Міністерство освіти і науки України
Київський Національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ДОПУСТИТИ ДО ЗАХИСТУ:
В.о. завідувача кафедри кібербезпеки
та захисту інформації
_____ Іван ПАРХОМЕНКО
«17» травня 2024 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи

галузь знань *12 Інформаційні технології*
(шифр і назва галузі знань)
спеціальність *125 Кібербезпека*
(код і назва спеціальності)
освітній ступень *магістр*
освітньо-наукова програма *Кібербезпека*
(назва освітньої програми)

на тему: «Механізм багатофакторної автентифікації у вебзастосунках реляційних СУБД»

Виконавець: студент ІІ курсу, групи КБм-22

Дмитро ЖЕБРАК
(підпис) (Ім'я, ПРІЗВИЩЕ)

	Ім'я, ПРІЗВИЩЕ	**Підпис**
Науковий керівник	Іван ПАРХОМЕНКО	
Нормоконтроль	Лариса МИРУТЕНКО	

Київ 2024

Міністерство освіти і науки України
Київський Національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ЗАТВЕРДЖЕНО:
В.о. завідувача кафедри кібербезпеки
та захисту інформації
_____ Іван ПАРХОМЕНКО
“_____” _____ 2023 року

ЗАВДАННЯ
на виконання кваліфікаційної роботи

спеціальності *125 Кібербезпека*
(код і назва спеціальності)

освітній ступень *магістр*

Здобувача(ки) КБм-22 (група) Жебрака Дмитра Віталійовича (прізвище ім’я по-батькові)

Тема кваліфікаційної роботи Механізм багатофакторної автентифікації у вебзастосунках реляційних СУБД

1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Рішення засідання кафедри кібербезпеки та захисту інформації факультету інформаційних технологій протокол № 5 від 15.11.2023 р.

2. МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Об’єкт досліджень	Процес багатофакторної автентифікації у вебзастосунках реляційних СУБД
Предмет досліджень	Механізм багатофакторної автентифікації у вебзастосунках реляційних СУБД
Мета	Розробка багатофакторної автентифікації у вебзастосунках СУБД, яка забезпечує захист інформації
Вихідні дані для проведення роботи	Автентифікація та її види, методи проходження автентифікації, шифрування, механізм багатофакторної автентифікації у вебзастосунках реляційниз СУБД

3. ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

Наукова новизна удосконалення комплексного механізму багатофакторної автентифікації, який інтегрується безпосередньо з вебзастосунками на базі реляційних СУБД, забезпечуючи захист інформації та зручність для користувачів

Практична цінність полягає в удосконалені механізму багафакторної автентифікації у вебзастосунках реляційних СУБД

4. ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Робота виконана у повному обсязі відповідно до теми.

5. ЕТАПИ ВИКОНАННЯ РОБОТИ

Найменування етапів робіт	Строки виконання робіт (початок-кінець)
Уточнення постановки задачі	17.11.2023 – 29.01.2024
Аналіз літературних джерел	30.01.2024 – 12.02.2024
Дослідження критичної точки компанії	13.02.2024 – 21.02.2024
Аналіз механізмів та принципів багатофакторної автентифікації	22.02.2024 – 26.02.2024
Розгляд систем управління базами даних	27.02.2024 – 04.03.2024
Вибір механізму автентифікації для розробки	05.03.2024 – 10.03.2024
Розробка і впровадження механізму багатофакторної автентифікації у вебзастосунках	11.03.2024 – 06.04.2018
Проведення аналізу отриманих результатів	07.04.2024 – 16.04.2024
Оформлення презентаці	17.04.2024 – 30.04.2024
Оформлення пояснювальної записки згідно методичних рекомендацій	01.05.2024 – 12.05.2024
Подача пакету документів на розгляд ЕК	13.05.2024 – 17.05.2024

6. РЕАЛІЗАЦІЯ РЕЗУЛЬТАТІВ ТА ЕФЕКТИВНІСТЬ

Економічний ефект Можливість інтеграції цього механізму безпеки безпосередньо з реляційними базами даних може сприяти зниженню витрат на розробку та підтримку інфраструктури забезпечення безпеки

Соціальний ефект	Підвищення рівня безпеки вебзастосунків реляційних СУБД сприятиме захисту конфіденційності та приватності користувачів, що позитивно позначаться на загальному рівні довіри до цих систем

7. ДОДАТКОВІ ВИМОГИ

Завдання видав _____ Іван ПАРХОМЕНКО
(підпис) (Ім'я, ПРІЗВИЩЕ)

Завдання прийняв
до виконання _____ Дмитро ЖЕБРАК
(підпис) (Ім'я, ПРІЗВИЩЕ)

Дата видачі завдання: 17.11.2023 р.
Термін подання кваліфікаційної роботи до ЕК 17.05.2024 р.

РЕФЕРАТ

Пояснювальна записка: 90 сторінки, 23 рисунка, 1 таблиця, 39 джерел.

Об’єкт дослідження – процес багатофакторної автентифікації у вебзастосунках реляційних СУБД.

Мета роботи – розробка багатофакторної автентифікації у вебзастосунках СУБД, яка забезпечує захист інформації.

Предмет дослідження – механізм багатофакторної автентифікації у вебзастосунках реляційних СУБД.

Методи дослідження – структурний аналіз, порівняння, системний підхід, моделювання.

В роботі проведено аналіз видів автентифікації та вивчено саме поняття. Досліджено поняття багатофакторної автентифікації у вебзастосунках реляційних СУБД, його створення та методи використання. Проаналізовано можливість використання шифрування при аутентифікації.

Запропоновано удосконалення механізму багатофакторної автентифікації у вебзастосунках реляційних СУБД.

Удосконалено безпечний механізм багатофакторної автентифікації у вебзастосунках реляційних СУБД.

Практичне цінність роботи полягає в удосконалені механізму багатофакторної автентифікації у вебзастосунках реляційних СУБД та можливості шифрування даних при їх використанні.

Результати здійснених у кваліфікаційній роботі досліджень можуть бути використані для впровадження в системи авторизації на будь-яких підприємствах, платформах тощо.

Напрямки подальших досліджень спрямовані на удосконалення даного механзму.

Ключові слова: аутентифікація, багатофакторна аутентифікація, паролі, криптографічне шифрування, алгоритми шифрування, безпечні механізми аутентифікації, реляційні бази даних.

ЗМІСТ

ВСТУП

У сучасному інформаційному суспільстві рівень розвитку ІКТ зростає з кожним днем. В останні роки їх інтенсивне використання та глобальне поширення, а також необмежений доступ населення до Інтернету призвели до експоненціального зростання кількості інформації. У зв'язку з цим виникає потреба подати інформацію в компактному, легкому у використанні, зручному та візуально приємному для користувача вигляді. Це допоможе користувачам швидко й легко знайти одразу те, що вони шукають серед великого об'єму інформації, витрачаючи при цьому мінімум часу та зусиль.

Сучасне життя вже неможливо уявити без різноманітних пристроїв, таких як мобільні телефони чи планшети, які стали неодмінним атрибутом кожного.

Об'єкт дослідження – процес багатофакторної автентифікації у вебзастосунках реляційних СУБД.

Мета роботи – розробка багатофакторної автентифікації у вебзастосунках СУБД, яка забезпечує захист інформації.

Предмет дослідження – механізм багатофакторної автентифікації у вебзастосунках реляційних СУБД.

Методи дослідження – структурний аналіз, порівняння, системний підхід, моделювання.

В роботі проведено аналіз видів автентифікації та вивчено саме поняття. Досліджено поняття багатофакторної автентифікації у вебзастосунках реляційних СУБД, його створення та методи використання. Проаналізовано можливість використання шифрування при аутентифікації.

Для реалізації мети були поставлені такі завдання:

- Проаналізувати принцип роботи компанії ТОВ «Декстра Мінерал»;
- Провести огляд механізмів багатофакторної автентифікації у вебзастосунках реляційних СУБД, їх створення та методи використання.
- Дослідити процес створення механізмів багатофакторної автентифікації у

вебзастосунках реляційних СУБД;

- Розробити безпечний механізм багатофакторної автентифікації у вебзастосунках реляційних СУБД.

РОЗДІЛ 1
АНАЛІЗ КОМПАНІЇ ТОВ "ДЕКСТРА МІНЕРАЛ"

1.1 Організаційної структури та бізнеси процеси ТОВ "ДЕКСТРА МІНЕРАЛ"

Товариство з обмеженою відповідальністю (ТОВ) «Декстра Мінерал» веде свою діяльність на українському ринку аграрних послуг. Підприємство утворено як оптово-роздрібне торговельне підприємство і зареєстроване за адресою: м. Вінниця, вулиця Некрасова, будинок 24. Основна інформація про ТОВ «Декстра Мінерал» представлена в таблиці 1.1.

Таблиця 1.1 – Основна інформація про ТОВ «Декстра Мінерал»

Повна назва	Товариство з обмеженою відповідальністю "Декстра Мінерал"
Скорочена назва	ТОВ "Декстра Мінерал"
ЄДРПОУ	44682062
Юридична адреса	21001, Вінницька область, Вінницький район, місто Вінниця, вулиця Некрасова, будинок 24
Статутний капітал	50000 грн
Основний вид діяльності	46.75 Оптова торгівля хімічними продуктами
Керівник:	Зеленко Денис Олександрович

Предметом діяльності ТОВ «Декстра Мінерал» є неспеціалізована оптова торгівля хімічними продуктами, закупівля та постачання необхідної продукції; продажів товару кінцевому покупцеві; консультаційні послуги; транспортні послуги; надання в оренду й експлуатацію власного чи орендованого майна.

Додатковими видами діяльності підприємства є: діяльність посередників у торгівлі сільськогосподарською сировиною, живими тваринами, текстильною сировиною та напівфабрикатами; технічне обслуговування та ремонт автотранспортних засобів; оптова торгівля іншими машинами й устаткованням; оптова торгівля твердим, рідким, газоподібним паливом і подібними продуктами;

оптова торгівля твердим, рідким, газоподібним паливом і подібними продуктами; інші види діяльності, не заборонені чинним законодавством України, доброчинницька діяльність.

На сьогоднішній день ТОВ «Декстра Мінерал» є одним з лідерів з багаторічним досвідом на ринку аграрних послуг з продажу якісних та ефективних добрив по всій Україні. Компанія також є офіційним дистриб‘ютором мінеральних і мікродобрив найбільших європейських виробників, таких як: Luvena, Siarkopol, Fosfan, Jiva, Grupa Azoty та ін[2].

Основною метою роботи ТОВ «Декстра Мінерал» є отримання прибутку. Прибуток - це ключовий показник організації. ТОВ «Декстра Мінерал» орієнтований на отримання максимального прибутку. Зростання прибутку в організації обумовлюється наступними чинниками: умови добробуту і розвиток гарних відносин серед персоналу; створення нових робочих місць; публічна відповідальність і імідж організації; технічна ефективність, високий рівень продуктивності праці, мінімізація витрат виробництва тощо. Крім цього, іншою метою підприємства є задоволення потреб покупців, надання консультацій щодо правильного використання продукції; створення нових робочих місць, скорочення безробіття, забезпечення місцевого ринку якісною продукцією; розвиток соціальної структури міста та області.

Основним завданням діяльності підприємства є задоволення потреб населення в товарах, продукції, роботах, послугах. Діяльність підприємства на споживчому ринку спрямована на досягнення певних стратегічних і тактичних завдань, які дозволяють забезпечити конкурентноздатність і ефективну господарську діяльність, здатність виживати на ринку.

Організаційно-правова форма підприємства - товариство з обмеженою відповідальністю. Статутний фонд господарського товариства поділений на частки визначених установчими документами розмірів, а учасники несуть відповідальність за своїми зобов'язаннями усім своїм майном; учасники, які повністю сплатили свої внески, несуть ризик збитків, пов'язаних із діяльністю товариства, у межах вартості своїх внесків. Товариство з обмеженою відповідальнісю «Декстра Мінерал»

утворено повністю як приватне підприємство. Товариство є юридичною особою, має самостійний баланс, печатку, штампи, бланки зі своїм найменуванням, зареєстрований у встановленому порядку товарний знак.

Організаційна структура підприємства має лінійно-функціональний тип, який зображено на рисунку 1.1. Персонал підприємства розподілений за відділами, дирекціями та функціональними підрозділами, які підпорядковуються безпосередньо директору підприємства.

Рисунок 1.1 – Організаційна структура управління ТОВ «Декстра Мінерал»

ТОВ «Декстра Мінерал» працює на ринку багато років. Весь цей час підприємство розвивається і продовжує підвищувати обсяги реалізації, перетворюючись в стабільну і ефективну організацію, здатну конкурувати на українському ринку в області продажу добрив, орієнтованої на вимоги клієнтів і високу якість продукції. За весь час роботи торгове підприємство зарекомендувало себе як надійний партнер.

1.2 Аналіз взаємодії підприємства з ринковою інфраструктурою

Проведемо оцінку ринкових можливостей підприємства ТОВ «Декстра Мінерал». Основною продукцією підприємства є добрива, які вирізняються високою якістю та ефективністю. Аналіз динаміки обсягів реалізації продукції ТОВ «Декстра Мінерал» дає змогу стверджувати (рис. 4.1), що протягом досліджуваного періоду спостерігається загальна тенденція збільшення у реалізації продукції на підприємстві.

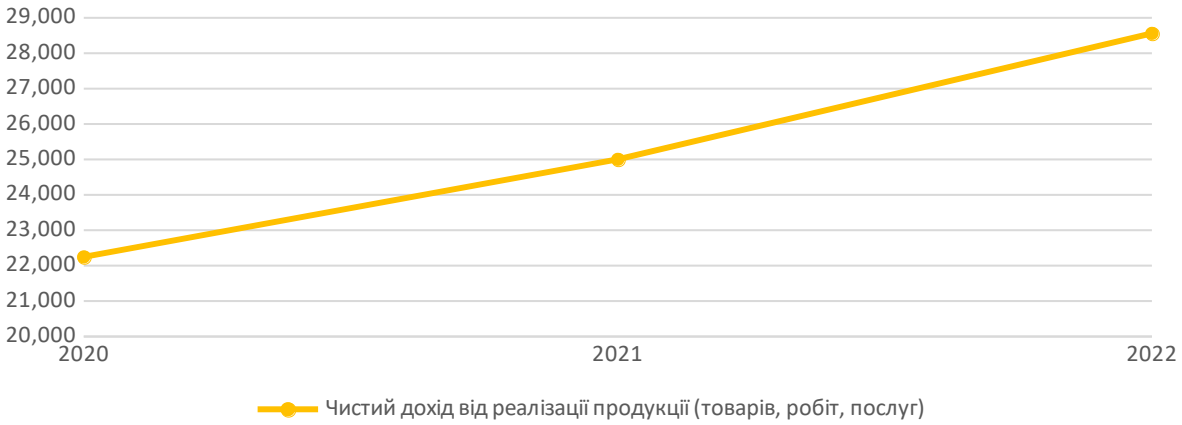

Рисунок. 1.2 – Аналіз динаміки обсягів реалізації продукції ТОВ «Декстра Мінерал»

Повномасштабна війна значним чином вплинула на ринок мінеральних добрив в Україні. Втрата виробничих потужностей та логістичних маршрутів, зміна ланцюжків постачання, скорочення прибутку фермерів – все це і не тільки – стало ключовими факторами змін українського ринку мінеральних добрив. Попри нові ризики та потрясіння, ТОВ «Декстра Мінерал» змогло забезпечити АПК мінеральними добривами та повністю виконали взяті на себе зобов'язання. За підсумками року набільші втрати спостерігались у імпортерів. Великі імпортери-трейдери тимчасово пішли з українського ринку через військові ризики. Це дозволило українським виробникам відчутно посилити свої позиції.

В системі ринкової інфраструктури ТОВ «Декстра Мінерал» приділяє важливе значення налагодженню взаємозв’язків з усіма ланками. Зв’язки підприємства з

зовнішніми контрагентами можна охарактеризувати як стібільні. Партнерами компанії є одні з найбільших підприємств Литви та Німеччини з виробництва мінеральних добрив. ТОВ «Декстра Мінерал» має статус офіційного дистриб‘ютора -мінеральних і -мікродобрив найбільших європейських виробників, таких як: Luvena, Siarkopol, Fosfan, Jiva, Grupa Azoty та ін. Успішне співробітництво з партнерами по бізнесу свідчить про те, що ТОВ «Декстра Мінерал» завжди виправдовуємо їх довіру., що в свою чергу дає змогу гарантувати якісну та ефективну продукцію для своїх клієнтів.

Продукція, що реалізується досліджуваним підприємством, затребувана в основному на ринку далеко прилеглих районів. Проте наявність на підприємстві власного автотранспорту дозволяє пропонувати доставку добрив в інші регіони – за умови попереднього замовлення такої послуги у службі доставки підприємства.

Аналіз стану збутової діяльності підприємства дає змогу визначити низку проблем, які ускладнюють рух торговельних потоків, а саме:

- украй погана якість автомобільних доріг у самому районі та прилеглих до нього. Район доставки має низький показник інтегральної транспортної доступності й, зокрема, нерівномірний розвиток і знос автомобільних доріг;
- перевезення на відстані, більші ніж 200 км, для підприємства нерентабельні. У зв’язку із цим необхідно шукати шляхи зниження сукупних витрат, у тому числі логістичних, що допоможе знаходити шляхи збуту продукції на віддалених ринках

У цілому, комплекс каналів збуту ТОВ «Декстра Мінерал» можна представити таким чином (рис. 4.2). Відповідно у збутовій діяльності досліджуваного підприємства можна виділити такі особливості:

- використання як багаторівневих логістичних каналів розподілу, так і каналів з нульовим рівнем (продаж продукції безпосередньо кінцевому споживачеві без посередника).

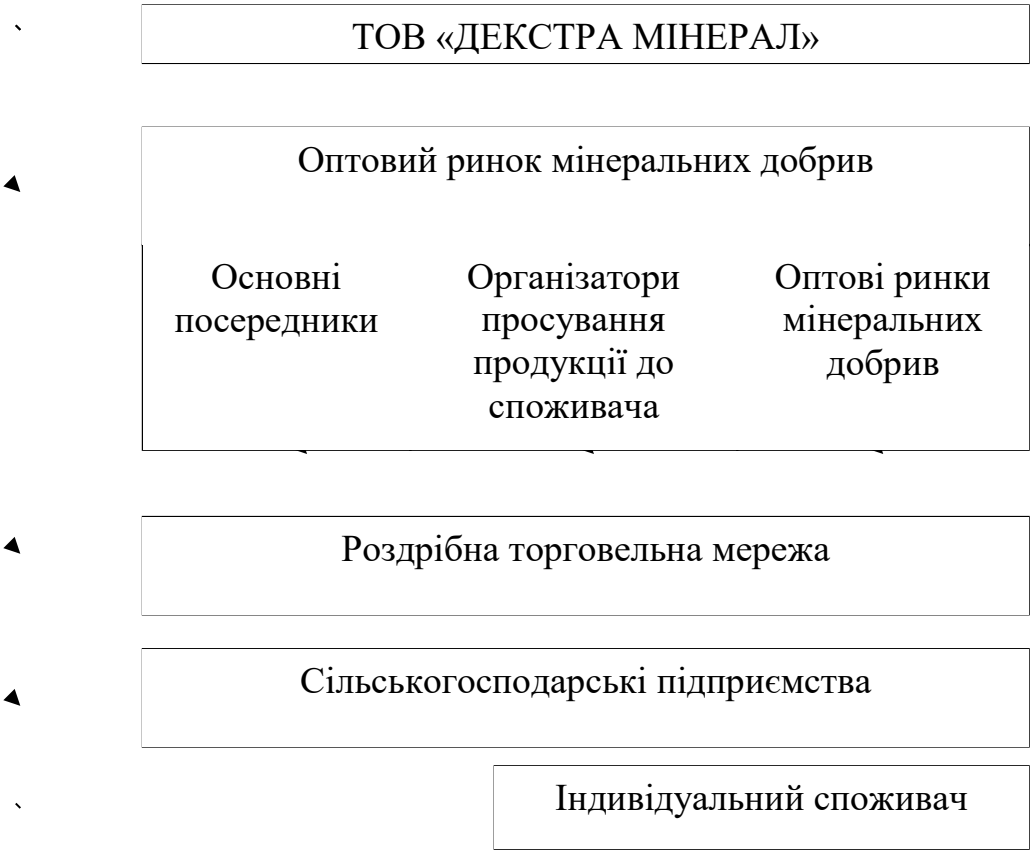

Рисунок.1.3 – Канали збуту продукції на ТОВ «Декстра Мінерал»

- канали розподілу орієнтовані конкретно на продукцію цієї галузі;
- багаторівневі канали є зовнішніми, тому що використовуються посередниками; багаторівневі канали є періодичними за дією, оскільки попит носить сезонний характер, у той час як однорівневі канали діють постійно.
- на підприємстві канали з нульовим рівнем є внутрішніми, бо підприємство реалізує продукцію при такій системі збуту безпосередньо зі своїх виробничих складів. Фірмових магазинів підприємство не має, що в цілому характерно для цієї галузі.

Таким чином, логістичні канали збуту ТОВ «Декстра Мінерал» включають канали розподілення всіх рівнів, основними елементами яких є оптові торговельні організації.

ТОВ «Декстра Мінерал» велику частину товару збуває через оптові ринки. Це пов’язано з тим, що основними споживачами продукції сільськогосподарської промисловості є великі підприємства, на їх частку припадає до 60% споживання.

Таким чином, підприємство за роки роботи зайняло гідне місце серед підприємств України, що реалізують аналогічну продукцію. Основну ставку підприємство робить на високу якість реалізуємої продукції. Підприємство продовжує освоювати нові ринки і розширювати номенклатуру товарів. Крім ТОВ «Декстра Мінерал» на ринку функціонує ще ряд дистриб'юторських підприємств, що реалізують аналогічну продукцію. Серед них ТОВ «Антал Агро», ТОВ «Макош», ТОВ «Ерідон», «Тетра Агро», «Галнафтохім», «Агролайф» та інші.

Підприємство міцно закріплює позиції на ринку і має великі перспективи розвитку. Це пояснюється тим, що ТОВ «Декстра Мінерал» - одне з небагатьох підприємств, що закуповує товар безпосередньо від виробників, реалізує дуже широкий асортимент товарів вищої якості за цінами, які цікаві як оптовим так і дрібно оптовим покупцям. До того ж, ТОВ «Декстра Мінерал» має добре розвинену систему логістики, що дозволяє виробляти доставку товару «від 1 коробки» вчасно і з мінімальними кількісними і якісними втратами.

Подальше підвищення ефективності логістичної системи на підприємстві та застосування принципів логістичного підходу дасть змогу: інтегрувати функції господарських зв’язків з функціями визначення потреби у перевезеннях продукції; скоординувати процеси підготовки до транспортування і безпосереднього перевезення продукції; оптимізувати сукупні витрати на збут шляхом залучення транспортних, комерційних організацій і клієнтів, економічно зацікавлених у підвищенні ефективності функціонування розподільно-збутової системи; скооперувати різні ланки логістичної системи в управлінні товарорухом; раціонально розподілити функції між суб’єктами управління. Подальші дослідження цієї проблеми для підприємства дадуть змогу запропонувати альтернативні варіанти розвитку ефективної логістичної системи для ТОВ «Декстра Мінерал» задля підвищення рівня його конкурентоспроможності.

1.3 Виявлення критичних точок в операційній діяльності компанії

Для виявлення критичних точок в операційній діяльності компанії Декстра Мінерал можна провести аналіз, зосереджений на ідентифікації ключових етапів та областей, які можуть суттєво вплинути на загальний хід бізнес-процесів. Ось кілька можливих кроків для визначення критичних точок:

1. Аналіз ланцюга постачання:
   - Оцінка надійності постачальників сировини та матеріалів.
   - Визначення чутливості до можливих затримок у постачанні.
2. Виробничий процес:
   - Визначення ключових етапів виробничого процесу та їх взаємозв'язок.
   - Аналіз потенційних точок збою чи затримок у виробництві.
3. Технологічна інфраструктура:
   - Оцінка стійкості та ефективності технічного обладнання.
   - Аналіз можливих ризиків, пов'язаних із сучаснізацією та технічним обслуговуванням.
4. Людський фактор:
   - Вивчення залежності від ключових співробітників та експертів.
   - Аналіз можливих проблем у навчанні та резервуванні кадрів.
5. Фінансова стійкість:
   - Оцінка стійкості фінансової позиції компанії та її здатності витримувати фінансові труднощі.
   - Аналіз можливих впливів змін валютних курсів чи зростання витрат.
6. Регуляторні аспекти:
   - Вивчення впливу змін у законодавстві або стандартах на операційну діяльність.
   - Аналіз ризиків, пов'язаних із невідповідністю регуляторним вимогам.
7. Інформаційна безпека:

- Оцінка рівня захисту інформації та даних компанії.
- Аналіз можливостей кібератак та їхніх наслідків.

Цей комплексний підхід допоможе виявити та визначити ключові точки, де можуть виникнути проблеми чи ризики, та розробити стратегії для їхнього управління та мінімізації впливу на операційну діяльність компанії Декстра Мінерал.

Після завершення планування надходження товарів на підприємство ТОВ "Декстра Мінерал" проводиться відбір постачальників. Для цього розробляється перелік можливих постачальників для кожного виду товарів. У випадку, коли на ринку відсутня монополія серед постачальників, вибір постачальників здійснюється покупцями. Для формування списку постачальників ТОВ «Декстра Мінерал» використовує аналіз ринку товарів та звертається до відповідних джерел інформації. Основні принципи взаємодії з постачальниками на підприємстві включають: послідовність, принциповість, прозорість, рівність.

На сьогоднішній день компанія користується всім доступною CRM-системою під назвою TRELLO.

Trello – це безкоштовна багато-платформна система для управління про різними проектами. Вона використовує так звану парадигму для керування проектами – канбан. Система Trello (http://trello.com/) є програмною реалізацією канбан-дошки - одного з основних інструментів управлінської методології, відомої як "канбан".

Віртуалізація цієї програми наступна: проекти зображені у вигляді дошок, що містять колони (списки.) Списки місять картки, в основному це картки клієнті. В картках є можливість ставити задачі. Картки можуть переходити з одного списку в інший, за допомогою перетягування. Карткам можна присвоювати відповідальних користувачів. Створення команд з користувачів та дошок. Система має підтримку тегів, у вигляді кольорових міток, які налаштовує сам користувач. Картки можуть містити коментарі, час, дату, переліки задач.

Інтерфейс працює за допомогою формату drag-and-drap, усі оновляються у фоні. Є деякі недоліки такі, як: система не працює в офлайні, відсутня можливість

редагувати коментарі, немає безпечної аутентифікації кожен раз перед початок роботи.

Відповідно, окремі ресурси в Trello представляють собою гнучко налаштовуваний аналог канбан-дошки (для спрощення назвемо його трелло-дошкою), яка складається з колонок, в які можна розміщувати окремі картки, що містять, в свою чергу, додаткову інформацію. Таким чином реалізується трьохрівнева ієрархія інформації. З своєї суті Trello є системою управління проектами, що працює через мережевий доступ до центрального серверу.

У вебсервісі Trello реалізовано ряд додаткових можливостей, які розробники називають "power-ups". Також існує інтеграція з достатньою кількістю інших Інтернет-ресурсів, таких як, наприклад, Evernote або соціальні мережі. Як і більшість Інтернет-сервісів, Trello забезпечує кілька видів доступу до створюваних трелло-дошок: повний (всі можливості змін), частковий (окремі види активностей, наприклад, коментування або голосування), пасивний (тільки перегляд). Самі дошки в системі можуть бути двох типів: приватні (доступ тільки для членів команди проекту) і публічні (доступ для всіх). Пошукові системи можуть індексувати лише публічні дошки в Trello. Є деякі недоліки такі, як: система не працює в офлайні, відсутня можливість редагувати коментарі, немає безпечної аутентифікації кожен раз перед початок роботи.

Останнім часом все частіше використовується багатофакторна автентифікація. Для її реалізації використовують два або більше фактори автентифікації, що значно підвищує рівень захищеності системи. З метою підвищення ефективності сервісів багатофакторної автентифікації виникає питання дослідження існуючих та розробки нових систем. Ці системи повинні забезпечувати надійну автентифікацію, оскільки через високу складність структур сучасних мереж та сервісів завдання автентифікації користувача часто вирішується не в повному обсязі або має надто високу складність.

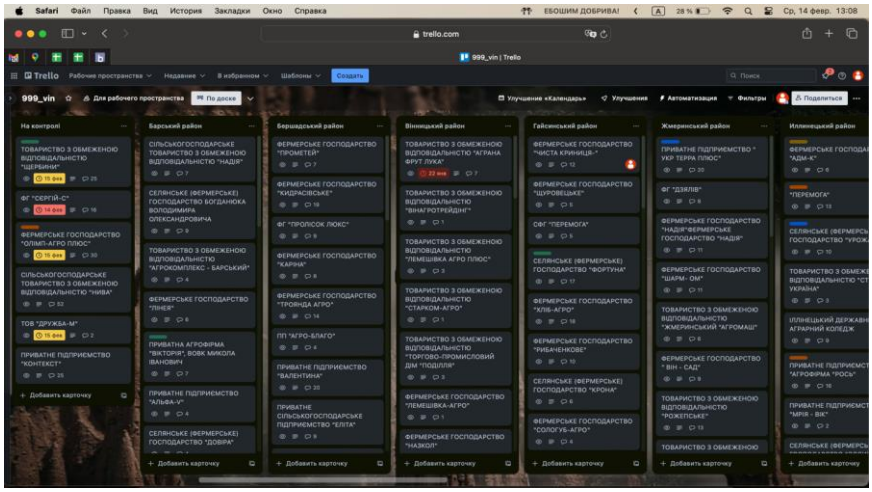

Рисунок. 1.4 – Робочий простір менеджера підприємства

В даній системі (рис. 1.4.) знаходяться всі дані про клієнтів всієї України, та кожен менеджер вносить інформацію про що він домовлявся та спілкувався з кожним із них.

Дана складова компанії є критичною точкою, адже стоїть питання кадрів, та доброчестності, типу коли людина звільняється вона не має забирати з собою базу і це треба контролювати. Тому в даній дипломній роботі ми будемо розробляти базу даних подібну по функціоналу до TRELLO на основі MySQL, де буде використовуватись механізм багатофакторної автентифікації. БД планується розроблятись у вебверсії, де кожен раз, як працівник починає свою роботу, буде зобовязаний ввести свій логін та пароль, а потім підтвердити свою спробу потрапити до БД за допомогою Google autentificator. Це дасть можливість контролювати базу і роботу співробітників.

Висновки до першого розділу

Таким чином, Товариство з обмеженою відповідальністю (ТОВ) «Декстра Мінерал» веде свою діяльність на українському ринку аграрних послуг. Предметом діяльності ТОВ «Декстра Мінерал» є неспеціалізована оптова торгівля хімічними продуктами, закупівля та постачання необхідної продукції; продажів товару кінцевому покупцеві; консультаційні послуги; транспортні послуги; надання в

оренду й експлуатацію власного чи орендованого майна.

Додатковими видами діяльності підприємства є: діяльність посередників у торгівлі сільськогосподарською сировиною, живими тваринами, текстильною сировиною та напівфабрикатами; технічне обслуговування та ремонт автотранспортних засобів; оптова торгівля іншими машинами й устаткованням; оптова торгівля твердим, рідким, газоподібним паливом і подібними продуктами.

Підприємство за роки роботи зайняло гідне місце серед підприємств України, що реалізують аналогічну продукцію. Основну ставку підприємство робить на високу якість реалізуємої продукції. Підприємство продовжує освоювати нові ринки і розширювати номенклатуру товарів. Крім ТОВ «Декстра Мінерал» на ринку функціонує ще ряд дистриб'юторських підприємств, що реалізують аналогічну продукцію. Серед них ТОВ «Антал Агро», ТОВ «Макош», ТОВ «Ерідон», «Тетра Агро», «Галнафтохім», «Агролайф» та інші.

На сьогоднішній день компанія користується всім доступною CRM-системою під назвою TRELLO. Дана складова компанії є критичною точкою, адже стоїть питання кадрів, та доброчестності, типу коли людина звільняється вона не має забирати з собою базу і це треба контролювати. Тому в даній дипломній роботі ми будемо розробляти базу даних подібну по функціоналу до TRELLO на основі MySQL, де буде використовуватись механізм багатофакторної автентифікації. БД планується розроблятись у вебверсії, де кожен раз, як працівник починає свою роботу, буде зобовязаний ввести свій логін та пароль, а потім підтвердити свою спробу потрапити до БД за допомогою Google autentificator. Це дасть можливість контролювати базу і роботу співробітників.

РОЗДІЛ 2
МЕХАНІЗМИ ТА ПРИНЦИПИ БАГАТОФАКТОРНОЇ АВТЕНТИФІКАЦІЇ

2.1 Методології багатофакторної автентифікації

Аутентифікація є кінцевою метою забезпечення безпеки системи, оскільки вона гарантує, що особа, яка намагається отримати доступ до системи, дійсно є тим, за кого себе видає. Відбуваючись після ідентифікації, яка зосереджена на визначенні особистих даних користувача, аутентифікація перевіряє ці дані для підтвердження їхньої вірності. Таким чином, після успішної аутентифікації користувача система переходить до третього етапу, а саме авторизації, де визначається, які конкретні дії чи ресурси користувач може використовувати.

Традиційні методи аутентифікації, такі як використання логінів та паролів, мають певні обмеження. Найзначнішим з них є відокремленість особи, що намагається авторизуватися, від методу, який використовується для підтвердження її особи. Це створює певні ризики щодо безпеки, оскільки хакери можуть використовувати скомпрометовані паролі або ідентифікаційні дані. Проте цей недолік може бути усунутий за допомогою біометричних параметрів, які асоціюють ідентичність користувача безпосередньо з його особистістю, забезпечуючи вищий рівень безпеки та надійності.

В сучасний період атаки на системи автентифікації стали звичайним явищем, і багато публікацій присвячені вивченню різних методів, видів і способів автентифікації. Для кращого розуміння кожного з цих видів необхідно вивчити, які методи та способи можна використовувати, а також які типи протоколів використовуються в цьому процесі.

Швидкі темпи розвитку обчислювальної техніки, автоматизованих інформаційних систем та нових технологій створюють благоприятні умови для зростання проблем безпеки інформації. Наприклад, такі явища, як промислове шпигунство та комп'ютерна злочинність, стають більш поширеними. Однією з основних загроз є несанкціонований доступ до конфіденційної інформації.

Це робить актуальним і важливим завдання захисту інформації для будь-якого сучасного підприємства, незалежно від його типу діяльності та форми власності. Без надійної системи захисту інформації, що включає як організаційні та нормативні заходи, так і програмно-апаратні засоби контролю безпеки інформації під час її обробки, зберігання та передачі в автоматизованих системах, підприємство не може ефективно розвиватися та здійснювати свою діяльність.

Сам процес автентифікації є процедурою перевірки відповідності пред'явленого ідентифікатора об'єкта комп'ютерної системи з метою підтвердження його належності до цього об'єкту.

Цей процес може бути здійснений за допомогою різних методів:

- Парольні: використовуються як одноразові, так і багаторазові паролі.
- Інфраструктура відкритих ключів (PKI): ґрунтується на асиметричній криптографії, де закритий ключ користувача може знаходитись на смарт-карті, криптографічному токені або знімному накопичувачі.
- Мобільна автентифікація: генерує одноразовий пароль (OTP) за допомогою спеціальної програми на смартфоні. Таким чином, смартфон виступає як OTP токен.
- Біометричні: використовуються фізіологічні характеристики користувача для перевірки.
- Інформація про користувача: така як номер телефону, дівоче прізвище матері, дата реєстрації та інше, що може використовуватися для відновлення логіна і пароля або для двофакторної автентифікації.
- Користувацькі дані: включають інформацію про точки доступу бездротового зв'язку та геодані про місце знаходження користувача.

Якщо узагальнити, на сьогоднішній день існують різноманітні методи аутентифікації, які використовуються для забезпечення безпеки доступу до систем та даних. Ось основні з них:

1. Парольні методи: Це найпоширеніший метод аутентифікації, де користувачі вказують паролі для доступу до систем. Паролі можуть бути одноразовими, генеруватися системою для кожної сесії, або багаторазовими, коли користувач сам

задає пароль. Прикладами є PIN-коди, слова, цифри або графічні ключі. Також до цього типу належать одноразові паролі, які можуть надсилатися користувачам через SMS або інші засоби.

2. Комбіновані методи: Ця форма аутентифікації використовує кілька методів одночасно для підтвердження ідентичності користувача. Наприклад, можуть використовуватися парольні методи разом із криптографічними сертифікатами для забезпечення більшого рівня безпеки.

3. Біометричні методи: Ці методи базуються на унікальних фізіологічних або поведінкових характеристиках користувача. Це може бути відбиток пальця, сітківка ока, тембр голосу, сканер обличчя або навіть ДНК. Біометричні дані використовуються для перевірки ідентичності користувача і забезпечення доступу до системи чи даних.

Ці методи аутентифікації можуть застосовуватися окремо або комбінуватися для створення більш надійних систем захисту. Кожен з них має свої переваги та обмеження, і вибір методу залежить від конкретних потреб та вимог безпеки організації.

Ці способи можна класифікувати наступним чином:

- Базова автентифікація: включає логін і пароль користувача, які передаються у вебзапиті. Однак це не надійний метод, оскільки зловмисник може легко перехопити пакети і використовувати отримані дані.
- Дайджест-автентифікація: використовує передачу хешованих паролів. Постійне оновлення хешу ускладнює розшифрування даних для зловмисника.
- HTTPS: забезпечує шифрування не лише логіна і пароля користувача, але і всіх інших даних, що передаються між клієнтом і сервером в Інтернеті.
- Автентифікація з пред'явленням цифрового сертифікату: використовує протоколи з запитом і відповіддю на них.
- Автентифікація за допомогою Cookies: браузер надсилає Cookies як одну з частин HTTP-запиту при кожній спробі підключення до ресурсу.
- Децентралізована автентифікація: використовується в протоколах, таких як OpenID, OpenAuth та OAuth, і базується на принципі роботи з

розподіленою системою ідентифікації.

Автентифікація може бути розділена на дві основні категорії в залежності від кількості методів, які використовуються: однофакторна та багатофакторна. Однофакторна автентифікація передбачає використання лише одного методу для підтвердження ідентичності користувача, тоді як багатофакторна використовує комбінацію різних методів.

Існують різні типи автентифікації залежно від можливостей засобів та рівня інформаційної безпеки:

1. Статична автентифікація: Цей тип захищає від несанкціонованого доступу шляхом використання постійних ідентифікаторів, таких як паролі. Вона має на меті унеможливити зловмисникам отримання доступу до інформації, використовуючи ідентифікатор користувача, який може бути вкрадений під час користування ресурсом чи сайтом.

2. Динамічна автентифікація: Цей тип базується на використанні змінних ідентифікаторів, які генеруються перед кожним сеансом. Вона призначена для запобігання активним атакам, але не забезпечує повного захисту.

3. Постійна автентифікація: Цей вид автентифікації захищає суб'єкт від несанкціонованої крадіжки або модифікації його ідентифікатора на будь-якому етапі роботи з інформацією. Вона забезпечує захист від атак навіть після того, як користувач вже автентифікувався.

В залежності від політики безпеки системи та рівня довіри, можна виділити наступні типи автентифікації:

1. Одностороння автентифікація: В цьому випадку користувач підтверджує своє право на доступ до ресурсу, але сам ресурс не перевіряє автентичність користувача. Це може включати, наприклад, введення пароля або іншого ідентифікатора для отримання доступу до системи.

2. Взаємна автентифікація: У цьому випадку перевіряється як автентичність користувача, так і власника ресурсу. Це забезпечує більш високий рівень безпеки, оскільки обидва боки перевіряються на автентичність за допомогою криптографічних методів, наприклад, обмінюючи цифрові підписи або сертифікати.

Також, важливо розуміти, що процес автентифікації полягає в порівнянні інформації, яку надає користувач, з тією, яку знає система. В залежності від типу інформації, її можна класифікувати за одним із наступних факторів:

1. Фактор знання: Це щось, що користувач знає. Сюди входять паролі або відповіді на секретні питання, які використовуються для підтвердження ідентичності користувача.

2. Речовий фактор: Це щось, чим користувач володіє. Це можуть бути смарт-картки, токени та інші фізичні пристрої або об'єкти, які використовуються для автентифікації.

3. Біофактор: Це щось, що є частиною користувача. Сюди входять дані, які можна отримати з біометричних сканерів, такі як відбитки пальців, геометрія руки, почерк або голос користувача, і які використовуються для перевірки ідентичності.

Зазначені зауваження відображають різноманітні можливості впровадження автентифікаційних методів залежно від потреб та вимог безпеки системи. Ось деякі ключові моменти:

1. Різноманітність методів реалізації: В залежності від конкретних вимог і контексту застосування, автентифікаційні методи можуть бути реалізовані різними способами, включаючи використання різних комбінацій методів.

2. Постійна автентифікація з високим рівнем безпеки: Багатофакторна автентифікація, що використовує декілька методів, є одним з найефективніших способів забезпечення високого рівня інформаційної безпеки.

3. Оптимальність двофакторної автентифікації: Двофакторна автентифікація, яка використовує комбінацію статичних та стійких методів, таких як багаторазові та одноразові паролі або багаторазовий пароль та біометричні дані, може бути оптимальним варіантом. Це забезпечує вищий рівень безпеки порівняно з одним методом автентифікації.

Узагальнюючи, важливо враховувати конкретні потреби та вимоги щодо безпеки і доступності при виборі і реалізації методів автентифікації.

2.2 Технологій використання одноразових паролей

2.2.1. Аналіз функціональності та застосування Google Authenticator

Забезпечення безпеки паролів та усвідомлення ризиків грають важливу роль у збереженні конфіденційності та безпеки користувачів в Інтернеті. Існує кілька основних практик, які можна рекомендувати для уникнення ризиків:

1. Унікальні паролі для різних сайтів: Використовуйте унікальні паролі для кожного вебсайту чи сервісу. Це запобігає розповсюдженню шкодливих дій в разі витоку пароля з одного ресурсу.

2. Завантаження з надійних ресурсів: Уникають завантаження програм і файлів з ненадійних джерел. Використовуйте лише офіційні магазини додатків та перевірені вебсайти для завантаження.

3. Обережність при переході за посиланнями: Не переходьте по сумнівним посиланням у електронних листах. Перевіряйте достовірність посилань перед їх відкриттям, особливо якщо лист надійшов від невідомого або неперевіреного вами відправника.

4. Дводіапазонна аутентифікація: Використовуйте двофакторну аутентифікацію (2FA), яка додає додатковий шар безпеки, наприклад, через введення коду, який надсилається на мобільний пристрій.

5. Регулярна зміна паролів: Змінюйте паролі регулярно для уникнення можливих атак, особливо якщо ви спільно використовуєте їх на різних сайтах.

Усвідомлення цих практик і дотримання їх може значно зменшити ризики кіберзагроз та забезпечити більшу безпеку в Інтернеті.

Так, правильне налаштування безпеки може значно зменшити ризики витоку та неправомірного доступу до особистих даних. Важливо дотримуватися певних обережних заходів для уникнення кіберзагроз. Декілька рекомендацій:

1. Перевірка адреси вебсторінки: Уважно перевіряйте адресу вебсторінки перед введенням особистих даних. Фішингові сторінки часто мають помилки або

подвійні літери в адресі.

2. Застосунок "Захисник пароля Google": Використовуйте застосунок "Захисник пароля Google", який попереджає користувача про фішингові сторінки та не дозволяє переходити за неправильними посиланнями.

3. Дотримання уважності при введенні даних: Будьте уважними при введенні своїх особистих даних на вебсайтах. Відмовляйтеся від введення інформації на підозрілих або непідтверджених ресурсах.

4. Регулярна зміна паролів: Змінюйте паролі регулярно, щоб ускладнити завдання зловмисникам у випадку витоку.

5. Оновлення програм та систем: Забезпечуйте регулярне оновлення всіх програм та операційних систем, оскільки це може містити важливі патчі для виправлення потенційних вразливостей.

Узагальнюючи, свідоме та обережне використання Інтернету разом із застосуванням додаткових заходів безпеки допоможе мінімізувати ризики кіберзагроз і зберегти особисті дані в безпеці.

Також, якщо шифрування HTTPS не активоване, це може відіграти роль відлякувального фактору, що має відтермінувати користувача від надання своїх особистих даних на даному вебресурсі. Проте немає абсолютної гарантії, що будь-який застосунок забезпечить захист користувача від різних видів шахрайства. І хоча існують різні платні інструменти для захисту від фішингу, вони не мають такої самої популярності, як, наприклад, "Google Authenticator" або "Google Password Alert".

"Google Authenticator" - це мобільний додаток, розроблений компанією Google, призначений для забезпечення додаткового шару безпеки у процесі багатофакторної аутентифікації. Додаток створений для генерації одноразових паролів (OTP) або кодів, які користувачі використовують разом із своїм основним паролем при вході в обліковий запис або при здійсненні інших важливих дій.

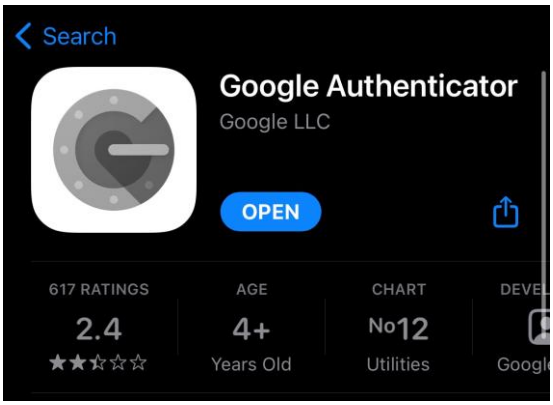

Рисунок. 2.1 - "Google Authenticator"

Основна ідея "Google Authenticator" полягає в тому, щоб користувачі мали унікальний одноразовий код, який змінюється через певний інтервал часу. Ці коди генеруються додатком і часто використовуються для багатофакторної аутентифікації, де, крім основного пароля, потрібно ввести ще й код з "Google Authenticator". Такий підхід робить доступ до облікового запису більш безпечним і ускладнює завдання зловмисникам, навіть якщо вони дізналися основний пароль.

Додаток підтримує стандартні алгоритми генерації одноразових кодів, такі як TOTP (Time-Based One-Time Password) та HOTP (HMAC-Based One-Time Password).

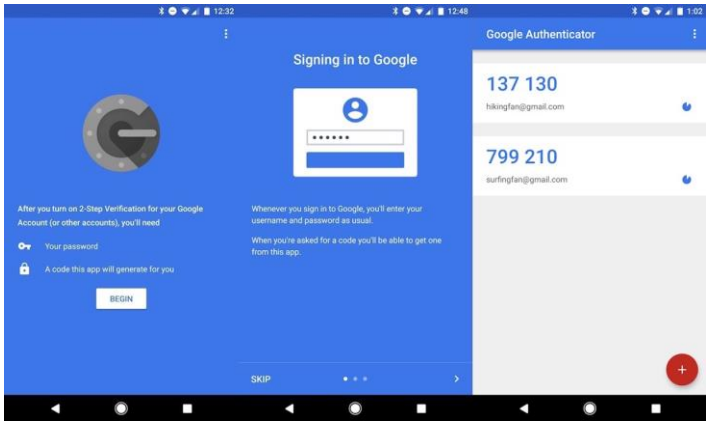

Рисунок. 2.2 - "Google Authenticator". Автентифікація через Google

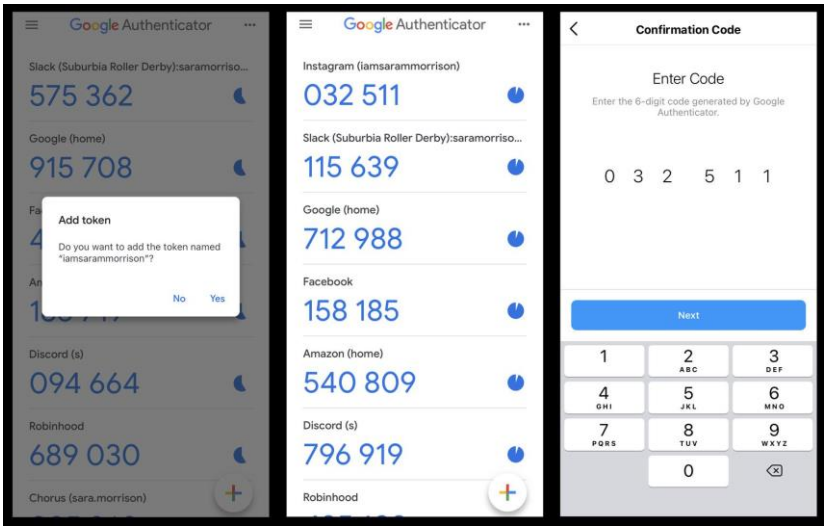

Рисунок. 2.3 - "Google Authenticator". Генерування кодів

Алгоритми генерації одноразових кодів, такі як TOTP (Time-Based One-Time Password) та HOTP (HMAC-Based One-Time Password), є стандартами для створення і використання одноразових паролів у багатофакторній аутентифікації.

1. TOTP (Time-Based One-Time Password):

- Принцип: Генерація коду ґрунтується на поточному часі та секретному ключі. Код змінюється через фіксований період часу (зазвичай 30 секунд).
- Процес генерації:

1. Користувач та сервер обирають та зберігають спільний секретний ключ.
2. З поточного часу обчислюється номер періоду.
3. Номер періоду та секретний ключ передаються через хеш-функцію, і генерується одноразовий код.

2. HOTP (HMAC-Based One-Time Password):

- Принцип: Генерація коду ґрунтується на лічильнику та секретному ключі. Код змінюється при кожній валідній події (наприклад, при кожному вході).
- Процес генерації:

1. Користувач та сервер обирають та зберігають спільний секретний

ключ і лічильник.

2. Лічильник збільшується при кожному використанні.

3. Лічильник та секретний ключ передаються через хеш-функцію, і генерується одноразовий код.

Обидва ці алгоритми використовують хеш-функції та секретні ключі для генерації кодів, і вони дозволяють забезпечити безпеку при багатофакторній аутентифікації.

Отже, для ефективного захисту своїх особистих даних від фішингових атак користувачеві слід встановити двофакторну аутентифікацію та самостійно розпізнавати фішингові сторінки, електронні листи та повідомлення. Цього буде достатньо для базового рівня безпеки і захисту від фішингу.

І хоча користування двофакторною аутентифікацією може включати деякі нюанси, вони здаються складними лише на перший погляд. Кожен вирішує для себе, яке є ідеальне співвідношення між безпекою і зручністю. У будь-якому випадку всі ці заходи повністю виправдовуються, коли мова йде про безпеку платіжних даних або особистої інформації, яку не слід відкривати чужим очам.

2.2.2. Дослідження можливостей та особливостей Authy як системи автентифікації

Authy - це платформа для двофакторної аутентифікації (2FA), яка надає інструменти для безпечного входу в облікові записи за допомогою мобільних пристроїв. Ось дослідження можливостей та особливостей Authy:

1. Багатоплатформенність: Authy доступний на різних платформах, включаючи мобільні пристрої (iOS, Android), веббраузери і різні настільні операційні системи. Це дозволяє користувачам отримати доступ до своїх кодів для двофакторної аутентифікації з будь-якого пристрою.

2. Синхронізація пристроїв: Authy автоматично синхронізує дані між різними пристроями користувача, що дозволяє отримати доступ до кодів 2FA на будь-якому пристрої, підключеному до облікового запису Authy.

3. Зручний інтерфейс користувача: Додаток Authy має зручний інтерфейс користувача, який дозволяє легко додавати нові облікові записи для двофакторної аутентифікації та керувати ними.

4. Безпека: Authy використовує сильне шифрування для захисту конфіденційної інформації користувачів, такої як секретні ключі для генерації одноразових кодів.

5. Резервне копіювання і відновлення: Authy надає можливість створювати резервні копії облікових записів і відновлювати їх у випадку втрати або заміни пристрою.

6. Підтримка більшості вебсайтів і сервісів: Authy підтримує широкий спектр вебсайтів і онлайн-сервісів, які підтримують двофакторну аутентифікацію через TOTP або аналогічні методи.

7. Автоматична адаптація до зміни часу: Authy автоматично адаптується до зміни часового поясу або переходу на літній/зимовий час для правильної генерації одноразових кодів.

8. Екстрені коди: Authy надає можливість налаштувати екстрені коди, які можна використовувати для входу в обліковий запис у випадку втрати доступу до мобільного пристрою або іншого проблемного сценарію.

Узагальнюючи, Authy - це потужний і зручний інструмент для забезпечення безпеки облікових записів через двофакторну аутентифікацію, який пропонує широкий функціонал та високий рівень безпеки.

2.2.3 Оцінка використання OATH Toolkit у контексті безпеки

OATH-Toolkit - це безкоштовний набір програмних засобів для аутентифікації з одноразовим паролем (OTP) з використанням алгоритмів HOTP/TOTP. Програмне забезпечення постачається з невеликим набором утиліт командного рядка, які охоплюють більшість операцій, пов'язаних з OTP.

Ця процедура конфігурації пояснює аутентифікацію OATH-Toolkit PAM з використанням демона OpenSSH. Додаткові приклади конфігурації знаходяться в

офіційній документації.

Набір інструментів OATH забезпечує компоненти для побудови систем аутентифікації одноразовими паролями. Він містить спільні бібліотеки C, інструменти командного рядка та модуль PAM. Підтримувані технології включають алгоритм HOTP на основі подій (RFC 4226), алгоритм TOTP на основі часу (RFC 6238) та Портативний Контейнер Симетричного Ключа (PSKC, RFC 6030) для управління даними секретного ключа. OATH означає Open AuTHentication, яка є організацією, яка визначає алгоритми.

До складу входять такі компоненти:

- liboath: Спільна та статична бібліотека C для обробки OATH.
- oathtool: Інструмент командного рядка для генерації та перевірки OTP.
- pam_oath: Модуль PAM для підключення аутентифікації входу для OATH.
- libpskc: Спільна та статична бібліотека C для обробки PSKC.
- pskctool: Інструмент командного рядка для обробки даних PSKC.

Для отримання додаткової інформації дивіться сторінку Документації.

Використання OATH Toolkit у контексті безпеки може бути дуже корисним, оскільки він надає можливість реалізувати системи аутентифікації з одноразовими паролями, такі як HOTP і TOTP. Ось деякі переваги та важливі аспекти використання OATH Toolkit з точки зору безпеки:

1. Зменшення ризику атак перехоплення паролів: Одноразові паролі зменшують ризик перехоплення і використання паролів зловмисниками, оскільки кожен пароль використовується тільки один раз. Кожен одноразовий пароль може бути використаний лише один раз, після чого він стає недійсним. Це ускладнює завдання зловмисників, оскільки навіть якщо вони зможуть перехопити пароль, вони не зможуть його використати пізніше. Крім того, такі паролі часто базуються на фізичних пристроях або мобільних додатках, що генерують паролі, що потребує фізичного доступу до цих пристроїв. Це додатково ускладнює завдання зловмисників. Атаки фішингу також ускладнюються, оскільки одноразові паролі діють лише на дуже короткий проміжок часу і не можуть бути використані пізніше. Використання одноразових паролів може додати додатковий шар безпеки до

існуючих методів аутентифікації, підвищуючи загальний рівень безпеки аутентифікаційних систем.

2. Сильна криптографія: OATH Toolkit використовує стандартні алгоритми криптографії, такі як HMAC для HOTP і SHA для TOTP, що забезпечує високий рівень безпеки. OATH Toolkit використовує стандартні алгоритми криптографії, такі як HMAC для HOTP і SHA для TOTP. Це забезпечує високий рівень безпеки, оскільки ці алгоритми є визнаними та перевіреними на практиці в галузі криптографії. HMAC (Hash-based Message Authentication Code) - це метод аутентифікації, який використовує хеш-функцію разом з секретним ключем для створення аутентифікаційного коду для повідомлення. SHA (Secure Hash Algorithm) - це криптографічна хеш-функція, яка використовується для створення унікального, неперевіреного значення з вхідного повідомлення. Використання таких стандартних алгоритмів забезпечує надійність і стійкість системи аутентифікації, оскільки вони вже пройшли перевірку на стійкість та безпеку в широкому спектрі застосувань.

3. Захист від атак фішингу: Оскільки кожен одноразовий пароль дійсний лише на короткий період часу, атаки фішингу, спрямовані на отримання паролів, стають набагато складнішими. Одноразові паролі відображаються лише на обмежений час, зазвичай лише декілька хвилин або секунд. Це робить атаки фішингу, спрямовані на отримання паролів, набагато складнішими. Зловмисники мають обмежений час для використання перехопленого пароля, оскільки він стає недійсним після закінчення визначеного терміну. Тому вони мають виконати атаку швидко після перехоплення пароля, що значно ускладнює завдання. Крім того, навіть якщо зловмисники використають перехоплений одноразовий пароль, вони не зможуть використати його пізніше, оскільки він буде недійсним. Таким чином, використання одноразових паролів додає додатковий шар безпеки та ускладнює атаки фішингу, забезпечуючи більш високий рівень безпеки для системи аутентифікації.

4. Підтримка стандартів безпеки: OATH Toolkit використовує стандарти безпеки, такі як RFC4226 та RFC6238 для HOTP і TOTP відповідно. Це гарантує сумісність із застосуваннями, що підтримують ці стандарти. RFC (Request for

Comments) - це документи, що описують протоколи, алгоритми та стандарти для Інтернету та мережевих технологій. RFC4226 визначає алгоритм HOTP (HMAC-based One-Time Password), який використовує HMAC для генерації одноразових паролів на основі подій. RFC6238 визначає алгоритм TOTP (Time-Based One-Time Password), який використовує хеш-функцію та часовий маркер для генерації одноразових паролів на основі часу. Використання цих стандартів дозволяє забезпечити сумісність із застосунками, що підтримують ці протоколи, і забезпечити високий рівень безпеки для систем аутентифікації, що використовують OATH Toolkit.

5. Можливість зберігання секретних ключів у безпечному форматі: За допомогою OATH Toolkit можна керувати секретними ключами у форматі PSKC, що дозволяє зберігати їх у безпечному, зашифрованому вигляді. Так, за допомогою OATH Toolkit можна керувати секретними ключами у форматі PSKC (Portable Symmetric Key Container). Цей формат дозволяє зберігати секретні ключі у безпечному, зашифрованому вигляді. PSKC є стандартизованим форматом, описаним у RFC6030, який дозволяє зберігати секретні ключі та інші конфіденційні дані у захищеному контейнері. Використання цього формату дозволяє підвищити безпеку зберігання секретних ключів, оскільки вони зберігаються у зашифрованому вигляді і можуть бути доступні лише авторизованим користувачам або системам. Такий підхід допомагає уникнути витоку секретних ключів і забезпечує додатковий шар безпеки для систем аутентифікації, що використовують OATH Toolkit.

У цілому, використання OATH Toolkit може підвищити рівень безпеки аутентифікації та зменшити ризики, пов'язані з витоком паролів та атаками на аутентифікаційні системи. Однак важливо правильно налаштувати та використовувати цей інструмент для максимального забезпечення безпеки.

2.2.4 OpenOTP і його переваги в сфері автентифікації

OpenOTP - це інноваційна система автентифікації, яка забезпечує безпеку та зручність для користувачів. Ось деякі з його переваг у сфері автентифікації:

1. Багатофакторна аутентифікація: OpenOTP підтримує різні методи аутентифікації, включаючи двофакторну (2FA) та багатофакторну (MFA) аутентифікацію. Це дозволяє використовувати комбінації факторів, такі як пароль, токен або біометричні дані, для підтвердження ідентичності користувача.

2. Безпека: OpenOTP використовує сучасні алгоритми шифрування та протоколи безпеки для захисту конфіденційності та цілісності користувальницьких даних під час процесу автентифікації.

3. Мобільний доступ: OpenOTP підтримує мобільні додатки для генерації одноразових паролів або здійснення аутентифікації за допомогою мобільних пристроїв. Це дозволяє користувачам отримувати доступ до своїх облікових записів навіть під час подорожей або роботи вдалині.

4. Легкість використання: OpenOTP має інтуїтивний і зручний інтерфейс, що робить процес автентифікації простим і зрозумілим для користувачів будь-якого рівня навичок.

5. Складність паролів: OpenOTP може вимагати складних паролів або використовувати інші методи аутентифікації, що зростають у складності, що забезпечує додатковий захист від несанкціонованого доступу.

6. Широкі можливості інтеграції: OpenOTP має різноманітні інтеграційні можливості з іншими системами, такими як VPN, електронна пошта, вебсайти тощо, що дозволяє легко впроваджувати інструменти безпеки в комплексні корпоративні середовища.

Узагальнюючи, OpenOTP надає високий рівень безпеки, зручність та гнучкість в реалізації різних методів аутентифікації, що робить його привабливим вибором для організацій, що прагнуть підвищити безпеку своїх систем.

2.3 Автентифікація через коди електронної пошти та SMS: переваги та недоліки

Найбільш поширеною, хоч і найменш безпечною, формою двофакторної аутентифікації є отримання кодів через SMS. Автентифікація через коди

електронної пошти та SMS - це один із способів двофакторної аутентифікації (2FA), коли користувач отримує унікальний код на свою електронну адресу або мобільний телефон для підтвердження своєї ідентичності. Однак цей метод є ненадійним через можливість перехоплення SMS та можливість підробки або зламування номерів телефонів через мобільних операторів. Хоча використання SMS-кодів краще, ніж використання лише паролів, рекомендується використовувати програмні засоби автентифікації або фізичні ключі безпеки, які є більш надійними. Відданий зловмисник може легко отримати доступ до SMS-кодів 2FA, звернувшись до мобільного оператора та замінивши вашу SIM-карту.

Якщо ви готові ввести двофакторну аутентифікацію для облікових записів вашої організації, ви можете скористатися ресурсом https://2fa.directory/, щоб швидко знайти інформацію та інструкції для підключення конкретних служб, таких як Gmail, Office 365, Facebook, Twitter тощо, і дізнатися, які типи двофакторної аутентифікації підтримуються різними службами.

Ось переваги та недоліки цього методу:

Переваги:

1. Простота використання: Багато людей вже мають електронну адресу або мобільний телефон, тому автентифікація через коди SMS або електронну пошту є досить простою для багатьох користувачів.

2. Широка доступність: Коди електронної пошти та SMS можуть бути використані на більшості мобільних телефонів та електронних пристроях, що робить їх досить універсальними.

3. Менша вартість: В порівнянні з іншими методами 2FA, які можуть потребувати спеціального обладнання або додаткового програмного забезпечення, використання кодів електронної пошти та SMS може бути більш економічно вигідним варіантом.

Недоліки:

1. Залежність від мережі: Використання SMS потребує наявності мобільного зв'язку, а для електронної пошти потрібне з'єднання з Інтернетом. Тому цей метод може бути неефективним у випадках обмеженої мережевої доступності або у

країнах з поганим покриттям мережі.

2. Затримки: У деяких випадках коди SMS можуть затримуватися або навіть не надходити вчасно через проблеми з мережею чи налаштуванням.

3. Потенційна вразливість до атак: SMS-повідомлення можуть бути перехоплені або підроблені, що створює потенційну вразливість для атак.

4. Більша ймовірність втрати доступу: У порівнянні з апаратними токенами або мобільними додатками для 2FA, коди електронної пошти та SMS можуть бути легше втрачені або викрадені, особливо якщо користувач має необхідність використовувати чужий телефон або комп'ютер для отримання коду.

Цей метод, що передбачає відправлення одноразового пароля у вигляді текстового повідомлення або введення комбінації цифр голосового меню під час телефонного дзвінка, вважається менш надійним у порівнянні з іншими методами двофакторної аутентифікації.

Основна перевага цього методу полягає у тому, що для його використання не потрібне підключення до Інтернету. Однак, необхідність наявності мобільної мережі може стати недоліком, адже в окремих ситуаціях може виникнути проблема з отриманням текстового повідомлення або дзвінком.

Крім того, існує ризик втрати SIM-картки або телефону, що може призвести до втрати доступу до одноразового пароля та, відповідно, до облікового запису.

У підсумку, хоча цей метод може бути зручним у деяких ситуаціях, він вважається менш надійним і потребує уважного врахування потенційних ризиків та недоліків.

2.4 Застосування геолокації в багатофакторній автентифікації

2.4.1 Технології геолокаційного збору даних: GPS, Wi-Fi, Сотові Мережі

Застосування геолокації у багатофакторній автентифікації включає використання інформації про місцезнаходження користувача для підтвердження його ідентичності. Ось деякі способи, які можуть використовуватися:

1. Геозони: Система може налаштовувати "геозони" - географічно обмежені області, де користувач зазвичай доступується до свого облікового запису. Якщо доступ здійснюється з незвичного місця, система може вимагати додаткової аутентифікації.

2. Аналіз відстані: Система може оцінювати відстань між двома місцями, де користувач зазвичай входить у свій обліковий запис. Велика відстань може викликати підозру і вимагати додаткової перевірки.

3. IP-адреса: Геолокація може бути визначена за допомогою IP-адреси, що використовується для з'єднання. Хоча цей метод може бути менш точним, він все ще може дати загальне уявлення про місцезнаходження користувача.

4. Мобільність пристрою: Інформація про рух або зміну місцезнаходження пристрою також може бути використана для аналізу. Наприклад, якщо користувач раптово пересувається з одного місця до іншого, система може вимагати додаткової аутентифікації для підтвердження.

Переваги використання геолокації у багатофакторній автентифікації включають підвищення безпеки шляхом додаткового шару перевірки місцезнаходження користувача та спрощення користувача, оскільки він може здійснювати доступ до свого облікового запису без додаткових кроків, якщо доступ здійснюється з звичного місця. Однак важливо також враховувати приватність користувачів і захист їх особистої інформації при зборі та використанні геоданих.

Технології геолокаційного збору даних включають в себе різні методи визначення місцезнаходження користувачів. Ось кілька основних технологій геолокаційного збору даних:

1. GPS (Global Positioning System): GPS є однією з найпоширеніших технологій геолокації, яка використовує супутникові сигнали для визначення місцезнаходження пристрою. Ця технологія дозволяє досягти високої точності визначення місцезнаходження, особливо на відкритих просторах. Багато людей використовують системи позиціонування на своїх навігаційних пристроях, смартфонах і планшетах. Однак, слабкою стороною цих рішень є те, що система глобального позиціонування (GPS) недоступна усередині приміщень через сильне

приглушення сигналів стінами та перекриттями будівель. Це відкриває нішу для розробки надійних рішень з позиціювання у приміщеннях. GPS-приймач працює шляхом обчислення власного місцезнаходження через вимірювання часу, який потрібно сигналам від GPS-супутників, щоб дійти до приймача. Кожен супутник надсилає повідомлення, яке містить інформацію про час, точку орбіти, з якої було надіслано повідомлення, а також загальний стан системи та приблизні дані орбіт інших супутників. GPS-приймач визначає час затримки у надходженні сигналу і обчислює відстань до супутників. Застосовуючи метод трилатерації, він визначає своє місце на основі цих відстаней. Отримані координати перетворюються в наочну форму і відображаються користувачеві на екрані.

2. Wi-Fi-геолокація: Wi-Fi-геолокація використовує визначення місцезнаходження за допомогою точок доступу Wi-Fi, які знаходяться в навколишній області. Пристрої зчитують ідентифікатори цих точок доступу і використовують їх для визначення місцезнаходження. Цей метод може бути особливо ефективним у місцях з великою концентрацією точок доступу, таких як міста та торгові центри.

3. Сотові мережі: Технологія геолокації на основі сотових мереж використовується для визначення місцезнаходження за допомогою інформації від сотових базових станцій. Принцип полягає в тому, що мобільні пристрої зв'язуються з близькими сотовими вежами, і їх місцезнаходження може бути приблизно визначено на основі сигналів, які вони отримують від цих станцій.

Кожен з цих методів має свої переваги і обмеження, і вони можуть використовуватися окремо або комбіновано для досягнення більшої точності визначення місцезнаходження.

2.4.2 Стратегії протидії фальсифікації геолокації

Протидія фальсифікації геолокації є важливим завданням для забезпечення безпеки та точності визначення місцезнаходження в різних системах. Ось декілька стратегій протидії фальсифікації геолокації:

1. Валідація даних: Перевірка отриманих даних геолокації на відповідність стандартам та реальним обмеженням може допомогти виявити спроби фальсифікації. Наприклад, перевірка на основі швидкості переміщення може допомогти виявити надмірно швидкі переміщення, які можуть бути підозрілими. Стратегії протидії фальсифікації геолокації можуть включати різноманітні методи перевірки та аналізу отриманих даних. Ось деякі з них: перевірка відповідності стандартам: перевірка даних геолокації на відповідність встановленим стандартам та параметрам може допомогти виявити аномалії та підозрілі відхилення. Наприклад, якщо координати знаходяться поза діапазоном прийнятних значень або не відповідають дійсному місцезнаходженню, це може свідчити про фальсифікацію; аналіз швидкості переміщення: перевірка швидкості переміщення може допомогти виявити надмірно швидкі переміщення, які не відповідають типовому режиму руху користувача. Наприклад, раптове переміщення великої відстані за короткий час може бути підозрілим та вказувати на фальсифікацію; перевірка наявності супутникових сигналів: перевірка доступності супутникових сигналів та їх якості може бути корисною стратегією. Якщо сигнали занадто слабкі або відсутні, це може свідчити про спробу фальсифікації, особливо якщо користувач знаходиться у місці з гарною видимістю неба; застосування алгоритмів машинного навчання: Використання алгоритмів машинного навчання для аналізу та виявлення аномалій у данних геолокації може допомогти автоматично виявляти спроби фальсифікації на основі складних патернів та залежностей у даних.

Ці стратегії можуть бути використані окремо або комбіновано для ефективного виявлення та протидії спробам фальсифікації геолокації.

2. Мультиплексування джерел: Використання кількох джерел для визначення геолокації, таких як GPS, Wi-Fi та сотові мережі, може допомогти ускладнити фальсифікацію, оскільки необхідно фальсифікувати дані з усіх цих джерел одночасно. Враховуючи, що кожне джерело може мати власні особливості та переваги, а також що дані з них можуть бути незалежно перевірені, використання комплексного підходу може ускладнити спроби підробки. Наприклад, якщо дані з одного джерела виявляються суперечливими або несумісними з іншими, це може

викликати підозри та підвищити рівень безпеки системи. Такий підхід може стати ефективним заходом протидії фальсифікації в контексті геолокації.

3. Криптографічні заходи безпеки: Використання криптографічних методів для захисту даних геолокації може забезпечити цілісність та конфіденційність даних та ускладнити їх фальсифікацію. Використання криптографічних методів для захисту даних геолокації може забезпечити їх цілісність та конфіденційність, а також ускладнити спроби їх фальсифікації. Це досягається за допомогою різних методів, таких як підписування даних цифровим підписом, шифрування та аутентифікація. Цифровий підпис: Дані геолокації можуть бути підписані цифровим підписом, що дозволяє перевірити їхню цілісність та автентичність. Це означає, що будь-яка зміна даних буде виявлена під час перевірки підпису, що ускладнює їх фальсифікацію. Шифрування: Дані геолокації можуть бути зашифровані перед передачею, що забезпечує їхню конфіденційність та ускладнює доступ до них для несанкціонованих користувачів. Аутентифікація: Використання аутентифікаційних методів, таких як обмін ключами або використання цифрових сертифікатів, може підтвердити ідентичність відправника та забезпечити впевненість у тому, що дані геолокації отримані від вірного джерела. Всі ці заходи сприяють підвищенню рівня безпеки та ускладнюють фальсифікацію даних геолокації, що робить їх більш надійними в застосуванні.

4. Аналіз аномалій: Проведення аналізу аномалій може допомогти виявити незвичайні патерни або поведінку, які можуть бути пов'язані з фальсифікацією геолокації. Наприклад, раптові зміни місцезнаходження без належних пояснень можуть бути підозрілими. Подібні аномалії можуть бути виявлені шляхом порівняння даних геолокації зі звичайними патернами руху користувача, а також за допомогою алгоритмів аналізу даних, спрямованих на виявлення відхилень від типової поведінки. Такий аналіз допомагає виявити потенційні загрози безпеці та ускладнює спроби фальсифікації геолокації.

5. Перевірка довірених джерел: Використання довірених джерел або серверів для перевірки та підтвердження даних геолокації може допомогти уникнути фальсифікації. Наприклад, використання блокчейн технологій для реєстрації та

підтвердження даних про місцезнаходження може забезпечити їхню цілісність та недоступність для змін. Використання довірених джерел або серверів для перевірки та підтвердження даних геолокації може допомогти уникнути фальсифікації. Наприклад, використання блокчейн технологій для реєстрації та підтвердження даних про місцезнаходження може забезпечити їхню цілісність та недоступність для змін. Коли дані про геолокацію фіксуються у блокчейні, вони стають неможливими до модифікації або вилучення без відповідного доступу до всієї мережі. Це дозволяє створити надійний механізм перевірки та підтвердження даних геолокації, що ускладнює спроби їх фальсифікації.

Ці стратегії можуть бути використані окремо або у поєднанні для забезпечення ефективної протидії фальсифікації геолокації в різних системах.

Висновки до другого розділу

Тому, аутентифікація є кінцевою метою забезпечення безпеки системи, оскільки вона гарантує, що особа, яка намагається отримати доступ до системи, дійсно є тим, за кого себе видає. Відбуваючись після ідентифікації, яка зосереджена на визначенні особистих даних користувача, аутентифікація перевіряє ці дані для підтвердження їхньої вірності. Таким чином, після успішної аутентифікації користувача система переходить до третього етапу, а саме авторизації, де визначається, які конкретні дії чи ресурси користувач може використовувати.

Якщо узагальнити, на сьогоднішній день існують різноманітні методи аутентифікації, які використовуються для забезпечення безпеки доступу до систем та даних.

Автентифікація може бути розділена на дві основні категорії в залежності від кількості методів, які використовуються: однофакторна та багатофакторна. Однофакторна автентифікація передбачає використання лише одного методу для підтвердження ідентичності користувача, тоді як багатофакторна використовує комбінацію різних методів.

"Google Authenticator" - це мобільний додаток, розроблений компанією

Google, призначений для забезпечення додаткового шару безпеки у процесі багатофакторної аутентифікації. Додаток створений для генерації одноразових паролів (OTP) або кодів, які користувачі використовують разом із своїм основним паролем при вході в обліковий запис або при здійсненні інших важливих дій.

РОЗДІЛ 3
АНАЛІЗ СИСТЕМ УПРАВЛІННЯ БАЗАМИ ДАНИХ

3.1 Типи СУБД та їх особливості

Роль систем управління базами даних (СУБД) полягає у забезпеченні організованого доступу до інформації, уникненні повторення даних та гарантуванні цілісності та безпеки даних.

Перші СУБД з'явилися ще в 1960-х роках і спрямовані були на зберігання даних у вигляді ієрархічних або мережевих структур. У наступному десятилітті, у 1970-х роках, з'явилися реляційні СУБД, які залишаються найбільш популярними до сьогоднішнього дня. У 1980-х роках стали з'являтися об'єктно-орієнтовані СУБД, які оптимально підходять для зберігання даних про складні об'єкти. Також важливим етапом в історії розвитку СУБД було створення мови структурованих запитів SQL у 1970-х роках.

Останнім часом спостерігається активний розвиток хмарних СУБД, що надають можливість користувачам отримувати доступ до баз даних через Інтернет.

Системи управління базами даних (СУБД) можна класифікувати за різними критеріями, такими як модель даних, тип доступу, архітектура, та інші. Ось деякі основні типи СУБД та їхні особливості:

1. Реляційні СУБД:

- Основані на моделі реляційних таблиць, де дані організовані у вигляді таблиць з рядками і стовпцями.
- Використовують мову структурованих запитів (SQL) для маніпулювання даними.
- Дуже поширені та мають широке застосування в бізнесі та інших галузях.

2. Об'єктно-орієнтовані СУБД:

- Орієнтовані на роботу з об'єктами, що дозволяє зберігати та обробляти дані як об'єкти з методами та властивостями.

- Підтримують спадкування, поліморфізм та інші принципи об'єктно-орієнтованого програмування.
- Зазвичай використовуються для зберігання складних структур даних, наприклад, в області наукових досліджень або геоінформаційних систем.

3. Ієрархічні та мережеві СУБД:

- Організовані у вигляді ієрархічної (деревоподібної) або мережевої структури, де кожен запис має одного або більше батьків.
- Ієрархічні СУБД використовують модель, де дані представлені у вигляді дерева з батьківськими та дочірніми вузлами.
- Мережеві СУБД використовують модель, де кожен запис може мати кілька батьківських та дочірніх вузлів, утворюючи складні мережеві структури.

4. Хмарні СУБД:

- Забезпечують доступ до баз даних через Інтернет, зазвичай через послуги хмарного обчислення.
- Надають гнучкість та масштабованість, оскільки користувачі можуть використовувати ресурси хмарної інфраструктури за потребою.
- Забезпечують високу доступність та автоматизоване резервне копіювання даних.

Це лише кілька основних типів СУБД, існують також гібридні та спеціалізовані рішення, які використовуються для конкретних завдань або галузей діяльності.

Системи керування базами даних можна класифікувати за різними параметрами. Серед таких параметрів можна виділити модель даних (наприклад, реляційні, ієрархічні, мережеві), функціональне призначення (оперативні або аналітичні), розподіл (централізовані або розподілені), тип використання (SQL-орієнтовані або NoSQL) та ліцензію (відкриті або пропрієтарні).

Системи управління базами даних (СУБД) відіграють важливу роль у збереженні та організації інформації. Вони забезпечують ефективне та структуроване зберігання даних, наприклад, в інтернет-магазині, де вони відповідають за зберігання інформації про товари, замовлення та клієнтів. Більше

того, за допомогою інструментів вилучення та оновлення можна керувати цими даними, забезпечуючи ефективну роботу з базою даних. Оптимізація запитів дозволяє забезпечити швидкий доступ до інформації, що особливо важливо у галузях, де необхідна миттєва реакція, наприклад, у медичних закладах для оперативного отримання медичних записів.

У роботі СУБД важливу роль відіграють функції аутентифікації та авторизації. Це означає, що система контролює доступ, вимагаючи від користувачів надавати свої облікові дані та визначаючи, які конкретні дії вони можуть здійснювати. Наприклад, в медичних установах такі заходи гарантують захист особистої медичної інформації. Шифрування даних додатково забезпечує захист конфіденційної інформації, додавши ще один рівень безпеки. Крім того, СУБД надають інструменти моніторингу та ведення журналу, які відстежують активність користувачів, допомагають виявляти можливі загрози та відновлювати дані після випадкових втрат.

3.2 Реляційні моделі СУБД

3.2.1 MySQL: функціонал та застосування

Реляційні моделі систем управління базами даних (СУБД) використовуються для зберігання та управління даними у вигляді таблиць, що містять рядки і стовпці. Кожна таблиця представляє собою відношення між даними, де кожен рядок відповідає окремому запису, а кожний стовпчик представляє атрибут цих записів. Взаємозв'язки між таблицями встановлюються за допомогою ключів, які пов'язують одну таблицю з іншою. Реляційні моделі СУБД забезпечують структуроване та ефективне зберігання даних, а також простий та зручний доступ до них за допомогою мови запитів SQL.

Цей вид баз даних є найдавнішим: теоретичні основи цього підходу встановив британський вчений Едгар Кодд у 1970 році. У цих базах даних інформація представлена у вигляді таблиць з рядків і стовпців. В рядках містяться дані про

об’єкти (значення властивостей), а стовпці є властивостями цих об’єктів (поля).

Складні взаємозв'язки між об'єктами в реляційних базах даних моделюються за допомогою зовнішніх ключів - посилань на інші таблиці. Це дозволяє проектувати базу даних з погляду нормалізації - мінімізації збільшення при описі властивостей об'єктів.

Наприклад, у випадку ресторанного меню, кожна страва має свою вагу, ціну, назву, калорійність і категорію, до якої вона належить: гарячі страви, холодні страви, перші страви, десерти, салати і т.д. Зв'язок між стравами та категорією встановлюється за допомогою поля-посилання на індекс категорії в таблиці страв.

Цей підхід дозволяє:

1. Мінімізувати обсяг бази даних: немає необхідності повторювати назву категорії для кожної страви.

2. Підвищити цілісність системи: у цьому прикладі всі страви пов'язані з категоріями меню, тому додавання страви без категорії неможливе, так само як і додавання категорії, якої не існує.

3. Спростити масштабування: нові страви можуть бути додані до існуючих категорій.

4. Підвищити стійкість до відмов: оптимальна організація схеми таблиць дозволяє запитам на вибірку і агрегацію працювати з меншою кількістю даних, тому вони виконуються швидше, ніж без нормалізації.

MySQL - це одна з найпопулярніших в світі систем управління базами даних (СУБД), яка використовується для зберігання, організації та отримання доступу до даних. Вона надає широкий функціонал і застосовується у багатьох галузях, включаючи веброзробку, аналітику даних, електронну комерцію та багато інших.

MySQL - це система керування базами даних (СКБД), яка використовується для зберігання та управління великими обсягами даних у реляційній формі. SQL (Structured Query Language) використовується для формування запитів до бази даних. Це дозволяє здійснювати вибірки, агрегації, угруповання даних, змінювати та видаляти записи, а також модифікувати структуру бази даних.

MySQL підтримує як нормалізацію, так і денормалізацію даних. Нормалізація

використовується для розбиття даних на окремі таблиці для уникнення дублювання та забезпечення цілісності. Денормалізація, навпаки, полягає у зведенні даних з різних таблиць для полегшення доступу до них.

Важливо враховувати, що використання денормалізації потребує обгрунтування, оскільки вона може призвести до втрати ефективності бази даних.

Переваги реляційного підходу включають можливість визначення складних відносин між об'єктами, підтримку нормалізації та денормалізації даних, а також використання структурованої мови запитів.

Проте, жорстка структура реляційних баз даних може бути недоліком, особливо у випадках, коли необхідно працювати з недостатньо структурованою інформацією.

Приклади реляційних баз даних, подібних до MySQL, включають MariaDB, PostgreSQL та SQLite.

Основний функціонал MySQL включає наступне:

1. Структурування даних: MySQL дозволяє створювати бази даних з різноманітними таблицями, які можуть містити дані будь-якого типу, такі як числа, текст, дати тощо.

2. Мову запитів: Вона підтримує стандартну мову запитів SQL, що дозволяє виконувати різноманітні операції з даними, такі як вибірка, вставка, оновлення та видалення.

3. Безпека даних: MySQL надає можливості для захисту даних, включаючи автентифікацію користувачів, контроль доступу та шифрування даних для забезпечення конфіденційності.

4. Оптимізація запитів: Вона має інструменти для оптимізації та покращення продуктивності запитів, такі як індекси, вигляди, транзакції та багато інших.

5. Реплікація та резервне копіювання: MySQL підтримує можливості реплікації, яка дозволяє створювати копії баз даних для резервування та реплікації даних на різних серверах для підвищення надійності та доступності.

6. Інтеграція з різними мовами програмування: Вона може бути легко інтегрована з різними мовами програмування, такими як PHP, Python, Java, що

робить її популярним вибором для розробників вебзастосунків.

MySQL широко використовується в різних сферах, включаючи веброзробку, аналіз даних, системи управління вмістом, електронну комерцію, телекомунікації та багато інших. Вона відома своєю швидкістю, надійністю та зручним інтерфейсом, що робить її однією з найбільш популярних виборів серед розробників.

3.2.2 Характеристики та унікальні особливості Oracle Database

Сервіси та продукти для баз даних Oracle пропонують замовникам високопродуктивні та вартісно-оптимізовані версії Oracle Database, що є передовою у світі конвергентною багатомодельною системою управління базами даних, а також мають в своєму арсеналі резидентні бази даних NoSQL та MySQL. Oracle Autonomous Database, доступна як на локальному рівні через Oracle Cloud@Customer, так і в Oracle Cloud Infrastructure, дозволяє замовникам спростити середовища реляційних баз даних та зменшити кількість навантажень на управління. Oracle представляє інтегровану векторну базу даних для підтримки генеративного штучного інтелекту та суттєвого підвищення продуктивності розробників. Oracle Database - це потужна та високоефективна система керування базами даних (СКБД), яка має кілька характеристик і унікальних особливостей:

1. Масштабованість: Oracle Database може легко масштабуватися від невеликих до дуже великих обсягів даних, що робить її ідеальним вибором для різних застосувань, від невеликих підприємств до великих корпорацій.

2. Висока продуктивність: Oracle Database відома своєю високою продуктивністю та швидкодією завдяки ряду оптимізаційних технологій, таких як оптимізатор запитів та кешування результатів.

3. Надійність: Система забезпечує високий рівень надійності завдяки механізмам резервування, відновлення після збоїв та вбудованим механізмам безпеки.

4. Розширені можливості: Oracle Database підтримує різні типи даних, включаючи структуровані та неструктуровані дані, а також може інтегруватися з

іншими технологіями та інструментами.

5. Висока безпека: Oracle Database володіє різноманітними заходами безпеки, включаючи автентифікацію, авторизацію, шифрування даних та аудит.

6. Широкий функціонал: Система має велику кількість функцій та можливостей, включаючи підтримку транзакцій, розподілену обробку даних, аналітичні можливості та багато іншого.

Унікальні особливості Oracle Database, які роблять її відомою та популярною серед користувачів, включають її потужність, масштабованість, високу продуктивність, надійність та широкий функціонал, що робить її одним з провідних вирішень у сфері управління базами даних.

3.2.3 Оцінка переваг та використання Microsoft SQL Server

Microsoft SQL Server - це програмне забезпечення типу реляційної системи управління базами даних (RDBMS), яке є широко використовуваним у різних сферах. Розроблене корпорацією Microsoft, воно є масштабованим і може бути використане на різних платформах, від персональних комп'ютерів до хмарних серверних мереж. Проте, при використанні Microsoft SQL Server слід враховувати вимоги до апаратного та програмного забезпечення для ефективної роботи.

З моменту свого випуску в 1989 році, Microsoft SQL Server завоював популярність серед користувачів баз даних і залишається одним із важливих інструментів у цій сфері. Розробка SQL Server була дуже перспективною, починаючи з першої версії SQL Server 1.0 і продовжується донині. Його функціональні можливості розширилися, і тепер він не лише реляційна система управління базами даних (RDBMS), але й має вбудовані інструменти для бізнес-аналітики та звітності. Як один з продуктів відомої компанії Microsoft, Microsoft SQL Server відіграє ключову роль у світі управління базами даних. Згідно з назвою, це програмне забезпечення здійснює управління базами даних за допомогою мови запитів SQL. SQL - це загальновизнана мова, що використовується для доступу до даних, збережених у базах даних.

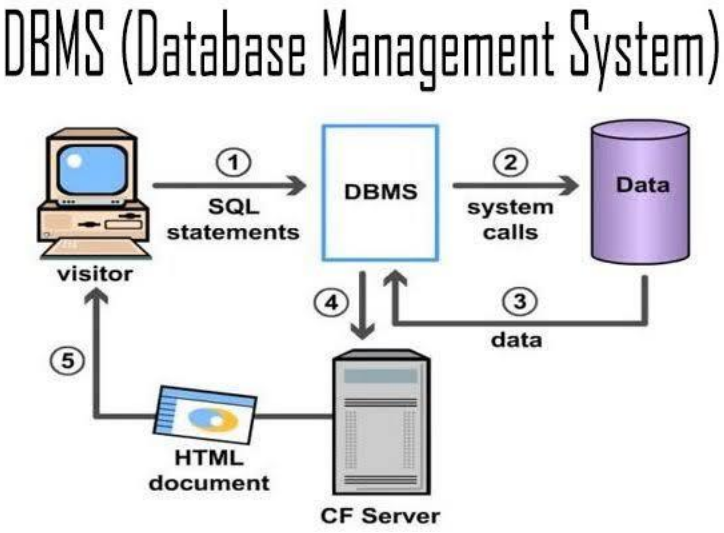

Рисунок 3.1 – У правління базами даних за допомогою мови запитів SQL

Ця функція має велике значення в світі програмування, оскільки багато розробників і програмістів використовують Microsoft SQL Server. Незалежно від вибраної мови програмування, наявність Microsoft SQL Server допомагає в розробці програм або додатків, які використовують сервери баз даних. Ще однією особливістю, якою цікавляться у Microsoft SQL Server, є його можливість створювати дзеркальне відображення та кластеризацію баз даних.

Наразі Microsoft SQL Server має більше 7 поколінь і доведено, що вони доступні на різних типах комп'ютерів, як 32-бітні, так і 64-бітні, та мають багато інших переваг. Однак, незважаючи на те, що цей продукт належить до великої компанії, Microsoft, це не означає, що він не має недоліків. Нижче наведено опис переваг Microsoft SQL Server:

1. Сумісність з різними версіями операційної системи Windows: Будучи продуктом Microsoft, який керується і підтримується тією ж компанією, що і операційні системи Windows, Microsoft SQL Server працює добре на всіх версіях ОС Windows. Від Windows XP до Windows 10, можна встановити Microsoft SQL Server, і він працюватиме безперервно та ефективно.

2. Можливість реалізації кластеризації даних: Кластеризація даних - це метод групування даних в різні кластери. Microsoft SQL Server має функціонал, який

полегшує реалізацію кластеризації даних, що стає важливою особливістю для розробників і програмістів. За допомогою цієї функції кластеризації даних можна спростити управління базами даних.

3. Централізоване управління базами даних: За допомогою Microsoft SQL Server управління базами даних може бути централізовано. Це дозволяє зменшити можливість помилок або проблем зв'язку при обробці та реалізації бази даних, оскільки вона використовує лише один комп'ютер як центр управління.

4. Можливість резервного копіювання бази даних: Microsoft SQL Server також має засіб резервного копіювання баз даних як запобіжний захід у разі виникнення перебоїв під час впровадження. Крім того, цю функцію резервного копіювання бази даних можна використовувати неодноразово, якщо буде побудовано подібну систему в майбутньому. Це також значно спрощує процес переміщення бази даних на інші пристрої.

5. Функції відновлення та відновлення даних: Microsoft SQL Server також має функції відновлення та відновлення даних, що стає корисним у випадку пошкодження частини бази даних, щоб можна було відновити та відновити дані.

Однак, нарікання на деякі недоліки продукту залишаються актуальними, і деякі з них не були вирішені або переборені дотепер.

1. Можна використовувати лише в ОС Windows: Microsoft SQL Server залишається доступним тільки для користувачів ОС Windows, виключаючи тих, хто працює на LINUX або MAC OS. Це обмеження може бути причиною того, що Microsoft утримує SQL Server тільки на платформі Windows, що відповідає їх стратегії.

2. Висока вартість придбання: Ціна Microsoft SQL Server є високою, що характерно для більшості програмного забезпечення для Windows. Однак ця вартість виправдана багатим функціоналом та можливостями, що надається.

3. Обмежений вибір мов програмування: SQL Server популярний серед програмістів, оскільки сумісний головним чином з мовою програмування .Net. Отже, користувачам, які використовують інші мови програмування, можуть знадобитися додаткові зусилля для роботи з ним.

4. Не підходить для масштабних баз даних: Хоча SQL Server має велику популярність, він може бути менш ефективним у великих обсягах даних порівняно з Oracle. Oracle вважається більш придатним для масштабних проектів, і це може бути перешкодою для використання SQL Server у деяких випадках.

3.3 Нереляційних моделі СУБД

3.3.1 Застосування MongoDB: особливості та перспективи

Протягом тривалого часу реляційні бази даних переважали та вважалися єдиною альтернативою. Проте з розвитком Інтернету, зростанням обсягу інформації та кількості пристроїв, стали з’являтися альтернативні рішення NoSQL. Спочатку це були прості сховища типу ключ-значення, а потім більш складні: графо-, атрибуто- та документо-орієнтовані. Усі вони були спрямовані на надання користувачам нових можливостей, які традиційні SQL рішення не могли забезпечити. Однією з таких систем є документо-орієнтована база даних MongoDB.

Документоорієнтовані бази даних вирішують обмеження реляційних БД, пов'язані з жорсткою структуризацією та типізацією даних. У цих базах даних інформація зберігається у структурованих форматах, таких як XML, JSON або BSON. Ключовим елементом є адресний доступ до даних за ключем, що дозволяє ефективно оптимізувати доступ до них.

Ці бази даних ідеально підходять для ситуацій, коли вам потрібно працювати з різноманітними структурами даних. Вони дозволяють швидко розробляти системи та сервіси, які опрацьовують дані з різним набором властивостей. Крім того, вони легко масштабуються та можуть змінювати структуру даних за потреби.

Деякі приклади документоорієнтованих баз даних включають MongoDB, RethinkDB, CouchDB, DocumentDB.

MongoDB є системою керування базами даних, входить до сімейства NoSQL, і використовує документо-орієнтовану модель. Це означає, що вся інформація зберігається у вигляді документів у форматі BSON, який базується на JSON і

призначений для бінарного кодування. У MongoDB дані організовані у колекціях, що відповідає таблицям у реляційних базах даних, документи в цих колекціях відповідають кортежам, а поля - атрибутам.

Одна з перших особливостей MongoDB - це простота представлення даних. Наприклад, якщо потрібно зберегти дані соціальної мережі, можна використовувати документи для адреси статті, коментарів користувачів, хештегів та зображень. Нормалізація таких даних у MongoDB здійснюється за допомогою ER-моделі, що дозволяє зображати відносини між сутностями.

3.3.2 Роль Couchbase у сучасних нереляційних СУБД

Couchbase є однією з передових нереляційних систем управління базами даних (NoSQL).

Couchbase Server, як система управління базами даних (СУБД) класу NoSQL поєднує у собі можливості двох популярних проектів: Apache CouchDB та Membase. Основна ідея полягає в створенні документоорієнтованих баз даних, схожих на ті, що використовуються в Apache CouchDB, у поєднанні зі сховищами даних у форматі "ключ-значення", як у Membase.

Одна з ключових переваг Couchbase Server полягає в його сумісності з протоколом memcached, що дозволяє використовувати його з існуючими додатками і системами, що використовують memcached. Це робить Couchbase привабливим варіантом для розширення або заміни існуючих систем.

Крім того, завдяки поєднанню функціональності CouchDB та Membase, Couchbase Server надає можливість використовувати мови запитів та індексів, схожі на ті, що використовуються в CouchDB, а також протокол доступу до даних, схожий на Membase. Це забезпечує високу сумісність з існуючими додатками, що використовують ці системи.

Різні SDK, підготовлені для різних мов програмування, таких як Java, Ruby, .NET, C/C++, PHP, Node.js, Go та Python, дозволяють розробникам легко інтегрувати Couchbase Server у свої додатки та розробляти з ними.

Ця функція дозволяє зберігати дані як на одному сервері, так і у вигляді розподіленої системи, яка розміщує дані на групі серверів. Вона також має вбудовані засоби для забезпечення високої доступності та самовідновлення в разі відмови одного з серверів. Дані можуть бути дубльовані на різних серверах та розподілені по різних дата-центрах. Система підтримує як однонаправлену, так і двонаправлену реплікацію. Крім того, можливе створення як первинних, так і вторинних індексів, а також індексів за декількома ключами. Для оптимізації продуктивності використовуються вбудовані механізми кешування в оперативній пам'яті та автоматична генерація індексів.

Крім можливості зберігання даних у форматі «ключ-значення», Couchbase також пропонує концепцію документоорієнтованого зберігання, де кожен документ має унікальний ідентифікатор, версію і набір іменованих полів у форматі «ключ-значення». Ця модель дозволяє визначати документи у форматі JSON без необхідності визначення схеми заздалегідь. Запити та індексація даних можуть виконуватися згідно з парадигмою MapReduce. Для організації псевдоструктурованого набору даних пропонується концепція формування представлень.

Ця база даних спеціалізується на розподіленому зберіганні, швидкості доступу та високій доступності даних. Ось кілька ключових ролей Couchbase у сучасних нереляційних СУБД:

1. Розподілена архітектура: Couchbase розроблений для роботи у розподіленій середовищі, де дані розміщені на кількох вузлах серверів. Це забезпечує високу доступність та масштабованість.

2. Швидкість та продуктивність: Couchbase відомий своєю високою швидкістю доступу до даних, особливо в умовах великого обсягу даних та великого навантаження. Він використовує усунення проблем блокування, кешування та інші техніки для оптимізації продуктивності.

3. Гнучкість сховища даних: Couchbase дозволяє зберігати структуровані, напівструктуровані та неструктуровані дані, що робить його відмінним вибором для сучасних додатків, які потребують різноманітності даних.

4. Підтримка SQL-подібних запитів: Couchbase включає N1QL (або Nickel), мову запитів, що схожа на SQL. Це дозволяє розробникам простіше працювати з базою даних, виконуючи різноманітні операції, такі як вибірка, оновлення та об’єднання даних.

5. Підтримка масштабування горизонтального: Couchbase легко масштабується горизонтально шляхом додавання нових вузлів до кластера. Це дозволяє обробляти великі обсяги даних та збільшувати продуктивність без перерви в роботі.

У цілому, Couchbase грає важливу роль у сучасних додатках, які вимагають швидкості, масштабованості та гнучкості у роботі з дани

3.3.3 Аналіз HBase: можливості та застосування

HBase є розподіленою системою керування даними, яка базується на моделі ключ-значення (key-value).

HBase - це база даних класу NoSQL з відкритим вихідним кодом, яка є частиною екосистеми Hadoop. Вона написана на Java та відноситься до категорії "сімейство стовпців", при цьому багато технічних рішень позичені з Google BigTable. HBase працює поверх розподіленої файлової системи HDFS і надає можливості, схожі на BigTable, для Hadoop, забезпечуючи надійний спосіб зберігання великих обсягів розріджених даних.

У HBase підтримується стиснення, операції в пам'яті та фільтр Блума для кожного базового стовпця, реалізовані відповідно до документації BigTable. Таблиці в HBase можуть бути використані як вхід та вихід для реалізації MapReduce у проекті Hadoop, і можуть бути отримані не лише через Java API, але і через API (REST, Avro, Thrift). Проекти Phoenix та Trafodion забезпечують SQL-доступ до даних, керованих HBase.

HBase застосовується для управління даними у ряді великих проектів, включаючи Facebook, який використовував HBase для платформи повідомлень у період з 2010 по 2018 рік (пізніше перейшов на MyRocks). Серед постійних

користувачів також є такі компанії, як Adobe, StumbleUpon, Twitter, Yahoo!, який експлуатує кластер HBase з 3 тисяч узлів

Проект HBase був розпочатий у 2006 році Чедом Уолтерсом і Джимом Келлерманом з компанії Powerset, які мали потребу у обробці великих обсягів даних для створення пошукової системи на природній мові. Перший прототип був заснований на статті, опублікованій у 2005 році співробітниками Google про систему Bigtable. У лютому 2007 року Майк Кафарелла написав початковий код системи, подальший розвиток якої вів Джим Келлерман. Невдовзі проект здобув велику увагу розробників і отримав високий пріоритет в Apache Software Foundation.

Перша версія HBase була включена до поставки Hadoop 0.15.0 у жовтні 2007 року. У травні 2010 року система перейшла з категорії підпроектів Hadoop у категорію проектів верхнього рівня фонду Apache. З 2012 року щорічно проводиться конференція розробників та користувачів системи HBaseCon.

У 2015 році вийшла версія 1.0 системи HBase, що була визнана як важливий момент фондом та незалежними спостерігачами, свідчачи про зрілість продукту. З другої половини 2010-х років інтерес до системи також зрос за рахунок появи SQL-інтерфейсів до неї, таких як Phoenix і Trafodion.

Основною особливістю HBase є те, що вона спроектована для зберігання великих обсягів даних на різних серверах у розподіленому середовищі, що дозволяє легко масштабувати систему.

Основні можливості та застосування HBase включають:

1. Горизонтальне масштабування: HBase дозволяє легко масштабувати систему шляхом додавання нових серверів у кластер. Це робить її ідеальним рішенням для зберігання великих обсягів даних, таких як веблоги, соціальні медіа або дані сенсорів Інтернету речей (IoT).

2. Підтримка високої доступності: HBase має вбудовані засоби для забезпечення високої доступності даних. Вона автоматично реплікує дані на різні сервери, що дозволяє уникнути втрати даних у разі відмови сервера.

3. Швидкий доступ до даних за ключем: Основна модель даних HBase базується на ключ-значення, що дозволяє швидко отримувати доступ до даних за їх

ключем.

4. Підтримка структурованих та неструктурованих даних: HBase дозволяє зберігати як структуровані, так і неструктуровані дані. Це робить її корисним інструментом для різноманітних застосувань, від аналітики великих даних до зберігання документів і зображень.

5. Можливості аналізу даних: HBase ідеально підходить для розв'язання задач аналізу великих обсягів даних, так як вона дозволяє ефективно зберігати і обробляти великі обсяги даних у реальному часі.

Узагальнюючи, HBase - це потужна розподілена система керування даними, яка відмінно підходить для зберігання великих обсягів даних і розв'язання різноманітних задач аналізу даних.

Висновки до третього розділу

Тому, роль систем управління базами даних (СУБД) полягає у забезпеченні організованого доступу до інформації, уникненні повторення даних та гарантуванні цілісності та безпеки даних.

Системи управління базами даних (СУБД) можна класифікувати за різними критеріями, такими як модель даних, тип доступу, архітектура, та інші.

Системи керування базами даних можна класифікувати за різними параметрами. Серед таких параметрів можна виділити модель даних (наприклад, реляційні, ієрархічні, мережеві), функціональне призначення (оперативні або аналітичні), розподіл (централізовані або розподілені), тип використання (SQL-орієнтовані або NoSQL) та ліцензію (відкриті або пропрієтарні).

Реляційні моделі систем управління базами даних (СУБД) використовуються для зберігання та управління даними у вигляді таблиць, що містять рядки і стовпці. Кожна таблиця представляє собою відношення між даними, де кожен рядок відповідає окремому запису, а кожний стовпчик представляє атрибут цих записів. Взаємозв'язки між таблицями встановлюються за допомогою ключів, які пов'язують одну таблицю з іншою. Реляційні моделі СУБД забезпечують структуроване та

ефективне зберігання даних, а також простий та зручний доступ до них за допомогою мови запитів SQL.

Нереляційні бази даних ідеально підходять для ситуацій, коли вам потрібно працювати з різноманітними структурами даних. Вони дозволяють швидко розробляти системи та сервіси, які опрацьовують дані з різним набором властивостей. Крім того, вони легко масштабуються та можуть змінювати структуру даних за потреби. Деякі приклади документоорієнтованих баз даних включають MongoDB, RethinkDB, CouchDB, DocumentDB.

РОЗДІЛ 4
РОЗРОБКА І ВПРОВАДЖЕННЯ БАГАТОФАКТОРНОЇ АВТЕНТИФІКАЦІЇ У ВЕБЗАСТОСУНКАХ РЕЛЯЦІЙНИХ СУБД

4.1 Програмна реалізація механізму багатофакторної автентифікації

Вебдодаток був реалізований з використанням різноманітних технологій для фронтенду та бекенду, забезпечуючи динамічну та інтерактивну роботу на сервері. Фронтенд частину розроблено з використанням HTML, CSS і React.js, що дозволяє створювати ефективні та користувачами зручні інтерфейси.

Бекенд був написаний мовою програмування PHP, що працює у поєднанні з MySQL базою даних. Ця комбінація технологій надає можливість для ефективного зберігання та обробки даних, що потрібні для вебдодатку.

Додаток також включає в себе функціонал двофакторної автентифікації (2FA), що забезпечує високий рівень безпеки для користувачів. Інтеграція з 2FA дозволяє забезпечити додатковий шар захисту для доступу до облікових записів користувачів. Крім того, в додатку використовуються різноманітні бібліотеки для роботи з різними сервісами, наприклад - Google. Ці бібліотеки дозволяють зручно взаємодіяти з іншими сервісами та використовувати їх функціонал у вебдодатку.

У вебдодатку також реалізовано можливість відправки та отримання даних з бази даних, що дозволяє користувачам ефективно взаємодіяти з інформацією, збереженою на сервері. Ця функціональність дозволяє користувачам зберігати, витягувати, оновлювати та видаляти дані з бази даних за допомогою вебінтерфейсу

Додаток також включає в себе роутинг для ефективного керування URL-адресами та навігації в додатку, а також механізми авторизації та реєстрації, що дозволяють користувачам створювати облікові записи та управляти своїм доступом до функціоналу додатку.

Крім того, в додатку існує можливість роботи на дошці, де користувачі можуть створювати картки для розташування в колонках, а також виконувати

різноманітні дії, такі як редагування, видалення, зміна місця розташування та інші. Цей функціонал дозволяє користувачам організовувати та управляти своїми завданнями, проекти або будь-якими іншими типами інформації у зручному інтерфейсі вебдодатку.

4.1.1 Опис модуля реєстрації та авторизації користувачів

В даному модулі ми маємо авторизацію та реєстрацію користувача.

Спочатку ми підвантажуємо функції бази, для подальшої роботи та запису потрібних нам данних. Наприклад запису нового користувача або ж перевірки його данних.

В __construct ми ініціліазуємо роботу згідно запитів з фронтендної частини (сайту)

Функція registerUser - реєструє користувача у базі.

Спочатку ми перевіряємо чи заповнені поля, далі робимо перевірку на те, чи присутній користувач з таким емейлом в нашій системі. Після цього - створюємо аккаунт.

Функція loginUser - забезпечує користувачеві вхід до сайту.

Спочатку ми перевіряємо чи заповнені поля. Далі отримуємо інформацію про користувача, якщо вона присутня, перевіряємо чи співпадають паролі.

Якщо користувач має додаткову авторизацію (2FA) - ми повертаємо його до вводу коду. Якщо ж він немає, тоді ми ініціалізуємо його сесію в системі.

Фукнція authLoginUser - забезпечує користувачеві вхід до сайту, якщо він має додаткову авторизацію (2FA).

Перевіривши введений в системі код, за допомогою модуля 2FA, ми передаємо емейл користувача, якого потрібно залогінити.

Частина коду функції __construct:

```
public function __construct() {
    if(isset($_POST['action'])) {
```

```
$this->database = new DataBase();
if($_POST['action'] == "register"){
```

Частина коду функції registerUser:

```
echo $this->registerUser($_POST['email'],$_POST['password']);
```

Частина коду функції loginUser:

```
echo $this->loginUser($_POST['email'],$_POST['password']);
```

Частина коду функції authLoginUser:

```
else if($_POST['action'] == "login_2fa"){
    echo $this->authLoginUser($_POST['email']);
public function authLoginUser($email){
    /*
     * Get user info
     */
    $user_info = $this->database >retrieveUserAccount($email);
```

4.1.2 Опис модуля двухфакторної авторизації

В даному модулі можна переглянути роботу модулю двухфакторної авторизації.

Спочатку ми підвантажуємо функції бази, для подальшої роботи та запису потрібних нам данних.

Далі підвантажуємо гугл бібліотеку та потрібні для роботи класи.

Прописуємо змінні для роботи з АПІ.

В __construct ми ініціліазуємо роботу АПІ, отримуєм секретний ключ, а далі працюємо взалежності від надісланої з фронта дії.

Функція createQRCode - створює зображення qr коду, яке дасть змогу авторизувати додаток та створити додатковий захист.

В ній ми підвантажуємо сервіс для геренації свг, створюємо зображення та передаєм на сайт.

Функція getSecretKey - нам потрібна для того, щоб шифрувати роботу з АПІ, та мати захист під час отримання та перевірки коду.

Спочатку ми отримуємо код з бази (якшо він існує). Якшо ж він відсутній, то ми генеруємо новий та записуємо до бази.

Функція setUser2FA - встановлює додаткову авторизацію для юзера.

Фукнція check2FAcode - перевіряє надісланий користувачем код ,та порівнює його з тим, що має бути, забезпечивши додатковий захист до адмін. панелі

Частина коду функції __construct:

```
public function __construct(){
    if(isset($_POST['action'])) {
        /*
         * Initialize Google2FA
         */
        $this->google2fa = new Google2FA();
```

Частина коду функції createQRCode:

```
private function createQRCode(){
    /*
     * Use service for create QR code
     */
    $this->google2fa->setQrcodeService(
        new Bacon(
            new \BaconQrCode\Renderer\Image\SvgImageBackEnd()
        )
    );
    /*
     * Get the QR code SVG
     */
    $QRCodeInline = $this->google2fa->getQRCodeInline(
        $this->app_name,
        $this->app_email ,
```

```
            $this->secret_key
        );
        $base64EncodedSvg = base64_encode($QRCodeInline);
```

Частина коду функції getSecretKey:

```
private function getSecretKey(){
        $database = new DataBase();
        $this->secret_key = $database->getSecretKey();
        if(!isset($this->secret_key) || $this->secret_key == null){
            /*
            * Generate Secret Key
            */
            $this->secret_key = $this->google2fa->generateSecretKey();
            /*
```

Частина коду функції setUser2FA:

```
private function setUser2FA(){
        $database = new DataBase();
        if ($database->setUserAuth()) {
            return
```

Частина коду функції check2FAcode**:**

```
            $database->setSecretKey($this->secret_key);
        }
    }
private function check2FAcode($code){
        if ($this->google2fa->verifyKey($this->secret_key, $code)) {
            /*
             * Code is valid
             */
            return
```

4.1.3 Опис модуля створення картки у базі даних

В даному модулі ми маємо повністю повний функціонал взаємодії з картками, та їх додатковим функціоналом : колонки та коментарі.

Спочатку ми підвантажуємо функції бази, для подальшої роботи та запису потрібних нам данних. Наприклад додавання карток, зміна колонок, додавання коментарів, оновлення даних картки, тощо.

В __construct ми ініціліазуємо роботу згідно запитів з фронтендної частини (сайту)

Ми маємо перелік запитів, що забезпечують усю роботу з картками. Ось їх опис:

1. add_card - createCard() - створення картки.
2. update_card - updateCard - оновлення даних в картці.
3. delete_card - deleteCard() - видалення картки.
4. change_card_column - updateCard() - оновлення колонки в картці.
5. get_cards - getCards() - отримання всіх карток користувача.

Також ми маємо ще 2 функції коментарів, які опишемо в модулі нижче.

Функція createCard - забезпечує створення нової картки. В нас є певний набір змінних, таких як (колонка, назва, дата, id користувача і т.п) , які ми записуємо в базу даних. Після створення картки, ми повертаємо її інформацію для роботи на дошці.

Функція updateCard - забезпечує оновлення даних по конкретній картці за допомогою ідентифікатора картки. Ми передаємо потрібні нам для оновлення дані, та за допомогою запиту до бази, оновлюєм дані , задаючи умови, що оновлення для поточного ідентифікатора картки. Також за допомогою даної функції змінюється конкретна колонка картки.

Функція deleteCard() - видаляє картку з бази данних для поточного ідентифікатора картки.

Функція getCards() - надає інформацію про усі картки для поточного користувача. Картки містять в собі усю інформацію по картці, а також коментарі, які

були прописані для цієї картки. Самі картки є масивом, що розподілений по колонках з набіром карток.

Частина коду функції __construct:

```
public function __construct() {
    /*
     * Check for a POST request
     */
    if(isset($_POST['action']))
        $this->database = new DataBase();
switch ($_POST['action'])
        case 'add_card' :
                                                    echo $this->database-
>createCard($_POST['card_title'],$_POST['column_id']);
            break;
```

Частина коду запиту update_card:

```
case 'update_card' :
            echo $this->database->updateCard(
              $_POST['card_id'],
              array(
                'description' => $_POST['description'],
                'title' => $_POST['card_title'],
                'column_id' => $_POST['column_id'],
                'start_date' => $_POST['start_date'],
                'finish_date' => $_POST['finish_date']
            break;
```

Частина коду запиту delete_card:

```
case 'delete_card' :
            echo $this->database > deleteCard($_POST['card_id']);
            break;
```

Частина коду запиту change_card_column:

```
case 'change_card_column' :
                echo $this->database->updateCard($_POST['card_id'],array('column_id' =>
$_POST['column_id']));
            break;
```

Частина коду запиту get_cards:

```
case 'get_cards' :
            echo $this->database->getCards();
            break;
```

Частина коду функції createCard:

```
public function createCard($card_title,$column_id)
$query = $this->db->prepare("INSERT INTO cards
(title,column_id,user_id,date_created,start_date) VALUES ('" . $card_title . "','" .
$column_id . "','" . $this->user_id . "','" . date('Y-m-d H:i:s') . "','" . date('Y-m-d H:i:s') .
"')");
```

Частина коду функції updateCard:

```
public function updateCard($card_id,$data){
    $sets = "";
    foreach ($data as $column => $value){
       $sets .= $column." = '".$value."',";
    }
    $sets = substr($sets, 0, -1);
       $query = $this->db->prepare("UPDATE cards SET ".$sets." WHERE id = " .
$card_id);
```

Частина коду функції deleteCard:

```
public function deleteCard($card_id){
    $query = $this->db->prepare("DELETE FROM cards WHERE id = ".$card_id);
    if ($query->execute()) {
        $query_comments = $this->db->prepare("DELETE FROM comments WHERE
card_id=".$card_id." AND user_id = ".$this->user_id);
       $query_comments->execute();
```

```
        return
```

Частина коду функції getCards:

```
public function getCards()
   {
       $query = $this->db->prepare("SELECT * FROM cards WHERE user_id ='" . $this->user_id . "'");
       $query->execute();
       $result = $query->get_result();
       if ($result->num_rows > 0) {
          $cards = [];
          while ($row = $result->fetch_assoc()) {
             $comments =  $this->getComments($row['id']);
             $row['comments'] = $comments;
             $cards[$row['column_id']][] = $row;
          }
          return
```

4.1.4 Опис модуля додавання коментарів у картку

Це частина модуля карток, яка взаємодія з коментарами. Надає змогу додавати, видаляти та отримувати їх.

Ми маємо 2 запити

1. add_comment - createComment() - створення коментаря.

2. delete_comment - deleteComment() - видалення коментаря.

Функція - createComment() забезпечує створення нового коментаря для заданої картки. Коментар має певний набір аргументів , це: ід. картки, ід. користувача, текст та дата створення. В цій функції ми повертаємо створений коментар для подальшої роботи з ним.

Функція - deleteComment() видаляє коментар з бази даних та картки для поточного ідентифікатора коментаря.

Частина коду функції та запиту add_comment - createComment():

```
public function createComment($card_id, $comment_text)
   {
                    $query = $this->db->prepare("INSERT INTO comments (card_id,user_id,text,date_created) VALUES (" . $card_id . "," . $this->user_id . ",'" . $comment_text . "','" . date('Y-m-d H:i:s') . "')");
      if ($query->execute()) {
         $inserted_id = $query->insert_id;
         $query_comment = $this->db->prepare("SELECT * FROM comments WHERE id = " . $inserted_id);
         $query_comment->execute();
         $result = $query_comment->get_result();
         return
```

Частина коду функції та запиту delete_comment - deleteComment():

```
public function deleteComment($comment_id){
            $query = $this->db->prepare("DELETE FROM comments WHERE id=".$comment_id." AND user_id = ".$this->user_id);
      if ($query->execute()) {
         return json_encode(array('status' => 'success'));
      }
      else return json_encode(array('status' => 'Database error'));
   }
```

4.1.5 Опис модуля статистики та прорахування середнього часу угоди

Цей модуль надає змогу отримати інформацію щодо карток.В таблиці присутня інформація щодо нових карток, які створились не пізніше чим 5 днів тому. Завершених угод, а саме ті, що мають дату закінчення. А також середній час укладання угоди.

Ми маємо функцію getAllCards() - яка повертає усі угоди користувача,

включаючи в себе завершені, нові, та ті що в процесі.

Фукнція getNewCards() - отримує усі картки, що були створені на протязі 5 останніх днів. Таким чином можна моніторити та дізнаватись, скільки в середньому сворюється угод за робочий тиждень, а також розуміти приблизну кількість угод на день.

Після того, як ми отримуємо всі дані, ми формуємо таблицю з інформацією.

Модуль є динамічним для кожного користувача, таким чином, кожний користувач має свою статистику. А отже ми можемо ККД та KPI кожного із співробітників.

Частина коду функції getAllCards():

```
public function getAllCards(){
    $query = $this->db->prepare("SELECT * FROM cards WHERE user_id ='" . $this->user_id . "'");
    $query->execute();
    $result = $query->get_result();
    if ($result->num_rows > 0) {
        $count_finish_cards = 0;
        $time_finish_cards = 0;
        while ($row = $result->fetch_assoc()) {
```

Частина коду функції getNewCards():

```
public function getNewCards(){
    $current_date = date('Y-m-d');
    $start_date = date('Y-m-d', strtotime('-5 days'));
    $query = $this->db->prepare("SELECT * FROM cards WHERE user_id ='" . $this->user_id . "' AND date_created BETWEEN '".$start_date."' AND '".$current_date."'");
    $query->execute();
    $result = $query->get_result();
    if ($result->num_rows > 0) {
        return $result->num_rows;
    } else {
```

```
return '0';
```

4.1.6 Опис модуля фільтр угод за часом

Ця частина модуля, дозволяє відстежувати статистку за весь час, за місяць, а також за 7 днів. Що є корисним, для розуміння прогресу чи навпаки спаду по кількості угод. Таким чином можна дізнатись показники, які співробітник отримав за поточний вказаний час.

Функція getCardsByFilter() - робить запит до бази даних та отримує статистику карток за заданий проміжок часу. Ця функція є частиною модуля статистики, проте працює як окрема функція, не підвантажуючи лишню інформацію.

Також, як і попередні функції модуля статистики, ця функція працює для конкретного користувача, і ще для конкретного проміжку часу.

Частина коду функції getCardsByFilter():

```
public function getCardsByFilter($filter){
    $filter_sql = "";
    if(isset($filter) && $filter != ""){
        $current_date = date('Y-m-d');
        $start_date = date('Y-m-d', strtotime('-'.$filter.' days'));
            $filter_sql = " AND date_created BETWEEN '".$start_date."' AND 
'".$current_date."'";
    }
     $query = $this->db->prepare("SELECT * FROM cards WHERE user_id ='" . $this-
>user_id . "'".$filter_sql);
    $query->execute();
    $result = $query->get_result();
    if ($result->num_rows > 0) {
        $count_finish_cards = 0;
        $time_finish_cards = 0;
```

4.2 Програмна демонстрація роботи бази даних у вебзастоунку.

Розглянемо демонстрацію кожного кроку роботи нового співробітника компанії, який щойно приступив до виконання своїх обовязків на пристроях компнанії (телефон та ноутбук).

Розглянемо логіку кожного кроку нового користувача а саме менеджера коли він приступає до виконання своїх обовязків на пристроях компанії телефоні та ноутбуці компанії.

Вивчимо ретельно дії, які проводяться в першу чергу.

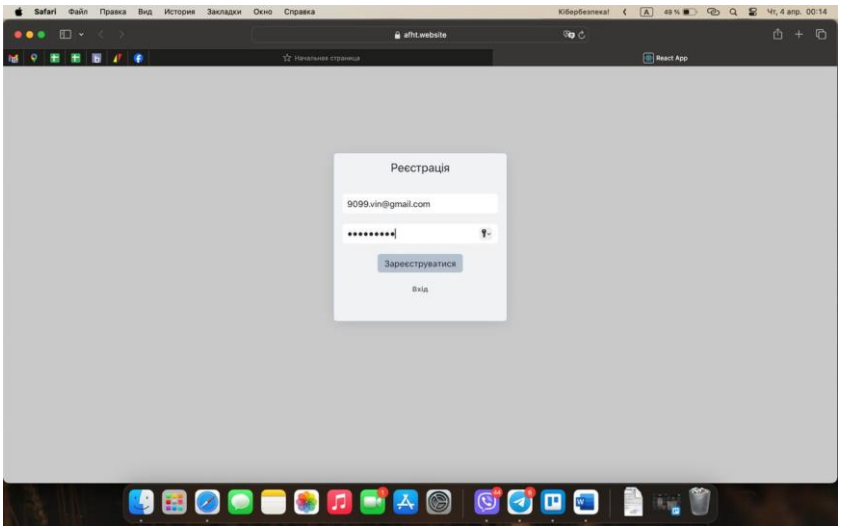

Рисунок 4.1 – Етап перший. Процес реєстрації

По-перше менеджер відкриває посилання нашої бази даних. По-друге, йому необхідно пройти реєстрацію. Тобто йому необхідно зареєструватися, для цього він повинен ввести свій логін та свій пароль, після чого він натискає кнопку “зареєструватися”, потім йому підтверджують реєстрацію та йому необхідно увійти.

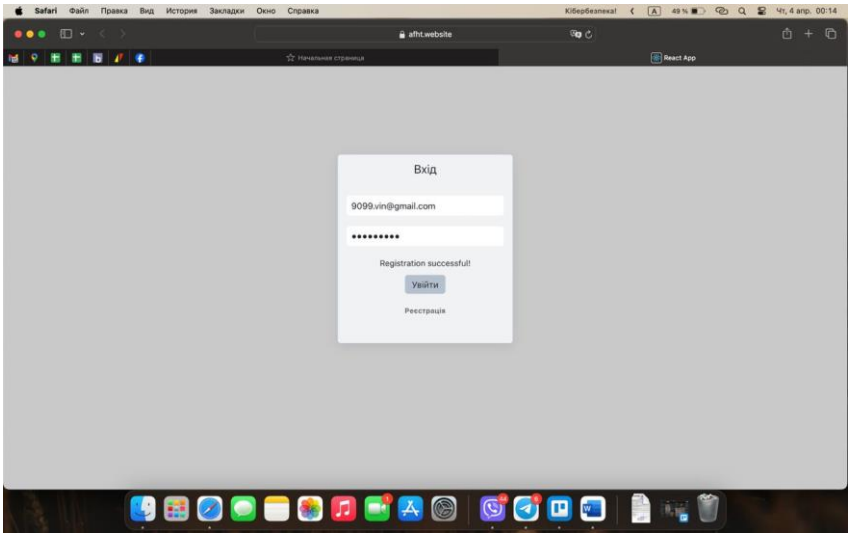

Рисунок 4.2 – Вхід до системи

Він входить під такими даними, що він вводив попередню і отримує у перший раз доступ до бази даних.

Це перша його авторизація і вона їде без багатофакторної складової, тобто вона не є багатофакторною.

В нього пуста база даних яку йому необхідно налаштувати і заповнити.

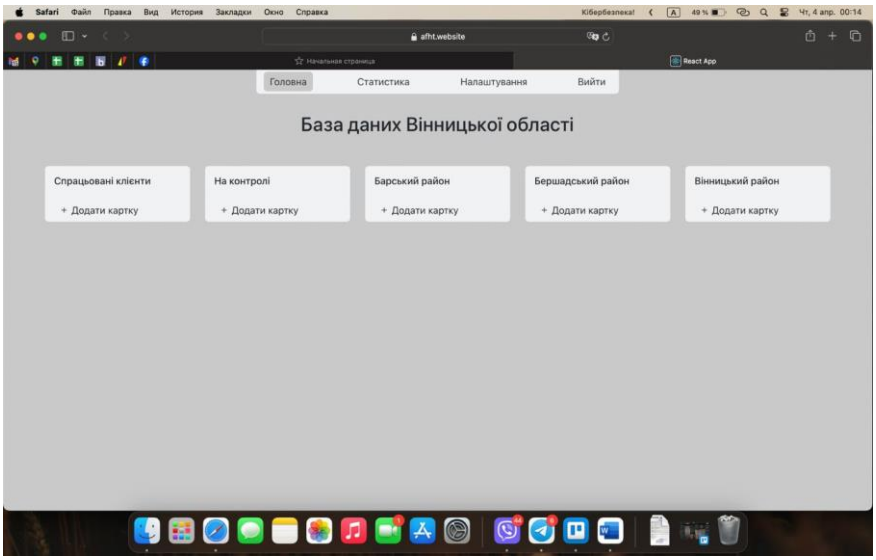

Рисунок 4.3 – База даних

По-четверте, наступним кроком іде підключення двофакторної аутентифікації.

Перше, що він робить коли він вже отримав доступ до бази даних, він виходить на налаштування і підключає собі двофакторну аутентифікацію на його акаунт.

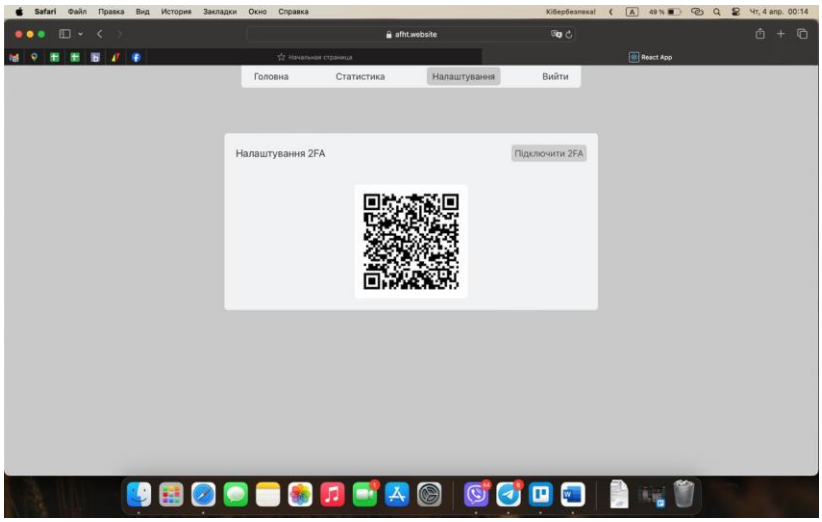

Рисунок 4.4 – Підключення двохфакторної аутентифікації у налаштуваннях

По-п’яте, він бере телефон, який йому надала компанія та інсталює на телефон додаток Google autentificator з Плей Маркета або Google Apple Store.

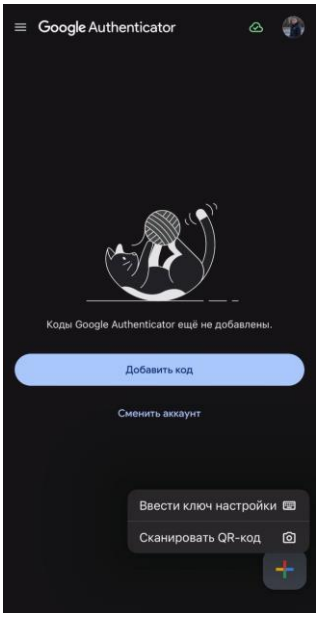

Рисунок 4.5 – Процес налаштування двохфакторної аутентифікації

Нажимає на “плюс” і нажимає на кнопку “сканувати QR код”.

Йому на ноутбуці компанія, який йому надала компанія для роботи є QR код та він його сканує телефоном.

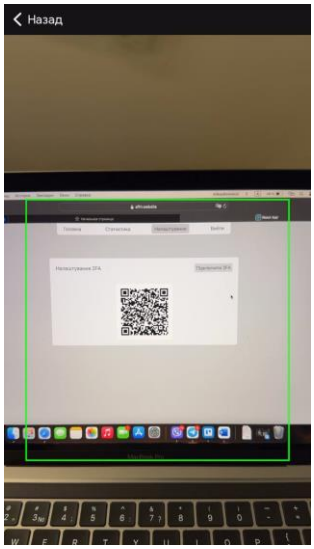

Рисунок 4.6 – Процес сканування QR кода

Телефоном автоматично в програму Google аутентификатор додається пристрій який буде авторизуватись.

І в нього є код, який оновлюється кожні 30 секунд.

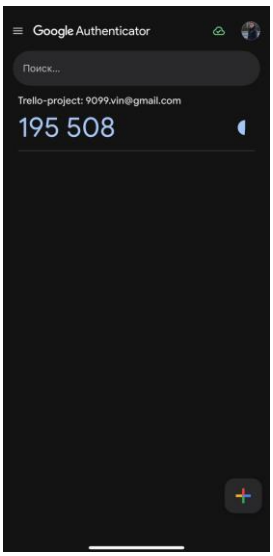

Рисунок 4.7 – Зразок кода

Потім, необхідно зробити наступний крок цього процесу. Відповідальна персона (менелжер) повинна заповнити свою базу даних. Мається на увазі увесь процес опрацювання даних клієнтів та фермерської бази: хто конкретно, особисті дані, адреса, район, уся додаткова інформація. Під час цієї роботи необхідно ретельно зосередитися, для того, щоб не втратити інформацію, а разом і з нею і клієнта.

Важливо також враховувати потенційних клієнтов, з якими ведуться перемовини та не втратити також і їх. Для цього ці дані можуть бути винесені до колонки “На контролі”.

Якщо вже є опрацьовані клієнти, то додається колонка “Опрацьовані клієнти”.

Далі, наступний крок, це власне функціонал карточки, в якому є наступне: є поле для того щоб назвати цю карточку, в даному випадку це, потім є поле для опису своєї карточки, а саме - адреса даного господарства та номер телефону контактної особи, потім є поля “Дата початку” та “Дата кінця”.

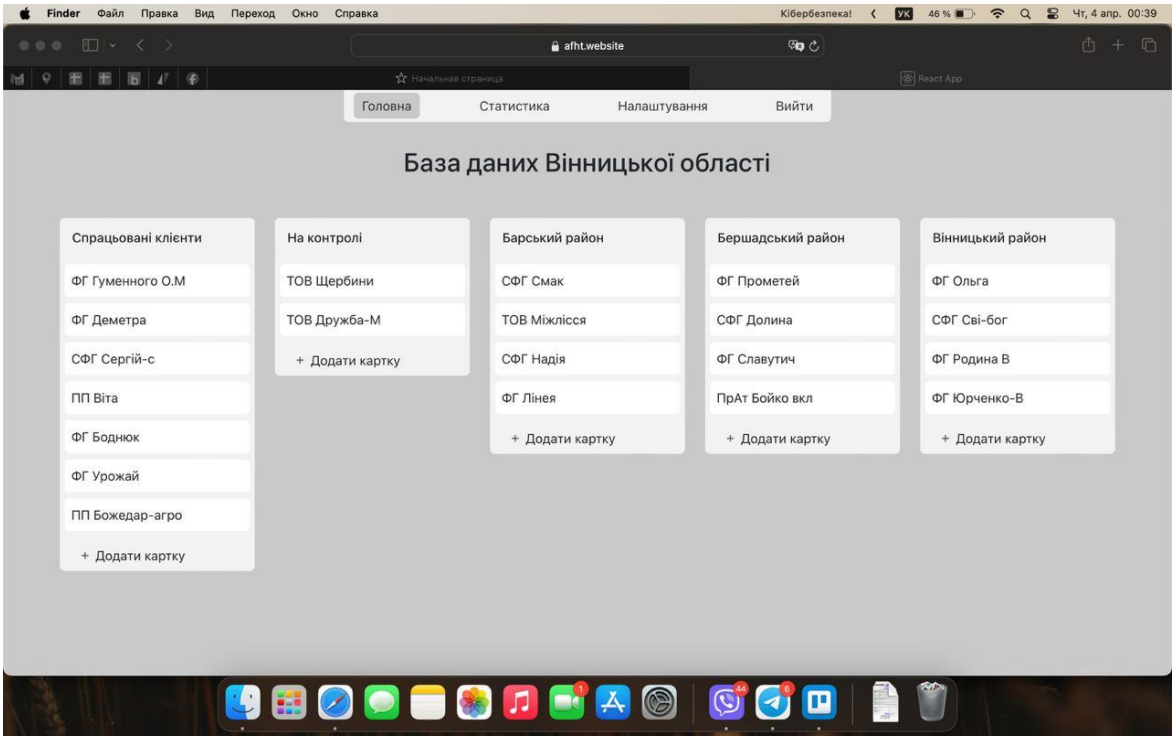

Рисунок 4.8 – База даних

Таким чином, менелжер повинен виставити дату початку спілкування з клієнтом і дату кінця, це тоді коли в нього була здійснена продажа, товар був завезений у господарство.

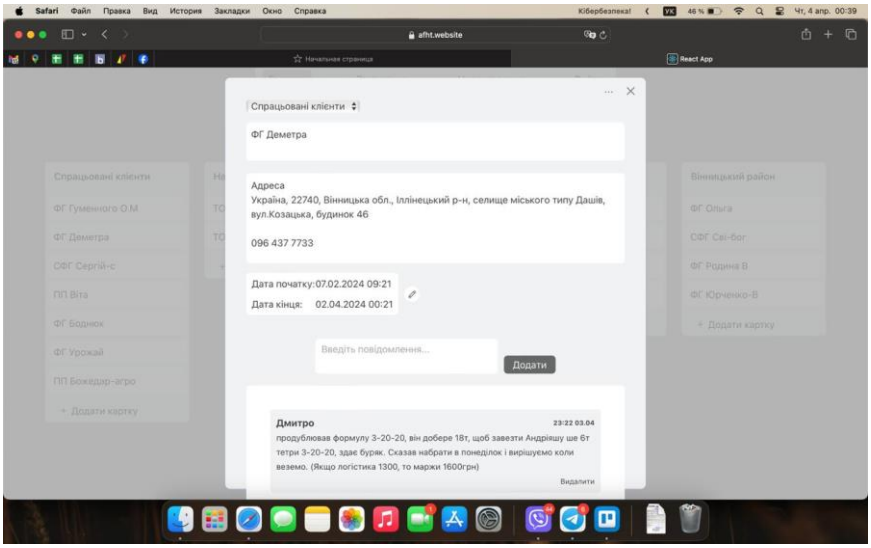

Рисунок 4.9 – Карточка клієнта

Також у нас є повідомлення, тобто після кожної розмови, скільки разів менеджер спілкується з клієнтом він записує коментар щодо цієї розмову для того щоб пам'ятати про що він говорив з ним попередній раз.

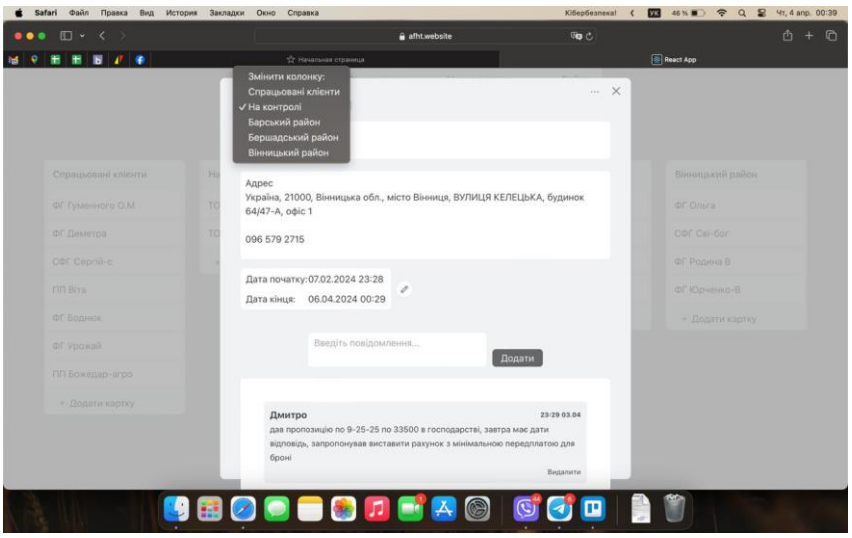

Рисунок 4.10 – Керування карточкою клієнта

Також функціонал карток дозволяє переміщати їх між колонками.

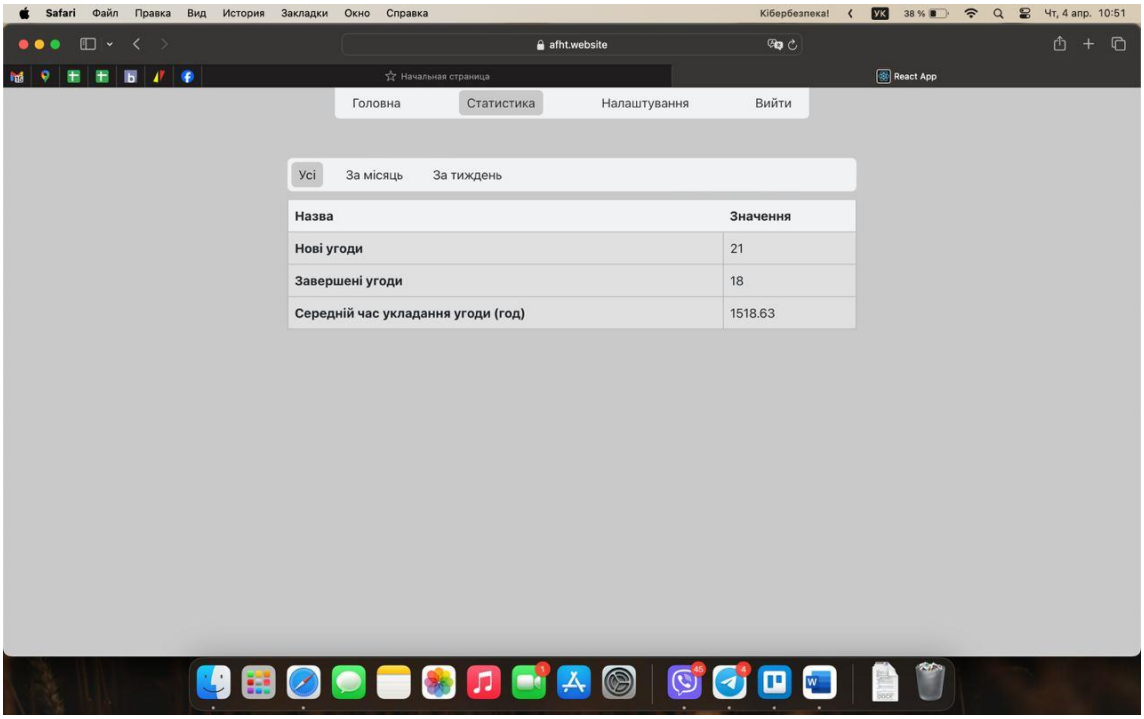

Рисунок 4.11 – Статистика бази даних стосовно угод

Тобто натискаємо, там де в нас “Спрацьовані клієнти” та можемо перенести цю карточку до іншої колонки, до якої бажаємо.

Також у базі даних присутня статистика, яка враховує три позиції: “Нові угоди”, “Завершені угоди” та “Середній час укладання угод”, тобто нові угоди, це нові карточки, які були додані до бази даних. Завершені угоди, це ті угоди, які враховуються системою як завершені відповідно до показників функції “Календар”. “Середній час укладання угоди” - це проміжок часу з початку укладання угоди до її завершення, місяць та тиждень.

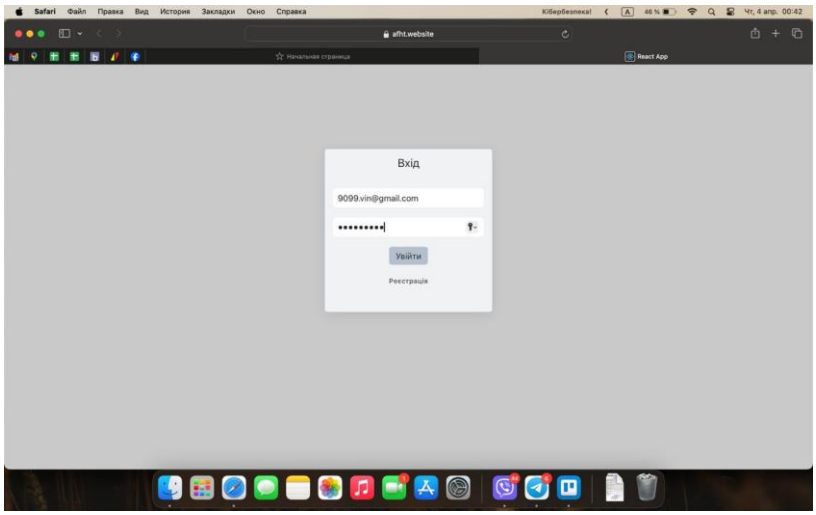

Рисунок 4.12 – Вхід до системи наступного дня

Після того як менеджер закінчив свій робочий день він вийшов з системи, минула ціла доба, наступного робочого дня він повертається до свого робочого місця та йому треба знову виконати вхід.

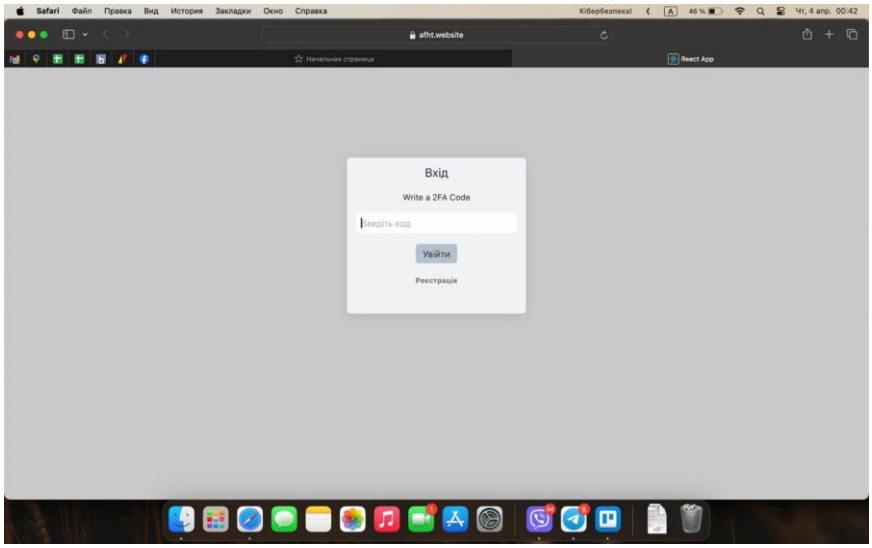

Рисунок 4.13 – Вхід до системи наступного дня за допомогою багатофакторної аутентифікації

Для цього він вводить свій логін і пароль, натискає “Увійти” його просить база даних підтвердити його вхід за допомогою багатофакторної аутентифікації.

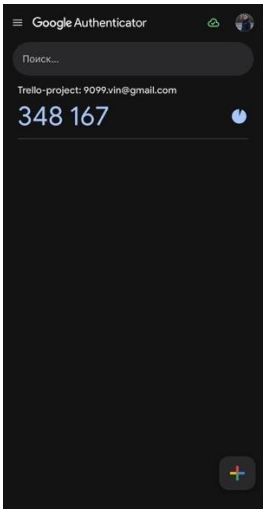

Рисунок 4.14 – Вхід до системи за допомогою кода на телефоні

У цей момент він розблоковує свій телефон та дивиться який код в нього там є на цю секунду. Це 348 167. Він вводить даний код в браузері в відповідне поле.

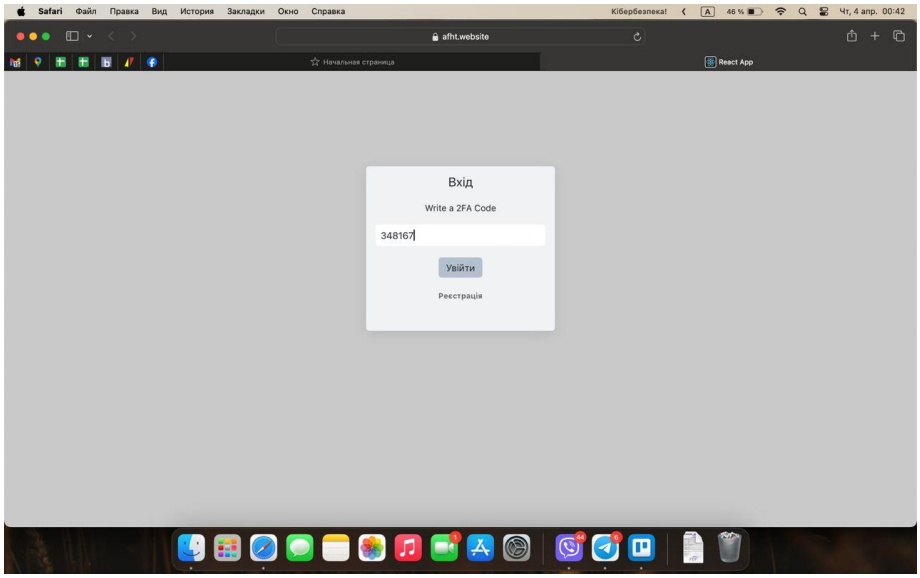

Рисунок 4.15. Вхід до системи за допомогою кода у браузері

Якщо код правильний, йому залишається натиснути “Увійти” і система надає йому доступ до бази даних.

4.3 Розгляд перспектив та викликів використання багатофакторної автентифікації у вебзастосунках

Багатофакторна автентифікація вебзастосунків відкриває нові перспективи для забезпечення безпеки користувачів. Цей метод, який використовує два або більше методи підтвердження ідентичності, дозволяє зменшити ризики крадіжки або злому аккаунтів. Використання різних факторів, таких як паролі, біометричні дані, або фізичні пристрої, створює додаткові перешкоди для потенційних зловмисників.

Однак впровадження багатофакторної автентифікації може стати складним завданням через його вартість та технічні виклики. Інтеграція з існуючими системами, забезпечення приватності та безпеки даних, а також навчання користувачів використовувати нові методи, все це вимагає уважного розгляду.

Отже, хоча багатофакторна автентифікація має значний потенціал для підвищення безпеки вебзастосунків, вона також вимагає уважного управління, щоб забезпечити оптимальний баланс між безпекою та зручністю користувачів.

Перспективи використання багатофакторної автентифікації у вебзастосунках включають:

1. Підвищена безпека: Багатофакторна автентифікація забезпечує більш високий рівень безпеки, оскільки навіть якщо один фактор автентифікації стає компромісним, інші фактори все ще залишаються на захисті.

2. Захист від фішингу та крадіжки ідентифікаторів: Оскільки для вторгнення необхідно мати доступ до кількох факторів, багатофакторна автентифікація зменшує ймовірність успішних атак фішингу та крадіжки ідентифікаторів.

3. Відповідність з вимогами безпеки: В ряді галузей, таких як фінансові послуги або медична сфера, використання багатофакторної автентифікації може бути обов'язковим для дотримання вимог щодо безпеки даних.

4. Користувацький комфорт: З вдосконаленням технологій багатофакторної автентифікації, користувачі можуть користуватися різними методами автентифікації, що їм зручні, наприклад, сенсорними сканерами відбитків пальців або розпізнаванням обличчя.

5. Зменшення витрат на обробку відновлення паролів: За допомогою багатофакторної автентифікації можна зменшити кількість випадків втрати або забування паролів, що може зменшити витрати на обслуговування користувачів.

Однак існують також виклики, пов'язані з використанням багатофакторної автентифікації:

1. Комплексність для користувачів: Деякі методи багатофакторної автентифікації можуть бути складними для користувачів або вимагати додаткових зусиль для реалізації.

2. Можливість блокування доступу: Якщо користувач втратить доступ до всіх своїх факторів автентифікації (наприклад, телефон), він може бути заблокований з вебзастосунку.

3. Вартість реалізації та підтримки: Впровадження та підтримка систем багатофакторної автентифікації може бути витратною для організацій.

4. Проблеми інтеграції з існуючими системами: Іноді існуючі системи не підтримують багатофакторну автентифікацію або її інтеграція може бути складною.

5. Приватність та безпека даних: Збір та обробка додаткових особистих даних для багатофакторної автентифікації може викликати проблеми з приватністю та безпекою даних.

Усі ці фактори потребують уважного розгляду при впровадженні багатофакторної автентифікації у вебзастосунках для забезпечення оптимального балансу між безпекою та зручністю для користувачів.

Висновки до четвертого розділу

Багатофакторна автентифікація вебзастосунків відкриває нові перспективи для забезпечення безпеки користувачів. Цей метод, який використовує два або більше методи підтвердження ідентичності, дозволяє зменшити ризики крадіжки або злому аккаунтів. Використання різних факторів, таких як паролі, біометричні дані, або фізичні пристрої, створює додаткові перешкоди для потенційних зловмисників.

Однак впровадження багатофакторної автентифікації може стати складним

завданням через його вартість та технічні виклики. Інтеграція з існуючими системами, забезпечення приватності та безпеки даних, а також навчання користувачів використовувати нові методи, все це вимагає уважного розгляду.

Отже, хоча багатофакторна автентифікація має значний потенціал для підвищення безпеки вебзастосунків, вона також вимагає уважного управління, щоб забезпечити оптимальний баланс між безпекою та зручністю користувачів.

ВИСНОВКИ

Сучасною реальністю є те, що процедура автентифікації є необхідною для забезпечення цілісності, конфіденційності та доступності інформації користувачів у різних системах та ресурсах. Використання автентифікації поширене в усіх сферах для контролю доступу, як до фізичних об'єктів, так і до інформації в онлайн-застосунках.

Під час виконання завдань дослідження було розглянуто основні аспекти автентифікації, її різновиди та області використання. Аналізуються проблеми та вразливості однофакторної автентифікації для вебзастосунків, і виокремлено переваги багатофакторної автентифікації.

У звіті з практики проведено аналіз існуючих популярних рішень, визначено мінімальні вимоги, які повинен включати відповідний сервіс. Виявлено, що більшість існуючих рішень спрямовані на спеціалізовані ринки, є складними та вартісними для користувача. Багато з них, хоча й ефективні в аутентифікації, включають комплексні рішення з безпеки для вебзастосунків, що може бути зайвим для деяких користувачів.

Висновок полягає в тому, що за результатами проведено дослідження було проаналізовано основні існуючи механізми багатофакторної автентифікації та виявлено критичну точку в роботі компанії: а саме використання застосунку Trello менеджерами компанії, що впливає на безпеку СУБД підприємства. На основі проведеного аналізу було вирішено, що найбільш ефективним автентифікатором, який підходить для вирішення практичного завдання диплому та забезпечить основні функції аналогів, але при цьому більш простим і лишений надлишкового функціоналу, є "Google Authenticator", який вирішено використовувати в дипломному завданні.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Концепція створення національної системи ідентифікації громадян України, іноземців та осіб без громадянства : Розпорядження Кабінету Міністрів України № 1428-2015-р від 23.12.2015 [Електронний ресурс]. – Режим доступу : https://zakon.rada.gov.ua/laws/main/1428-2015-%D1%80

2. Бідюк П. Сучасні методи біометричної ідентифікації / П.Бідюк, В.Бондарчук // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. – 2009. – Вип. 1 (18). – С. 137–146

3. Горбенко І.Д. Інфраструктури відкритих ключів. Електронний цифровий підпис. Теорія та практика: монографія./ І.Д. Горбенко, Ю.І. Горбенко – Харків: «Форт», 2010. – 608 с.

4. Горбенко І.Д. Прикладна криптологія. Теорія. Практика. Застосування: монографія./ І.Д. Горбенко, Ю.І. Горбенко – Харків: «Форт», 2012. – 880 с.

5. Гуменюк І.В. Методика забезпечення захисту інформації в інформаційно-телекомунікаційних системах / І.В. Гуменюк, В.Л. Барилюк, М.В. Файдюк // Забезпечення інформаційної безпеки держави у воєнній сфері: проблеми та шляхи їх вирішення : мат-ли науково-практичної конференції. – Київ : НУОУ ім. І.Черняховського, 2019. – С. 66–67

6. Гуменюк І.В. Біометрична ідентифікація у кіберпросторі на основі розпізнавання обличчя / І.В. Гуменюк, М.С. Басараба, О.В. Некрилов // Проблеми теорії та практики інформаційного протиборства в умовах ведення гібридних війн : тези доповідей наук.-практ. конф. 24–25 жовтня 2019 р. – Житомир : ЖВІ, 2019. – С. 205–207

7. Кавуненко Я. О. Оптимізація застосування комплексного методу захисту інформації від витоку по каналам побічних електромагнітних випромінювань / Я. О. Кавуненко. // Харків, ХНУРЕ, Матеріали XXII міжнародного молодіжного форуму «Радіоелектроніка та молодь у XXI столітті». Том 4. – 2018. – С. 124 – 125

8. Кавуненко Я. О. Аналіз криптографічних систем і перспектива

використання протоколів у групах КОС / Я. О. Кавуненко. // Харків, ХНУРЕ, Матеріали XXIV міжнародного молодіжного форуму «Радіоелектроніка та молодь у XXI столітті». – у друці (рік виходу – 2020)

9. Коваль Л.Г. Методи і технології біометричної ідентифікації за результатами літературних джерел / Л.Г. Коваль, С.М. Злепко, Г.М. Новіцький // Вчені записки ТНУ ім. В.І. Вернадського. Серія : Технічні науки. – 2019. – Т. 30 (69), Ч. 1, № 2. – С. 104–112

10. Кумченко Ю.О. Інформаційна технологія ідентифікації персоналу на основі комплексу біометричних параметрів : дис. на здобуття наукового ступеня канд. техн. наук / Ю.О. Кумченко. – 2017. – 143 с

11. Миронов Ю. Б. Сильні та слабкі сторони геолокації та навігації [Електронний ресурс] / Ю. Б. Миронов // Науковий вісник Херсонського державного університету. Серія «Технічні науки». – 2014. – Випуск 6, частина 5. – С.26-30. – Режим доступу : http://www.ej.kherson.ua/journal/a _06/260.pdf

12. Нємкова О.А. Біометрична ідентифікація у кіберпросторі / О.А. Нємкова // Системи обробки інформації. – 2015. – Вип. 7 (132). – С. 118–121

13. Нечипоренко О.В. Біометрична ідентифікація і автентифікація особи за геометрією обличчя / О.В. Нечипоренко, Я.В. Корпань // Вісник Хмельницького національного університету. Серія : Технічні науки. – 2016. – № 4. – С. 133–138

14. Ніколаєв С.С. Залежність якості детектора облич на ознаках Хаара від варіативності навчальної вибірки / С.С. Ніколаєв, Ю.О. Тимошенко, К.Ю. Матвіїв // Наукові вісті НТУУ «КПІ» : міжнародний науковотехнічний журнал. – 2017. – № 6 (116). – С. 38–46.

15. Подоляка Н. В. Комплексний підхід при виборі методології оцінки ризиків серверної кімнати як об'єкту підвищеного ризику компанії / 88 Н. В. Подоляка, Я. О. Кавуненко. // Харків, ХНУРЕ, Матеріали XXIV міжнародного молодіжного форуму «Радіоелектроніка та молодь у XXI столітті». – у друці (рік виходу – 2020).

16. Поповський В. В. Основи криптографічного захисту інформації в телекомунікаційних системах. Навчальний посібник. Частина 1 / В. В. Поповський, А. В. Персіков. – Харків: СМИТ, 2010. – 352 с.

17. Принцип дії GPS [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/GPS

18. Радіолокація [Електронний ресурс] / Держ. служба статистики України. – К. : Август Трейд, 2013. – Режим доступу : http://library.oseu.edu.ua/files/StatSchorichnyk Ukrainy 2012.pdf

19. Різник О., Дзюба Д., Чернодуб А. «Біокон» – система біометричної ідентифікації користувача комп'ютерної мережі. Системи підтримки прийняття рішень. Теорія і практика: зб. доп. наук.-прак. конф. з міжнар. участю. «СППР 2009». Київ, 2009. С. 189 – 193.

20. Свидрук І. І. Дослідження методів геолокації: автореф. дис. на здобуття наук. ступеня канд. екон. наук : 08.00.04 «Методи геолокації» / Ірена Ігорівна Свидрук. - Львів, 2007. - 20 с

21. Спеціалізоване програмне забезпечення біометричної ідентифікації/аутентифікації користувачів інформаційно-телекомунікаційних систем на основі геометрії обличчя : заявка про реєстрацію авторського права на твір № АПС 95-19 ; дата затвердження заявки 28.10.2019

22. Царьов Р.Ю. Біометричні технології: навч. посіб. [для вищих навчальних Ц18 закладів] / Р.Ю. Царьов, Т. М. Лемеха. – Одеса: ОНАЗ ім. О.С. Попова, 2016. – 140 с.: іл.

23. Daugman J.G. High Confidence Visual Recognition of Persons by a Test ou Statistical Independence. IEEE Transactions On Pattern Analysis and Machine Intelligence. Vol. 15. No.11. pp.1148-1161. 1993.

24. Jain A. Introduction to Biometrics [Text] /A. Jain, A. Ross.// Handbook of Biometrics. Springer. – 2008. – p. 1–22.

25. International Organization for Standardization [Електронний ресурс]. – 2012. – Режим доступу до ресурсу: http://www.iso.org

26. ISO/IEC 19785-1:2015 Information technology — Common Biometric Exchange Formats Framework — Part 1: Data element specification [Електронний ресурс]. – 2015. – Режим доступу до ресурсу: https://www.iso.org/ru/standard/66179.html

27. R. Wildes, J.C. Asmuth, G.L. Green, S.C. Hsu, R.J. Kolczynski, J.R. Matey and S.E. McBride. A system for automated iris recognition. In Proceedings ou the IEEE Workshop on Applications of Computer Vision. pp. 121-128. 1994.

28. Authentication using the Google APIs Client [Електронний ресурс] – Режим доступу до ресурсу: https://developers.google.com/api-clientlibrary/javascript/start/start-js.

29. Dominic, Ehiwe & Kayode, Akinola & Ominike, Akpovi. (2018). Social Network Application User Authentication: 2FA with Encrypted Image. American Journal of Computing and Engineering Vol.3, Issue 1 No.1, pp 1 – 10

30. Features [Електронний ресурс] // authy. – 2019. – Режим доступу до ресурсу: https://authy.com/features/.

31. FreeOTP [Електронний ресурс] – Режим доступу до ресурсу: https://freeotp.github.io/.

32. Getting Started [Електронний ресурс] – Режим доступу до ресурсу: https://developers.google.com/api-client-library/javascript/start/start-js.

33. Getting Started [Електронний ресурс] – Режим доступу до ресурсу: https://developers.google.com/api-client-library/javascript/start/start-js.

34. HOTP: An HMAC-Based One-Time Password Algorithm [Електронний ресурс] // Network Working Group. – 2005. – Режим доступу до ресурсу: https://tools.ietf.org/html/rfc4226.

35. OATH Certification [Електронний ресурс] // OATH Authentication. – 2019. – Режим доступу до ресурсу: https://openauthentication.org/oathcertification/.

36. OCRA: OATH Challenge-Response Algorithm [Електронний ресурс] // Internet Engineering Task Force. – 2011. – Режим доступу до ресурсу: https://tools.ietf.org/html/rfc6287.

37. Open source version of Google Authenticator [Електронний ресурс] – Режим доступу до ресурсу: https://github.com/google/google-authenticator

38. Understanding GPS Principles and Applications [Електронний ресурс] – Режим доступу до ресурсу: http://www.worldcat.org/title/understanding-gps-principles-and-applications-principles-andapplications/oclc/437160311

39. Ivan Parkhomenko, Dmytro Zhebrak MULTI-FACTOR AUTHENTICATION IN WEB APPLICATIONS X Міжнародна науково-практична конференція (Information technology and implementations) – IT&I-2023 – pp. 160–161.