Вінницький національний технічний університет

Факультет менеджменту та інформаційної безпеки

Кафедра менеджменту та безпеки інформаційних систем

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Захищений консолідований інформаційний ресурс системного аналізу безпеки енергетичної інфраструктури регіону»

Виконав: ст. 2-го курсу, групи 1КІТС-22м
спеціальності 125– Кібербезпека
Освітня програма – Кібербезпека
інформаційних технологій та систем
(шифр і назва напряму підготовки, спеціальності)

Гуменюк В.В.
(прізвище та ініціали)

Керівник: д.т.н., проф. каф МБІС
Яремчук Ю.Є.
(прізвище та ініціали)
«04» грудня 2023 р.

Опонент: к.т.н., доц., доцент каф. ОТ
Кожем'яко А. В.
(прізвище та ініціали)
«04» грудня 2023 р.

Допущено до захисту
Голова секції УБ кафедри МБІС

Юрій ЯРЕМЧУК
"04" грудня 2023 р.

Вінниця ВНТУ - 2023 рік

Кафедра менеджменту та безпеки інформаційних систем

Рівень вищої освіти ІІ-й (магістерський)
Галузь знань 12 – Інформаційні технології
Спеціальність 125 – Кібербезпека
Освітньо-професійна програма - Кібербезпека інформаційних технологій та систем

ЗАТВЕРДЖУЮ
Голова секції УБ кафедри МБІС

_____**Юрій ЯРЕМЧУК**
"20" березня 2023 р.

З А В Д А Н Н Я
на магістерську кваліфікаційну роботу студенту
_____ Гуменюк В`ячеслав Володимирович_____
(прізвище, ім`я, по-батькові)

1. Тема роботи:

«Захищений консолідований інформаційний ресурс системного аналізу безпеки енергетичної інфраструктури регіону»
Керівник роботи_:_____д.т.н., проф. каф. МБІС Яремчук Ю.Є._____
(прізвище, ім`я, по-батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу від "18" вересня 2023 року № 247
2. Строк подання студентом роботи_за тиждень до захисту.__
3. Вихідні дані до роботи:
Стандарти, електронні джерела, підручники та наукові статті по темі, які стосуються теми магістерської кваліфікаційної роботи.
4. Зміст текстової частини:
Для досягнення мети роботи було поставлено наступні задачі: дослідити потреби організацій та звичайних людей у засобах захисту інформації; проаналізувати поширені способи захисту інформації та визначити їх недоліки; У першому розділі було проведено аналіз критичних інфраструктур та методів аналізу безпеки; У другому розділі було здійснено розробку бази даних та її нормалізаці, також було здійснено розробку методу захисту інформаційного ресурсу; В третьому розділі було здійснено реалізацію бази даних, модулю захисту та системи аналізу безпеки інфраструктури

5. Перелік ілюстративного матеріалу (з точним зазначенням обов`язкових креслень)
У дугому розділі магістерської кваліфікаційної роботи наведено 3 рисунків
третьому розділі – 10 рисунків

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Основна частина			
I	Яремчук Ю.Є. д.т.н., проф. каф. МБІС		
II	Яремчук Ю.Є. д.т.н., проф. каф. МБІС		
III	Яремчук Ю.Є. д.т.н., проф. каф. МБІС		
Економічна частина			
IV	Причепа І.В., к.е.н., доц. каф. ЕПВМ		

7. Дата видачі завдання 20 вересня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи		Примітка
1.	Визначення напрямку магістерської роботи, формулювання теми	20.09.2023	31.09.2023	
2.	Аналіз предметної області обраної теми	01.10.2023	15.10.2023	
3.	Розробка роботи	16.10.2023	26.10.2023	
4.	Написання магістерської роботи на основі розробленої теми	27.10.2023	15.11.2023	
5.	Передзахист магістерської кваліфікаційної роботи	16.11.2023	24.11.2023	
6.	Виправлення, уточнення, корегування магістерської кваліфікаційної роботи	27.11.2023	04.12.2023	
7.	Захист магістерської кваліфікаційної роботи	11.12.2023	17.12.2023	

Студент (підпис) Гуменюк В.В.

Керівник роботи (підпис) Яремчук Ю.Є.

АНОТАЦІЯ

УДК 004.56.5(043.2)

Гуменюк В.В, Розробка захищеного консолідованого інформаційного ресурсу системного аналізу безпеки енергетичної інфраструктури регіону. Магістерська кваліфікаційна робота зі спеціальності 125 – «Кібербезпека», освітня програма «Кібербезпека інформаційних технологій та систем». Вінниця: ВНТУ, 2023. 137 с.

На укр.мові. Бібліогр.: 45 назв; рис.: 26; табл. 10.

У магістерській кваліфікаційній роботі здійснено розробку захищеного консолідованого інформаційного ресурсу системного аналізу безпеки енергетичної інфраструктури регіону.

В першому розділі роботи здійснено аналіз теоретичного матеріалу обраної галузі: досліджено актуальність, особливості та забезпечення безпеки об’єктів критичної інфраструктури, проведено аналіз методів системного аналізу безпеки об’єктів та аналіз сучасних методів автентифікації користувачів.

У другому розділі роботи описано особливості розробки інформаційного ресурсу аналізу безпеки енергетичної інфраструктури та забезпечення захисту розробленого ресурсу від несанкціонованого доступу. Проведена розробка бази даних консолідованого інформаційного ресурсу та проведено нормалізацію відношень бази даних.

У третьому розділі роботи здійснено практичну реалізацію бази даних консолідованого інформаційного ресурсу, також реалізовано систему аналізу безпеки енергетичної інфраструктури. Розроблено програмний модуль забезпечення захисту, та досліджено результати роботи.

У четвертому розділі роботи проаналізовано економічну доцільність розробки, продемонстровано її високий комерційний потенціал та можливість подальшого впровадження.

Ключові слова: енергетична інфраструктура, критична інфраструктура, системний аналіз, консолідація, безпека, інформаційний ресурс.

ABSTRACT

Humenyuk V.V. Development of a Secure Consolidated Information Resource for System Analysis of Energy Infrastructure Security in the Region. Master's Thesis in Cybersecurity, Educational Program "Cybersecurity of Information Technologies and Systems." Vinnitsa: VNTU, 2023. 137 p.

In Ukrainian language. Bibliographer: 45 titles; figures: 26; tables: 10.

In this master's thesis, the development of a secure consolidated information resource for system analysis of energy infrastructure security in the region is presented.

The first chapter provides an analysis of the theoretical material in the chosen field, exploring the relevance, features, and security aspects of critical infrastructure objects. The chapter also includes an analysis of methods for the system analysis of object security and an examination of modern user authentication methods.

The second chapter describes the peculiarities of developing an information resource for the security analysis of energy infrastructure and ensuring protection against unauthorized access. The development of the database of the consolidated information resource is outlined, including the normalization of database relations.

The third chapter focuses on the practical implementation of the database of the consolidated information resource and the system for the security analysis of energy infrastructure. It includes the development of a security software module and an exploration of the results of its operation.

The fourth chapter analyzes the economic feasibility of the development, demonstrating its high commercial potential and possibilities for further implementation.

Keywords: environmental infrastructure, critical infrastructure, system analysis, consolidation, security, information resource.

ЗМІСТ

ВСТУП 9

1 ТЕОРЕТИЧНІ ЗАСАДИ СТВОРЕННЯ КОНСОЛІДОВАНОГО ІНФОРМАЦІЙНОГО РЕСУРСУ АНАЛІЗУ БЕЗПЕКИ ЕНЕРГЕТИЧНОЇ ІНФРАСТРУКТУРИ 12

1.1 Актуальність, особливості та основні принципи забезпечення безпеки об'єктів критичної інфраструктури 12

1.2 Загальна характеристика сфери діяльності "енергетична інфраструктура", як частина критичної інфраструктури регіону 17

1.3 Консолідація інформації для забезпечення безпеки енергетичної інфраструктури 22

1.4 Аналіз методів системного аналізу безпеки об’єктів 26

1.5 Аналіз сучасних методів автентифікації користувачів до інформаційного ресурсу 32

1.6 Висновки та постановка задачі 44

2 РОЗРОБКА ЗАХИЩЕНОГО КОНСОЛІДОВАНОГО ІНФОРМАЦІЙНОГО РЕСУРСУ 46

2.1 Особливості розробки інформаційного ресурсу аналізу безпеки енергетичної інфраструктури 46

2.2 Розробка бази даних консолідованого інформаційного ресурсу 48

2.3 Проектування ER-моделі бази даних 52

2.4 Нормалізації відношень бази даних 53

2.5 Проектування звітів. 57

2.6 Забезпечення захисту розробленого ресурсу від несанкціонованого доступу. 60

2.7 Висновки до розділу. 65

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАХИЩЕНОГО КОНСОЛІДОВАНОГО ІНФОРМАЦІЙНОГО РЕСУРСУ ТА СИСТЕМНИЙ АНАЛІЗ БЕЗПЕКИ ФІНАНСОВОЇ ІНФРАСТРУКТУРИ 67

3.1 Обґрунтування вибору СУБД 67

3.2 Обґрунтування вибору мови програмування 71

3.3 Реалізація бази даних інформаційного ресурсу 75

3.4 Реалізація запитів та звітів 78

3.5 Реалізація програмних модулів забезпечення захисту інформаційного ресурсу 82

3.6 Реалізація системи аналізу безпеки енергетичної інфраструктури регіону 86

3.7 Висновки до розділу 92

4 ЕКОНОМІЧНА ЧАСТИНА 93

4.1 Оцінювання комерційного потенціалу розробки програмного забезпечення 93

4.2 Прогнозування витрат на виконання наукової роботи та впровадження її результатів 97

4.3 Прогнозування комерційних ефектів від реалізації результатів розробки 106

4.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності 108

4.5 Висновки до розділу 112

ВИСНОВКИ 114

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 116

ДОДАТКИ 121

Додаток А. Технічне завдання 122

Додаток Б. Лістинг програми 125

Додаток В. Ілюстративний матеріал 136

Додаток Г. Протокол перевірки на антиплагіат 145

ВСТУП

Актуальність. Енергетична інфраструктура регіону є критично важливою для його нормального функціонування. Вона містить мережі електропостачання, газопостачання, теплопостачання, а також інші об'єкти, які забезпечують безперебійне постачання енергії.

Однак енергетична інфраструктура є вразливою до атак. Зловмисники можуть атакувати енергетичну інфраструктуру з різних причин, наприклад, для отримання економічної вигоди, політичної мети або просто для завдавання шкоди.

Атаки на енергетичну інфраструктуру можуть мати серйозні наслідки. Вони можуть призвести до відключення електроенергії, газопостачання або теплопостачання, що може спричинити економічні втрати, порушення життєдіяльності населення та інші негативні наслідки.

У 2022 році в Україні відбулася низка кібератак на енергетичну інфраструктуру. Ці атаки призвели до порушення роботи енергомереж, що спричинило перебої в постачанні електроенергії.

У 2023 році у США було повідомлено про серію фізичних атак на енергетичні об'єкти. Ці атаки призвели до пожеж на електростанціях та інших пошкоджень.

Ці приклади свідчать про те, що енергетична інфраструктура є вразливою до атак. Розробка захищеного консолідованого інформаційного ресурсу системного аналізу безпеки енергетичної інфраструктури регіону дозволить підвищити рівень безпеки енергетичної системи та захистити її від атак.

Для підвищення безпеки енергетичної інфраструктури регіону необхідно розробити ефективні методи її захисту. Одним із таких методів є створення захищеного консолідованого інформаційного ресурсу системного аналізу безпеки енергетичної інфраструктури регіону.

Розробка захищеного консолідованого інформаційного ресурсу системного аналізу безпеки енергетичної інфраструктури регіону є важливою для підвищення

безпеки енергетичної інфраструктури. Проєкт має низку переваг, які дозволять покращити видимість стану енергетичної інфраструктури, підвищити ефективність управління безпекою та створити єдиний центр управління безпекою енергетичної інфраструктури регіону.

Мета і задачі дослідження. Метою роботи є розробка та впровадження захищеного консолідованого інформаційного ресурсу, який має сприяти системному аналізу та ефективному забезпеченню безпеки енергетичної інфраструктури регіону.

Задачами дослідження є:

– аналіз сучасних методів системного аналізу безпеки та аналіз сучасних методів забезпечення захисту інформаційного ресурсу;

– розробка бази даних для консолідованого інформаційного ресурсу, нормалізація відношень бази даних та забезпечення захисту створеного консолідованого інформаційного ресурсу;

– проєктування та розробка інтерфейсу користувача та реалізація програмного засобу у вигляді веб-сторінки;

– тестування розробки та аналіз отриманих результатів;

– економічне обґрунтування доцільності впровадження здійсненої розробки.

Об’єкт дослідження – енергетична інфраструктура регіону, включаючи електростанції, підстанції та енергетичні мережі, що визначають життєво важливу інфраструктуру сучасного суспільства.

Предмет дослідження – розроблений консолідований інформаційний ресурс системного аналізу безпеки енергетичної інфраструктури.

Новизна роботи: вперше розроблено захищений консолідований інформаційний ресурс аналізу безпеки енергетичної інфраструктури регіону, впроваджено інноваційний підхід до захисту енергетичних систем від кіберзагроз, використання сучасних методів системного аналізу та розробка новаторських інформаційних рішень для ефективного забезпечення безпеки.

Практична цінність: розроблено інформаційний ресурс який має велике значення для оперативного аналізу та моніторингу безпеки енергетичної інфраструктури регіону, сприяючи підвищенню рівня стійкості та надійності.

За тематикою роботи опубліковано 3 публікації, зокрема 1 статтю у фаховому виданні та 2 тези доповідей на науковій конференції [41 – 43].

1 ТЕОРЕТИЧНІ ЗАСАДИ СТВОРЕННЯ КОНСОЛІДОВАНОГО ІНФОРМАЦІЙНОГО РЕСУРСУ АНАЛІЗУ БЕЗПЕКИ ЕНЕРГЕТИЧНОЇ ІНФРАСТРУКТУРИ

У першому розділі магістерської кваліфікаційної роботи розглядається аналіз забезпечення безпеки об'єктів критичної інфраструктури, а також аналіз енергетичної інфраструктури, розглянутої як складову критичної інфраструктури, а також аналіз процесу консолідації інформації.

На основі цього аналізу сформовано висновки та здійснено постановку задач для подальших досліджень та розробки.

1.1 Актуальність, особливості та основні принципи забезпечення безпеки об'єктів критичної інфраструктури

Сучасне суспільство великою мірою покладається на критичну інфраструктуру для забезпечення свого функціонування та надання важливих послуг у різних сферах, включаючи електроенергетику, транспорт, комунікації, фінанси, та багато інших. Однак зростаючий рівень технологічного розвитку та загрози, пов'язані з кібератаками й тероризмом, роблять об'єкти критичної інфраструктури особливо уразливими. Забезпечення безпеки цих об'єктів стає надзвичайно важливим завданням для забезпечення стійкості та безпеки суспільства.

Критична інфраструктура (КІ) - це комплексні об'єкти та системи, які є незамінними для нормального функціонування сучасного суспільства та забезпечують надання важливих послуг і функцій у різних сферах життя. Ця інфраструктура містить різноманітні галузі та об'єкти, і від її надійного функціонування залежить безперебійне надання послуг та забезпечення стабільності суспільства [1].

Критична інфраструктура може бути розділена на кілька ключових категорій (рис. 1.1).

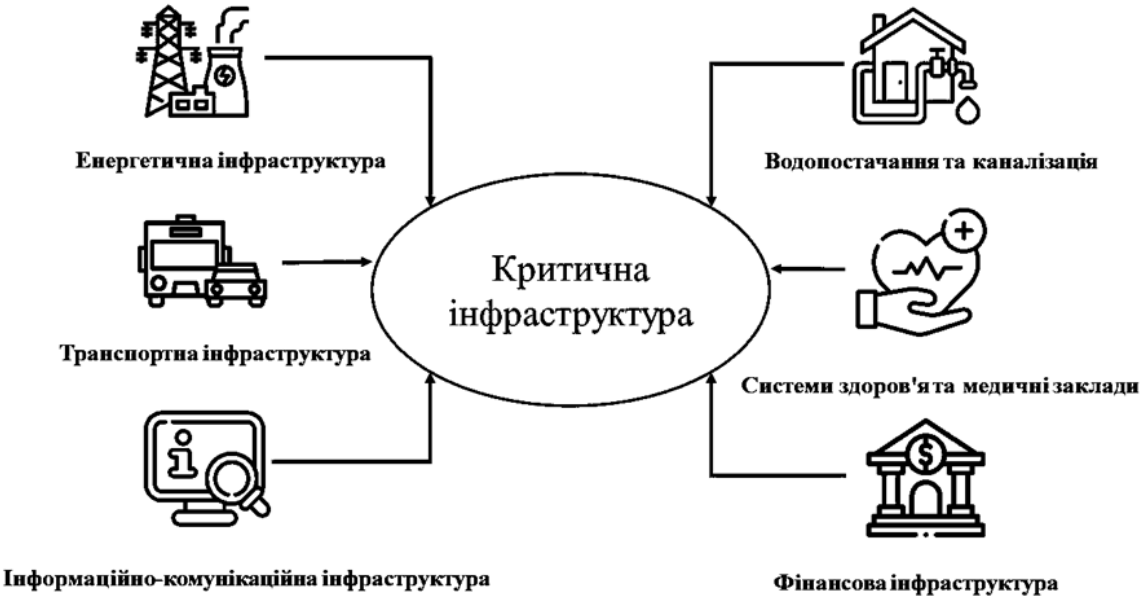


Рисунок 1.1 – Ключові категорії критичної інфраструктури

Детальніше розглянемо ці категорії:

– енергетична інфраструктура. Сюди входять електростанції, підстанції, газопроводи, нафтопроводи та інші об'єкти, які забезпечують постачання енергії для освітлення, опалення та роботи промислових підприємств;

– транспортна інфраструктура. Ця категорія включає дороги, мости, залізничні мережі, аеропорти та порти, які забезпечують транспортування людей і товарів;

– інформаційно-комунікаційна інфраструктура. Системи зв'язку, інтернет, телекомунікаційні мережі та інші технології, які забезпечують обмін інформацією та комунікацію;

– водопостачання та каналізація. Це включає системи постачання питної води та водовідведення, які є життєво важливими для забезпечення гігієнічних стандартів та здоров'я населення;

– системи здоров'я та медичні заклади. Медичні установи, лікарні, лікарські препарати та обладнання, які забезпечують медичну допомогу населенню;

– фінансова інфраструктура. Банки, фінансові установи та платіжні системи, які забезпечують функціонування економічного сектору.

Ці категорії об'єктів критичної інфраструктури мають вирішальне значення для стабільності та функціонування сучасного суспільства, і будь-які відмови чи загрози їх нормальному функціонуванню можуть мати серйозні наслідки для громадської безпеки, економічної стабільності та якості життя населення [2].

Критична інфраструктура є ключовим компонентом сучасного суспільства, оскільки вона забезпечує функціонування важливих систем і послуг, таких як електроенергетика, транспорт, комунікації, фінанси та інші. Внаслідок технологічного розвитку і зростаючої загрози кібератак і терористичних актів, забезпечення безпеки об'єктів критичної інфраструктури набуває особливого значення. Недоліки в цій галузі можуть призвести до серйозних наслідків, включаючи припинення надання важливих послуг, екологічні катастрофи та втрату життів.

Розглядаючи критичну інфраструктуру, важливо враховувати, що вона має свої особливості, які роблять її унікальною і вимагають специфічних підходів до забезпечення її безпеки. Для забезпечення безпеки КІ необхідно враховувати ряд особливостей:

– висока важливість;

Важливість КІ відображається в тому, що від неї залежить доступ до основних ресурсів, таких як енергія, вода, транспорт, та інші. Якщо КІ виявиться недоступною або неробочою, це може призвести до серйозних наслідків для суспільства.

– великі розміри та складність;

КІ може складатися з великої кількості різних об'єктів та систем, які можуть бути фізичними або інформаційними. Ця складність робить їх керування та забезпечення безпеки складним завданням.

– інтерконектованість;

Різні компоненти КІ часто взаємодіють між собою. Відмова в одній частині КІ може призвести до каскадного ефекту і відмови в інших галузях.

– вразливість до загроз;

КІ піддається різним загрозам, включаючи природні катастрофи, кібератаки, терористичні акти та інші небезпеки. Розробка стратегій та заходів для запобігання та реагування на ці загрози є критично важливою.

– інновації та технології.

Розвиток нових технологій та збільшення автоматизації в КІ може підвищити продуктивність і ефективність, але також може збільшити ризики в області кібербезпеки та інших аспектів безпеки [3].

Після детального розгляду особливостей критичної інфраструктури, деякі з яких описують її високу важливість, складність та вразливість до різних загроз, можна перейти до основних принципів забезпечення безпеки цих об'єктів. Розуміння особливостей КІ допомагає сформулювати та впровадити ефективні стратегії безпеки, враховуючи її унікальні вимоги та виклики. Розглянемо основні принципи забезпечення безпеки об'єктів критичної інфраструктури більш детально:

– принцип ідентифікації та класифікації загроз;

Даний принцип передбачає вивчення та класифікацію потенційних загроз, які можуть вплинути на об'єкти критичної інфраструктури. Цей аналіз повинен враховувати як природні, так і техногенні фактори, включаючи технічні відмови, кібератаки, природні катастрофи, терористичні акти тощо. Загрози повинні бути класифіковані залежно від рівня їх серйозності та потенційного впливу на інфраструктуру.

– принцип захисту та запобігання;

Цей принцип містить впровадження заходів та політик, спрямованих на зменшення ризику виникнення загроз та вразливостей. Вони можуть бути спрямовані як на фізичний захист, так і на захист інформації, мережі, а також забезпечення безперебійної роботи об'єктів КІ [4].

– принцип виявлення та реагування;

Даний принцип передбачає створення систем виявлення загроз та постійного моніторингу стану об'єктів критичної інфраструктури. Якщо загроза виявлена, повинні бути розроблені механізми та плани реагування для оперативного відновлення нормального функціонування та мінімізації збитків.

– принцип відновлення і відновлення;

Цей принцип включає розробку планів відновлення та відновлення інфраструктури після подій, що призвели до відмов чи пошкоджень. Плани відновлення мають забезпечити якнайшвидше та ефективне відновлення нормального функціонування для забезпечення безперебійності надання важливих послуг.

– принцип моніторингу та аналізу;

Даний принцип передбачає постійний моніторинг стану безпеки об'єктів критичної інфраструктури та аналіз інформації про можливі загрози. Аналіз допомагає виявляти тенденції, прогнозувати потенційні загрози та покращувати стратегії безпеки.

– принцип співпраці та координації.

Співпраця та координація між різними суб'єктами, включаючи державні органи, приватних операторів та громадські організації, є важливою для ефективного забезпечення безпеки КІ. Взаємодія та обмін інформацією допомагають покращити реагування на загрози та координувати дії у випадку надзвичайних ситуацій.

Ці принципи становлять основу стратегій забезпечення безпеки об'єктів критичної інфраструктури та спрямовані на забезпечення стійкості та надійності цих систем перед різними загрозами [5].

Актуальність забезпечення безпеки об'єктів критичної інфраструктури надзвичайно важлива в сучасному світі, оскільки вони відіграють вирішальну роль у забезпеченні стабільності суспільства та наданні важливих послуг.

Особливості цих об'єктів включають їх велику важливість, складність, вразливість до різних загроз, інтерконектованість та постійні зміни у сфері технологій.

Основні принципи забезпечення безпеки об'єктів критичної інфраструктури включають ідентифікацію та класифікацію загроз, захист та запобігання, виявлення та реагування, відновлення і відновлення, моніторинг та аналіз, а також співпрацю та координацію. Ці принципи спрямовані на створення ефективних стратегій забезпечення безпеки, які допомагають забезпечити стійкість і надійність критичної інфраструктури перед різними загрозами, і враховують її унікальні особливості та виклики.

1.2 Загальна характеристика сфери діяльності "енергетична інфраструктура", як частина критичної інфраструктури регіону

Сфера енергетичної інфраструктури є однією з найважливіших галузей, що забезпечують функціонування сучасного суспільства. Енергетична інфраструктура містить широкий спектр систем і об'єктів, які забезпечують виробництво, передачу, розподіл і використання різних видів енергії, що необхідні для економічного розвитку та підтримання життєдіяльності нації.

Ця сфера діяльності стала необхідною частиною сучасного способу життя, якій довіряють економічний розвиток, комфорт та безпеку громадян. Проте, з огляду на свою важливість, енергетична інфраструктура стала об'єктом підвищеної уваги з боку різних загроз та викликів, що включають природні катастрофи, кібератаки, терористичні акти та інші небезпеки.

У цьому контексті важливо проводити аналіз і вивчення енергетичної інфраструктури як частини критичної інфраструктури регіону, для розуміння її особливостей, визначення загроз та розробки стратегій забезпечення її безпеки та надійності.

Характеристика енергетичної інфраструктури включає такі аспекти:

– типи енергії;

Енергетична інфраструктура може включати виробництво електроенергії з різних джерел, включаючи ядерну, вугільну, газову, водну, вітряну та сонячну енергію, а також виробництво і розподіл природного газу, нафти та інших видів енергії.

– передача та розподіл;

Енергетична інфраструктура містить системи передачі та розподілу енергії, які транспортують електроенергію та інші види енергії від виробництва до споживачів через мережі та транспортні системи.

– електростанції;

Це включає електростанції різних типів, включаючи термальні, ядерні, гідроелектростанції та станції на відновлюваних джерелах енергії.

– транспорт і логістика;

Для перевезення нафти, газу та інших видів енергії потрібні транспортні системи, такі як нафтопроводи, газопроводи та транспортні вагони.

– інфраструктура та обладнання;

Енергетична інфраструктура включає в себе обладнання, таке як турбіни, трансформатори, газові компресори та інші технічні пристрої.

– керування та моніторинг;

Для ефективного керування інфраструктурою потрібні системи моніторингу, керування та автоматизації, які допомагають забезпечити безперебійну роботу та реагувати на відмови та аварії.

– забезпечення безпеки;

Енергетична інфраструктура піддається різним загрозам, включаючи природні катастрофи, кібератаки, терористичні акти та інші небезпеки. Забезпечення безпеки є критично важливою частиною функціонування енергетичної інфраструктури.

– споживачі.

Енергетична інфраструктура обслуговує широкий спектр споживачів, включаючи побутові, комерційні та промислові підприємства, що залежать від

надання послуг енергетичної інфраструктури для забезпечення нормальної діяльності.

Забезпечення безпеки та надійності енергетичної інфраструктури є надзвичайно важливою завданням для забезпечення сталого розвитку та життєдіяльності суспільства [6].

Об'єкти енергетичної інфраструктури, які відносять до критичної інфраструктури, включають різні компоненти, які є незамінними для забезпечення безпеки, сталості та функціонування суспільства. Ці об'єкти можуть відрізнятися в залежності від конкретного регіону та сфери діяльності. Відповідно до постанови Кабінету Міністрів України від 09.10.2020 № 1109, в якій визначено порядок віднесення об’єктів до об’єктів критичної енергетичної інфраструктури — до критичної енергетичної інфраструктури належать такі об’єкти:

- електроенергетика;
- нафтова промисловість;
- нафтова промисловість;
- ядерна енергетика.

Вони мають ключове значення для забезпечення енергетичних потреб суспільства та економіки та відзначаються своєю великою важливістю та специфічними особливостями. Розглянемо кожну з цих галузей докладніше та які послуги вони надають.

Електроенергетика:

- виробництво електричної енергії. Включає в себе всі види електростанцій, які виробляють електроенергію, від термальних та ядерних станцій до станцій на відновлюваних джерелах енергії;
- забезпечення функціонування ринку електричної енергії. Включає в себе всі аспекти організації ринку електроенергії, встановлення тарифів та контролю за торгівлею електроенергією;

– управління системами передачі та енергопостачання. Включає в себе керування системами передачі електроенергії, визначення навантаження та планування роботи мереж передачі;

– розподіл електричної енергії. Включає в себе мережі та підстанції, які розподіляють електроенергію до кінцевих споживачів.

Нафтова промисловість:

– видобуток нафти. Включає в себе видобуток нафти з нафтових свердловин та бурових платформ;

– передача (транзит) нафти та нафтопродуктів. Включає нафтопроводи, термінали та інфраструктуру для транспортування та транзиту нафти та нафтопродуктів;

– очищення, переробка та обробка нафти. Містить різні процеси обробки та переробки нафти, від очищення до виробництва нафтопродуктів;

– експлуатація нафтопроводів. Містить системи для транспортування та розподілу нафти та нафтопродуктів;

– зберігання та постачання нафти та нафтопродуктів. Містить резервуари для зберігання нафти та системи постачання нафтопродуктів до споживачів.

Газова промисловість:

– видобуток газу. Містить видобуток природного газу з газових свердловин;

– переробка та очищення газу. Містить обробку та очищення природного газу перед транспортуванням;

– передача (транзит) газу. Включає системи транспортування та транзиту природного газу через газопроводи;

– розподіл газу. Включає системи розподілу природного газу до кінцевих споживачів;

– експлуатація газотранспортної системи. Включає обслуговування та управління газотранспортними мережами;

– зберігання природного газу. Містить підземні сховища та резервуари для зберігання природного газу;

– забезпечення роботи систем зрідження природного газу; Включає системи зрідження газу для транспортування та зберігання.

Ядерна енергетика:

– виробництво електричної енергії на атомних станціях. Містить роботу атомних реакторів для генерації електроенергії.

– експлуатація атомних станцій. Включає обслуговування та управління атомними станціями.

– виробництво, переробка та зберігання ядерного палива. Містить виробництво ядерного палива, його переробку та зберігання.

– поводження з радіоактивними відходами та захоронення радіоактивних відходів. Містить обробку, зберігання та захоронення радіоактивних відходів, що виникають під час роботи атомних станцій.

Всі ці галузі енергетики мають важливе значення для забезпечення енергетичних потреб суспільства та економіки, і їх безпека та надійність є критично важливими аспектами для сталого розвитку.

Важливість енергетичної інфраструктури полягає у тому, що енергія необхідна для життєдіяльності людей, функціонування промисловості, комерції, освіти, медицини та багатьох інших аспектів сучасного життя. Ось як енергетична інфраструктура належати до критичної інфраструктури [7]:

– економічна важливість. Енергетична інфраструктура є основним двигуном економіки. Практично всі галузі промисловості та послуги залежать від безперебійного постачання енергії. Аварія або відмова в енергопостачанні може призвести до серйозних економічних втрат;

– соціальна важливість. Енергія необхідна для освіти, охорони здоров'я, тепла в будинках, доступу до інформації та багатьох інших аспектів

повсякденного життя. Брак енергії може вплинути на безпеку та комфорт громадян;

– функціонування критичних інфраструктур. Багато інших критичних інфраструктур, таких як водопостачання, транспорт, телекомунікації та медичні послуги, також залежать від надійного енергопостачання. Енергетична інфраструктура є невід'ємною частиною їх функціонування;

– безпека. Наявність електроенергії для роботи систем безпеки, таких як системи контролю та відеоспостереження, є важливою для запобігання злочинності та підвищення загального рівня безпеки;

– забезпечення надзвичайних ситуацій. Енергетична інфраструктура має грати важливу роль у підтримці функціонування критичних послуг під час надзвичайних ситуацій, таких як природні катастрофи, техногенні аварії або терористичні акти;

– інформаційна безпека. Багато систем управління енергетичною інфраструктурою пов'язані з інформаційними технологіями, і їх безпека є ключовою для запобігання кіберзагрозам та кібератакам на критичні об'єкти.

З урахуванням цих аспектів енергетична інфраструктура стає невід'ємною частиною критичної інфраструктури регіону і вимагає особливої уваги, стратегій забезпечення безпеки та роботи у надзвичайних ситуаціях для забезпечення сталості та безпеки суспільства.

1.3 Консолідація інформації для забезпечення безпеки енергетичної інфраструктури

Енергетична інфраструктура стала безсумнівною системою сучасного суспільства, і її надійність та безпека мають критичне значення для забезпечення сталого розвитку та функціонування інших галузей господарства. Однак зростаюча кількість кіберзагроз та інцидентів в кіберпросторі ставить під загрозу безпеку енергетичних систем, створюючи нагальну потребу у вдосконаленні заходів з кібербезпеки.

Кібербезпека стає ключовим викликом для сучасної енергетичної інфраструктури, оскільки кібератаки можуть призвести до серйозних порушень роботи енергетичних систем, екологічних катастроф та економічних втрат.

Консолідація інформації для забезпечення безпеки енергетичної інфраструктури є критично важливим аспектом забезпечення безпеки та надійності енергетичних систем. Враховуючи величезну важливість енергетичної інфраструктури для суспільства та економіки, а також зростання кіберзагроз та ризиків, які існують у цій галузі, необхідно приділяти особливу увагу забезпеченню кібербезпеки.

Консолідація інформації є процесом об'єднання різних джерел та форм інформації в єдиний, централізований ресурс або систему. Цей процес має на меті створити зібрану інформацію більш доступною, зрозумілою та корисною для подальшого аналізу, використання та прийняття рішень. Консолідація інформації грає важливу роль в різних сферах, включаючи кібербезпеку, енергетику, бізнес-аналітику, наукове дослідження та інші області [8].

Ключові аспекти консолідації інформації для забезпечення кібербезпеки енергетичної інфраструктури включають:

– збір даних;

Спільний збір інформації з різних джерел, таких як системи моніторингу, сенсори, журнали подій, системи керування та інші, дозволяє створити повну картину стану енергетичної інфраструктури. Це містить інформацію про роботу об'єктів, поточний стан систем передачі та постачання енергії, а також дані про кіберінциденти та потенційні загрози.

– структурування та нормалізація даних;

Після збору інформації, важливо стандартизувати та нормалізувати дані для того, щоб зробити їхню обробку та аналіз більш ефективними. Це містить перетворення даних у єдиний формат та забезпечення їхньої взаємодії між собою. Це допомагає уникнути проблем інтеграції та дублювання даних.

– збір реального часу та аналіз в реальному часі;

Багато інформації в енергетичній інфраструктурі надходить у режимі реального часу. Збір та аналіз даних в реальному часі дозволяє виявляти загрози та інциденти миттєво, що сприяє швидкому реагуванню та прийняттю необхідних заходів.

– перевірка та валідація даних;

Збір інформації повинен супроводжуватися процесом перевірки та валідації даних, щоб виявити та коригувати помилки, а також визначити достовірність інформації. Це важливо для запобігання поширенню неправдивої інформації та недостовірних даних.

– захист інформації;

Збір і агрегація інформації повинні супроводжуватися заходами з кібербезпеки, щоб запобігти несанкціонованому доступу до даних та зберегти їхню конфіденційність. Це містить шифрування даних під час передачі, контроль доступу до інформації та захист від кібератак.

– резервне копіювання та відновлення даних;

Інформацію слід регулярно резервувати та забезпечувати можливість відновлення даних в разі аварій, витоків інформації чи інших непередбачуваних ситуацій.

– співпраця та обмін інформацією.

Співпраця між різними гравцями, такими як урядові органи, енергетичні компанії, кібербезпекові організації та інші сторони, є важливою для обміну інформацією та спільного реагування на кіберзагрози.

Консолідація інформації є невід'ємною частиною забезпечення кібербезпеки енергетичної інфраструктури, оскільки вона допомагає створити централізовану платформу для виявлення, аналізу та реагування на кіберзагрози та забезпечити надійність та безпеку роботи енергетичних систем [9].

Поза ключовими аспектами, які були розглянуті раніше, існують ще деякі важливі аспекти, які стосуються консолідації інформації для забезпечення кібербезпеки енергетичної інфраструктури:

– моніторинг та аналіз віддалених об'єктів;

Енергетичні системи можуть містити велику кількість розподілених об'єктів, включаючи підстанції, електростанції та нафтопроводи, які знаходяться в різних місцях. Важливо мати систему моніторингу та аналізу, яка забезпечує доступ до інформації з віддалених об'єктів та дозволяє виявляти проблеми в реальному часі.

– автоматизована система прийняття рішень;

Консолідація інформації може бути пов'язана з автоматизованими системами прийняття рішень, які допомагають реагувати на кіберзагрози та інциденти швидко та ефективно. Ці системи можуть містити алгоритми виявлення аномалій та реагування на них.

– використання штучного інтелекту (ШІ) та аналітики даних;

ШІ та аналітика даних можуть бути використані для автоматичного аналізу великих обсягів інформації та виявлення нестандартних паттернів, що можуть свідчити про кіберзагрози. Це допомагає забезпечити більш ефективну реакцію на потенційні загрози.

– візуалізація даних;

Візуалізація даних графічним чи картографічним способом допомагає користувачам швидше розуміти складні дані та сприяє прийняттю рішень. Інтерактивні графіки та мапи можуть бути корисними для відстеження стану енергетичної інфраструктури.

– створення стандартів і протоколів.

Важливо розробляти стандарти та протоколи для обміну та зберігання інформації в галузі кібербезпеки енергетичної інфраструктури. Це допомагає різним сторонам співпрацювати та обмінюватись інформацією більш ефективно.

Враховуючи ключові аспекти консолідації інформації для забезпечення кібербезпеки енергетичної інфраструктури, наведено схему (рис. 1.2) консолідації даних, спрямованої на ефективне управління енергетичною інфраструктурою.

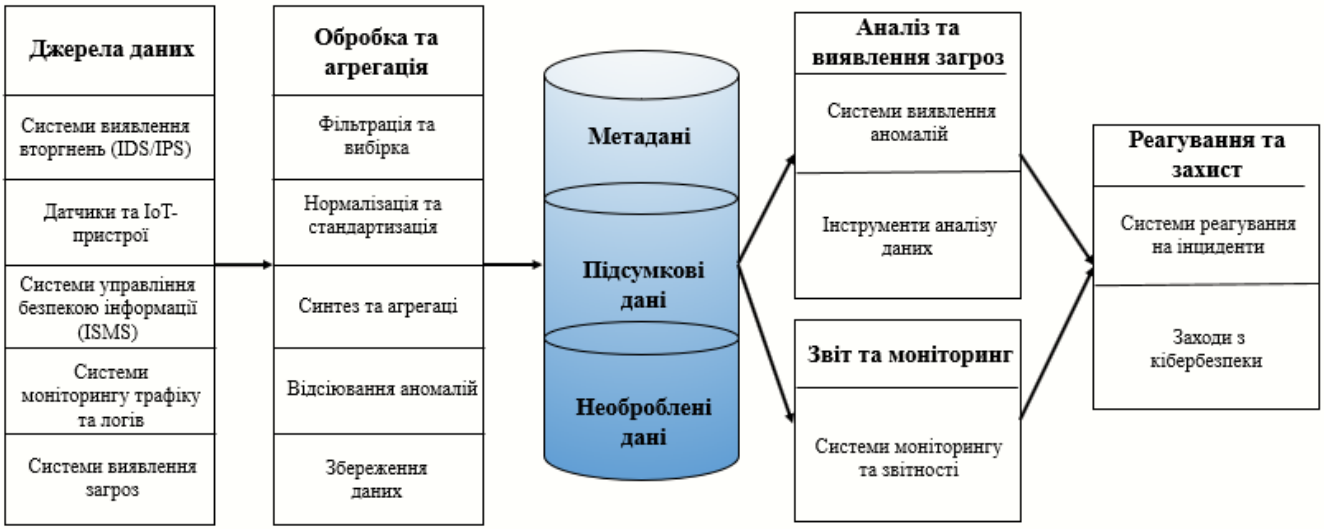


Рисунок 1.2 – Схема консолідації інформації

Вищезазначена схема виступає важливим інструментом, спрямованим на інтеграцію та оптимізацію інформації, яка надходить із різноманітних джерел, з метою забезпечення високого рівня безпеки та ефективності функціонування енергетичної системи.

Загалом, консолідація інформації для забезпечення кібербезпеки енергетичної інфраструктури — це складний та багатогранний процес, який вимагає інтеграції різноманітних технологій та підходів для забезпечення надійності та безпеки енергетичних систем.

1.4 Аналіз методів системного аналізу безпеки об’єктів

В сучасному світі, де технологічний прогрес та складні системи взаємодіють на різних рівнях, аналіз та забезпечення безпеки об'єктів стає ключовим завданням для забезпечення стійкості та ефективності різноманітних систем. Системний підхід до безпеки визначається вивченням та вдосконаленням методів, спрямованих на комплексне розуміння та управління безпековими аспектами об'єктів. У даному контексті важливим є аналіз методів системного аналізу безпеки об'єктів, оскільки він спрямований на розгляд взаємодії компонентів системи, ідентифікацію потенційних ризиків та вирішення завдань

управління безпекою для досягнення оптимальної функціональності та стабільності.

Мета аналізу методів системного аналізу безпеки об'єктів полягає в розгляді та критичному оцінюванні існуючих підходів до забезпечення безпеки об'єктів у рамках системного підходу. Системний аналіз в цьому контексті визначається як комплексна методологія вивчення, моделювання та управління системами, спрямована на досягнення їхньої ефективності та безпеки. Важливими елементами аналізу є ідентифікація потенційних загроз, оцінка ризиків та розробка стратегій протидії для забезпечення оптимального рівня безпеки об'єктів.

Один із основних методів системного аналізу в контексті безпеки об'єктів — це аналіз ризиків. Цей метод містить ідентифікацію можливих загроз, визначення вразливостей системи та оцінку потенційних наслідків подій. Результати аналізу ризиків служать основою для прийняття рішень з розробки та впровадження ефективних стратегій забезпечення безпеки [10].

Аналіз ризиків є ключовим етапом у системному підході до забезпечення безпеки об'єктів. Цей процес визначається як систематична оцінка потенційних загроз та визначення ймовірностей та наслідків можливих подій, що можуть вплинути на об'єкт. Аналіз ризиків дозволяє ефективно ідентифікувати та управляти ризиками, що може призвести до збитків чи непередбачених витрат.

Основні етапи аналізу ризиків можна відобразити у вигляді таблиці 1.1:

Таблиця 1.1 – Основні етапи аналізу ризиків

Етап	Опис
Ідентифікація ризиків	Визначення Загроз. Виявлення потенційних небезпек, які можуть вплинути на об'єкт.
	Визначення Вразливостей. Аналіз вразливостей системи перед можливими загрозами.

Продовження табл. 1.1

Оцінка ризиків	Оцінка Ймовірності. Визначення того, наскільки ймовірно виникнення конкретного ризику.
	Оцінка Наслідків. Визначення масштабу та важливості можливих наслідків.
Визначення прийнятного ризику	Встановлення Критеріїв Прийнятності. Визначення рівнів ризику, які організація готова прийняти.
	Порівняння з Критеріями. Порівняння оцінок ризиків з прийнятними критеріями.
Управління ризиками	Розробка Стратегій Зменшення Ризиків. Визначення та реалізація заходів для зниження ймовірності або впливу ризиків.
	Створення Систем Управління Ризиками. Розробка плану управління ризиками для реагування на ідентифіковані загрози.
Моніторинг та оновлення	Постійний Моніторинг. Систематичний аналіз та моніторинг ризиків на всіх етапах життєвого циклу об'єкта.
	Оновлення Стратегій. Адаптація стратегій управління ризиками відповідно до змін у середовищі або стані об'єкта.

Ця таблиця надає компактний та структурований огляд ключових етапів аналізу ризиків, які дозволяють ефективно управляти можливими загрозами та визначати стратегії безпеки.

Аналіз ризиків дозволяє організаціям ефективно взаємодіяти з невизначеністю та негативними впливами на їхню діяльність. Правильно проведений аналіз ризиків надає можливість приймати обґрунтовані рішення,

забезпечуючи оптимальний рівень безпеки та резильєнтність у відношенні до можливих загроз [11].

Ще одним методом системного аналізу є моделювання систем безпеки. Цей підхід дозволяє розв'язувати проблеми безпеки, використовуючи моделі для представлення взаємодії між різними частинами системи та аналізу впливу зовнішніх факторів.

Моделювання систем безпеки є ефективним підходом для вивчення та аналізу взаємодії різних компонентів системи та їх впливу на загальний стан безпеки. Цей підхід дозволяє створювати абстракції реальних систем та проводити експерименти для визначення оптимальних стратегій забезпечення безпеки.

Основні етапи моделювання систем безпеки включають:

– визначення системи;

Ідентифікація компонентів. Визначення окремих частин системи та їх взаємодії.

Опис Взаємозв'язків. Встановлення, як компоненти взаємодіють між собою та як це впливає на безпеку.

– розробка моделей;

Статичні моделі. Використання статичних діаграм та графіків для відображення структури системи.

Динамічні моделі. Створення моделей, що враховують часові та просторові аспекти взаємодії компонентів.

– імітаційне моделювання;

Створення сценаріїв. Розробка сценаріїв, які відображають потенційні загрози та події.

Запуск симуляцій. Використання імітаційних інструментів для відтворення взаємодії компонентів та аналізу результатів.

– оцінка результатів;

Аналіз показників безпеки. Визначення ключових показників ефективності та безпеки системи.

Ідентифікація вразливостей. Виявлення можливих вразливостей та розробка стратегій для їх запобігання.

– оптимізація системи.

Вдосконалення частин. Розробка оптимальних стратегій для окремих компонентів системи.

Оптимізація системи в цілому. Пошук оптимальних рішень для забезпечення загальної безпеки.

Моделювання систем безпеки дозволяє аналізувати різні сценарії та прогнозувати наслідки подій, що сприяє розробці ефективних стратегій безпеки. Воно також дозволяє виявляти слабкі місця та вразливості системи, що сприяє подальшому удосконаленню систем безпеки в цілому.

Системний аналіз також може включати методи імітаційного моделювання та аналізу сценаріїв, що дозволяють визначити реакцію системи на різні умови та ідентифікувати можливі проблеми та слабкі місця [13].

Методи імітаційного моделювання та аналізу сценаріїв є потужним інструментарієм для вивчення та вдосконалення систем безпеки. Ці методи дозволяють створювати віртуальні моделі систем, відтворювати різні сценарії подій та визначати їхні наслідки.

Основні методи імітаційного моделювання та аналізу сценаріїв та їх застосування наведені в таблиці 1.2:

Таблиця 1.2 – основні методи імітаційного моделювання та аналізу сценаріїв

Метод	Опис	Застосування
Метод масового обслуговування (Discrete Event Simulation, DES)	ES використовує моделі, що передбачають зміни стану системи при виникненні подій. Вони визначають, як система реагує на різні події в часі	DES застосовується для моделювання складних систем, де події відбуваються в конкретні моменти часу
Система динаміки	Цей метод використовує зв'язані диференціальні рівняння для моделювання поведінки систем в часі. Він дозволяє враховувати зміни в часі та їх вплив на різні компоненти системи	Системна динаміка застосовується для аналізу взаємодії між різними елементами системи та їх впливу на безпеку
Метод Монте-Карло	Метод Монте-Карло використовує випадкові експерименти для моделювання великої кількості можливих сценаріїв. Він дозволяє аналізувати статистичні характеристики та ризики	Метод Монте-Карло застосовується для вирішення завдань, де важко або неможливо розв'язати аналітичні рівняння
Агентне моделювання	Агентне моделювання створює моделі індивідуальних агентів та визначає їх взаємодію в системі. Кожен агент має власні характеристики та стратегії	Агентне моделювання використовується для аналізу децентралізованих систем та взаємодії між окремими агентами
Метод випадкових подій	Метод Випадкових Подій використовує статистичні зразки для визначення ймовірностей та розподілу результатів випадкових подій	Цей метод допомагає аналізувати невизначеність та враховувати стохастичні аспекти безпекових сценаріїв

Ці методи імітаційного моделювання та аналізу сценаріїв дозволяють отримувати глибше розуміння взаємодії складних систем та ефективно аналізувати різні можливі сценарії подій для покращення систем безпеки.

Після аналізу вищенаведених методів системного аналізу безпеки об'єктів можна зробити наступні висновки:

– метод масового обслуговування (DES). Підходить для вивчення часових параметрів та подійно-орієнтованих систем, але вимагає значних обчислювальних ресурсів при складних моделях;

– системна динаміка. Забезпечує аналіз динаміки систем в часі та взаємодії компонентів, але має високі обчислювальні витрати та обмежену здатність враховувати невизначеність;

– метод Монте-Карло. Ефективний для аналізу статистичних характеристик і ризиків, але вимагає значних обчислювальних ресурсів та може бути менш точним при складних моделях;

– агентне моделювання. Забезпечує гнучкість та можливість моделювання різних сценаріїв, але вимагає великих обчислювальних ресурсів та точного визначення характеристик агентів;

– метод випадкових подій. Дозволяє аналізувати системи з великою невизначеністю та стохастичністю, але може вимагати значних обчислювальних ресурсів та бути менш точним при обмежених ресурсах.

Кожен метод має свої унікальні переваги та недоліки, і вибір конкретного методу повинен бути обґрунтованим і здійснюватися на основі конкретної задачі та умов використання.

1.5 Аналіз сучасних методів автентифікації користувачів до інформаційного ресурсу

В сучасному цифровому світі, де обмін інформацією є неодмінною складовою повсякденного життя, питання забезпечення безпеки та конфіденційності стає більш нагальним ніж коли-небудь. Автентифікація

користувачів, яка є входом до цього цифрового простору, відіграє визначальну роль у забезпеченні надійного захисту інформаційних ресурсів.

Зростання кількості та складності загроз вимагає вдосконалення методів автентифікації. Від традиційних логінів і паролів до сучасних технологій, таких як біометричні дані та двоетапна аутентифікація, розробники та адміністратори інформаційних систем стикаються з завданням обрати ефективний та зручний метод захисту доступу.

Існує безліч методів автентифікації, які використовуються для забезпечення доступу до інформаційних ресурсів. Враховуючи велику кількість існуючих методів автентифікації, для аналізу було обрано найпопулярніші методи автентифікації, а саме [16]:

- ідентифікатор та пароль;
- двофакторна аутентифікація;
- біометрична аутентифікація;
- карти доступу;
- сертифікати так ключі;
- одноразові паролі;
- SMS-аутентифікація;
- соціальна аутентифікація.

Ці методи можуть комбінуватися для створення більш надійних та безпечних систем аутентифікації в залежності від конкретних потреб та вимог.

Ідентифікатор та пароль — це один з найпоширеніших методів аутентифікації в інтернеті та інших інформаційних системах. Користувачі вказують унікальний ідентифікатор (зазвичай ім'я користувача або електронна пошта) та супровідний пароль для отримання доступу до облікового запису. Ось деякі ключові аспекти цього методу [18]:

- простота використання. Ідентифікатор та пароль легко введені та можуть бути запам’ятовані користувачами;

– ризики безпеки. Однак ідентифікатори та паролі можуть бути вкрадені або взяті на обробку атаками, такими як перехоплення паролів, фішингові атаки, атаки перебору паролів тощо;

– двофакторна аутентифікація. З метою збільшення безпеки ідентифікатор та пароль може бути посилено за допомогою двофакторної аутентифікації, де, крім пароля, використовується додатковий елемент, такий як код, отриманий на мобільний пристрій;

– політики паролів. Для підвищення безпеки важливо встановлювати вимоги до паролів, такі як довжина, використання різних символів і періодична зміна;

– кількість спроб. Деякі системи мають захист від атак перебору паролів, забороняючи доступ після кількох невірних спроб;

– управління паролями. Управління паролями може бути складним завданням для користувачів, і за останні роки виникли інструменти для автоматизації та полегшення цього процесу;

Ідентифікатор та пароль залишаються одним із основних методів аутентифікації, хоча в останні роки спостерігається зростання інтересу до біометричної аутентифікації, двофакторної аутентифікації та інших інноваційних методів забезпечення безпеки [19].

Двофакторна аутентифікація (2FA) - це метод безпеки, що використовується для підтвердження ідентичності користувача, вимагаючи два різних елементи аутентифікації. Це зазвичай містить щось, що користувач знає (наприклад, пароль) і щось, що користувач має або щось, що він є (наприклад, біометричні дані). Кілька основних аспектів двофакторної аутентифікації:

– чинники аутентифікації;

Щось, що ви знаєте: наприклад, пароль або PIN-код.

Щось, що ви маєте: токен безпеки, смарт-карта або інший фізичний пристрій.

Щось, що ви є: біометричні дані, такі як відбиток пальця або розпізнавання обличчя.

– різні форми 2FA;

SMS-коди. Коди, які висилаються на мобільний пристрій користувача. Мобільні додатки. Генерація одноразових кодів за допомогою спеціального додатка. Апаратні токени. Фізичні пристрої, які генерують одноразові коди. Біометрика. Використання фізичних характеристик користувача для аутентифікації. EMV (Europay, MasterCard, Visa) карти. Використання чипів на кредитних чи дебетових картах.

– захист від атак;

2FA допомагає уникнути багатьох атак, таких як фішинг та атаки перехоплення паролів, оскільки навіть якщо зловмисник отримує доступ до одного елемента аутентифікації, він все одно потребує другий для успішного входу.

– популярність і використання;

2FA стала стандартним заходом безпеки в багатьох онлайн-сервісах, таких як банки, електронні поштові сервіси, соціальні мережі та інші.

– розвиток.

Існують інновації, такі як використання біометрики у комбінації з іншими чинниками аутентифікації або впровадження адаптивних систем, які аналізують контекст входу.

Загально кажучи, двофакторна аутентифікація додає додатковий рівень безпеки до звичайного ідентифікатора та пароля, зменшуючи ризик несанкціонованого доступу до облікового запису.

Біометрична аутентифікація — це метод аутентифікації, який використовує фізичні характеристики користувача для підтвердження його ідентичності. Основна ідея полягає в тому, що кожна людина має унікальні біологічні риси, які можна використовувати для ідентифікації. Кілька ключових аспектів біометричної аутентифікації:

– види біометричних даних;

Відбиток пальця. Аналіз унікальних рис на поверхні пальця.

Розпізнавання обличчя. Визначення унікальних особливостей обличчя для ідентифікації.

Розпізнавання райдужки. Використання унікальних характеристик райдужки для аутентифікації.

Голосове визначення. Аналіз унікальних особливостей голосу користувача.

– переваги;

Висока надійність. Біометричні дані зазвичай є унікальними для кожної особи, що робить цей метод дуже надійним.

Зручність використання. Користувачам не потрібно запам'ятовувати паролі чи носити додаткові пристрої — їм лише потрібно використовувати свої фізичні характеристики [20].

– виклики та питання;

Приватність та безпека даних. Збереження і захист біометричних даних є ключовим аспектом, оскільки ці дані можуть бути цінним об'єктом для крадіжки.

Необхідність спеціалізованого обладнання: Деякі види біометричної аутентифікації вимагають спеціалізованого обладнання для зчитування біометричних даних.

– Застосування;

Біометрична аутентифікація широко використовується у сферах фінансів, медицини, громадської безпеки, мобільних пристроїв, комп'ютерів та інших областях.

– розвиток технологій.

Інновації містять розширення використання біометричних технологій у мобільних пристроях, розпізнавання динамічних характеристик (наприклад, рукопису) та комбіновані підходи.

Біометрична аутентифікація продовжує розвиватися, і вона стає все більш популярною завдяки своїм перевагам у забезпеченні безпеки та зручності для користувачів.

Карти доступу — це фізичні чи електронні карти, які використовуються для забезпечення доступу до приміщень, систем або інших областей. Цей метод аутентифікації є популярним у багатьох сферах і містить декілька ключових аспектів:

– типи карт доступу;

Пластикові карти з магнітною смугою. Карти, на яких зберігається інформація на магнітній смузі.

Смарт-карти. Карти з вбудованими чипами, які можуть зберігати та обробляти дані.

Проксіміті-карти. Карти, які використовують технологію RFID або NFC для безконтактного доступу.

Безконтактні картки. Використовують технології, які не вимагають прямого контакту з читачем.

– читачі карт доступу;

Читачі карт з магнітною смугою. Використовуються для зчитування інформації з магнітних смуг.

RFID-читачі. Сприймають сигнали від RFID-карт для безконтактного доступу.

Смарт-картові читачі. Читають інформацію з вбудованих чипів у смарт-картах.

– використання в організаціях;

Карти доступу широко використовуються для фізичного доступу до офісів, лабораторій, складів та інших областей.

Вони також можуть використовуватися для електронного доступу до комп'ютерних систем або мереж.

– заходи безпеки;

Шифрування. Деякі карти доступу можуть використовувати шифрування для захисту інформації, що передається між карткою і читачем.

Багаторівневий доступ. Картки можуть використовуватися в комбінації з іншими методами аутентифікації для створення багаторівневих систем безпеки.

– втрата чи крадіжка карт;

Загроза полягає в тому, що якщо карта доступу втрачена або вкрадена, зловмисник може намагатися використовувати її для отримання неправомірного доступу.

– інтеграція з іншими системами;

В деяких випадках карти доступу можуть бути інтегровані з іншими системами, такими як системи обліку робочого часу чи системи безпеки.

Використання карт доступу є популярним і ефективним методом фізичного та логічного контролю доступу в організаціях.

Сертифікати та ключі використовуються у криптографії для забезпечення безпеки електронних комунікацій та інших інформаційних процесів.

Ключові аспекти їх використання:

– криптографічні ключі;

Симетричні ключі. Використовують один і той самий ключ для як шифрування, так і розшифрування даних. Забезпечують швидке шифрування, але виникає проблема безпеки обміну ключами.

Асиметричні ключі (публічний та приватний). Використовують пару ключів, один для шифрування та інший для розшифрування. Приватний ключ зберігається конфіденційно, а публічний ключ відомий для інших. Забезпечують безпеку обміну даними, але можуть бути менш ефективними за симетричні ключі.

– сертифікати;

SSL/TLS-сертифікати. Використовуються для шифрування даних, що передаються між веб-сервером і користувачем. Підтверджують автентичність веб-сайту.

Сертифікати електронного підпису. Використовуються для підтвердження автентичності відправника документа або повідомлення.

Сертифікати для VPN-з'єднань. Використовуються для безпечного з'єднання з віртуальною приватною мережею.

– процес отримання та використання сертифікатів;

Запит на сертифікат. Заява на отримання сертифіката, яку робить користувач чи організація.

Верифікація. Авторитет сертифікації перевіряє, чи дані у запиті є вірними, перед тим як видати сертифікат [21].

Видача сертифіката. Якщо верифікація успішна, сертифікат видано.

Використання сертифіката. Сертифікат використовується для підтвердження ідентичності та забезпечення безпеки обміну даними.

– інфраструктура відкритих ключів (PKI);

PKI включає всі компоненти, що визначають, як видаються, зберігаються, передаються та перевіряються сертифікати та ключі.

Забезпечує безпеку та автентичність у відкритих мережах.

– регулювання та стандарти.

Існують стандарти, такі як X.509, які визначають формати для сертифікатів та ключів.

Деякі аспекти роботи з сертифікатами можуть регулюватися законодавством.

Сертифікати та ключі грають важливу роль у забезпеченні конфіденційності та інтегритету даних, а також у визначенні автентичності користувачів та систем.

Одноразовий пароль (OTP) - це вид автентифікаційного коду, який діє лише один раз під час конкретного сеансу або транзакції. OTP широко використовується для забезпечення додаткового рівня безпеки під час входу в систему, здійснення фінансових операцій або інших конфіденційних дій. Ось кілька ключових аспектів OTP:

– генерація OTP;

Часові базові (Time-based OTP): Генеруються на основі часу та секретного ключа, що відомого як seed. Вони діють протягом короткого періоду (зазвичай 30 секунд).

Засновані на подіях (Event-based OTP): Генеруються при кожній події, такій як натискання кнопки на пристрої або підписання операції.

– доставка OTP;

SMS: Коди відправляються на мобільний телефон користувача через текстове повідомлення.

Мобільні додатки: Коди генеруються у мобільних додатках, таких як Google Authenticator чи Authy.

Електронна пошта. OTP може бути відправлений на електронну пошту користувача.

Фізичні пристрої. OTP може бути генерований на спеціальних апаратних пристроях.

– часове обмеження;

Більшість OTP має обмежений термін дії для запобігання використанню протягом довгого часу в разі його викрадення.

– безпека;

Використання OTP додає безпеки, оскільки кожен код може бути використаний лише один раз, що ускладнює завдання зловмисників.

– поширеність в банківському секторі;

Багато банків використовують OTP для підтвердження операцій через Інтернет або для здійснення транзакцій.

– стандарти та протоколи.

Протоколи, такі як HOTP (HMAC-based One-Time Password) та TOTP (Time-based One-Time Password), визначають стандарти для генерації та перевірки OTP.

Використання одноразових паролів допомагає усунути ризик витоку паролів та забезпечити додатковий рівень безпеки для електронних сервісів та транзакцій.

SMS-аутентифікація - це метод аутентифікації, при якому одноразовий код (OTP) або інша форма автентифікаційної інформації надсилається користувачеві через текстове повідомлення (SMS). Цей метод широко використовується для забезпечення безпеки та підтвердження ідентичності користувача при доступі до різних систем, особливо в банківському секторі та електронних сервісах. Ключові аспекти SMS-аутентифікації [22]:

– використання одноразових кодів;

Користувач отримує унікальний одноразовий код через SMS. Цей код може використовуватися лише один раз і має обмежений термін дії.

– процес SMS-аутентифікації;

Користувач вводить свій ідентифікатор (наприклад, номер телефону) на веб-сайті чи в додатку. Система генерує одноразовий код та надсилає його користувачеві через SMS. Користувач вводить отриманий код для завершення аутентифікації.

– доставка через SMS;

Код може бути відправлений на мобільний телефон користувача через SMS-повідомлення. Після отримання SMS, користувач повинен ввести код у відповідне поле в системі.

– безпека;

SMS-аутентифікація додає додатковий рівень безпеки, оскільки вимагає наявності фізичного доступу до мобільного телефону користувача.

Однак існують зауваження стосовно безпеки SMS-аутентифікації, зокрема, вразливості до атак, таких як SIM-картковий обман або перехоплення SMS.

– застосування;

SMS-аутентифікація широко використовується в банківському секторі, електронних комерційних платформах, соціальних мережах та інших сервісах.

– можливі ризики.

Хоча SMS-аутентифікація є популярною, існують ризики, такі як можливість перехоплення SMS або атаки на мобільні оператори.

SMS-аутентифікація є зручним та широко застосовуваним методом, але важливо враховувати потенційні ризики та шукати більш безпечні альтернативи, такі як додатки для генерації одноразових кодів чи фізичні токени.

Соціальна аутентифікація — це метод аутентифікації, при якому користувач використовує свій обліковий запис з іншого веб-сервісу чи соціальної мережі для входу на інший веб-сайт чи додаток. Замість того, щоб створювати та запам'ятовувати новий пароль, користувач використовує свої існуючі облікові дані для підтвердження своєї ідентичності. Кілька ключових аспектів соціальної аутентифікації [23]:

– соціальні мережі та сервіси;

Популярні соціальні мережі, такі як Facebook, Google, Twitter, або LinkedIn, часто використовуються як ідентифікатори для аутентифікації.

– процес аутентифікації;

Користувач обирає опцію "Увійти за допомогою Facebook" чи іншої соціальної мережі на веб-сайті, чи додатку.

Система направляє користувача на сторінку відповідної соціальної мережі, де користувач підтверджує свою ідентичність.

Після підтвердження, інформація від соціальної мережі передається до веб-сайту чи додатку для автоматичного входу.

– переваги соціальної аутентифікації;

Зручність для користувача. Уникнення необхідності запам'ятовувати та управляти багатьма паролями.

Швидкий вхід. Процес входу стає більш швидким та простим.

– безпека;

Додаткові заходи безпеки. Багато сервісів, які використовують соціальну аутентифікацію, дозволяють встановлювати додаткові заходи безпеки, такі як підтвердження паролям чи двофакторна аутентифікація.

– дозвіл на обмін інформації;

Залежно від конкретної реалізації, користувач може надавати додаткові дозволи на обмін інформацією між веб-сайтом та соціальною мережею.

– питання приватності;

Важливо бути уважним до питань приватності, оскільки інформація про користувача може передаватися між платформами.

– реалізація на веб-сайтах та додатках;

Багато веб-сайтів та додатків надають можливість увійти за допомогою різних соціальних облікових записів.

Соціальна аутентифікація дозволяє зробити процес входу більш зручним для користувачів та може зменшити кількість паролів, які потрібно запам'ятовувати. Однак важливо ретельно керувати заходами безпеки та обліком приватності при використанні цього методу.

Враховуючи аналіз різних методів аутентифікації, можна зазначити такі результати у таблиці (табл. 1.3) порівняння:

Таблиця 1.3 – порівняння методів автентифікації

Метод автентифікації	Простота впровадження (1-10)	Рівень безпеки	Зручність для користувача	Залежність від додат-кових засобів	Поширеність використання
Ідентифікатор та пароль	10	Низький	7	Ні	Висока
Двофакторна аутентифікація	8	Високий	6	Так	Середня
Біометрична аутентифікація	7	Високий	8	Так	Середня
Одноразові Паролі	8	Високий	7	Так	Середня
SMS-аутентифікація	9	Середній	6	Так	Середня
Соціальна Аутентифікація	9	Середній	8	Так	Середня

Аналіз таблиці вказує на те, що вибір конкретного методу є завданням, яке вимагає збалансованого підходу та врахування ряду ключових параметрів.

Ідентифікація за допомогою ідентифікаторів та паролів, несумнівно, є широко використовуваним та досить простим методом, проте його безпека є вразливою, а зручність для користувача середня. З іншого боку, двофакторна аутентифікація та біометричні методи виявляються високими за безпекою, проте рівень їх зручності для користувача та простота впровадження можуть варіюватися в залежності від реалізації [25].

Одноразові паролі через SMS або інші канали забезпечують високий рівень безпеки, але при цьому можуть стикатися з питаннями простоти впровадження та можливістю перехоплення кодів. Використання токенів безпеки підвищує рівень безпеки, але може вимагати фізичних пристроїв та мати середню зручність для користувача.

Соціальна аутентифікація, хоча зручна для користувача та поширена, зазнає критики через свою залежність від сторонніх сервісів та низький рівень безпеки.

У кінцевому підсумку, розуміння цих аспектів є критичним для розробки та вдосконалення методів аутентифікації в цифровій ері, що відкриває нові можливості та ставить перед собою завдання підвищення рівня безпеки та вдосконалення зручності використання для користувачів.

1.6 Висновки та постановка задачі

Отже, в даному розділі було проведено аналіз сучасних стратегій забезпечення безпеки інформаційних ресурсів енергетичної інфраструктури. Розглянуті методи системного аналізу для оцінки безпеки об’єктів енергетичної галузі, яка є критичною складовою інфраструктури регіону.

У ході загального огляду теоретичного матеріалу сформульовано наступні завдання:

– розробити захищений консолідований інформаційний ресурс, спрямований на забезпечення безпеки енергетичної інфраструктури;

– розробити алгоритм оцінювання безпеки енергетичних об'єктів;

– створити базу даних для зберігання інформації щодо безпеки енергетичних систем;

– провести дослідження коректності та ефективності функціонування розробленого інформаційного ресурсу.

Здійснення всіх поставлених завдань сприятиме досягненню головної мети цієї роботи — розробки захищеного консолідованого інформаційного ресурсу для системного аналізу безпеки енергетичної інфраструктури регіону.

2 РОЗРОБКА ЗАХИЩЕНОГО КОНСОЛІДОВАНОГО ІНФОРМАЦІЙНОГО РЕСУРСУ

У даному розділі буде здійснено аналіз особливостей створення інформаційного ресурсу аналізу безпеки енергетичної інфраструктури, буде здійснена розробка бази даних та розроблено систему забезпечення захисту, після чого буде сформовано висновки по роботі виконаній в даному розділі.

2.1 Особливості розробки інформаційного ресурсу аналізу безпеки енергетичної інфраструктури

Розробка інформаційного ресурсу для аналізу безпеки енергетичної інфраструктури — це складний процес, що вимагає уваги до багатьох аспектів.

Збір та обробка даних є ключовим етапом розробки інформаційного ресурсу для аналізу безпеки енергетичної інфраструктури. В рамках розробки інформаційного ресурсу для аналізу безпеки енергетичної інфраструктури важливо звернути увагу на кілька ключових аспектів, які визначають якість та ефективність системи. До цих аспектів відносять контроль якості даних, конфіденційність та безпеку даних, реалізацію механізмів резервного копіювання, стандартизацію та нормалізацію даних, а також транспортування та зберігання інформації.

Контроль якості даних передбачає визначення метрик якості, систем моніторингу та механізмів виявлення та виправлення помилок. Це забезпечує точність, повноту та актуальність інформації. Зокрема, важливо регулярно перевіряти метрики та вживати заходів для усунення виявлених недоліків.

Конфіденційність та безпека даних вимагають використання сучасних методів шифрування, механізмів аутентифікації та авторизації, а також заходів фізичної безпеки для захисту від несанкціонованого доступу. Системи моніторингу безпеки допомагають виявляти та реагувати на можливі загрози чи аномалії [27].

Реалізація механізмів резервного копіювання містить визначення критичних даних, автоматизоване створення резервних копій та тестування можливості відновлення інформації. Забезпечення ефективного управління резервними копіями дозволяє ефективно впоратися з можливими проблемами.

Стандартизація та нормалізація даних передбачають використання загальноприйнятих стандартів, розробку уніфікованих схем даних та застосування метаданих для полегшення їх інтерпретації. Це сприяє уникненню розбіжностей та забезпечує однаковий формат для подальшого аналізу.

Транспортування та зберігання даних містить вибір ефективних протоколів передачі, конфігурацію систем зберігання, методи резервного копіювання та інтеграцію з іншими системами. Це дозволяє забезпечити надійність, безпеку та доступність інформації.

Цей процес також включає вивчення галузевих стандартів і нормативів, які регулюють обробку та захист даних у сфері енергетики. Це важливо для визначення конкретних вимог, що встановлюються для інформаційних ресурсів в даній галузі.

Збір вимог від користувачів є іншим ключовим етапом, який передбачає консультації з потенційними користувачами та іншими стейкхолдерами. Це дозволяє отримати інсайти щодо їхніх конкретних потреб та вимог, щоб забезпечити розробку ресурсу, який відповідає їхнім очікуванням [28].

Технічний аналіз включає розроблення технічних вимог, таких як забезпечення конфіденційності, цілісності та доступності даних. На цьому етапі визначаються технічні характеристики та вимоги до архітектури системи. Такий аналіз допомагає створити технічну основу для розробки безпечного та ефективного інформаційного ресурсу.

Політики захисту даних містить розробку політик та правил, які встановлюють стандарти та процедури для обробки, зберігання та передачі інформації. Класифікація даних за ступенем чутливості та введення правил для їх обробки є також частиною цього процесу. У цьому контексті розробляються механізми для захисту конфіденційної інформації.

Аналіз законодавства містить юридичний аналіз, спрямований на дотримання відповідних нормативів та законів, які стосуються обробки та захисту даних. Забезпечення відповідності цим вимогам включає розробку стратегій та заходів для ефективного контролю за юридичною відповідністю системи. Оцінка ризиків та регулярний юридичний аудит допомагають впевнитися в тому, що система дотримується усіх вимог законодавства та нормативів.

2.2 Розробка бази даних консолідованого інформаційного ресурсу

Проєктування структури бази даних є важливим етапом у розробці консолідованого інформаційного ресурсу. Цей процес містить розробку логічної та фізичної моделей бази даних, визначення таблиць, відносин між ними, ключів, індексів та інших елементів, що визначають структуру та організацію зберігання інформації.

Для консолідованого інформаційного ресурсу, спрямованого на аналіз безпеки енергетичної інфраструктури, потрібно використовувати таблиці для ефективного зберігання та управління інформацією. Таблиця "Користувачі" міститиме дані про користувачів системи, такі як ім'я, прізвище, логін, хеш пароля та роль. Таблиця "Аналізи" буде зберігати результати проведених аналізів, включаючи назву, дату проведення, опис, результати, відповідального аналітика та пов'язаний об'єкт. В "Енергетичних Системах" будуть дані про різні енергетичні системи та їх параметри, такі як тип, технічні характеристики та дата введення в експлуатацію.

"Енергетичні Об'єкти" міститимуть інформацію про окремі об'єкти, включаючи назву, тип, місцезнаходження та власника.

У "Заходах Безпеки" буде інформація про заходи, що спрямовані на забезпечення безпеки енергетичних об'єктів, такі як опис, дата введення в дію та пов'язаний об'єкт. Ці таблиці допоможуть в організації та зберіганні інформації про користувачів, результати аналізів, енергетичні системи та об'єкти, а також заходи безпеки [29].

Першим кроком в методі проєктування бази даних сутність-зв'язок є визначення сутностей, які повністю відображатимуть інформаційні потреби користувача майбутньої бази даних.

Виділимо сутності, які будуть містити інформацію:

- Users
- Analysis
- EnergySystem
- EnergyObjects
- SecurityMeasuares

Наступним кроком визначимо зв’язки між обраними сутностями:

- Users здійснює Analysis
- Analysis включають EnergySystem
- EnergySystem містить EnergyObjects
- EnergyObjects мають SecurityMeasuares

Наступним кроком визначаємо атрибути обраних сутностей та ключі для кожної із сутностей:

- Users (<User id>, name, email, password);
- Analysis (<Analysis id>, analysis name, user id, system id, analys result, analys date);
- EnergySystem (<EnergySystem id>, system name, system description, system technical specification);
- EnergyObjects (<EnergyObjects id>, object name, object type, object location, id system);
- SecurityMeasuares (<SecurityMeasuares id>, measuares description, measuares implementation date, measuares type, object id);

Наступним кроком визначимо типи зв’язків між сутностями:

Визначення ступня зв`язку та класу належності сутностей «Users» та «Analysis» (рис. 2.1).

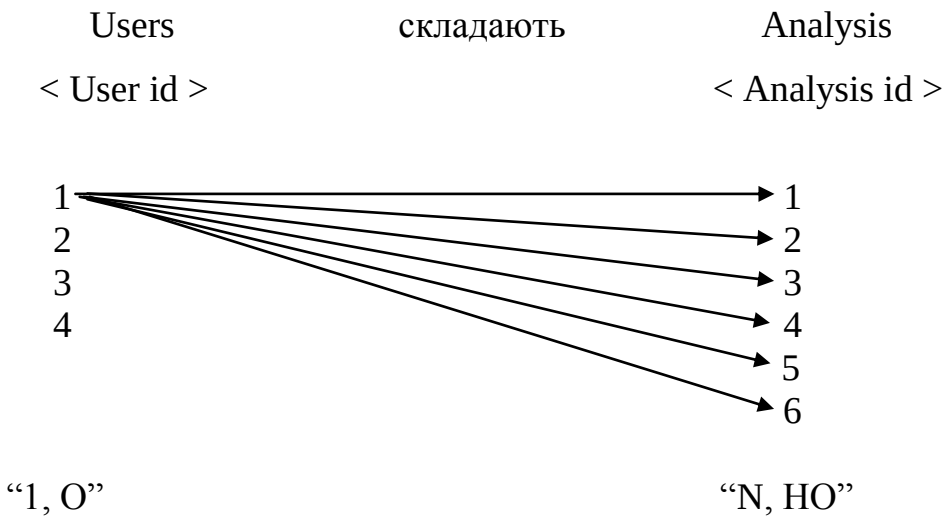


Рисунок 2.1 – Аналіз зв’язку сутностей «Users» та «Analysis

Далі ступні зв`язку та класу належності сутностей «Analysis» та «EnergySystem» (рис. 2.2).

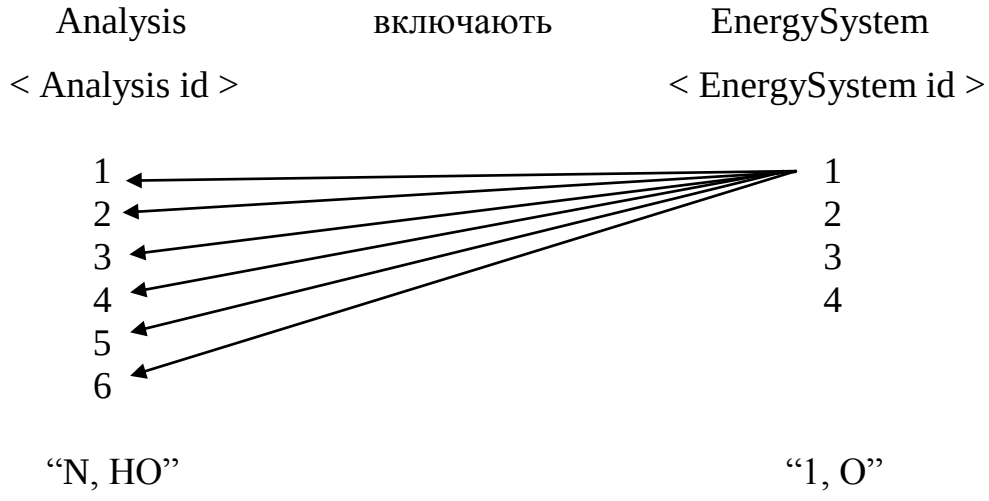


Рисунок 2.2 – Аналіз зв’язку сутностей «Analysis» та «EnergySystem»

Визначення ступеня зв`язку та класу належності сутностей «EnergySystem» та «EnergyObjects» (рис. 2.3).

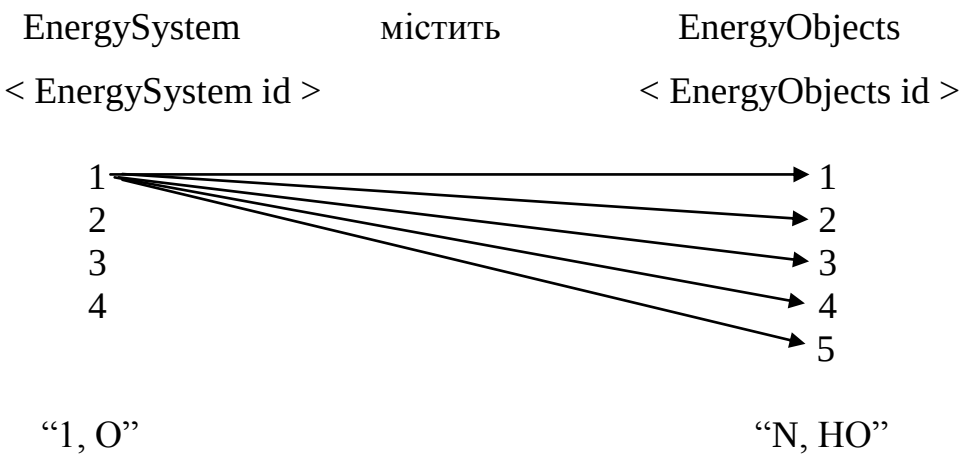


Рисунок 2.3 – Аналіз зв’язку сутностей «EnergySystem» та «EnergyObjects»

Визначення ступеня зв`язку та класу належності сутностей «EnergyObjects» та «SecurityMeasuares» (рис. 2.4).

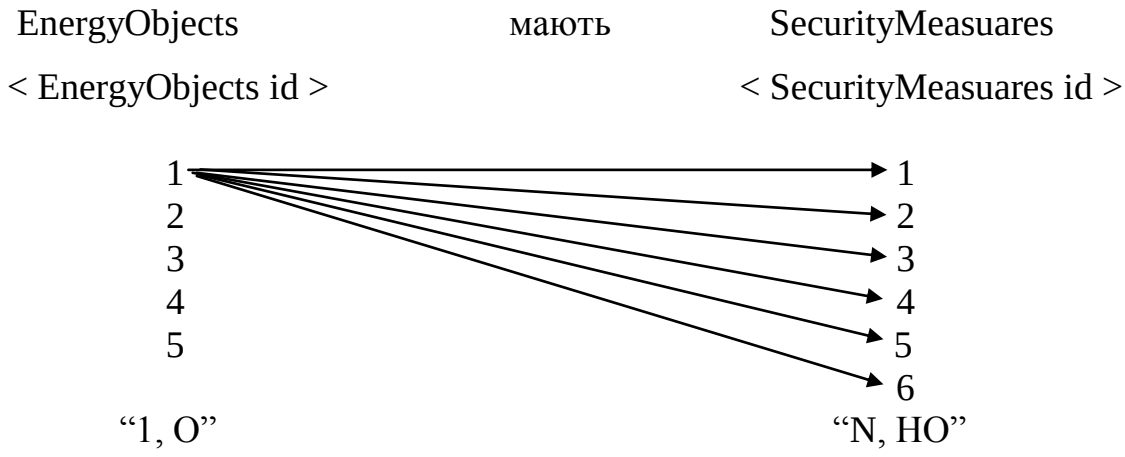


Рисунок 2.4 – Аналіз зв’язку сутностей «EnergyObjects» та «SecurityMeasuares»

Цей процес розробки бази даних стримувався на створенні системи, структурованої та ефективної, яка в змозі враховувати важливі аспекти аналізу безпеки енергетичної інфраструктури, забезпечуючи високу якість зберігання та обробки інформації для підтримки прийняття рішень та вчасної реакції на потенційні загрози.

2.3 Проектування ER-моделі бази даних

Проектування ER-моделі бази даних - це процес створення абстрактної моделі бази даних, що використовує концепції сутність-зв'язок (ER) для представлення інформації та взаємозв'язків між різними об'єктами домену. ER-модель дозволяє описати структуру даних, визначити, як дані будуть зберігатися та взаємодіяти між собою.

Результатом проектування буде ER-модель зображена на рисунку 2.5.

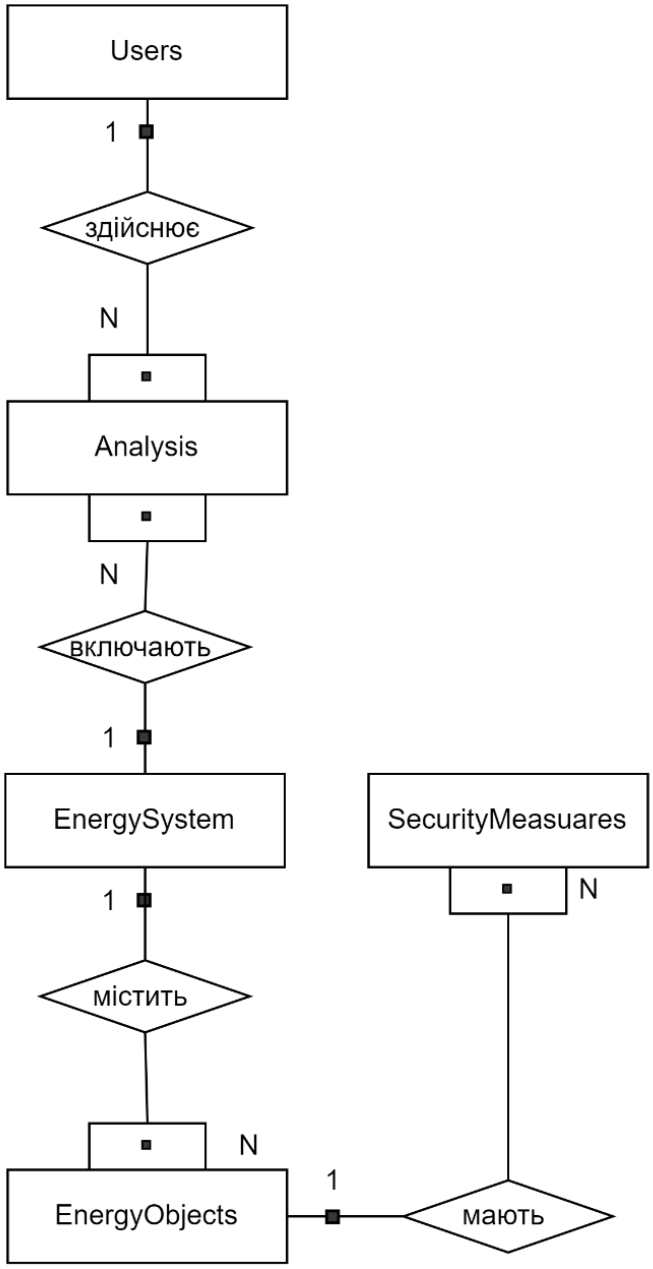


Рисунок 2.5 - ER-модель

Таким чином, в результаті проектування було отримано такі попередні відношення:

– Users (<User id>, name, email, password);

– Analysis (<Analysis id>, analysis name, user id, system id, analys result, analys date);

– EnergySystem (<EnergySystem id>, system name, system description, system technical specification);

– EnergyObjects (<EnergyObjects id>, object name, object type, object location, id system);

– SecurityMeasuares (<SecurityMeasuares id>, measuares description, measuares implementation date, measuares type, object id);

2.4 Нормалізації відношень бази даних

Нормалізація відношень в базах даних є важливим етапом у процесі проєктування та управління інформаційними системами. Цей процес спрямований на досягнення оптимальної організації даних з метою поліпшення ефективності та надійності БД. Однією з ключових переваг нормалізації є мінімізація дублювання інформації, що сприяє консистентності та використанню ресурсів.

Додатково, цей процес дозволяє зменшити обсяг збережених даних, що сприяє економії дискового простору та оптимізації продуктивності системи. Ефективність запитів покращується завдяки розподілу інформації між таблицями, уникаючи зайвих обчислень. Крім того, нормалізація дозволяє забезпечити цілісність даних, запобігаючи аномаліям введення та модифікації інформації.

Важливим аспектом є також легкість модифікації та розширення бази даних без значних труднощів. Це стає особливо актуальним у змінних умовах вимог до даних в сучасних інформаційних середовищах. Загалом, нормалізація відношень в БД виступає ключовим елементом для забезпечення ефективності та стабільності функціонування систем, що робить її необхідною складовою процесу розробки та управління базами даних.

Нормальні форми в базах даних є ключовими концепціями, спрямованими на удосконалення структури даних та усунення аномалій, що можуть виникнути при їх використанні. Перша нормальна форма (1НФ) вимагає, щоб значення кожного атрибута в таблиці було атомарним, тобто не розкладалося на менші частини. Це запобігає повторенню даних та забезпечує атомарність полів.

Друга нормальна форма (2НФ) передбачає виконання 1НФ і додає умову, щоб кожен неключовий атрибут залежав від усіх ключових атрибутів, а не лише від частини ключа. Це допомагає уникнути залежностей між атрибутами та виправдовується бажанням уникнути аномалій, пов'язаних зі змінами у базі даних.

Третя нормальна форма (3НФ) враховує попередні нормальні форми та додає вимогу, щоб кожен неключовий атрибут не залежав від інших неключових атрибутів, тобто не мав транзитивних залежностей від ключа. Це сприяє уникненню дублювання даних та підтримує консистентність інформації в таблицях [30].

Використання цих нормальних форм дозволяє створювати оптимізовані та структуровані схеми даних, сприяючи надійному та легкому управлінню базами даних.

Визначимо інформацію, яка необхідна при роботі захищеного консолідованого інформаційного ресурсу системного аналізу безпеки.

– Users (<User id>, name, email, password);

– Analysis (<Analysis id>, analysis name, user id, system id, analys result, analys date);

– EnergySystem (<EnergySystem id>, system name, system description, system technical specification);

– EnergyObjects (<EnergyObjects id>, object name, object type, object location, id system);

– SecurityMeasuares (<SecurityMeasuares id>, measuares description, measuares implementation date, measuares type, object id);

Для реалізації поставленої задачі універсальне відношення буде мати вигляд:

R(User id, name, email, password, Analysis id, analysis name, user id, system id, analys result, analys date, EnergySystem id, system name, system description, system technical specification, EnergyObjects id, object name, object type, object location, id system, SecurityMeasuares id, measuares description, measuares implementation date, measuares type, object id).

Проведемо нормалізацію відношень за трьома формами.

Перша нормальна форма:

R1(User id, name, email, password, Analysis id, analysis name, user id, system id, analys result, analys date, system name, system description, system technical specification, objects id, object name, object type, object location, id system, SecurityMeasuares id, measuares description, measuares implementation date, measuares type).

Друга нормальна форма:

R2(<system id> Analysis id, analysis name, user id, analys result, analys date, system name, system description, system technical specification, objects id, object name, object type, object location, id system, SecurityMeasuares id, measuares description, measuares implementation date, measuares type).

R3 (<User id>, name, email, password)

Третя нормальна форма:

R4(<User id>, name, email, password);

(<Analysis id>, analysis name, user id, system id, analys result, analys date);

R5 (<EnergySystem id>, system name, system description, system technical specification);

R6 (<EnergyObjects id>, object name, object type, object location, id system);

R7 (<SecurityMeasuares id>, measuares description, measuares implementation date, measuares type, object id);

R8 (<Analysis id>, analysis name, user id, system id, analys result, analys date);

Виходячи з проєктування, кінцевими відношеннями будуть: R4, R5, R6, R7, R8. На їх основі й відбуватиметься розробка бази даних захищеного консолідованого інформаційного ресурсу системного аналізу безпеки.

Отже, з наведених вище схем можна розробити схему майбутньої бази даних (рис. 2.6).

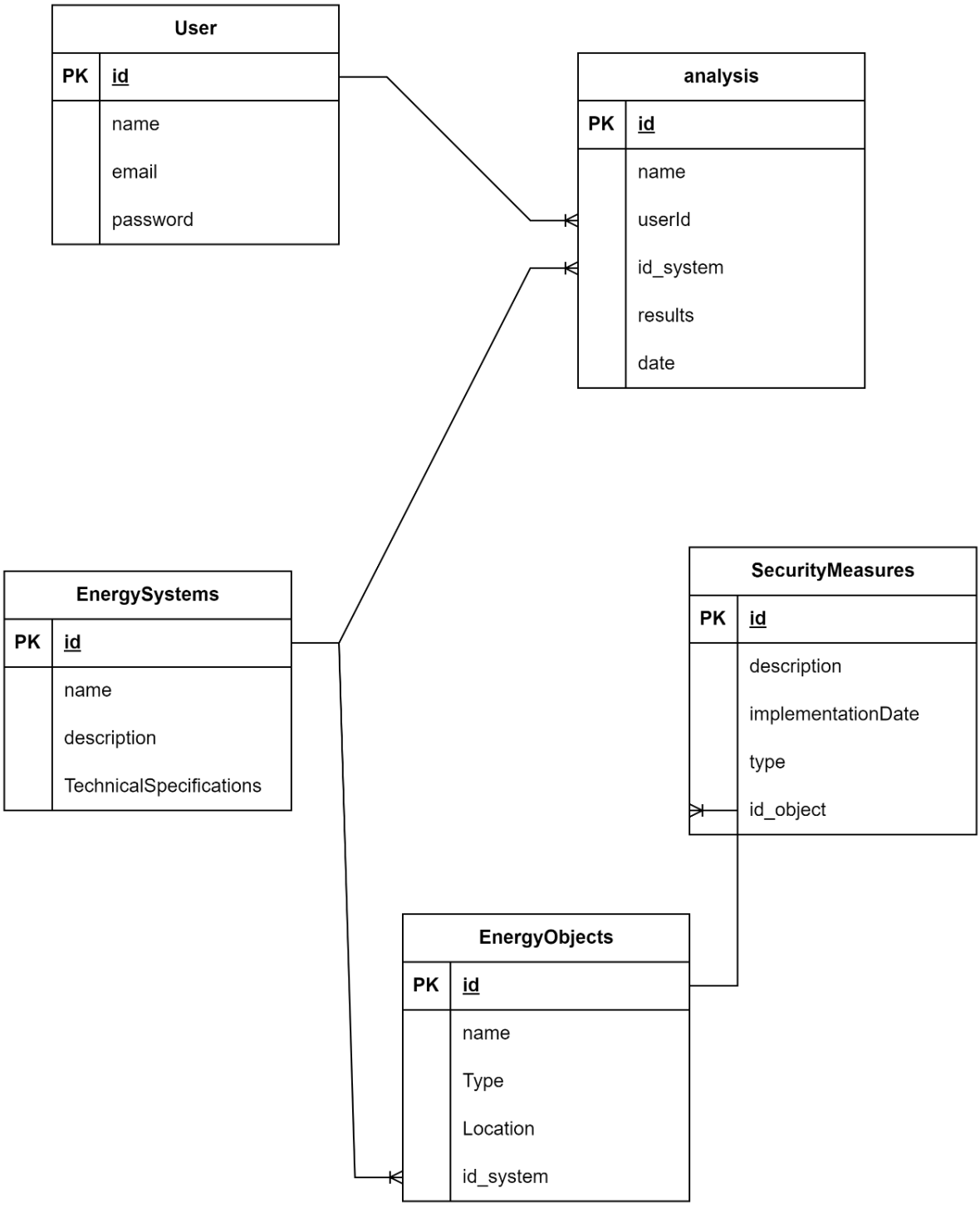


Рисунок 2.6 - Схема бази даних консолідованого інформаційного ресурсу

Отже, розробка бази даних для консолідованого інформаційного ресурсу, спрямованого на аналіз безпеки енергетичної інфраструктури, становить собою ключовий етап проєкту. Процес містить складне проєктування структури бази даних, визначення таблиць та взаємозв'язків між ними, а також встановлення стандартів для збору, обробки та аналізу інформації.

Структура бази даних включала таблиці, що сприймали та узагальнювали інформацію про користувачів системи, результати проведених аналізів, характеристики енергетичних систем та об'єктів, а також заходи безпеки. Кожна таблиця мала свої унікальні поля та взаємозв'язки з іншими таблицями для забезпечення цілісності та ефективності бази даних.

Застосування стандартів безпеки, регулярне резервне копіювання та систематична перевірка якості даних в системі були реалізовані для забезпечення високого рівня надійності та стабільності інформаційного ресурсу.

2.5 Проектування звітів.

Однією з ключових мет цілей створення консолідованого інформаційного ресурсу є подання зведеної статистичної інформації. Ця інформація надалі допоможе підвищити ефективність роботи. Для успішної роботи в даній сфері необхідно мати повний обсяг інформації, яка буде представлена у зручному форматі та надавати можливості для автоматичного узагальнення, систематизації та подальшого аналізу даних, вже відсортованих, згрупованих та підрахованих.

Звіт є представленням даних у спеціальному форматі, яке може бути відображене на екрані, збережене у файлі або надруковане. Звіти використовуються для створення документів на основі таблиць і запитів. На відміну від форм, вони не призначені для введення даних у таблиці. При друку таблиць і запитів інформація виводиться практично в тому вигляді, у якому вона зберігається. Часто виникає потреба представити дані у вигляді звітів, які мають традиційний вигляд і легко читаються. Детальний звіт включає всю інформацію з таблиці або запиту, але має заголовки та розділений на сторінки із верхніми і

нижніми колонтитулами. Типовою особливістю будь-якого звіту є можливість перегляду записів з однієї або декількох таблиць. Важливо пам'ятати, що звіт може бути пов'язаний або з однією таблицею, або з запитом, який має доступ до однієї або декількох таблиць. Для створення звіту потрібно вибрати поля запиту або таблиці. Якщо даних з однієї таблиці недостатньо, то, ймовірно, звіт потрібно пов'язати із запитом. У випадку використання даних з декількох таблиць такого вибору немає, і звіт доведеться пов'язати із запитом.

Для розробленого інформаційного ресурсу розробимо наступні звіти:

- список користувачів системи;
- список проведених аналізів безпеки ;
- список проведених аналізів безпеки інфраструктури;
- список енергетичних інфраструктур;
- список об’єктів енергетичної інфраструктури;
- список методів захисту об’єкта інфраструктури;
- рейтинг енергетичних інфраструктур по результатах аналізів безпеки;
- рейтинг енергетичних інфраструктур за кількістю об’єктів інфраструктури;
- рейтинг об’єктів інфраструктури за кількістю засобів захисту;
- звіт по типам об’єктів енергетичної інфраструктури.

Звіт про список користувачів системи буде відображати список користувачів. Атрибутами при створенні звіту будуть: ім’я користувача, електрона пошта користувача.

Звіт про список проведених аналізів безпеки буде відображати список проведених аналізів безпеки всіх енергетичних інфраструктур. Атрибутами при створенні звіту будуть: назва інфраструктури, результат аналізу, дата проведення аналізу.

Звіт про список проведених аналізів безпеки буде відображати список проведених аналізів безпеки конкретної енергетичної інфраструктури.

Атрибутами при створенні звіту будуть: назва інфраструктури, результат аналізу, дата проведення аналізу.

Звіт про список енергетичних інфраструктур буде відображати список всіх енергетичних інфраструктур. Атрибутами при створенні звіту будуть: назва інфраструктури, опис інфраструктури.

Звіт про список об’єктів енергетичної інфраструктури буде відображати список об’єктів енергетичної інфраструктури. Атрибутами при створенні звіту будуть: назва об’єкта, тип об’єкта, локація об’єкта.

Звіт про список методів захисту об’єкта інфраструктури буде відображати список методів захисту об’єкта інфраструктури. Атрибутами при створенні звіту будуть: опис захисту, дата встановлення, тип захисту.

Звіт про рейтинг енергетичних інфраструктур за результатами аналізів безпеки буде відображати рейтинговий список енергетичних інфраструктур за результатами аналізів безпеки. Атрибутами при створенні звіту будуть: назва інфраструктури, середній результат аналізу безпеки [31].

Звіт про рейтинг енергетичних інфраструктур за кількістю об’єктів інфраструктури буде відображати рейтинговий список енергетичних інфраструктур за кількістю об’єктів інфраструктури. Атрибутами при створенні звіту будуть: назва інфраструктури, кількість об’єктів інфраструктури.

Звіт про рейтинг об’єктів інфраструктури за кількістю засобів захисту буде відображати рейтинговий список об’єктів інфраструктури за кількістю засобів захисту. атрибутами при створенні звіту будуть: Назва об’єкту, кількість засобів захисту.

Звіт за типами об’єктів енергетичної інфраструктури буде відображати рейтинг об’єктів інфраструктури по їх типу. Атрибутами при створенні звіту будуть: тип об’єктів, кількість.

2.6 Забезпечення захисту розробленого ресурсу від несанкціонованого доступу.

Забезпечення захисту розробленого ресурсу від несанкціонованого доступу вимагає комплексного та добре продуманого підходу.

Двоетапна аутентифікація (2FA) є важливим елементом для ефективного захисту розробленого ресурсу з кількох причин. Вона додає додатковий шар безпеки, оскільки для отримання доступу користувач повинен пройти через два етапи перевірки: зазвичай, щось, що вони знають (наприклад, пароль), та щось, що вони мають (наприклад, мобільний пристрій або безпековий токен).

2FA ускладнює завдання зловмисників, які вкрадуть або підкуплять паролі. Навіть якщо зловмисники отримають доступ до пароля, їм також потрібно мати фізичний доступ до другого фактора аутентифікації. Це усуває ризик, пов'язаний із використанням лише одного елемента для перевірки особи.

У випадку непередбачених або підозрілих активностей користувач може швидко змінити або відключити другий фактор аутентифікації, що заблокує зловмиснику доступ. Це дає можливість вчасно реагувати на можливі загрози безпеки [32].

2FA також зменшує вплив витоків паролів, оскільки навіть у випадку витоку бази даних, зловмисники не матимуть повного доступу до облікових записів без другого фактора аутентифікації.

Важливо відзначити, що 2FA може використовувати різні форми ідентифікації, такі як відбитки пальців, розпізнавання обличчя, одноразові паролі, що робить процес аутентифікації ще більш надійним та відмінним. Загалом, 2FA є ефективним інструментом для забезпечення безпеки ресурсу, враховуючи його здатність ускладнювати процеси аутентифікації та забезпечувати додатковий рівень захисту.

Для впровадження буде використовуватися двоетапна аутентифікація з використанням Time-Based One-Time Password (TOTP) є ефективним засобом забезпечення додаткового рівня безпеки під час входу в систему чи доступу до облікового запису. У цьому методі користувач реєструється із сервером,

обмінюючись унікальним спільним секретним ключем. Після цього, за допомогою спеціального пристрою або мобільного додатку, генерується одноразовий пароль (TOTP) на основі поточного часу та спільного ключа.

Особливістю є те, що цей пароль діє протягом обмеженого часового проміжку, забезпечуючи використання його лише протягом короткого періоду. При вході в систему користувач вводить свої звичайні облікові дані (логін і пароль) разом із відповідним одноразовим паролем, що генерується його пристроєм.

Цей підхід додає додатковий рівень безпеки, оскільки для успішного входу потрібно не лише знати постійні облікові дані, але й мати фізичний доступ до пристрою, на якому генерується одноразовий пароль. 2FA з TOTP є інтегрованим та зручним методом, адаптованим для використання з різними мобільними додатками.

Також в розроблений ресурс буде впроваджено захист AAA (Authentication, Authorization, and Accounting), який охоплює комплекс заходів, спрямованих на забезпечення безпеки та ефективного управління доступом до інформаційних ресурсів. Кожна з компонент AAA відіграє ключову роль в цьому контексті.

Аутентифікація визначає особу, яка намагається отримати доступ. Це може включати введення логіна та пароля, використання біометричних даних, токенів та інших методів. Головна мета — запобігання несанкціонованому доступу та підробці облікових даних [35].

Авторизація визначає права доступу користувача після успішної аутентифікації. Це містить визначення того, які ресурси та дії користувач може використовувати. Принцип найменшого доступу стверджує, що користувачеві надаються лише ті права доступу, які необхідні для його обов'язків.

Облік включає запис інформації про дії користувачів, такі як вхід, вихід, зміни прав доступу. Зібрана інформація може використовуватися для аудиту та виявлення можливих загроз безпеці.

Отже, розробимо схему захисту розробленого ресурсу від несанкціонованого доступу, яка буде містити розглянуті вище технології (рис. 2.7).

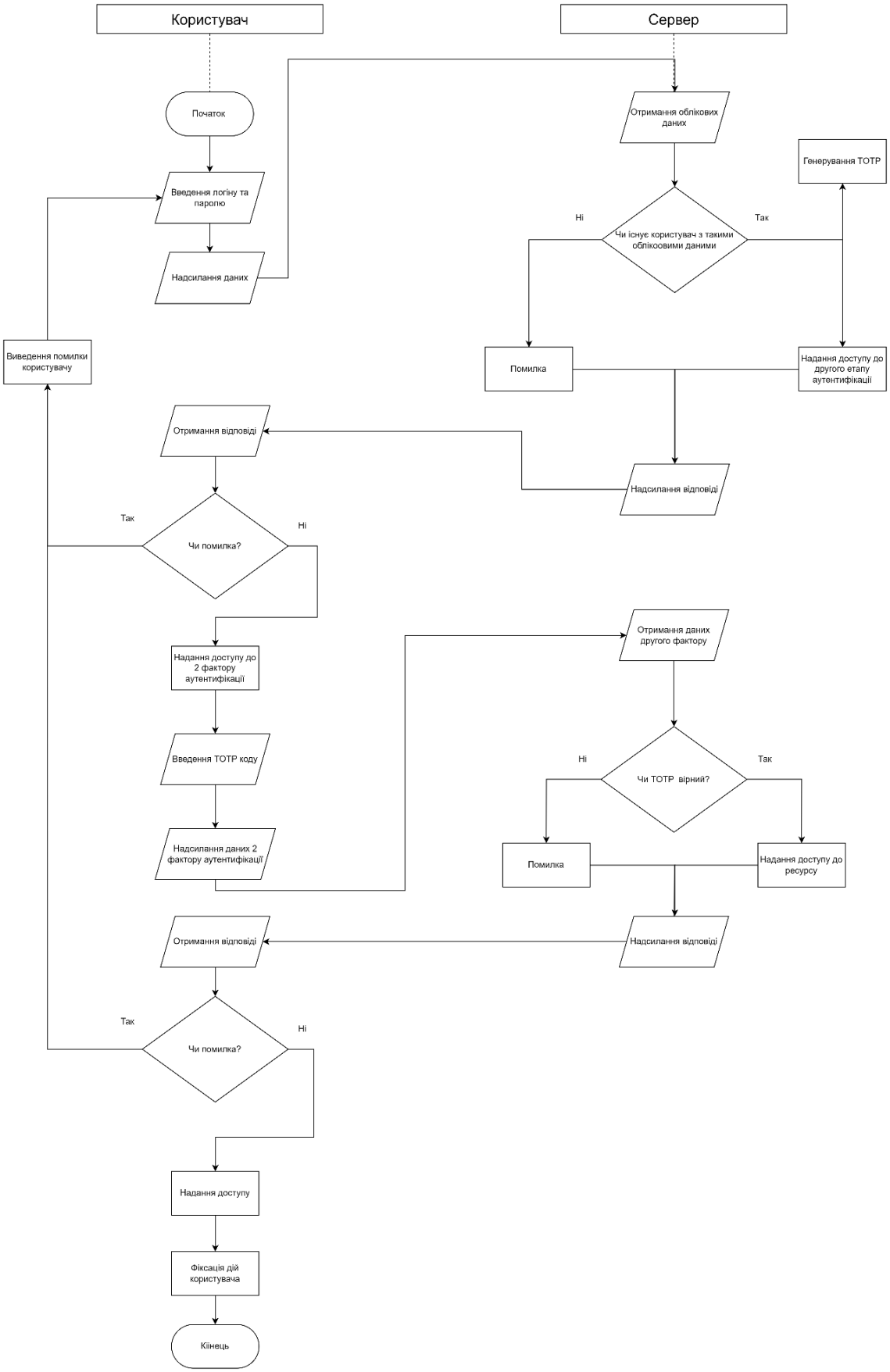


Рисунок 2.7 - Алгоритм захисту від несанкціонованого доступу на основі AAA та TOTP

Розглянемо алгоритм від несанкціонованого доступу на основі AAA та TOTP більш детально по кроках.

Крок 1. Початок.

Початок алгоритму відбувається на стороні користувача коли він переходить на сторінку входу в систему.

Крок 2. Введення логіну та паролю.

На даному кроці користувач вводить логін та пароль від свого облікового запису на стороні користувача.

Крок 3. Надсилання даних.

Користувач надсилає свої облікові дані на сервер.

Крок 4. Отримання облікових даних.

Серверна частина приймає запит з обліковими даними користувача.

Крок 5. Чи існує користувач з такими обліковими даними?

Серверна частина перевіряє чи існує користувач з такими обліковими даними.

Якщо так:

Крок 5.1 Генерування TOTP

Система генерує TOTP код, тобто одноразовий код для другого етапу аутентифікації користувача.

Крок 5.2 Надання доступу до другого етапу аутентифікації

Сервер надає доступ до другого етапу аутентифікації користувачу.

Якщо ні:

Крок 5.3 Помилка

Сервер створює помилку через те що такого користувача не існує.

Крок 6. Надсилання відповіді.

Сервер надсилає на сторону користувача відповідь про надання доступу до наступного етапу аутентифікації або про помилку.

Крок 7. Отримання відповіді.

Сторона користувача отримує відповідь від сервера.

Крок 8. Чи помилка?

Сторона користувача перевіряє відповідь, яка прийшла з сервера.

Якщо так:

Крок 8.1. Виведення помилки користувачу.

У випадку, якщо введені користувачем облікові дані неправильні або користувача з такими обліковими даними не існує, система надсилає помилку.

Користувач отримує сповіщення про помилку та перенаправляється на крок 2 для повторного введення облікових даних.

Цей цикл повторюється до тих пір, поки користувач не надасть правильні облікові дані або не вирішить припинити спроби входу.

Якщо ні:

Крок 9. Надання доступу до другого фактору аутентифікації.

Система надає користувачу доступ до другого фактору аутентифікації.

Крок 10. Введення TOTP коду.

Користувач має ввести TOTP код.

Крок 11. Надсилання даних 2 фактору аутентифікації.

Сторона користувача надсилає TOTP код на серверну частину.

Крок 12. Отримання даних другого фактору.

Серверна частина приймає запит з TOTP кодом від користувача

Крок 13. Чи TOTP вірний?

Перевірка чи користувач ввів вірний TOTP код.

Якщо так:

Крок 13.1. Надання доступу до ресурсу

Сервер надає користувачу доступ до ресурсу

Якщо ні:

Крок 13.2. Помилка

Сервер не надає користувачу доступ до ресурсу і створює помилку

Крок 14. Надсилання відповіді

Сервер надсилає користувачу відповідь з наданням доступу або помилкою.

Крок 15. Отримання відповіді

Користувацька частина отримує відповідь від сервера.

Крок 16. Чи помилка?

Користувацька частина перевіряє чи отримана відповідь містить помилку.

Якщо так:

Повернення користувача до кроку 8.1.

Якщо ні:

Крок 17. Надання доступу

Користувачу надається доступ до ресурсу

Крок 18. Фіксація дій користувача.

Всі дії, які виконуються користувачем фіксуються.

Крок 19. Кінець

Кінець алгоритму.

Отже, можна сформулювати висновок, що розроблена схема захисту, заснована на використанні спільних принципів AAA та 2FA, встановлює високий стандарт безпеки, гарантуючи ефективний захист від різноманітних загроз, таких як атаки на паролі, фішинг та підміни сесій.

Цей підхід також відповідає вимогам до відповідності та законодавчих норм щодо захисту конфіденційної інформації.

Впроваджений метод захисту від несанкціонованого доступу, з акцентом на використанні AAA та 2FA, виявляється ключовим елементом, що сприяє створенню довіри до системи та забезпеченню високої надійності захисту в умовах постійно зростаючих кіберзагроз.

Цей підхід дозволяє ефективно впоратися із викликами сучасного кіберпростору, а також враховує вимоги до безпеки, які накладаються законодавством та стандартами конфіденційності даних.

2.7 Висновки до розділу.

У процесі розширеного аналізу та розробки інформаційного ресурсу для системного аналізу безпеки енергетичної інфраструктури виявлено, що цей комплекс вимагає високоекспертних підходів та урахування специфічних особливостей сфери енергетики.

Належна розробка ключових компонентів, таких як створення бази даних та захист від несанкціонованого доступу, визначає успішне та ефективне функціонування інформаційного ресурсу.

Створення консолідованої бази даних виявилося нетривіальним завданням, оскільки вимагало детальної уваги до деталей та визначення структури, яка враховує всі необхідні аспекти аналізу безпеки в енергетичній галузі.

В результаті аналізу та визначення вимог до бази даних була створена концептуальна основа, яка не лише враховує специфіку індустрії, але й забезпечує зручний та ефективний доступ до інформації.

Забезпечення захисту розробленого ресурсу від несанкціонованого доступу стало необхідною частиною процесу розробки. Використання принципів AAA (Authentication, Authorization, and Accounting) та двоетапної аутентифікації (2FA) стало ключовим для запобігання можливим загрозам та збереження конфіденційності інформації [36].

Узагальнюючи, розглянуті аспекти розробки інформаційного ресурсу підкреслюють важливість комплексного підходу та високого рівня експертизи для забезпечення надійності та ефективності системи.

Аналіз інформаційної безпеки, створення бази даних та застосування сучасних засобів захисту є критичними для успішної розробки та реалізації інформаційного ресурсу в енергетичній сфері.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАХИЩЕНОГО КОНСОЛІДОВАНОГО ІНФОРМАЦІЙНОГО РЕСУРСУ ТА СИСТЕМНИЙ АНАЛІЗ БЕЗПЕКИ ФІНАНСОВОЇ ІНФРАСТРУКТУРИ

В даному розділі буде здійснено обґрунтування вибору СУБД та мови програмування, після чого буде здійснена реалізація бази даних, модулів захисту та системи оцінювання безпеки інфраструктури, після чого буде здійснений висновок по виконаній роботі.

3.1 Обґрунтування вибору СУБД

Для забезпечення безпеки енергетичної інфраструктури необхідно мати ефективну систему аналізу безпеки. Така система повинна дозволяти збирати, зберігати та аналізувати інформацію про потенційні загрози та вразливості енергетичної інфраструктури.

Одним із ключових компонентів системи аналізу безпеки є база даних (БД). БД повинна бути надійною, доступною та захищеною.

СУБД (система управління базами даних) - це програмне забезпечення, яке дозволяє зберігати, керувати та отримувати доступ до даних. СУБД забезпечує структурування даних, а також забезпечує безпеку та цілісність даних.

СУБД використовується в багатьох різних сферах, включаючи бізнес, уряд, освіту та медицину. Наприклад, СУБД використовуються для зберігання інформації про клієнтів, продукти, продажі, фінанси та багато іншого.

Існує багато різних типів СУБД, кожен з яких має свої переваги та недоліки. Деякі поширені типи СУБД включають:

– реляційна СУБД;

Цей тип СУБД є найпоширенішим і використовується в більшості бізнес-додатків. Реляційні СУБД зберігають дані у вигляді таблиць, які пов'язані один з одним за допомогою ключів.

– нереляційна СУБД;

Цей тип СУБД не використовує реляційну модель даних. Нереляційні СУБД часто використовуються для зберігання великих обсягів даних, таких як дані аналітики або дані машинного навчання.

– СУБД в пам’яті.

Цей тип СУБД зберігає дані в оперативній пам'яті, що може значно підвищити продуктивність. СУБД в пам’яті часто використовуються для транзакційних додатків, таких як електронна комерція або фінансові системи.

СУБД, яка використовується для розробки захищеного консолідованого інформаційного ресурсу системного аналізу безпеки енергетичної інфраструктури, повинна відповідати наступним вимогам:

– безпека. СУБД повинна забезпечувати високий рівень безпеки даних, що зберігаються в ній. Вона повинна мати вбудовані засоби захисту від несанкціонованого доступу, модифікації та знищення даних;

– здатність до масштабування. СУБД повинна забезпечувати можливість масштабування в залежності від зростання обсягу даних, що зберігаються в ній;

– стійкість до відмов. СУБД повинна забезпечувати стійкість до відмов, щоб гарантувати безперервну роботу системи системного аналізу безпеки;

– легкість використання. СУБД повинна бути легкою у використанні, щоб забезпечити доступ до даних для користувачів системи системного аналізу безпеки.

Деякі з найбільш популярних СУБД включають:

– Microsoft SQL Server – корпоративна СУБД, розроблена компанією Microsoft;

– Oracle Database – корпоративна СУБД, розроблена компанією Oracle;

– MySQL - відкрита СУБД, розроблена компанією Oracle;

– PostgreSQL - відкрита СУБД, розроблена спільнотою розробників;

– MongoDB - документна СУБД, розроблена компанією MongoDB.

PostgreSQL є відкритою СУБД, яка відповідає всім вимогам, що висуваються до СУБД для розробки захищеного консолідованого інформаційного ресурсу системного аналізу безпеки енергетичної інфраструктури.

PostgreSQL має вбудовані засоби захисту від несанкціонованого доступу, модифікації та знищення даних. Наприклад, PostgreSQL підтримує такі механізми безпеки, як:

– ролі та привілеї. Дозволяють контролювати доступ користувачів до даних;

– механізм шифрування. Дозволяє шифрувати дані, що зберігаються в СУБД;

– відстеження змін. Дозволяє відстежувати зміни, внесені до даних.

PostgreSQL також забезпечує можливість масштабування в залежності від зростання обсягу даних, що зберігаються в ній. PostgreSQL підтримує такі механізми масштабування, як:

– паралельні операції. Дозволяють виконувати операції з базою даних паралельно, що може значно підвищити продуктивність;

– розподілені бази даних. Дозволяють зберігати дані в декількох серверах, що може забезпечити більшу масштабованість і стійкість до відмов.

PostgreSQL є стійкою до відмов СУБД. PostgreSQL підтримує такі механізми стійкості до відмов, як:

– автоматична реплікація. Дозволяє автоматично реплікувати дані на кілька серверів, що забезпечує безперервність роботи системи в разі відмови одного з серверів.

– функція завантаження/збереження. Дозволяє зберігати та відновлювати базу даних, що може бути використано для відновлення системи в разі відмови.

PostgreSQL є легкою у використанні СУБД. PostgreSQL має простий і інтуїтивно зрозумілий інтерфейс користувача, що дозволяє користувачам системи системного аналізу безпеки легко отримувати доступ до даних.

Крім того, PostgreSQL є відкритою СУБД, що означає, що вона безплатна і доступна для використання будь-ким. Це може бути важливим фактором для деяких організацій, які прагнуть заощадити кошти або забезпечити прозорість своїх систем.

Інші СУБД, які можна розглядати для цієї задачі, включають Microsoft SQL Server і Oracle Database. Однак ці СУБД є власним програмним забезпеченням, що означає, що вони платні. Крім того, вони можуть бути менш гнучкими та масштабованими, ніж PostgreSQL.

PostgreSQL є відкритою СУБД, яка відповідає всім технічним вимогам, що висуваються до СУБД для розробки захищеного консолідованого інформаційного ресурсу системного аналізу безпеки енергетичної інфраструктури.

PostgreSQL забезпечує високий рівень сумісності зі стандартами SQL, що дозволяє ефективно використовувати стандартні SQL-запити та оптимізує взаємодію з базою даних. Це важливо для консолідованого інформаційного ресурсу, оскільки він повинен підтримувати різні джерела даних, які можуть використовувати різні версії SQL.

PostgreSQL дозволяє використовувати складні типи даних, що дозволяє ефективно моделювати та зберігати різнорідну інформацію, що є важливим для консолідованого інформаційного ресурсу. Наприклад, PostgreSQL підтримує географічні типи даних, які можуть бути використані для зберігання інформації про розташування об'єктів енергетичної інфраструктури.

PostgreSQL має географічні розширення, що дозволяють ефективно обробляти геопросторові дані. Це може бути важливим для аналізу енергетичної інфраструктури, оскільки вона часто має географічний характер. Наприклад, PostgreSQL можна використовувати для аналізу даних про розташування об'єктів енергетичної інфраструктури, щоб виявити потенційні загрози.

PostgreSQL підтримує різноманітні функціональності для аналізу даних, включаючи вбудовані агрегації, віконні функції та інші. Це забезпечує потужні можливості для системного аналізу безпеки. Наприклад, PostgreSQL можна використовувати для аналізу даних про історію збоїв енергетичної

інфраструктури, щоб виявити закономірності та тенденції, які можуть вказувати на потенційні загрози.

PostgreSQL дозволяє використовувати тригери та процедури. Це дозволяє реалізувати автоматичні перевірки безпеки та реакції на події в реальному часі. Наприклад, можна використовувати тригери для перевірки даних на наявність несанкціонованих змін [39].

PostgreSQL забезпечує різноманітні механізми автентифікації та авторизації, що дозволяє строго контролювати доступ до даних та системних ресурсів. Це важливо для забезпечення безпеки даних енергетичної інфраструктури.

PostgreSQL підтримує SSL-захист, який забезпечує безпеку транзакцій та з'єднань. Це критично важливо для конфіденційності інформації про енергетичну інфраструктуру.

PostgreSQL має можливості аудиту, які дозволяють створювати деталізований журнал подій. Це може бути використано для виявлення потенційних загроз.

PostgreSQL відповідає всім вимогам, що висуваються до СУБД для розробки захищеного консолідованого інформаційного ресурсу системного аналізу безпеки енергетичної інфраструктури. Вона має високі технічні характеристики, потужні функціональні можливості та ефективні безпекові механізми.

3.2 Обґрунтування вибору мови програмування

JavaScript (JS) - це мова програмування, яка використовується для створення інтерактивних вебдодатків. Вона є однією з найпопулярніших мов програмування у світі, і її використовують мільйони розробників.

JavaScript є об'єктноорієнтованою мовою програмування, що означає, що вона використовує об'єкти для організації даних та коду. JavaScript також є

динамічною мовою програмування, що означає, що типи даних визначаються під час виконання.

Мова програмування, яка використовується для розробки захищеного консолідованого інформаційного ресурсу системного аналізу безпеки енергетичної інфраструктури, повинна відповідати наступним вимогам:

– безпека;

Мова програмування повинна мати вбудовані механізми безпеки, які захищають від несанкціонованого доступу, модифікації та знищення даних.

– здатність до масштабування;

Мова програмування повинна забезпечувати можливість масштабування в залежності від зростання обсягу даних, що зберігаються в ній.

– легкість використання.

Мова програмування повинна бути легкою у використанні, щоб забезпечити доступ до даних для користувачів системи системного аналізу безпеки.

JavaScript відповідає всім вимогам, що висуваються до мови програмування для розробки захищеного консолідованого інформаційного ресурсу системного аналізу безпеки енергетичної інфраструктури.

JavaScript має вбудовані механізми безпеки, які захищають від несанкціонованого доступу, модифікації та знищення даних. Наприклад, JavaScript підтримує такі механізми безпеки, як:

– операції безпеки;

JavaScript підтримує операції безпеки, такі як typeof, instanceof та in, які можуть використовуватися для перевірки безпеки даних.

– функції безпеки;

JavaScript підтримує функції безпеки, такі як eval() та setTimeout(), які можуть використовуватися з обережністю, щоб уникнути потенційних проблем безпеки.

– стандарти безпеки.

JavaScript підтримує стандарти безпеки, такі як DOM Level 2 Security та HTML5 Security, які забезпечують додаткові засоби захисту від несанкціонованого доступу, модифікації та знищення даних.

JavaScript є динамічною мовою програмування, що дозволяє легко масштабувати вебдодатки. Наприклад, JavaScript підтримує такі механізми масштабування, як:

– асинхронні операції;

JavaScript підтримує асинхронні операції, які дозволяють виконувати кілька операцій одночасно.

– середовище node.js.

Node.js - це середовище виконання, яке дозволяє використовувати JavaScript для створення серверних застосунків. Node.js забезпечує ефективну масштабованість для серверних застосунків, написаних на JavaScript.

JavaScript є популярною мовою програмування, яка має велике співтовариство розробників. JavaScript також має простий синтаксис, що полегшує його вивчення та використання.

jQuery - це фреймворк JavaScript, який спрощує розробку вебдодатків. Він надає широкий спектр функцій, які спрощують взаємодію з Document Object Model (DOM) і анімацію. Він також надає ряд додаткових функцій, таких як обробка подій, обробка даних та Ajax.

jQuery також має ряд функцій безпеки, які можуть використовуватися для захисту даних. Наприклад, jQuery підтримує такі функції безпеки, як:

– відстеження змін;

jQuery підтримує відстеження змін у даних, що може використовуватися для виявлення несанкціонованих змін;

– захист від XSS;

jQuery підтримує захист від XSS-атаків, який може використовуватися для захисту від несанкціонованого введення даних;

– захист від CSRF.

jQuery підтримує захист від CSRF-атаків, який може використовуватися для захисту від несанкціонованого доступу до даних.

jQuery має ряд переваг, які роблять його популярним фреймворком JavaScript. До них відносяться:

– простота;

jQuery має простий синтаксис, що полегшує його вивчення та використання.

– відкритість;

jQuery є відкритим фреймворком JavaScript, що означає, що він безплатний для використання та модифікації.

– модульність;

jQuery є модульним фреймворком JavaScript, що дозволяє легко додавати нові функції та можливості.

– підтримка;

jQuery має велике співтовариство розробників, які підтримують і розвивають фреймворк.

jQuery є потужним інструментом для розробки вебдодатків. Він має ряд переваг, які роблять його популярним вибором для розробників вебдодатків.

Express - це мікрофреймворк Node.js, який використовується для створення вебдодатків на основі Node.js. Він забезпечує базову структуру для вебдодатків, включаючи маршрутизацію, обробку запитів та відповідей, а також управління станом.

Express є популярним вибором для розробки вебдодатків, оскільки він простий у використанні та забезпечує широкий спектр функцій. Він також добре масштабується, що робить його придатним для великих вебдодатків.

Express є хорошим вибором для розробки захищеного консолідованого інформаційного ресурсу системного аналізу безпеки енергетичної інфраструктури. Він має такі переваги, які роблять його хорошим вибором для цього завдання:

– безпека;

Express підтримує ряд функцій безпеки, які можуть використовуватися для захисту даних.

– здатність до масштабування;

Express має ряд функцій, які сприяють масштабуванню вебдодатків.

– легкість використання.

Express є відносно простим у використанні фреймворком. Це може бути важливою перевагою для розробників, які не мають досвіду роботи з Node.js.

Express також має ряд функцій безпеки, які можуть використовуватися для захисту даних. Наприклад, Express підтримує такі функції безпеки, як:

– захист від атак методом переповнення буфера. Express підтримує захист від атак методом переповнення буфера, який може використовуватися для захисту від несанкціонованого втручання в код;

– захист від атак методом SQL-ін'єкції. Express підтримує захист від атак методом SQL-ін'єкції, який може використовуватися для захисту від несанкціонованого доступу до даних.

JavaScript з використанням jQuery, Express є хорошим вибором для розробки захищеного консолідованого інформаційного ресурсу системного аналізу безпеки енергетичної інфраструктури. Вони мають високі технічні характеристики, потужні функціональні можливості та ефективні безпекові механізми.

3.3 Реалізація бази даних інформаційного ресурсу

Реалізація бази даних для інформаційного ресурсу є важливим етапом у розробці системи.

Для розробки була вибрана система управління базою даних Postgres. Розробимо базу даних за допомогою бібліотеки Express js та бібліотеки Sequelize.

Для початку реалізуємо підключення потрібних модулів та підключення бази даних.

```
const express = require('express');
const { Sequelize, DataTypes } = require('sequelize');
```

```
const sequelize = new Sequelize({
 dialect: 'postgres',
 host: '127.0.0.1',
 username: 'your_username',
 password: 'your_password',
 database: 'your_database',});
```

Наступним кроком оголосимо модель користувачів.

```
const User = sequelize.define('User', {
 id: {
  type: DataTypes.INTEGER,
  primaryKey: true,
  autoIncrement: true,  },
 name: {
  type: DataTypes.STRING, },
 email: {
  type: DataTypes.STRING,
  unique: true,  },
 password: {
  type: DataTypes.STRING,  },});
```

В результаті отримаємо таблицю з користувачами, яку заповнимо даними (рис. 3.1).

	id [PK] integer	email character varying	password character vary	created_at timestamp without ti	updated_at timestamp without time	userName character varying
1	8	asasddassf@gmail.com	$2a$05$Lk...	2023-10-28 02:25...	2023-10-28 02:25:0...	Vova
2	9	asasdd1assf@gmail.com	$2a$05$c...	2023-10-28 02:38...	2023-10-28 02:38:2...	Andrey
3	10	asas1dd1assf@gmail.com	$2a$05$A...	2023-10-28 02:38...	2023-10-28 02:38:2...	Kola
4	11	asas11dd1assf@gmail.com	$2a$05$y8...	2023-10-28 02:38...	2023-10-28 02:38:2...	Oleg
5	12	a1sas11dd1assf@gmail.com	$2a$05$x...	2023-10-28 02:38...	2023-10-28 02:38:2...	Vika
6	13	a1sas111dd1assf@gmail.com	$2a$05$n...	2023-10-28 02:38...	2023-10-28 02:38:3...	Igor

Рисунок 3.1 – Заповнена таблиця користувачі

Далі оголосимо модель аналізів.

```
const Analysis = sequelize.define('Analysis', {
 id: {
  type: DataTypes.INTEGER,
  primaryKey: true,
  autoIncrement: true,  },
 analysisName: {
  type: DataTypes.STRING,  },
 analysisResult: {
  type: DataTypes.STRING,  },
 analysisDate: {
  type: DataTypes.DATE,  },});
```

В результаті отримаємо таблицю з результатами аналізу безпеки(рис. 3.2).

	analysisName character varying	analysisResult character varying	analysisDate date	userId bigint	id_system bigint	id [PK] integer
1	Аналіз безпе...	Були виявлені наступні недолі...	2023-11-23	1	1	1
2	Аналіз безпе...	Були виявлені наступні недолі...	2023-11-23	1	1	2
3	Аналіз безпе...	Були виявлені наступні недолі...	2023-11-23	3	2	3
4	Аналіз безпе...	Були виявлені наступні недолі...	2023-11-23	1	1	4

Рисунок 3.2 – Заповнена таблиця результати аналізу безпеки

Далі оголосимо модель де буде зберігатися інформація про енергетичні системи.

```
const EnergySystem = sequelize.define('EnergySystem', {
 id: {
  type: DataTypes.INTEGER,
  primaryKey: true,
  autoIncrement: true,  },
 systemName: {
  type: DataTypes.STRING,  },
 systemDescription: {
  type: DataTypes.STRING,  },
 systemTechnicalSpecification: {
  type: DataTypes.STRING,  },});
```

В результаті отримаємо таблицю з енергетичними інфраструктурами (рис. 3.3).

	id [PK] integer	systemName character var	systemDescription character varying	systemTechnicalSpecif character varying
1	1	Вінницька	Віницька енергет...	Віницька енергети...
2	2	Чернігівс...	Чернігівська ене...	Чернігівська енер...
3	3	Львівська	Львівська енерге...	Львівська енергет...
4	4	Київська	Київська енергет...	Київська енергети...

Рисунок 3.3 – Заповнена таблиця енергетичних інфраструктур

Далі оголосимо модель з енергетичними об’єктами.

```
const EnergyObjects = sequelize.define('EnergyObjects', {
 id: {
  type: DataTypes.INTEGER,
  primaryKey: true,
  autoIncrement: true,  },
 objectName: {
  type: DataTypes.STRING,  },
 objectType: {
  type: DataTypes.STRING,  },
 objectLocation: {
  type: DataTypes.STRING,  },});
```

В результаті отримаємо таблицю з об’єктами енергетичної інфраструктури (рис. 3.4).

	id [PK] integer	objectName character vary	objectType character varying	objectLocation character varyin
1	1	Підстанція	Електростанція	Віннцька обл
2	2	Електрос...	Електростанція	Віннцька обл
3	3	Гідроеле...	Електростанція	Віннцька обл
4	4	Теплоеле...	Електростанція	Віннцька обл

Рисунок 3.4 – Заповнена таблиця об’єктів інфраструктури

Наступним кроком реалізуємо сутність з засобами забезпечення захисту об’єкта інфраструктури.

```
const SecurityMeasures = sequelize.define('SecurityMeasures', {
id: {
  type: DataTypes.INTEGER,
  primaryKey: true,
  autoIncrement: true,  },
measuresDescription: {
  type: DataTypes.STRING,  },
measuresImplementationDate: {
  type: DataTypes.DATE,  },
measuresType: {
  type: DataTypes.STRING,  },});
```

Також реалізуємо зв’язки між всіма сутностями бази даних.

```
SecurityMeasures.belongsTo(EnergyObjects);
EnergyObjects.belongsTo(EnergySystem);
EnergyObjects.hasMany(SecurityMeasures);
EnergySystem.hasMany(Analysis);
EnergySystem.hasMany(EnergyObjects);
Analysis.belongsTo(User);
Analysis.belongsTo(EnergySystem);
```

Далі реалізуємо функції, завдяки яким можна буде діставати інформацію з сутностей бази даних для подальшої роботи з нею.

```
app.get('/analyses', async (req, res) => {
 try {
  const analyses = await Analysis.findAll({
    include: [User, EnergySystem], });
  res.json(analyses);
 } catch (error) {
  console.error(error);
  res.status(500).json({ error: 'Internal Server Error' });  }});
app.get('/users', async (req, res) => {
 try {
  const users = await User.findAll();
  res.json(users);
 } catch (error) {
  console.error(error);
  res.status(500).json({ error: 'Internal Server Error' });  }});
```

Отже, можна зробити висновок, що практична реалізація бази даних пройшла успішно, що дозволяє перейти до наступних етапів розробки.

3.4 Реалізація запитів та звітів

Запит в базі даних є конструкцією або командою, яка використовується для взаємодії з базою даних, зокрема для отримання, вставлення, оновлення чи видалення даних. SQL, або мова структурованих запитів, є засобом виразного вираження таких запитів. Звіт в базі даних натомість, є візуалізованою формою

представлення даних з метою полегшення їх розуміння та аналізу. Звіти можуть бути операційними, стратегічними чи тактичними, залежно від цілей вивчення інформації.

Використання SQL для створення запитів та звітів має свої переваги. SQL надає простий та ефективний спосіб взаємодії з базами даних, виражаючи операції з даними. Ця мова є стандартом для роботи з реляційними базами даних та забезпечує ефективний доступ до даних. Використання SQL також гарантує стандартизацію процесу та сприяє безпеці, оскільки можливе параметризоване використання SQL-запитів для захисту від потенційних атак.

SQL є масштабованим інструментом, який дозволяє працювати як з невеликими, так і з великими обсягами даних. Використання цієї мови спрощує процес аналізу та дозволяє швидко створювати звіти для відображення результатів. Такий підхід дозволяє ефективно використовувати інструменти баз даних та забезпечує зручність управління інформацією.

Для початку реалізуємо запит для отримання інформації про користувачів системи:

```
SELECT * FROM User;
```

Далі реалізуємо запит для отримання інформації про результати аналізів безпеки:

```
SELECT * FROM Analysis;
```

Далі реалізуємо запит для отримання інформації про енергетичні інфраструктури:

```
SELECT * FROM EnergySystem;
```

Далі реалізуємо запит для отримання інформації про об’єкти критичної інфраструктури :

```
SELECT * FROM EnergyObjects;
```

Далі реалізуємо запит для отримання інформації про засоби захисту об’єкта інфраструктури:

```
SELECT * FROM SecurityMeasures;
```

Наступним кроком реалізуємо запит для звіту по рейтингу енергетичних інфраструктур за результатами їх аналізів безпеки:

```
SELECT
  EnergySystem.systemName,
  EnergySystem.systemDescription,
  COUNT(Analysis.id) AS analysisCount
FROM
  EnergySystem
LEFT JOIN
  Analysis ON EnergySystem.id = Analysis.EnergySystemId
GROUP BY
  EnergySystem.id, EnergySystem.systemName, EnergySystem.systemDescription
ORDER BY
  analysisCount DESC;
```

Реалізуємо даний звіт на сторінці користувача (рис. 3.5).

Рисунок 3.5 – Звіт рейтингу енергетичних інфраструктур за результатами їх аналізів безпеки

Далі реалізуємо запит для звіту рейтинг енергетичних інфраструктур за кількістю об’єктів інфраструктури:

```
SELECT
  EnergySystem.systemName,
  EnergySystem.systemDescription,
  COUNT(EnergyObjects.id) AS energyObjectsCount
FROM
  EnergySystem
LEFT JOIN
  EnergyObjects ON EnergySystem.id = EnergyObjects.EnergySystemId
GROUP BY
  EnergySystem.id, EnergySystem.systemName, EnergySystem.systemDescription
ORDER BY
  energyObjectsCount DESC;
```

Реалізуємо даний звіт на сторінці користувача (рис. 3.6).

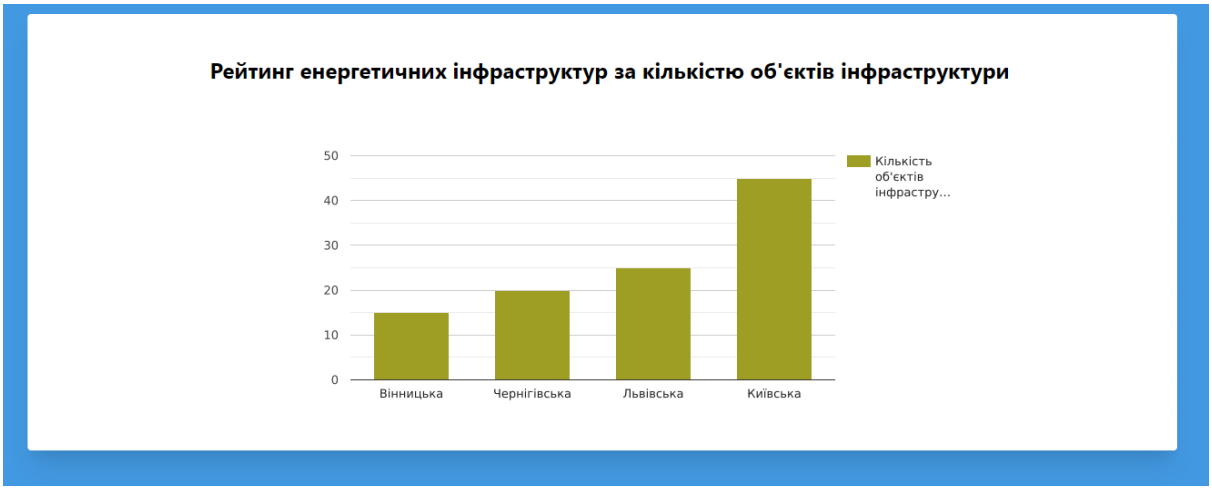


Рисунок 3.6 – Звіт рейтингу енергетичних інфраструктур за кількістю об'єктів інфраструктури

Далі реалізуємо запит для звіту рейтинг об’єктів інфраструктури за кількістю засобів захисту:

```
SELECT
  EnergyObjects.objectName,
  EnergyObjects.objectType,
  COUNT(SecurityMeasures.id) AS securityMeasuresCount
FROM
  EnergyObjects
LEFT JOIN
  SecurityMeasures ON EnergyObjects.id = SecurityMeasures.EnergyObjectId
GROUP BY
  EnergyObjects.id, EnergyObjects.objectName, EnergyObjects.objectType
ORDER BY
  securityMeasuresCount DESC;
```

Реалізуємо даний звіт на сторінці користувача (рис. 3.7).

Рисунок 3.7 – Звіт рейтингу об’єктів інфраструктури за кількістю засобів захисту

Далі реалізуємо запит для звіту рейтинг об’єктів інфраструктури за їх типами:

```
SELECT
  EnergyObjects.objectType,
  COUNT(EnergyObjects.id) AS objectTypeCount
FROM
  EnergyObjects
GROUP BY
  EnergyObjects.objectType
ORDER BY
  objectTypeCount DESC;
```

Реалізуємо даний звіт на сторінці користувача (рис. 3.8).

Рисунок 3.8 – Звіт рейтингу об’єктів інфраструктури за їх типами

Отже, можна зробити висновок, що реалізація звітів та запитів пройшла успішно і користувачі зможуть побачити інформацію в зручному для аналізу вигляді.

3.5 Реалізація програмних модулів забезпечення захисту інформаційного ресурсу

Розробка програмних модулів забезпечення для захисту інформаційного ресурсу є важливим етапом в розробці системи. Заходи безпеки повинні містити захист від несанкціонованого доступу, забезпечення конфіденційності даних, виявлення та реагування на інциденти, а також забезпечення цілісності даних.

Для початку розробимо функцію реєстрації користувача та додавання його в базу даних.

```
router.post('/register', async (req, res) => {
 try {
  const { username, password } = req.body;
  const existingUser = await User.findOne({ where: { username } });
  if (existingUser) {
   return res.status(409).json({ message: 'Користувач з таким іменем вже існує' });    }
  const newUser = await User.create({
   username,
   password,   });
  res.status(201).json({ message: 'Користувач успішно зареєстрований' });
 } catch (error) {
  console.error(error);
  res.status(500).json({ message: 'Помилка сервера' });  }});
module.exports = router;
```

Далі розробимо функціонал авторизації користувача.

```
router.post('/login', async (req, res) => {
 try {
  const { username, password } = req.body;
  const existingUser = await User.findOne({ where: { username } });
  if (!existingUser) {
   return res.status(401).json({ message: 'Користувач не знайдений' });   }
  const isPasswordValid = await bcrypt.compare(password, existingUser.password);
  if (!isPasswordValid) {
   return res.status(401).json({ message: 'Невірний пароль' });    }
  res.status(200).json({ message: 'Авторизація пройшла успішно' });
 } catch (error) {
  console.error(error);
  res.status(500).json({ message: 'Помилка сервера' });  }});
```

Далі розробимо функцію генерації одноразового ключа для двофакторної ідентифікації користувача.

```
  const token = jwt.sign({ userId: existingUser.id }, 'your-secret-key', { expiresIn: '1h' });
  res.status(200).json({ token });
 } catch (error) {
  console.error(error);
  res.status(500).json({ message: 'Помилка сервера' });
 }});
```

Далі реалізуємо функціонал на стороні користувача. Для початку реалізуємо сторінку реєстрації користувача.

```
<div className='w-full flex flex-col justify-between items-center'>
      {isSingUp && (
       <CustomInput
        type="text"    required={true}
        name='userName'
        value={state.userName}   placeholder='Name'
        onChange={onChangeHandler}    )}
      <CustomInput
       type="email"
       required={true}
       name='email'
       value={state.email}
       placeholder='Email'
       onChange={onChangeHandler}
```

Також розробимо форму авторизації користувача.

```
{isSingUp && (
   <CustomInput
    required={true}
    minLength={8}
    name='confirmPassword'
    placeholder='Confirm assword'
    type="password"
    value={state.confirmPassword}
    onChange={onChangeH  </div>
 <div className='flex flex-col justify-between items-center w-full'><button  type="submit"
    className="mt-3  w-full  text-white  bg-gray-800  hover:bg-gray-900  focus:outline-none  focus:ring-4 focus:ring-gray-300  font-medium  rounded-lg  text-sm  px-5  py-2.5  mb-2  dark:bg-gray-800  dark:hover:bg-gray-700 dark:focus:ring-gray-700 dark:border-gray-700">
    {isSingUp ? 'SingUp' : 'SingIn'}
   </button>
```

Також розробимо функціонал другого фактору автентифікації користувача.

```
const CustomInput: React.FC<CustomInputProps> = ({ type, value, onChange, placeholder, required, name, minLength }) => {
 return (
  <input  name={name}  className={inputClases}
   required={required}
   type={type}
   value={value}
   onChange={onChange}
   minLength={minLength}
   placeholder={placeholder}  /> );
```

В результаті розробки були такі сторінки користувача, як сторінка авторизації (рис. 3.9).

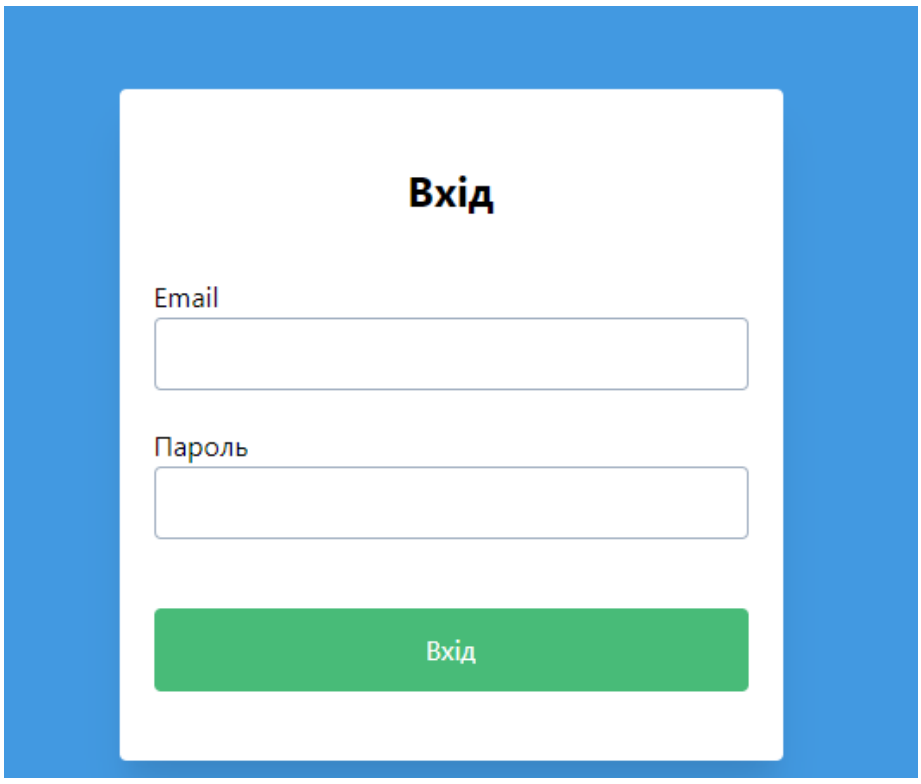


Рисунок 3.9 – Сторінка входу

Також була реалізована сторінка реєстрації користувача (рис. 3.10).

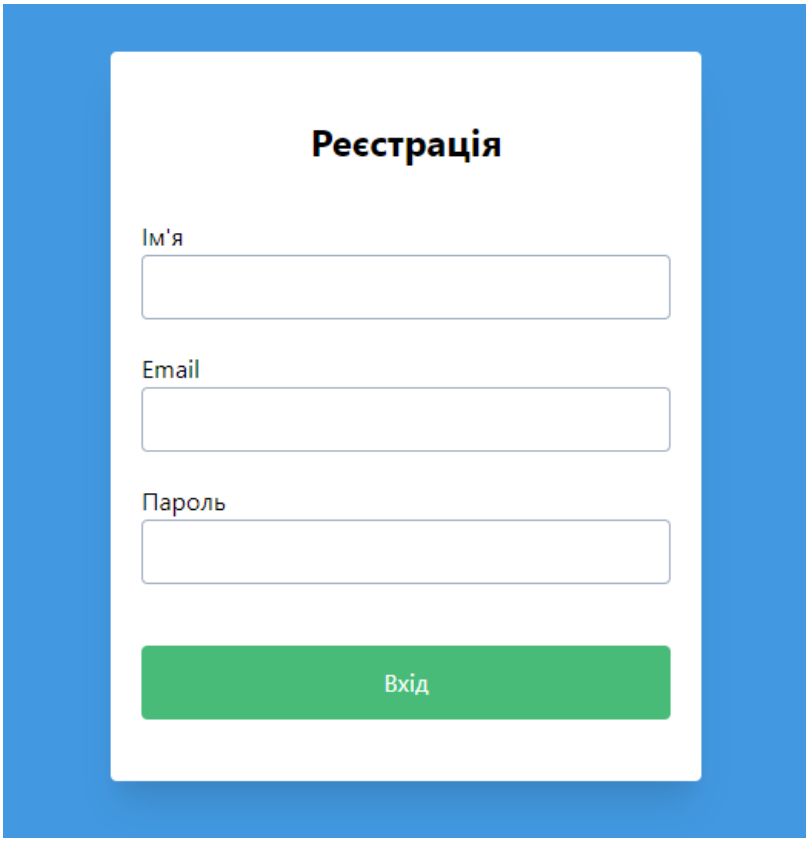


Рисунок 3.10 – Сторінка реєстрації

Та сторінка введення другого фактору авторизації (рис. 3.11).

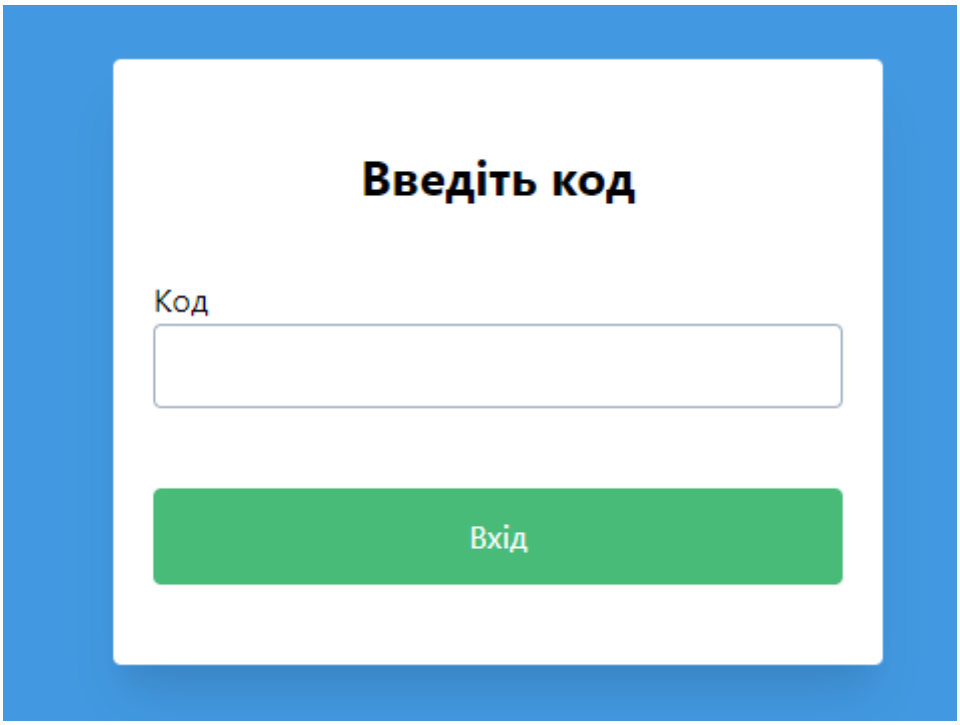


Рисунок 3.11 – Сторінка проходження другого фактору авторизації

Отже, можна зробити висновок, що розробка модуля захисту була здійснена успішно, під час реалізації була створена серверна логіка та логіка на стороні користувача.

3.6 Реалізація системи аналізу безпеки енергетичної інфраструктури регіону

Початковим кроком в реалізації системи аналізу безпеки енергетичної інфраструктури регіону є реалізація логіки роботи системи. Отже, спочатку реалізуємо логіку додавання енергетичної інфраструктури.

```
router.post('/add', async (req, res) => {
 try {
  const { systemName, systemDescription, technicalSpecification } = req.body;
  const newSystem = await EnergySystem.create({
   systemName,
   systemDescription,
   technicalSpecification,   });
  res.status(201).json({ message: 'Енергетична система додана успішно', system: newSystem });
 } catch (error) {
  console.error(error);
  res.status(500).json({ message: 'Помилка сервера' });   }});
```

Далі реалізуємо інтерфейс на сторінці користувача.

```
<div class="container">
  <div class="form-container" id="login-form">
   <h1>Додавання енергетичної інфраструктури</h1>
   <form>
    <label for="password">Назва</label>
    <input type="password" id="password" name="password" required>
      <label for="password">Опис</label>
    <input type="password" id="password" name="password" required>
      <label for="password">Специфікації</label>
    <input type="password" id="password" name="password" required>
    <button type="submit">Додати</button>
   </form>
  </div>
```

В результаті чого була отримана наступна сторінка (рис. 3.12).

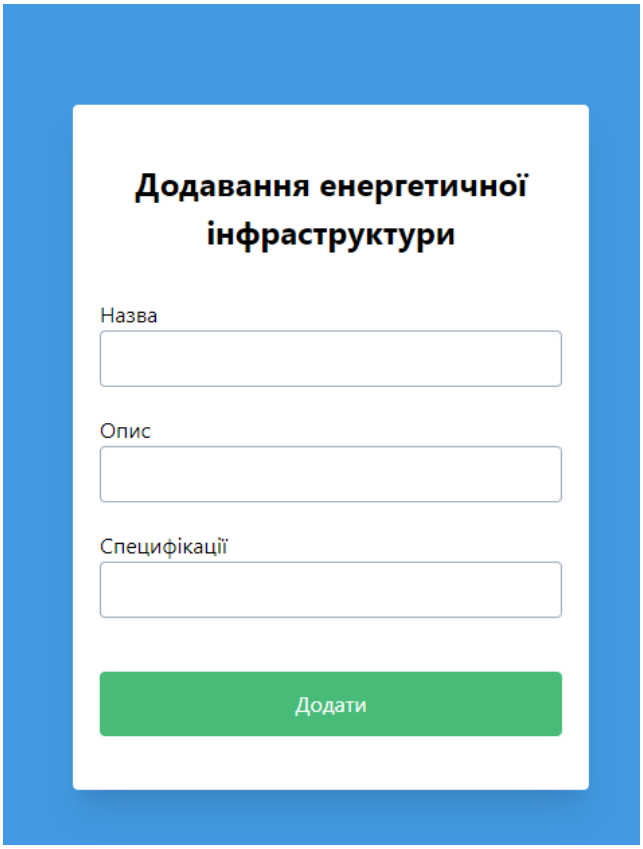


Рисунок 3.12 – Сторінка додавання інфраструктури

Наступним кроком реалізуємо логіку додавання об’єктів інфраструктури.

```
router.post('/add', async (req, res) => {
 try {
   const { objectName, objectType, objectLocation, systemId } = req.body;
   const newObject = await EnergyObjects.create({
     objectName,
     objectType,
     objectLocation,
     systemId,    });
   res.status(201).json({ message: 'Об\'єкт інфраструктури додано успішно', object: newObject });
 } catch (error) {
   console.error(error);
   res.status(500).json({ message: 'Помилка сервера' });  }});
```

Також реалізуємо відповідну сторінку на стороні користувача.

```
<div class="container">
   <div class="form-container" id="login-form"> <h1>Додавання об'єкта інфраструктури</h1>    <form>
      <label for="password">Назва</label>
      <input type="password" id="password" name="password" required>
        <label for="password">Локація</label>
      <input type="password" id="password" name="password" required>
        <label for="password">Тип</label>
      <input type="password" id="password" name="password" required>
      <button type="submit">Додати</button>     </form>    </div>
   <div class="form-container" id="signup-form" style="display: none;">
```

В результаті була отримана наступна сторінка (рис 3.13).

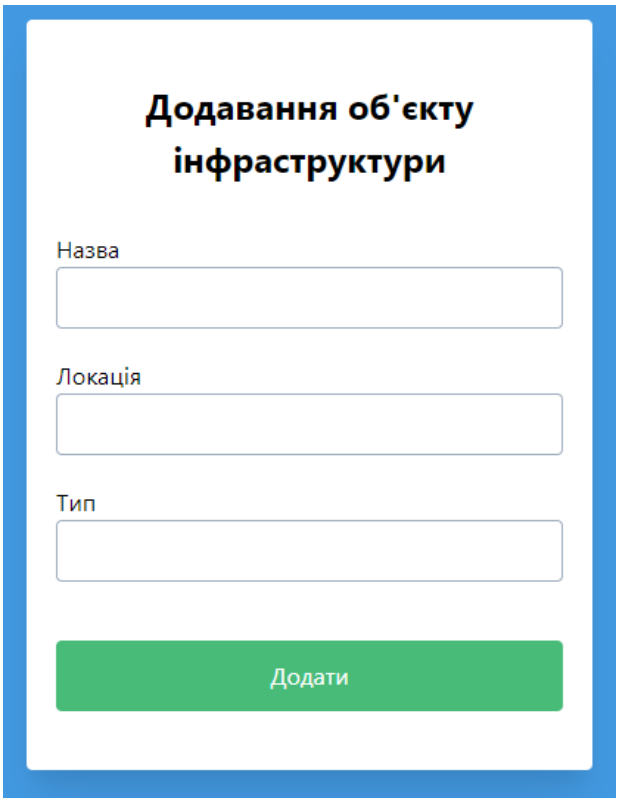


Рисунок 3.13 – Сторінка додавання об’єкта інфраструктури

Далі реалізуємо логіку додавання методів захисту об’єктів інфраструктури.

```
router.post('/add', async (req, res) => {
 try {   const { measuresDescription, implementationDate, measuresType, objectId } = req.body;
   const newMeasure = await SecurityMeasures.create({
     measuresDescription,
     implementationDate,
     measuresType,
     objectId,    });
   res.status(201).json({ message: 'Засіб захисту додано успішно', measure: newMeasure });
 } catch (error) {
   console.error(error);
   res.status(500).json({ message: 'Помилка сервера' });  }});
```

Також реалізуємо відповідну сторінку на стороні користувача.

```
<div class="container">
   <div class="form-container" id="login-form">
     <h1>Додавання засобів захисту</h1>
     <form>
       <label for="password">Назва</label>
       <input type="password" id="password" name="password" required>
         <label for="password">Дата</label>
       <input type="password" id="password" name="password" required>
         <label for="password">Тип</label>
       <input type="password" id="password" name="password" required>
          <label for="password">Опис</label>
       <input type="password" id="password" name="password" required>
       <button type="submit">Додати</button>
     </form>
   </div>
```

В результаті була отримана наступна сторінка (рис 3.14).

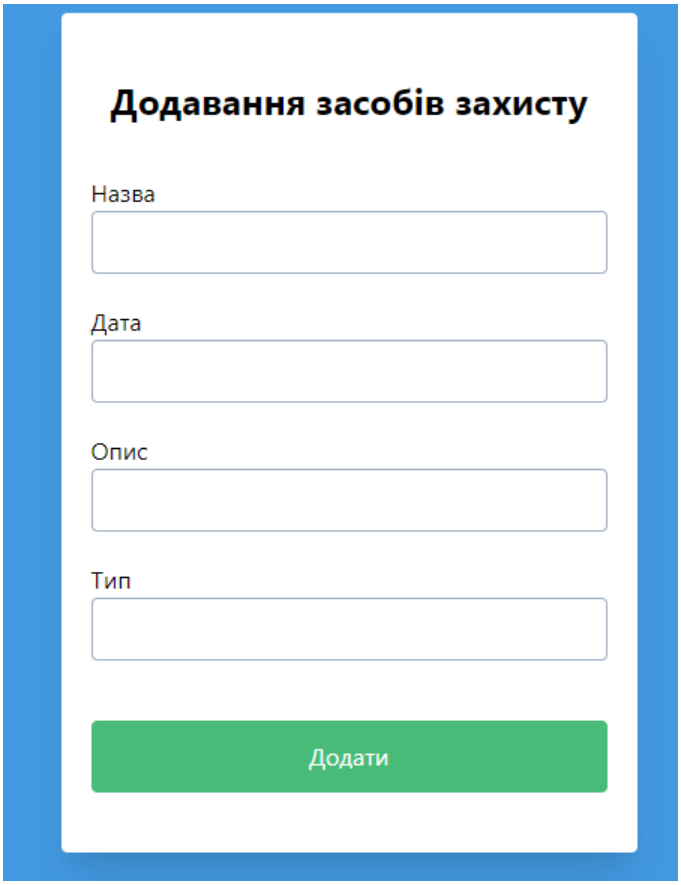


Рисунок 3.14 – Сторінка додавання засобів захисту

Наступним кроком реалізуємо логіку оцінювання безпеки енергетичної інфраструктури регіону. Спочатку реалізуємо логіку проходження питань (рис. 3.7).

```
 const app = express();
app.use(bodyParser.json());
app.get('/questions', (req, res) => {
  res.json(questions);});
app.post('/submit', (req, res) => {
  const userAnswers = req.body.answers;
  let score = 0;
  questions.forEach((question, index) => {
    if (userAnswers[index] === question.correctOption) {
      score++;  } });
  res.json({ score, totalQuestions: questions.length });});
app.listen(PORT, () => {
```

Також реалізуймо відповідну логіку та сторінку на стороні користувача.

```
<script>
    async function fetchQuestions() {
      const response = await fetch('/questions');
      const questions = await response.json();
      return questions   }
    async function displayQuestions() {
      const questions = await fetchQuestions();
```

```
const container = document.getElementById('questions-container');
questions.forEach((question, index) => {
 const questionElement = document.createElement('div');
 questionElement.innerHTML = `
  <p>${index + 1}. ${question.text}</p>  <ul>
   ${question.options.map((option, optionIndex) => `<li><input type="radio" name="q${index}" value="${optionIndex}">${option}</li>`).join('')}
  </ul>  container.appendChild(questionElement);  });
async function submitAnswers() {  const userAnswers = [];
 const answerInputs = document.querySelectorAll('input[type="radio"]:checked');
 answerInputs.forEach(input => {
  userAnswers.push(parseInt(input.value)); });
 const response = await fetch('/submit', {
  method: 'POST',  headers: {  'Content-Type': 'application/json',  },
 body: JSON.stringify({ answers: userAnswers }),  });
 const result = await response.json();
 alert(`Ваш результат: ${result.score}/${result.totalQuestions}`); }
```

В результаті реалізації була отримана наступна сторінка (рис. 3.15).

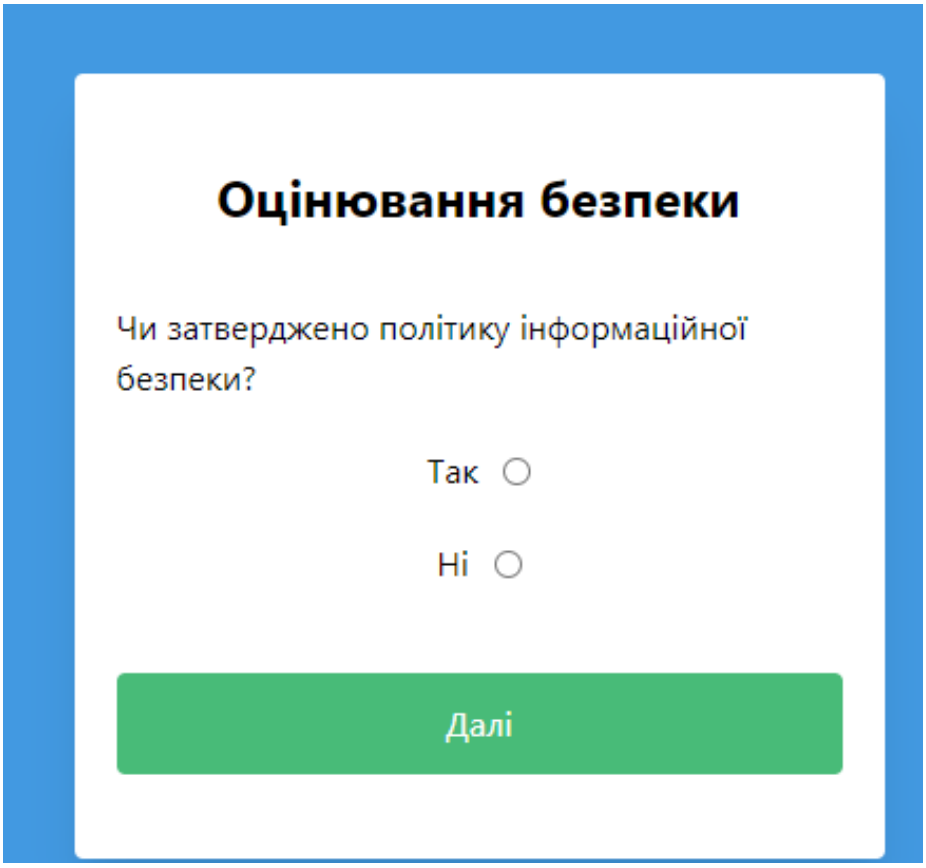


Рисунок 3.15 – Сторінка проходження оцінювання безпеки

Наступним кроком реалізуємо логіку оцінювання та запису результатів в базу даних.

```
app.post('/save-result', async (req, res) => {
 try {
  const { userId, score, totalQuestions } = req.body;
  const result = await TestResult.create({
```

```
    userId,
    score,
    totalQuestions,   });
  res.status(201).json({ message: 'Результат тестування збережено успішно', result });
 } catch (error) {
  console.error(error);
  res.status(500).json({ message: 'Помилка сервера' });  }});
```

Також реалізуємо сторінку виведення результатів.

```
<body>
 <h1>Результати тестування</h1>
 <% if (results.length === 0) { %>  <p>Немає результатів тестування.</p>
 <% } else { %>   <table>    <thead>
     <tr>      <th>ID</th>
       <th>User ID</th>
       <th>Score</th>
       <th>Total Questions</th>  </tr>     </thead>
   <tbody>   <% results.forEach(result => { %>  <tr>
        <td><%= result.id %></td>
        <td><%= result.userId %></td>
        <td><%= result.score %></td>
        <td><%= result.totalQuestions %></td>   </
<% }); %>  </tbody>   </table>  <% } %>
```

В результаті реалізації була отримана наступна сторінка (рис. 3.16).

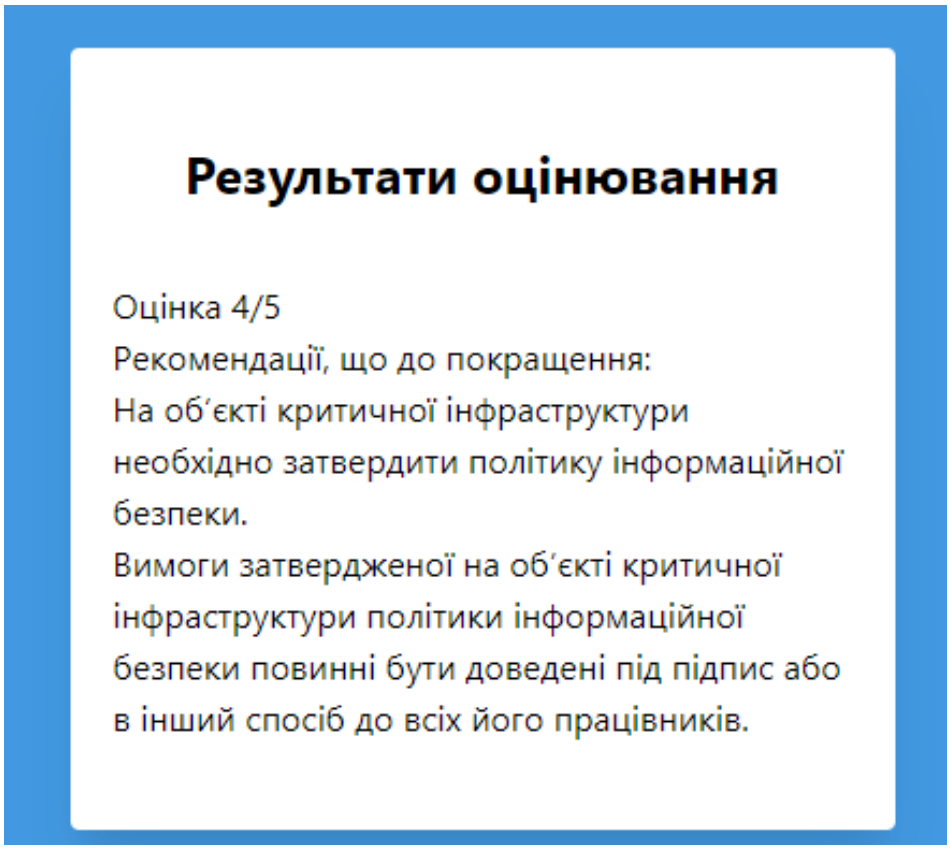


Рисунок 3.16 – Сторінка результатів

Отже, в ході дослідження та розробки було досягнуто успішної реалізації, що визначається створенням працюючої логіки системи оцінювання безпеки.

Додатково були успішно розроблені та впроваджені сторінки, спрямовані на забезпечення зручного та ефективного користування системою.

Розробка інтерфейсу, спрямованого на зручність користування, дозволяє ефективно взаємодіяти з системою та максимально використовувати її можливості для забезпечення безпеки та ефективності управління інформаційними ресурсами.

3.7 Висновки до розділу

У ході даного дослідження та розробки інформаційного ресурсу були визначені, обґрунтовані та успішно реалізовані ключові аспекти, спрямовані на створення та ефективний захист інформаційного середовища. Обрана система управління базами даних PostgreSQL виявилася високоефективною, забезпечуючи надійність та розширюваність для оптимального управління обсягами даних. Вибір мови програмування JavaScript для реалізації серверної частини обґрунтований її широким застосуванням у сфері веброзробки та універсальністю використання на обох сторонах веб-додатка.

Створена база даних інформаційного ресурсу оптимально використовує реляційні таблиці та зовнішні ключі для ефективного управління та зберігання великих обсягів даних. Забезпечення інформаційного ресурсу вбудовано через застосування моделі ролей, а також механізмів автентифікації та авторизації, що гарантує конфіденційність та цілісність інформації.

Обрані технології та підходи були обґрунтовані їх ефективністю, актуальністю та відповідністю завданням інформаційного ресурсу. У висновку можна визначити, що розроблені рішення сприяють покращенню якості та безпеки управління інформаційними ресурсами, забезпечуючи оптимальні умови для їх розвитку та функціонування в сучасному інформаційному середовищі.

4 ЕКОНОМІЧНА ЧАСТИНА

У даному розділі виконано аналіз економічного потенціалу розробки захищеного консолідованого інформаційного ресурсу системного аналізу безпеки енергетичної інфраструктури регіону, яка полягає в розробці захищеного консолідованого інформаційного ресурсу для аналізу інформаційної безпеки енергетичної інфраструктури регіону, що включає оцінку комерційних можливостей, прогноз витрат на виконання наукової роботи та впровадження його результатів, а також прогноз комерційних вигід від реалізації розробленого продукту та розрахунок ефективності вкладених інвестицій та часу їх повернення.

На основі проведеного аналізу буде здійснено висновок стосовно економічної доцільності розробки захищеного консолідованого інформаційного ресурсу системного аналізу безпеки енергетичної інфраструктури регіону.

4.1 Оцінювання комерційного потенціалу розробки програмного забезпечення

Мета здійснення технологічного аудиту полягає у визначенні комерційного потенціалу розробки, яка виникла в результаті проведеної науково-технічної діяльності.

В рамках магістерської кваліфікаційної роботи була розроблена захищена консолідована інформаційна система для системного аналізу безпеки енергетичної інфраструктури регіону, яка фактично реалізована у формі веб-сервісу. Для проведення технологічного аудиту залучено трьох незалежних експертів.

У межах даної роботи такими експертами є викладачі кафедри МБІС:

– Карпінець В. В. (к.т.н., доцент каф. МБІС ВНТУ);
– Яремчук Ю. Є. (д.т.н., проф. МБІС ВНТУ);
– Грицак А. В. (доц., викл. каф. МБІС ВНТУ).

Оцінювання комерційного потенціалу здійснимо за критеріями, що наведені в таблиці 4.1

Таблиця 4.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в реальних умовах
Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в	Технічні та споживчі влас-тивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в	Технічні та споживчі властивості продукту значно кращі, ніж в
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї

Продовження табл. 4.1

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвес-тицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь- які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці.

Таблиця 4.2 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Прізвище, ініціали, посада експерта		
	1 – Карпінець В. В.	2 – Яремчук Ю. Є.	3 – Грицак А. В.
1	4	4	4
Ринкові переваги (недоліки):			
2	3	4	3
3	4	4	3

Продовження таблиці 4.2

4	4	3	4
5	3	3	4
Ринкові перспективи			
6	4	4	4
7	3	4	3
Практична здійсненність			
8	4	4	4
9	3	4	4
10	4	3	4
11	4	3	3
12	3	4	3
Сума балів	$СБ_1 = 43$	$СБ_1 = 44$	$СБ_1 = 43$
Середньоарифметична сума балів $СБ_с$	$СБ = 43,3$		

На підставі інформації, поданої в таблиці 4.2, можна зробити висновок щодо рівня комерційного потенціалу розробки. Порівняємо отримані результати з рівнями комерційного потенціалу, які викладені в таблиці 4.3.

Таблиця 4.3 – Рівні комерційного потенціалу розробки

Середньоарифметична сума балів СБ, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 – 10	Низький
11 – 20	Нижче середнього
21 – 30	Середній
31 – 40	Вище середнього
41 – 48	Високий

На основі проведених досліджень встановлено, що рівень комерційного потенціалу розробки, присвяченої темі "Захищений консолідований інформаційний ресурс системного аналізу безпеки екологічної інфраструктури

регіону", складає 43,3 бали. Згідно з таблицею 4.3, це свідчить про високу комерційну важливість проведення цих досліджень.

4.2 Прогнозування витрат на виконання наукової роботи та впровадження її результатів

Прогнозування витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технічної роботи включає три ключові етапи, які ретельно розглядають різні аспекти витрат та впливають на всі аспекти виконання проекту.

На першому етапі проводиться визначення витрат, які напряму пов'язані із зусиллями та ресурсами, витраченими виконавцями цього розділу роботи. Сюди включаються витрати на оплату праці, навчання та інші витрати, які безпосередньо пов'язані із здійсненням цієї конкретної роботи.

Другий етап передбачає розрахунок загальних витрат на виконання всієї роботи. Це включає витрати на матеріали, обладнання, послуги та інші загальні витрати, що стосуються всього проекту.

Третій етап охоплює прогнозування загальних витрат на виконання впровадження результатів даної роботи. Це включає витрати на впровадження розробок, рекламу, підготовку персоналу та інші витрати, пов'язані з реалізацією отриманих результатів.

Важливо зауважити, що на цьому етапі використовується конкретна структура витрат, враховуючи, що для розробки інформаційної технології використовується лише один розробник програмного забезпечення.

Основна заробітна плата $З_о$:

$$З_о = \frac{М}{Т_р} \times t, \text{грн} \quad (4.1)$$

Де М – місячний посадовий оклад – 30 000 грн.;

$Т_р$ – число робочих днів в місяць; приблизно $Т_р$ = 24 днів;

t – число робочих днів роботи – 42 днів.

Таким чином:

$$З_о = \frac{30\,000}{24} \times 42 = 60\,000\ (грн.)$$

Таблиця 4.4 – Витрати по заробітній платі

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату
Керівник	32 000	1333,3	50	66 666,7
Веб-розробник	30 000	1250	42	52 500
Всього				119 166,7

Витрати на основну заробітну плату робітників ($З_р$) за відповідними найменуваннями робіт розраховується за формулою:

$$З_р = \sum_{i=1}^{n} C_i \cdot t_i ,$$

(4.2)

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

t_i – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_р \cdot t_{зм}} ,$$

(4.3)

де M_M – розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), приймемо $Мм$=6700,00 грн;

K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду;

K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

T_p – середнє число робочих днів в місяці, приблизно T_p = 24 дні;

$t_{зм}$ – тривалість зміни, год.

$C_1 = 6700{,}00 \cdot 1{,}10 \cdot 1{,}65 / (24 \cdot 8) = 63{,}34$ грн.

$З_{р1} = 63{,}34 \cdot 6{,}00 = 380{,}02$ грн.

Таблиця 4.5. Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Установка електронно-обчислювального обладнання	6	2	1,1	63,34	380,02
Підготовка робочого місця дослідника	2,4	2	1,1	63,34	152,01
Інсталяція програмного забезпечення	2,2	5	1,7	97,88	215,34
Всього					747,36

$З_Р = 747{,}36$

Додаткова заробітна плата $З_д$ працівників:

$$З_{дод} = (З_о + З_р) \cdot \frac{Н_{дод}}{100\%},$$

(4.4)

де $Н_{дод}$ – норма нарахування додаткової заробітної плати. Приймаємо 10%.

$З_{дод}$ = (119 166,7+ 747,36) · 10 / 100% = 11 916,6 грн.

Нарахування на заробітну плату Зн розробника становить:

$$З_н = (З_о + З_р + З_{дод}) \cdot \frac{Н_{зп}}{100\%}$$

(4.5)

де $Н_{зп}$ – норма нарахування на заробітну плату. Приймаємо 22%.

Зн = ((119 166,7 + 747,36 + 11 916,6) · 22 / 100% = 29 002,7 грн.

Розрахунок витрат на комплектуючі

Витрати на комплектуючі (Кв), які використовують при проведенні НДР, відсутні.

Спецустаткування для наукових (експериментальних) робіт

До статті «Спецустаткування для наукових (експериментальних) робіт» належать витрати на виготовлення та придбання спецустаткування, яке необхідне для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування та встановлення. В дослідній роботі витрати на спец устаткування відсутні.

Розрахунок витрат на програмне забезпечення

До даної статті витрат належать витрати на придбання необхідного програмного забезпечення та витрати на їх проектування, формування та встановлення.

Балансову вартість програмного забезпечення розраховуємо за формулою:

$$В_{прг} = \sum_{i=1}^{k} Ц_{iпрг} \cdot С_{прг.i} \cdot К_i \ , \qquad (4.6)$$

де $Ц_{iпрг}$ – ціна придбання одиниці програмного засобу даного виду, грн;

$C_{прг.і}$ – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, (K_i = *1,10…1,12);*

k – кількість найменувань програмних засобів.

$\text{Впрг} = 13\,299 \cdot 2 \cdot 1{,}12 = 29\,789{,}8$*грн.*

Отримані результати зведемо до таблиці:

Таблиця 4.6 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
ОС Windows 10 Pro for Workstations	2	13 299	29 789,8
DataGrip	1	8700	9 744
Всього			39 533,8

Амортизація обладнання, комп’ютерів та приміщень, які використовувались під час виконання даного етапу роботи розраховуємо за формулою:

$$\text{А} = \frac{\text{Ц} \times \text{Т}}{12 \times \text{Т}_\text{в}} \tag{4.7}$$

де Ц – загальна балансова вартість обладнання, приміщення тощо, грн.;

Т – фактична тривалість використання, міс;

$\text{Т}_\text{о}$ – термін використання обладнання, приміщення тощо, роки.

Розробка програмного забезпечення ведеться приблизно 2 місяці.

Для офісного приміщення $\text{А} = \frac{87\,000 \times 2}{12 \times 20} = 725$ грн.;

Для ноутбука $\text{А} = \frac{38\,500 \times 2}{12 \times 4} = 1\,604{,}1$ грн.;

Розрахунки зведено до таблиці 4.7:

Таблиця 4.7 – Амортизаційні відрахування

Найменування	Балансова вартість (грн.)	Термін використання (років)	Фактична тривалість використання, (міс.)	Величина амортизаційних відрахувань, (грн.)
Офісне приміщення	87 000	20	2	725
Ноутбук Lenovo IdeaPad Gaming 3 15ARH7	33 999	4	2	1 416,6
Ноутбук Apple New MacBook Air M1 13.3" 256Gb MGN63	38 499	4	2	1 604,1
			Всього	3 745,7

Витрати на матеріали, що були використані під час виконання даного етапу роботи, розраховуються за формулою:

$$К = \sum_{1}^{п} Н_i \times Ц_i \times К_i - \sum_{j=1}^{n} В_j \times Ц_{в\,j}\ (грн.) \quad (4.8)$$

$Н_j$ – норма витрат матеріалу j-го найменування, кг;

n – кількість видів матеріалів;

$Ц_j$ – вартість матеріалу j-го найменування, грн/кг;

$К_j$ – коефіцієнт транспортних витрат, (Кj = 1,1 … 1,15);

$В_j$ – маса відходів j-го найменування, кг;

$Ц_{вj}$ – вартість відходів j-го найменування, грн/кг.

Таблиця 4.8 – Витрати на матеріали

Найменування матеріалів	Ціна за од, грн	Норма витрат, од	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
Комп'ютерна мишка	750 грн.	2	0	0	1650 грн.
Килимок для миші	450 грн.	2	0	0	990 грн.
Папір	215 грн	1	0	0	236,5 грн.
Канцелярський набір	200 грн	2	0	0	440 грн.
				Всього	3 316,5 грн.

Витрати на силову електроенергію $В_е$ розраховуються за формулою:

$$В_е = \sum_{i=1}^{п} \frac{W_{yt} \times t_i \times Ц_в \times К_{впі}}{\eta_i} \text{ (грн.)} \quad (4.9)$$

$Ц_в$ – вартість 1 кВт – год. (на сьогодні для підприємців вартість 7,50 грн./кВт-год.);

W_{yt} – установлена потужність обладнання;

t_i – фактична кількість годин роботи обладнання;

$К_{впі}$ – коефіцієнт використання потужності, $К_{впі} < 1$;

η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

Таблиця 4.9 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Ноутбук Lenovo IdeaPad Gaming 3 15ARH7	0,6 кВт	350 год.	1 542,5 грн.
Ноутбук Apple New MacBook Air M1 13.3" 256Gb MGN63	0,6 кВт	294 год.	1 259,5 грн
Робоче місце розробника	0.2 кВт	350 год.	514,2 грн.
		Всього	3 316,2 грн.

Інші витрати I_B охоплюють:

– витрати на управління організацією;

– оплату службових відряджень;

– витрати на утримання, ремонт та експлуатацію, основних засобів;

– витрати на опалення, водопостачання, охорону праці тощо.

Інші витрати $B_{ін}$ можна прийняти як 100% від суми основної заробітної плати розробника:

$$I_B = 119\,166{,}7 \times 1 = 119\,166{,}7\ (\text{грн.})$$

Накладні (загальновиробничі) витрати

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов’язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов’язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов’язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуємо як 100…150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{нзв} = (З_о + З_р) \cdot \frac{H_{нзв}}{100\%} ,$$

(4.10)

де $H_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати», приймемо $Н_{нзв}$ = 130%.

$B_{нзв}$ = (119 166,7+ 747,36) · 130 / 100% = 155 888,3грн.

Сума всіх попередніх статей витрат дає витрати на виконання даної частини роботи – $В_{заг}$

$$В_{заг} = 119\,166{,}7 + 747{,}36 + 11\,916{,}6 + 29\,002{,}7 + 39\,533{,}8 + 3\,745{,}7 + 3\,316{,}2 + 3\,316{,}5 + 119\,166{,}7 + 155\,888{,}3 = 385\,800{,}6(\text{грн.})$$

Проведемо прогнозування загальних витрат ЗВ на виконання та впровадження виконаної наукової роботи. Прогнозування здійснюється за формулою:

$$ЗВ = \frac{В_{заг}}{\beta}, \text{грн.}$$

(4.11)

β – коефіцієнт, який характеризує етап виконання даної роботи.

Так, якщо розробка знаходиться:

- на стадії науково-дослідних робіт, то $\beta \approx 0{,}1$;
- на стадії технічного проєктування, то $\beta \approx 0{,}2$;
- на стадії розробки конструкторської документації, то $\beta \approx 0{,}3$;
- на стадії розробки технології, то $\beta \approx 0{,}4$;
- на стадії розробки дослідного зразка, то $\beta \approx 0{,}5$;
- на стадії розробки промислового зразка, то $\beta \approx 0{,}7$;
- на стадії впровадження, то $\beta \approx 0{,}9$.

$В_{заг}$ – загальна вартість всієї наукової роботи.

$$ЗВ = \frac{385\,800{,}6}{0.9} = 428\,667{,}33\ (грн.)$$

Отже, прогноз загальних витрат ЗВ на виконання та впровадження результатів виконаної роботи складає 428 667,33 (грн.)

4.3 Прогнозування комерційних ефектів від реалізації результатів розробки

У цьому розділі проведено кількісний прогноз та оцінку очікуваної вигоди та можливого прибутку, які можуть виникнути внаслідок впровадження результатів наукової роботи у майбутньому.

У сучасних умовах ринкової конкуренції ключовим показником позитивного впливу, який підприємство отримує від впровадження розробок, є збільшення чистого прибутку. Оцінка зростання чистого прибутку може бути проведена в теперішній вартості грошей.

Підвищення чистого прибутку внаслідок впровадження розробки справить вплив на отримання додаткових коштів для підприємства, що сприятиме покращенню його фінансових показників. Тривалість реалізації даної наукової роботи та впровадження її результатів приблизно оцінюється у 8 місяців. Очікується, що позитивні результати від впровадження розробки будуть помітні вже протягом першого місяця.

Цей період впровадження не лише дозволить отримати перші позитивні вигоди, але й створить передумови для стійкого зростання прибутковості підприємства в подальшому. Таким чином, реалізація проекту відзначиться швидким ефектом і позитивним внеском у фінансовий успіх підприємства.

Детальний прогноз позитивних результатів та їх кількісне оцінювання по роках буде виконано.

Обчислення збільшення чистого прибутку підприємства ΔП для кожного із років, протягом яких передбачається отримання позитивних результатів від впровадження розробки, розраховується за відповідною формулою:

$$\Delta П_i = \sum_{1}^{n} (\pm \Delta Ц_о \times N + Ц_о \times \Delta N)_i \times \lambda \times \rho \times (1 - \frac{\vartheta}{100}),$$

(4.12)

$\pm\Delta Ц_о$ – зміна вартості програмного продукту від впровадження результатів науково-технічної розробки, приймемо 13000,00 грн;

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки, приймемо 400 користувачів;

$Ц_б$ – вартість програмного продукту у році до впровадження результатів розробки, приймемо 257 000,00 грн;

ΔN – збільшення кількості споживачів продукту, у періоди часу, що аналізуються, від покращення його певних характеристик;

– протягом першого року – збільшення на 30 одиниць;

– протягом другого року – додаткове зростання на 55 одиниць;

– протягом третього року – додаткове зростання на 70 одиниць.

– протягом четвертого року – додаткове зростання на 75 одиниць.

λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2023 році ставка податку на додану вартість складає 20%, а коефіцієнт $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту. Приймемо ρ =38%;

ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2023 році $\vartheta = 18\%$;

Збільшення чистого прибутку $\Delta П_1$ протягом першого року складе:

$$\Delta П_1 = (13000 \times 400 + 270\,000{,}00 \times (30)) \times 0{,}8333 \times 0{,}38 \times \left(1 - \frac{0{,}18}{100}\right) = 4\,203\,917{,}5\ (\text{грн.})$$

Обчислимо збільшення чистого прибутку $\Delta П_2$ протягом другого року:

$$\Delta П_2 = (13000 \times 400 + 270\,000{,}00 \times (30 + 55)) \times 0{,}8333 \times 0{,}38 \times \left(1 - \frac{0{,}18}{100}\right) = 8\,897\,765{,}2(\text{грн.})$$

Збільшення чистого прибутку $\Delta П_3$ протягом третього року становитиме:

$$\Delta П_3 = (13000 \times 400 + 270\,000{,}00 \times (30 + 55 + 70)) \times 0{,}8333 \times 0{,}38 \times \left(1 - \frac{0{,}18}{100}\right) = 14\,871\,753{,}3\ (\text{грн.})$$

Збільшення чистого прибутку $\Delta П_4$ протягом четвертого року становитиме:

$$\Delta П_4 = (13000 \times 400 + 270\,000{,}00 \times (30 + 55 + 70 + 75)) \times 0{,}8333 \times 0{,}38 \times \left(1 - \frac{0{,}18}{100}\right) = 21\,272\,454{,}7(\text{грн.})$$

Отже, відповідно до обчислень, комерційна вигода від впровадження розробки, як і передбачалося, буде суттєвою і виявиться в зростанні чистого прибутку підприємства.

4.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності

Ключовими критеріями, які визначають обґрунтованість фінансування певним інвестором наукової розробки, є абсолютна та відносна ефективність інвестицій, а також термін їх повернення.

На першому етапі проводиться розрахунок теперішньої вартості інвестицій (PV), які вкладено у наукову розробку.

Величина початкових інвестицій PV, які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$\text{PV} = k_{інв} \times ЗВ \tag{4.13}$$

$k_{інв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо $k_{інв}$=3;

$ЗВ$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 428 667,33 грн.

$$\text{PV} = 3 \times 428\,667{,}33 = 1\,286\,002$$

Другий етап включає розрахунок очікуваного зростання прибутку (ΔПі), який отримає підприємство чи організація від впровадження результатів наукової розробки. Цей розрахунок виконується для кожного року, починаючи з першого року впровадження.

Таке збільшення прибутку також було розраховане нами раніше та становить:

$$\Delta П_1 = 4\,203\,917{,}5 \text{ (грн.)}$$

$$\Delta П_2 = 8\,897\,765{,}2 \text{ (грн.)}$$

$$\Delta П_3 = 14\,871\,753{,}3 \text{ (грн.)}$$

$$\Delta П_4 = 21\,272\,454{,}7 \text{ (грн.)}$$

На третьому етапі створюємо вісь часу, на якій відтворюємо всі фінансові транзакції, включаючи інвестиції та прибутки, що відбуваються протягом виконання науково-дослідної роботи та впровадження її результатів.

Рисунок 4.1 характеризує рух платежів (інвестицій та додаткових прибутків).

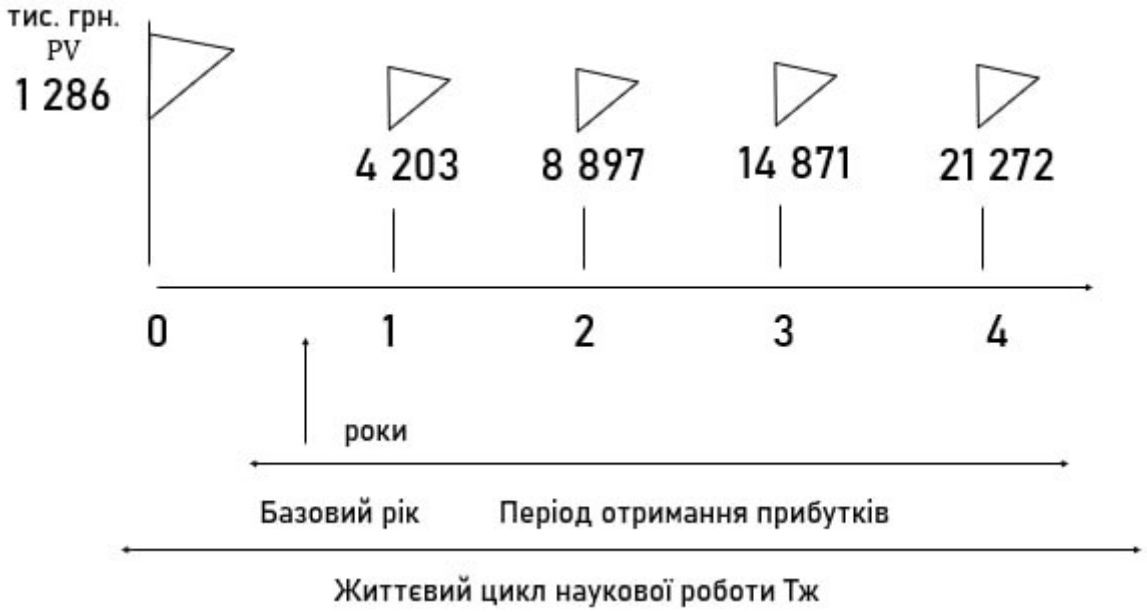


Рисунок 4.1 – Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів

На четвертому етапі проведемо обчислення абсолютної ефективності вкладених інвестицій (Еабс) відповідно до наступної формули:

$$\text{Е}_{\text{абс}} = (\text{ПП} - \text{PV}), (\text{грн.})$$

(4.9)

ПП – приведена вартість всіх чистих прибутків, що їх отримає підприємство (організація) від реалізації результатів наукової розробки, грн.;

PV – теперішня вартість інвестицій, грн.

Приведена вартість всіх чистих прибутків ПП розраховується за формулою:

$$\text{ПП} = \sum_{1}^{\text{T}} \frac{\Delta\text{П}_1}{(1+\tau)^t}, (\text{грн})$$

(4.14)

$\Delta\text{П}_1$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн,;

Т – період часу, протягом якого виявляються результати впровадженої НДДКР, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні – 0,25;

t – період часу (в роках) від моменту отримання чистого прибутку до точки «0»;

$$ПП = \frac{4\ 203\ 917,5}{(1+0,25)^1} + \frac{8\ 897\ 765,2}{(1+0,25)^2} + \frac{14\ 871\ 753,3}{(1+0,25)^3} + \frac{21\ 272\ 454,7}{(1+0,25)^4} = 25\ 385\ 238,9(\text{грн.})$$

$$Е_{абс} = 25\ 385\ 238,9 - 1\ 286\ 002 = 24\ 099\ 236,9\ (\text{грн.})$$

Оскільки $Е_{абс} > 0$, встановлено, що проведення наукових досліджень для розробки програмного продукту та його подальше впровадження призведуть до отримання прибутку. Це підтверджує обґрунтованість проведення досліджень. Однак цей факт ще не гарантує зацікавленості інвестора у фінансуванні даної програми.

На п'ятому етапі розраховуємо відносну (щорічну) ефективність вкладених інвестицій в наукову розробку $Е_в$ за відповідною формулою:

$$Е_в = \sqrt[Т_ж]{1 + \frac{Е_{абс}}{PV}} - 1 \quad (4.15)$$

$Е_{абс}$ – абсолютна ефективність вкладених інвестицій, грн.;

PV – теперішня вартість інвестицій, грн.;

$Т_ж$ – життєвий цикл наукової розробки, роки.

$$Е_в = \sqrt[4]{1 + \frac{24\ 099\ 236,9}{1\ 286\ 002}} - 1 = 1,1 \text{ або } 110\%$$

Порівняємо $Е_в$ з мінімальною (бар'єрною) ставкою дисконтування τ_{min}, яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатись не будуть.

У загальному вигляді мінімальна (бар'єрна) ставка дисконтування τ_{min} визначається за формулою:

$$\tau_{min} = d + f$$

(4.16)

d – середньозважена ставка за депозитними операціями в комерційних банка;

f – показник, що характеризує ризикованість вкладень; f = 0,3.

d = 0,1.

$$\tau_{min} = 0{,}1 + 0{,}3 = 0{,}4$$

Оскільки $E_в$ = 110% > τ_{min} = 40%, то у інвестора є потенційна зацікавленість у фінансуванні даної наукової розробки.

На шостому етапі проведемо розрахунок часу окупності вкладених інвестицій у реалізацію наукового проекту, $Т_{ок}$ за формулою:

$$Т_{ок} = \frac{1}{Е_в}, \text{рік}$$

$$Т_{ок} = \frac{1}{1{,}1} = 0{,}91 \text{ (року)} \quad (4.17)$$

Зважаючи на те, що період повернення інвестицій у реалізацію наукового проекту становить менше трьох років, можна зробити висновок, що фінансування нової розробки є обґрунтованим.

4.5 Висновки до розділу

У даному розділі проведено оцінку комерційного потенціалу розробки програмного засобу для захищеного консолідованого інформаційного ресурсу системного аналізу безпеки енергетичної інфраструктури регіону. Технологічний аудит був виконаний з участю трьох незалежних експертів, які визначили, що рівень комерційного потенціалу розробки перевищує середній рівень.

Згідно проведеного оцінювання, розробка виявилася якісною та конкурентоспроможною. Рівень комерційного потенціалу розробки становить 43,3, що відповідає рівню "високий". З розрахунків витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи видно, що загальні витрати на розробку складають 428 667,33 (грн.).

Розрахована абсолютна ефективність вкладених інвестицій в сумі 24 099 236,9 (грн.) свідчить про те, що інвестор отримає прибуток від комерціалізації програмного продукту. Щорічна ефективність вкладених інвестицій в наукову розробку складає 110%, що перевищує мінімальну бар'єрну ставку дисконтування у 40%, свідчачи про зацікавленість інвесторів у фінансуванні нової розробки.

Термін окупності вкладених інвестицій у реалізацію проекту становить 0,91 року, що також підтверджує доцільність фінансування нової розробки. З урахуванням отриманих економічних показників, можна вважати, що запропонована розробка програмного засобу має високий комерційний потенціал і, отже, є обґрунтованою для подальшого впровадження.

ВИСНОВКИ

У даній магістерській кваліфікаційній роботі було проведено глибоке дослідження та розробку на тему "Захищений консолідований інформаційний ресурс системного аналізу безпеки енергетичної інфраструктури регіону". Вибір даної тематики обумовлений актуальністю, враховуючи існуючі виклики та потреби у сфері енергетики. Завдяки сучасному рівню технологічного розвитку, швидкому процесу цифровізації та росту обсягів інформації про енергетичні ресурси, постають значущі завдання щодо забезпечення кібербезпеки та конфіденційності даних у галузі енергетики.

Зростання кількості цифрових атак та загроз безпеці вимагає вдосконалення заходів для захисту енергетичних об'єктів та інфраструктури. Таким чином, розробка захищеного консолідованого інформаційного ресурсу для системного аналізу безпеки енергетичної інфраструктури набуває належної важливості і відповідає актуальним викликам та потребам енергетичної галузі.

Основний фокус дослідження спрямований на аналіз та визначення системного підходу до забезпечення безпеки енергетичної інфраструктури в конкретному регіоні через впровадження розробленого консолідованого інформаційного ресурсу. Робота включає в себе аналіз теоретичних матеріалів, визначення особливостей критичної енергетичної інфраструктури та встановлення принципів, які слід враховувати при розробці консолідованого інформаційного ресурсу.

Досліджено принципи та методи збору та обробки даних для системного аналізу, а також проаналізовано питання забезпечення безпеки енергетичних об'єктів. Сформульовані висновки на основі проведеного аналізу та визначені ключові завдання для подальших етапів дослідження.

У другому розділі детально розглянуто створення "Захищеного консолідованого інформаційного ресурсу аналізу безпеки інфраструктури". Акцент приділено особливостям розробки інформаційного ресурсу для аналізу

безпеки енергетичної інфраструктури, а також розглянуті вимоги, необхідні для забезпечення ефективності та надійності ресурсу.

У рамках розділу виконано розробку бази даних консолідованого інформаційного ресурсу, використовуючи метод сутність-зв'язок. Визначено сутності та їх взаємозв'язки для оптимального зберігання та організації інформації про безпеку енергетичної інфраструктури. Проведено нормалізацію відношень бази даних для поліпшення структури та уникнення аномалій.

У третьому розділі здійснено практичну реалізацію бази даних, враховуючи специфіку енергетичної сфери, та проведено програмну реалізацію звітів з бази даних. Також виконано практичну реалізацію системи системного аналізу безпеки, аналізовано системну безпеку енергетичної інфраструктури регіону на основі впроваджених програмних рішень.

Четвертий розділ роботи включає аналіз економічної доцільності розробки та впровадження програмного забезпечення. Наведені економічні показники підтверджують високий комерційний потенціал розробленого продукту.

У підсумку можна визнати, що магістерська робота успішно досягла своєї основної мети, представивши розробку захищеного консолідованого інформаційного ресурсу для системного аналізу безпеки енергетичної інфраструктури регіону, і вказала на шляхи подальшого вдосконалення цієї системи в майбутньому.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Cloudflare Application Services | Security and Performance. Cloudflare. URL: https://www.cloudflare.com/application-services/?utm_source=google&utm_medium=cpc&utm_campaign=DG_EMEA_ENG_G_Search_Generic_Beta_Applications-Security&utm_content=Beta_Generic_Applications-Security_Core&utm_term=cyber+security&campaignid=7170000011271632 2&adgroupid=58700008486001012&creativeid=662071359069&gclid=Cj0KCQiA67CrBhC1ARIsACKAa8Q17aOUR4pjkbG6p-6JfA6-aLhFR_-ig7BqZ7VtZrN6wUzk1Nhp6nkaAgwEEALw_wcB&gclsrc=aw.ds).

2. Кібербезпека. URL: https://www.span.eu/ua/рішення-та-послуги/сервіси-з-безпеки/кібербезпека/?gad_source=1&gclid=Cj0KCQiA67CrBhC1ARIsACKAa8REXQS8JOs2ZaaOW6g5OyaeVWM8ksz3M6viwjX_pclBefnGsQgibfMaAgYHEALw_wcB.

3. Contributors to Wikimedia projects. Information security - Wikipedia. Wikipedia, the free encyclopedia. URL: https://en.wikipedia.org/wiki/Information_security

4. Imperva. URL: https://www.imperva.com/learn/data-security/information-security-infosec/.

5. What is information security (infosec)?. Cisco. URL: https://www.cisco.com/c/en/us/products/security/what-is-information-security-infosec.html.

6. What is information security (infosec)? Goals, types and applications. Exabeam. URL: https://www.exabeam.com/explainers/information-security/information-security-goals-types-and-applications/.

7. Yasar K., Wright G., Teravainen T. What is information security (infosec)? – techtarget definition. Security. URL: https://www.techtarget.com/searchsecurity/definition/information-security-infosec.

8. What is information security? | IBM. IBM in Deutschland, Österreich und der Schweiz | IBM. URL: https://www.ibm.com/topics/information-security.

9. What is information security? - geeksforgeeks. GeeksforGeeks. URL: https://www.geeksforgeeks.org/what-is-information-security/.

10. INFOSEC - Glossary | CSRC. NIST Computer Security Resource Center | CSRC. URL: https://csrc.nist.gov/glossary/term/INFOSEC.

11. Wright G. What is critical infrastructure? | Definition from TechTarget. WhatIs.com. URL: https://www.techtarget.com/whatis/definition/critical-infrastructure#:~:text=Critical%20infrastructure%20is%20the%20collection,each%20nation%20considers%20critical%20varies.

12. Critical infrastructure sectors | CISA. Cybersecurity and Infrastructure Security Agency CISA. URL: https://www.cisa.gov/topics/critical-infrastructure-security-and-resilience/critical-infrastructure-sectors.

13. Contributors to Wikimedia projects. Critical infrastructure - Wikipedia. Wikipedia, the free encyclopedia. URL: https://en.wikipedia.org/wiki/Critical_infrastructure.

14. Critical infrastructure. Migration and Home Affairs. URL: https://home-affairs.ec.europa.eu/pages/page/critical-infrastructure_en.

15. Critical infrastructure security and resilience | cybersecurity and infrastructure security agency CISA. Home Page | CISA. URL: https://www.cisa.gov/topics/critical-infrastructure-security-and-resilience.

16. Critical infrastructure | homeland security. Home | Homeland Security. URL: https://www.dhs.gov/science-and-technology/critical-infrastructure.

17. Five things you need to know about critical infrastructures - Institute for Environment and Human Security. Institute for Environment and Human Security. URL: https://ehs.unu.edu/blog/5-facts/5-things-about-critical-infrastructures.html.

18. What is critical infrastructure?. BlackBerry. Intelligent Security. Everywhere. URL: https://www.blackberry.com/us/en/solutions/endpoint-security/industry-4-0/critical-infrastructure.

19. Cyber and infrastructure security centre website. Cyber and Infrastructure Security Centre Website. URL: https://www.cisc.gov.au/what-is-the-cyber-and-infrastructure-security-centre/defining-critical-infrastructure.

20. Critical infrastructure protection. EU Science Hub. URL: https://joint-research-centre.ec.europa.eu/scientific-activities-z/critical-infrastructure-protection_en.

21. Critical infrastructure - Glossary | CSRC. NIST Computer Security Resource Center | CSRC. URL: https://csrc.nist.gov/glossary/term/critical_infrastructure.

22. Critical infrastructure. Cyberwatching. URL: https://www.cyberwatching.eu/cybersecurity-and-privacy-project-clusters/critical-infrastructure.

23. What is critical infrastructure? Why does critical infrastructure security matter?. Palo Alto Networks. URL: https://www.paloaltonetworks.com/cyberpedia/what-is-critical-infrastructure.

24. Critical national infrastructure | NPSA. National Protective Security Authority | NPSA. URL: https://www.npsa.gov.uk/critical-national-infrastructure-0.

25. What is critical infrastructure? | IBM. IBM in Deutschland, Österreich und der Schweiz | IBM. URL: https://www.ibm.com/topics/critical-infrastructure.

26. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. Вінниця : ВНТУ, 2021. 42 с.

27. Кавецький В. В. Економічне обґрунтування інноваційних рішень: практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепа. Вінниця : ВНТУ, 2016. 113 с.

28. What is authorization? - examples and definition - auth0. *Auth0*. URL: https://auth0.com/intro-to-iam/what-is-authorization .

29. Academy B. Seed Phrase | Binance Academy. Binance Academy. URL: https://academy.binance.com/en/glossary/seed-phrase.

30. Visual studio code. Visual Studio Code. URL: https://code.visualstudio.com/).

31. What is authorization? - examples and definition - auth0. *Auth0*. URL: https://auth0.com/intro-to-iam/what-is-authorization .

32. Visual studio code. Visual Studio Code. URL: https://code.visualstudio.com/).

33. Редактор коду visual studio code. Habr. URL: https://habr.com/ua/articles/490754

34. JavaScript. URL: https://developer.mozilla.org/en-US/docs/Web/JavaScript.

35. Electron. Electron. URL: https://www.electronjs.org/ .

36. Node.js. Node.js. URL: https://nodejs.org/uk .

37. The Modern JavaScript. URL: https://javascript.info/ .

38. Sufiyan T. What is node.js: a comprehensive guide. *Simplilearn.com*. URL: https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-nodejs).

39. Megida D. What is javascript? A definition of the JS programming language. freeCodeCamp.org. URL: https://www.freecodecamp.org/news/what-is-javascript-definition-of-js/

40. What is javascript? A basic introduction to JS for beginners. Hostinger Tutorials. URL: https://www.hostinger.com/tutorials/what-is-javascript).

41. Удосконалення стеганографічного методу вбудовування крихких цифрових водяних знаків для підвищення захищеності потокового відео від несанкціонованої модифікації / О. В. Салієва, А. В. Грицак, В. В. Гуменюк, О. П. Білик // Вимірювальна та обчислювальна техніка в технологічних процесах. – № 3, 2023. – С. 197–205.

42. Особливості управління морально-етичним станом населення під час війни в інформаційному просторі / А. А. Шиян, І. В. Абрамчук, В. В. Гуменюк // Матеріали VI Міжнародної науково-практичної конференції «Інформаційна безпека та комп`ютерні технології»: тези доповідей, 20-21 квітня 2023 р. – Кропивницький: ЦНТУ, 2023. – С. 91.

43. Гуменюк В. В. Системний аналіз безпеки енергетичної інфраструктури регіону на основі захищеного консолідованого інформаційного

ресурсу [Електронний ресурс] / В. В. Гуменюк, Ю. Є. Яремчук // Міжнародна науково-практична інтернет-конференція «Молодь в науці: дослідження, проблеми, перспективи (МН-2024)»: тези доповідей, Вінниця, 2023 р. – Електрон. текст. дані. – 2023. – Режим доступу: https://conferences.vntu.edu.ua/index.php/mn/mn2024/paper/view/19690.

44. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. Вінниця : ВНТУ, 2021. 42 с.

45. Кавецький В. В. Економічне обґрунтування інноваційних рішень: практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепа. Вінниця : ВНТУ, 2016. 113 с.

ДОДАТКИ

Додаток А. Технічне завдання

Вінницький національний технічний університет
Факультет менеджменту та інформаційної безпеки
Кафедра менеджменту та безпеки інформаційних систем

ЗАТВЕРДЖУЮ
Голова секції "Управління інформаційною безпекою" кафедри МБІС
д.т.н., професор
_____ Юрій ЯРЕМЧУК
"20" вересня 2023 р.

ТЕХНІЧНЕ ЗАВДАННЯ

до магістерської кваліфікаційної роботи на тему:
Захищений консолідований інформаційний ресурс системного аналізу безпеки енергетичної інфраструктури регіону
08-72.МКР.009.00.000.ТЗ

Керівник магістерської кваліфікаційної роботи
д.т.н., проф.
Яремчук Ю.Є.. _____

Вінниця – 2023 р.

1. Найменування та область застосування

Захищений консолідований інформаційний ресурс системного аналізу безпеки енергетичної інфраструктури регіону

2.Підстава для розробки

Розробка виконується на основі наказу ректора ВНТУ № 247 від 18 вересня 2023 р.

3. Мета та призначення розробки

3.1 Мета розробки: розробка захищеного консолідованого інформаційного ресурсу системного аналізу безпеки енергетичної інфраструктури регіону

3.2 Призначення: консолідація та системний аналіз безпеки енергетичної інфраструктури та захист цих даних.

4. Джерела розробки

4.1. Ахрамович В. М. Ідентифікація й аутентифікація, керування доступом // Сучасний захист інформації. – 2016. №4.– С. 47-51.

4.2. Бурячок В.Л. Політика інформаційної безпеки: підручник. / В.Л.Бурячок, Р.В.Грищук, В.О.Хорошко / За заг. ред. докт. техн. наук, проф. В.О. Хорошка. – К.: ПВП «Задруга», 2014. – 222 с.

4.3. Єсін В.І. Безпека інформаційних систем і технологій / В.І.Єсін, О.О. Кузнецов, Л.С. Сорока. – Харків: ХНУ імені В.Н. Каразіна, 2013. – 632 с.

4.4. ZakariaOmar, ZangooeiToomaj, MohdAfiziMohdShukran. Enhancing Mixing Recognition-Based and Recall-Based Approach in Graphical Password Scheme. IJACT, Vol. 4, No. 15, pp. 189-197, 2012.

5. Вимоги до програми

5.1 Вимоги до функціональних характеристик:

5.1.1 Програмний засіб повинен мати зручний, легкий у використанні інтерфейс користувача;

5.1.2 Реалізація методу не повинна вимагати спеціальних ліцензійних програмних додатків;

5.1.3 Програмний засіб повинен виконувати процес автентифікації користувачів у системі.

5.2 Вимоги до надійності:

5.2.1 Програмний засіб повинен працювати без помилок, у випадку виникнення критичних ситуацій необхідно передбачити виведення відповідних повідомлень;

5.2.2 Бази даних повинні бути налаштовані на автоматичне створення резервних копій;

5.2.3 Програмний засіб повинен виконувати свої функції.

5.3 Вимоги до складу і параметрів технічних засобів:

– процесор – Pentium 1500 МГц і подібні до них;

– оперативна пам’ять – не менше 512 Mb;

– середовище функціонування – операційна система сімейство Windows;

– вимоги до техніки безпеки при роботі з програмою повинні відповідати існуючим вимогам та стандартам з техніки безпеки при користуванні комп’ютерною технікою.

6. Вимоги до програмної документації

6.1 Обов'язкова поетапна інструкція для майбутніх користувачів, наведена у пункті 3.4

7. Вимоги до технічного захисту інформації

7.1 Необхідно забезпечити захист розроблюваного програмного засобу від несанкціонованого використання.

7.2 Неможливість отримання доступу незареєстрованих користувачів до інформаційних ресурсів.

8. Техніко-економічні показники

8.1 Цінність результатів використання даного проекту повинна перевищувати витрати на його реалізацію.

8.2 Має бути реалізований таким чином, щоб підходити для використання широкого загалу.

9. Стадії та етапи розробки

№	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи		Примітка
1.	Визначення напрямку магістерської роботи, формулювання теми	20.09.2023	31.09.2023	
2.	Аналіз предметної області обраної теми	01.10.2023	15.10.2023	
3.	Розробка роботи	16.10.2023	26.10.2023	
4.	Написання магістерської роботи на основі розробленої теми	27.10.2023	15.11.2023	
5.	Передзахист магістерської кваліфікаційної роботи	16.11.2023	24.11.2023	
6.	Виправлення, уточнення, корегування магістерської кваліфікаційної роботи	27.11.2023	04.12.2023	
7.	Захист магістерської кваліфікаційної роботи	11.12.2023	17.12.2023	

10. Порядок контролю та прийому

10.1 До приймання магістерської кваліфікаційної роботи надається:

– ПЗ до магістерської кваліфікаційної роботи;
– програмний додаток;
– презентація;
– відзив керівника роботи;
– відзив опонента

Технічне завдання до виконання прийняв _____ Гуменюк В.В.

Додаток Б. Лістинг програми

```
import { HttpStatus, Inject, Injectable } from '@nestjs/common';
import { QuizzesModel } from './entities/quizzes.model';
import { DeleteResult, Repository } from 'typeorm';
import { InjectRepository } from '@nestjs/typeorm';
import { JWTPayload } from 'src/auth/interfaces/jwtAuth0.payload.interface';
import { CompanyModel } from 'src/companies/entiites/companies.model';
import { MembersModel, Roles } from 'src/companies/entiites/members.model';
import { GeneralResponse } from 'src/dto/responseTemplate.dto';
import { UsersService } from 'src/users/users.service';
import { QuizDto } from './dto/quizzes.dto';
import { PaginateQuery, Paginated, paginate } from 'nestjs-paginate';
import { PassedQuestiosnDto } from './dto/passedQuestions.dto';
import { ResultOfQuizForRedis } from './interfaces/pasedQyizResultForRedis.interface';
import { MemberRatingModel } from './entities/memberRating.model';
import { UserRatingModel } from './entities/userRating.model';
import { QuizzesResultsModel } from './entities/quizzesResults.model';
import { DatesOfLastQuizzesPassedModel } from './entities/datesOfLastQuizeesPased.model';
import { CACHE_MANAGER } from '@nestjs/cache-manager';
import { Cache } from 'cache-manager';

@Injectable()
export class QuizzesService {
 constructor(
  @InjectRepository(QuizzesModel)
  private quizzesRepository: Repository<QuizzesModel>,
  @InjectRepository(CompanyModel)
  private companyRepository: Repository<CompanyModel>,
  @InjectRepository(MembersModel)
  private membersRepository: Repository<MembersModel>,
  @InjectRepository(MemberRatingModel)
  private membersRatingRepository: Repository<MemberRatingModel>,
  @InjectRepository(UserRatingModel)
  private userRatingRepository: Repository<UserRatingModel>,
  @InjectRepository(QuizzesResultsModel)
  private quizzesResultsRepository: Repository<QuizzesResultsModel>,
  @InjectRepository(DatesOfLastQuizzesPassedModel)
  private dateOfLastQuizzesPassedRepository: Repository<DatesOfLastQuizzesPassedModel>,
  private userService: UsersService,
  @Inject(CACHE_MANAGER) private cacheService: Cache,
 ) { }

 private async isCompanyAdminOrOwner(
  email: string,
  companyId: number,
 ): Promise<boolean> {
  const company = await this.companyRepository.findOne({
   where: { id: companyId },
  });
  if (!company) {
   return false;
  }
  const user = await this.userService.getUserByEmail(email);
  if (!user) {
   return false;
  }
  const member = await this.membersRepository.findOne({
```

```
    where: { userId: user.id, companyId: companyId, role: Roles.ADMIN },
   });
   if (member || company.ownerId === user.id) {
    return true;
   } else {
    return false;
   }
  }

  public async addNewQuiz(
   payload: JWTPayload,
   companyId: number,
   dto: QuizDto,
  ): Promise<GeneralResponse<QuizzesModel>> {
   try {
    const company = await this.companyRepository.findOne({
     where: { id: companyId },
    });
    if (!company) {
     throw new Error('Company not found');
    }
    const isAdminOrOwner = await this.isCompanyAdminOrOwner(
     payload.email,
     companyId,
    );
    if (!isAdminOrOwner) {
     throw new Error('Quiz can create only by company owner or admin');
    }

    const newQuiz = await this.quizzesRepository.create({
     ...dto,
     companyId: companyId,
    });
    await this.quizzesRepository.save(newQuiz);
    return new GeneralResponse(
     newQuiz,
     'Quiz is created',
     HttpStatus.CREATED,
    );
   } catch ({ error, message }) {
    return new GeneralResponse(
     message,
     'Quiz not created',
     HttpStatus.BAD_REQUEST,
    );
   }
  }

  public async deleteQuiz(
   payload: JWTPayload,
   quizId: number,
  ): Promise<GeneralResponse<DeleteResult>> {
   try {
    const quiz = await this.quizzesRepository.findOne({
     where: { id: quizId },
    });
    if (!quiz) {
     throw new Error('Quiz not found');
```

```
    }
    const company = await this.companyRepository.findOne({
      where: { id: quiz.companyId },
    });
    if (!company) {
      throw new Error('Company not found');
    }
    const isAdminOrOwner = await this.isCompanyAdminOrOwner(
      payload.email,
      quiz.companyId,
    );
    if (!isAdminOrOwner) {
      throw new Error('Quiz can delete only by company owner or admin');
    }
    return new GeneralResponse(
      await this.quizzesRepository.delete(quizId),
      'Quiz is deleted',
      HttpStatus.OK,
    );
  } catch ({ error, message }) {
    return new GeneralResponse(
      message,
      'Quiz not deleted',
      HttpStatus.BAD_REQUEST,
    );
  }
}

public async updateQuiz(
  payload: JWTPayload,
  quizId: number,
  dto: QuizDto,
): Promise<GeneralResponse<QuizzesModel>> {
  try {
    const quiz = await this.quizzesRepository.findOne({
      where: { id: quizId },
    });
    if (!quiz) {
      throw new Error('Quiz not found');
    }
    const isAdminOrOwner = await this.isCompanyAdminOrOwner(
      payload.email,
      quiz.companyId,
    );
    if (!isAdminOrOwner) {
      throw new Error('Quiz can delete only by company owner or admin');
    }
    await this.quizzesRepository.update(quizId, dto);
    return new GeneralResponse(
      await this.quizzesRepository.findOne({ where: { id: quizId } }),
      'Quiz is updated',
      HttpStatus.OK,
    );
  } catch ({ error, message }) {
    return new GeneralResponse(
      message,
      'Quiz is not updated',
      HttpStatus.BAD_REQUEST,
```

```
    );
  }
}

public async getQuizById(
  quizId: number,
): Promise<GeneralResponse<QuizzesModel>> {
  try {
    const quiz = await this.quizzesRepository.findOne({
      where: { id: quizId },
    });
    if (!quiz) {
      throw new Error('Quiz not found');
    }
    return new GeneralResponse(quiz, 'Get quiz by id', HttpStatus.OK);
  } catch ({ error, message }) {
    return new GeneralResponse(
      message,
      'Quiz is not updated',
      HttpStatus.BAD_REQUEST,
    );
  }
}

public async allCompanyQuizzes(
  query: PaginateQuery,
  companyId: number,
): Promise<GeneralResponse<Paginated<QuizzesModel>>> {
  const paginateQuizzes = paginate(query, this.quizzesRepository, {
    sortableColumns: [
      'id',
      'quizName',
      'quizDescription',
      'created_at',
      'updated_at',
    ],
    nullSort: 'last',
    defaultSortBy: [['id', 'DESC']],
    searchableColumns: ['quizName', 'quizDescription', 'id'],
    select: [
      'id',
      'quizName',
      'quizDescription',
      'questions',
      'frequency',
      'companyId',
      'created_at',
      'updated_at',
    ],
    where: { companyId: companyId },
  });
  return new GeneralResponse(
    await paginateQuizzes,
    'Quizzes list',
    HttpStatus.OK,
  );
}
```

```
public async passQuiz(
  payload: JWTPayload,
  quizId: number,
  passedQuestions: PassedQuestiosnDto[],
): Promise<GeneralResponse<QuizzesResultsModel>> {
  try {
    const quiz = await this.quizzesRepository.findOne({
      where: { id: quizId },
    });
    if (!quiz) {
      throw new Error('Quiz not found');
    }
    const user = await this.userService.getUserByEmail(payload.email);
    if (!user) {
      throw new Error('User is not found');
    }
    const company = await this.companyRepository.findOne({
      where: { id: quiz.companyId },
    });
    if (!company) {
      throw new Error('Company not found');
    }
    const member = await this.membersRepository.findOne({
      where: { userId: user.id, companyId: company.id },
    });
    if (!member) {
      throw new Error('Member not found');
    }
    const resultOfQuizPassed: {
      numberOfAllQuestin: number;
      numberOfCorectAnswers: number;
    } = {
      numberOfAllQuestin: quiz.questions.length,
      numberOfCorectAnswers: 0,
    };
    if (quiz.questions.length !== passedQuestions.length) {
      throw new Error('The number of questions does not match');
    }

    //redis data
    const resultForRedis: ResultOfQuizForRedis = {
      memberId: member.id,
      companyId: company.id,
      quizId: quiz.id,
      passedQuestionsResult: [],
    };


    //checking the results of the quiz

    passedQuestions.map((value, index) => {
      const question = quiz.questions[index];
      const compareAnsvers = question.corectAnswer.reduce(
        (score: number, element: number) =>
          score + Number(value.answers.includes(element)),
        0,
      );
      const points = compareAnsvers / question.corectAnswer.length;
```

```
  const fines =
    value.answers.filter(
      (element) => !question.corectAnswer.includes(element),
    ).length / question.corectAnswer.length;
  const totalPoints = points - fines;
  if (totalPoints > 0) {
    resultOfQuizPassed.numberOfCorectAnswers += totalPoints;
  }
  resultForRedis.passedQuestionsResult.push({
    question: value.question,
    answers: value.answers,
    numberCorectAnswers:
      totalPoints > 0 ? Number(totalPoints.toFixed(3)) : 0,
  });
});

// update member rating
const memberRating = await this.membersRatingRepository.findOne({
  where: { memberId: member.id },
});
if (!memberRating) {
  const averageScore = Number(
    (
      resultOfQuizPassed.numberOfCorectAnswers /
      resultOfQuizPassed.numberOfAllQuestin
    ).toFixed(3),
  );
  const newMemberRating = await this.membersRatingRepository.create({
    memberId: member.id,
    allCorrectlyPassedQuestions: Number(
      resultOfQuizPassed.numberOfCorectAnswers.toFixed(3),
    ),
    allPassedQuestions: resultOfQuizPassed.numberOfAllQuestin,
    averageScore: averageScore,
  });
  await this.membersRatingRepository.save(newMemberRating);
} else {
  const updateMemberAllCorrectlyPassedQuestions = Number(
    (
      Number(memberRating.allCorrectlyPassedQuestions) +
      resultOfQuizPassed.numberOfCorectAnswers
    ).toFixed(3),
  );
  const updateMemberAllPassedQuestions =
    memberRating.allPassedQuestions +
    resultOfQuizPassed.numberOfAllQuestin;
  const updateMemberAverageScore = Number(
    (
      updateMemberAllCorrectlyPassedQuestions /
      updateMemberAllPassedQuestions
    ).toFixed(3),
  );
  await this.membersRatingRepository.update(memberRating.id, {
    allCorrectlyPassedQuestions: updateMemberAllCorrectlyPassedQuestions,
    allPassedQuestions: updateMemberAllPassedQuestions,
    averageScore: updateMemberAverageScore,
  });
}
```

```
//update user rating
const userRating = await this.userRatingRepository.findOne({
  where: { userId: user.id },
});
if (!userRating) {
  const averageScore = Number(
    (
      resultOfQuizPassed.numberOfCorectAnswers /
      resultOfQuizPassed.numberOfAllQuestin
    ).toFixed(3),
  );
  const newUserRating = await this.userRatingRepository.create({
    userId: user.id,
    allCorrectlyPassedQuestions: Number(
      resultOfQuizPassed.numberOfCorectAnswers.toFixed(3),
    ),
    allPassedQuestions: resultOfQuizPassed.numberOfAllQuestin,
    averageScore: averageScore,
  });
  await this.userRatingRepository.save(newUserRating);
} else {
  const updateUserRatingAllCorrectlyPassedQuestions = Number(
    (
      Number(userRating.allCorrectlyPassedQuestions) +
      resultOfQuizPassed.numberOfCorectAnswers
    ).toFixed(3),
  );
  const updateUserRatingAllPassedQuestions =
    userRating.allPassedQuestions + resultOfQuizPassed.numberOfAllQuestin;
  const updateUserRatingAverageScore = Number(
    (
      updateUserRatingAllCorrectlyPassedQuestions /
      updateUserRatingAllPassedQuestions
    ).toFixed(3),
  );
  await this.userRatingRepository.update(userRating.id, {
    allCorrectlyPassedQuestions:
      updateUserRatingAllCorrectlyPassedQuestions,
    allPassedQuestions: updateUserRatingAllPassedQuestions,
    averageScore: updateUserRatingAverageScore,
  });
}

//update date of last quiz passed
const dateOfLastQuizPassed =
  await this.dateOfLastQuizzesPassedRepository.findOne({
    where: { memberId: member.id, quizId: quiz.id },
  });
if (!dateOfLastQuizPassed) {
  const newDateOfLastQuizPassed =
    await this.dateOfLastQuizzesPassedRepository.create({
      memberId: member.id,
      quizId: quiz.id,
      dateOfQuizLastPassed: new Date(),
    });
  await this.dateOfLastQuizzesPassedRepository.save(
    newDateOfLastQuizPassed,
  );
```

```
    } else {
      await this.dateOfLastQuizzesPassedRepository.update(
        dateOfLastQuizPassed.id,
        { dateOfQuizLastPassed: new Date() },
      );
    }

    //save quiz result
    const quizResult = await this.quizzesResultsRepository.create({
      quizId: quiz.id,
      memberId: member.id,
      correctlyPassedQuestions: Number(
        resultOfQuizPassed.numberOfCorectAnswers.toFixed(3),
      ),
      passedQuestions: resultOfQuizPassed.numberOfAllQuestin,
    });
    await this.quizzesResultsRepository.save(quizResult);

    //save data to redis
    await this.cacheService.set(quizResult.id.toString(), resultForRedis, 60 * 60 * 24 * 2);

    return new GeneralResponse(quizResult, 'Quiz is passed', HttpStatus.OK);
   } catch ({ error, message }) {
    return new GeneralResponse(
      message,
      'Quiz is not passed',
      HttpStatus.BAD_REQUEST,
    );
   }
  }
}
import {
  ForbiddenException,
  HttpStatus,
  Injectable,
  UnauthorizedException,
} from '@nestjs/common';
import { CreateUserDto } from 'src/users/dto/createUser.dto';
import { UsersService } from 'src/users/users.service';
import { LoginDto } from './dto/userLogin.dto';
import { comparePassword } from 'src/utils/bcrypt.utils';
import { UserModel } from 'src/users/user.model';
import { JwtService } from '@nestjs/jwt';
import { GeneralResponse } from 'src/dto/responseTemplate.dto';
import { UserDto } from 'src/users/dto/user.dto';
import { v4 as uuidv4 } from 'uuid';
import { GenerateTokensDto } from './dto/generateTokens.dto';
import { EmailService } from 'src/email/email.service';
@Injectable()
export class AuthService {
  constructor(
    private usersService: UsersService,
    private jwtService: JwtService,
    private emailService: EmailService,
  ) {}

  public async login(
    userDto: LoginDto,
```

```
): Promise<GeneralResponse<GenerateTokensDto>> {
  try {
    const user = await this.validateUser(userDto);
    if (!user) {
      throw new Error('wrong email or password');
    }
    return new GeneralResponse(
      await this.generateTokens(user),
      'Authorization complete',
      HttpStatus.OK,
    );
  } catch ({ error, message }) {
    return new GeneralResponse(
      message,
      'Authorization not complete',
      HttpStatus.BAD_REQUEST,
    );
  }
}

public async registration(
  userDto: CreateUserDto,
): Promise<GeneralResponse<UserDto>> {
  return this.usersService.createUser(userDto);
}

public async getMe(
  payload,
): Promise<
  GeneralResponse<UserModel> | GeneralResponse<GeneralResponse<UserModel>>
> {
  const user = await this.usersService.getUserByEmail(payload.email);
  if (user) {
    return new GeneralResponse(user, 'Get me info', HttpStatus.OK);
  }
  const newAuth0User = await this.usersService.createUser({
    userName: payload.email,
    email: payload.email,
    password: await uuidv4(),
  });
  return new GeneralResponse(newAuth0User, 'Get me info', HttpStatus.OK);
}

public async refreshTokens(body: {
  token: string;
}): Promise<GeneralResponse<GenerateTokensDto>> {
  try {
    const payload = await this.jwtService.verifyAsync(body.token, {
      secret: process.env.JWT_REFRESH_SECRET,
    });
    const user = await this.usersService.getUserByEmail(payload.email);
    if (!user) {
      throw new ForbiddenException('Access Denied');
    }
    return new GeneralResponse(
      await this.generateTokens(user),
      'Tokens refreshed',
      HttpStatus.OK,
```

```
    );
  } catch (error) {
    throw new UnauthorizedException();
  }
}

private async generateTokens(user: UserModel): Promise<GenerateTokensDto> {
  const payload = {
    email: user.email,
  };

  const [accessToken, refreshToken, actionToken] = await Promise.all([
    this.jwtService.signAsync(payload, {
      secret: process.env.JWT_KEY,
      expiresIn: '15m',
    }),
    this.jwtService.signAsync(payload, {
      secret: process.env.JWT_REFRESH_SECRET,
      expiresIn: '7d',
    }),
    this.jwtService.signAsync(payload, {
      secret: process.env.JWT_ACTION_SECRET,
      expiresIn: '7d',
    }),
  ]);
  return {
    accessToken,
    refreshToken,
    actionToken,
  };
}

private async validateUser(userDto: LoginDto): Promise<UserModel> {
  const user = await this.usersService.getUserByEmail(userDto.email);
  const passwordEquals = await comparePassword(
    userDto.password,
    user.password,
  );
  if (user && passwordEquals) {
    return user;
  }
  return null;
}

public async sendEmailForgotPassword(
  email: string,
): Promise<GeneralResponse<string>> {
  try {
    const user = await this.usersService.getUserByEmail(email);
    if (!user) {
      throw new Error('User not found');
    }
    const token = await this.jwtService.signAsync(
      { email: email },
      {
        secret: process.env.EMAIL_TOKEN_SECRET,
        expiresIn: '15m',
      },
```

```
      );
      await this.emailService.sendEmailForgotPassword(token, email);
      return new GeneralResponse('Done', 'Email send', HttpStatus.OK);
    } catch ({ error, message }) {
      return new GeneralResponse(
        message,
        'Email not send',
        HttpStatus.BAD_REQUEST,
      );
    }
  }

  public async changePassword(
    token: string,
    password: string,
  ): Promise<GeneralResponse<string>> {
    try {
      const payload: { email: string } = await this.jwtService.verifyAsync(
        token,
        { secret: process.env.EMAIL_TOKEN_SECRET },
      );
      if (payload) {
        await this.usersService.changePassword(payload.email, password);
        return new GeneralResponse('Done', 'Password chenged', HttpStatus.OK);
      } else {
        throw new Error('Bad token');
      }
    } catch ({ error, message }) {
      return new GeneralResponse(
        message,
        'Password not chenged',
        HttpStatus.BAD_REQUEST,
      );
    }
  }
}
```

Додаток В. Ілюстративний матеріал

Захищений консолідований інформаційний ресурс системного аналізу безпеки енергетичної інфраструктури регіону

Виконав студент групи КІТС-22М Гуменюк В.В.

Науковий керівник: д.т.н., проф. каф МБІС Яремчук Ю.Є.

Актуальність роботи

Енергетична інфраструктура регіону є критично важливою для його нормального функціонування. Вона містить мережі електропостачання, газопостачання, теплопостачання, а також інші об'єкти, які забезпечують безперебійне постачання енергії.

Однак енергетична інфраструктура є вразливою до атак. Зловмисники можуть атакувати енергетичну інфраструктуру з різних причин, наприклад, для отримання економічної вигоди, політичної мети або просто для завдавання шкоди. Тому потрібно слідкувати за її постійною безпекою.

ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Об'єкт дослідження – енергетична інфраструктура регіону, включаючи електростанції, підстанції та енергетичні мережі, що визначають життєво важливу інфраструктуру сучасного суспільства.

Предмет дослідження – розроблений захищений консолідований інформаційний ресурс системного аналізу безпеки енергетичної інфраструктури.

НОВИЗНА РОБОТИ

Новизна роботи: впровадження інноваційного захищеного консолідованого інформаційного ресурсу системного аналізу безпеки енергетичних систем, щоб забезпечити захист від кіберзагроз, з використанням сучасних методів системного аналізу та розробки новаторських інформаційних рішень для ефективного забезпечення захисту інформаційного ресурсу.

Критична інфраструктура

Критична інфраструктура (КІ) - це комплексні об'єкти та системи, які є незамінними для нормального функціонування сучасного суспільства та забезпечують надання важливих послуг і функцій у різних сферах життя. Ця інфраструктура містить різноманітні галузі та об'єкти, і від її надійного функціонування залежить безперебійне надання послуг та забезпечення стабільності суспільства

Енергетична інфраструктура

Сфера енергетичної інфраструктури є однією з найважливіших галузей, що забезпечують функціонування сучасного суспільства. Енергетична інфраструктура містить широкий спектр систем і об'єктів, які забезпечують виробництво, передачу, розподіл і використання різних видів енергії, що необхідні для економічного розвитку та підтримання життєдіяльності нації.

Ця сфера діяльності стала необхідною частиною сучасного способу життя, якій довіряють економічний розвиток, комфорт та безпеку громадян. Проте, з огляду на свою важливість, енергетична інфраструктура стала об'єктом підвищеної уваги з боку різних загроз та викликів, що включають природні катастрофи, кібератаки, терористичні акти та інші небезпеки.

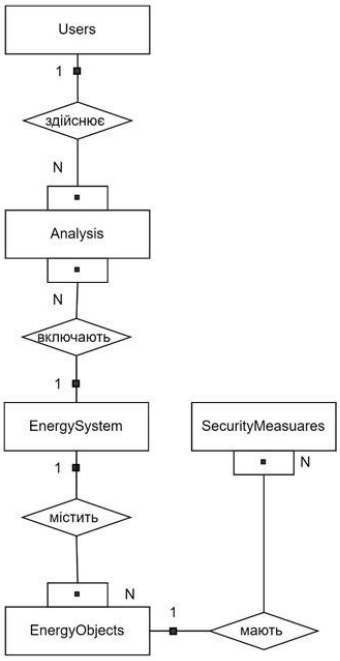
Розробка захищеного консолідованого інформаційного ресурсу

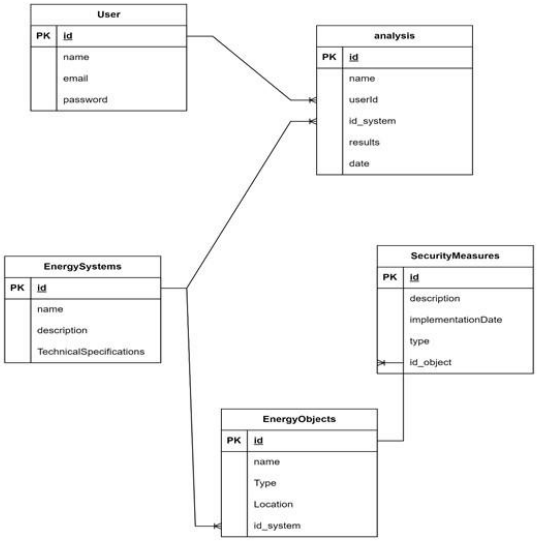
При розробці було сплановано, після чого реалізовано основні модулі та функції захищеного консолідованого інформаційного ресурсу аналізу безпеки енергетичної інфраструктури регіону.

Для початку були враховані всі особливості, для реалізації даного ресурсу. Також була розроблена база даних зберігання даних аналізів та інших даних для функціонування інформаційного ресурсу, були спроектовані звіти для зручного та зрозумілого виводу інформації.

Також велику увагу було приділено захисту інформаційного ресурсу, був спроектований алгоритм, який забезпечує захист ресурсу від несанкціонованого доступу.

База даних





Захист від несанкціонованого доступу

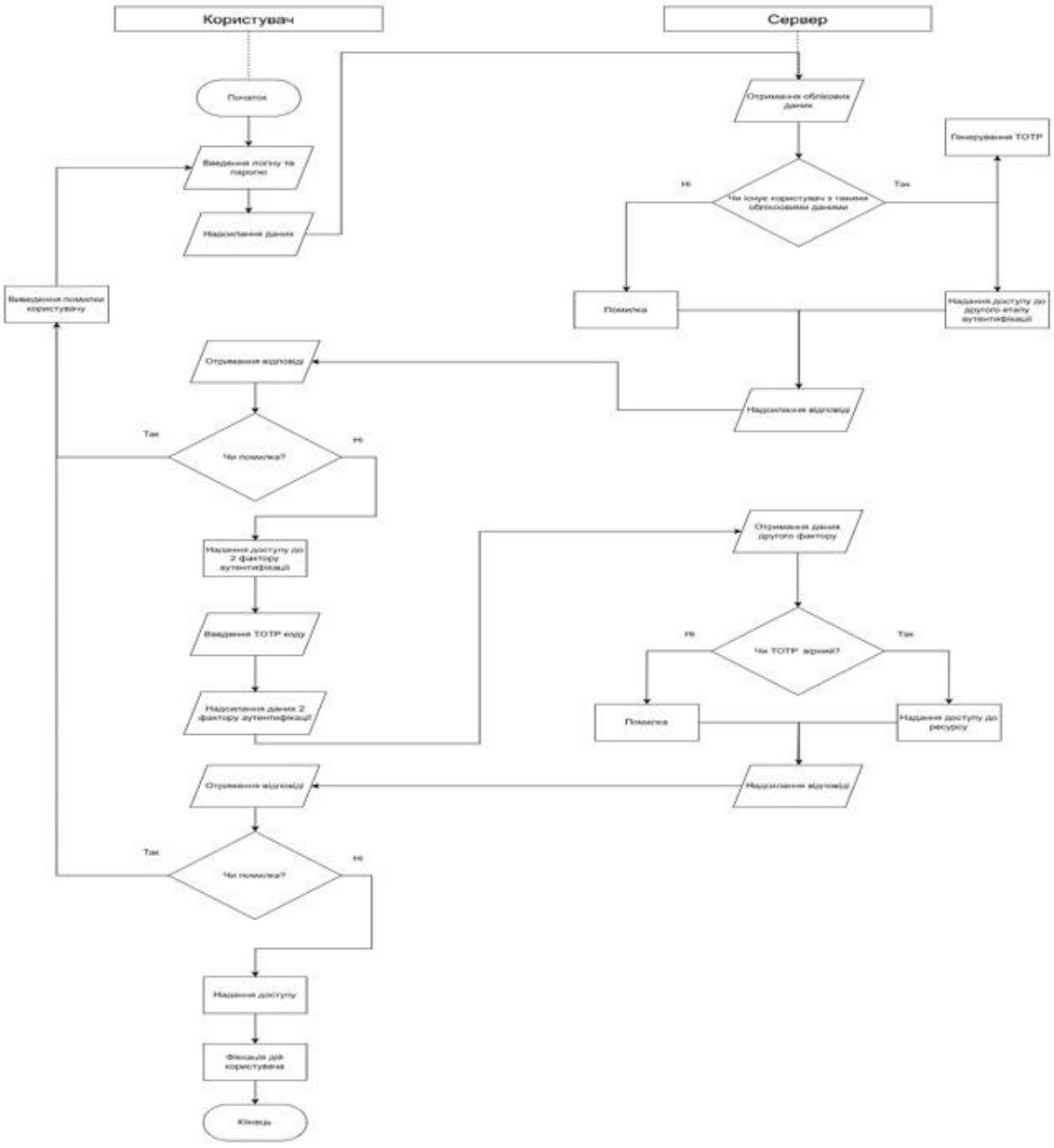
Врозроблений ресурс було впроваджено захист AAA (Authentication, Authorization, and Accounting), який охоплює комплекс заходів, спрямованих на забезпечення безпеки та ефективного управління доступом до інформаційних ресурсів. Кожна з компонент AAA відіграє ключову роль в цьому контексті.

Також була впроваджена система двоетапної аутентифікації з використанням Time-Based One-Time Password (TOTP), яка є ефективним засобом забезпечення додаткового рівня безпеки під час входу в систему чи доступу до облікового запису.

Розроблений алгоритм захисту

Розроблений алгоритм захисту поєднує в собі декілька методів захисту таких, як:

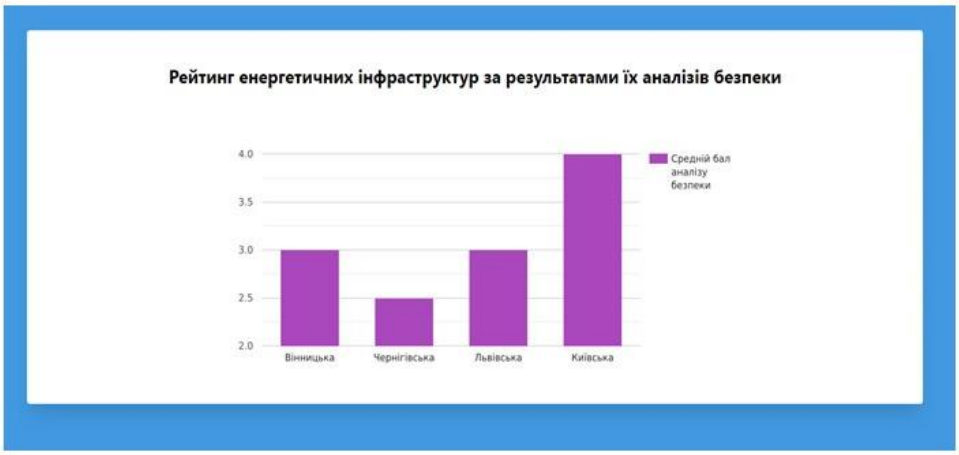
- TOTP – це одноразовий пароль, який генерується для користувача та має обмежений час життя.
- 2FA – двоетапна аутентифікація, яка забезпечує додатковий шар захисту, для перевірки другого фактору використовується TOTP, який генерується на сервері.
- AAA – даний метод захисту фіксує всі дії користувача після проходження аутентифікації , та забезпечує авторизацію користувача.



Використані технології

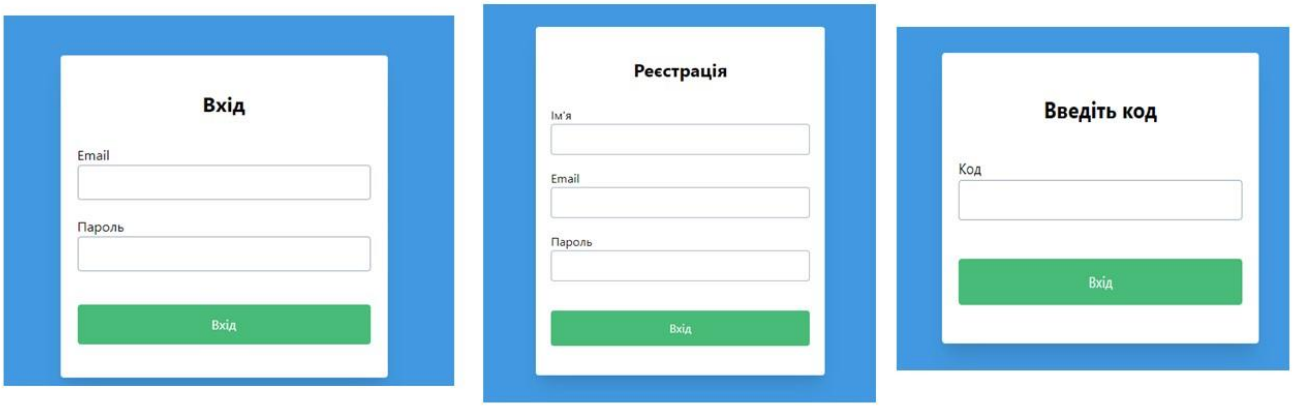
PostgreSQL

Реалізовані звіти

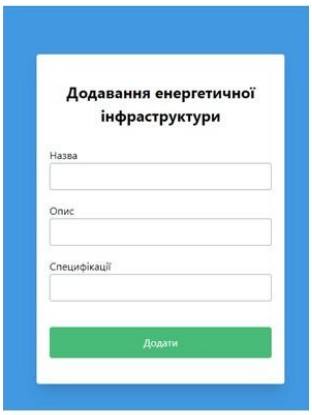


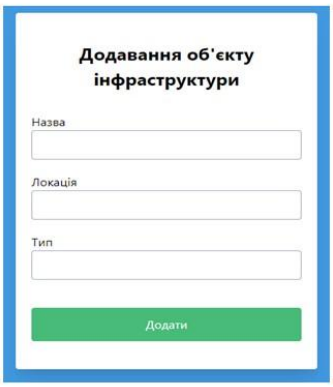
Реалізовані звіти

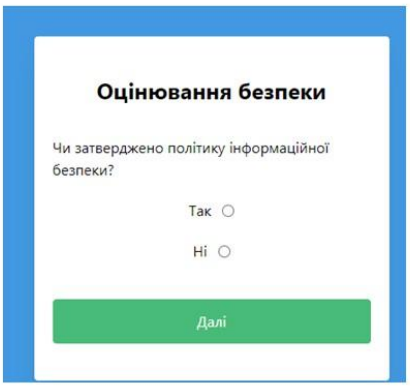
Інтерфейс реєстрації та входу



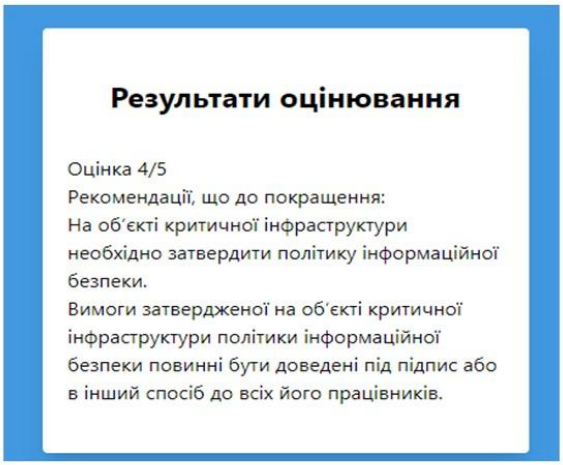
Інтерфейс







Інтерфейс



Висновок

В ході виконання роботи було досліджено принципи та методи збору та обробки даних для системного аналізу, а також проаналізовано питання забезпечення безпеки енергетичних об'єктів. . У підсумку можна визнати, що магістерська робота успішно досягла своєї основної мети, представивши розробку захищеного консолідованого інформаційного ресурсу для системного аналізу безпеки енергетичної інфраструктури регіону, і вказала на шляхи подальшого вдосконалення цієї системи в майбутньому.

Дякую за увагу!

Додаток Г. Протокол перевірки на антиплагіат

ПРОТОКОЛ
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ
ЗАПОЗИЧЕНЬ

Назва роботи: Захищений консолідований інформаційний ресурс системного аналізу безпеки енергетичної інфраструктури регіону

Тип роботи: магістерська кваліфікаційна робота
(БДР, МКР)

Підрозділ: Кафедра менеджменту та безпеки інформаційних систем
Факультет менеджменту та інформаційної безпеки
(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність 96 % Схожість 4 %

Аналіз звіту подібності (відмітити потрібне):
1. Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
2. Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
3. Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку _____ Коваль Н.П.
(підпис) (прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи _____ Гуменюк В.В.
(підпис) (прізвище, ініціали)

Керівник роботи _____ Яремчук Ю.Є.
(підпис) (прізвище, ініціали)