

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики  
Кафедра теорії та технології програмування

**Кваліфікаційна робота**  
**на здобуття ступеня бакалавра**  
за спеціальністю 122 «Комп'ютерні науки»  
на тему:

**Веб застосунок для виявлення шахрайських транзакцій з  
використанням технологій Machine Learning**

Виконав студент 4-го курсу бакалавра  
Ігор БУТЕНКО

\_\_\_\_\_  
(підпис)

Науковий керівник:  
доцент, кандидат фіз.-мат. наук  
Тарас ПАНЧЕНКО

\_\_\_\_\_  
(підпис)

Засвідчую, що в цій роботі немає  
запозичень з праць інших авторів без  
відповідних посилань.

Студент

\_\_\_\_\_  
(підпис)

Роботу розглянуто і допущено до  
захисту на засіданні кафедри теорії та  
технології програмування  
«08» травня 2023 р.,  
протокол № 16  
Завідувач кафедри  
Микола НІКІТЧЕНКО

\_\_\_\_\_  
(підпис)

## РЕФЕРАТ

**Обсяг роботи:** основний текст 50 сторінок, 33 ілюстрації, 1 таблиця, 21 джерело посилання.

ВЕБСАЙТ, МАШИННЕ НАВЧАННЯ, ПЕРЕВІРКА ТРАНЗАКЦІЙ, ШАХРАЙСЬКІ ТРАНЗАКЦІЇ, REACT.JS, TENSORFLOW.

**Актуальність теми:** У сучасному світі електронних платежів зростаюча кількість онлайн-транзакцій створює нові виклики для підприємств і користувачів, оскільки шахраї намагаються зловживати системою.

Використання технологій машинного навчання дозволяє автоматично виявляти підозрілі активності, що допомагає попередити фінансові збитки. Тож дана тема є дуже актуальною у сучасності, яка характеризується швидкою цифровізацією та автоматизацією платіжних засобів.

**Об'єктом даної роботи** є шахрайські транзакції та їх виявлення.

**Предметом роботи** є створення клієнт-серверного веб застосунку для виявлення шахрайських транзакцій.

**Метою кваліфікаційної роботи** є розробка веб застосунку аналізу транзакцій з метою оцінки їх підозрілості на шахрайство.

**Методи розробки:** використання бібліотеки TensorFlow, для створення ML моделі класифікації транзакцій, та технології React JS для створення веб інтерфейсу управління цією системою.

**Результати роботи:** проаналізовано переваги та недоліки конкурентів, та на їх основі створено веб застосунок верифікації транзакцій.

**Структура та обсяг.** Кваліфікаційна робота складається з вступу, двох розділів та висновків.

У першому розділі, розглядається наявні на ринку конкуренти, описується їхні переваги та недоліки, які були виконані та доповнені при створенні системи для збільшення функціоналу кваліфікаційної роботи. розповідається про використані технології системи. А також пояснення призначень та цілій

створення веб застосунку верифікації транзакцій, про вимоги а також планування кваліфікаційної роботи.

У другому розділі розповідається про реалізацію двох модулів, а саме ML моделі та веб інтерфейсу.

У висновку було проаналізовано отриманий результат кваліфікаційної роботи. Також були запропоновані майбутні плани покращення роботи, збільшення ефективності системи та інші ідеї, які можна впровадити.

Загалом, розроблена система буде корисною для платіжних сервісів або фірм, що надають фінансові послуги.

## ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ	6
ВСТУП	7
РОЗДІЛ 1. ОГЛЯД РИНКУ ТА ТЕХНОЛОГІЙ	9
1.1 Огляд наявних на ринку технологій	9
1.1.1 Застосунок «SEON»	9
1.1.2 Застосунок «Sift»	10
1.1.3 Застосунок «Kount»	11
1.1.4 Підсумки	12
1.2 Огляд використаних технологій	14
1.2.1 Технологія React JS	14
1.2.2 Бібліотека MUI React	14
1.2.3 Шаблон Material Dashboard	15
1.2.4 Бібліотека Axios	15
1.2.5 Технологія GitLab	16
1.2.6 Застосунок Draw.io	17
1.2.7 Бібліотека TensorFlow	17
РОЗДІЛ 2. РЕАЛІЗАЦІЯ СИСТЕМИ	19
2.1 Реалізація <i>ML</i> моделі	19
2.1.1 Дослідження існуючих моделей	19
2.1.2 Обрання набору даних для тренування та використання моделі	21
2.1.2.1 Credit Card Fraud Detection	21
2.1.2.2 Credit Card Transactions Fraud Detection Dataset	22

2.1.3 Створення функціонала щодо обробки даних	23
2.1.4 Створення функціонала щодо тренування моделі та перевірки транзакцій	25
2.2 Реалізація веб інтерфейсу	26
2.2.1 Розробити use-case діаграму та макети сторінок	26
2.2.1.1 Use-case діаграма користувача	26
2.2.1.2 Інформаційна сторінка	27
2.2.1.3 Сторінка перевірки транзакцій	28
2.2.1.4 Сторінка перегляду останніх перевірених транзакцій	29
2.2.1.5 Сторінка створення правил	30
2.2.1.6 Сторінка перегляду правил	31
2.2.2 Дослідити та обрати технологію створення веб сторінок	32
2.2.3 Розробити базову сторінку інтерфейсу	33
2.2.4 Підключити інтерфейс до серверної частини	34
2.2.5 Створити основну сторінку з даними	34
2.2.6 Створити сторінку із останніми перевіреними транзакціями	36
2.2.7 Створити сторінку перевірки транзакцій	38
2.2.8 Розробити CRUD інтерфейс для моделі правил	38
2.2.9 Впровадити деякі покращення	42
ВИСНОВКИ	44
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	47

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

БД – База даних;

JS – JavaScript;

AML – Anti Money Laundering, противодействие отмыванию денег;

ML – Machine Learning, машинне навчання;

CRUD – create, read, update, delete — операції створення, читання, оновлення та видалення;

REST – Representational State Transfer, передача репрезентативного стану;

HTTP – HyperText Transfer Protocol, протокол передачі даних, що використовується в комп'ютерних мережах;

API – Application Programming Interface, прикладний програмний інтерфейс;

UI - User interface, інтерфейс користувача;

MUI – Material-UI - Magic User Interface, об'єктно орієнтовна система;

IDE – Integrated Development Environment, інтегроване середовище розробки;

DOM - Document Object Model, незалежний програмний інтерфейс;

HTML - HyperText Markup Language, мова гіпертекстової розмітки;

URL - Uniform Resource Locator, веб-адреса;

ПЗ – Програмне забезпечення.

XGBoost - Extreme Gradient Boosting.

## ВСТУП

Сучасність характеризується стрімкими змінами та інноваційними технологіями, які неперервно проникають у всі сфери нашого життя. Одним з прикладів стала цифровізація та автоматизація платежів. Готівкові розрахунки все більш замінюються електронними та безготівковими операціями, де гроші пересилаються за допомогою онлайн систем. Ця тенденція призводить до збільшення зручності та ефективності платіжних процесів, а це своєю чергою сприяє розвитку глобальної економіки. Завдяки цифровізації та автоматизації платежів ми маємо можливість швидко здійснювати транзакції, керувати фінансами та робити покупки у будь-якій точці світу [1].

Через поширення цифрових платіжних систем, питання безпеки транзакцій постає у центрі уваги. Реальну загрозу для користувачів являють шахрайські транзакції. Хакери та шахраї постійно шукають нові способи обману та крадіжки грошей через електронні платіжні системи.

Для цього компанії розробляють системи виявлення шахрайських транзакцій та їх запобігання. Ці системи використовують комплексний підхід, який може складатись із аналізу даних, машинного навчання, алгоритмів тощо, задля виявлення шахрайських транзакцій та їх подальшого блокування.

Важливість такої системи виявлення та запобігання шахрайським транзакціям є надзвичайно високою. Така система забезпечує захист коштів та мінімізує ризики для користувачів платіжних систем. Розвиток та вдосконалення таких систем стає невід'ємною частиною безпеки та надійності в цифровій епосі.

### **Актуальність роботи та підстави для її виконання.**

Збільшення популярності електронних платежів загострює питання безпеки коштів користувачів.

За статистикою Nilson Report [2] кількість шахрайських транзакцій збільшується за останні роки, тож фірми шукають способи виявлення та запобігання таким транзакціям. Для забезпечення безпеки електронних платежів,

компанії впроваджують різноманітні системи перевірки та аутентифікації, такі як двофакторна аутентифікація, виявлення шахрайської активності, аналіз поведінки користувачів та інші. Помічником у цій темі постають моделі машинного навчання, які можуть навчитись на існуючих даних фірми, визначити шаблони шахрайства та на основі цих шаблонів перевіряти транзакції.

**Об'єктом роботи** є шахрайські транзакції та їх виявлення.

**Предметом роботи** є створення клієнт-серверної системи для виявлення шахрайських транзакцій. Використовуючи технології машинного навчання, а також, технології розробки веб застосунків.

**Методи розробки:** використання технологій та бібліотек для створення моделі машинного навчання для розпізнавання шахрайських транзакцій, а також використання технологій та фреймворків для розробки веб інтерфейсу управління та моніторингу цієї системи.

**Мета й завдання роботи.**

Метою роботи була розробка системи виявлення шахрайських транзакцій та веб інтерфейсу для їх перегляду та перевірки.

Для досягнення мети було поставлено та виконано наступні завдання:

- Розробка ML моделі:
  - Дослідження існуючих моделей та обрання найкращої.
  - Обрання набору даних для тренування та використання моделі.
  - Створення функціоналу щодо обробки даних.
  - Створення функціоналу щодо тренування та перевірки транзакцій.
- Розробка веб інтерфейсу:
  - Розробка use-case діаграми користувача та макетів сторінок.
  - Дослідження та обрання технології створення веб застосунків.
  - Реалізація веб інтерфейсу за обраною технологією.

## **РОЗДІЛ 1. ОГЛЯД РИНКУ ТА ТЕХНОЛОГІЙ**

### **1.1 Огляд наявних на ринку технологій**

Сьогодні існують різні системи, розроблені для виявлення та запобігання шахрайським схемам та несанкціонованим транзакціям. Однак, важливо розуміти, що не всі ці системи можуть бути придатними для кожної компанії. Кожна фірма має унікальні потреби та ризики, які потребують індивідуального підходу при виборі та налаштуванні системи виявлення шахрайських транзакцій.

Далі представлені три застосунки наявні на ринку, що зроблені для виявлення шахрайства серед транзакцій [3].

#### **1.1.1 Застосунок «SEON»**

На рис. 1 зображена сторінка застосунку «SEON» [4], який пропонує інструменти та технології для виявлення шахрайських транзакцій. Серед цих інструментів можна виділити такі, як зняття відбитків пристроїв, збагачення даних та перевірку соціальних мереж. Використання цих функцій дозволяє системі виявляти підозрілі транзакції та запобігати шахрайству.

Проте, варто зазначити деякі недоліки цієї системи. Seon не надає інтеграцію на місці для корпоративних клієнтів, що може бути недоцільним для підприємств зі складною структурою або вимогами щодо налаштувань. Також система не має функції перевірки посвідчень особи та рахунків, що може бути важливим для підтвердження легітимності користувачів та забезпечення додаткового рівня безпеки.

Transaction ID	Score	Status	Transaction ID	Merchant	Amount	Date
3fd5d6b747a1	0	APPROVE	a66b36ba19fc	blagovest.igov@raiffaisen.bg		1/21/22 3:28:02
dd3e18fefa51	0	REVIEW	218013469581482	darthvader71		1/21/22 1:54:48
2a409a717f05	0	APPROVE	2180139581482	darthvader71		1/21/22 1:53:00
dc3ce7c042d3	42.25	DECLINE	cd762c81-ee28-43a6-b943-853904d54377	barracuda rowenadesacagita@yahoo.co.id		1/21/22 12:19:5
0c70e822c878	2	APPROVE	cd762c81-ee28-43a6-b943-853904d54375	barracuda rowenadesacagita@yahoo.co.id		1/21/22 12:19:3
3c967f02d17b	70.25	DECLINE	4687513	Exampleaccount sahufigz8ewrgs@gmx.de	25 USD	1/20/22 7:53:47
f6fe4e265a79	69.25	DECLINE	4687513	Exampleaccount sahufigz8ewrgs@gmx.de	25 USD	1/20/22 7:53:18
abd91bbade12	69.25	DECLINE	4687513	Exampleaccount sahufigz8ewrgs@gmx.de	25 USD	1/20/22 7:52:41
d056238a4f68	66.25	DECLINE	4687513	Exampleaccount sahufigz8ewrgs@gmx.de	25 USD	1/20/22 7:34:53
601dc47422c6	65.25	DECLINE	4687513	Exampleaccount sahufigz8ewrgs@gmx.de	25 USD	1/20/22 7:32:25
4c62833a9d33	66.25	DECLINE	4687513	Exampleaccount sahufigz8ewrgs@gmx.de	25 USD	1/20/22 3:40:42
b098398f55a3	67.25	DECLINE	4687513	Exampleaccount sahufigz8ewrgs@gmx.de	25 USD	1/20/22 3:38:45

Рисунок 1 - Приклад вигляду застосунку SEON

Також треба помітити, що SEON є безкоштовним тільки до двох тисяч транзакцій кожного місяця, якщо транзакцій буде більше, доведеться придбати про версію застосунку, яка має обмеження в чотири тисячі транзакцій.

### 1.1.2 Застосунок «Sift»

«Sift» [5] - це швидкозростаючий постачальний антиризикових рішень, який спеціалізується на наданні захисту від шахрайства для різних веб сайтів. Вони пропонують різноманітні продукти, серед яких модуль автентифікації користувачів та захисту від атак, а також інструменти для захисту платежів.

Варто зазначити, що Sift має обмежене розуміння щодо прийнятих рішень і заходів, пов'язаних із результатами перевірки транзакцій. Також обмежені заходи щодо боротьби з відмиванням грошей. Це становить деякі складнощі щодо точності перевірок, а також складнощі для компаній, в яких збільшений ризик відмивання грошей. Неможливість збагачення даних є ще одним з недоліків системи. Злочинці постійно шукають нові способи шахрайства, і неможливість швидко адаптуватись під нові реалії зменшує точність перевірок з часом. На рис.2 можна побачити одну з сторінок застосунку.

**Risky Users**

Payment Protection Score

1-10 of 761

91 j Bentley632@icloud.com

Auto Block Transactions - Users - Mar 21, 2023 4:21 am

Decision

**Risk Summary**

Identity	Order	Devices	Behavior
<ul style="list-style-type: none"> <li>Billing name: frequent changes</li> <li>Phone: connected users</li> <li>Phone is risky</li> <li>Name: known bad</li> </ul>	<ul style="list-style-type: none"> <li>Order: high velocity</li> <li>Shipping address is risky</li> <li>Shipping address: risky location</li> </ul>	<ul style="list-style-type: none"> <li>Device: connected users</li> <li>Device is risky</li> <li>IP: risky location</li> <li>IP / Billing mismatch</li> </ul>	<ul style="list-style-type: none"> <li>Shipping address: frequent changes</li> <li>Billing address: frequent changes</li> <li>Previously failed orders</li> <li>Payment method: frequent changes</li> </ul>

**Order History**

Attempted Orders: 10,778.52+

Authorized Amount: 3,452.43+

Refunds or Voids: —

Chargebacks Received: 3,323.58+

Order	Amount	Items	Payment	CVV / AVS	Billing Name	Txn Status	Failed Txns	Decision
Mar 21, 2023 12037	£31	CREAM CUPID HE... PACK OF 72 RETR...	7827	—	Jeffrey Thompson	X Sale	1 failure	Chase paymentech
Mar 21, 2023 429710	£81	WOODEN PICTUR... +2 more	7757	—	Fannie Otto	✓ Sale	—	—
Mar 21, 2023 106219	€79	GLASS STAR FRO... STRAWBERRY LU...	40240	—	Rosa Majors	✓ Sale	—	—
Mar 21, 2023 60373	€168	JAM MAKING SET ... CIRCUS PARADE L...	3192	—	Sonya Gheen	✓ Sale	—	—
Mar 21, 2023 20683	£275	SET 7 BABUSHKA ... +2 more	7757	—	Pamela Miller	X Sale	1 failure	—

order 1-5 of 100+

**Top Payment Protection Signals**

- Unique Shipping Add in Orders (Month)
  - 20
- Unique billing postal codes (past month)
  - 21
- Unique Billing Names (past month)
  - 21
- Create orders in the last hour
  - 6
- Unique billing postal codes (past day)
  - 7
- Unique billing addresses (past month)
  - 21
- Number of users with the same iOS device ID
  - 9
- Unique payment fingerprints in the last month
  - 21
- Unique billing addresses (past day)

Рисунок 2 - Приклад сторінки підозрілого користувача на «Sift»

### 1.1.3 Застосунок «Kount»

«Kount» [6] - це гігант у сфері виявлення шахрайства, зі значною кількістю клієнтів по всьому світу. Вони спеціалізуються на наданні інноваційних технологій для управління ризиками та обслуговування корпоративних клієнтів. Система включає інструмент для оптимізації для управління зворотними платежами та захисту від відмивання грошей. Див. рис. 3.

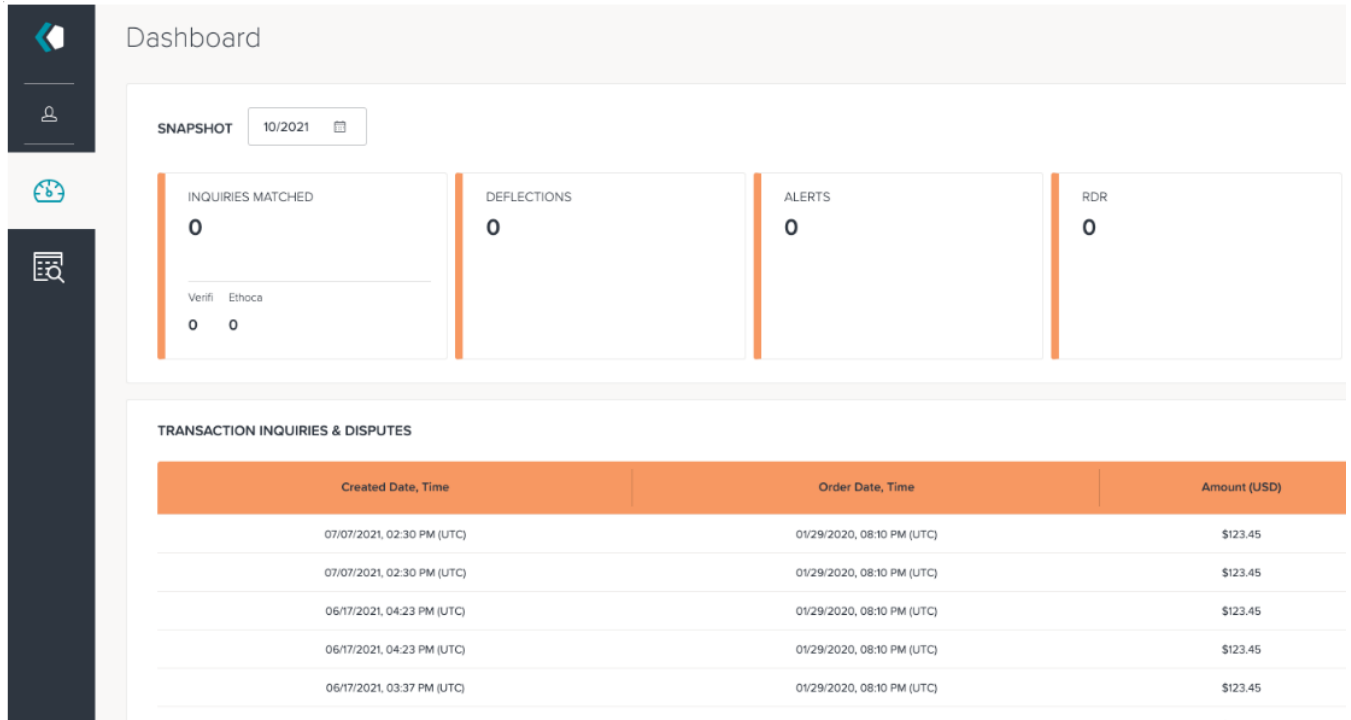


Рисунок 3 - Сторінка останніх транзакцій застосунку «Kount»

Проте, варто відзначити, що система має обмежені можливості в галузі машинного навчання, що може вплинути на ефективність їх системи виявлення шахрайських транзакцій. Недостатнє використання машинного навчання може обмежити їх здатність впізнавати нові шахрайські схеми та адаптуватись до змін методів атак.

#### 1.1.4 Підсумки

Можна відзначити що всі конкуренти, що розглядаються, є провідними постачальниками рішень щодо виявлення шахрайства та захисту від неправомірних зворотних платежів. У них є схожі переваги, такі як широке охоплення галузей, правила, які можна налаштовувати і продукти, які допомагають боротися з шахрайством.

Проте кожна система має свої недоліки, зокрема всі системи мають обмежені можливості машинного навчання. А у деяких з них є обмеження в збагаченні даних, що може негативно позначитися на точності протидії шахрайства.

Важливо зазначити, що жодна система неспроможна ідеально відповідати вимогам всіх фірм. Кожна компанія має унікальні потреби та умови, які можуть вимагати додаткового налаштування або додаткових інструментів. Таким чином фірмам доводиться проводити додаткові роботи з переробки чи доповнення систем, щоб вони повністю відповідали вимогам.

Таким чином у кваліфікаційній роботі було приділено особливу увагу до можливих умов замовника щодо функціоналу та налаштування системи. Також звернено увагу на недолік, такий як неможливість збагачення даних через те, що шахраї кожного дня знаходять нові засоби, які допомагають обходити вже існуючі правила моделей. Було розроблено сторінку із статистикою та діаграмами, а також сторінку яка дозволить верифікувати транзакції. Ця верифікація не надає остаточний результат, як шахрайська транзакція чи ні, а надає число, яке характеризує ймовірність шахрайства. Також було розроблено сторінку із останніми перевіреними транзакціями, які можна буде позначити іншим результатом, для майбутнього збагачення даних.

## **1.2 Огляд використаних технологій**

**При розробці системи було використано наступні технології:**

1. Веб технологія React JS.
2. Додаткові матеріали MUI для React.
3. Шаблон Material Dashboard.
4. Бібліотека Axios для створення HTTP запитів на API.
5. Веб платформа GitLab.
6. Додаток для створення діаграм Draw.io.
7. Технологія машинного навчання TensorFlow.

### **1.2.1 Технологія React JS**

*React.js* [7] – це бібліотека JS для швидкої та гнучкої розробки користувацьких інтерфейсів. Технологія використовує компонентний підхід щодо розробки, що дозволяє розробляти незалежні елементи які можна

повторно використовувати в інтерфейсі. Віртуальний *DOM* допомагає покращити продуктивність додатків шляхом ефективного оновлення лише необхідних частин.

Однонаправлений потік даних є ще однією перевагою *React.js*. Зміни даних викликають переписування лише потрібних компонентів, що підвищує продуктивність додатків. *React.js* також має підтримку різних інструментів та бібліотек, таких як *React Router* для керування маршрутизацією та *Axios* для виконання *HTTP*-запитів.

### 1.2.2 Бібліотека MUI React

*Mui-react* [8] є популярною бібліотекою користувацького інтерфейсу для розробки веб додатків, які використовують технологію *React.js*. Ця бібліотека надає готові компоненти та стилі, засновані на дизайні *Material Design* від *Google*. *Mui-react* пропонує широкий набір готових компонентів, таких як кнопки, форми, таблиці, модальні вікна та інші, які легко використовувати та налаштовувати в додатку.

Бібліотека також забезпечує гнучкість налаштування компонентів, реактивне оновлення інтерфейсу та підтримку різноманітних тем та варіантів компонентів. Завдяки активній спільноті розробників, *Mui-react* є надійним інструментом для розробки сучасних та естетично привабливих користувацьких інтерфейсів.

### 1.2.3 Шаблон Material Dashboard

*Material Dashboard* [9] – це адміністративний шаблон, розроблений на базі *material ui*, який призначений для створення сучасних та естетично привабливих панелей управління. Він використовує стилі та компоненти що відповідають принципам *Material Design* [10], та пропонує відповідний дизайн для забезпечення зручної навігації на різних пристроях.

Шаблон надає готові компоненти, такі як графіки, таблиці, форми та панелі, які можна легко налаштовувати відповідно до вимог проекту. Він

спрощує розробку панелі управління системою, оскільки має заздалегідь підготовлені компоненти, які можна швидко використовувати.

Окрім цього, Material Dashboard забезпечує інтеграцію із `mui`, що дозволяє використовувати усі переваги цієї бібліотеки. Він також має добре написану документацію, яка допомагає розробникам швидко ознайомитись із наявною функціональністю шаблону і ефективно її використовувати.

Загалом, цей шаблон дозволяє прискорити розробку панелей управління, забезпечуючи високу якість та стильний дизайн, а також надаючи зручні інструменти та готові компоненти для ефективної роботи. Він підходить для розробки веб інтерфейсів та панелей управління системами.

#### 1.2.4 Бібліотека Axios

*Axios* [11] – це бібліотека JS, яка забезпечує зручний та ефективний спосіб взаємодії з сервером за допомогою `http` запитів. Вона є популярним інструментом, що дозволяє легко виконувати мережеві запити як із браузера, так і з інших середовищ.

Бібліотека пропонує простий API, що дозволяє відправляти різні типи запитів, такі як GET, POST, PUT та DELETE, з передачею параметрів та заголовків. Це дозволяє точно налаштувати взаємодію з сервером та передавати необхідні дані. Axios забезпечує легкість використання та зрозумілість, що робить його ідеальним вибором для виконання запитів у JavaScript застосунках.

Одне з важливих особистостей технології – це підтримка обіцянок, це, в свою чергу, дозволяє зручно працювати з асинхронними запитами та обробляти отримані дані з використанням синтаксису обіцянок. Це спрощує обробку запитів та дозволяє легко управляти помилками, які можуть виникати під час взаємодії з сервером.

#### 1.2.5 Технологія GitLab

GitLab [12] – це інструмент управління розробкою програмного забезпечення, який надає повний цикл розробки в єдиному середовищі. Він

поєднує в собі систему керування версіями Git, а також широкий спектр функціональності для ефективного управління задачами та спільної роботи над проектами.

За допомогою GitLab розробники можуть зручно працювати з кодом, зберігаючи його в репозиторіях Git та використовувати потужні можливості системи керування версіями. Він надає засоби для організації робочих процесів, включаючи створення відстеження задач, управління проблемами, спринтами та релізами. Також, можливість автоматичного процесу збирання, тестування та розгортання коду сприяє надійній розробці програмного забезпечення.

Також, GitLab надає розробникам інструменти інтеграції із хмарними сервісами. Він надає розробнику можливість розгортання застосунку на своєму хмарному сервісі, так і на хмарному сервісі GitLab, що дозволяє обрати найбільш підходящий варіант для своїх потреб.

### **1.2.6 Застосунок Draw.io**

Draw.io [13] – це веб додаток, призначений для створення різноманітних графічних візуалізацій, таких як діаграми, схеми, макети тощо. Він надає зручний спосіб візуалізувати складні концепції у вигляді діаграм.

Додаток пропонує широкий вибір елементів та символів, що дозволяє створювати гнучкі візуальні моделі та відображати зв'язки між об'єктами. Він забезпечує інструменти для маніпулювання моделями на полотні як їхнє розміщення, кольори, стилі тощо.

Одним з ключових аспектів Draw.io є доступність онлайн. Щоб їм користуватись, не потрібно встановлювати додаток, достатньо користуватись їм безпосередньо у веб браузері. Крім того, застосунок підтримує інтеграцію з різними платформами та сервісами такими як Google, що дозволяє зберігати візуалізації в хмарі та обмінюватись їм із іншими.

Загалом, Draw іо є універсальним інструментом для створення графічних візуалізацій з великим набором функціональних можливостей та зручним доступом онлайн. Він допомагає створювати зрозумілі та привабливі графічні візуалізації різноманітних систем та концепцій.

### **1.2.7 Бібліотека TensorFlow**

TensorFlow [14] – це відкрите програмне забезпечення для машинного навчання, що було розроблене компанією Google, яке надає широкий набір інструментів для побудови та навчання моделей штучного інтелекту. Воно володіє розподіленою архітектурою, що дозволяє прискорити процес навчання моделей та забезпечити масштабованість.

Однією з ключових особливостей технології є його підтримка автоматичного диференціювання, що дозволяє швидко обчислювати похідні функцій та використовувати їх у процесі навчання моделей. Це, в свою чергу, спрощує створення складних моделей та оптимізацію їх параметрів.

Загалом, TensorFlow є потужним інструментом машинного навчання, що дозволяє розробникам створювати та навчати складні моделі штучного інтелекту. Він сприяє прискоренню процесу навчання, підтримує диференціювання та надає доступ до широкого набору ресурсів, що допомагають у розвитку та дослідженнях ML моделей.

## РОЗДІЛ 2. РЕАЛІЗАЦІЯ СИСТЕМИ

Реалізація системи складається з двох основних модулів, а саме ML моделі та веб інтерфейсу для моніторингу та перевірки транзакцій за допомогою цієї моделі.

### 2.1 Реалізація *ML* моделі

Мета цього модуля полягає у дослідженні, вивченні та реалізації моделей машинного навчання. Обрання найкращих моделей для верифікації транзакцій а також обробці даних таким чином, щоб їх можна було використовувати в моделях а також покращувати результати.

Розробка була поділена на такі етапи:

1. Дослідження існуючих моделей, та обрання найкращих.
2. Обрання датасету для тренування та використання моделі.
3. Створення функціоналу щодо обробки даних.
4. Створення функціоналу щодо тренування моделі та перевірки транзакцій.

#### 2.1.1 Дослідження існуючих моделей

По перше, було визначено задачу машинного навчання як задачу регресійного аналізу, бо нам треба не визначити чи є транзакція шахрайською чи ні, а знайти оцінку ймовірності шахрайства, тобто значення від нуля до сотні.

Далі треба обрати модель або декілька моделей, які будуть використані при розробці системи.

Дослідивши статтю порівняння моделей машинного навчання [15], було обрано основною моделлю XGBoost для знаходження оцінки шахрайства, яка показала найкращі результати у швидкості навчання та точності результатів.

Див. рис. 4 та рис. 5

## Performance Comparison using SKLearn's 'Make\_Classification' Dataset

(5 Fold Cross Validation, 1MM randomly generated data sample, 20 features)

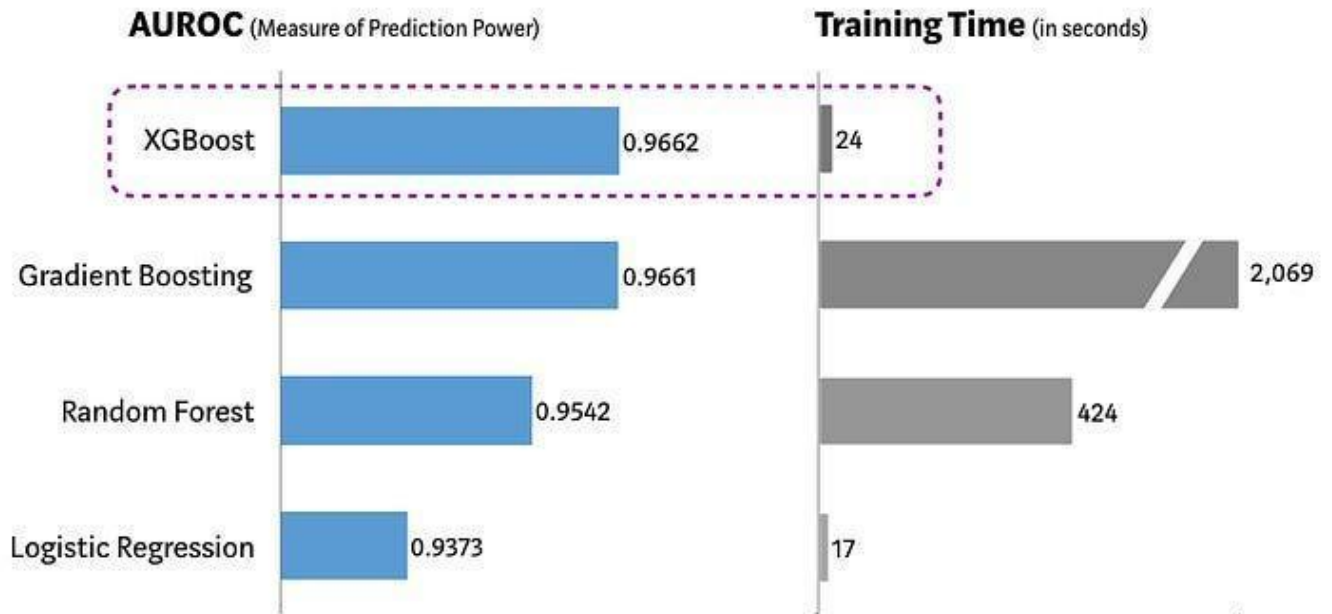


Рисунок 4 - Порівняння моделей

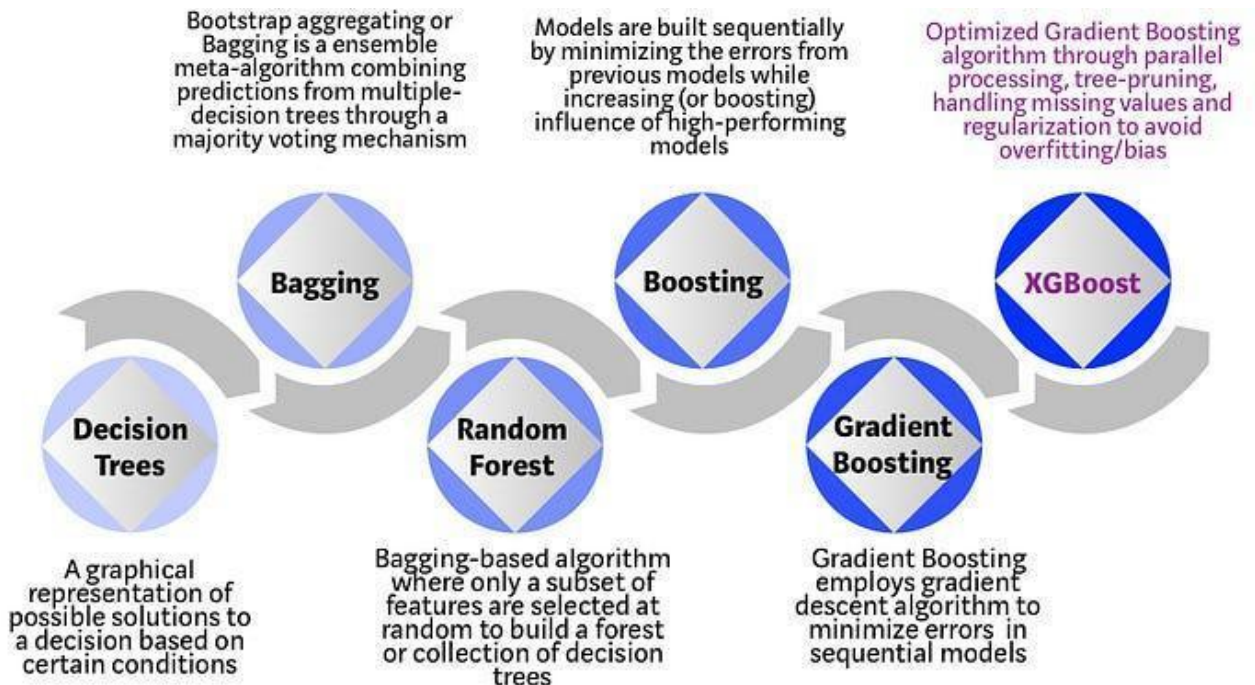


Рисунок 5 - Результат порівняння моделей

Хоча, незважаючи на відверту перевагу моделі, були обрані одразу три моделі, а саме *Random Forest*, *XGBoost* та модель *Transformer* [16], для можливості подальшого комбінування результатів перевірки та знаходження

більш точної оцінки шахрайства. Остання модель, в свою чергу, в перспективі подальшого розвитку проекту може давати навіть кращі результати за *XGBoost* через можливість тонкого налаштування а також його здатність обробляти не тільки числові значення, а також і текстові, що дозволяє використовувати його для перевірки імен користувачів тощо. У перспективі це може дозволити знаходити шаблони імен яких, наприклад не існує, що майже доказує шахрайство транзакції, або знаходження підозрілих шаблонів.

### **2.1.2 Обрання набору даних для тренування та використання моделі**

Для розробки моделі, потрібні данні, за якими модель зможе навчитись і тестуватись. Для цього було знайдено та порівняно два варіанти набори даних із сайту Kaggle[17]. Метою порівняння було виявлення найкращих даних, які будуть забезпечувати максимальну ефективність вивчення моделі а також буде інформативним як для користувача, так і для моделі .

#### **2.1.2.1 Credit Card Fraud Detection**

Цей набір даних [18] має тридцять колонок із значеннями див.рис.6. Всі значення мають числову форму, що буде гарно сприйматись *ML* моделлю, а також скоротить час, затрачений на підготовку даних.

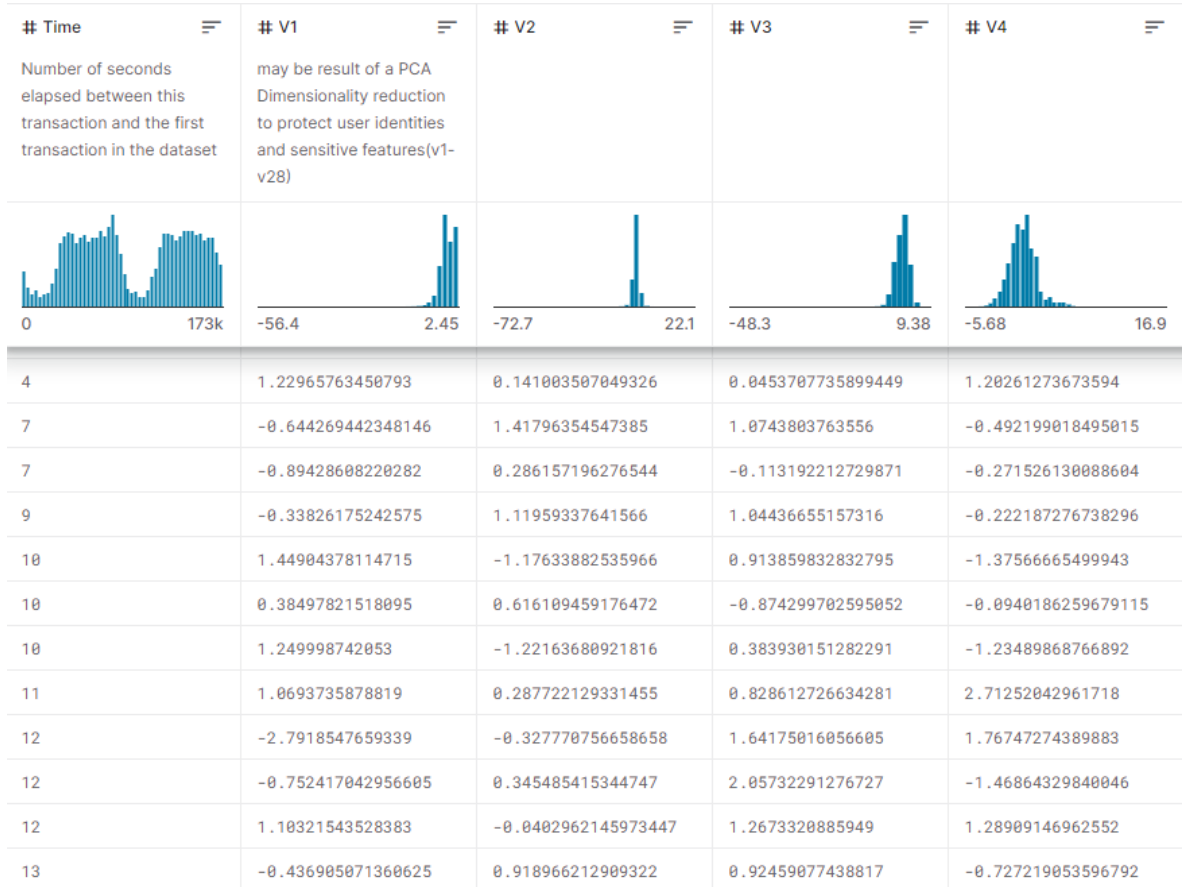


Рисунок 6 - Вигляд першого датасету

Однак, як раз через такі значення, дані не несуть ніякої інформації для користувача. Тобто користувач не буде знати які данні він надає на перевірку, та що впливає на результат.

### 2.1.2.2 Credit Card Transactions Fraud Detection Dataset

Цей набір [19] містить в собі сорок шість колонок із даними див.рис.7. Але, треба зауважити що більшість даних мають текстову форму, та далеко не всі з них можна перевести в числовий вигляд. Але такі дані будуть найкращими в нашому випадку, через їх схожість із реальними транзакціями, це в свою чергу дозволить користувачу розуміти що саме він надає на перевірку, а також покаже реальну точність моделі.

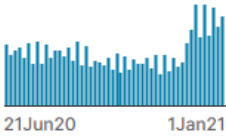


trans_date_trans_...	# cc_num	merchant	category	# amt
		<b>693</b> unique values	gas_transport 10% grocery_pos 9% Other (446796) 80%	
2020-06-21 12:14:25	2291163933867244	fraud_Kirlin and Sons	personal_care	2.86
2020-06-21 12:14:33	3573030041201292	fraud_Sporer-Keebler	personal_care	29.84
2020-06-21 12:14:53	3598215285024754	fraud_Swaniawski, Nitzsche and Welch	health_fitness	41.28
2020-06-21 12:15:15	3591919803438423	fraud_Haley Group	misc_pos	60.05
2020-06-21 12:15:17	3526826139003047	fraud_Johnston-Casper	travel	3.19
2020-06-21 12:15:37	30407675418785	fraud_Daugherty LLC	kids_pets	19.55
2020-06-21 12:15:44	213180742685905	fraud_Romaguera Ltd	health_fitness	133.93
2020-06-21 12:15:50	3589289942931264	fraud_Reichel LLC	personal_care	10.37
2020-06-21 12:16:10	3596357274378601	fraud_Goyette, Howell and Collier	shopping_pos	4.37
2020-06-21 12:16:11	3546897637165774	fraud_Kilback Group	food_dining	66.54
2020-06-21 12:16:20	2242542703101233	fraud_Feil, Hilpert and Koss	food_dining	7.01

Рисунок 7 - Вигляд другого датасету

За результатами порівняння був обраний другий датасет через його схожість із реальними транзакціями.

### 2.1.3 Створення функціоналу щодо обробки даних

Як було сказано раніше, усі моделі, окрім *Transformer* можуть обробляти тільки числові дані. Тож перед тренуванням моделі, треба обробити датасет таким чином, щоб відкинути непотрібні значення, або значення, які не можна перевести в числовий вигляд.

Також, важливим кроком буде змінити деякі значення так, щоб вони надавали максимальну значущість для моделі [20]. Таким прикладом можна навести координати звідки та куди йдуть кошти. Самі координати не дають багато інформації щодо транзакції, а ось якщо ці координати перевести у відстань між місцями звідки та куди йдуть кошти, це дасть як нам, корисну

інформацію. Наприклад транзакція, відстань в якій є усього лише пара кілометрів, буде небезпечною із меншою ймовірністю, ніж транзакція яка йде на відстань від п'яти тисяч кілометрів. Див.рис.8

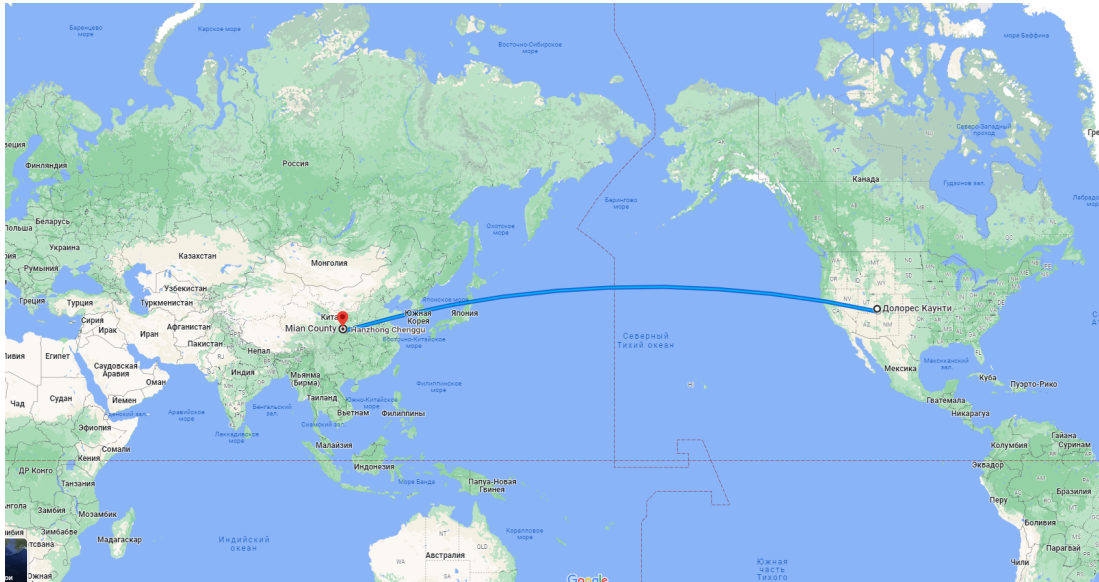


Рисунок 8 - Приклад відстані між точками

Тож було створено декілька функцій обробки даних такі як:

- **Гаряча категорія.** Функція, що створює нову колонку під назвою гаряча категорія, що каже чи є транзакція із категорії, мають найбільший відсоток шахрайств.
- **Час дня.** Функція що переводить звичайну дату в день тижня, рік та час дня.
- **Вихідний.** Функція що створює колонку, яка відображає чи є транзакція зроблена у вихідний день, чи ні.
- **Стать.** Функція, що переводить текстові значення колонки статі з «Female», «Male» у одиницю або нуль.
- **Роки.** Функція, що переводить дату народження користувача у кількість років цього користувача.
- **Відстань.** Функція, що переводить координати транзакції у відстань між початком транзакції та куди вона йшла.
- Багато інших функцій.

### 2.1.4 Створення функціоналу щодо тренування моделі та перевірки транзакцій

У цьому етапі було розроблено функції обміну інформацій із веб інтерфейсом через backend API

А саме розроблено такі функції:

- Train model – що приймає в себе дані, за якими модель, в подальшому, вивчається.
- Verify transactions – ця функція приймає список транзакцій та повертає список оцінок ймовірностей шахрайства кожної транзакції.
- Verify transaction – ця функція приймає транзакцію, та повертає її оцінку ймовірності шахрайства.
- Verify user transaction – ця функція приймає транзакцію юзера, яку треба перевірити, та його останні транзакції.

Важливо детальніше розповісти про останню функцію, а саме чому вона, для перевірки однієї транзакції, приймає декілька останніх транзакцій користувача. Це зроблено із метою дізнатись шаблон поведінки цього користувача, і збільшити ймовірність точної відповіді.

Можна навести два приклади:

Якщо людина кожного дня купляє каву, розраховуючись карткою, то можна зробити висновок, що якщо вона знову купить каву в тому ж місці в той же час, то це навряд чи буде небезпечний переказ. А ось якщо його шаблон поведінки раптом змінився, це може збільшити ймовірність шахрайства.

Інший приклад це відстань між останніми двома транзакціями. Якщо перша транзакція була зроблена в одній країні, а друга через декілька хвилин вже в іншій, або якщо відстань між транзакціями буде великою, а час між ними навпаки маленьким, то можна зробити висновок, що картка була зламана, і треба її блокувати

## 2.2 Реалізація веб інтерфейсу

Веб інтерфейс це модуль системи, який надає користувачу можливість переглядати та перевіряти транзакції, помічати їх як шахрайська, підозріла або безпечна, а також надає повний *CRUD* функціонал щодо моделі правила.

Розробка веб інтерфейсу складалася із таких частин:

1. Розробити use-case діаграму та макети сторінок.
2. Дослідити та обрати технологію створення веб сторінок.
3. Розробити базову сторінку інтерфейсу.
4. Підключити інтерфейс до серверної частини.
5. Створити основну сторінку з даними
6. Створити сторінку із останніми перевіреними транзакціями
7. Створити сторінку перевірки транзакцій.
8. Розробити *CRUD* інтерфейс для моделі правил
9. Впровадити деякі покращення.

### 2.2.1 Розробити use-case діаграма та макети сторінок

#### 2.2.1.1 Use-case діаграма користувача

Важною частиною розробки будь-якого застосунку є його планування. Одним із способів планування є створення use-case діаграм, діаграм класів тощо. Також для початкового розуміння вигляду сторінок можна розробити базові макети.

Для виконання цього завдання було використано застосунок draw.io за допомогою якого було створено наступні діаграми.

В першу чергу було створено use-case діаграму можливостей користувача, яка зображає дії, які користувач може виконати при використанні застосунку.

Див рис 10

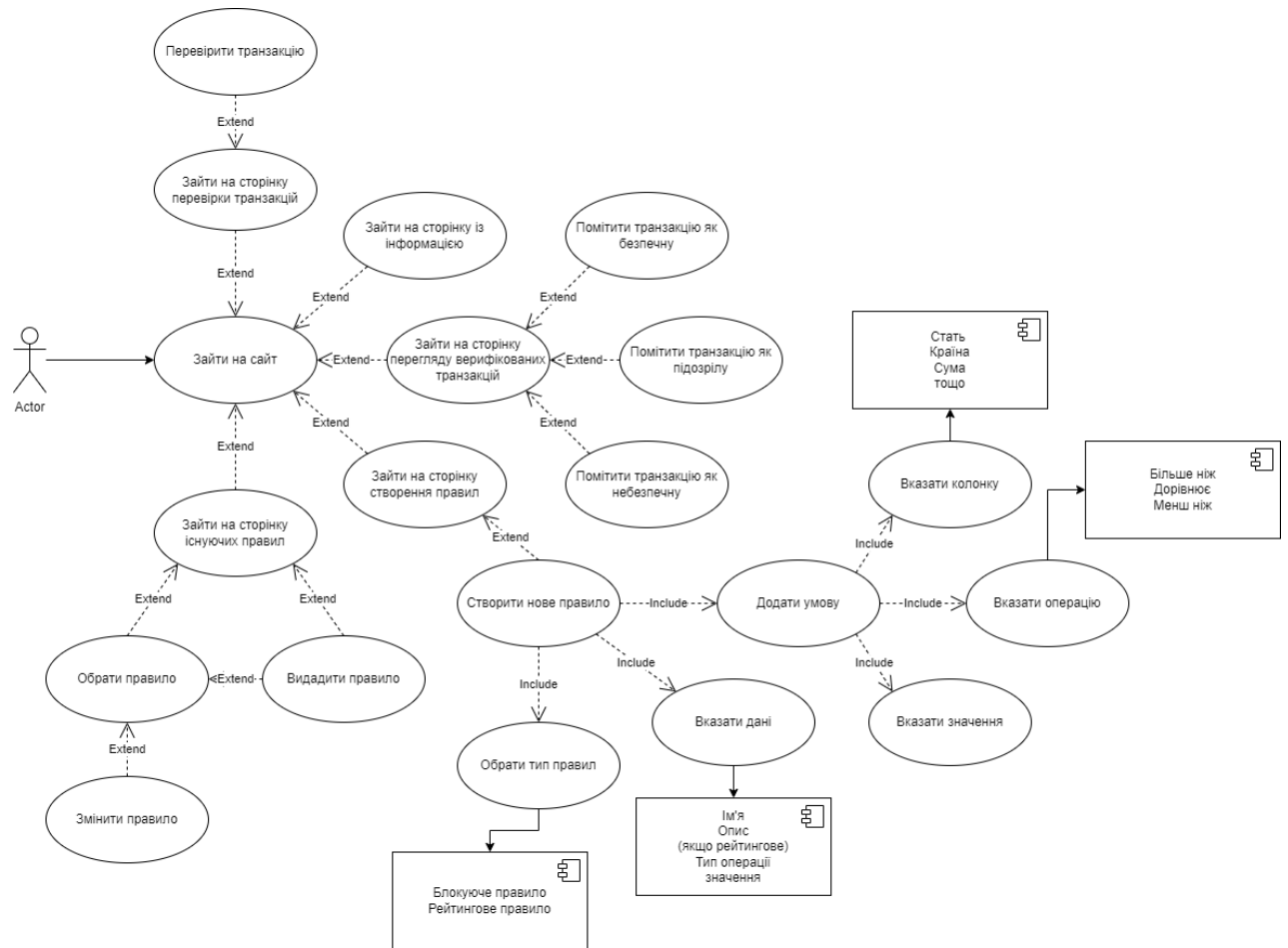


Рисунок 10 - Use-case діаграма користувача

Сторінки, які треба розробити за цим планом:

1. Інформаційна сторінка.
2. Сторінка перевірки транзакцій.
3. Сторінка перегляду останніх перевірених транзакцій.
4. Сторінка створення правил.
5. Сторінка перегляду правил.

### 2.2.1.2 Інформаційна сторінка

На першому макеті див. рис.11 зображено макет вигляду початкової сторінки веб інтерфейсу, а саме сторінку із інформацією. Були створені місця під текстову інформацію, наприклад кількість перевірених транзакцій за місяць. Чотири графіки, які можуть містити інформацію про останні перекази, або

категорії, за якими найбільш частіше трапляються шахрайства. А також колонка під останні транзакції, яка буде динамічно відображати останні перевірені транзакції.

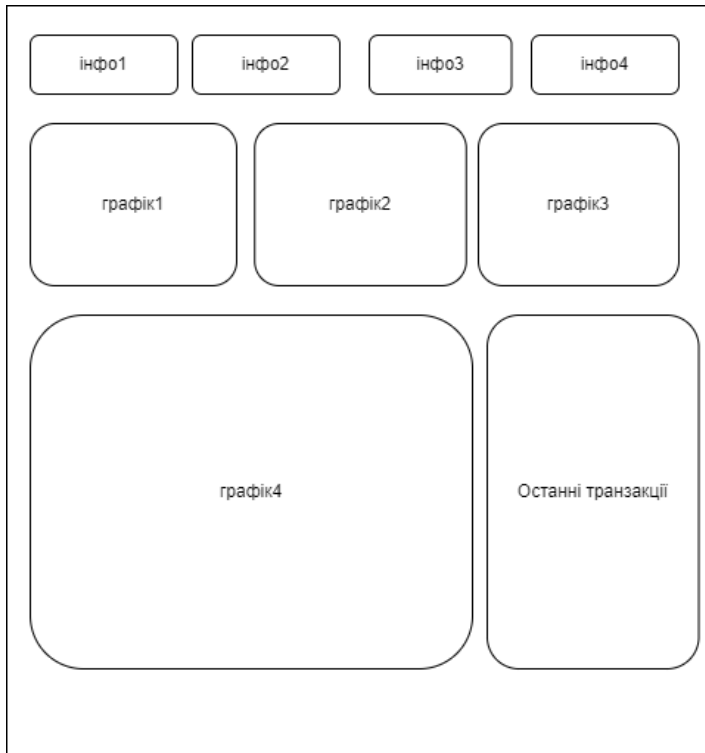


Рисунок 11 - Макет інформаційної сторінки

### 2.2.1.3 Сторінка перевірки транзакцій

На цій сторінці повинні бути зображені користувачі, та їхні транзакції, які можуть бути верифіковані. Макет див. рис.12. При натисканні на кнопку перевірки, транзакція буде відправлятися на backend за допомогою API та буде перевірятись. Після перевірки кнопка буде замінюватись на отримане, після верифікації, значення.

The image shows a wireframe for a transaction verification page, divided into two sections for 'User1' and 'User2'. Each section consists of a header box with the user's name, followed by a list of three transactions. Each transaction row includes a 'Перевірити' (Verify) button on the left and the transaction name on the right. The bottom corners of the transaction lists are rounded.

User1	
Перевірити	Транзакція 1
Перевірити	Транзакція 2
Перевірити	Транзакція 3

User2	
Перевірити	Транзакція 1
Перевірити	Транзакція 2
Перевірити	Транзакція 3

Рисунок 12 - Макет сторінки перевірки транзакцій

#### 2.2.1.4 Сторінка перегляду останніх перевірених транзакцій

Далі було створено макет сторінки із останніми перевіреними транзакціями див. рис.13. Ця сторінка повинна показувати інформацію об останніх перевірених платежів. Також вона повинна надавати можливість ручного позначення транзакції як безпечну, підозрілу або небезпечну. Таким чином це стане основою для майбутнього збагачення даних.

Транзакція1	Безпечна	Підозріла	Небезпечна
Транзакція2	Безпечна	Підозріла	Небезпечна
Транзакція3	Безпечна	Підозріла	Небезпечна
Транзакція4	Безпечна	Підозріла	Небезпечна
Транзакція5	Безпечна	Підозріла	Небезпечна
Транзакція6	Безпечна	Підозріла	Небезпечна
Транзакція7	Безпечна	Підозріла	Небезпечна
Транзакція8	Безпечна	Підозріла	Небезпечна
Транзакція9	Безпечна	Підозріла	Небезпечна
Транзакція10	Безпечна	Підозріла	Небезпечна

Рисунок 13 - Макет сторінки перевірених транзакцій

### 2.2.1.5 Сторінка створення правил

Розроблено макет сторінки створення правил див.рис.5. На цій сторінці повинен бути вибір типу правила, а саме блокуюче правило, або рейтингове. Базова інформація об правилі: його ім'я та додаткова інформація, а також, у випадку якщо правило рейтингове, тоді і сам рейтинг та його операція, така як додавання або множення. Можна додавати або видаляти умови спрацювання правила. Щоб зробити умову, треба вказати колонку транзакції, яка буде перевірятись, операцію перевірки, а саме порівняння колонки із вказаним значенням.

The mockup shows a 'Rule builder' interface. At the top, there is a header bar with the text 'Rule builder' and two buttons: 'Блокуюче' (Blocking) and 'Рейтингове' (Rating). Below this, there are two input fields: 'Ім'я' (Name) and 'Опис' (Description). Underneath, there is a section for 'Умови' (Conditions) with a 'Додати' (Add) button. The bottom part of the interface features two rows of rule configuration. Each row contains four buttons: 'Колонка1' (Column 1), 'Операція1' (Operation 1), 'Значення1' (Value 1), and 'Видалити' (Delete). The second row has identical buttons labeled '2'.

Рисунок 14 - Макет сторінки створення правил

#### 2.2.1.6 Сторінка перегляду правил

Було створено макет сторінки перегляду правил див.рис.6. За цим макетом будуть створені дві сторінки для перегляду блокуючих та рейтингових правил окремо один від одного. На цій сторінці будуть зображені правила, їхні умови для спрацювання а також кнопки, за якими можна буде переглянути, змінювати або видаляти правила.

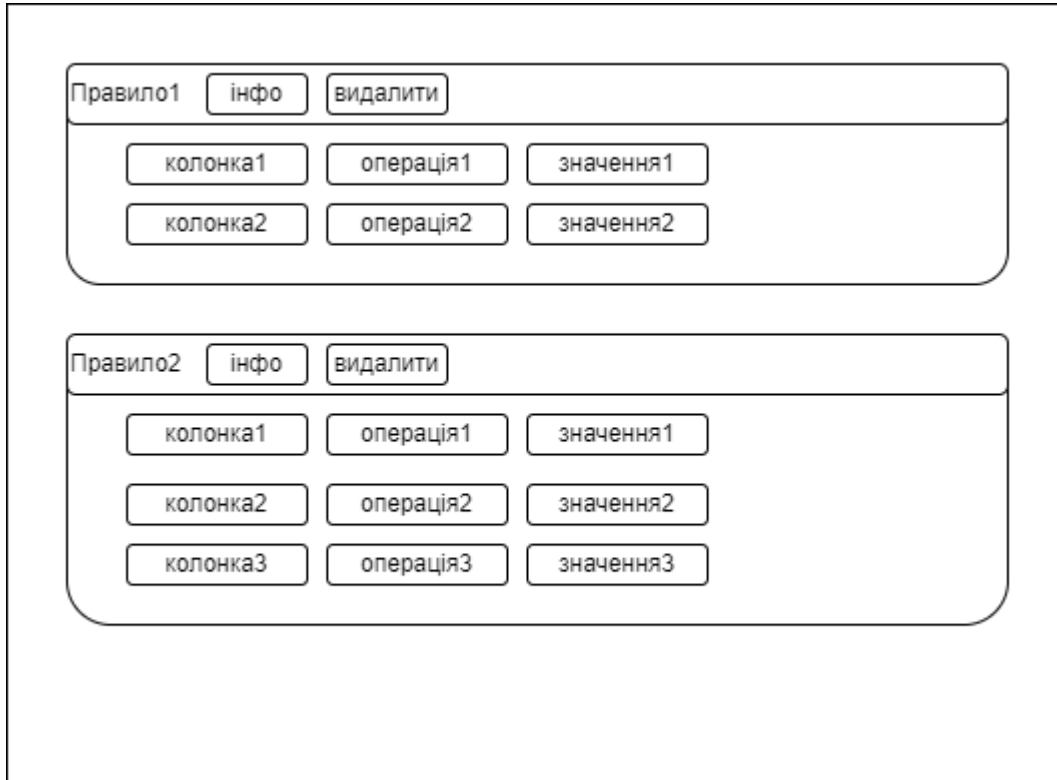


Рисунок 15 - Макет сторінки перегляду правил

### 2.2.2 Дослідити та обрати технологію створення веб сторінок

Для дослідження технологій створення веб сторінок були обрані дві технології, а саме Angular та React. Розглянемо таблицю порівняння цих технологій див. табл. 1.

	Angular	React
Тип	Повноцінний структурний фреймворк	JS – орієнтовна бібліотека
Призначення	Розробка динамічних веб застосунків	Створення інтерактивних UI компонентів
Складність	Може бути складним для початківців	Відносно простий
Рейтинг на <i>GitHub</i>	86 тисяч	203 тисячі

Структура застосунку	Фіксована і комплексна платформа	Гнучка, компонентно-орієнтована
Ідеальні випадки використання	Розробка комплексних корпоративних застосунків	Сучасні веб застосунки із частими змінами даних

Таблиця 1

За розглянутою інформацією, взятою із статті Angular vs React [21], була обрана технологія React JS за багатьма критеріями, такими як, призначення, рейтинг, популярність, складність та випадками використання тощо.

### 2.2.3 Розробити базову сторінку інтерфейсу.

Як базу для React веб інтерфейсу був обраний шаблон Material DashBoard, який є доповненням до бібліотеки MUI, що дозволить швидко і легко створювати та налаштовувати під весь необхідний функціонал. Цей шаблон вже має систему роутінгу між сторінками, бічну панель переходу між сторінками а також сторінку із інформаціями та графіками, яка може знадобитись при створенні сторінки з даними див.рис.16.

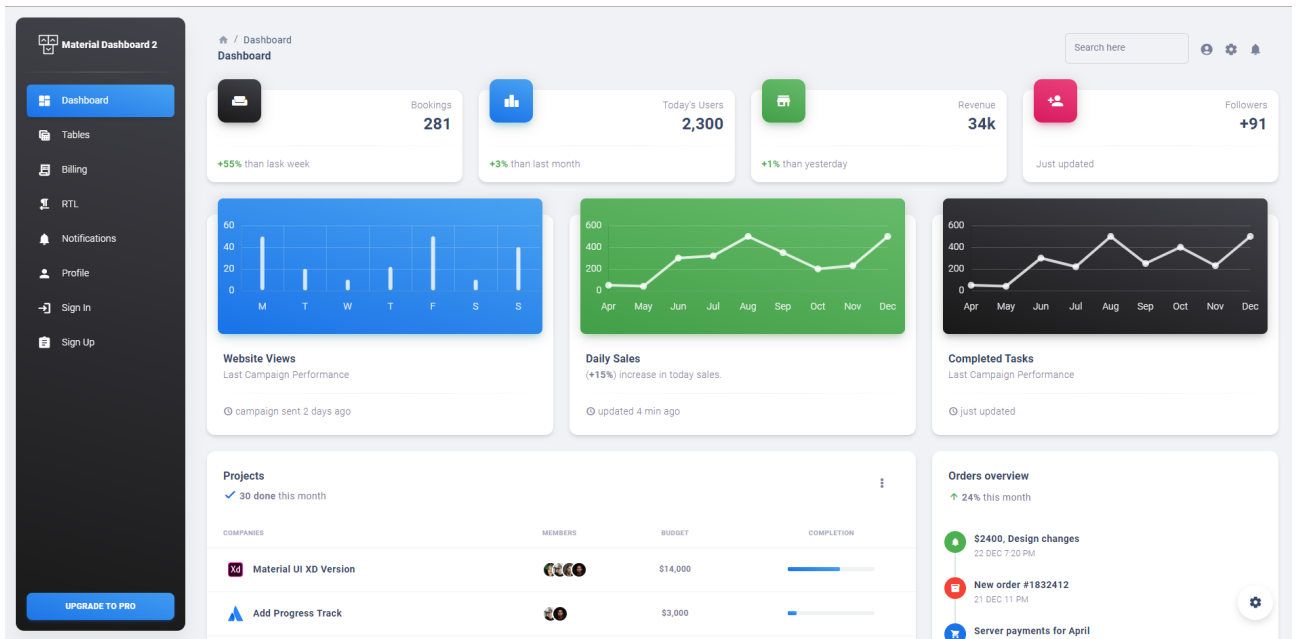


Рисунок 16 - Базова сторінка веб інтерфейсу

На початку видалимо усі сторінки із роутінгу таким чином, щоб вони не відображалися в бічній панелі. Також приберемо кнопку знизу, авторів тощо. Отримаємо пустий макет веб інтерфейсу який вже можна в подальшому доповнювати.

#### 2.2.4 Підключити інтерфейс до серверної частини.

Для обміну інформації із сервером треба зробити HTTP запити на серверну API. Щоб вирішити цю задачу було використано бібліотеку axios, яка дозволяє робити HTTP запити, відловлювати помилки а також взаємодіяти із обіцянками, що в свою чергу робить запити на сервер асинхронними, тобто замість того, щоб витратити час на очікування результату, можна почати їх обробляти, і підставити, коли вони будуть отримані.

Щоб зробити запит в axios на початку вказали протокол, за яким буде проходити приєднання до серверу, а саме https, що робить наш запит захищеним. Далі вказали адресу сервера. Вказали метод запиту, а саме get, post, put або delete, і, за необхідністю, додаткові аргументи.

### 2.2.5 Створити основну сторінку з даними

За основу сторінки візьмемо головну сторінку шаблону, та змінимо її під наші умови. Для цього додали діаграму співвідношення суми шахрайських транзакцій до безпечних та заповнимо її даними з обраного датасету. Див.рис.17

Amount USD to Fraud percentage

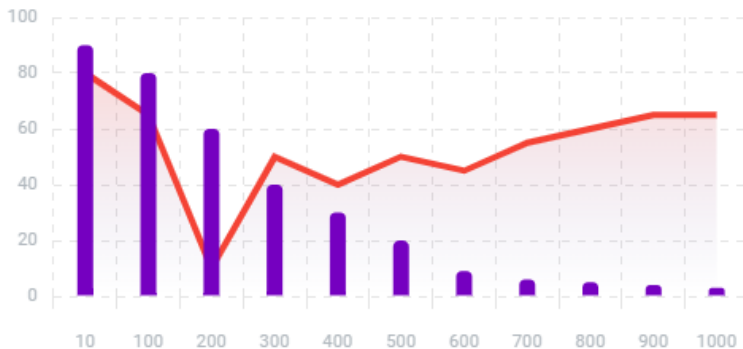


Рисунок 17 - Діаграма співвідношення кількості транзакцій

Одразу ж видно, що зазвичай шахраї роблять більш великі транзакції порівняно із звичайними.

Далі зробили графік, який зображає кількість шахрайських переказів щодо днів тижня. Див.рис.18.



Рисунок 18 - Діаграма кількості шахрайств щодо днів тижня

Можна побачити, що більшість небезпечних транзакцій перепадає на вихідні дні тижня

Також додали графік категорій, на які перепадає більшість шахрайських платежів. Див.рис.19.

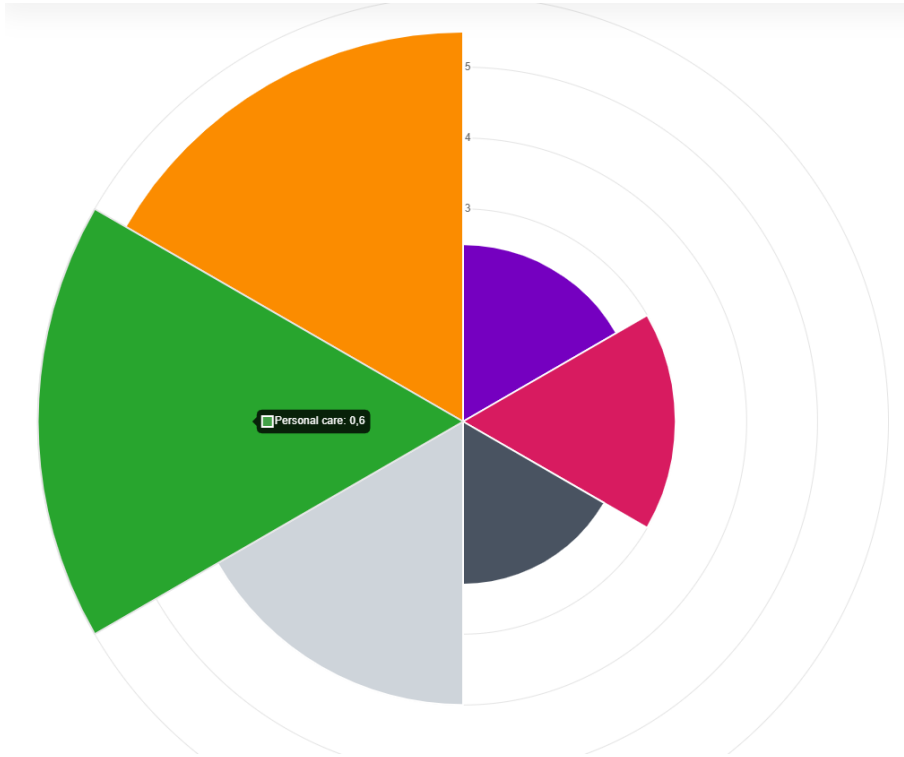


Рисунок 19 - Діаграма категорій

Додали поле із останніми перевіреними транзакціями, які будуть приймати данні із сервера див.рис.21. Де показуємо зеленим кольором ті перевірки, в яких рейтинг склав менше тридцяти, червоним, де більше сімдесяти та оранжевим у інших випадках.

#### Last verified transactions

↑ 24% this month

- ✔ Transaction #cddd362416bb3291825759b92565fb74  
2023.05.23 01:10
- ✔ Transaction #71e72ff15419352217cc0d24c86bfcd0  
2023.05.23 01:10
- ✔ Transaction #af7f43b7e7661fb2e26dca56d67f7473  
2023.05.23 01:10
- ✔ Transaction #2a844672a41f8936988b60ccfb82cf7e  
2023.05.23 01:10
- ✔ Transaction #e4cae85f3740ac7df0fad7b3ddcad715  
2023.05.23 01:10
- ▲ Transaction #34c6238ee0a06cb8c5dbb10793af1189  
2023.05.23 01:10

Рисунок 21 - Компонент останніх перевірених транзакцій

## 2.2.6 Створити сторінку із останніми перевіреними транзакціями

Мета цієї сторінки це показати перевірені платежі та надати можливість користувачу позначати їх на свій розсуд як безпечні, підозрілі або небезпечні. Для цього створено окремий компонент, який буде зображати цю сторінку та підключено його до роутінгу, щоб він з'явився у бічній навігаційній панелі.

Далі, щоб узяти із серверу дані, зробили запит та зробили таблицю, куди будуть завантажуватись данні. Справа додали кнопки для позначки транзакцій. Отримали такий результат див. рис. 22 та рис. 23

FRAUD SCORE	IS FRAUD	TRANS TIME	CATEGORY	AMOUNT USD	GENDER	CITY	STREET	UNIX TIME	ID	MAR
30.36	Not fraud	2020-06-27T03:03:17+03:00	gas transport	77.0	M	Titusville	32941 Krystal Mill Apt. 552	1372302197	34c6238ee0a06cb8c5dbb10793af1189	<input type="checkbox"/>
12.46	Not fraud	2020-06-25T15:52:04+03:00	travel	3.74	M	Titusville	32941 Krystal Mill Apt. 552	1372175524	cddd362416bb3291825759b92565fb74	<input checked="" type="checkbox"/>
0.37	Not fraud	2020-06-26T23:35:36+03:00	shopping net	550.85	M	Titusville	32941 Krystal Mill Apt. 552	1372289736	af7f43b7e7661fb2e26dca56d67f7473	<input type="checkbox"/>
0.2	Not fraud	2020-06-26T19:10:03+03:00	health fitness	22.99	M	Titusville	32941 Krystal Mill Apt. 552	1372273803	e4cae85f3740ac7df0fad7b3ddcad715	<input checked="" type="checkbox"/>
0.05	Not fraud	2020-06-26T16:27:34+03:00	shopping pos	6.61	M	Titusville	32941 Krystal Mill Apt. 552	1372264054	71e72ff15419352217cc0d24c86bfcd0	<input checked="" type="checkbox"/>
0.04	Not fraud	2020-06-26T04:21:10+03:00	misc net	9.9	M	Titusville	32941 Krystal Mill Apt. 552	1372220470	2a844672a41f8936988b60ccfb82cf7e	<input type="checkbox"/>
0.03	Not fraud	2020-06-25T11:01:28+03:00	misc net	34.32	M	Titusville	32941 Krystal Mill Apt. 552	1372158088	d02236fab774c399c99eac7aa781970	<input checked="" type="checkbox"/>
0.02	Not fraud	2020-06-27T04:04:38+03:00	grocery net	79.02	M	Titusville	32941 Krystal Mill Apt. 552	1372305878	7bf2fea95f1ca6a7270dfffb379737e0e	<input type="checkbox"/>
0.01	Not fraud	2020-06-25T12:17:22+03:00	shopping pos	5.48	M	Titusville	32941 Krystal Mill Apt. 552	1372162642	d1bef7db3673cd36498f50b8fbaa4da4	<input checked="" type="checkbox"/>
0.01	Not fraud	2020-06-24T20:42:41+03:00	health fitness	34.66	M	Titusville	32941 Krystal Mill Apt. 552	1372106561	feec47c8e8f86113fe93f54cd754d12c	<input checked="" type="checkbox"/>

Рисунок 22 - Компонент останніх перевірених транзакцій 1

All Verified Transactions						CLEAR ALL		
AMOUNT USD	GENDER	CITY	STREET	UNIX TIME	ID	MARK AS NOT FRAUD	MARK AS SUSPICIOUS	BLOCK
77.0	M	Titusville	32941 Krystal Mill Apt. 552	1372302197	34c6238ee0a06cb8c5dbb10793af1189	MARK AS NOT FRAUD	MARK AS SUSPICIOUS	BLOCK
3.74	M	Titusville	32941 Krystal Mill Apt. 552	1372175524	cd3d362416bb3291825759b92565fb74	NOT FRAUD	SUSPICIOUS	BLOCK
550.85	M	Titusville	32941 Krystal Mill Apt. 552	1372289736	af7f43b7e7661fb2e26dca56d67f7473	MARK AS NOT FRAUD	MARK AS SUSPICIOUS	BLOCK
22.99	M	Titusville	32941 Krystal Mill Apt. 552	1372273803	e4cae85f3740ac7df0fad7b3ddcad715	NOT FRAUD	SUSPICIOUS	BLOCKED
6.61	M	Titusville	32941 Krystal Mill Apt. 552	1372264054	71e72ff15419352217cc0d24c86bfd0	NOT FRAUD	MARK AS SUSPICIOUS	BLOCKED
9.9	M	Titusville	32941 Krystal Mill Apt. 552	1372220470	2a844672a41f8936988b60ccfb82cf7e	MARK AS NOT FRAUD	MARK AS SUSPICIOUS	BLOCK
34.32	M	Titusville	32941 Krystal Mill Apt. 552	1372158088	d02236fafb774c399c99eac7aa781970	NOT FRAUD	MARK AS SUSPICIOUS	BLOCKED
79.02	M	Titusville	32941 Krystal Mill Apt. 552	1372305878	7bf2fea95f1ca6a7270dff379737e0e	MARK AS NOT FRAUD	MARK AS SUSPICIOUS	BLOCK
5.48	M	Titusville	32941 Krystal Mill Apt. 552	1372162642	d1bef7db3673cd36498f50b8fbaa4da4	NOT FRAUD	SUSPICIOUS	BLOCK
34.66	M	Titusville	32941 Krystal Mill Apt. 552	1372106561	feec47c8e8f86113fe93f54cd754d12c	NOT FRAUD	SUSPICIOUS	BLOCKED

Рисунок 23 - Компонент останніх перевірених транзакцій 2

Також із метою тестування системи, була додана кнопка зліва зверху, яка стирає усі результати, дозволяючи повторно їх перевірити.

### 2.2.7 Створити сторінку перевірки транзакцій.

За принципом попередньої сторінки, розроблено компонент та додано його до роутінгу. Створено таблиці із транзакціями, та додано кнопку перевірки транзакції, яка, у випадку, якщо транзакція вже була верифікована, заміниться на оцінку ймовірності шахрайства цієї транзакції. Див рис.15.

FRAUD SCORE	VERIFIED TIME	TRANS TIME	MERCHANT	CATEGORY	AMOUNT	GENDER	STATE	CITY	STREET
VERIFY		2020-06-23T00:18:42+03:00	Zemlak Group	misc net	461.23	M	PA	Morrisdale	4138 David Fall
VERIFY		2020-06-23T14:22:55+03:00	Fadel Inc	health fitness	81.79	M	PA	Morrisdale	4138 David Fall
0.07	2023-05-22T22:26:57.1182415+00:00	2020-06-23T18:12:55+03:00	Pouros-Haag	shopping pos	1.78	M	PA	Morrisdale	4138 David Fall
0.05	2023-05-22T22:26:56.5994073+00:00	2020-06-23T21:11:09+03:00	Pouros-Conroy	shopping pos	5.44	M	PA	Morrisdale	4138 David Fall
VERIFY		2020-06-23T21:12:50+03:00	Willms, Kris and Bergnaum	shopping pos	7.4	M	PA	Morrisdale	4138 David Fall
0.05	2023-05-22T22:26:58.1273265+00:00	2020-06-24T18:34:33+03:00	Terry-Huel	shopping net	212.07	M	PA	Morrisdale	4138 David Fall
VERIFY		2020-06-24T19:44:04+03:00	Konopelski, Schneider and Hartmann	food dining	47.26	M	PA	Morrisdale	4138 David Fall
VERIFY		2020-06-25T03:33:27+03:00	Kovacek, Dibbert and Ondricka	grocery pos	78.28	M	PA	Morrisdale	4138 David Fall
VERIFY		2020-06-25T05:29:41+03:00	Kuvalis Ltd	gas transport	72.6	M	PA	Morrisdale	4138 David Fall
VERIFY		2020-06-25T20:33:55+03:00	Friesen-Ortiz	personal care	35.78	M	PA	Morrisdale	4138 David Fall

Рисунок 24 - Компонент перевірки транзакцій

## 2.2.8 Розробити CRUD інтерфейс для моделі правил

Треба розробити інтерфейс створення правил, їх перегляд, видалення та зміни.

Для було розроблено чотири компоненти:

- Компонент створення правил.
- Компонент перегляду усіх блокуючих правил.
- Компонент перегляду усіх рейтингових правил.
- Компонент редагування правил.

Реалізуємо компонент створення правил. За умови, потрібно мати можливість змінювати тип правила, а також додавати умови, за якими правило буде виконуватись див.рис. 25.

Rules Builder2  
Rules Builder2

Rules Builder Blocking Scoring

Name \* ⓘ

Description

Parts ADD PART

Feature	Operation	Value	
Amount	Biger than	2000	❌
City	Equals	Melitopol	❌
Gender	Operation		❌

SAVE

Рисунок 25 - Компонента створення правил 1

Також, було додано валідацію, яка не дозволить створити правило, у разі неповних даних. Була реалізована можливість зміни типу правила див.рис.26.

Rules Builder2  
Rules Builder2

Rules Builder Blocking Scoring

Name \* ⓘ

Score Type  
Mult

Score  
1.3

Description

Parts ADD PART

Feature	Operation	Value	
Amount	Biger than	2000	❌

SAVE

Рисунок 26 - Компонента створення правил 2

Далі, коли з'явилася можливість створювати нові правила, треба додати можливість їх переглядати, для цього було реалізовано компоненти блокуючих та рейтингових правил. Див. рис. 27 та рис. 28.

DISPLAY NAME	OPERATION	VALUE
Amount	Biger than	1000
Gender	Equals	male
City	Equals	libertad

DISPLAY NAME	OPERATION	VALUE
Gender	Equals	man
Amount	Biger than	1000

DISPLAY NAME	OPERATION	VALUE
Amount	Biger than	2000

Рисунок 27 - Компонента блокуючих правил

FEATURE	OPERATION	VALUE
amt	1	2000

FEATURE	OPERATION	VALUE
city	2	mamaya

FEATURE	OPERATION	VALUE
city	2	mayamy

Рисунок 28 - Компонента рейтингових правил

Як можна побачити, були створені кнопки деталей та видалення правил, які будуть оброблювати *put* та *delete* запити.

Останньою компонентою CRUD є сторінка редагування правил, яку було виконано за принципом створення, але додані додаткові кнопки видалення та повернення на попередню сторінку. Див.рис.20.

The screenshot displays a web interface for editing rules. At the top, a breadcrumb trail shows the path: Home / Rule-Info / 37025190 1cbf-4e35-52ea-08db5a34ee7d. Below this, the rule ID '37025190 1cbf-4e35-52ea-08db5a34ee7d' is displayed. The main section is titled 'Rules Builder' and features two tabs: 'Blocking' (selected) and 'Scoring'. Below the tabs, there are two text input fields: 'Name \*' containing 'rul1' and 'Description' containing 'de1'. A section titled 'Parts' includes an 'ADD PART' button and a list of three rule parts. Each part consists of a 'Feature' field, an 'Operation' field, and a 'Value' field, with a red trash icon to its right. The parts are: 1) Feature: Amount, Operation: Bigger than, Value: 1000; 2) Feature: Gender, Operation: Equals, Value: male; 3) Feature: City, Operation: Equals, Value: libertad. At the bottom, there are three buttons: 'SAVE' (green), 'RETURN' (blue), and 'DELETE' (red).

Рисунок 29 - Компонента редагування правил

Усі компоненти були підключені до системи роутінгу, окрім останнього компоненту, бо, щоб змінити правило, його треба обрати через компоненту правил

### 2.2.9 Впровадити деякі покращення

Незважаючи на те, що весь потрібний функціонал був реалізований, можна було покращити час очікування запитів. Через те, що сервер системи знаходиться у хмарі, усі запити оброблюються не моментально, і треба якось зобразити для користувача процес виконання запиту. Таким чином, в усіх GET запитах, було впроваджено skeletons, які зображали завантаження даних див.рис.30 та рис.31.

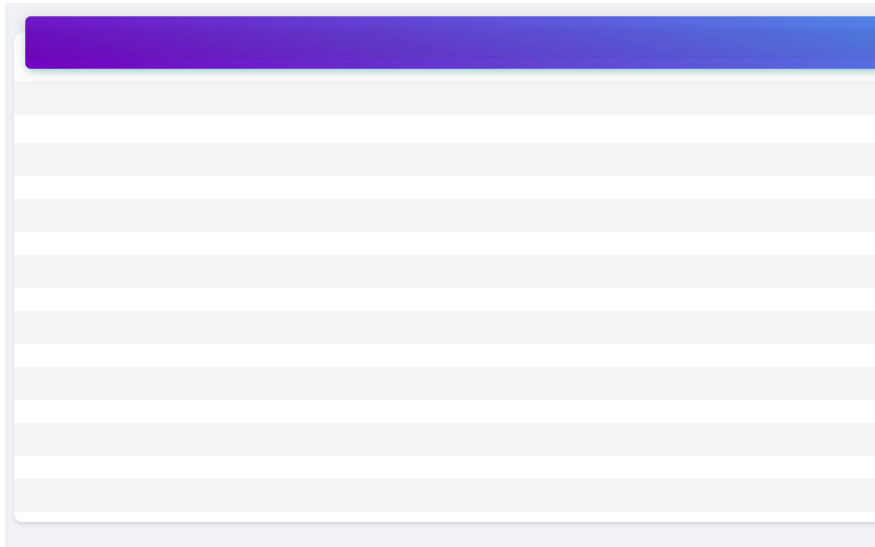


Рисунок 30 - Skeletons в транзакціях

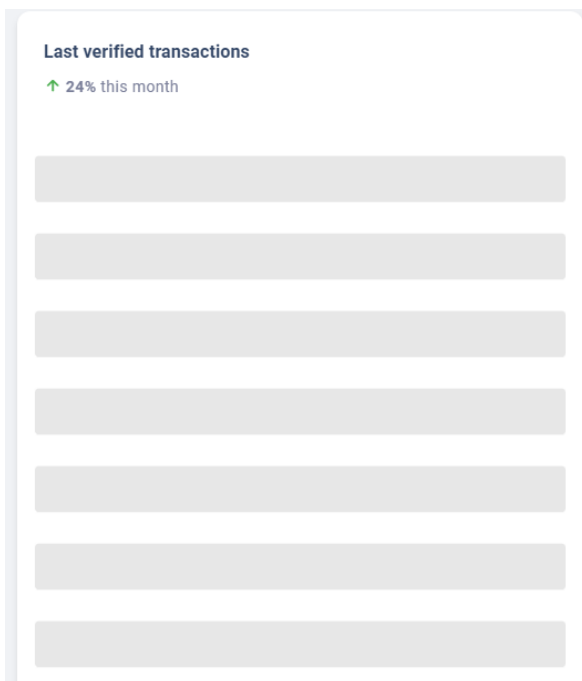


Рисунок 31 - Skeletons в останніх перевірених транзакціях

Далі впроваджено повідомлення про те, що транзакція почала перевірятись у правому нижньому кутку застосунку. Див. рис. 32

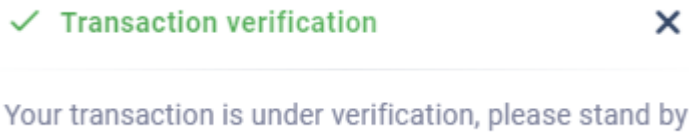


Рисунок 32 - Повідомлення про верифікацію

А також, щоб позбутися випадкового видалення правил, було впроваджено підтвердження своїх дій див.рис.33. Те саме було зроблено і для редагування правил.

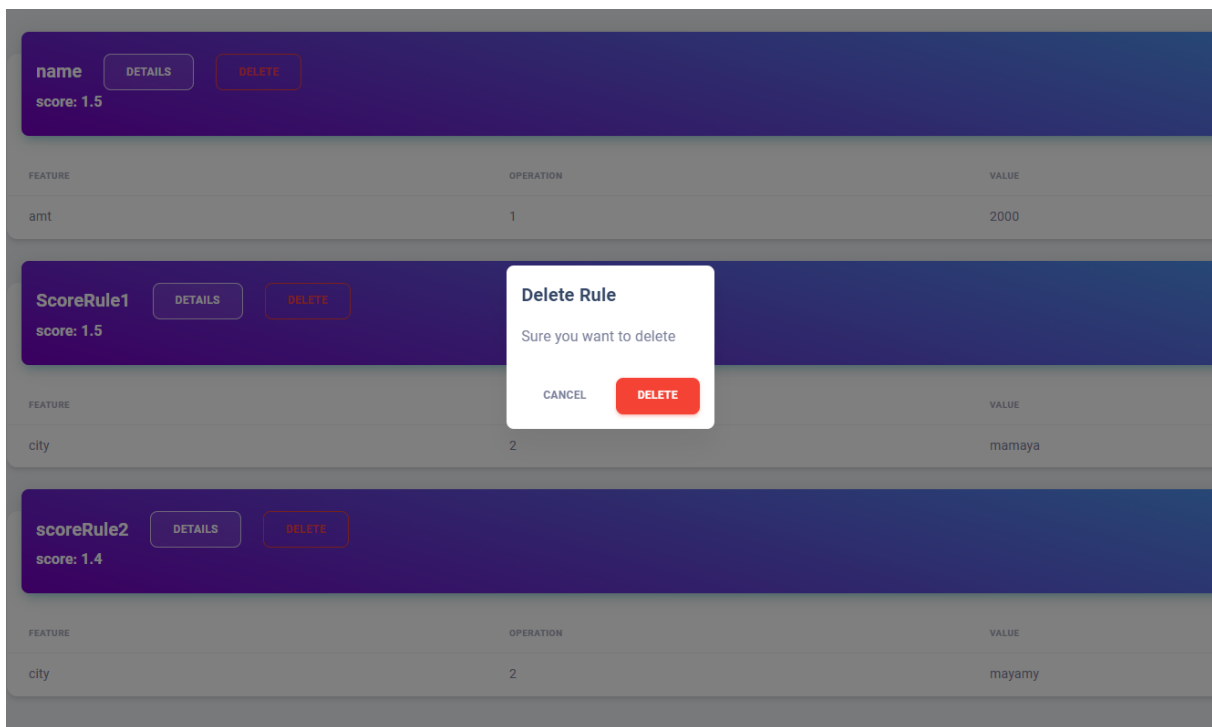


Рисунок 33 - Діалогове вікно видалення правил

## ВИСНОВКИ

На основі поставленої мети розробки – розробки системи виявлення шахрайських транзакцій та веб-інтерфейсу для їх перегляду та перевірки, а також основи для подальшого збагачення даних були виконані наступні завдання:

- Виконані завдання розробки ML моделі:
  - Була визначена задача машинного навчання, а саме задача регресійного аналізу. Досліджені існуючі ML моделі, та обрана найкраща.
  - Було проаналізовано два набори даних, які можна використати для навчання моделі, та обрано той, що більш схожий на реальні транзакції, задля забезпечення розуміння користувача, що саме він перевіряє, а також для надання більш реальної оцінки якості моделі.
  - Створено функціонал щодо обробки даних для можливості їхнього використання в моделі а також для покращення результатів оцінок транзакцій на шахрайства. А саме реалізовано функції: гаряча категорія, час дня, вихідні, стать, роки, відстань та інші.
  - Створено функціонал щодо тренування моделі та її використання. А саме реалізовано такі функції: тренування моделі, перевірки однієї транзакції, перевірки списку транзакцій, перевірки транзакції юзера, що дозволяє дослідити шаблон поведінки користувача, та на основі цього шаблону зробити більш точну оцінку ймовірності шахрайства.
- Виконані завдання розробки веб інтерфейсу:
  - Було розроблено use-case діаграму користувача застосунку, на його основі зроблено макети потрібних сторінок. А саме макети інформаційної сторінки, сторінки перевірки транзакцій, перегляду транзакцій, сторінок створення правил та їх перегляду.

- Було порівняно дві технології фронтенд розробки, а саме React js та angular. За результатами порівнянь було обрано технологію React Js.
- За обраною технологією веб розробки, а також шаблону Material dashboard реалізовано основу веб інтерфейсу.
- Підключено веб інтерфейс до backend API із використанням технології axios.
- Розроблено головні компоненти веб інтерфейсу. Компонента інформаційної сторінки надає декілька інформаційних графіків щодо останніх результатів перевірок. Компонента сторінки перевірки транзакцій надає таблицю із усіма транзакціями, які можна перевірити, натиснувши на кнопку, якщо транзакція вже перевірена, кнопка замінюється на результат перевірки. Компонента сторінки останніх перевірених транзакцій надає таблицю із перевіреними транзакціями, які можна помітити як безпечну, підозрілу або небезпечну, для подальшого збагачення даних. Компонента сторінки створення правил надає можливість створювати блокуючі правила, та правила оцінки. Компонента перегляду правил надає таблицю із усіма правилами, їх типом та умовами правил.
- На кінець, були впровадженні деякі додаткові покращення застосунку, такі як skeletons для усіх get запитів, повідомлення щодо виконання деяких запитів а також діалогові вікна підтвердження видалення або редагування правил.

Нижче приведені деякі плани на майбутнє:

- Покращити *ML* модель шляхом використання одразу декількох моделей і знаходження середнього між ними.
- Розвиток *ML* моделі transformer, незважаючи на те, що *XGBoost* показав себе із гарної сторони, у перспективі подальшого розвитку, трансформер

може бути більш корисним, через можливість обробляти текстові дані, що дозволить знаходити багато залежностей між назвами.

- Знаходження нових ідей, щодо обробки отриманих даних, що в свою чергу може призвести до покращення точності моделі машинного навчання.
- Використовувати правила задля блокування або зміни рейтингу. Хоча *CRUD* для правил вже був реалізованим, правила все що ні як не впливають на верифікацію, у подальшому треба буде одразу блокувати платежі, в яких збіглися умови блокуючих правил, та збільшувати або зменшувати рейтинг транзакцій, при збіганні рейтингових правил.
- Покращення веб інтерфейсу.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. *Digital trends*: стаття, що розповідає про сучасний стан онлайн платежів [Електронний ресурс] – режим доступу до ресурсу: <https://www.paymentsjournal.com/navigating-the-future-top-digital-payment-trends-to-watch/>
2. *Nilson Report*: журнал дослідження шахрайських транзакцій за останні роки [Електронний ресурс] – Режим доступу до журналу: [https://nilsonreport.com/upload/content\\_promo/NilsonReport\\_Issue1209.pdf](https://nilsonreport.com/upload/content_promo/NilsonReport_Issue1209.pdf)
3. *Fraud detection systems*: стаття порівняння існуючих на ринку систем виявлення шахрайства [Електронний ресурс] – Режим доступу до ресурсу: <https://seon.io/resources/comparisons/fraud-detection-and-prevention-software-tools/>
4. *Seon*: офіційний сайт системи seon [Електронний ресурс] – Режим доступу до ресурсу: <https://seon.io/>
5. *Sift*: офіційний сайт системи sift [Електронний ресурс] – Режим доступу до ресурсу: <https://sift.com/>
6. *Kount*: офіційний сайт системи Kount [Електронний ресурс] – Режим доступу до ресурсу: <https://kount.com/>
7. *Fullstack React* : підруч. для веб розробників / [А. Accomazzo, А. Lerner, N. Murray, С. Allsopp, та ін.]. – К: «Fullstack.io», 2017. – 825 с.
8. *MUI React*: бібліотека компонентів для швидкого і зручного налаштування [Електронний ресурс] – Режим доступу до ресурсу: <https://mui.com/>
9. *Material DashBoard*: шаблон, що надає більш детальне налаштування *mui* компонентів [Електронний ресурс] – Режим доступу до ресурсу: <https://www.creative-tim.com/product/material-dashboard>
10. *Mew K. Learning Material Design: master material design and create beautiful, animated interfaces for mobile and web applications* / К. Mew. – К.: Packt Publishing, 2015. – 187 с.

11. *Axios-http*: онлайн документація бібліотеки Axios [Електронний ресурс] – Режим доступу до ресурсу: <https://axios-http.com/docs/intro>
12. *GitLab*: офіційний сайт технології [Електронний ресурс] – Режим доступу до ресурсу: <https://about.gitlab.com/>
13. *Draw.io*: застосунок для створення діаграм [Електронний ресурс] – Режим доступу до ресурсу: <https://app.diagrams.net/>
14. *Moroney L. AI and Machine Learning for Coders: a programmer's guide to artificial intelligence* / L. Moroney. – К.: L. Moroney. 2020. – 390 с.
15. *XGBoost Algorithm*: це стаття, в якій порівнюються різні моделі машинного навчання [Електронний ресурс] – Режим доступу до ресурсу: <https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d>
16. *ML model Transformer*: стаття що розповідає про модель машинного навчання трансформер [Електронний ресурс] – режим доступу до ресурсу: <https://medium.com/inside-machine-learning/what-is-a-transformer-d07dd1fbec04>
17. *Kaggle*: веб застосунок із великою кількістю наборів даних [Електронний ресурс] – Режим доступу до ресурсу: <https://www.kaggle.com>
18. *Credit Card Fraud Detection*: перший набір даних, узятий із сайту Kaggle [Електронний ресурс] – Режим доступу до ресурсу: <https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud>
19. *Credit Card Transactions Fraud Detection Dataset*: другий набір даних, узятий з сайту Kaggle [Електронний ресурс] – Режим доступу до ресурсу: <https://www.kaggle.com/datasets/kartik2112/fraud-detection>
20. *Pyle D. Data Preparation for Data Mining* / D. Pyle. – К.: Morgan Kaufmann, 1999. – 466 с.
21. *Angular vs React*: це стаття, в якій порівнюються дві технології розробки веб застосунків [Електронний ресурс] – Режим доступу до ресурсу:

<https://www.simform.com/blog/angular-vs-react/#:~:text=React%20is%20a%20JavaScript%20library%2C%20whereas%20Angular%20is%20a%20TypeScript,has%20a%20smaller%20bundle%20size.>