("my did", did);
      this.ceramic.did = did;
    },
```

```
    async connect() {
      if (window.ethereum == null) {
        console.error("No injected Ethereum provider found");
      }
      try {
        this.loading = true;
        await this.authenticateWithEthereum(window.ethereum);
        await this.getProfileFromCeramic();
        this.loading = false;
      } catch (error) {
        console.error(error);
        this.loading = false;
      }
    },

    async getProfileFromCeramic() {
      try {
        //use the DIDDatastore to get profile data from Ceramic
        console.log("getProfileFromCeramic", this.datastore);
        let did = this.datastore.id;
        did = did.substring(did.lastIndexOf(":") + 1);

        this.blockchainProfile = await this.datastore.get("BasicProfile");
        this.$emit("onConnect", did, this.datastore);
      } catch (error) {
        console.error(error);
      }
    },
  },
  computed: {
    btnText() {
      if (this.blockchainProfile) return "Connected";
      else return "Sign in with the wallet";
    },
  },
};
</script>
<style scoped></style>
```

6. Файл FaceRecognition.vue

```
<template>
  <v-card
    height="580"
    :loading="cameraLoad"
  >
    <v-card class="camera-box">
      <div>
        <video
          ref="camera"
          class="camera"
          autoplay
        ></video>
      </div>
      <v-overlay
        absolute
        opacity="0.55"
        :value="overlay"
      >
        <div
          class="d-flex flex-column justify-center align-center"
          v-if="countDown > 0"
        >
          <div
            class="text-center text-h6"
            style="margin-bottom: 180px"
          >
            Please stay still while we collecting <br />
            and processing your biometry information
          </div>
          <span
            class="text-h1 font-weight-thin"
            style="color: rgba(255, 255, 255, 0.5)"
            >{{ countDown }}</span
          >
        </div>
        <v-progress-circular
```

```
            v-else
            indeterminate
            size="64"
          ></v-progress-circular>
        </v-overlay>
      </v-card>
      <canvas
        v-show="false"
        id="photoTaken"
        ref="canvas"
        class="camera"
        width="640"
        height="480"
      >
      </canvas>

      <div
        class="d-flex flex-column justify-center align-center"
        v-if="shouldRegister"
      >
        <v-alert
          color="warning"
          outlined
          class="mt-4 mx-3"
          type="info"
        >
          Seems you not register two-factor authentification. <br />

          Please be sure that you are the only person in the camera and
press
          <strong>Register</strong> button
        </v-alert>
        <v-btn
          @click="register()"
          x-large
          outlined
          :loading="overlay"
          color="success"
          class="mb-3"
```

```
        rounded
      >
        Register
      </v-btn>
    </div>

    <div
      class="d-flex flex-column justify-center align-center"
      v-else
    >
      <v-alert
        outlined
        class="mt-4 mx-3"
        type="info"
      >
        Please be sure that you are the only person in the camera and
press
        <strong>Login</strong> button
      </v-alert>
      <v-btn
        @click="login()"
        x-large
        outlined
        :loading="loginLoading"
        color="success"
        rounded
      >
        Login
      </v-btn>
    </div>
  </v-card>
</template>

<script>
import axios from "axios";

export default {
  name: "FaceRecognition",
  data: () => ({
```

```
  shouldRegister: false,
  overlay: false,
  loginLoading: false,
  countDown: 15,
  captures: [],
  cameraLoad: false,
}),

props: {
  walletId: String,
},

async mounted() {
  this.cameraLoad = true;
  await this.checkTFARegistration();
  this.setupCamera();
  this.cameraLoad = false;
},

methods: {
  setupCamera() {
    const constraints = (window.constraints = {
      audio: false,
      video: true,
    });

    navigator.mediaDevices
      .getUserMedia(constraints)
      .then((stream) => {
        this.$refs.camera.srcObject = stream;
      })
      .catch((error) => {
        alert(
          "May the browser didn't support or there is some errors.",
          error
        );
      });
  },
```

```
async checkTFARegistration() {
  let path = `http://localhost:5000/check/${this.walletId}`;

  await axios
    .get(path)
    .then((response) => {
      this.shouldRegister = !response.data.data;
      console.log(this.shouldRegister);
    })
    .catch((error) => {
      console.log(error);
    });
},

async register() {
  try {
    this.overlay = true;
    await this.recordSeries();
    await this.checkTFARegistration();
    this.overlay = false;
  } catch (error) {
    console.log(error);
    this.overlay = false;
  }
},

async recordSeries() {
  this.captures = [];
  this.countDown = 15;
  for (this.countDown; this.countDown > 0; this.countDown--) {
    let url = await this.makePhotoWithInterval();
    this.captures.push(url);
  }
  await this.sendBiometryInformation();
},

makePhotoWithInterval() {
  return new Promise((resolve) =>
    setTimeout(() => {
```

```
      resolve(this.getCaptureUrl());
    }, 200)
  );
},

getCaptureUrl() {
  this.$refs.canvas
    .getContext("2d")
    .drawImage(this.$refs.camera, 0, 0, 640, 480);
  return this.$refs.canvas.toDataURL("image/png");
},

async sendBiometryInformation() {
  console.log(this.captures.length);
  let path = `http://localhost:5000/load_faces`;
  let data = {
    images: this.captures,
    name: this.walletId,
  };
  console.log(data);
  await axios.post(path, data);
},

async login() {
  console.log("this.walletId", this.walletId);
  let path = `http://localhost:5000/find_face`;
  let image = this.getCaptureUrl();
  let data = {
    image: image,
    name: this.walletId,
  };
  await axios
    .post(path, data)
    .then((response) => {
      console.log(response.data);
      let recognitionStatus = response.data.data;
      this.$emit("faceRecognized", recognitionStatus);
      console.log(recognitionStatus);
    })
```

```
        .catch((error) => {
          console.log(error);
        });
    },
  },
};
</script>

<style>
.camera {
  width: 100%;
  height: 100%;
}

.camera-box {
  width: 500px;
  height: 375px;
}
</style>
```

7. Файл MainPage.vue

```
<template>
  <v-container style="height: 100%">
    <v-row style="height: 100%">
      <v-col
        class="d-flex justify-center align-center"
        v-if="walletId == null"
        cols="12"
      >
        <v-btn
          x-large
          rounded
          color="success"
          @click="authDialog = true"
        >
          Connect
        </v-btn>
      </v-col>
      <v-col
```

```
        v-else
        cols="12"
      >
        <v-col class="d-flex justify-end">
          <v-btn
            large
            rounded
            text
            @click="walletId = null"
            color="secondary"
          >
            Log out
          </v-btn>
        </v-col>

        <profile-page :datastore="dataStore"> </profile-page>
      </v-col>
    </v-row>
    <auth-dialog
      :dialog="authDialog"
      @close="authDialog = false"
      @authFinished="onAuthFinished"
    >
    </auth-dialog>
  </v-container>
</template>

<script>
import ProfilePage from "./ProfilePage.vue";
import AuthDialog from "./AuthDialog.vue";

export default {
  name: "MainPage",

  data: () => ({
    walletId: null,
    authDialog: false,
    edit: false,
    dataStore: null,
```

```
  }),

  components: {
    ProfilePage,
    AuthDialog,
  },

  methods: {
    onAuthFinished(walletId, dataStore) {
      console.log("onAuthFinished", walletId, dataStore);
      this.walletId = walletId;
      this.dataStore = dataStore;
    },
  },
  computed: {},
};
</script>
<style scoped></style>
```

8. Файл ProfilePage.vue

```
<template>
  <v-form
    ref="form"
    v-model="valid"
    v-if="profileToEdit"
    lazy-validation
    :loading="loading"
  >
    <div class="text-h3 text-center mb-4">Blockchain user profile</div>
    <v-text-field
      v-model="profileToEdit.name"
      prepend-inner-icon="mdi-account"
      label="Name"
      :readonly="!edit"
    ></v-text-field>

    <v-text-field
      v-model="profileToEdit.gender"
```

```
    label="Gender"
    prepend-inner-icon="mdi-gender-male-female-variant"
    :readonly="!edit"
  ></v-text-field>

  <v-textarea
    v-model="profileToEdit.description"
    rows="2"
    prepend-inner-icon="mdi-account-details"
    label="Short description"
    :readonly="!edit"
  ></v-textarea>

  <v-text-field
    v-model="profileToEdit.emoji"
    label="Emoji"
    prepend-inner-icon="mdi-emoticon-cool-outline"
    :readonly="!edit"
  ></v-text-field>

  <v-menu
    ref="menu"
    v-model="menu"
    :close-on-content-click="false"
    transition="scale-transition"
    offset-y
    min-width="auto"
    :disabled="!edit"
  >
    <template v-slot:activator="{ on, attrs }">
      <v-text-field
        :value="profileToEdit.birthDate"
        label="Date Of Birth"
        prepend-inner-icon="mdi-calendar"
        readonly
        v-bind="attrs"
        v-on="on"
      ></v-text-field>
    </template>
```

```
    <v-date-picker
      ref="picker"
      no-title
      v-model="profileToEdit.birthDate"
      :max="new Date().toISOString().substr(0, 10)"
      min="1900-01-01"
      @change="saveDOB"
    ></v-date-picker>
  </v-menu>

  <div v-if="edit == true">
    <v-btn
      color="danger"
      outlined
      rounded
      large
      class="me-3"
      @click="cancel()"
    >
      Cancel
    </v-btn>

    <v-btn
      color="success"
      outlined
      rounded
      large
      @click="update()"
    >
      Update
    </v-btn>
  </div>
  <div v-else>
    <v-btn
      color="warning"
      outlined
      rounded
      large
      @click="edit = true"
```

```
      >
        Edit profile
      </v-btn>
    </div>
  </v-form>
</template>

<script>
import BasicProfile from "@datamodels/identity-profile-basic";
import { DIDDataStore } from "@glazed/did-datastore";

export default {
  name: "ProfilePage",

  data: () => ({
    valid: true,
    profileToEdit: null,
    loading: false,
    edit: false,
    menu: false,
  }),

  props: {
    profile: BasicProfile,
    datastore: DIDDataStore,
  },

  async mounted() {
    this.profileToEdit = await this.datastore.get("BasicProfile");
  },

  watch: {
    profile(newVal) {
      console.log("watch--profile", newVal);
      this.profileToEdit = newVal;
    },
  },
  methods: {
    validate() {
```

```
      return this.$refs.form.validate();
    },

    saveDOB(date) {
      this.$refs.menu.save(date);
    },

    cancel() {
      console.log(this.profileToEdit, this.profile);
      this.profileToEdit = this.profile;
      this.edit = false;
    },

    async update() {
      this.edit = false;
      await this.onSubmit();
    },

    async onSubmit() {
      try {
        this.loading = true;
        //use the DIDDatastore to merge profile data to Ceramic
        await this.datastore.merge("BasicProfile", this.profileToEdit);

        //use the DIDDatastore to get profile data from Ceramic
        this.profileToEdit = await this.datastore.get("BasicProfile");

        this.loading = false;
      } catch (error) {
        this.loading = false;
        console.error(error);
      }
    },
  },
  computed: {},
};
</script>
<style scoped></style>
```