МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет
імені Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

ДОПУЩЕНО ДО ЗАХИСТУ
Завідувач кафедри інтелектуальних
інформаційних систем, д-р техн. наук, проф.
_____Ю. П. Кондратенко
«____»_____2024 р.

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

ВЕБПЛАТФОРМА ДЛЯ ТОРГІВЛІ ТОВАРАМИ ПОДВІЙНОГО ПРИЗНАЧЕННЯ З KYC-ВЕРИФІКАЦІЄЮ

Спеціальність 122 «Комп'ютерні науки»

122 – КРБ – 401.22010113

Виконав студент 4-го курсу, групи 401
_____ В. М. Лялюк
«18» червня 2024 р.

Керівник: ст.викладач
_____І. С. Бурлаченко
«18» червня 2024 р.

Миколаїв – 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет ім. Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

Рівень вищої освіти **бакалавр**
Спеціальність **122 «Комп'ютерні науки»**
(шифр і назва)
Галузь знань **12 «Інформаційні технології»**
(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри інтелектуальних
інформаційних систем, д-р техн. наук, проф.
_____ Ю. П. Кондратенко
«____» _____2024 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи

Видано студенту групи 401 факультету комп'ютерних наук Лялюку Володимиру Миколайовичу.

1. Тема кваліфікаційної роботи «Вебплатформа для торгівлі товарами подвійного призначення з KYC-верифікацією».

Керівник роботи Бурлаченко Іван Сергійович, старший викладач кафедри КІ.

Затв. наказом Ректора ЧНУ ім. Петра Могили від «28» грудня 2023 р. № 271

2. Строк представлення кваліфікаційної роботи студентом «18» червня 2024 р.

3. Вхідні (початкові) дані до роботи: дані, що надає користувач, моделі для роботи алгоритму

Очікуваний результат: вебплатформа для торгівлі товарами подвійного призначення з KYC-верифікацією.

4. Перелік питань, що підлягають розробці (зміст пояснювальної записки):

- постановка задачі. Огляд основних технологій;
- огляд можливостей бібліотеки TensorFlow;
- планування проекту. Проектування бази даних;
- реалізація.

5. Перелік графічного матеріалу: презентація.

6. Завдання до спеціальної частини: «Вимоги до приміщення з комп’ютерним обладнанням. Перевірочний розрахунок природного освітлення для приміщення»

7. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис
Спеціальна частина з охорони праці	Алексєєва А. О., доцент кафедри екології	

Керівник роботи __старший викладач Бурлаченко І. С._____
(наук. ступінь, вчене звання, прізвище та ініціали)

_____
(підпис)

Завдання прийнято до виконання ___Люляк В. М._____
(прізвище та ініціали)

_____
(підпис)

Дата видачі завдання «_14_» _____січня____ 2024 р.

КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної роботи

Тема: Вебплатформа для торгівлі товарами подвійного призначення з KYC-верифікацією _____

№	Найменування роботи	Початок	Закінчення	Примітки
1	Подання заяви на затвердження теми та керівників КРБ	10.11.2023	15.11.2023	
2	Отримання завдання на виконання КРБ	10.01.2024	15.01.2024	
3	Складання календарного плану роботи на весь період виконання КРБ	16.01.2024	30.01.2024	
4	Отримання завдання на переддипломну практику	15.04.2024	29.04.2024	
5	Проходження переддипломної практики, збір та аналіз матеріалів до КРБ	29.04.2024	11.05.2024	
6	Розробка звіту з переддипломної практики	12.05.2024	15.05.2024	
7	Виконання КРБ: аналіз сучасного стану вебплатформ із інтеграцією KYC, огляд існуючих технологій, розробка ПЗ	13.05.2024	22.06.2024	
8	Перший попередній захист КРБ на засіданні комісії кафедри	27.05.2024	27.05.2024	
9	Доробка та остаточне оформлення КРБ	28.05.2024	09.06.2024	
10	Другий попередній захист КРБ на засіданні комісії кафедри	10.06.2024	10.06.2024	
11	Подання КРБ рецензенту	13.06.2024	13.06.2024	
11	Подання КРБ, її електронної копії та інших документів (відгуку, рецензії) до захисту	18.06.2024	21.06.2024	
12	Захист КРБ перед екзаменаційною комісією (ЕК)	25.06.2024	28.06.2024	

Розробив студент __Лялюк В. М._____ _____
(прізвище, ім'я, по батькові студента) (*підпис*)

Керівник роботи __старший викладач Бурлаченко І. С._____
(*посада, прізвище, ім'я, по батькові)* *(підпис)*

«_29_» _____01_____ 2024 р.

АНОТАЦІЯ

кваліфікаційної роботи студента групи 401 ЧНУ ім. Петра Могили
Лялюка Володимира Миколайовича
Тема: «Вебплатформа для торгівлі товарами подвійного призначення з KYC-верифікацією»

Актуальність інтеграції KYC систем в платформу торгівлі товарами подвійного призначення полягає визначенні та ідентифікації осіб, які можуть бути включені до чорних списків.

Предметом кваліфікаційної роботи бакалавра є KYC верифікація користувачів вебплатформа для торгівлі товарами подвійного призначення.

Об’єктом кваліфікаційної роботи бакалавра є процес роботи вебплатформи, що включає ідентифікацію та верифікацію особистості користувачів за допомогою методів штучного інтелекту та аналізу даних.

Метою роботи є забезпечення важливого етапу безпекового шару вебплатформи, мінімізуючи ризик вступу в комерційні відносини з небажаними або ризикованими контрагентами, що може призвести до небажаних наслідків для платформи та її користувачів.

Пояснювальна записка складається зі вступу, чотирьох розділів, висновків та додатків.

У першому розділі розглядаються існуючі аналоги вебзастосунків із KYC-верифікацією і з’ясовано сучасний стан проблеми та основні методи її вирішення.

У другому розділі розібрано роботу бібліотеки Tensorflow.js і обрано спосіб її використання.

У третьому розділі детально розглянуто архітектуру та функціональність клієнту та серверу системи Аутентифікації і KYC-верифікакції, включаючи інтеграцію з сервісами для обробки файлів та розпізнавання облич. Ключові аспекти включають розробку інтерактивних модулів роботи, які забезпечують динамічну взаємодію з користувачами та ефективне управління контентом.

У четвертому розділі описано програмну реалізацію всі заплановані задачі, підключені необхідні сервіси і протестовано роботу застосунку.

Кваліфікаційна робота містить 15 рисунків, 15 використаних джерел та презинтацію.

Ключові слова: вебзастосунок, KYC-верифікація, безпека.

ABSTRACT
of the bachelor thesis by a student of group 401,

Petro Mohyla Black Sea National University

Volodymyr Lialiuk Mykolaiovych

Title: «Web Platform for Trading Dual-Use Goods with KYC Verification»

The relevance of integrating KYC systems into a platform for trading dual-use goods lies in identifying and verifying individuals who may be included on blacklists.

The subject of the bachelor's thesis is KYC verification of users on a web platform for trading dual-use goods.

The object of the bachelor's thesis is the operational process of the web platform, which includes identifying and verifying user identities using artificial intelligence methods and data analysis.

The aim of the thesis is to provide an important security layer for the web platform, minimizing the risk of entering into commercial relationships with undesirable or risky counterparts, which could lead to adverse consequences for both the platform and its users.

The explanatory note consists of an introduction, four sections, conclusions, and appendices.

The first section reviews existing web applications with KYC verification, explores the current state of the problem, and identifies main methods for its resolution.

The second section discusses the operation of the Tensorflow.js library and selects a method for its use.

The third section thoroughly examines the architecture and functionality of the frontend and backend of the Authentication and KYC verification system, including integration with services for file processing and facial recognition. Key aspects include the development of interactive modules that provide dynamic user interaction and effective content management.

The fourth section describes the software implementation of all planned tasks, connected necessary services, and tested the application's operation.

The web platform for trading dual-use goods with KYC verification includes an interactive assistant capable of providing personalized recommendations and support.

The qualification thesis contains 15 figures, 15 sources used and presentation. Key words: web application, KYC-verification, security.

ЗМІСТ

ВСТУП

У сучасному світі торгівля товарами подвійного призначення –це не тільки комерційна діяльність, але й область, що вимагає строгого регулювання та високого рівня безпеки. Товари подвійного призначення, такі як технології, що можуть бути використані як у цивільних, так і військових аплікаціях, потребують особливої уваги з боку регуляторів і бізнесу. Це ставить підвищені вимоги до механізмів перевірки та валідації угод, особливо в контексті міжнародних санкцій та експортного контролю.

Вебплатформа для торгівлі товарами подвійного призначення з KYC (Know Your Customer)-верифікацією – це інноваційне рішення, яке використовує сучасні технології для забезпечення прозорості, законності та безпеки торгових операцій. Основною її функцією є ідентифікація та перевірка суб'єктів торгівлі з метою попередження ризиків, пов'язаних з фінансовими шахрайствами, відмиванням грошей, а також порушеннями регулятивних норм.

Актуальність інтеграції KYC систем в платформу торгівлі товарами подвійного призначення полягає визначенні та ідентифікації осіб, які можуть бути включені до чорних списків. Завдяки детальній перевірці особистих даних та їх порівнянню з такими базами даних як «Миротворець», система KYC дозволяє швидко виявляти осіб, які підпадають під санкції або мають обмеження на участь у торгівлі певними категоріями товарів.

Платформа використовує передові алгоритми для збору, аналізу та зберігання даних про користувачів. Використання біометричних даних, документів, що підтверджують особу, та інших форм ідентифікації забезпечує надійну верифікацію осіб, які беруть участь у торгівлі. Платформа постійно синхронізується з глобальними базами даних, що містять інформацію про санкції, експортні обмеження, а також чорні списки, забезпечуючи дотримання всіх відповідних міжнародних та національних правил.

Додаткова перевірка особистості дозволяє покращити процеси перевірки угод на предмет їх відповідності регулятивним вимогам. Це значно підвищує швидкість і ефективність обробки транзакцій. Високий рівень захисту даних забезпечується завдяки сучасним криптографічним методам та стратегіям кібербезпеки, які захищають платформу від несанкціонованого доступу та кібератак.

Платформа надає розширені інструменти для генерації звітів та аналітики, що дозволяє користувачам отримувати детальну інформацію про угоди, аудиторські сліди та історію транзакцій. Ця вебплатформа відіграє ключову роль у глобальній торгівлі, знижуючи ризики та підвищуючи довіру до ринку товарами подвійного призначення. Її розвиток і удосконалення можуть стати значним кроком уперед у створенні більш безпечного та регульованого торгового середовища.

Метою роботи є забезпечення важливого етапу безпекового шару вебплатформи, мінімізуючи ризик вступу в комерційні відносини з небажаними або ризикованими контрагентами, що може призвести до небажаних наслідків для платформи та її користувачів.

Предметом кваліфікаційної роботи бакалавра є KYC верифікація користувачів вебплатформа для торгівлі товарами подвійного призначення.

Об’єктом кваліфікаційної роботи бакалавра є процес роботи вебплатформи, що включає ідентифікацію та верифікацію особистості користувачів за допомогою методів штучного інтелекту та аналізу даних.

1 ОГЛЯД ОСНОВНИХ ТЕХНОЛОГІЙ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Основні поняття і визначення

Know Your Customer (KYC) або "Знай свого клієнта" – це процес, який використовують фінансові установи, такі як банки, інвестиційні компанії та фінансові технологічні компанії (фінтех), для ідентифікації та перевірки особи своїх клієнтів. Ця процедура стала важливим компонентом сучасної фінансової системи і є частиною заходів, спрямованих на забезпечення законності та прозорості у фінансових операціях.

Основні компоненти KYC включають ідентифікацію клієнта, перевірку та моніторинг. Під час ідентифікації фінансова установа збирає від клієнта базову особисту інформацію, таку як ім'я, адреса, дата народження та документи, що підтверджують особу. Найчастіше для цього використовують такі документи, як паспорт, водійське посвідчення або інші офіційні документи, які можуть підтвердити особу клієнта. На стадії перевірки компанія перевіряє, чи відповідають надані документи реальній особі клієнта. Це може включати порівняння фото з документа з особою клієнта або використання сторонніх служб для перевірки автентичності документів. Додатково, компанії можуть перевіряти фінансовий профіль клієнта, його кредитну історію, а також наявність у нього будь-яких правових або фінансових обмежень. Моніторинг включає відстеження діяльності клієнта та періодичне оновлення інформації про клієнтів, особливо якщо їх обставини змінюються.

KYC є важливим для боротьби з відмиванням грошей, запобігання фінансовому шахрайству, забезпечення відповідності нормативним вимогам та захисту репутації фінансових установ. Однією з головних причин застосування KYC є запобігання відмиванню грошей, оскільки процес дозволяє виявляти та відстежувати підозрілі фінансові операції. KYC також допомагає запобігти фінансовому шахрайству, забезпечуючи, що особа, яка здійснює операції, є тим, за

кого себе видає. Це знижує ризик шахрайських транзакцій. Багато країн мають законодавчі вимоги щодо ідентифікації клієнтів для фінансових установ. KYC забезпечує відповідність цим вимогам, що допомагає уникнути юридичних проблем та штрафів. Проведення належного KYC також допомагає фінансовим установам захистити свою репутацію, оскільки вони знижують ризик бути пов'язаними з нелегальною діяльністю.

AML (Anti-Money Laundering) – це комплекс заходів та процедур, спрямованих на запобігання відмиванню грошей, отриманих незаконним шляхом. AML має на меті виявлення, попередження та розслідування злочинних фінансових операцій. Цей процес включає в себе різні кроки, такі як ідентифікація та верифікація клієнтів, моніторинг транзакцій, виявлення підозрілої діяльності, а також співпрацю з правоохоронними органами.

TBML (Trade-Based Money Laundering) – це тип відмивання грошей, при якому фінансові злочинці використовують торговельні операції для приховування та переміщення незаконно отриманих коштів. TBML передбачає використання міжнародної торгівлі як інструменту для переведення незаконних коштів з однієї країни в іншу, маскуючи ці транзакції як легітимні торговельні операції.

Онлайн-платформа – це цифровий інтерфейс або вебсайт, який надає різноманітні послуги або функції через інтернет. Вона слугує як основа для взаємодії між користувачами або між користувачами та компанією, забезпечуючи доступ до певних продуктів, інформації чи спільноти. Онлайн-платформи можуть бути орієнтовані на різні сфери діяльності, такі як електронна комерція, соціальні мережі, освіта, фінансові послуги, розваги або комунікації. Ці платформи часто використовуються для полегшення транзакцій, обміну інформацією, спільного використання ресурсів або взаємодії в реальному часі між користувачами, і вони грають ключову роль у сучасній цифровій економіці.

Верифікація – це процес підтвердження або перевірки достовірності та точності певної інформації, особи або системи. У фінансовому та технологічному контекстах, верифікація часто стосується перевірки особистих даних або

документів клієнта, щоб переконатися, що вони є справжніми та відповідають реальності. Цей процес допомагає запобігти шахрайству, забезпечує відповідність нормативним вимогам і гарантує, що лише авторизовані особи отримують доступ до певних ресурсів або послуг.

1.2 Існуючі платформи з інтеграцією KYC

Підхід використання KYC (Know Your Customer) для вебплатформ, які займаються торгівлею товарами подвійного призначення, не є поширеним, особливо на місцевому ринку в Україні. Мій застосунок пропонує новаторське рішення для локальних сервісів, що вирішує потребу в підвищеній регуляторній відповідності та перевірці клієнтів у торгівлі такими чутливими товарами.

Один з помітних прикладів такого підходу можна побачити в діяльності Executive Office for Control and Non-Proliferation (EOCN) в OAE. EOCN надає онлайн-сервіс для стратегічних товарів, який включає товари подвійного призначення, технології та програмне забезпечення, що можуть бути використані як у цивільних, так і у військових цілях. Їх платформа сприяє імпорту, експорту та реекспорту таких товарів через систему дозволів, яка пов'язана з різними питаннями безпеки, такими як нерозповсюдження та регіональна стабільність.

Торгівля товарами подвійного призначення часто вимагає складних регуляторних вимог, оскільки ці предмети можуть бути використані як для легітимних цивільних, так і для потенційно шкідливих військових цілей. Традиційний підхід до торгівлі такими товарами покладався на суворі заходи експортного контролю, але не завжди інтегрував процедури KYC так само, як фінансові послуги чи інші галузі. Це створило прогалину у забезпеченні належної перевірки суб'єктів, які займаються такою торгівлею, підвищуючи ризики зловживань або відхилень чутливих товарів.

Впроваджуючи KYC у цю сферу, мій застосунок прагне встановити новий стандарт для локальних сервісів, забезпечуючи, щоб усі клієнти були належним чином ідентифіковані та їхній фон перевірений перед будь-якими транзакціями, що

стосуються товарів подвійного призначення. Такий підхід не лише відповідає міжнародним нормативним очікуванням, але й покращує безпеку та довіру до процесу торгівлі, допомагаючи запобігти зловживанням або небажаному розповсюдженню чутливих технологій.

Платформа EOCN демонструє, як така система може бути реалізована ефективно. Процес KYC на їх платформі включає реєстрацію, активацію облікового запису та подання різних документів для перевірки особи та легітимності суб'єктів, що займаються стратегічними товарами. Цей приклад демонструє, як інтеграція процедур KYC у процес торгівлі може покращити відповідність і запобігти незаконній діяльності.

Крім того, такі платформи, як AKS iQ, пропонують рішення, засновані на штучному інтелекті, для забезпечення відповідності в торгівлі, включаючи KYC, протидію відмиванню грошей (AML) та протидію відмиванню грошей у торгівлі (TBML). Ці платформи демонструють, що зростає потреба у всебічній перевірці особистості та моніторингу транзакцій у торгівлі, особливо коли мова йде про чутливі предмети, такі як товари подвійного призначення.

Отже, впровадження KYC для платформ торгівлі товарами подвійного призначення є інноваційним підходом, який пропонує значні переваги з точки зору регуляторної відповідності, безпеки та довіри до процесу торгівлі. Використовуючи ці знання та застосовуючи їх на місцевому ринку в Україні, мій застосунок виступає як піонерське рішення, яке вирішує важливу прогалину на ринку.

1.3 Переваги і недоліки розробки вебплатформи для торгівлі товарами подвійного призначення з KYC

Переваги розробки вебплатформи для торгівлі товарами подвійного призначення з KYC:

– впровадження KYC допомагає дотримуватися законів і нормативних актів, які стосуються торгівлі товарами подвійного призначення, забезпечуючи відсутність торгівлі з підозрілими або нелегальними суб'єктами;

– це важливо для уникнення штрафів та інших санкцій від урядів і міжнародних організацій;

– KYC сприяє прозорості і забезпечує захист від шахрайства, що підвищує довіру клієнтів та партнерів до платформи;

– це може сприяти розвитку бізнесу та залученню більшої кількості користувачів і партнерів;

– коли платформа торгує товарами, які можуть бути використані для військових або інших небезпечних цілей, KYC допомагає запобігати фінансуванню тероризму та відмиванню грошей, що є ключовими проблемами національної та глобальної безпеки.

Недоліки розробки вебплатформи для торгівлі товарами подвійного призначення з KYC:

– впровадження KYC потребує додаткових фінансових витрат на розробку системи, збір та обробку даних клієнтів, а також на навчання персоналу та підтримку процесу. Це може бути особливо обтяжливим для малих або нових підприємств;

– KYC вимагає від користувачів надання додаткової інформації та проходження перевірки, що може відлякати потенційних клієнтів або зробити процес покупки менш зручним;

– деякі користувачі можуть не бажати надавати свої особисті дані або проходити додаткові перевірки;

– збір та зберігання персональних даних клієнтів збільшує ризик витоку або крадіжки інформації, що може призвести до втрати довіри та юридичних проблем для компанії.

Це особливо актуально для вебплатформ, які оперують чутливою інформацією.

Розробка вебплатформи для торгівлі товарами подвійного призначення з KYC має свої плюси і мінуси. З одного боку, така платформа забезпечує відповідність нормативним вимогам, підвищує довіру з боку клієнтів і партнерів та запобігає відмиванню грошей і фінансуванню тероризму. Ці фактори можуть зробити бізнес більш надійним та успішним, особливо у складних галузях торгівлі чутливими товарами.

Однак, необхідність додаткових витрат на впровадження та підтримку системи KYC, а також потенційні ускладнення користувацького досвіду можуть бути суттєвими перешкодами. Особливо це стосується малого бізнесу або нових підприємств, для яких додаткові фінансові та операційні витрати можуть стати вирішальними факторами.

1.4 Спосіб імплементації KYC для вебплатформи

Вебзастосунок буде складатися із трьох основних компонентів: серверного застосунку, клієнту та клієнту для адміністративної панелі. Включатиме функціональність для процесу "Know Your Customer" (KYC), що передбачає поетапну перевірку користувачів на основі їхньої суми покупок [1]. Планується, що застосунок дозволятиме користувачам купувати речі з реєстрацією, але без KYC, якщо сума їхніх покупок не перевищує 10 тисяч гривень. Однак, якщо сума покупок стає більшою за 10 тисяч гривень, користувач повинен зареєструватися та пройти перший етап KYC. Якщо сума покупок сягає 50 тисяч гривень, користувач повинен пройти другий етап KYC. Нарешті, якщо сума покупок перевищує 100 тисяч гривень, користувач повинен пройти третій етап KYC. Кожен із цих етапів передбачає введення користувачем певних особистих даних, які потім мають бути переглянуті та затверджені адміністратором через адміністративну панель.

Для реалізації цієї системи обрано такі технології:

– для серверного застосунку використовується фреймворк NestJS, оскільки він пропонує масштабовану та ефективну структуру для розробки серверного коду на TypeScript;

– для роботи з базою даних ви використовуєте Prisma, що забезпечує зручний інтерфейс для взаємодії з PostgreSQL, яку обрано як СУБД;

– для клієнту ви використано React, оскільки він забезпечує інтерактивний та динамічний досвід для користувачів та забезпечує високу швидкість розробки;

– адміністративна панель також буде розроблена на React, щоб забезпечити зручний та інтуїтивно зрозумілий інтерфейс для адміністраторів.

Процес взаємодії у системі передбачає, що користувачі можуть купувати товари без реєстрації, якщо сума їхніх покупок не перевищує 10 тисяч гривень. У цьому випадку вони можуть здійснити покупку без введення жодної особистої інформації. Коли ж сума покупок користувача перевищує 10 тисяч гривень, система автоматично вимагає від нього зареєструватися та пройти перший етап KYC. Цей етап може включати базову інформацію, таку як ім'я, адреса, та підтвердження електронної пошти.

Після введення цієї інформації, вона передається на сервер, де зберігається у базі даних PostgreSQL через Prisma, а потім відображається в адміністративній панелі для розгляду адміністратором. Адміністратор переглядає інформацію через панель на React та приймає рішення про затвердження або відхилення. Це рішення передається назад на сервер через NestJS, де оновлюється статус користувача. Коли сума покупок користувача перевищує 50 тисяч гривень, він повинен пройти другий етап KYC, який може включати надання більш детальної інформації, такої як документ, що посвідчує особу. Аналогічно, ця інформація передається на сервер зберігається у базі даних, і передається в адміністративну панель для розгляду адміністратором. Коли сума покупок користувача перевищує 100 тисяч гривень, він повинен пройти третій етап KYC, який може включати додаткові підтвердження або документи. Процедура проходження та перегляду цього етапу аналогічна попереднім етапам. Таким чином, ваша система забезпечує ефективний та безпечний процес KYC, який дозволяє користувачам здійснювати покупки, відповідаючи нормативним вимогам. Використання NestJS, Prisma, PostgreSQL та

React забезпечує сучасну та надійну технологічну платформу для реалізації цієї функціональності.

Цей вебзастосунок демонструє гнучкий та адаптивний підхід до управління процесом KYC, дозволяючи користувачам здійснювати покупки, не наражаючи їх на надмірні перевірки, доки це не стане необхідним. Така стратегія забезпечує не тільки кращий користувацький досвід, але й дотримання нормативних вимог, що є надзвичайно важливим для захисту інтересів як компанії, так і її клієнтів. Інтегруючи реєстрацію та перевірку особистості в залежності від сум покупок, система автоматично пристосовується до різних рівнів ризику та фінансових операцій.

Адміністративна панель, створена за допомогою React, надає адміністраторам зручний інтерфейс для перегляду та затвердження KYC-форм користувачів. Це дозволяє адміністраторам ефективно управляти процесом перевірки, забезпечуючи швидку ідентифікацію потенційних ризиків та реагування на них. Водночас, використання NestJS та Prisma на сервері дозволяє забезпечити безперебійний і безпечний потік даних між різними компонентами системи, а PostgreSQL служить надійним сховищем для збереження всієї необхідної інформації.

1.5 Опис основних використаних технологій

NestJS – це прогресивний фреймворк для розробки серверних додатків Node.js, який використовує TypeScript як основну мову програмування. Він створений з акцентом на продуктивність, масштабованість і підтримку модульної архітектури. NestJS[2] базується на експресивному синтаксисі і використовує принципи об'єктно-орієнтованого програмування, функціонального програмування та реактивного програмування.

Фреймворк побудований на основі Express або Fastify, і дозволяє розробникам створювати ефективні та масштабовані серверні додатки з мінімальним зусиллям. NestJS також має вбудовану підтримку декількох бібліотек,

таких як TypeORM, Mongoose, Passport, та інших, що полегшує інтеграцію з різними базами даних та идентификационными системами.

Однією з ключових особливостей NestJS є його модульна структура, яка дозволяє легко розділяти функціональність додатка на окремі, незалежні модулі. Це сприяє кращій організації коду та полегшує підтримку та розширення додатка в майбутньому .

PostgreSQL – це потужна, відкрита реляційна система управління базами даних (RDBMS), яка наголошує на стандарти SQL та багатому наборі функцій. Вона відома своєю стабільністю, надійністю та повноцінною підтримкою транзакцій. PostgreSQL відома як "Postgres" і забезпечує високу продуктивність та масштабованість, що робить її підходящою для застосувань будь-якого розміру, від невеликих до великих корпоративних систем.

Ключові особливості PostgreSQL:

– PostgreSQL підтримує значну частину стандарту SQL та розширює його додатковими функціями, включаючи розширені типи даних, функції та індекси;

– система дозволяє користувачам створювати власні функції, типи даних, оператори та інші елементи, що робить її дуже гнучкою та придатною для широкого спектру застосувань;

– PostgreSQL забезпечує повну підтримку транзакцій із властивостями атомарності, узгодженості, ізольованості та стійкості (ACID), що є важливим для надійної роботи бази даних;

– PostgreSQL пропонує кілька видів індексів, включаючи B-Tree, Hash, GiST, GIN та інші, що підвищує продуктивність запитів та пошуку.

Prisma – це сучасний ORM (Object-Relational Mapping) інструмент для TypeScript та Node.js, який дозволяє розробникам працювати з базами даних більш ефективно та зручно. Prisma спрямований на покращення продуктивності розробки шляхом абстрагування та автоматизації взаємодії з базами даних.

Основні переваги Prisma включають:

– Prisma автоматично генерує типи TypeScript на основі вашої бази даних, що допомагає уникнути помилок та підвищує безпеку типів у коді;

– Prisma спрощує роботу з різними базами даних, такими як PostgreSQL, MySQL, SQLite та MongoDB;

– Prisma має інтуїтивний інтерфейс, зрозумілий навіть для новачків у ORM;

– Prisma дозволяє налаштовувати та розширювати схему бази даних за допомогою Prisma Schema Language, що дозволяє легко адаптуватися до змін у вимогах.

Prisma складається з трьох основних частин:

– автоматично генерується клієнт, який дозволяє взаємодіяти з базою даних за допомогою простих у використанні методів;

– інструмент для управління міграціями бази даних, який допомагає контролювати та відстежувати зміни у схемі;

– інтерфейс користувача для роботи з даними в базі даних, який полегшує перегляд і маніпулювання даними.

React, також відомий як ReactJS, – це популярна JavaScript-бібліотека для створення інтерфейсів користувача. Вона була розроблена та підтримується Facebook, а також використовується багатьма провідними технологічними компаніями. React дозволяє розробникам створювати інтерактивні та динамічні вебдодатки з простотою та ефективністю.

Основні особливості React:

– React заснований на концепції компонентів, які є автономними та багаторазовими блоками коду[3], що відповідають за певну частину інтерфейсу користувача;

– цей підхід полегшує створення та управління складними інтерфейсами, розбиваючи їх на менші, більш керовані частини;

– React використовує віртуальний DOM (Document Object Model) для підвищення продуктивності;

– коли стан компонент змінюється, React порівнює нову версію віртуального DOM з попередньою та оновлює тільки ті частини реального DOM, які змінилися;

– це робить оновлення інтерфейсу швидшими та ефективнішими;

– у React дані передаються в одному напрямку – зверху вниз;

– це спрощує відстеження потоку даних та усунення помилок, оскільки зміни в даних завжди йдуть від батьківських компонентів до дочірніх;

– React використовує JSX, синтаксичне розширення JavaScript, яке дозволяє писати компоненти за допомогою розмітки, схожої на HTML;

– це робить код більш читабельним та зрозумілим, особливо для розробників, які знайомі з HTML.

Переваги React:

– React має простий та зрозумілий API, що робить його доступним для розробників з різним рівнем досвіду;

– його компонентна архітектура та підтримка JSX надають велику гнучкість у проектуванні та реалізації інтерфейсів користувача;

– React має велику та активну спільноту, а також багатий набір бібліотек та інструментів, які підтримують його та розширюють можливості;

– це включає популярні бібліотеки для управління станом, такі як Zustand, Redux та MobX, та інструменти для тестування, такі як Jest та Enzyme;

– завдяки використанню віртуального DOM та ефективному управлінню компонентами, React забезпечує високу продуктивність, що робить його підходящим для створення швидких та чуйних вебдодатків;

– React легко інтегрується з іншими фреймворками та бібліотеками, що робить його гнучким для використання в різних контекстах;

– наприклад, він може бути використаний разом з Redux для управління станом, з React Router для керування навігацією або з Styled Components для стилізації компонентів;

– React Native, побудований на базі React, дозволяє розробляти крос-платформні мобільні додатки, які працюють як на iOS, так і на Android.

Tailwind CSS – це утилітарний CSS-фреймворк для створення користувацьких інтерфейсів [4]. Він надає набір готових класів, які дозволяють розробникам швидко та ефективно стилізувати елементи без необхідності писати власний CSS-код. Tailwind CSS розроблений таким чином, щоб надати максимальну гнучкість та мінімізувати час, витрачений на створення стилів.

Основні особливості Tailwind CSS:

– Tailwind використовує утилітарні класи, які є атомарними та виконують конкретні стилістичні завдання, такі як розмір, колір, відступи тощо;

– це дозволяє легко комбінувати класи для досягнення бажаного зовнішнього вигляду, що значно покращує швидкість розробки та зменшує складність CSS-коду;

– Tailwind дозволяє легко налаштовувати та розширювати стандартні стилі за допомогою конфігураційного файлу;

– це означає, що розробники можуть адаптувати дизайн до власних потреб без необхідності писати новий CSS;

– завдяки використанню атомарних класів, Tailwind сприяє мінімалістичному підходу до стилізації, що допомагає уникнути дублювання та зменшити розмір CSS-коду;

– це також робить компоненти більш багаторазовими та легкими для підтримки;

– Tailwind підтримує Just-In-Time (JIT) компіляцію, яка генерує лише ті класи, які фактично використовуються в проекті;

– це значно підвищує продуктивність та зменшує розмір фінального CSS-файлу.

Переваги Tailwind CSS:

– Tailwind дозволяє розробникам швидко стилізувати інтерфейс, використовуючи вже готові класи, замість написання нового CSS;

– Tailwind забезпечує послідовність у стилізації завдяки використанню попередньо визначених класів та чітко визначених масштабів для відступів, розмірів та інших властивостей;

– утилітарний підхід Tailwind сприяє масштабованості та підтримці великих проектів, оскільки код стає більш передбачуваним та керованим.

Axios – це популярна JavaScript-бібліотека, яка використовується для здійснення HTTP-запитів. Вона побудована на основі нативної функціональності браузера для роботи з мережевими запитами[5] і пропонує простий і ефективний інтерфейс для відправки та отримання даних від сервера.

Основні особливості Axios:

– Axios забезпечує інтуїтивно зрозумілий інтерфейс для здійснення HTTP-запитів;

– використання бібліотеки є досить простим, оскільки запити можуть бути здійснені через методи get, post, put, delete та інші;

– Axios використовує проміси для обробки асинхронних операцій, що робить код більш читабельним та легким для підтримки;

– також Axios підтримує async/await синтаксис, що додає ще більше зручності у використанні;

– Axios автоматично перетворює JSON-дані у JavaScript-об'єкти і навпаки;

– це спрощує роботу з даними та зменшує обсяг коду, який потрібно писати вручну;

– Axios надає вбудовану систему для обробки помилок, що дозволяє легко обробляти несправності та забезпечує надійність додатків;

– Axios дозволяє налаштовувати запити за допомогою конфігураційних об'єктів та використовувати перехоплювачі для модифікації запитів або відповідей перед тим, як вони будуть відправлені або отримані;

– це дає можливість додавати заголовки, встановлювати тайм-аути, обробляти глобальні помилки та виконувати інші операції.

Переваги Axios:

– Axios працює не тільки у браузерах, але й у середовищі Node.js, що робить його корисним для різних типів додатків;

– Axios забезпечує зручний інтерфейс та інтуїтивно зрозумілий API, що спрощує здійснення HTTP-запитів і обробку відповідей;

– завдяки своїй гнучкості та можливості конфігурації, Axios підходить як для невеликих, так і для великих додатків.

Zustand – це легка бібліотека для управління станом у React-додатках. Бібліотека забезпечує простий, але потужний спосіб керування глобальним станом [6] без використання складних шаблонів чи залежності від контексту.

Основні особливості Zustand:

– Zustand має інтуїтивно зрозумілий API, який дозволяє створювати та використовувати стан у React-додатках з мінімальними зусиллями;

– він використовує функціональний підхід до управління станом, де стан описується за допомогою створених функцій-редакторів;

– Zustand підтримує як глобальний, так і локальний стан;

– це означає, що ви можете використовувати його для зберігання як спільних, так і специфічних для компонентів даних;

– завдяки використанню JavaScript Proxy та оптимізованих алгоритмів, Zustand забезпечує високу продуктивність та мінімізує непотрібні перерендери компонентів;

– Zustand може працювати як у браузері, так і в серверному середовищі, що робить його придатним для рендерингу на сервері (SSR) або інших середовищах, таких як React Native.

Переваги Zustand:

– Zustand має невеликий розмір, що робить його ідеальним для додатків, які прагнуть мінімізувати вагу залежностей;

– завдяки простому та зрозумілому API, Zustand дозволяє легко створювати та використовувати стан у React-компонентах без зайвої складності.

1.6 Постановка задачі

Перша задача полягає у розборі роботи бібліотеки Tensorflow і моделей із методами, що будуть використані для забезпечення роботи KYC-верифікації.

Другою задачею є вибір способу імплементації функціоналу, який система повинна забезпечувати, а саму перевірку особистості користувачів та інших особистих даних.

Третя задача включає створення зручного інтерфейсу, що сприяє легкому доступу та управлінню торговельними процесами для всіх користувачів платформи, включаючи адміністрацію. Клієнт і адміністративний клієнт повинні бути інтуїтивно зрозумілими, забезпечувати швидкий доступ до необхідних функцій і відповідати сучасним стандартам дизайну та функціональності.

Третя задача полягає в плануванні архітектури серверного застосунку, модулів, які будуть створені, описання їх взаємодії та призначення. Також у плануванні двох клієнт застосунків, один із яких, є застосунком для користувачів, самою платформою, а інший – адміністративна панель.

Четверта задача полягає у проектуванні бази даних, яка ефективно підтримує необхідний функціонал вашого додатку. Це включає створення оптимізованої схеми бази даних з використанням засобів нормалізації для забезпечення цілісності даних, а також планування індексації для покращення швидкості запитів.

П'ята задача включає розробку серверного застосунку за допомогою NestJS, TypeScript, та інтеграцію з базою даних через Prisma ORM. Основна увага буде приділена створенню RESTful API для обробки запитів від клієнтських застосунків, забезпеченню безпеки через аутентифікацію та авторизацію, а також логіки для обробки і зберігання даних.

Шоста задача зосереджується на реалізації клієнтських застосунків за допомогою React, що включає розробку інтуїтивно зрозумілих інтерфейсів для кінцевих користувачів і адміністраторів. Це охоплює імплементацію реактивних компонентів для відображення та управління даними в реальному часі, інтеграцію

з API сервера для забезпечення взаємодії та обміну даними між клієнтом і сервером.

Реалізація цих задач потребує залучення сучасних технологій та методик розробки, зокрема використання мови програмування JavaScript, фреймворку NestJS для серверного застосунку, React для клієнту та системи управління базами даних PostgreSQL. Такий підхід дозволить створити масштабовану, ефективну та високопродуктивну платформу, яка задовольнить потреби користувачів та відповідатиме всім регулятивним вимогам.

Висновки до розділу 1

Вибір технологій був обґрунтований і цілеспрямований, орієнтований на створення стійкої, масштабованої та високопродуктивної вебплатформи для торгівлі товарами подвійного призначення з KYC-верифікацією.

NestJS забезпечує надійність серверного застосунку, оптимізований для масштабування та легкої інтеграції з іншими сервісами та базами даних, зокрема з PostgreSQL, яка відома своєю стабільністю та розширеними функціональними можливостями. Це дозволяє впевнено керувати великими об'ємами даних, що є критично важливим для систем KYC, де потрібно зберігати великі масиви чутливої інформації.

Prisma як сучасний ORM інструмент оптимізує взаємодію з базою даних, забезпечуючи чистоту коду та легкість управління даними, що сприяє швидкому розгортанню та підтримці бази даних.

Зі сторони клієнту, використання React дозволяє створювати інтуїтивно зрозумілі та динамічні користувацькі інтерфейси, що забезпечують зручність і ефективність взаємодії з користувачем. Це особливо важливо для платформ KYC, де користувацький досвід може вплинути на сприйняття безпеки та надійності системи.

Використання цих технологій разом створює міцну основу для розробки вебплатформи, яка не тільки відповідає технічним вимогам сучасного ринку, але й

забезпечує необхідний рівень безпеки та приватності для дотримання регулятивних стандартів. Такий підхід забезпечує високу адаптивність та можливість майбутнього розширення функціоналу, зберігаючи при цьому легкість управління та високу продуктивність.

2 МОДЕЛІ І МЕТОДИ ДЛЯ KYC ВЕРИФІКАЦІЇ

2.1 Опис TensorFlow.js

TensorFlow.js — це бібліотека машинного навчання, яка дозволяє виконувати операції зі штучним інтелектом прямо в браузері, без необхідності використання серверних ресурсів. Це розширення оригінальної бібліотеки TensorFlow від Google, адаптоване для веброзробників з метою впровадження можливостей машинного навчання у клієнт-додатки. TensorFlow.js дозволяє розробникам навчати моделі безпосередньо у браузері або використовувати попередньо навчені моделі для виконання різноманітних завдань, таких як виявлення об'єктів, розпізнавання мови та аналіз зображень.

Однією з ключових особливостей TensorFlow.js є її інтеграція з сучасними вебтехнологіями, такими як WebGL для прискорення обчислень за допомогою графічних процесорів (GPU), що значно збільшує швидкість виконання складних моделей машинного навчання. Це робить TensorFlow.js ідеальним інструментом для створення інтерактивних додатків з використанням штучного інтелекту, які вимагають миттєвої відповіді від моделі, наприклад, в додатках для розпізнавання облич або перекладу мови в реальному часі.

Основна технічна особливість TensorFlow.js полягає у використанні JavaScript API для інтеграції моделей машинного навчання у вебдодатки. Бібліотека підтримує виконання як попередньо навчених моделей, так і моделей, які можуть бути навчені безпосередньо в браузері. Навчання моделей в браузері можливе завдяки WebGL, технології, яка дозволяє TensorFlow.js використовувати графічний процесор клієнта для прискорення обчислень.

TensorFlow.js надає розробникам можливість використання різних архітектур нейронних мереж, таких як конволюційні нейронні мережі (CNN), рекурентні нейронні мережі (RNN), та інші складні архітектури, які використовуються для задач обробки зображень, аналізу тексту та інших видів даних. Бібліотека також

включає інструменти для трансформації та нормалізації даних, що є критично важливими для підготовки даних перед тренуванням моделей.

Для полегшення імпорту моделей, створених у Python-версії TensorFlow, TensorFlow.js підтримує конвертацію цих моделей за допомогою TensorFlow SavedModel або формату Keras model. Це дозволяє розробникам легко мігрувати свої існуючі моделі до вебсередовища, відкриваючи двері для нових вебдодатків, які використовують передові можливості штучного інтелекту для покращення користувацького досвіду і забезпечення нових функціональностей.

В цілому, TensorFlow.js відкриває широкі можливості для веброзробників, дозволяючи їм використовувати передові технології машинного навчання для покращення досвіду користувачів, оптимізації взаємодій з додатками та створення нових інноваційних рішень у вебпросторі.

2.2 Використані моделі

Для реалізації сервер-системи, яка займається порівнянням людей на двох фотографіях, можна використовувати моделі глибокого навчання, які були обговорені раніше: ssdMobilenetv1 для виявлення облич, faceLandmark68Net для визначення ключових точок на обличчі та faceRecognitionNet для порівняння облич. Ця система може бути імплементована з використанням бібліотеки TensorFlow.js, що дозволяє ефективно виконувати машинне навчання прямо на сервері.

2.1.1 Модель ssdMobilenetv1

Модель ssdMobilenetv1 є ефективним рішенням для виявлення об'єктів у реальному часі, зокрема для виявлення облич. Ця модель базується на архітектурі Single Shot Multibox Detector (рис. 2.1), що дозволяє їй визначати різні об'єкти на зображеннях за один прохід. Використання MobileNet як базової мережі допомагає збалансувати точність та швидкість, роблячи ssdMobilenetv1 ідеальною для застосувань, де необхідні швидке виявлення облич та мінімальне споживання

ресурсів. Це робить її популярною вибором для вбудованих систем і мобільних застосунків.

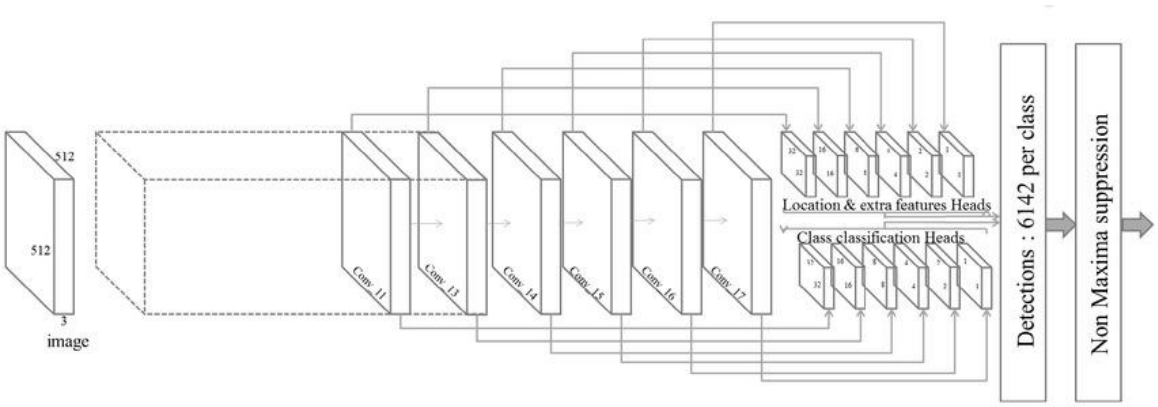

Рисунок 2.1 – Архітектура ssdMobilenetv1 [3]

ssdMobilenetv1 оптимізована для швидкості, що дозволяє їй працювати в реальному часі навіть на пристроях з обмеженими обчислювальними ресурсами, такими як смартфони та планшети. Використання легковагової архітектури MobileNet дозволяє зменшити кількість параметрів, що необхідні для моделі, забезпечуючи при цьому достатню точність для багатьох застосувань. Завдяки цьому ssdMobilenetv1 стає відмінним рішенням для розробників, які потребують швидкого виявлення об'єктів без значного навантаження на процесор.

Крім того, модель може бути натренована або адаптована для розпізнавання різних типів об'єктів, що робить її універсальною для багатьох сценаріїв використання. Підтримка високої модульності та адаптивності є важливими характеристиками, які дозволяють використовувати ssdMobilenetv1 у широкому спектрі застосувань, від систем безпеки до інтерактивних медіа. Ця модель не тільки використовується для виявлення облич, але й може ефективно виявляти інші об'єкти, забезпечуючи розробникам гнучкість у створенні складних застосунків машинного навчання.

2.1.2 Модель faceLandmark68Net

Модель faceLandmark68Net використовується для точного виявлення 68 ключових точок на обличчі людини, що включає точки навколо очей, носа, рота та контуру обличчя (рис. 2.2). Ця технологія дозволяє розробникам аналізувати різні аспекти обличчя, такі як вирази, позу та орієнтацію голови, що є особливо корисним у застосунках, пов'язаних із розпізнаванням емоцій, розширеною реальністю та інтерактивними іграми. Точність цієї моделі в виявленні точок дозволяє створювати деталізовані та реалістичні анімації обличчя, а також покращує здатність системи до адаптації в різних умовах освітлення і позиціях голови.

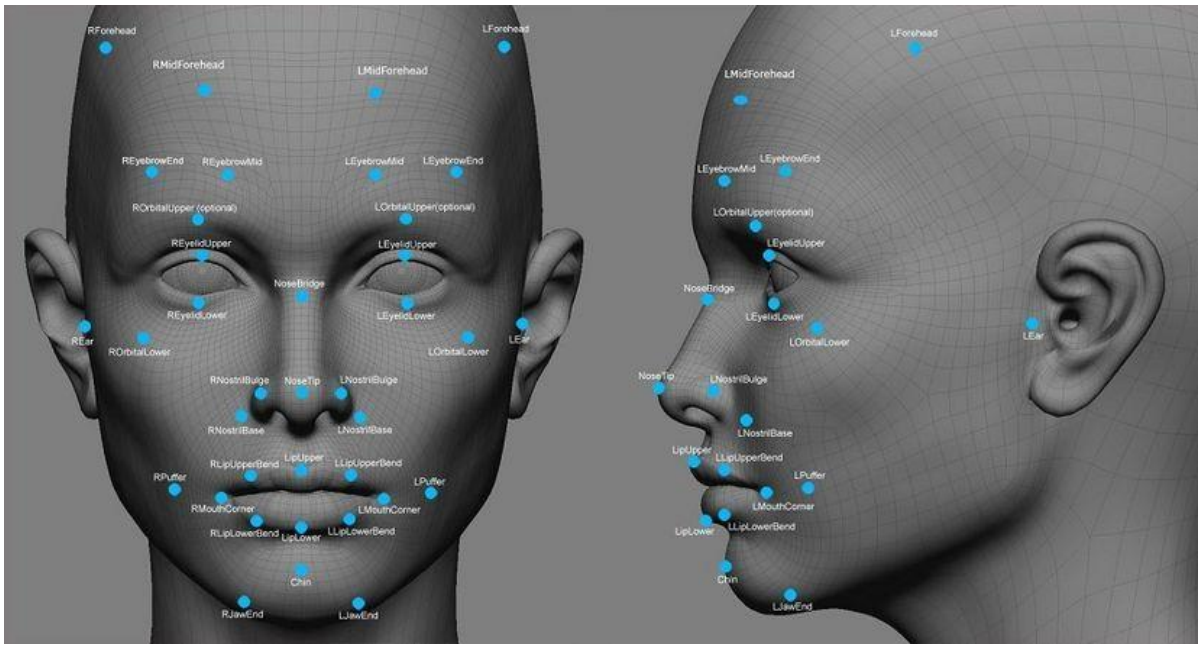

Рисунок 2.2 – Точки орієнтирів обличчя [4]

Завдяки своїй ефективності та високій точності, faceLandmark68Net широко застосовується не тільки в академічних і дослідницьких цілях, але й у комерційних продуктах. Наприклад, вона використовується для створення персоналізованих досвідів у косметичній промисловості, де додатки можуть автоматично пропонувати продукти або стилі макіяжу, базуючись на аналізі особливостей

обличчя користувача. Також ця технологія знайшла застосування в системах відеонагляду та безпеки, де можливість точно ідентифікувати особи та їхні вирази облич може служити засобом покращення безпеки і контролю.

Крім того, faceLandmark68Net застосовується в системах біометричної ідентифікації, дозволяючи створювати детальні біометричні профілі на основі структури обличчя. Точне визначення локалізації кожної з 68 точок може допомогти у покращенні систем доступу, які використовують розпізнавання облич як один з методів аутентифікації. Ця модель стає ключовим інструментом в розробці інтерфейсів, що здатні адаптуватися до емоційного стану користувача, забезпечуючи більш природну та інтуїтивну взаємодію.

Однією з ключових переваг faceLandmark68Net є її здатність інтегруватися з іншими моделями розпізнавання облич, що дозволяє створювати комплексні рішення, здатні виконувати багатофункціональні завдання, як-от автентифікація особи, моніторинг уваги або навіть автоматичне генерування відеоконтенту. Її можливість працювати в реальному часі робить цю модель незамінною для застосунків, де швидка реакція є критично важливою, наприклад, у інтерактивних системах, що вимагають негайної відповіді на дії користувача.

2.1.3 Модель faceRecognitionNet

Модель faceRecognitionNet розроблена для високоточного розпізнавання облич. Використовуючи глибокі нейронні мережі, вона аналізує зображення обличчя, порівнюючи його з базою даних зареєстрованих осіб, щоб ідентифікувати або верифікувати особу. Ця технологія знаходить застосування в різноманітних областях, включно з системами безпеки, контролю доступу та ідентифікаційними системами в банківській справі, де важлива висока точність ідентифікації особи для запобігання шахрайству і забезпечення конфіденційності.

Одна з важливих характеристик faceRecognitionNet полягає в її здатності ефективно працювати навіть у менш ідеальних умовах освітлення або при частковому закритті обличчя, що робить її надійним інструментом для реальних

застосунків. Модель також адаптована до розпізнавання облич з різних ракурсів, забезпечуючи високу точність навіть при бічному перегляді або наклоні голови. Ця адаптивність є критично важливою для застосувань, де користувачі можуть не завжди знаходитись перед камерою під оптимальним кутом.

Завдяки можливостям архітектури (рис. 2.3), faceRecognitionNet є відмінним вибором для інтеграції в розумні домашні системи та персональні асистенти, де може використовуватися для персоналізації взаємодії з користувачами. Це включає персональні привітання, автоматичне налаштування систем дому згідно з уподобаннями користувача та більш безпечне і зручне управління доступом до домашніх пристроїв. Її універсальність і надійність роблять модель faceRecognitionNet цінним інструментом у великому спектрі застосувань, що вимагають точного розпізнавання особи.

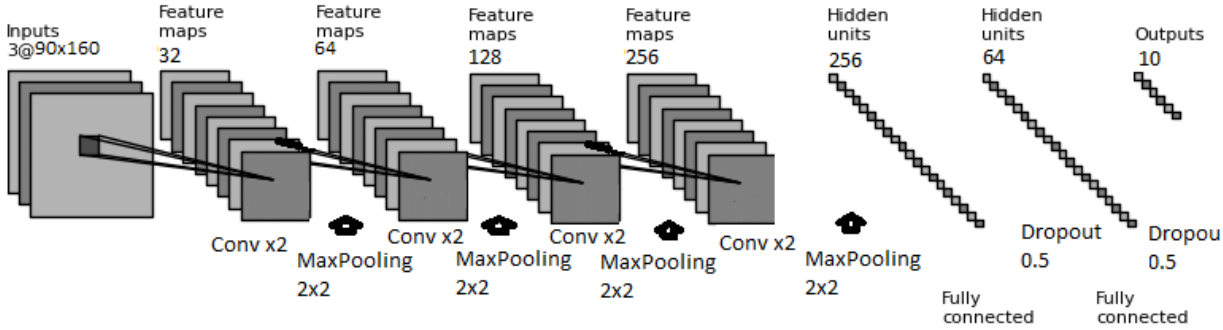

Рисунок 2.3 – Архітектура faceRecognitionNet[5]

Модель здатна аналізувати і зберігати великі обсяги даних облич, використовуючи їх для створення надійних ідентифікаційних процедур. Це особливо корисно в умовах, де безпека є пріоритетом, наприклад, в банківському секторі або на стратегічно важливих об'єктах. FaceRecognitionNet також використовується для персоналізації рекламних та маркетингових кампаній, аналізуючи демографічні характеристики та переваги користувачів на основі їхніх облич.

2.3 Метод реалізації

Для реалізації модуля на сервері, який використовує технології машинного навчання для порівняння осіб на двох фотографіях у рамках вебдодатку, розробленого на платформі NestJS, необхідно виконати кілька кроків. Модуль повинен інтегруватися з бібліотекою TensorFlow.js [7] і використовувати глибокі нейронні мережі для визначення та порівняння облич.

Перш за все, слід створити новий модуль у NestJS, який буде відповідати за обробку фотографій. В рамках цього модулю потрібно розробити сервіс, який міститиме методи для завантаження фотографій, виявлення облич на них та подальшого порівняння цих облич. Сервіс повинен приймати два зображення у форматі, прийнятному для вебзапитів, наприклад, у формі Base64 [8] або як посилання на зображення, яке зберігається на сервері.

На другому етапі, використовуючи TensorFlow.js, необхідно імплементувати методи для виявлення облич на зображеннях. Можна використати попередньо навчені моделі, такі як ssdMobilenetv1 для виявлення облич, faceLandmark68Net для виявлення ключових точок обличчя та faceRecognitionNet для отримання векторів особливостей облич, що дозволяють порівнювати обличчя на різних фотографіях.

Після обробки фотографій і отримання векторів особливостей для кожного обличчя, слід розробити алгоритм порівняння цих векторів. Це може бути реалізовано через визначення косинусної подібності між двома векторами, що дасть змогу визначити, наскільки однаковими є обличчя на двох фотографіях. Важливо зазначити, що точність порівняння залежатиме від якості вхідних зображень та різноманітності навчального набору даних, на якому були навчені моделі.

Нарешті, слід забезпечити належну інтеграцію сервісу з іншими частинами системи, зокрема з контролерами та інтерфейсом користувача, а також впровадити адекватні механізми обробки помилок та безпеки для захисту даних користувачів.

Розроблення такого модуля вимагає глибокого розуміння машинного навчання, веброзробки та основних принципів роботи з зображеннями.

Висновки до розділу 2

У цьому розділі детально розглянуто застосування технологій машинного навчання, зокрема TensorFlow.js, для виконання задач KYC верифікації безпосередньо у веббраузері. Обговорено переваги інтеграції машинного навчання в клієнтські додатки, що дозволяє здійснювати обробку та аналіз даних на стороні користувача без залучення серверних ресурсів. Важливість цього підходу особливо проявляється у задачах, які потребують швидкої відповіді та високої приватності даних.

Значну увагу приділено можливостям TensorFlow.js, таким як навчання моделей прямо у браузері та їх швидке виконання завдяки використанню WebGL та GPU. Це відкриває двері для створення інноваційних вебдодатків, які можуть виконувати складні задачі обробки зображень, такі як розпізнавання облич, аналіз виразів обличчя або навіть більш комплексні системи для перевірки документів і персональних даних.

Також обговорено специфічні моделі машинного навчання, такі як ssdMobilenetv1, faceLandmark68Net та faceRecognitionNet, які можуть використовуватися для визначення особистості в рамках KYC. Кожна з цих моделей має свої особливості та призначення, від швидкого виявлення облич до точного аналізу ключових точок і розпізнавання осіб. Використання цих технологій сприяє не тільки забезпеченню безпеки та дотриманню регуляторних вимог, але й підвищує зручність та ефективність обслуговування користувачів.

3 ПЛАНУВАННЯ ПРОЕКТУ. ПРОЕКТУВАННЯ БАЗИ ДАНИХ

3.1 Функціонал застосунку

3.1.1 Аутентифікація. Менеджмент даних користувачів застосунку

Задля імплементації KYC верифікації необхідно реалізувати роботу із даними користувачів, так, щоб при перевірці усі необхідні дані завжди були присутні та ними було легко оперувати та варто зважати на їх непостійність. Деякі дані користувача як посвідчення особи, актуальність номеру телефона, місце проживання можуть змінюватися і мають як можна частіше актуалізовуватися не надто навантажуючи цим процес взаємодії із користувача застосунком. Тож треба реалізувати аутентифікацію, задля збереження особи користувача у системі.

Для неї обрано спосіб із двома JWT токенами [9]. Де дані користувача передаються за допомогою зашифрованого токена, який можна прочитати, але не можна непомітно для системи підмінити. JWT-токен складається з трьох частин:

– Header – об’єкт JSON, який містить інформацію про тип токена та алгоритм його шифрування;

– Payload – об’єкт JSON, який містить дані, потрібні для авторизації користувача;

– Signature – рядок, який створюється за допомогою секретного коду, Headera та Payload. Цей рядок служить для верифікації токена.

У підході із двома токенами 1 із них є короткочасним і використовується для входу у систему, а другий є на випадок коли час життя першого токену закінчується, в такому разі за допомогою нього застосунок впізнає користувача і може генерувати йому нові токени без додаткових дій із клієнтської сторони. Додатково, використання JWT сприяє масштабуванню системи. JWT токени легко передавати через мережеві з'єднання, оскільки вони просто вбудовуються в заголовки HTTP. Це робить технологію ідеальною для систем, які використовують

мікросервіси або розподілені архітектури, де потрібно забезпечувати аутентифікацію та авторизацію користувачів на різних серверах або платформа

Зі сторони клієнтського застосунку користувач має сторінки реєстрації,аутентифікації та відновлення паролю на випадок втрати доступу до попереднього. А зі сторон серверу реалізовані логіка обробки отриманих від користувача даних та функції генерації токенів та валідації токенів[10]. Так при будь-яких операціях користувача у вебмагазині, застосунок матиме доступ до даних про нього.

Уже аутенифікованому у системі користувачу доступний функціонал менеджменту особистих даних, у що входять оновлення інформації даної при реєстрації, яка включає особисті дані такі як ім’я і прізвище та облікові дані як пароль і емейл, проходження KYC форм і актуалізація даних та відстеження статусу їх погодження.

Заохочення користувачів до оновлення своїх даних також є критично важливим аспектом, особливо у контексті KYC, де актуальність інформації може вплинути на легальність та дійсність проведених транзакцій. Надаючи користувачам інструменти для легкого оновлення своїх особистих даних через користувацький інтерфейс, можна не тільки підвищити їхнє задоволення від використання платформи, але й забезпечити дотримання нормативних вимог.

Зі сторони адмін панелі, адміністратори мають можливість переглядати, оновлювати, видаляти дані юзера, призупиняти акаунт переглядати дані із форм KYC верифікації і погоджувати користувачу новий рівень чи відхиляти запит.

3.1.2 Оперування товарами. Замовлення

Основною особливістю платформи для продажу товарів подвійного призначення є не тільки можливість купівлі товарів, але й великий акцент на управлінні асортиментом через адміністраторську панель. Кожен товар у системі класифікується за категоріями, що дозволяє користувачам легко навігувати та знаходити необхідні продукти. Адміністратор має можливість додавати нові

категорії або редагувати існуючі, що робить асортимент гнучким та адаптованим до змін у попиті та ринкових умовах.

Кожен товар може бути оснащений відгуками від покупців, що сприяє більшій довірі між користувачами і підвищує прозорість при купівлі. Система відгуків дозволяє користувачам залишати свої коментарі та оцінки для товарів, які вони придбали, тим самим допомагаючи іншим зробити обізнані вибори. Адміністратори, у свою чергу, можуть моніторити відгуки для забезпечення відповідності стандартам якості та вчасно реагувати на можливі проблеми або питання з боку клієнтів.

Ціноутворення товарів є ключовим аспектом управління асортиментом. Адміністратори мають інструменти для встановлення та оновлення цін, залежно від змін у виробництві, податках або логістиці[11]. Ціни можуть бути динамічно адаптовані до ринкових тенденцій, що дозволяє підтримувати конкурентоспроможність платформи.

Зміни у наявності товарів є критично важливими для підтримки неперервності продажів і запобігання ситуаціям, коли покупці не можуть знайти потрібний товар в наявності. Адміністраторська панель включає функціональність для відстеження запасів кожного товару, дозволяючи адміністраторам оперативно реагувати на зміни в наявності, як наприклад, швидке замовлення додаткових одиниць у разі їх нестачі.

Завдяки цим інструментам, платформа не тільки спрощує процес покупки для кінцевого користувача, але й надає адміністраторам потужний набір функцій для ефективного управління асортиментом, ціноутворенням, наявністю товарів та взаємодії з клієнтами через систему відгуків. Такий підхід не лише підвищує задоволення користувачів, але й сприяє підтриманню високих стандартів обслуговування та управління на платформі.

3.1.3 Зберігання файлів

Коли справа доходить до впровадження рівнів KYC (Know Your Customer) процесів, платформа забезпечує ретельне верифікування ідентичності користувачів. Для другого і третього рівнів KYC необхідно, щоб користувачі надали свої фотографії або скан-копії документів, що підтверджують особу. Це може включати паспорт, водійське посвідчення, або інші урядові документи. Такі заходи не тільки забезпечують відповідність платформи регулятивним вимогам, але й сприяють підвищенню довіри між користувачами та платформою.

З метою забезпечення безпеки та конфіденційності користувачів, всі фотографії та скановані документи зберігаються у Google Cloud Bucket Storage [12]. Використання хмарних сховищ дозволяє не тільки легко масштабувати зберігання відповідно до зростання кількості користувачів, але й забезпечує високий рівень захисту даних. Файли завантажуються і зберігаються у зашифрованому вигляді, що мінімізує ризики витоку інформації та недозволеного доступу.

Ця підхід також спрощує процеси внутрішнього аудиту і перевірки, оскільки адміністратори мають змогу швидко доступатися до верифікованих даних у випадку потреби. Використання хмарних технологій дозволяє платформі зосередитися на основних аспектах бізнесу, не переймаючись за управління інфраструктурою зберігання.

Надійність та прозорість такого підходу до зберігання документів забезпечує не лише відповідність законодавчим і регулятивним вимогам, але й підвищує лояльність та задоволеність користувачів, знаючи, що їхні особисті дані знаходяться у безпеці. Таким чином, платформа забезпечує високий рівень сервісу та безпеки, здійснюючи обробку та зберігання критично важливої інформації в надійному та ефективному форматі.

3.1.4 Структура бази даних

Таблиці спроектовані для обробки інформації про користувачів, адміністраторів, замовлення, товари, відгуки та інші аспекти системи. Детальний опис кожної таблиці та її призначення (рис. 3.1).

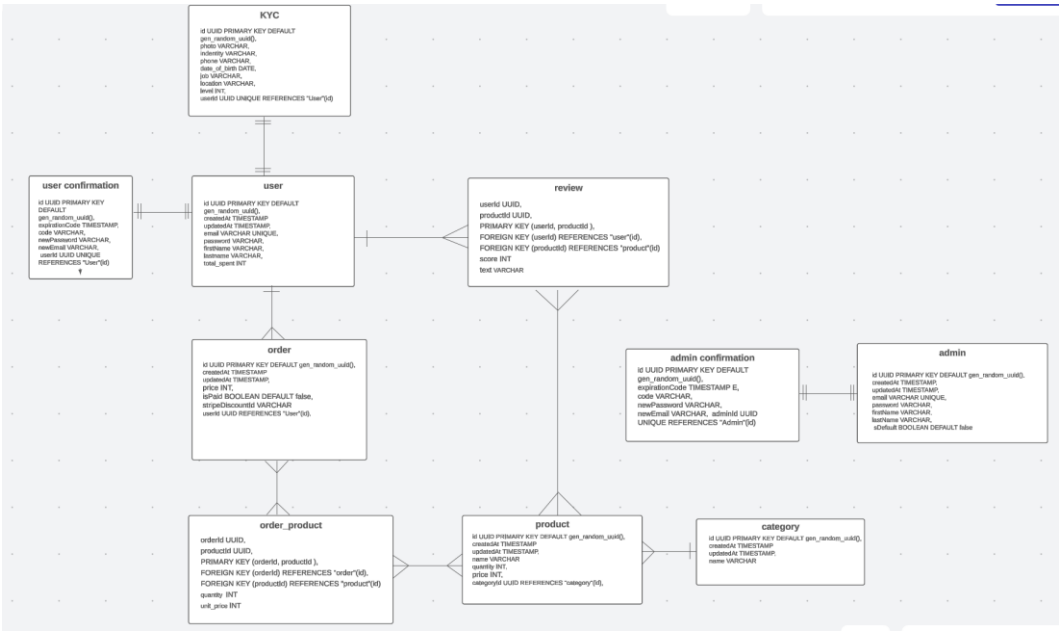

Рисунок 3.1 – Сутнісно-зв'язкова діаграма бази даних

1. Таблиця user – містить основну інформацію про користувачів платформи. Ці дані використовуються для аутентифікації користувача в системі. Ключові атрибути включають:

– id (UUID) – унікальний ідентифікатор для кожного користувача;

– created_at, updated_at – часові мітки створення та останнього оновлення користувацького профілю;

– email – електронна адреса користувача, яка використовується для входу;

– password – зашифрований пароль для безпеки облікового запису;

– first_name, last_name – ім'я та прізвище користувача;

– total_spent – сума, витрачена користувачем на платформі, яка може використовуватися для аналітики та маркетингу.

2. Таблиця admin – подібна до таблиці user, але призначена для адміністраторів платформи:

– id (UUID) – унікальний ідентифікатор для кожного адміністратора;

– email, password – використовуються для аутентифікації адміністратора;

– first_name, last_name – особиста інформація для ідентифікації адміністратора;

– is_default – показник, чи є цей адміністратор первинним (за замовчуванням).

3. Таблиця KYC є важливою складовою бази даних у системах, де потрібно забезпечувати відповідність законодавчим вимогам та перевірку особистості користувачів. Ця таблиця має критичне значення для фінансових послуг, онлайн-торгівлі, а також будь-яких інших платформ, що вимагають підвищеної безпеки та верифікації особи. Зберігає детальну персональну інформацію, отриману при проходжені користувачем KYC верифікацій, необхідну для процедур KYC:

– id (UUID) – унікальний ідентифікатор;

– photo, identity – фотографія та документ, що підтверджує особу;

– phone, date_of_birth, job, local_id, livein – інші особисті дані, які можуть бути використані для повної верифікації.

4. Таблиця order відслідковує замовлення, зроблені на платформі:

– id (UUID) – унікальний ідентифікатор замовлення;

– created_at, updated_at, price, is_paid – інформація про час створення, останнє оновлення, загальну вартість замовлення та статус оплати;

– stripeID, isPaidBocounted – інтеграція з системою Stripe для обробки платежів.

5. Таблиця order_product реалізує багато-до-багатьох відношення між замовленнями та продуктами:

– order_id, product_id – зовнішні ключі, що з'єднують замовлення та продукти;

– quantity, unit_price – кількість замовленого товару та ціна за одиницю.

6. Таблиця product – зберігає інформацію про id (UUID), name, price, quantity, category_id – унікальний ідентифікатор, назва, ціна, кількість на складі та зв'язок з категорією.

7. Таблиця category – управляє категоріями продуктів.

8. Таблиці user_confirmation та admin_confirmation обробляють зміни у даних користувачів та адміністраторів містять поля id, email, password_hash, code, newEmail – поля, необхідні для підтвердження змін у профілях.

Ці таблиці разом формують комплексну систему для управління користувачами, адміністраторами, продуктами, замовленнями та взаємодії на платформі.

3.2 Основні частини системи. Їх організація

3.2.1 Застосунки системи

У пункту 1 вже визначено, що проект складається з 3 основних частин, кожна з яких має свої особливості та вимоги до реалізації:

а) окремий серверний застосунок, в якому відбувається реалізація бази даних і робота із нею, бізнес логіка та інтеграція сторонніх сервісів; Для реалізацію використовується фреймворк Nest.js;

б) користувацький клієнт застосунок – окремий клієнтський застосунок, який слугує вебсайтом магазину товарів подвійного призначення. Його основна роль – забезпечення зручного інтерфейсу для кінцевих користувачів. Застосування React дозволяє створювати динамічний та відгуковий UI;

в) адміністраторський клієнт застосунок – окремий клієнтський застосунок, який виступає у ролі адмін панелі, де реалізований функціонал для керування і моніторингу роботою всього проекту та даними із серверу Для реалізації використовується фреймворк React.

Розділення адміністраторського та користувацького клієнтів в одному проекті не тільки поліпшує організацію коду, але й оптимізує досвід користувачів

та адміністраторів [13]. Адміністраторський інтерфейс, який призначений виключно для внутрішнього використання, містить розширені інструменти для моніторингу, управління даними користувачів, аналізу продажів та налаштувань системи. Ці функції вимагають вищого рівня доступу та безпеки, оскільки вони включають чутливі операції, які не можуть бути доступними для звичайних користувачів.

З іншого боку, користувацький клієнт орієнтований на кінцевого користувача, надаючи інтуїтивно зрозумілий і зручний інтерфейс для перегляду та покупки товарів. Ця частина фокусується на забезпеченні плавного користувацького досвіду з мінімумом перешкод для користувачів [14], що сприяє збільшенню задоволеності та лояльності клієнтів.

Рішення про розділення цих двох аспектів дозволяє кожній частині концентруватися на своїх специфічних потребах без компромісів в плані проектування чи безпеки[15]. Такий підхід також знижує складність кожного клієнту, оскільки відповідальності чітко поділені та визначені. Оскільки сайт призначений лише для власника, немає необхідності в розробці складних рольових систем чи інтеграції зі сторонніми сервісами, що могли б ускладнити архітектуру та управління проектом.

Отож у результаті проект складається із двох клієнтських застосунків та одного серверного.

3.2.2 Використання мультирепозиторію

Для кращої організації кодової бази вирішено об'єднати усі 3 застосунки в одному репозиторії. Мультирепозиторій може надати кілька ключових переваг порівняно з окремими репозиторіями для кожного застосунку [16]. Ось деякі з основних переваг:

– спільне використання коду, мультирепозиторії легше спільно використовувати код між різними застосунками;

– керування версіями і релізами в єдиному репозиторії дозволяє легше синхронізувати випуски між проектами. Всі проекти можуть використовувати однакові версії залежних бібліотек, що зменшує ризики несумісності та помилки в роботі застосунків;

– один репозиторій може використовувати спільні налаштування для розгортання, тестування та інших процесів DevOps. Це зменшує кількість скриптів та конфігурацій, які потрібно створювати та підтримувати;

– управління кількома проектами в одному репозиторії дозволяє централізувати задачі, що спрощує трекінг проблем та розвиток проектів. Командам легше координувати зусилля і спільно працювати над завданнями;

– централізоване управління проектами може допомогти оптимізувати використання серверних та розробницьких ресурсів, оскільки команда використовує спільну інфраструктуру для розробки, тестування та розгортання;

– корінь мультирепозиторію міститиме файли конфігурацій лінтера і тайпскрипта та залежності усіх застосунків.

3.3 Структура серверної частини

3.3.1 Архітектура

В основі NestJS лежить архітектурний паттерн, схожий на той, який використовується в Angular. Застосунок у NestJS складається з декількох ключових компонентів [17], які взаємодіють між собою для забезпечення гнучкості і функціональності.

Контролери в NestJS – це класи, що відповідають за обробку вхідних запитів і повернення відповідей клієнту. Вони служать точками входу для запитів від користувачів і часто асоційовані з певними маршрутами URL. Контролери використовують декоратори для вказівки маршрутів і можливих дій, що дозволяє легко розпізнати, які HTTP запити вони обробляють.

Провайдери в NestJS можуть бути класами, фабриками, значеннями або сервісами, що використовуються для абстракції та інкапсуляції логіки, яка не пов'язана безпосередньо з мережевими запитами [18]. Наприклад, сервіси, які виконують бізнес-логіку або взаємодіють з базою даних, є провайдерами. Вони впроваджуються через конструктори класів із використанням системи впровадження залежностей, що дозволяє легко замінювати їх під час тестування.

Модулі в NestJS організовують код застосунку і визначають межі контексту. Кожен модуль містить компоненти, такі як контролери, провайдери та імпорти інших модулів [19]. Завдяки цьому модулі можуть бути легко перевикористані в інших частинах застосунку або навіть у інших застосунках. Модулі забезпечують капсуляцію і зниження зв'язності компонентів, що сприяє чистоті та організованості коду.

Middleware – це функції, які викликаються перед методами контролера. Вони використовуються для виконання коду до того, як запит досягне контролера. Middleware може виконувати різні завдання, такі як логування, перевірка аутентифікації, збір статистики або модифікація вхідних запитів.

Фільтри винятків у NestJS дозволяють обробляти винятки, що виникають під час виконання запитів. Вони можуть перехоплювати винятки від контролерів і змінювати відповіді, що відправляються користувачам. Це корисно для відправлення користувачеві зрозумілих повідомлень про помилки і для забезпечення того, що сервер не розкриває чутливу внутрішню інформацію через неконтрольовані винятки.

Інтерцептори в NestJS дозволяють вам перехоплювати виклики методів перед тим, як вони будуть виконані, і після того, як вони повернули результат. Це може використовуватися для модифікації результатів, логування, затримок у відповідях або інших аспектів обробки запитів.

3.3.2 Модулі user і admin

Так як основною метою проекту є реалізація KYC для веб сайту для продажу товарів подвійного призначення, першими модулями є модуль user, який вміщатиме в собі усю логіку із роботи із користувачами та контролери, що прийматимуть запити із клієнту, їх даними і рівнями доступу і модуль admin який уміщатиме в собі логіку роботи з даними адміністраторів.

3.3.3 Модуль auth

Далі є модуль auth, в якому реалізовується функціонал аутентифікації застосунку. Як вже було описано використовується метод із 2 JWT токенами. Для його реалізацію використано стандартні для таких задач бібліотеки passport і passport-jwt. Ці бібліотеки допомагають імплементувати надійну та гнучку систему аутентифікації, що базується на JSON Web Tokens (JWT). За допомогою них реалізовано JWT стратегію.

Цей модуль не матиме прямого доступу до бази даних і робота із нею здійснюється за допомогою модулів user і admin. Для збереження чистоти коду вирішено розділити сервіс логіку цього модуля на 3 сервіси: для роботи із JWT токенами, для роботи із записами користувачів і ще один сервіс для роботи з адміністраторами.Так само і контролери для запитів із користувацького і адмінського клієнтів будуть окремі.Задля легшої обробки запитів, також додана перевірка ролей, які додаються як поле у JWT токен. У результаті архітектуру застосунку можна зобразтити у вигляді діаграми (рис. 3.2).

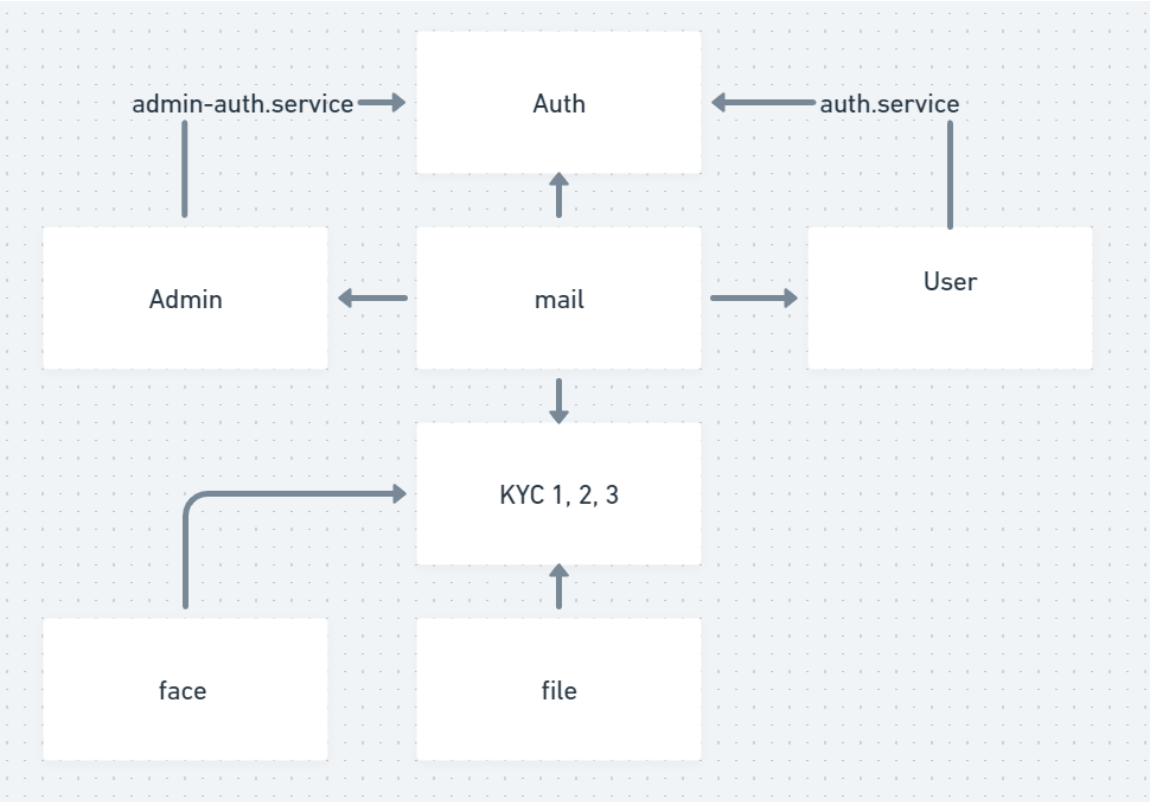

Рисунок 3.2 – Блокова діаграма серверних модулів, що використовуються у процесі верифікації користувачів

Висновки до розділу 3

У цьому розділі детально розглянуто ключові аспекти планування та проектування бази даних для вебплатформи, що включає реалізацію KYC верифікацій, аутентифікації користувачів, а також управління та обробку замовлень і товарів подвійного призначення. Особлива увага приділена архітектурі системи, зокрема впровадженню JWT токенів для забезпечення безпеки та ідентифікації користувачів.

Аналіз функціональних можливостей застосунку показує, що платформа ефективно інтегрує модульну структуру, що дозволяє легко масштабувати проект та вносити зміни без значного впливу на існуючу систему. Важливим елементом є також здатність адміністрування, яка включає повний спектр інструментів для

управління користувачами, товарними запасами, а також для моніторингу і верифікації KYC даних.

Цей розділ демонструє глибоке розуміння вимог до системи і забезпечує міцну основу для наступних етапів розробки та впровадження проекту. Важливо зазначити, що приділено належну увагу не тільки технічним аспектам проектування, але й взаємодії користувачів з системою, що є критично важливим для забезпечення високої задоволеності користувачів та дотримання нормативних вимог.

Завершення цього розділу надає чітке бачення щодо структури та функціональності майбутньої системи, підкреслюючи стратегічні напрями розвитку та ключові технологічні рішення, які будуть впроваджені в рамках проекту.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ

4.1 Аутентифікація

4.1.1 Серверна частина

Модуль аутентифікації складається з кількох основних компонентів, розташованих у відповідних піддиректоріях: const, dto, services, і strategy, кожна з яких відіграє свою роль у процесі аутентифікації.

Модуль аутентифікації розділений на декілька підкомпонентів для забезпечення модульності та легкості управління кодом. Конфігураційні константи та типи помилок визначаються у папці const. Об'єкти передачі даних (DTO – Data Transfer Objects), які використовуються для перевірки вхідних даних користувача при реєстрації та вході, розміщені в папці dto. Логіка аутентифікації і видачі токенів, реалізована в сервісах у папці services, а валідація та екстракція токенів здійснюються за допомогою стратегій у папці strategy.

“JWTService” відповідає за генерацію токенів доступу та оновлення. Під час генерації токенів, сервіс використовує допоміжні методи з JwtStrategy для створення та надсилання токенів у вигляді куків у відповіді до клієнта. Це дозволяє забезпечити безпечну та ефективну аутентифікацію на клієнтській стороні.

“AuthService” керує реєстрацією нових користувачів, активацією акаунтів та входом користувачів. Сервіс використовує “UserService” для доступу до даних користувачів та “MailService” для надсилання листів з кодом активації. Реєстрація користувачів включає перевірку наявності електронної пошти в базі даних і, у разі відсутності акаунту, створення нового користувача з хешованим паролем. Активація користувачів відбувається через перевірку наданого коду активації. Забезпечений функціонал відновлення забутого паролю

Також наявний “AuthAdminService” задля забезпечення специфічного для адміністративної панелі функціоналу.

“JwtStrategy” реалізує стратегію аутентифікації на основі токенів JWT, використовуючи бібліотеку Passport. Стратегія описує методи для екстракції

токену з куків, валідації токену та асоціації користувача з отриманими даними. Зокрема, використовуються два типи токенів: доступу та оновлення, що дозволяє забезпечити безперервний доступ користувачів до системи.

У контексті безпеки важливо відмітити використання хешування паролів з bcrypt, що забезпечує захист від атак методом підбору. Також значну увагу приділено безпечному зберіганню та передачі JWT токенів за допомогою HTTPS та куків із належними налаштуваннями безпеки.

4.1.2 Клієнтська частина

На стороні клієнту вебдодатку реалізовано систему аутентифікації, що використовує бібліотеку React і організована у вигляді двох основних модулів: sign-in (рис. 4.2) та sign-up (рис. 4.1).

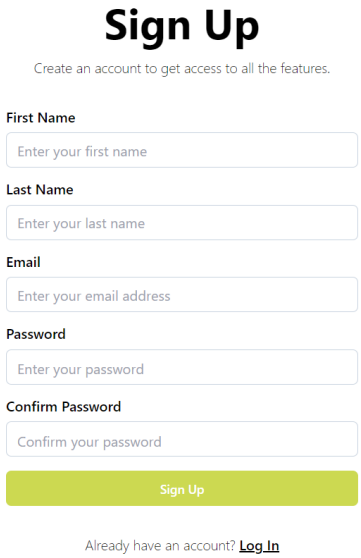

Рисунок 4.1 – UI сторінки реєстрації

Кожен модуль відповідає за окрему сторінку інтерфейсу користувача і інкапсулює всю необхідну логіку для виконання специфічних дій, таких як реєстрація або вхід користувачів. Кожен з модулів включає кілька підпапок і файлів:

– компоненти (components), які виконують роль візуального представлення функціоналу на сторінках входу та реєстрації;

– хуки (hooks) кастомні визначають і керують логікою взаємодії з API та управлінням станом форм [21]. Наприклад, хук useSignUp керує процесом реєстрації, включаючи валідацію даних і взаємодію з сервісом аутентифікації.

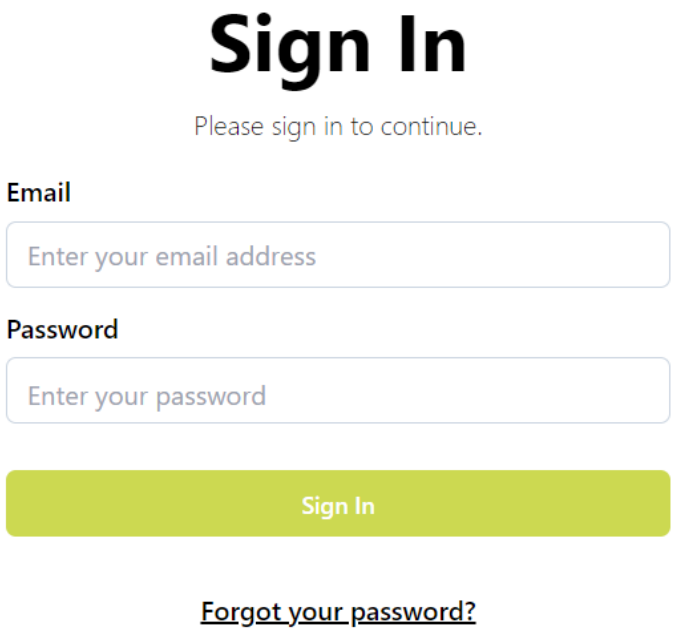

Рисунок 4.2 – UI сторінки аутентифікації

AuthService є основним сервісом для роботи з API. Цей сервіс використовує HttpService для відправлення запитів до сервера, які включають перевірку стану сесії користувача, реєстрацію нових користувачів, вхід в систему, активацію акаунту, повторну відправку коду активації, та вихід з системи[22].

У модулі sign-up реалізовано компонент SignUp, який використовує форму для збору даних користувача. Хук useSignUp керує логікою подання форми, включаючи перевірку валідності паролю, відправку даних на сервер через

AuthService та перенаправлення на сторінку активації після успішної реєстрації. Логіка валідації форми реалізована через validateFormValues, яка забезпечує додаткову перевірку введених даних перед відправкою на сервер[23].

Ця структура дозволяє чітко розділити логіку обробки даних та візуальне представлення, забезпечуючи легкість підтримки та розширення додатку. Кастомні хуки допомагають ізолювати специфічні процеси аутентифікації, роблячи код більш чистим і зрозумілим.

Для адмінського інтерфейсу клієнту вебдодатку також реалізована система аутентифікації, яка використовує бібліотеку React і подібно організована у вигляді модулів sign-in та sign-up. Цей інтерфейс особливо адаптований для адміністраторів, дозволяючи їм керувати доступом до конфіденційних розділів адмін панелі та взаємодіяти з сервісами, які вимагають вищих прав доступу. Модулі включають спеціалізовані компоненти та хуки, які забезпечують не тільки базову функціональність входу та реєстрації, а й додаткові можливості, такі як перегляд і управління користувацькими ролями та правами.

Реалізована сторінка відновлення паролю (рис. 4.3).

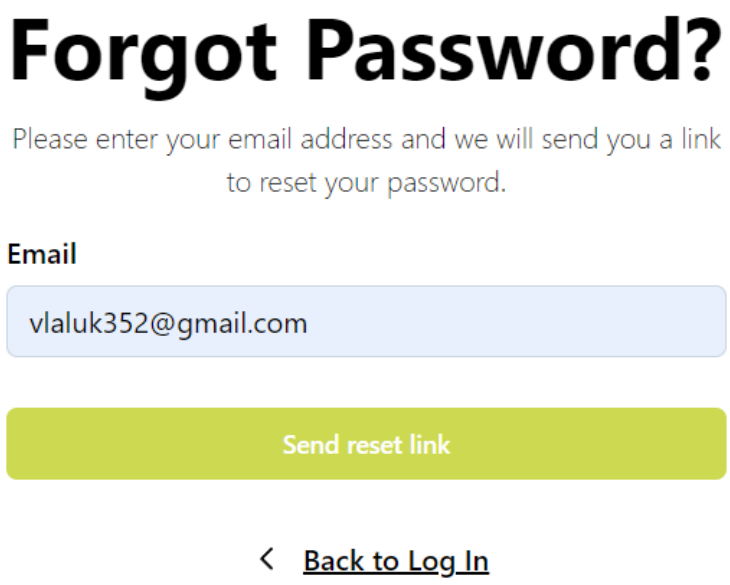

Рисунок 4.3 – UI сторінки відновлення паролю

Компоненти в адмін-модулях мають збільшені вимоги до безпеки та додаткові інструменти для моніторингу стану сесій адміністраторів. Кастомні хуки, такі як “useAdminSignIn”, включають розширені перевірки безпеки та логіку сесійного управління, що важливо для забезпечення надійного доступу до адміністративних функцій. “AuthService” для адміністраторів адаптовано під потреби керування вищими рівнями доступу та включає методи для спеціальних дій, як-от блокування акаунтів та перевірка дійсності сесій на основі рівнів доступу. Ця адаптація забезпечує додатковий рівень захисту та контролю, що є критичним для адміністративного інтерфейсу.

4.2 KYC Верифікація

4.2.1 Серверний функціонал

У реалізації системи верифікації KYC використовуються два основних сервіси: “KYCService” і “KYCManagementService”, які взаємодіють із базою даних через Prisma ORM[25], що дозволяє забезпечити ефективну обробку даних користувачів. Сервіс “KYCService” відповідає за базові функції, такі як отримання актуальних статусів верифікації користувачів та оновлення цих статусів. Цей сервіс забезпечує також завантаження та зберігання документів користувачів, використовуючи для цього інтегрований файловий сервіс, який дозволяє генерувати та зберегти публічні посилання на завантажені документи.

Для більш детального управління даними результу KYC-верифікації в системі використовується “KYCManagementService”. Цей сервіс надає адміністраторам можливість переглядати всі запити, що очікують на обробку, та змінювати статуси KYC записів в залежності від результатів перевірки. Використання такого підходу дозволяє централізовано управляти процесом верифікації, забезпечуючи високий рівень безпеки та відповідність законодавчим вимогам.

Методи, які реалізовані в “KYCService”, дозволяють не тільки отримати доступ до інформації про статус кожного етапу KYC, але й здійснювати необхідні

зміни у відповідності до вимог регуляторів. Ці методи забезпечують високу міру гнучкості та контролю над процесом верифікації, включаючи оновлення статусів на "очікування" після подання документів користувачами. Це важливо, оскільки процес верифікації може вимагати додаткових перевірок та залучення додаткових ресурсів.

Крім того, обробка виключень в системі реалізована за допомогою виключень HttpException, які використовуються для повідомлення про помилки у випадках, коли інформація про користувача не знайдена або інші помилки запитів. Це забезпечує чітке та ефективне керування помилками, дозволяючи розробникам швидко ідентифікувати та вирішувати потенційні проблеми в системі.

Нарешті, інтеграція системи захисту маршрутів через JwtAuthGuard є критично важливою для забезпечення того, щоб лише аутентифіковані користувачі мали доступ до функціоналу системи. Це важливо не тільки для безпеки даних, але й для відповідності нормативним вимогам щодо обробки персональних даних користувачів. Використання JSON Web Tokens (JWT) для цих цілей є стандартною практикою, що дозволяє ефективно управляти сесіями користувачів та забезпечувати контроль доступу на рівні кожного запиту. Взаємодія з контролером в системі KYC здійснюється через клас “KYCController”, який є відповідальним за обробку HTTP запитів від клієнтів. Завдяки цьому контролеру, користувачі можуть подавати запити на верифікацію через вебінтерфейс, які потім обробляються відповідними сервісами. Контролер використовує механізми інтерсепції файлів та аутентифікації, щоб забезпечити безпечну та ефективну обробку даних.

Контролер реалізує серію ендпоінтів, кожен з яких призначений для різних рівнів KYC верифікації. Наприклад, за допомогою POST запиту до ендпоінту “/kyc1” користувачі можуть подати дані для першого рівня KYC. Аналогічно, інші ендпоінти, такі як “/kyc2” і “/kyc3”, дозволяють подавати документи та інформацію для вищих рівнів верифікації. Це структуроване відокремлення запитів дозволяє системі легко масштабуватися та обробляти різноманітні запити ефективно.

З метою забезпечення безпеки даних, контролер використовує JwtAuthGuard, який перевіряє аутентифікацію користувачів перед обробкою їх запитів. Це забезпечує, що лише аутентифіковані користувачі можуть взаємодіяти з системою, знижуючи ризик несанкціонованого доступу до чутливої інформації.

Крім того, контролер використовує AnyFilesInterceptor, який дозволяє забезпечити безпечне завантаження файлів користувачів. Цей интерцептор фільтрує завантажені файли, приймаючи лише ті, що відповідають визначеним типам (наприклад, зображення у форматах JPEG або PNG). Такий підхід допомагає уникнути завантаження потенційно шкідливих файлів, що може бути використане для атак на сервер.

Використання технології мультипарт-форми, що обробляється контролером, є критично важливим для роботи з файлами. Контролер налаштований так, щоб обробляти багаточастинні форми, дозволяючи користувачам завантажувати декілька файлів одночасно, що є зручним і необхідним для комплексної верифікації документів KYC.

У підсумку, KYCController відіграє ключову роль у взаємодії користувачів із системою, дозволяючи безпечно та ефективно збирати, обробляти та верифікувати дані. Цей контролер є основним зв'язком між користувачем і сервером, забезпечуючи високий рівень безпеки та зручності для кінцевих користувачів системи

Клас FaceService, який входить до складу “kyc-management.service”, є важливим елементом для роботи з біометричною верифікацією користувачів в системі KYC. Цей сервіс забезпечує обробку зображень облич за допомогою бібліотеки face-api.js, що є однією з провідних бібліотек для розпізнавання облич в JavaScript. Застосування таких технологій в KYC процесах забезпечує додатковий рівень безпеки, перевіряючи, що особа, яка подає документи, дійсно є власником цих документів.

При ініціалізації сервісу “FaceService” відбувається завантаження необхідних моделей нейронної мережі з локального сховища. Ці моделі включають

ssdMobilenetv1, яка відповідає за визначення облич на зображенні, faceLandmark68Net, яка ідентифікує ключові точки на обличчі, та faceRecognitionNet, що забезпечує порівняння облич. Це дозволяє системі ефективно розпізнавати обличчя на різних зображеннях та порівнювати їх між собою.

Однією з основних функцій FaceService є метод bufferToTensor, який перетворює буфер зображення на тензор, що може бути використаний для подальшого аналізу у face-api.js. Цей метод використовує бібліотеку canvas для створення об'єкта канвасу, на який наноситься зображення. Після цього зображення перетворюється на тензор, що є необхідним для виконання обчислень пов'язаних з нейронними мережами.

Метод compareFaces відповідає за порівняння двох облич. Він використовує попередньо зазначене зображення з локального сховища як базове (для порівняння), і зображення отримане від користувача. Обидва зображення перетворюються на тензори, детектуються особливості облич і порівнюються за допомогою евклідової відстані між дескрипторами облич. Якщо розрахована відстань між обличчями менша за певний поріг, то вважається, що обличчя належать одній і тій же особі.

Цей метод важливий для забезпечення відповідності між документами, які подаються користувачем, та його особою, що є критично необхідним для забезпечення високого рівня довіри та безпеки у фінансових та юридичних транзакціях. Цей процес не тільки допомагає запобігати шахрайству, але й підтверджує легітимність дій користувача у системі.

Сервіс FaceService, який використовується в “kyc-management.service”, є важливим елементом для роботи з біометричною верифікацією користувачів в системі KYC. Цей сервіс забезпечує обробку зображень облич за допомогою бібліотеки face-api.js, що є однією з провідних бібліотек для розпізнавання облич в JavaScript. Застосування таких технологій в KYC процесах забезпечує додатковий

рівень безпеки, перевіряючи, що особа, яка подає документи, дійсно є власником цих документів.

При ініціалізації сервісу “FaceService” відбувається завантаження необхідних моделей нейронної мережі з локального сховища. Ці моделі включають ssdMobilenetv1, яка відповідає за визначення облич на зображенні, faceLandmark68Net, яка ідентифікує ключові точки на обличчі, та faceRecognitionNet, що забезпечує порівняння облич. Це дозволяє системі ефективно розпізнавати обличчя на різних зображеннях та порівнювати їх між собою.

Однією з основних функцій FaceService є метод bufferToTensor, який перетворює буфер зображення на тензор, що може бути використаний для подальшого аналізу у face-api.js. Цей метод використовує бібліотеку canvas для створення об'єкта канвасу, на який наноситься зображення. Після цього зображення перетворюється на тензор, що є необхідним для виконання обчислень пов'язаних з нейронними мережами.

Метод compareFaces відповідає за порівняння двох облич. Він використовує попередньо зазначене зображення з локального сховища як базове (для порівняння), і зображення отримане від користувача. Обидва зображення перетворюються на тензори, детектуються особливості облич і порівнюються за допомогою евклідової відстані між дескрипторами облич. Якщо розрахована відстань між обличчями менша за певний поріг, то вважається, що обличчя належать одній і тій же особі.

Цей метод важливий для забезпечення відповідності між документами, які подаються користувачем, та його особою, що є критично необхідним для забезпечення високого рівня довіри та безпеки у фінансових та юридичних транзакціях. Цей процес не тільки допомагає запобігати шахрайству, але й підтверджує легітимність дій користувача у системі.

4.2.2 Клієнтська реалізація

На стороні клієнтського додатку для користувачів, що використовує бібліотеку React і організована у вигляді модуля KYC. Модуль відповідає за окрему одноіменну сторінку (рис. 4.4 – рис. 4.6) і інкапсулює всю необхідну логіку для виконання специфічних дій, таких як реєстрація або вхід користувачів.

Модуль включає кілька підпапок і файлів:

– компоненти (components), які виконують роль візуального представлення функціоналу на сторінках входу та реєстрації;

– хуки (hooks) кастомні визначають і керують логікою взаємодії з API та управлінням станом форм. Наприклад, хук “useKycLevel1” інкапсулює логіку необхідну для роботи із станами елементів і відправлення запитів на сервер, що стосуються першого рівня верифікації.

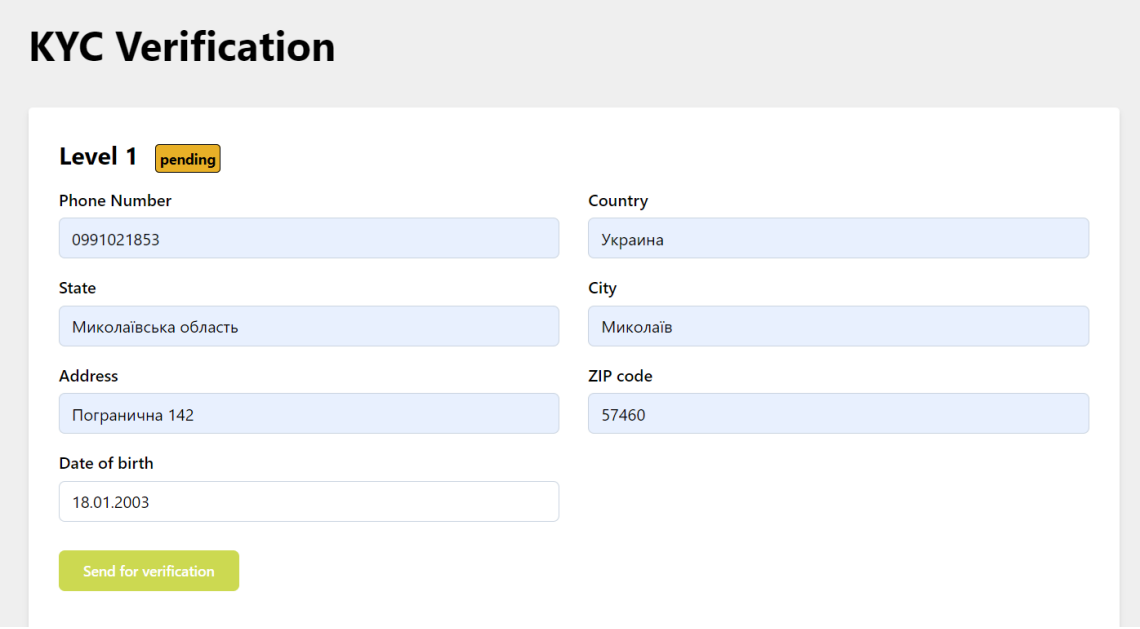

Рисунок 4.4 – UI сторінки форми для надання даних першого рівня

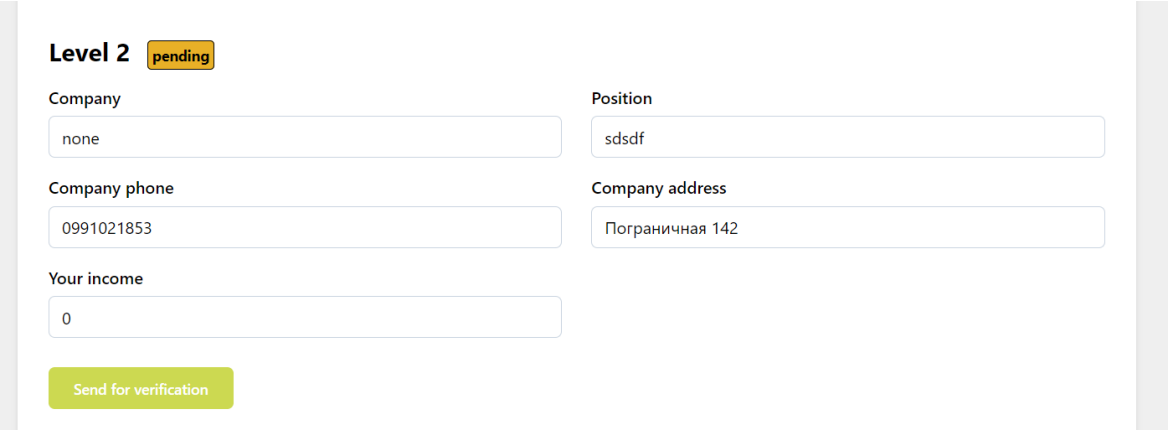

Рисунок 4.5 – UI сторінки форми для надання даних другого рівня

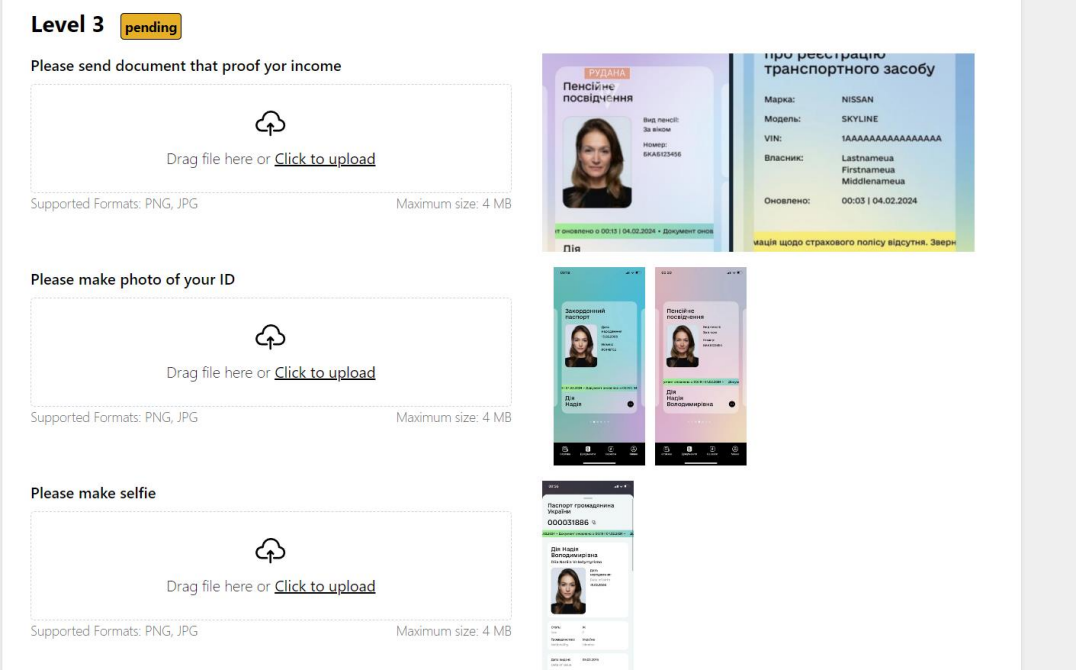

Рисунок 4.6 – UI сторінки форми для надання даних першого рівня

“KYCService” є основним сервісом для роботи з API. Цей сервіс використовує HttpService для відправлення запитів до сервера, які включають перевірку стану сесії користувача, реєстрацію нових користувачів, вхід в систему, активацію акаунту, повторну відправку коду активації, та вихід з системи.

У модулі “kyc” реалізовано компоненти для кожного рівня, кожен з яких використовує форму для збору даних користувача. Хуки керують логікою подання форм, включаючи перевірку валідності даних, відправку даних на сервер через

“kycService” та перенаправлення на сторінку активації після успішної реєстрації. Логіка валідації форми реалізована через validateFormValues, яка забезпечує додаткову перевірку введених даних перед відправкою на сервер.

Ця структура дозволяє чітко розділити логіку обробки даних та візуальне представлення, забезпечуючи легкість підтримки та розширення додатку. Кастомні хуки допомагають ізолювати специфічні процеси аутентифікації, роблячи код більш чистим і зрозумілим.

Для адмінського інтерфейсу клієнту вебдодатку реалізована сторінка із таблицею користувачів (рис. 4.7).

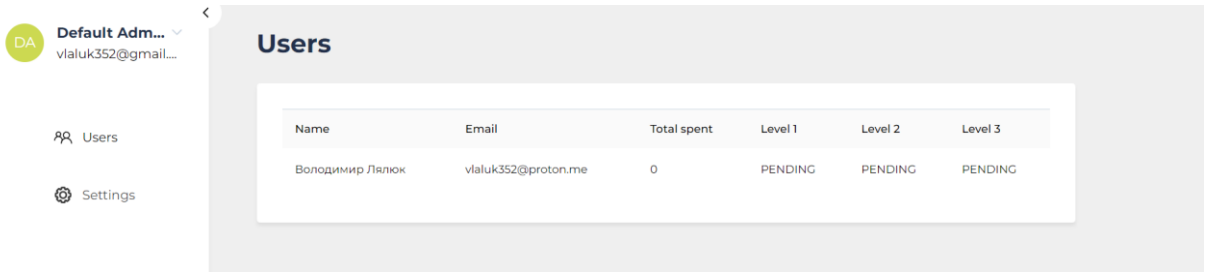

Рисунок 4.7 – UI сторінки із таблицею користувачів

Ця сторінка показує головну інформацію про користувачів. За допомогою неї, адміністратор може відстежити, які користувачі зробили запит на перевірку даних. Модуль включають спеціалізовані компоненти та хуки, які забезпечують вес потрібний функціонал сторінки, наприклад, перевірку фото користувача або зміна статусу верифікації користувачів. По кліку на рядок у таблиці адмін може перейти на специфічну сторінку користувача, на чий рядок було натиснуто (рис. 4.8).

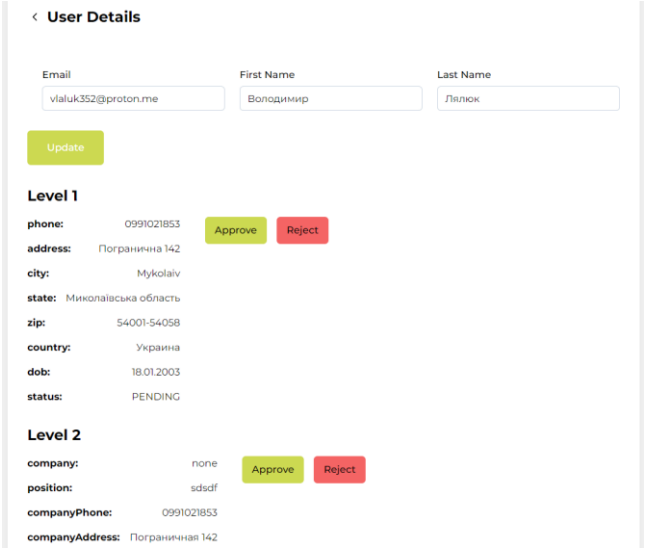

Рисунок 4.8 – UI сторінки адміністративної панелі із інформацією користувача

На цій сторінці також реалізований інтерефс для перегляду і обробки даних KYC-верифікації користувача (рис. 4.9).

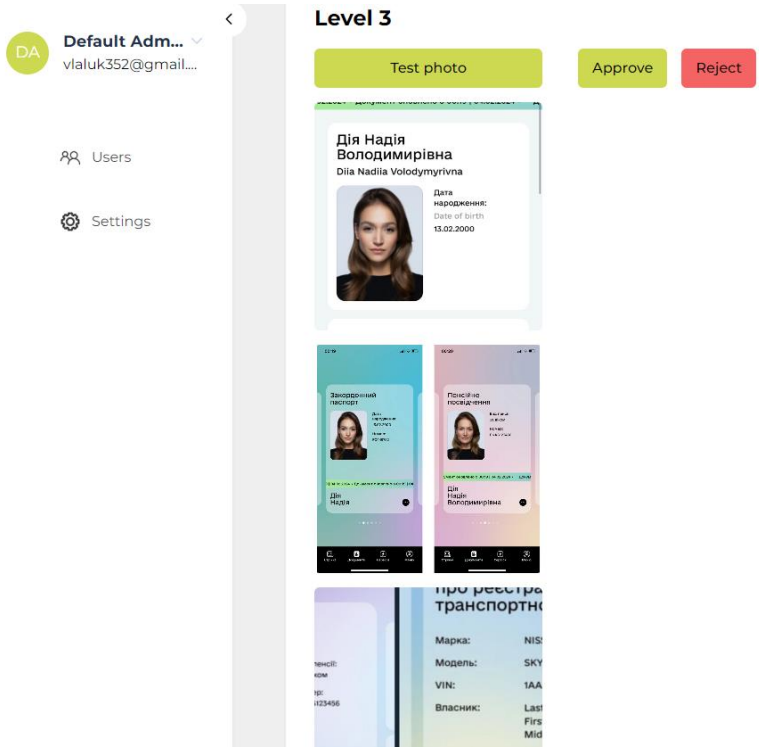

Рисунок 4.9 – UI сторінки адміністративної панелі із інформацією користувача третього рівня

Висновки до розділу 4

У цьому розділі детально розглянуто архітектуру та функціональність аутентифікації та системи верифікації KYC, що включає в себе серверну імплементацію та інтеграцію з контролером. Основним аспектом системи є впровадження каскадної моделі верифікації, що дозволяє ефективно збільшувати рівень перевірки залежно від вимог і ризиків, пов'язаних із конкретними користувацькими операціями. Використання NestJS та Prisma як основних технологічних рішень дозволяє забезпечити надійність, безпеку і масштабованість системи. Модульна структура серверного застосунку сприяє гнучкому розширенню і адаптації системи до змінних вимог ринку та законодавства.

Інтеграція з контролером через використання NestJS дозволяє ефективно управляти процесами KYC на кожному рівні верифікації, забезпечуючи їх автоматизацію та взаємодію з клієнтом. Реалізація взаємодії з базою даних за допомогою Prisma підкреслює високу продуктивність і оптимізацію звернень до бази, що є критично важливим для систем, які обробляють велику кількість даних. Роль FaceService як частини системи, що відповідає за біометричну верифікацію, підкреслює високий технологічний рівень безпеки, забезпечуючи можливість використання сучасних методів обробки зображень та машинного навчання для забезпечення автентичності користувацьких даних.

Загалом, розглянута система демонструє ефективну інтеграцію різноманітних технологій та методик, що спрямовані на забезпечення високого рівня безпеки та відповідності регулятивним вимогам. Впровадження інноваційних технічних рішень та модульний підхід до розробки сприяють адаптивності та масштабованості системи, що є ключовими факторами для успішного використання в широкому спектрі застосувань.

ВИСНОВКИ

Центральне місце у роботі займала розробка такої платформи, що включала використання передових технологій і методів розробки програмного забезпечення. Через інтеграцію KYC процесів з використанням TensorFlow.js для аналізу даних, створено систему, яка автоматично верифікує особистість користувачів, перевіряє їхні документи та звіряє інформацію з міжнародними базами даних про санкції та обмеження. Це значно підвищує безпеку та надійність угод, мінімізуючи ризики та забезпечуючи дотримання всіх необхідних регулятивних вимог.

Розроблено інтуїтивно зрозумілі інтерфейси для кінцевих користувачів і адміністраторів, забезпечили ефективність та швидкість обробки даних завдяки оптимізованій архітектурі серверних та клієнтських застосунків, використовуючи сучасні технології як NestJS для серверного застосунку та React для клієнтських.

Результати цієї роботи не тільки сприяють безпечнішому і прозорішому середовищу для торгівлі товарами подвійного призначення, але й покладають основу для подальших досліджень та розвитку вебплатформ, здатних забезпечувати комплексні перевірки в масштабах глобальної торгівлі. Таким чином, кваліфікаційна робота бакалавра стала значним внеском у розвиток інформаційних технологій у сфері торгівлі товарами подвійного призначення та забезпечення безпеки.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. KYC verification process – 3 steps to know your customer compliance, URL: https://shuftipro.com/blog/kyc-verification-process-3-steps-to-know-your-customer-compliance (Last accessed: 03.06.2024).

2. NestJs documentation, URL: https://docs.nestjs.com/ (Last accessed: 07.06.2024).

3. React documentation, URL: https://react.dev/ (Last accessed: 07.06.2024).

4. Tailwind CSS Documentation, Tailwind CSS, URL: https://tailwindcss.com/docs/ (Last accessed: 07.06.2024).

5. Axios Documentation, Axios, URL: https://axios-http.com/docs/intro (Last accessed: 07.06.2024).

6. Zustand Overview, Zustand Documentation, URL: https://docs.pmnd.rs/zustand/getting-started/introduction (Last accessed: 07.05.2024).

7. TensorFlow.js guide, URL: https://www.tensorflow.org/js/guide (Last accessed: 07.06.2024).

8. React guide, URL: https://react.dev (Last accessed: 12.06.2024).

9. Richards J., Millett P. Practical Web Development. Packt Publishing, 2015. 276 c.

10. Stearns J. N. CSS: The Missing Manual. O'Reilly Media, 2015. 650 c.

11. Flanagan D. JavaScript: The Definitive Guide. O'Reilly Media, 2020. 706 c.

12. West, Ben. "Mastering TypeScript." Packt Publishing, 2019. 372 c.

13. Cantelon, Mike; Harter, Marc; Holowaychuk, T.J.; Rajlich, Nathan. "Node.js in Action." Manning Publications, 2017. 384 c.

14. Meckfessel, Michael. "HTTP/2 in Action." Manning Publications, 2019. 320 c.

15. Haverbeke, Marijn. "Eloquent JavaScript: A Modern Introduction to Programming." No Starch Press, 2018. 472 c.

16. Crockford, Douglas. "JavaScript: The Good Parts." O'Reilly Media, 2008. 176 с.

17. Niederst Robbins, Jennifer. "Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics." O'Reilly Media, 2018. 810 с.

18. Souders, Steve. "High Performance Web Sites: Essential Knowledge for Frontend Engineers." O'Reilly Media, 2007. 170 с.

19. Zakas, Nicholas C. "Understanding ECMAScript 6: The Definitive Guide for JavaScript Developers." No Starch Press, 2016. 352 с.

20. Duckett, Jon. "HTML & CSS: Design and Build Websites." Wiley, 2011. 512 с.

21. Fowler, Martin. "Refactoring: Improving the Design of Existing Code." Addison-Wesley, 2018. 448 с.

22. Skiena, Steven. "The Algorithm Design Manual." Springer, 2008. 730 с.

23. Daigneau, Robert. "Service Design Patterns: Fundamental Design Solutions for SOAP/WSDL and RESTful Web Services." Addison-Wesley, 2011. 352 с.

24. Brown, Kyle; Craig Walls, Craig; Whittle, Gregor Hohpe. "Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions." Addison-Wesley, 2003. 736 с.

25. Tidwell, Jenifer. "Designing Interfaces: Patterns for Effective Interaction Design." O'Reilly Media, 2010. 578 с.