Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
(інститут)
Факультет інформаційних технологій
(факультет)
Кафедра інформаційних систем та технологій
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА

кваліфікаційної роботи ступеня бакалавра
(бакалавра, спеціаліста, магістра)

студента Кваші Олега Олександровича
(ПІБ)

академічної групи 123-20ск-1
(шифр)

спеціальності 123 Комп'ютерна інженерія
(код і назва спеціальності)

за освітньо-професійною програмою 123 Комп'ютерна інженерія
(офіційна назва)

на тему «Комп'ютерна система кафедри інформаційних технологій та комп'ютерної інженерії з реалізацією побудови та налаштування корпоративної мережі та з підтримкою веб-застосунку "Телеграм-бот ІТКІ" для профорієнтаційної роботи»
(назва за наказом ректора)

Керівники	**Прізвище, ініціали**	**Оцінка за шкалою**		**Підпис**
		рейтинговою	**інституційною**	
кваліфікаційної роботи	доц. Каштан В.Ю.			
розділів:				
розробка апаратної частини	доц. Ткаченко С.М.			
розробка корпоративної мережі	ас. Бешта Л.В.			
Рецензент	проф. Лактіонов І.С.			
Нормоконтролер	проф. Цвіркун Л.І.			

Дніпро

2023

ЗАТВЕРДЖЕНО:

завідувач кафедри
інформаційних систем
та комп'ютерних технологій
(повна назва)
_____ Гнатушенко В.В.
(підпис) (прізвище, ініціали)
« » 2023 року

ЗАВДАННЯ

на кваліфікаційну роботу ступеня бакалавр

студента Кваша О.О. **академічної групи** 123-20ск-1
(прізвище та ініціали) (шифр)

спеціальності 123 Комп'ютерна інженерія

за освітньо-професійною програмою 123 Комп'ютерна інженерія
(офіційна назва)

на тему «Комп'ютерна система кафедри інформаційних технологій та комп'ютерної інженерії з реалізацією побудови та налаштування корпоративної мережі та з підтримкою веб-застосунку "Телеграм-бот ІТКІ" для профорієнтаційної роботи»

затверджену наказом ректора НТУ «Дніпровська політехніка» від 16.05.2023 № 350-с

Розділ	Зміст	Термін виконання
Стан питання та постановка завдання	На основі матеріалів виробничих практик, інших науково-технічних джерел конкретизується предмет та мету роботи та виконується постанова завдання	10.05.2023
Розробка апаратної частини	На основі аналізу підприємства формулюються технічні вимоги до комп'ютерної системи та розробляється апаратна частина системи	17.05.2023
Розробка корпоративної мережі	Виконується розрахунок налаштувань корпоративної мережі та перевірка роботи системи, розробляються методи та налаштування обладнання для захисту інформації в системі	24.05.2023
Розробка компонента системи	Виконується детальна розробка компонента системи	31.05.2023

Завдання видано _____ доц. Каштан В.Ю.
(підпис керівника) (прізвище, ініціали)

Дата видачі 19.04.2023

Дата подання до екзаменаційної комісії 01.07.2023

Прийнято до виконання _____ Кваша О.О.
(підпис студента) (прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка містить: 91 сторінок., 38 рисунків., 7 таблиць, 1 додаток, 10 джерел.

VPN, КОМП’ЮТЕРНА МЕРЕЖА, LAN, VLAN, REST, HTTP

Об’єкт: комп’ютерна система кафедри інформаційних технологій та комп'ютерної інженерії з реалізацією побудови та налаштування корпоративної мережі та з підтримкою веб-застосунку "Телеграм-бот ІТКІ" для профорієнтаційної роботи.

Мета: створення комп’ютерної системи для кафедри інформаційних технологій та комп'ютерної інженерії в НТУ "Дніпровська політехніка".

Система орієнтована на збереження і відтворення заданої інформації, збір даних, взаємодію з абітурієнтами і здобувачами освіти, використовуючи інтернет-месенджер Telegram.

Система була виконана з використанням найсучасніших технологій у веб-розробці. Складається з чотирьох основних сервісів: REST API, frontend, Node-RED, сервер PostgreSQL. Всі сервіси, окрім серверу PostgreSQL, взаємодіють з іншими складовими за протоколом HTTP. Всі інтерфейси взаємодії з додатком потребують аутентифікації користувача, всі данні користувачів зберігаються у зашифрованому вигляді.

Розробка комп’ютерної мережі була виконана відповідно до завдання на кваліфікаційну роботу бакалавра.

Розроблена схема мережі реалізована у вигляді логічної топології в програмі Cisco Packet Tracer.

Результати перевірки спроектованої мережі у вигляді таблиць, графіків описані і наводяться у пояснювальній записці або додатках.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

VPN	– віртуальна приватна мережа (англ. Virtual Private Network);
КМ	– комп’ютерна мережа;
VLAN	– віртуальна локальна комп'ютерна мережа (англ. Virtual Local Area Network);
WAN	– глобальна мережа (англ. Wide Area Network);
IP-адреса	– унікальний ідентифікатор комп'ютера локальної мережі;
LAN	– локальна комп'ютерна мережа (англ. Local Area Network);
REST	– передача стану представлення (англ. Representational State Transfer), архітектурний стиль для побудови розподілених систем;
SSH	– безпечна оболонка, (англ. Secure Shell), протокол мережевого зв'язку та криптографічний протокол, що забезпечує безпечне з'єднання та взаємодію між віддаленими пристроями;
HTTP	– протокол передачі гіпертексту (англ. Hypertext Transfer Protocol), протокол, що використовується для передачі даних у веб-переглядачах та веб-серверах;
TCP/IP	– протокол передачі даних/інтернет-протокол (англ. Transmission Control Protocol/Internet Protocol), набір протоколів, що використовуються для забезпечення комунікації і передачі даних в комп'ютерних мережах.

ВСТУП

Розвиток інформаційних технологій останнім часом диктує необхідність використання комп'ютерних мереж у всіх сферах діяльності. Кафедра інформаційних технологій та комп'ютерної інженерії не є винятком і потребує реалізації комп'ютерної системи з побудови та налаштування корпоративної мережі та з підтримкою веб-застосунку "Телеграм-бот ІТКІ" для профорієнтаційної роботи.

Комп'ютерна мережа – це система взаємопов'язаних комп'ютерів та інших пристроїв, які можуть обмінюватися даними та ресурсами через засоби зв'язку, такі як кабелі, радіохвилі, інфрачервоне випромінювання або супутникові зв'язки.

"Телеграм-бот ІТКІ" – це комплексний веб-додаток, розроблений за клієнт-серверною архітектурою з використанням таких технологій як Java Spring Boot, Angular, Docker, Node-RED. Цей додаток призначений для розгортання на сервері (або кількох серверах). Основною метою цього додатку є забезпечення зв’язку з абітурієнтами та студентами (розповсюдження інформації за допомогою broadcast чат-повідомлень та моніторинг аудиторії), забезпечення автоматизації відтворення важливої інформації, забезпечення додаткового сервісу за допомогою команд.

Spring Boot – це фреймворк для розробки Java-додатків, який дозволяє розробляти додатки швидко і просто. Він заснований на фреймворку Spring Framework, але додає до нього ряд зручних інструментів, які дозволяють використовувати Spring з меншими зусиллями. Spring Boot є дуже популярним в світі Java-розробки, і він широко використовується для розробки веб-додатків, мікросервісів та інших типів додатків в різних сферах, включаючи фінанси, технології, медіа та інші галузі [5].

Angular – це фреймворк для створення веб-додатків, розроблений компанією Google. Він дозволяє розробникам створювати високопродуктивні, масштабовані та підтримувані додатки з великою кількістю функціональності.

Angular є дуже популярним фреймворком в світі веб-розробки та використовується в багатьох відомих проектах, таких як Google, Microsoft, IBM, PayPal, і багатьох інших. Він також має велику та активну спільноту розробників, яка постійно покращує його функціональність та продуктивність [6].

Node-RED – це візуальний інструмент для розробки програмного забезпечення, який базується на Node.js. Він має потужні засоби для обробки та виконання HTTP-запитів, що дозволяє розробникам створювати різні види веб-додатків, такі як веб-сервери, мікросервіси, чат-боти та інші [7].

Для будь-якої компанії веб-розробка на Java, Angular та Docker є дуже важливою і потрібною. Ці технології забезпечують можливість створювати надійні та ефективні клієнт-серверні додатки, які забезпечують зручну та швидку роботу з веб-сторінками. Багато відомих компаній використовують ці технології для створення своїх продуктів. Наприклад, Docker використовується в Amazon Web Services, Microsoft Azure та Google Cloud Platform для створення хмарних сервісів, а Angular використовується в Google, Microsoft, Apple і багатьох інших компаніях для створення клієнт-серверних додатків. Java є однією з найпопулярніших мов програмування для створення веб-додатків та клієнт-серверних додатків.

Актуальність даної роботи полягає в огляді рішень проектування та розробки веб-додатків і корпоративної мережі кафедри інформаційних технологій та комп'ютерної інженерії.

Мета цієї роботи: розробка комп'ютерної системи для кафедри, яка буде включати у себе побудову та налаштування корпоративної мережі, що дозволить об'єднати всі комп'ютери та пристрої на кафедрі в єдину систему, а також створення веб-застосунку "Телеграм-бот ІТКІ" для профорієнтаційної роботи.

1 СТАН ПИТАННЯ І ПОСТАНОВКА ЗАВДАННЯ

1.1 Стисла характеристика галузі та умов застосування комп’ютерної системи

Кафедра «Інформаційних технологій і комп'ютерної інженерії» спеціалізується на підготовці фахівців у галузі інформаційних технологій та комп'ютерної інженерії. Кафедра має декілька напрямів підготовки, включаючи бакалаврські, магістерські та аспірантські програми.

Студенти кафедри ІТКІ отримують знання та практичні навички в таких областях, як програмування, бази даних, мережі, безпека інформації, інтернет речей та багато іншого. Кафедра також пропонує різноманітні курси, семінари та майстер-класи з технічних та наукових питань у галузі ІТ.

Крім того, кафедра ІТКІ займається науковою діяльністю в галузі інформаційних технологій. Досягнення вчених кафедри охоплюють такі теми, як штучний інтелект, обробка природної мови, комп'ютерне зорове сприйняття та інші сфери застосування ІТ.

У загальному, кафедра ІТКІ НТУ "Дніпровська політехніка" є важливим центром підготовки фахівців у галузі ІТ та наукових досліджень у цій галузі.

На кафедрі ІТКІ НТУ "Дніпровська політехніка" також працює активний науковий колектив, який займається розробкою нових технологій та застосуванням існуючих у різних галузях, таких як медицина, автоматизація виробництва, електроенергетика, транспорт та інші.

Кафедра ІТКІ має сучасну інфраструктуру та обладнання, необхідні для проведення високоякісної підготовки студентів та проведення наукових досліджень. Кафедра забезпечує своїх студентів можливостями для отримання практичного досвіду та співпраці з провідними компаніями в галузі ІТ.

Кафедра ІТКІ активно співпрацює з науково-дослідними інститутами, університетами та компаніями з інших країн, що дозволяє забезпечити високий рівень освіти та наукових досліджень.

Також на кафедрі ІТКІ організовуються різноманітні заходи для студентів, які сприяють їхньому особистісному та професійному розвитку. Студенти можуть приєднатися до різних клубів та гуртків, що дозволяє їм розвивати свої навички та отримувати нові знання.

У загальному, кафедра ІТКІ є однією з провідних науково-освітніх установ в галузі інформаційних технологій та комп'ютерної інженерії в Україні. Вона забезпечує високий рівень підготовки фахівців та ведення наукових досліджень, що дозволяє випускникам кафедри успішно працювати в різних сфера.

Кваліфікацій робота полягає у розробці корпоративної мережі з можливостями віддаленого доступу за допомогою VPN та розробці веб-застосунку "Телеграм-бот ІТКІ".

1.2 Характеристика і структура об’єкта впровадження

Кафедру ІТКІ було відкрито в березні 1996 року. Першою її назвою було – кафедра «Геоінформатики». Почалась діяльність кафедри ІТКІ за наказу ректора Державної гірничої академії України (попередня назва НТУ «ДП», далі ДГАУ) академіка Півняка Г.Г. №4 від 28 березня 1996 року. На той час, в склад кафедри входили: професор, доктор технічних наук Яковлев С.В. (завідувач кафедри), доцент, кандидат фізико математичних наук Саричева Л.В. (заступник завідувача кафедрою), Нікулін С.Л. (асистент). Окрім навчальних занять, кафедра ще проводила, науково-дослідницькі роботи для обробки просторово-прив’язаних даних. У той час, під керівництвом кафедри знаходилась науково-дослідницька лабораторія геоінформаційних технологій. Основне приміщення кафедри це аудиторія 1/79. На той час кафедра ІТКІ належала до Геолого-розвідувального факультету (ГРФ).

З кожним роком існування, ІТКІ відзначала зростання своєї діяльності, зокрема з готування фахівців у напрямі "Інформаційні управляючі системи та технології", зі спеціалізацією "Геоінформаційні системи та технології". Вже з перших років функціонування кафедра стала однією з найбільш ранніх в

Україні, яка забезпечувала підготовку фахівців з розробки геоінформаційних систем (ГІС). У 1996 році ІТ-галузь України ще тільки починала свій розвиток, тому фахівці з такого нового сектору для українського ІТ-бізнесу були надзвичайно цінними. Це було актуально й до цього дня. Кафедра активно просувалася в різні сфери діяльності, що відображалося в залученні нових фахівців, зокрема викладачів-стажистів, таких як І.М. Гаркуша, а також викладачів звання доцента А.В. Кожевникова, Г.М. Коротенка, Л.Г. Ахметшини та професора О.М. Ахметшини. У 1997 році була створена філія кафедри у Міському комунальному підприємстві "Земград" у місті Дніпропетровську, а в 1998 році – у Державному інформаційно-геологічному фонді України "Геоінформ" у місті Києві.

Ця історія свідчить про те, як кафедра геоінформатики заснувала та розвивалася протягом десятиліть. Вона починалася з викладання окремих дисциплін та курсів, пов'язаних з геоінформатикою, та розвивалася до створення власних спеціальностей, готуючи ІТ-спеціалістів з компетенціями, необхідними для роботи з геоданими та іншими суміжними областями.

У 1999 році кафедра відкрила спеціальність "Комп'ютерний еколого-економічний моніторинг", яка зосереджувалася на застосуванні комп'ютерних технологій для вирішення проблем екології та економіки. У 2001 році була створена ще одна спеціальність – "Інтелектуальні системи прийняття рішень", яка зосереджувалася на вивченні методів та технологій прийняття рішень в умовах нечіткої та неповної інформації.

Протягом років кафедра активно готувала ІТ-спеціалістів з компетентностями в системному, геоінформаційному та інтелектуальному аналізі з залученням можливостей просторових систем управління базами даних та систем штучного інтелекту. Кафедра займалася вивченням та використанням різних програмних засобів, пов'язаних з обробкою геоданих.

У 2003 році кафедра геоінформатики була перейменована в кафедру геоінформаційних систем, відображаючи зміну спрямувань та основних напрямків досліджень та робіт.

На початку 2007 року кафедра мала в своєму складі докторів наук Б.С. Бусигіна, О.М. Ахметшина, В.І. Кузьменка, В.А. Воронова та кандидатів наук Л.В. Саричевої, Г.М. Коротенка, С.Л. Нікуліна, Л.Г. Ахметшиної, В.О. Трусова, В.Л. Кожевникова, А.В. Кожевникова, В.О. Салікова, А.М. Мільцина, В.К. Дорошкевича, В.С. Сенькіна, старшого викладача І.М. Гаркуші, асистентів Г.М. Бабенка, О.С. Заколесника, О.В. Качанова та М.А. Левченка, а також студентів К.Л. Сергеєвої, Є.П. Зацепіна та В.Ю. Воловіченка.

За 13 років, на момент 2020 року, багато випускників кафедри зуміли досягти певних посад у своїх професійних сферах. Серед них доцент К.Л. Сергеєва, асистент Д.В. Іванов, завідувач лабораторією О.В. Коробко та інженер О.В. Качанов.

У 2017 році була відкрита нова перспективна спеціальність "Інформаційні системи та технології" на кафедрі. Тоді студенти мали можливість працювати у трьох спеціалізованих комп'ютерних класах: 1/78, 1/76 та 4/55. Кафедра пропонує студентам широкий спектр сучасних дисциплін фундаментальної та загальноосвітньої спрямованості. Навчання студентів зосереджено на професійному оволодінні різноманітними інформаційними технологіями та програмними продуктами. У комп'ютерних класах студенти працюють з різними версіями операційних систем MS Windows та вивчають особливості програмних систем з відкритим кодом на базі ядер Linux. Крім того, вони оволодівають об'єктно-орієнтованими та спеціалізованими мовами програмування, такими як C/C++, Java, C#, Python, Ruby, SQL, VBA, HTML, JavaScript. Навчання умінню взаємодіяти зі складними інформаційними системами проводиться в інтегрованих середовищах розробки програм, пакетах математичного моделювання, а також у ГІС та СУБД, таких як Visual C/C++, IntelliJ IDEA, PyCharm, Netbeans, Code::Blocks, MathCAD, Matlab, ArcGIS, Oracle PL/SQL Server, MS SQL Server та інші.

Крім проведення навчальної діяльності, кафедра займається науковими дослідженнями, що стосуються проектування, розробки та впровадження

інформаційних і геоінформаційних систем в гірничо-геологічних організаціях. Одним з важливих аспектів науково-дослідної роботи кафедри є участь в міжнародних конференціях та семінарах. Кафедра має розвинуті міжнародні зв'язки з різними організаціями, такими як Німеччина, Канада, Китай, Словаччина, США, Туреччина, Казахстан та Узбекистан. Колектив кафедри активно займається виконанням держбюджетних та господарських науково-дослідних проектів [4].

1.3 Стислі відомості про технології збору та передачі інформації для об’єкта впровадження

У цьому проекті було побудовано мережу для двох будівель, а саме для першого, четвертого корпусів НТУ «ДП» (об’єднанні), і прийомної комісії Павлоградського коледжу Державного вищого навчального закладу "Національний гірничий університет" (рисунки 1.1 – 1.2).

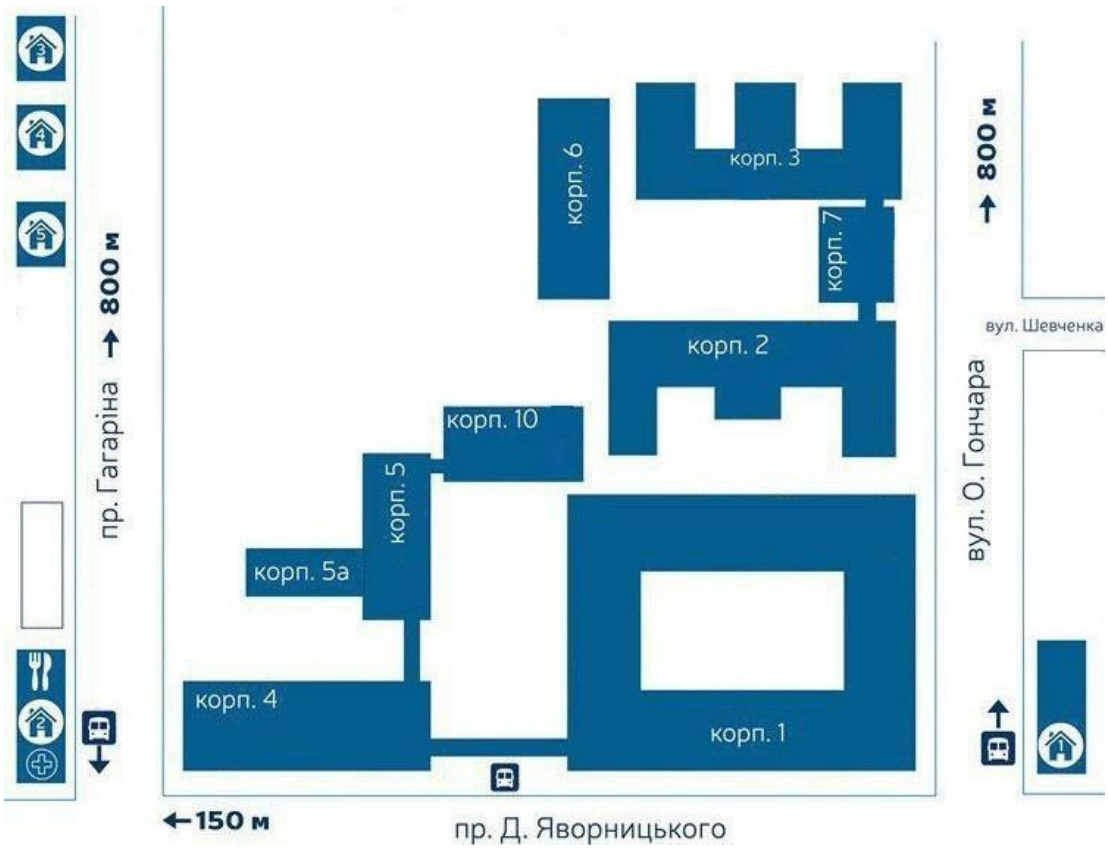

Рисунок 1.1 – Схема розміщення корпусів НТУ «ДП»

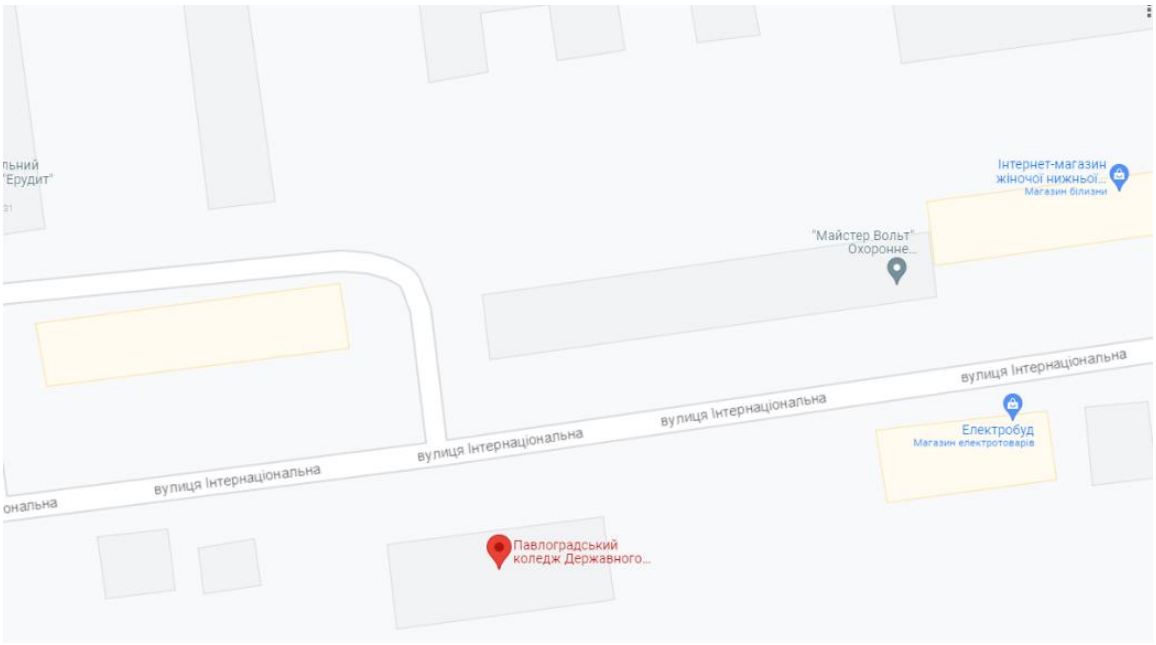

Рисунок 1.2 – Схема розміщення Павлоградського фахового коледжу

Об’єднання цих будівель одною інфраструктурою дозволить забезпечити будівлі таким як:

– поділ ресурсів, дозволе ощадливо використовувати ресурси, наприклад, управляти периферійними пристроями, такими, як принтери, зовнішні мережеві пристрої зберігання інформації, модеми і т.д. з усіх підключених робочих станцій;

– поділ даних надає можливість доступу і керування базами даних з периферійних робочих місць, що потребують інформації;

– поділ програмних засобів надає можливість одночасного використання централізованих, раніше встановлених програмних засобів;

– поділ ресурсів процесора, що забезпечує використання обчислювальних потужностей для обробки даних іншими системами, що входять в мережу;

– розрахований на багато користувачів режим - одночасне використання централізованих прикладних програмних засобів, зазвичай заздалегідь встановлених на сервері додатка.

В якості середи передачі в локальних мережах використовується «Кабель Digitus CAT 5e F-UTP, AWG 24/1, outdoor, black». Вита пара це мережевий шнур, всередині якого міститься зв'язка з декількох скручених в джгути і ізольованих провідників. Зазвичай провід має 2 або 4 пари внутрішніх жил. Їх жгутування забезпечує всім провідникам рівні в плані перешкод умови. Напруга на провідники однієї пари подається різна. За рахунок цього забезпечується інформаційний сигнал. Він як раз таки і полягає в цій різниці. Існують різні модифікації. Кожна - відрізняється калібром і діаметром провідника, а також опором і площею поперечного перерізу. ЛМ підприємства працює на стандарті Fast Ethernet.

Fast Ethernet - високошвидкісний різновид мережі Ethernet, що забезпечує швидкість передачі 100 Мбіт / с. Мережі Fast Ethernet сумісні з мережами, виконаними за стандартом Ethernet.

Стандарт визначає три типи середовища передачі для Fast Ethernet:

– 100BASE-T4 (зчетверена кручена пара);
– 100BASE-TX (здвоєна кручена пара);
– 100BASE-FX (оптоволоконний кабель).

Крім мережі, було створено, веб-додаток для серверу. Цей веб-додаток було розроблено за клієнт-серверною архітектурою. Він складається з чотирьох основних сервісів: Java REST API, Node-RED bot, Angular frontend, PostgreSQL server (рисунок 1.3).

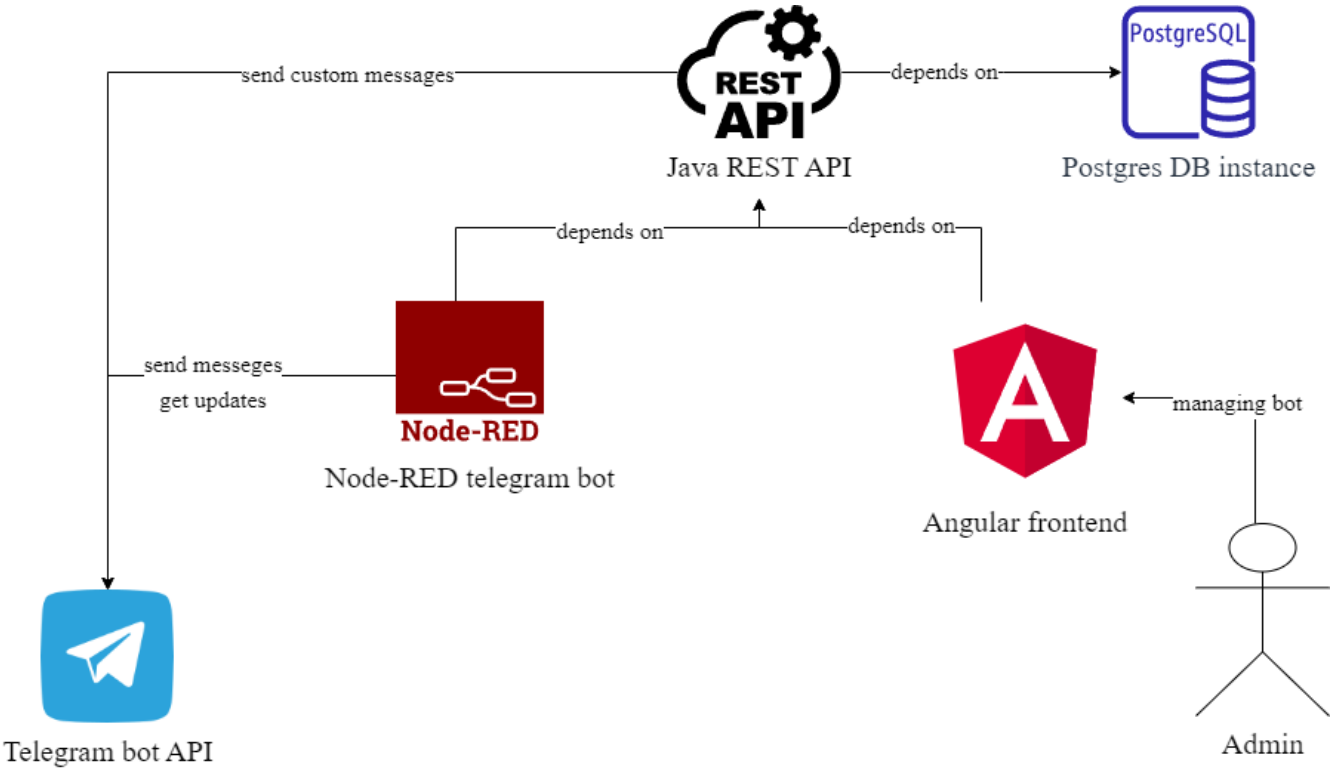

Рисунок 1.3 – Діаграма архітектури веб-додатку

1.4 Принципи, технічні способи інформаційного забезпечення об’єкта впровадження

Інформаційне забезпечення в комп'ютерних мережах включає в себе принципи, технічні способи та математичні методи, які забезпечують передачу та обробку інформації між комп'ютерами. Основною метою інформаційного забезпечення є забезпечення безперебійної та надійної комунікації між різними вузлами комп'ютерної мережі. Нижче представлені деякі ключові принципи, технічні способи та математичні методи, які використовуються для цього.

Принципи інформаційного забезпечення:

– принципи модульності – мережеве програмне забезпечення повинно бути розбите на незалежні модулі, що дозволяє розподілити роботу та спростити підтримку та розширення мережі;

– принципи стандартизації – використання загальноприйнятих стандартів для забезпечення сумісності та обміну даними між різними пристроями та мережами;

– принципи надійності – розробка механізмів, які забезпечують надійну передачу даних та виявлення помилок;

– принципи безпеки – застосування шифрування, аутентифікації та авторизації для забезпечення конфіденційності, цілісності та доступу до даних.

Технічні способи інформаційного забезпечення:

– протоколи передачі даних: наприклад, TCP (Transmission Control Protocol) – це протокол транспортного рівня, який сприяє передачі пакетів від джерела до призначення. Цей протокол вимагає встановлення з'єднання перед комунікацією між обчислювальними пристроями в мережі;

– маршрутизація – визначення оптимального шляху для передачі даних між підмережами;

– фільтрація та мережевий контроль – використання firewall, proxy-серверів та інших механізмів для контролю доступу до мережевих ресурсів та захисту від небажаного трафіку;

– виявлення та відновлення помилок – використання контрольних сум, повторних запитів та механізмів відновлення для забезпечення цілісності даних та надійної передачі.

Математичні методи інформаційного забезпечення:

– криптографія – наука про методи захисту інформації шляхом шифрування та дешифрування даних;

– теорія інформації – математична галузь, що вивчає передачу, обробку та збереження інформації;

– теорія кодування – розробка ефективних методів кодування та декодування для забезпечення надійної передачі даних через шумні канали.

1.5 Аналітичний огляд існуючих способів обробки та передачі інформації, принципів побудови об’єкта проектування, відомих рішень у галузі

Прикладом сучасної комп’ютерної мережі є Cisco Enterprise Architecture model. Ця модель полегшує проектування більших та масштабованих мереж.

При зростанні складності мереж необхідно використовувати більш модульний підхід до проектування, ніж просто ядро WAN та LAN, distribution та access рівні. Архітектура розбиває мережу на функціональні області та модулі. Ці області та модулі моделі корпоративної архітектури Cisco включають:

- область корпоративного кампусу;
- модуль корпоративного центру обробки даних;
- модуль філіалу корпорації;
- модуль дистанційних співробітників корпорації.

Cisco Enterprise Architecture model зберігає концепцію розподілу та доступу, з'єднуючи користувачів, WAN-послуги та серверні ферми через швидкі кампусні магістралі. Модульний підхід до проектування повинен бути керівником для мережевого архітектора. У менших мережах рівні можуть об'єднуватися в один рівень, навіть в один пристрій, але функції залишаються.

На рисунку 1.4 показана модель корпоративної архітектури Cisco. Область корпоративного кампусу містить інфраструктуру кампусу, що складається з ядра, distribution в будівлі та access в будівлю, з модулем центру обробки даних. Область enterprise edge складається з Інтернету, e-commerce, VPN та модулів WAN, які з'єднують підприємство з установами постачальника послуг. Область SP edge забезпечує Інтернет, загальну мережу телефонного зв'язку (PSTN) та WAN-послуги підприємству.

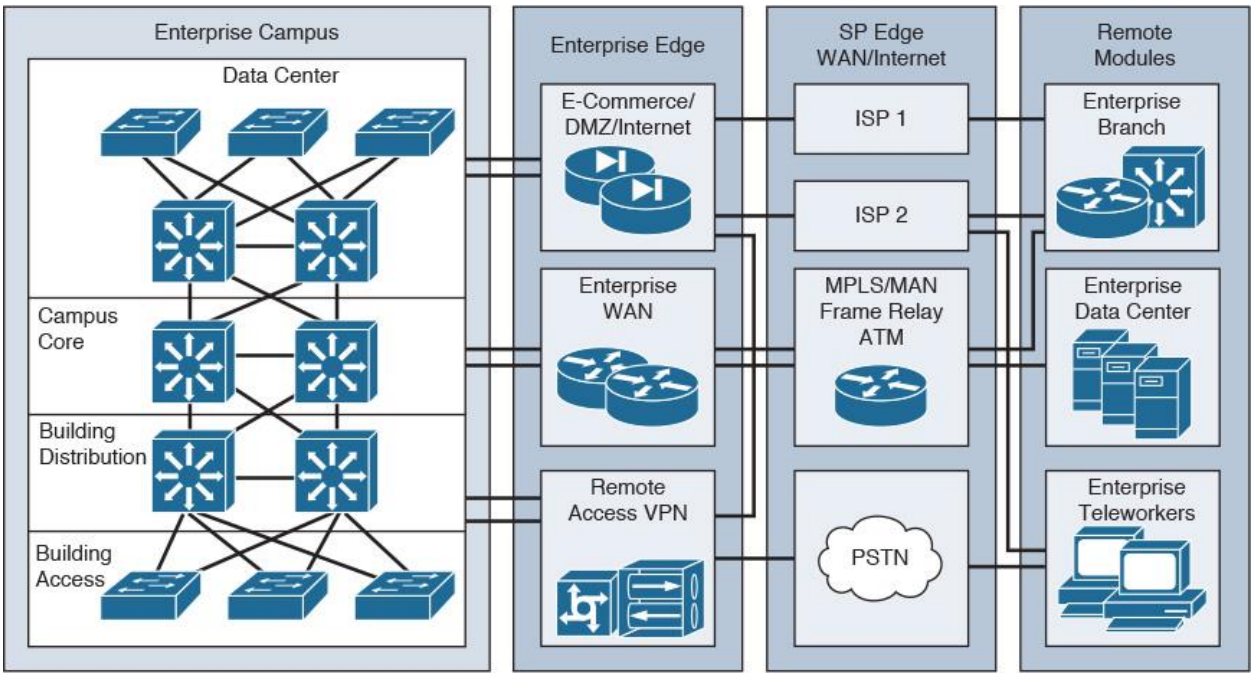

Рисунок 1.4 – Cisco Enterprise Architecture model

Сервери управління мережею розташовані в інфраструктурі кампусу, але вони пов'язані з усіма компонентами корпоративної мережі для моніторингу та управління.

Область enterprise edge з'єднується з distribution модулем enterprise edge. У невеликих та середніх місцях distribution модуль може об'єднуватися з компонентом кампусної магістралі. Він забезпечує зв'язок з вихідними послугами.

Distribution рівень мережі є точкою ізоляції між access та core рівнями мережі.

Access рівень забезпечує користувачам доступ до місцевих сегментів мережі. Access рівень характеризується комутованими сегментами LAN в кампусі. Мікросегментація за допомогою комутаторів LAN забезпечує високу пропускну здатність для робочих груп, зменшуючи кількість пристроїв на Ethernet-сегментах.

1.6 Завдання і мета роботи

Основним завданням даної кваліфікаційної роботи є проектування корпоративної мережі для кафедри інформаційних технологій та комп'ютерної інженерії в НТУ «ДП» та розробка веб-додатку "Телеграм-бот ІТКІ" для профорієнтаційної роботи.

При проектуванні корпоративної мережі були поставлені такі завдання як:

– виконати аналіз існуючих прикладів проектування мереж;

– побудувати топологію мережі;

– провести розрахунки IP-адресації для локальних підмереж за методом VLSM;

– налаштувати мережеве обладнання та кінцеві вузли (ПК та сервери);

– налаштувати безпеку мереж (port security, ACL списки доступу, підтримку служби AAA);

– налаштувати віддалений доступ до мережі через VPN.

Побудова і налаштування корпоративної мережі повинно бути виконано в Cisco Packet Tracer.

При розробці веб-додатку "Телеграм-бот ІТКІ" були поставлені наступні завдання:

– розробити сервіс, що буде обробляти запити з телеграму і надсилати відповіді користувачам;

– розробити сервіс, що буде реалізовувати CRUD-операції з даними для застосування у всьому додатку та виконувати кастомні запити до телеграму;

– розробити зручний інтерфейс взаємодії з веб-додатком через телеграм-клієнти та браузер;

– розробити панель для адміністрування веб-додатку;

– забезпечити віртуалізацію всіх сервісів для швидкого розгортання веб додатку на сервері;

– забезпечити безпеку додатку через використання хешування даних, аутентифікації, авторизації.

Весь вихідний код повинно бути надано на веб-сервісі для зберігання та спільної розробки програмного забезпечення GitHub.

1.7 Визначення можливих напрямків рішення поставлених завдань

Для вирішення завдань кваліфікаційної роботи з проектування корпоративної мережі було обрано наступну топологію мережі (рисунок 1.5).

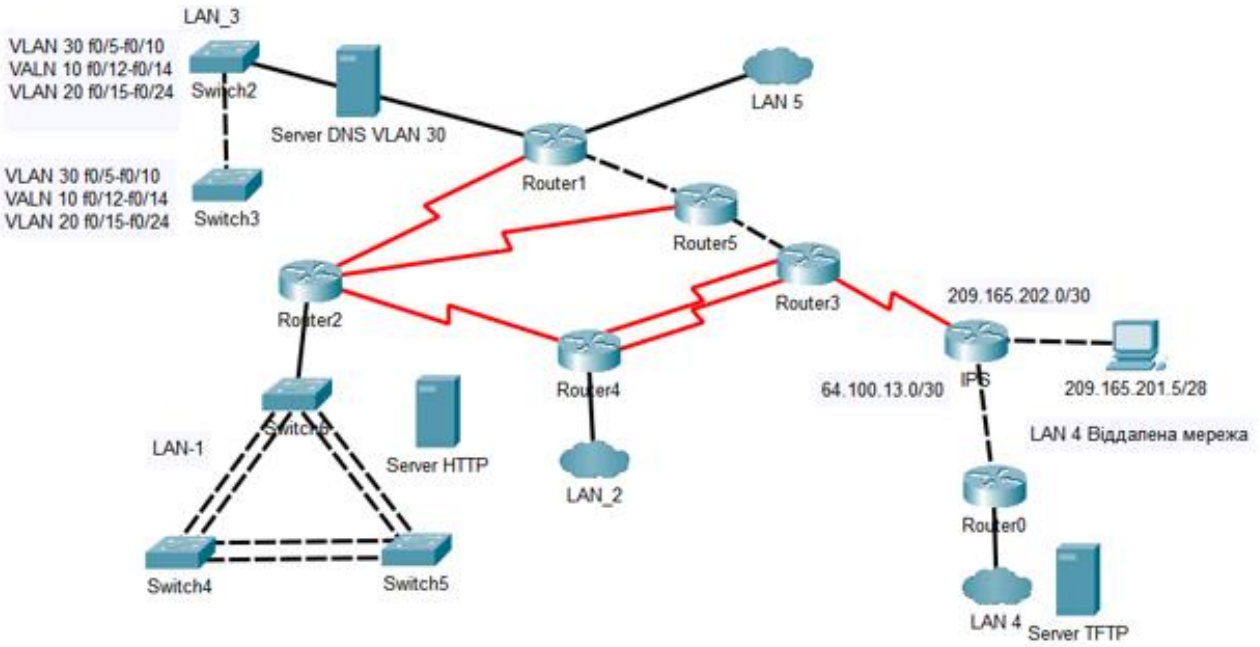

Рисунок 1.5 – Загальна топологія корпоративної мережі

Для вирішення завдань з розробки веб-додатку, було обрано такі технології як:

– Node-RED для побудови сервісу, що буде обробляти events з телеграму;

– Java з Spring Boot framework для побудови АПІ для забезпечення, аутентифікації, авторизації, доступу до даних, кастомних запитів до телеграму, хешування важливої інформації про адміністраторів додатку;

– Angular для побудови графічного інтерфейсу панелі адміністрування і взаємодії з АПІ;

– JWT (JSON Web Token) для реалізації Serverless аутентифікаї:

– PosrgreSQL, база даних для зберігання інформації, про адміністраторів, користувачів та ін;

– Docker для віртуалізації розроблених сервісів.

2 РОЗРОБКА АПАРАТНОЇ ЧАСТИНИ КОМП'ЮТЕРНОЇ СИСТЕМИ

2.1 Вимоги до системи в цілому

2.1.1 Вимоги до структури і функціонування системи

Розроблена комп'ютерна система призначена для проведення профорієнтаційної роботи кафедри ІТКІ в НТУ «ДП» (web-застосунок) та об'єднання обладнання кафедри ІТКІ в одну корпоративну мережу.

Відповідно до організаційної структури кафедри ІТКІ, прийомної комісії в НТУ «ДП» та структури віддаленої мережі, корпоративна мережа буде складатися з 5 підсистем таких як:

- підсистема прийомної комісії НТУ «ДП»;
- підсистема адміністрації кафедри ІТКІ в НТУ «ДП»;
- підсистема кабінету для викладачів в НТУ «ДП»;
- підсистема комп'ютерного класу в НТУ «ДП»;
- підсистема прийомної комісії Павлоградського коледжу.

Для кожної підсистеми корпоративної мережі потрібно створити окрему підмережу.

Базова IP-адреса для корпоративної мережі – 172.23.0.0/21.

Вимоги до максимальної кількості вузлів кожної підмережі: LAN1 – 88 вузлів, LAN2 – 7 вузлів, LAN3 – 41 вузлів, LAN4 – 85 вузлів, LAN5 –22 вузлів.

Кожна підмережа повинна мати доступ до глобальної мережі Інтернет та мережевих ресурсів корпоративної мережі.

Потрібно забезпечити доступ до віддалених мереж через глобальну мережу Інтернет за допомогою VPN-тунелю.

Сучасні web-додатки повинні забезпечувати швидкий, зручний та захищений сервіс для користувачів, незалежно від контексту веб додатку.

Структура web-додатку повинна бути гнучкою для забезпечення можливого розширення функціоналу та/або розширення кількості технічного обладнання.

Відповідно до функцій які повинен забезпечувати web-додаток він повинен складатися з таких підсистем як:

– підсистема збереження даних (БД);

– підсистема взаємодії з даними та зовнішніми сторонніми системами (API);

– підсистема взаємодії з Telegram API;

– підсистема взаємодії з адміністратором бота (графічний інтерфейс).

Всі підсистеми web-додатку повинні бути виконані як окремі проекти з використанням найсучасніших технологій у web-розробці.

Всі взаємодії між підсистемами додатку, повинні виконуватися за протоколом HTTP/HTTPS, або протоколами для взаємодії з базами даних (JDBC).

2.1.2 Вимоги до захисту від несанкціонованого доступу

Потрібно забезпечити захист підмереж від підключення пристроїв з глобальної мережі за локальними IP-адресами пристроїв (за виключенням віддалених підмереж), використовуючи ACL-списки доступу.

Потрібно забезпечити захист портів на комутаторах, що під’єднані до серверного обладнання: дозволити під’єднання до порту пристроїв тільки з двома унікальними MAC-адресами; MAC-адреси повинні визначатися динамічно; при спрацюванні обмеження, не виключати порт, а виводити відповідне повідомлення.

2.2 Вимоги до функцій, які виконує КС

Створена корпоративна мережа повинна забезпечувати такі функції як:

– передача даних між кінцевими вузлами корпоративної мережі;

– віддалений доступ за допомогою VPN;

– захист мережевого обладнання за допомогою ACL та служби AAA;

– доступ в Інтернет з будь-якої підмережі;

– доступ по глобальній IP-адресі до web-серверу.

Розроблений додаток повинен забезпечувати наступні функції:

– миттєва відповідь на команди користувача в месенджері Telegram;

– збір інформації з офіційного сайту НТУ «ДП»;

– кешування даних з API;

– керування даними API через графічний інтерфейс у браузері;

– надсилання повідомлень, файлів та фотографій усім користувачам бота в телеграмі;

– моніторинг поточної аудиторії бота;

– аутентифікація та авторизація адміністраторів.

2.3 Вимоги до видів забезпечення КС

2.3.1 Вимоги до інформаційного забезпечення КС

Розроблена корпоративна мережа використовується для з'єднання і підключення комп'ютерів або серверів, які розташовуються у тому ж будинку або декількох будівлях. Вона забезпечує доступ до однієї інформаційної системи, сприяючи швидкій передачі даних. Ця мережа дозволяє користувачам спільно обробляти дані, передавати інформацію та отримувати спільний доступ до кінцевих вузлів, мережевого обладнання та Інтернету.

Основними середовищами передачі даних у корпоративній мережі є вита пара категорії Cat 5e та Serial кабелі. Зв’язок з віддаленою мережею здійснюється через глобальну мережу Інтернет.

Корпоративна мережа буде використовуватися для забезпечення внутрішнього документообігу, проведення онлайн лекцій і занять в комп’ютерних класах, забезпечення доступу до серверів НТУ «ДП».

Корпоративна мережа повинна забезпечити розподілений доступ до налаштування мережевого обладнання через аутентифікацію з використанням RADIUS серверу та локальної аутентифікації.

Web-додаток повинен мати можливість розгортання на одному або кількох серверах та забезпечувати покращення профорієнтаційної роботи кафедри ІТКІ.

Всі компоненти комп'ютерної системи (корпоративна мережа та web-додаток) повинні відповідати міжнародним стандартам якості.

2.3.2 Вимоги до програмного забезпечення КС

Кафедра ІТКІ, займається підготовкою студентів за такими спеціальностями як «Комп'ютерна інженерія» та «Інформаційні системи та технології». Це висуває додаткові вимоги до програмного забезпечення обладнання, а саме: вимоги до інструментів взаємодії і створення баз даних, вимоги до інструментів розробки програмного забезпечення, вимоги до створення звітів та використання web-додатків.

Тому кожен ПК корпоративної мережі повинен забезпечувати:

- розробку додатків на мові Java (Intellij IDEA, Eclipse);
- розробку додатків на мові python (PyCharm);
- розробку додатків на мові C++ (Code::Blocks, MS Visual Studio);
- СУБД реляційної бази даних (MS SQL Server, MySQL Server);
- інтерфейс взаємодії з різними видами СУБД (DBeaver);
- розробку додатків на мові C# (MS Visual Studio);
- проектування і налаштування комп'ютерних мереж з обладнанням Cisco (Cisco Packet Tracer);
- роботу з електронною документацією (Microsoft Office);
- використання web-ресурсів (Google Chrome).

Для серверного обладнання на якому буде здійснено розгортання розробленого web-додатку є додаткова вимога – Docker.

Docker потрібен для скачування образів сервісів і запуску їх у ізольованому контейнері [9].

2.4 Вимоги до надійності системи

У разі збоїв у маршрутизаторах, корпоративна мережа повинна мати можливість використовувати резервні шляхи. Для забезпечення цього, потрібно налаштувати резервні маршрутизатори та динамічну маршрутизацію OSPF.

Корпоративна мережа, повинна мати резервні ліні для передачі інформації між мережами (два інтерфейси на маршрутизаторі) та в середині локальних підмереж (агрегація каналів на комутаторах).

Web-додаток повинен мати можливість кешувати дані, щоб забезпечити відповідь користувачу, навіть у відсутності доступу до розробленого API та/або web-сайту НТУ «ДП».

Web-додаток повинен обробляти блокування бота користувачами в телеграмі, видаляти їх в цьому разі і пропускати їх у черзі на відсилання повідомлення.

2.5 Вимоги до чисельності та кваліфікації персоналу

Для функціонування, кафедра ІТКІ повинна мати не менше 27 працівників педагогічного складу та не менше 1 працівника навчально допоміжного складу.

Кожен працівник педагогічного складу повинен мати, щонайменше диплом магістра у сфері технічної освіти.

Кожен працівник навчально допоміжного складу повинен мати допуск по електробезпеці не нижче третього розряду.

2.6 Розробка специфікації апаратних засобів КС

Для об’єднання кінцевих пристроїв в локальних підмережах потрібно реалізувати топологію «пасивної» зірки. Реалізувати цю топологію можна за допомогою комутатора як центрального мережевого вузла.

В якості комутатора було використано модель Cisco 2960-24TT. Цей комутатор належить до серії Cisco Catalyst 2960 і призначений для використання в невеликих або середніх мережевих середовищах. Комутатор має 24 порти Ethernet, що дозволяє підключати до 24 мережевих пристроїв одночасно. Усі 24 порти підтримують 10/100 Mbps швидкість передачі даних. Також цей комутатор має два інтерфейси Gigabit Ethernet для забезпечення швидкісного проходження трафіку між мережами для кількох хостів.

Для маршрутизації трафіку між мережами було обрано маршрутизатори Cisco ISR4331. Маршрутизатори Cisco ISR4331 спеціально розроблені для використання в корпоративних мережах і надають широкий спектр функцій і можливостей для побудови надійних і безпечних мережевих інфраструктур.

В якості ПК для робочих місць було обрано модель Dell Latitude 5420. Ця модель ноутбуку оснащена процесором Intel Core i5-1135G, екраном 14 дюймів з роздільною здатністю 1920x1080 та матрицею IPS, двома слотами для оперативної пам’яті (базовий об’єм 16 ГБ), вбудованим SSD-накопичувачем з об’ємом 512 ГБ, вбудованим акумулятором з ємністю 63 Вт*год. Ця модель Dell Latitude володіє надійною конструкцією, що відповідає вимогам для використання в освітніх закладах. Вона також має зручну клавіатуру та точпад, а також відмінну підтримку з боку Dell у разі потреби в гарантійному обслуговуванні.

В якості обладнання для web, TFTP, DNS серверів було обрано готову модель – Cisco UCS C240 M5 Rack Server. Ця модель має підтримку двох процесорів Intel Xeon Scalable з підтримкою до 28 ядер на процесор, до 3 ТБ оперативної пам'яті DDR4 ECC REG, до 24 гарячозамінних жорстких дисків формату 2,5 дюйма або 12 дисків формату 3,5 дюйма.

Таблиця 2.1 – Специфікація обладнання

Позиція	Найменування та тех. характеристика	Тип, марка, позначення документа	Одиниці виміру	Кількість
1	Cisco ISR4331 3x GE RJ-45, 4x Se	Kvasha_Router_0-5	Шт.	6
2	Cisco Catalyst 2960-24TT 24x FE RJ-45, 2x GE RJ-45	Kvasha_Switch_0-7	Шт.	8
3	Dell Latitude 5420, 14" IPS Full HD, Intel Core i5-1135G7 (4.2 ГГц), RAM 16 ГБ	PC1-35, PC0 VLAN17, PC19 VLAN17, PC20 VLAN17, PC17 VLAN17, PC38 VLAN37, PC21 VLAN27, PC20 VLAN 27, PC22 VLAN27	Шт.	43
4	Cisco UCS C240 M5 Rack Server Intel Xeon Scalable, DDR4 ECC REG	Server DNS, VLAN 37, Server HTTP, Server TFTP, Server0	Шт.	4

2.7 Вибір і обґрунтування структурної схеми комплексу технічних засобів КС

Основними складовими структури схеми комплексу технічних засобів розробленої комп’ютерної системи є мережеве обладнання (маршрутизатори та комутатори), середа передачі інформації (STP Cat 5e, UTP Cat 5e, Serial CAB 6060x), кінцеві пристрої (ПК та сервери).

Далі можна побачити структурну схему комплексу технічних засобів комп’ютерної системи з додатковим розділенням на рівень ядра та рівень доступу (Рис 2.1). На цій схемі можна побачити, що у розробленої корпоративної мережі є два ядра (для локальних підмереж НТУ «ДП» та віддаленої мережі). Ядро корпоративної мережі забезпечує маршрутизацію трафіку, налаштування безпеки та DHCP для локальних підмереж.

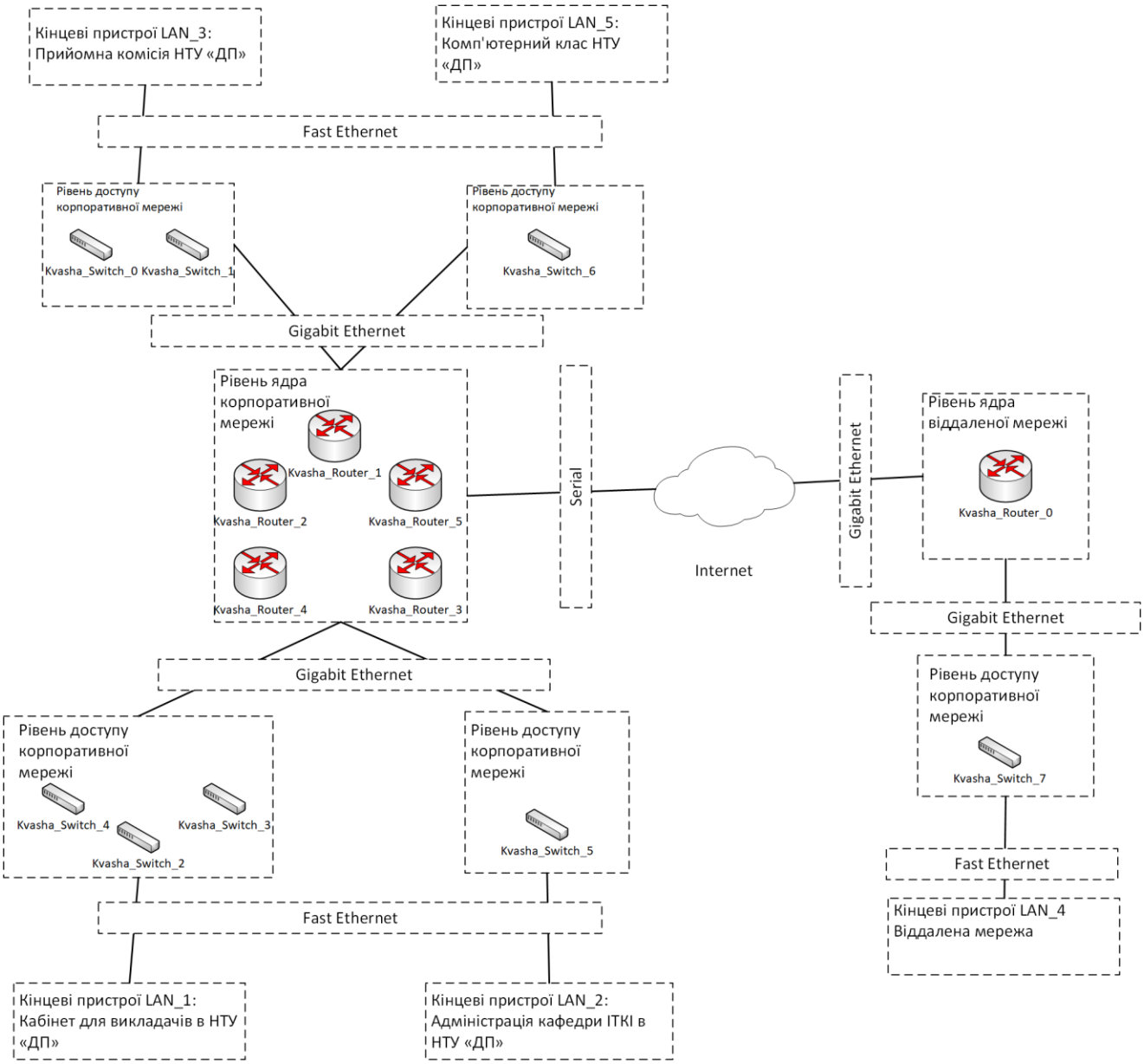

Рисунок 2.1 – Структурна схема комплексу технічних засобів КС

3 РОЗРОБКА КОРПОРАТИВНОЇ МЕРЕЖІ

3.1 Розрахунок інтенсивності трафіку вихідного трафіку найбільшої локальної мережі підприємства

Для розрахунку ключових характеристик вихідного трафіку потрібно, щоб мережа кафедри ІТКІ була завантажена на 100%.

Для об’єднання в пристроїв в найбільшій підмережі було обрано комутатори Cisco 2960-24TT та маршрутизатори Cisco ISR 4331.

У вхідних даних маємо:

– кількість вузлів в найбільшій підмережі: N = 88;

– середній показник інтенсивності трафіку: μ = 117 (кадрів/с);

– середня довжина вихідного повідомлення в найбільшій мережі: l = 650 (байт);

– кількість портів в комутаторі рівня доступу: n = 24;

– максимальна затримка передачі пакету: ≤6 мс.

З урахуванням цих даних можна розрахувати пропускну здатність мережі на рівні доступу та пропускну здатність мережі на рівні розподілу,

Пропускна здатність мережі на рівні доступу буде дорівнювати:

$$Рр.д = \mu \cdot l \cdot n \cdot 8 = 117 \cdot 650 \cdot 24 \cdot 8 = 14{,}6 \text{ (Мбіт/с)}.$$

Після цього можна розрахувати пропускну здатність цієї мережі на рівні розподілу:

$$Рр.р = \mu \cdot l \cdot N \cdot 8 = 117 \cdot 650 \cdot 88 \cdot 8 = 53{,}53 \text{ (Мбіт/с)}.$$

Комутатор рівня розподілу використовує лінію передачі з пропускною здатністю 1000 Мбіт/с. Тому загальне навантаження на комутатор не повинно перевищувати:

$$\mu вих = 1000 \cdot 1000000 / (650 \cdot 8) = 192308 \text{ (пакетів/с)}.$$

Через те, що кожне джерело в найбільшій підмережі буде виробляти 117 пакетів/с, буде невступне обмеження до приєднання до комутатора на рівні розподілу:

$$Nдж = μвих / μ=192308 / 117=1643 (джерел).$$

Цього достатньо для самої великої мережі (на 88 вузлів).

Далі для розрахунку коефіцієнта затримки, потрібно визначити інтенсивність вихідного трафіку:

$$λ = N · μ = 88 · 117 = 10296 (пакетів/с).$$

З цього, коефіцієнт затримки на рівні розподілу буде дорівнювати:

$$ρ = λ / μвих = 10296 / 192308 = 0,05.$$

Після цього можна розрахувати коефіцієнт зайнятості комутатора на рівні розподілу:

$$r = ρ / (1 - ρ) = 0,05 / (1 - 0,05) = 0,052.$$

При цьому, середня затримка кадру через чергу M/M/1 буде становити:

$$T = 1 / ((μвих - λ))=1 / (192308 - 10296) = 5,5 (мкс)$$

Далі потрібно розрахувати середню довжину черги пакетів:

$$L\_чер=ρ^2 / (1 - ρ) = (0,05)^2 / (1 - 0,05) = 0,0026.$$

Тепер можна розрахувати середній час одного пакету у черзі:

$$T\_оч = L\_чер / λ = 0,0026 / 10296 = 2,52 (мкс).$$

Середній час пакету у черзі не перевищує необхідне максимальне значення затримки пакету (6 мс).

Коефіцієнт затримки є прямо пропорційним пропускній здатності каналу і обернено пропорційним довжині кадру, тому пропускна здатність буде дорівнювати:

$$b = λ · l = 10296 · 650 · 8 = 53,54 (Мбіт/с)$$

З цих результатів можна сказати, що лінії розподілу на 1 Гбіт/с більш ніж достатньо.

3.2 Розрахунок схеми адресації корпоративної мережі

За основу адресного простору для побудови корпоративної мережі для кафедри ІТКІ в НТУ «ДП» та віддаленої мережі було узято адресу 172.23.0.0/21.

Розрахунок IP-адресації у всіх підмережах було проведено за методом VLSM. Цей метод дозволяє більш ефективно використовувати доступні IP-адреси, забезпечуючи кращу масштабованість і економію ресурсів мережі.

В Таблиці 3.1 наведено інформацію про необхідну кількість адрес для пристроїв в кожній підмережі.

Таблиця 3.1 – Мінімальна необхідна кількість вузлів для підмереж

Назва підмережі	LAN1	LAN2	LAN3	LAN4	LAN5
Кількість вузлів	88	7	41	85	22

Окрім просторів адрес для підмереж наведених в таблиці 3.1, потрібно ще створити простори для усіх зав’язків між маршрутизаторами. Всього цих зв’язків, згідно з рисунком 1.5, буде 9. Кожна з них повинна бути спроможна забезпечити щонайменше 2 вузли адресами. В якості, основи для адрес їх мереж було обрано адресу 10.1.7.0/24, де другий актет це номер мережі.

Також згідно рисунку 1.5, підмережа LAN3 повинна мати в собі чотири VLAN-мережі. Кожна з них потребує свого власного простору адрес всередині LAN3. Для цього адресу підмережі LAN3 потрібно розбити ще на 4 VLAN-мережі (VLAN 17, VLAN 27, VLAN 37, VLAN 99).

Для того щоб поділити корпоративну мережу на підмережі за допомогою методу VLSM потрібно визначити найбільшу підмережу і вибрати для неї оптимальну маску. Для спроектованої корпоративної мережі це буде LAN1 (88 вузлів). Оптимальною маскою для цієї мережі буде 255.255.255.128 (11111111.11111111.11111111.10000000), що наддасть простір для 126 хостів. Для підмережі LAN1 з адресою 172.23.0.0/25 діапазон допустимих адрес буде 172.23.0.1 - 172.23.0.126, broadcast адресою буде 172.23.0.127.

Тож далі потрібно взяти наступну найбільшу підмережу LAN4 (85 вузлів). Для неї буде наступна адреса – 172.23.0.128. Найбільш оптимальною маскою для неї буде також 255.255.255.128. Допустимим діапазоном адрес для LAN4 буде 172.23.0.129 – 172.23.0.254. Тож адресою для наступної мережі

буде 172.23.1.0. Так само потрібно проробити для всіх інших підмереж, на виході буде отримана наступна таблиця (Таблиця 3.2).

Таблиця 3.2 – Адресація в підмережах корпоративної мережі

Назва підмережі	Кількість вузлів	Адреса підмережі	Маска підмережі у десятковому форматі	Діапазон допустимих адрес	
LAN1	88	172.23.0.0	255.255.255.128	172.23.0.1	172.23.0.126
LAN4	85	172.23.0.128	255.255.255.128	172.23.0.129	172.23.0.254
LAN3	41	172.23.1.0	255.255.255.192	172.23.1.1	172.23.1.62
LAN5	22	172.23.1.64	255.255.255.224	172.23.1.65	172.23.1.94
LAN2	7	172.23.1.96	255.255.255.240	172.23.1.97	172.23.1.110
VLAN 17	14	172.23.1.0	255.255.255.240	172.23.1.1	172.23.1.14
VLAN 27	13	172.23.1.16	255.255.255.240	172.23.1.17	172.23.1.30
VLAN 37	13	172.23.1.32	255.255.255.240	172.23.1.33	172.23.1.46
VLAN 99	3	172.23.1.48	255.255.255.248	172.23.1.49	172.23.1.54
WAN1	2	10.1.7.0	255.255.255.0	10.1.7.1	10.1.7.254
WAN2	2	10.2.7.0	255.255.255.0	10.2.7.1	10.2.7.254
WAN3	2	10.3.7.0	255.255.255.0	10.3.7.1	10.3.7.254
WAN4	2	10.4.7.0	255.255.255.0	10.4.7.1	10.4.7.254
WAN5	2	10.5.7.0	255.255.255.0	10.5.7.1	10.5.7.254
WAN6	2	209.165.202.0	255.255.255.252	209.165.202.1	209.165.202.2
WAN7	2	10.7.7.0	255.255.255.0	10.7.7.1	10.7.7.254
WAN8	2	10.8.7.0	255.255.255.0	10.8.7.1	10.8.7.254
WAN9	2	10.9.7.0	255.255.255.0	10.9.7.1	10.9.7.254

3.3 Розрахунок схеми адресації пристроїв

Згідно з базовою адресою і Таблицею 3.2, загальна кількість доступних адрес становить $2^{(32 - 21)} – 2 = 2046$, всі підмережі розраховані методом VLSM можуть містити 358 вузлів, всього необхідно адрес у підмережах 243. З цього можна розрахувати відсоток адресного простору, що використовує розрахована мережа, він буде дорівнювати: 243 / 358 · 100% = 67.9%. У

випадку розрахунків за допомогою методу з маскою постійної довжини це буде 243 / 2046 · 100% = 11,9%.

Далі наведено таблицю адресації всіх маршрутизаторів в розрахованій мережі.

Таблиця 3.3 – Схема адресації маршрутизаторів

Ім’я пристрою	Інтерфейс	IP-адреса	Маска	Шлюз	VLAN	Інтерфейс підключеного пристрою
Kvasha_Router_0	Gig0/0/0	172.23.0.129	/25	-	-	Gig0/2
	Gig0/0/1	10.7.7.2	/24	-	-	Gig0/0/0
Kvasha_Router_1	Gig3/0	10.4.7.1	/24	-	-	Gig0/0/1
	Se0/0	10.1.7.2	/24	-	-	Se0/2/1
	Gig1/0	172.23.1.65	/27	-	-	Gig0/2
	Gig2/0.17	172.23.1.1	/28	-	17	Gig0/1
	Gig2/0.27	172.23.1.17	/28	-	27	Gig0/1
	Gig2/0.37	172.23.1.33	/28	-	37	Gig0/1
	Gig2/0.99	172.23.1.49	/29	-	99	Gig0/1
Kvasha_Router_3	Se0/1/1	209.165.202.1	/30	-	-	Se0/1/1
	Se0/1/1	10.8.7.2	/24	-	-	Se0/2/1
	Se0/2/1	10.9.7.2	/24	-	-	Se0/2/0
	Gig0/0/1	10.5.7.2	/24	-	-	Gig0/0/0
Kvasha_Router_4	Se0/2/1	10.3.7.2	/24	-	-	Se0/1/1
	Se0/1/1	10.8.7.1	/24	-	-	Se0/2/0
	Se0/2/0	10.9.7.1	/24	-	-	Se0/2/1
	Gig0/0/1	172.23.1.97	/28	-	-	Gig0/2
Kvasha_Router_5	Se0/1/1	10.2.7.2	/24	-	-	Se0/2/0
	Gig0/0/1	10.4.7.2	/24	-	-	Gig0/0/0
	Gig0/0/0	10.5.7.1	/24	-	-	Gig0/0/1
Kvasha_ISP	Se0/1/1	209.165.202.2	/30	-	-	Se0/1/1
	Gig0/0/1	209.165.201.1	/28	-	-	Fa0
	Gig0/0/0	10.7.7.1	/30	-	-	Gig0/0/1

Далі наведено таблицю адресації всіх комутаторів для яких було налаштовано SVI.

Таблиця 3.4 – Схема адресації комутаторів

Ім’я пристрою	Інтерфейс	IP-адреса	Маска	Шлюз	VLAN	Інтерфейс підключеного пристрою
Kvasha_Switch_0	Fa0/5	-	-	-	27	Fa0
	Fa0/6	-	-	-	27	Fa0
	Fa0/12	-	-	-	17	Fa0
	Fa0/13	-	-	-	17	Fa0
	Fa0/15	-	-	-	37	Fa0
	VLAN99	172.23.1.51	/29	-	99	-
Kvasha_Switch_1	Fa0/5	-	-	-	27	Fa0
	Fa0/6	-	-	-	27	Fa0
	Fa0/12	-	-	-	17	Fa0
	Fa0/13	-	-	-	17	Fa0
	VLAN99	172.23.1.50	/29	-	99	-
Kvasha_Switch_2	VLAN1	172.23.0.2	/25	-	1	-
Kvasha_Switch_3	VLAN1	172.23.0.3	/25	-	1	-
Kvasha_Switch_4	VLAN1	172.23.0.4	/25	-	1	-
Kvasha_Switch_5	VLAN1	172.23.1.98	/28	-	1	-
Kvasha_Switch_6	VLAN1	172.23.1.66	/27	-	1	-
Kvasha_Switch_7	VLAN1	172.23.0.130	/25	-	1	-

Далі наведено таблицю адресації кінцевих вузлів (ПК та серверів). Всі ПК отримують IP-адреси за налаштованими DHCP пулами адрес (за виключенням перших 10 можливих IP-адрес). Всі сервери маюсь статичну IP-адресу. Разом з DHCP пулами адрес, маршрутизатори також надають адреси default gateway та DNS server.

Таблиця 3.5 – Схема адресації кінцевих вузлів

Ім’я пристрою	Інтерфейс	IP-адреса	Маска	Шлюз	VLAN	Інтерфейс підключеного пристрою
PC17/20/19/0 VLAN 17	NIC	172.23.1.11 - 172.23.1.14	/28	172.23.1.1	17	Fa0/12-13
PC21/18/20/22 VLAN 27	NIC	172.23.1.11 - 172.23.1.14	/28	172.23.1.1	27	Fa0/5-6
Server DNS VLAN 37	NIC	172.23.1.46	/28	172.23.1.33	37	Fa0/15
PC36 VLAN 37	NIC	172.23.1.43 - 172.23.1.46	/28	172.23.1.33	37	Fa0/16
PC1-4/24-27	NIC	172.23.0.11 - 172.23.0.126	/25	172.23.0.1	1	Fa0/1-4
PC5-8	NIC	172.23.1.106 - 172.23.1.110	/28	172.23.1.97	1	Fa0/1-4
PC13-16/32-35	NIC	172.23.0.139 - 172.23.0.254	/25	172.23.0.129	1	Fa0/6-9, Fa0/1-4
Server TFTP	NIC	172.23.0.145	/25	172.23.0.129	1	Fa0/5
PC9-12/20-23	NIC	172.23.1.75 - 172.23.1.94	/27	172.23.1.65	1	Fa0/1-8
Server0	NIC	209.165.201.5	/28	209.165.201.1	1	Gig0/0/1

3.4 Розробка топологічної схеми корпоративної мережі

Для побудови корпоративної мережі була обрана топологія «пасивна зірка». Дана топологія має такі переваги як:

– простота установки і підключення, топологія "пасивна зірка" дуже проста у встановленні. Всі вузли підключаються до центрального комутатора або концентратора за допомогою окремих кабелів. Це полегшує процес налаштування мережі;

– легкість управління, ця топологія дозволяє централізовано керувати мережею. Через центральний комутатор або концентратор можна керувати трафіком, моніторити підключені вузли і виконувати необхідні налаштування;

– масштабованість, топологія "пасивна зірка" дозволяє додавати нові вузли до мережі без великих зусиль. Для додавання нового вузла достатньо прокласти додатковий кабель до центрального комутатора або концентратора. Це зручно у випадку, коли мережа постійно розширюється;

– висока надійність, у топології "пасивна зірка" вузли не залежать один від одного. Якщо один вузол виходить з ладу, то це не впливає на роботу інших вузлів в мережі. В разі виникнення проблеми з одним вузлом можна швидко і легко виявити і виправити її без впливу на решту мережі;

– покращена безпека, топологія "пасивна зірка" може забезпечити покращену безпеку в порівнянні з іншими топологіями. Оскільки всі дані проходять через центральний комутатор або концентратор, можна застосовувати різні методи шифрування та захисту, щоб забезпечити конфіденційність і цілісність даних.

Далі можна побачити розроблену логічну топологію (Рисунок 3.1).

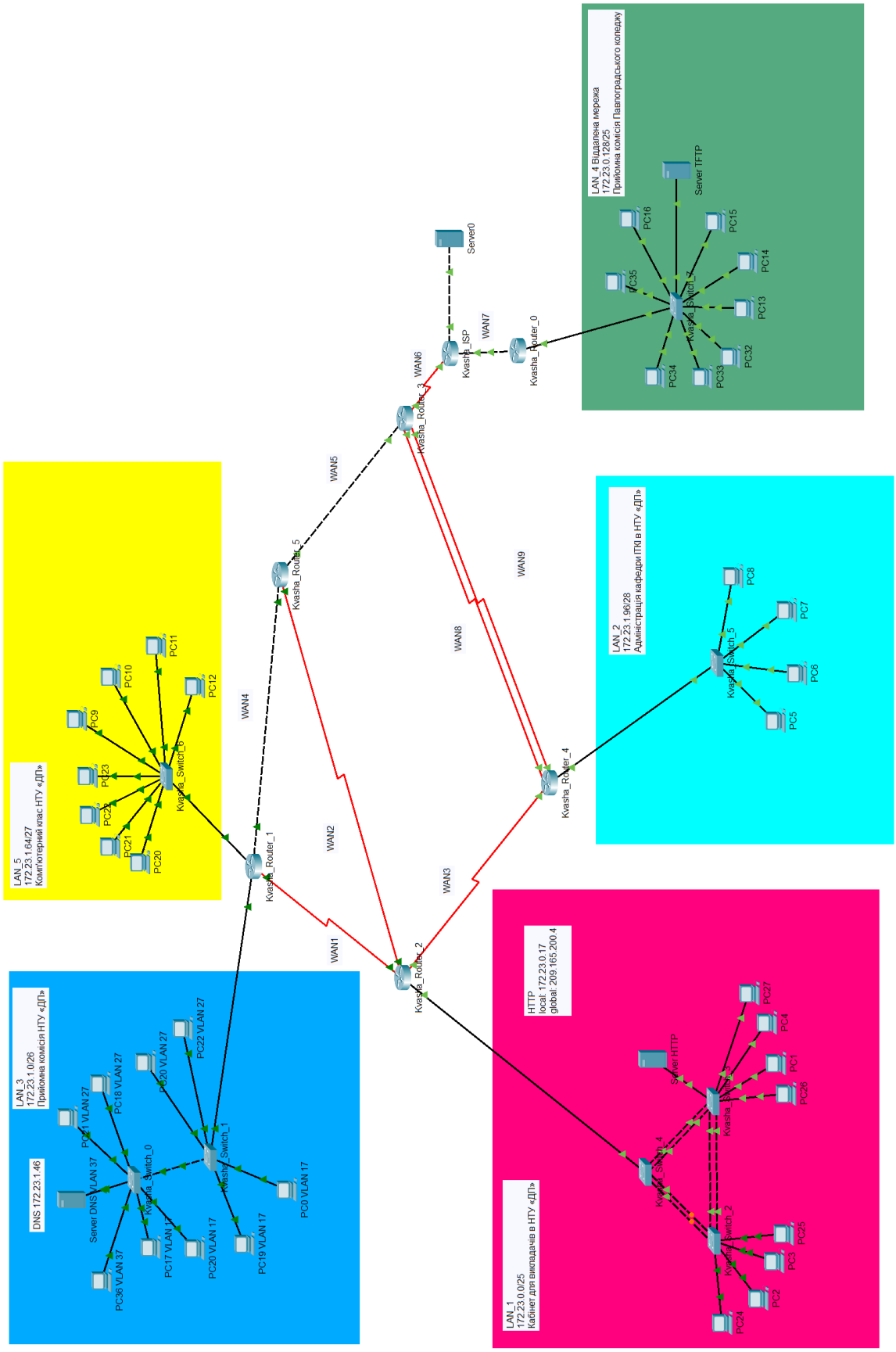

Рисунок 3.1 – Логічна топологія корпоративної мережі

Для забезпечення надійності системи, було створено дублюючі шляхи між Kvasha_Switch_2-4 та Kvasha_Router_3-4.

Крім цього розроблена логічна топологія містить повнозв’язні з’єднання між Kvasha_Router_1, Kvasha_Router_2, Kvasha_Router_5 та між Kvasha_Switch_2-4. Ці з’єднання орім покращення надійності системи надають ще менший TTL при проходженні трафіку.

Віддалена мережа пов’язана з корпоративною мережею НТУ «ДП» через глобальну мережу Інтернет за допомогою VPN-тунелю.

Спроектована комп'ютерна мережа об’єднує у собі три будівлі: два корпуси НТУ «ДП» (об’єднані) та кімнату прийомної комісії Павлоградського коледжу Державного вищого навчального закладу "Національний гірничий університет". Тому фізична топологія складається з трьох частин: другий поверх першого корпусу НТУ «ДП», перший поверх четвертого корпусу, віддалена мережа.

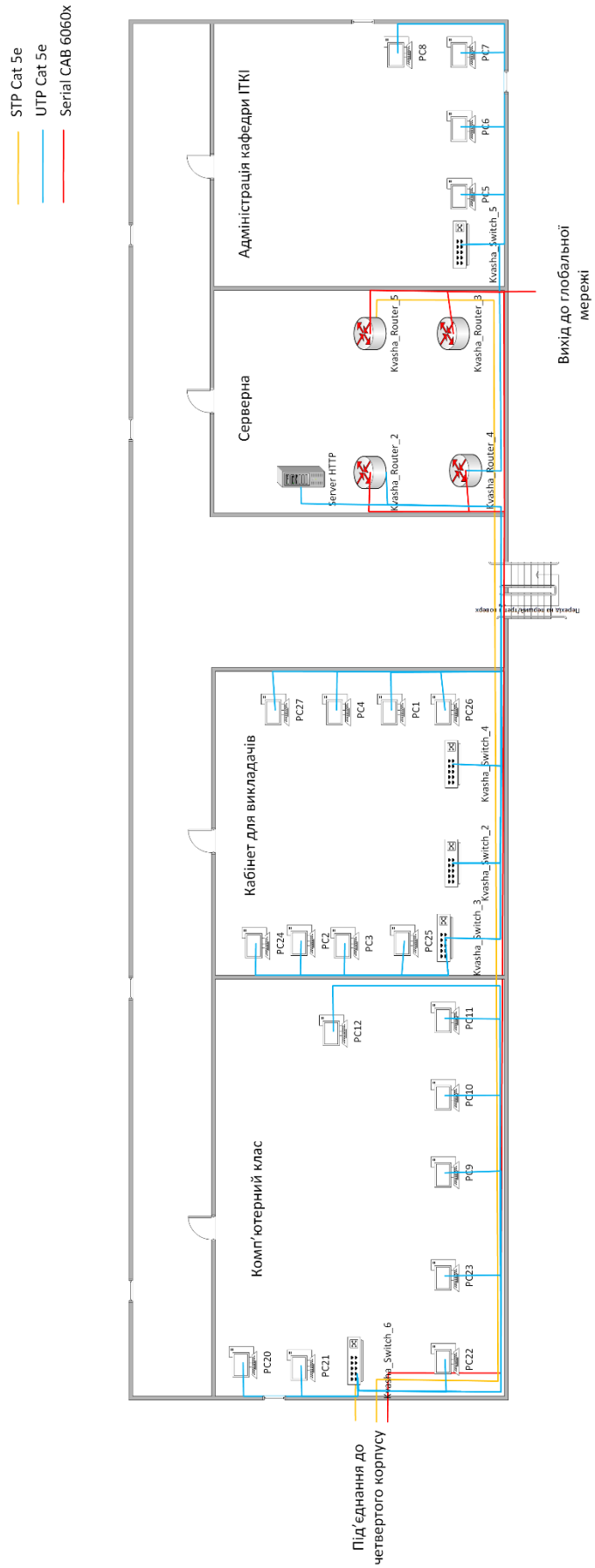

Рисунок 3.2 – Фізична топологія першого корпусу НТУ «ДП»

На Рисунку 3.2 можна побачити найбільшу частину розробленої корпоративної мережу. Вона під’єднання до мережі інтернет та четвертого корпусу. Також саме в цій топології розташовано web-сервер, що може бути використаний для розгортання web-додатків.

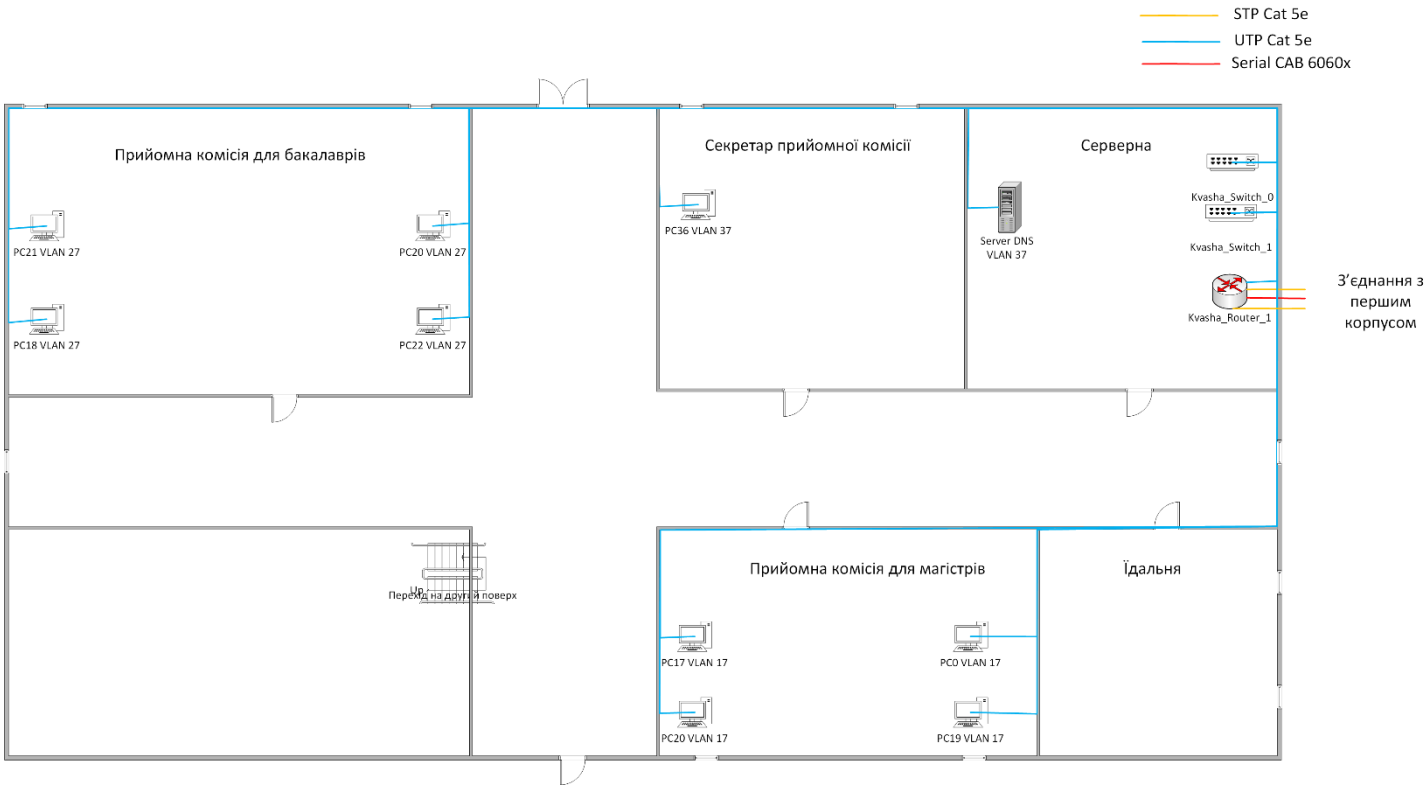

Рисунок 3.3 – Фізична топологія четвертого корпусу НТУ «ДП»

На Рисунку 3.3 можна побачити фізичну топологію прийомної комісії НТУ «ДП» (перший поверх четвертого корпусу). Саме в цій топології розташовано локальний DNS-сервер. Прямого під’єднання до глобальної мережі ця топологія не має. Для доступу до глобальної мережі, вона використовує лінії проведені до першого корпусу.

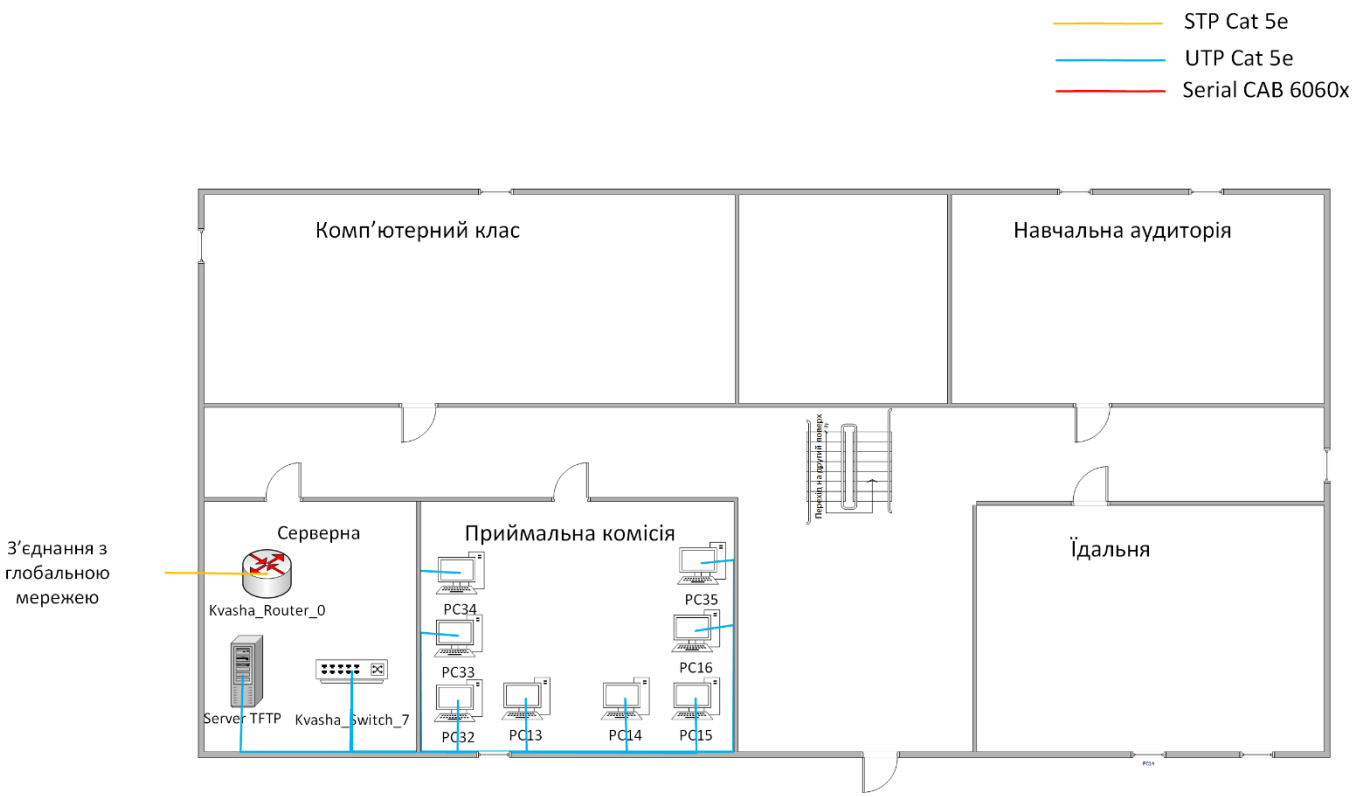

Рисунок 3.4 – Фізична топологія віддаленої мережі

На Рисунку 3.4 можна побачити фізичну топологію віддаленої мережі. Ця мережа містить TFTP-сервер. Вона розташована в іншому місті і для з'єднання з корпоративною мережею використовує VPN-тунель з IPsec.

3.5 Налаштування та перевірка роботи комп'ютерної мережі

3.5.1 Базове налаштування конфігурації пристроїв

Для всього мережевого обладнання було виконано базові налаштування. Ці налаштування включають в себе:

- налаштування назви приладу;
- налаштування назви домену (еквівалентне назві приладу);
- створення локального користувача;
- встановлення паролю на привілейований режим;
- встановлення паролю на консольний порт;
- встановлення паролю на лінії VTY;

– встановлення повідомлення в стартовому банері;

– налаштування шифрування паролів;

– налаштування SSH.

Алгоритм базових налаштувань для всього мережевого обладнання однаковий за виключенням назв пристроїв. Credentials для всіх приладів однакові.

Далі наведено приклад налаштування для Kvasha_Switch_0.

Сперш, на пристрої було налаштовано назву приладу та доменне ім’я. Доменне ім’я також є необхідним для налаштування доступу до обладнання через SSH.

Switch>enable

Switch(config)#hostname Kvasha_Switch_0

Kvasha_Switch_0(config)#ip domain-name Kvasha_Switch_0

Після цього було створено користувача з іменем 12320sk1_Kvasha та паролем admincisco. Для всіх приладів ім’я користувача та пароль однакові. Цей користувач необхідний для аутентифікації через SSH.

Kvasha_Switch_0(config)#username 12320sk1_Kvasha password admincisco

Після цього було встановлено пароль «class» на перехід до привілейованого режиму.

Kvasha_Switch_0(config)#enable password class

Далі було налаштовано пароль на лінії інтерфейсу консольного порту. Для цього було виконано три команди: перша – перехід до налаштувань потрібного інтерфейсу, друга – встановлення паролю, третя – встановлення запиту паролю при вході через цю лінію [3].

Kvasha_Switch_0(config)#line console 0

Kvasha_Switch_0(config-line)#password cisco

Kvasha_Switch_0(config-line)#login

Після цього було налаштовано пароль на всіх лініях VTY, так само як і для line console 0.

Kvasha_Switch_0(config-line)#line vty 0 15

Kvasha_Switch_0(config-line)#password cisco

Kvasha_Switch_0(config-line)#login

Далі було налаштовано повідомлення стартового банеру. Шаблон повідомлення для всіх пристроїв однаковий «Welcome to the <назва приладу>». В команді для встановлення повідомлення, початок та кінець повідомлення символізують знаки #.

Kvasha_Switch_0(config)#banner motd #Welcome to the Kvasha_Switch_0#

Після цього було налаштовано шифрування паролів, щоб не можна було переглянути паролі, наприклад через показ конфігураційного файлу пристрою.

Kvasha_Switch_0(config)#service password-encryption

Після цих налаштувань можна налаштувати SSH як транспортний протокол для під'єднання до пристрою через VTY-лінії. Перед цим потрібно згенерувати ключ за алгоритмом RSA. В якості довжини ключа було обрано 1024 біт.

Kvasha_Switch_0(config)#crypto key generate rsa

How many bits in the modulus [512]: 1024

Kvasha_Switch_0(config)#line vty 0 15

Kvasha_Switch_0(config-line)#login local

Kvasha_Switch_0(config-line)#transport in ssh

3.5.2 Налаштування маршрутизаторів корпоративної мережі

Перше, що було налаштовано на маршрутизаторах, це – підключені інтерфейси. Для того, щоб вони працювали, потрібно їх включити та налаштувати IP-адресу. Також, якщо це Serial інтерфейс тоді потрібно ще налаштувати пропускну здатність на 128Кб/с, швидкість передачі даних для синхронізації на 128000біт/с та вартість маршруту для OSPF на 7500.

Першим було налаштовано Kvasha_Router_0. Далі йде налаштування портів на цьому маршрутизаторі.

Kvasha_Router_0(config)#interface GigabitEthernet0/0/1

Kvasha_Router_0(config-if)#no shutdown

Kvasha_Router_0(config-if)#ip address 10.7.7.2 255.255.255.0

Kvasha_Router_0(config)#interface GigabitEthernet0/0/0

Kvasha_Router_0(config-if)#no shutdown

Kvasha_Router_0(config-if)#ip address 172.23.0.129 255.255.255.128

Всі IP-адреси встановлено згідно з Таблицею 3.3.

Після цього на маршрутизаторі можна налаштувати DHCP-сервер для встановлення IP-адрес підключених ПК.

Kvasha_Router_0(config)#ip dhcp pool polllan4

Kvasha_Router_0(dhcp-config)#default-router 172.23.0.129

Kvasha_Router_0(dhcp-config)# dns-server 209.165.201.5

Kvasha_Router_0(dhcp-config)#network 172.23.0.128 255.255.255.128

Kvasha_Router_0(dhcp-config)#exit

Kvasha_Router_0(config)#ip dhcp excluded-address 172.23.0.129 172.23.0.138

В цих командах було налаштовано пул адрес для DHCP-серверу згідно Таблиці 3.5. Після цього було налаштовано шлюз за замовчанням та адресу DNS-серверу. В кінці, з цього пулу було виключено 10 перших адрес.

Так само налаштування інтерфейсів було виконано для Kvasha_Router_1.

Сперш були налаштовані всі інтерфейси, окрім інтерфейсу для LAN3.

Kvasha_Router_1(config)#interface Serial0/0

Kvasha_Router_1(config-if)#clock rate 128000

Kvasha_Router_1(config-if)# bandwidth 128

Kvasha_Router_1(config-if)# ip ospf cost 7500

Kvasha_Router_1(config-if)#no shutdown

Kvasha_Router_1(config-if)#ip add 10.1.7.2 255.255.255.0

Kvasha_Router_1(config)#interface GigabitEthernet3/0

Kvasha_Router_1(config-if)#no shutdown

Kvasha_Router_1(config-if)#ip add 10.4.7.1 255.255.255.0

Kvasha_Router_1(config)#interface GigabitEthernet1/0

Kvasha_Router_1(config-if)#no shutdown

Kvasha_Router_1(config-if)#ip add 172.23.1.65 255.255.255.224

Після цього на Kvasha_Router_1 можна налаштувати DHCP пул для LAN5.

Kvasha_Router_1(dhcp-config)#ip dhcp pool polllan5

Kvasha_Router_1(dhcp-config)#default-router 172.23.1.65

Kvasha_Router_1(dhcp-config)#network 172.23.1.64 255.255.255.224

Kvasha_Router_1(dhcp-config)# dns-server 172.23.1.46

Kvasha_Router_1(dhcp-config)#exit

Kvasha_Router_1(config)#ip dhcp excluded-address 172.23.1.65 172.23.1.74

Далі так само було налаштовано порти на Kvasha_Router_2.

Kvasha_Router_2#configure terminal

Kvasha_Router_2(config)#interface GigabitEthernet0/0/1

Kvasha_Router_2(config-if)#Kvasha_Router_2(config-if)#

Kvasha_Router_2(config-if)#exit

Kvasha_Router_2(config)#interface GigabitEthernet0/0/1

Kvasha_Router_2(config-if)#no shutdown

Kvasha_Router_2(config-if)#ip address 172.23.0.1 255.255.255.128

Kvasha_Router_2(config)#interface Serial0/2/1

Kvasha_Router_2(config-if)#no shutdown

Kvasha_Router_2(config-if)#clock rate 128000

Kvasha_Router_2(config-if)# bandwidth 128

Kvasha_Router_2(config-if)# ip ospf cost 7500

Kvasha_Router_2(config-if)#ip ad 10.1.7.1 255.255.255.0

Kvasha_Router_2(config-if)#exit

Kvasha_Router_2(config)#interface Serial0/2/0

Kvasha_Router_2(config-if)#no shutdown

Kvasha_Router_2(config-if)#clock rate 128000

Kvasha_Router_2(config-if)# bandwidth 128

Kvasha_Router_2(config-if)# ip ospf cost 7500

Kvasha_Router_2(config-if)#ip address 10.2.7.1 255.255.255.0

Kvasha_Router_2(config-if)#exit

Kvasha_Router_2(config)#interface Serial0/1/1

Kvasha_Router_2(config-if)#ip address 10.3.7.1 255.255.255.0

Kvasha_Router_2(config-if)#no shutdown

Kvasha_Router_2(config-if)#clock rate 128000

Kvasha_Router_2(config-if)# bandwidth 128

Kvasha_Router_2(config-if)# ip ospf cost 7500

Після цього на Kvasha_Router_2 можна налаштувати DHCP-пул для LAN1.

Kvasha_Router_2(config)#ip dhcp pool polllan1

Kvasha_Router_2(dhcp-config)#default-router 172.23.0.1

Kvasha_Router_2(dhcp-config)#network 172.23.0.0 255.255.255.128

Kvasha_Router_2(dhcp-config)# dns-server 172.23.1.46

Kvasha_Router_2(dhcp-config)#exit

Kvasha_Router_2(config)#ip dhcp excluded-address 172.23.0.1 172.23.0.10

Після цього можна налаштувати інтерфейси для Kvasha_Router_3. Цей маршрутизатор не має неопосередкованого під’єднання до локальних мереж, тому в ньому можна не налаштовувати, DHCP-пул.

Kvasha_Router_3(config)#interface Serial0/1/1

Kvasha_Router_3(config-if)#ip address 209.165.202.1 255.255.255.252

Kvasha_Router_3(config-if)#clock rate 128000

Kvasha_Router_3(config-if)# bandwidth 128

Kvasha_Router_3(config-if)# ip ospf cost 7500

Kvasha_Router_3(config)#interface GigabitEthernet0/0/1

Kvasha_Router_3(config-if)#no shutdown

Kvasha_Router_3(config-if)#ip address 10.5.7.2 255.255.255.0

Kvasha_Router_3(config)#interface Serial0/2/0

Kvasha_Router_3(config-if)#no shutdown

Kvasha_Router_3(config-if)#ip address 10.8.7.2 255.255.255.0

Kvasha_Router_3(config-if)#clock rate 128000

Kvasha_Router_3(config-if)# bandwidth 128

Kvasha_Router_3(config-if)# ip ospf cost 7500

Kvasha_Router_3(config)#interface Serial0/2/1

Kvasha_Router_3(config-if)#no shutdown

Kvasha_Router_3(config-if)#clock rate 128000

Kvasha_Router_3(config-if)# bandwidth 128

Kvasha_Router_3(config-if)# ip ospf cost 7500

Kvasha_Router_3(config-if)#ip address 10.9.7.2 255.255.255.0

Після цього можна налаштувати порти для Kvasha_Router_4. Сперш, було налаштовано порт для LAN2.

Kvasha_Router_4(config)#interface GigabitEthernet0/0/1

Kvasha_Router_4(config-if)#no shutdown

Kvasha_Router_4(config-if)#ip ad 172.23.1.97 255.255.255.240

Після цього можна налаштувати DHCP-пул для LAN2.

Kvasha_Router_4(config)#ip dhcp pool polllan2

Kvasha_Router_4(dhcp-config)#default-router 172.23.1.97

Kvasha_Router_4(dhcp-config)#network 172.23.1.96 255.255.255.240

Kvasha_Router_4(dhcp-config)#dns-server 172.23.1.46

Kvasha_Router_4(dhcp-config)#exit

Kvasha_Router_4(config)#ip dhcp excluded-address 172.23.1.97 172.23.1.106

Після цього можна налаштувати Serial інтерфейси для Kvasha_Router_4.

Kvasha_Router_4(config)#interface Serial0/2/1

Kvasha_Router_4(config-if)#no shutdown

Kvasha_Router_4(config-if)#clock rate 128000

Kvasha_Router_4(config-if)# bandwidth 128

Kvasha_Router_4(config-if)# ip ospf cost 7500

Kvasha_Router_4(config-if)#ip address 10.1.7.2 255.0.0.0

Kvasha_Router_4(config-if)#ip address 10.3.7.2 255.255.255.0

Kvasha_Router_4(config)#interface Serial0/1/1

Kvasha_Router_4(config-if)#no shutdown

Kvasha_Router_4(config-if)#clock rate 128000

Kvasha_Router_4(config-if)# bandwidth 128

Kvasha_Router_4(config-if)# ip ospf cost 7500

Kvasha_Router_4(config-if)#ip address 10.8.7.1 255.255.255.0

Kvasha_Router_4(config-if)#exit

Kvasha_Router_4(config)#interface Serial0/2/0

Kvasha_Router_4(config-if)#ip address 10.9.7.1 255.255.255.0

Kvasha_Router_4(config-if)#clock rate 128000

Kvasha_Router_4(config-if)# bandwidth 128

Kvasha_Router_4(config-if)# ip ospf cost 7500

Після цього, можна налаштувати порти для Kvasha_Router_5.

Kvasha_Router_5(config-if)#interface Serial0/1/1

Kvasha_Router_5(config-if)#no shutdown

Kvasha_Router_5(config-if)#clock rate 128000

Kvasha_Router_5(config-if)# bandwidth 128

Kvasha_Router_5(config-if)# ip ospf cost 7500

Kvasha_Router_5(config-if)#ip add 10.2.7.2 255.255.255.0

Kvasha_Router_5(config)#interface GigabitEthernet0/0/1

Kvasha_Router_5(config-if)#no shutdown

Kvasha_Router_5(config-if)#ip add 10.4.7.2 255.255.255.0

Kvasha_Router_5(config)#interface GigabitEthernet0/0/0

Kvasha_Router_5(config-if)#no shutdown

Kvasha_Router_5(config-if)#ip add 10.5.7.1 255.255.255.0

Після цього було налаштовано інтерфейси для маршрутизатора провайдера (Kvasha_ISP).

Kvasha_ISP(config)#interface Serial0/1/1

Kvasha_ISP(config-if)#no shutdown

Kvasha_ISP(config-if)#clock rate 128000

Kvasha_ISP(config-if)# bandwidth 128

Kvasha_ISP(config-if)# ip ospf cost 7500

Kvasha_ISP(config-if)#ip address 209.165.202.2 255.255.255.252

Kvasha_ISP(config)#interface GigabitEthernet0/0/1

Kvasha_ISP(config-if)#no shutdown

Kvasha_ISP(config-if)#ip address 209.165.201.1 255.255.255.240

Kvasha_ISP(config)#interface GigabitEthernet0/0/0

Kvasha_ISP(config-if)#no shutdown

Kvasha_ISP(config-if)#ip address 10.7.7.1 255.255.255.0

Після налаштувань портів для всіх маршрутизаторів та DHCP-пулів, можна виконати налаштування динамічної маршрутизації за допомогою протоколу OSPF.

Сперш, було виконано налаштування для маршрутизації на Kvasha_Router_0. В цих командах ми повинні вказати до яких мереж підключено відповідний маршрутизатор. Також якщо є підключення до локальної мережі, то потрібно зробити цей інтерфейс пасивним, щоб зменшити кількість пакетів при обміні інформацією про шляхи між маршрутизаторами.

Kvasha_Router_0(config)#route ospf 1

Kvasha_Router_0(config-router)#network 10.7.7.0 0.0.0.255 area 0

Kvasha_Router_0(config-router)#network 172.23.0.128 0.0.0.127 area 0

Kvasha_Router_0(config-router)#passive-interface Gig0/0/0

Так само було пророблено для Kvasha_Router_1.

Kvasha_Router_1(config)#route ospf 1

Kvasha_Router_1(config-router)#network 10.0.0.0 0.255.255.255 area 0

Kvasha_Router_1(config-router)#network 172.23.1.0 0.0.0.255 area 0

Kvasha_Router_1(config-router)#passive-interface Gig2/0

Kvasha_Router_1(config-router)#passive-interface Gig1/0

В цьому прикладі маємо два інтерфейси з підключенням до локальних мереж.

Такі налаштування OSPF було прорoблено для всіх інших маршрутизаторів, згідно логічної топології (Рисунок 3.1) та таблиці адресації (Таблиця 3.2).

Окрім цього було налаштовано статичну маршрутизацію на Kvasha_Router_0, щоб можна було звертатися до Server HTTP за глобальною адресою з віддаленої мережі.

Kvasha_Router_0(config)#ip route 209.165.200.0 255.255.255.0 10.7.7.1

3.5.3 Налаштування агрегування каналів LACP

LACP (англ. Link Aggregation Control Protocol) – це протокол керування агрегацією з'єднань. Він використовується для об'єднання фізичних мережевих портів у логічні групи, що називаються агрегованими з'єднаннями або LAGs (Link Aggregation Groups). LACP дозволяє керувати і контролювати агрегацію з'єднань між мережевими пристроями, такими як комутатори.

Його потрібно налаштувати для комутаторів Kvasha_Switch_2-4.

Першим, було налаштовано Kvasha_Switch_2. Для цього потрібно вибрати потрібні інтерфейси, перевести їх у trunk-режим та розділити на групи за допомогою команди channel-group.

Kvasha_Switch_2(config)#int range fa0/21-24

Kvasha_Switch_2(config-if-range)#switchport mode trunk

Kvasha_Switch_2(config-if-range)#int range fa0/21-22

Kvasha_Switch_2(config-if-range)#channel-group 1 mode active

Kvasha_Switch_2(config-if-range)#int range fa0/23-24

Kvasha_Switch_2(config-if-range)#channel-group 2 mode active

Для інших комутаторів при розділенні на групи потрібно буде вказати відповідні номери на попередніх комутаторах.

Далі було налаштовано Kvasha_Switch_3 таким самим чином.

Kvasha_Switch_3(config)#int range fa0/21-22

Kvasha_Switch_3(config-if-range)#channel-group 1 mode passive

Kvasha_Switch_3(config-if-range)#int range fa0/23-24

Kvasha_Switch_3(config-if-range)#switchport mode trunk

Kvasha_Switch_3(config-if-range)#channel-group 3 mode active

Та налаштовано Kvasha_Switch_4.

Kvasha_Switch_4(config-if)#int range fa0/23-24

Kvasha_Switch_4(config-if-range)#channel-group 2 mode passive

Kvasha_Switch_4(config-if-range)#int range fa0/21-22

Kvasha_Switch_4(config-if-range)#channel-group 3 mode passive

3.5.4 Налаштування VLAN

Для налаштування VLAN, спершу, потрібно налаштувати sub-інтерфейсі на маршрутизаторі, що буде слугувати шлюзом для цієї мережі. Згідно логічної топології це буде Kvasha_Router_1. Для Kvasha_Router_1 були створені 3 sub-інтерфейси, номери після крапки, були виставлені згідно номеру VLAN до якого буде відноситися цей порт.

Kvasha_Router_1(config)#interface GigabitEthernet2/0

Kvasha_Router_1(config-if)#no shutdown

Kvasha_Router_1(config-if)#exit

Kvasha_Router_1(config)#interface GigabitEthernet2/0.17

Kvasha_Router_1(config-subif)#encapsulation dot1Q 17

Kvasha_Router_1(config-subif)#ip add 172.23.1.1 255.255.255.240

Kvasha_Router_1(config)#interface GigabitEthernet2/0.27

Kvasha_Router_1(config-subif)#encapsulation dot1Q 27

Kvasha_Router_1(config-subif)#ip add 172.23.1.17 255.255.255.240

Kvasha_Router_1(config-subif)#exit

Kvasha_Router_1(config)#interface GigabitEthernet2/0.37

Kvasha_Router_1(config-subif)#encapsulation dot1Q 37

Kvasha_Router_1(config-subif)#ip add 172.23.1.33 255.255.255.240

Kvasha_Router_1(config-subif)#exit

Самі VLAN повинні бути налаштовані згідно з потреб, описаних у Таблиці 3.6.

Таблиця 3.6 – Опис VLAN

Номер VLAN	Ім’я VLAN	Порт	Примітка
1	Default	-	Не використовується
17	Master	fa0/12-14	Для приймальної комісії на магістратуру
27	Bachelor	fa0/5-10	Для приймальної комісії на бакалаврат
37	Secretary	fa0/15-24	Для секретаря приймальної комісії та DNS-серверу
99	Management	SVI	Для управління пристроями
100	Native	G0/1-2	Транковый канал 802.1Q

Для налаштувань VLAN їх потрібно створити на Kvasha_Switch_0 та Kvasha_Switch_1. Після цього потрібно перевести порти, що будуть належати необхідному VLAN у режим access та призначити необхідний VLAN.

Далі йде налаштування VLAN для Kvasha_Switch_0.

З початку, VLAN були створені на комутаторі. З відповідними назвами.

Kvasha_Switch_0(config)#vlan 17

Kvasha_Switch_0(config-vlan)#name Master

Kvasha_Switch_0(config-vlan)#exit

Kvasha_Switch_0(config)#vlan 27

Kvasha_Switch_0(config-vlan)#name Bachelor

Kvasha_Switch_0(config-vlan)#exit

Kvasha_Switch_0(config)#vlan 37

Kvasha_Switch_0(config-vlan)# name Secretary

Kvasha_Switch_0(config-vlan)#exit

Kvasha_Switch_0(config)#vlan 99

Kvasha_Switch_0(config-vlan)#name Management

Kvasha_Switch_0(config-vlan)#exit

Kvasha_Switch_0(config)#vlan 100

Kvasha_Switch_0(config-vlan)#name Native

Kvasha_Switch_0(config-vlan)#exit

Після цього були налаштовані відповідні порти для VLAN.

Kvasha_Switch_0(config)#int range fa0/12-14

Kvasha_Switch_0(config-if-range)#switchport mode access

Kvasha_Switch_0(config-if-range)#switchport access vlan 17

Kvasha_Switch_0(config-if-range)#exit

Kvasha_Switch_0(config)#int range fa0/5-10

Kvasha_Switch_0(config-if-range)#switchport mode access

Kvasha_Switch_0(config-if-range)#switchport access vlan 27

Kvasha_Switch_0(config-if-range)#exit

Kvasha_Switch_0(config)#int range fa0/15-24

Kvasha_Switch_0(config-if-range)#switchport mode access

Kvasha_Switch_0(config-if-range)#switchport access vlan 37

Kvasha_Switch_0(config-if-range)#exit

Після цього можна налаштувати SVI для VLAN 99.

Kvasha_Switch_0(config)#int vlan 99

Kvasha_Switch_0(config-if)#ip add 172.23.1.51 255.255.255.248

Kvasha_Switch_0(config-if)#exit

Kvasha_Switch_0(config)#ip default-gateway 172.23.1.49

Далі потрібно налаштувати порт для VLAN100. Щоб трафік з інших VLAN міг через нього проходити.

Kvasha_Switch_0(config)#int Gig0/2

Kvasha_Switch_0(config-if)#switchport trunk native vlan 100

Kvasha_Switch_0(config-if)#switchport trunk allowed vlan 17,27,37,99,100

Такі самі налаштування, були виконані на Kvasha_Switch_1.

3.5.5 Налаштування динамічного NAT

NAT (англ. Network Address Translation) – це процес перетворення мережевих адрес (IP-адрес) між двома різними мережами. NAT широко використовується в комп'ютерних мережах, зокрема в Інтернеті, для забезпечення з'єднання між пристроями з різними локальними IP-адресами та глобальними IP-адресами.

Його потрібно налаштувати для того, щоб у кінцевих вузлів в локальних підмережах був доступ до глобальної мережі. Потрібно застосувати динамічний NAT для транслювання локальних адрес в глобальні, в межах 209.165.200.5/24 – 209.165.200.30/24. До маршрутизатору провайдеру під’єднано маршрутизатор Kvasha_Router_3. Тому саме на ньому потрібно налаштувати динамічний NAT.

Для того щоб налаштувати динамічний NAT потрібно створити пул адрес на які потрібно робити трансляцію, створити ACL список для дозволу адрес з локальних підмереж, налаштувати два інтерфейси на outside та inside. До інтерфейсів що налаштовано на inside потрібно застосувати створений пул адрес та список доступу. Далі наведено налаштування динамічного NAT на Kvasha_Router_3.

Kvasha_Router_3(config)#ip nat pool Internet 209.165.200.5 209.165.200.30 netmask 255.255.255.0

Kvasha_Router_3(config)#access-list 1 permit 172.23.0.0 0.0.255.255

Kvasha_Router_3(config)#int s0/1/1

Kvasha_Router_3(config-if)#ip nat outside

Kvasha_Router_3(config-if)#exit

Kvasha_Router_3(config)#int s0/2/1

Kvasha_Router_3(config-if)#ip nat inside

Kvasha_Router_3(config-if)#int s0/2/0

Kvasha_Router_3(config-if)#ip nat inside

Kvasha_Router_3(config-if)#int Gig0/0/1

Kvasha_Router_3(config-if)#ip nat inside

Kvasha_Router_3(config-if)#exit

Kvasha_Router_3(config)#ip nat inside source list 1 pool Internet

Для того щоб до локального HTTP серверу можна було звертатися за глобальною адресою (209.165.200.4). Потрібно налаштувати статичний NAT на локальну адресу цього серверу.

Kvasha_Router_3(config)#ip nat inside source static 172.23.0.17 209.165.200.4

Окрім цього потрібно забезпечити транслювання адрес 172.23.0.17 та 209.165.200.4 за доменним іменем «123.dnipro.ua». Для цього було налаштовано Server0 та Server DNS VLAN 37, відповідним чином (Рисунки 3.5 – 3.6).

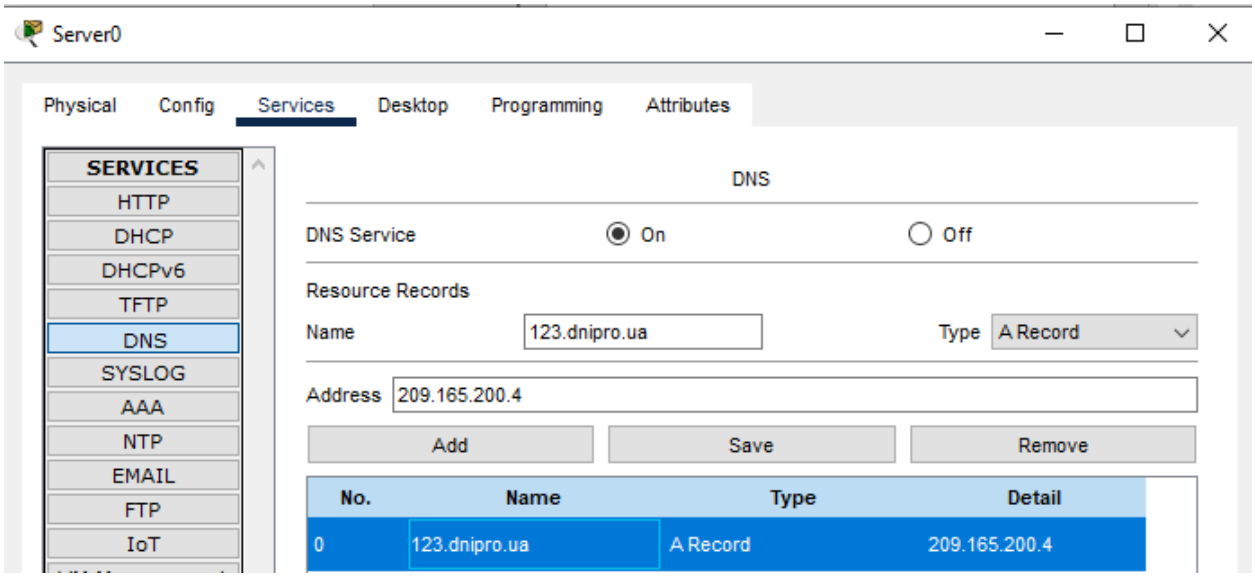

Рисунок 3.5 – Налаштування DNS на Server0

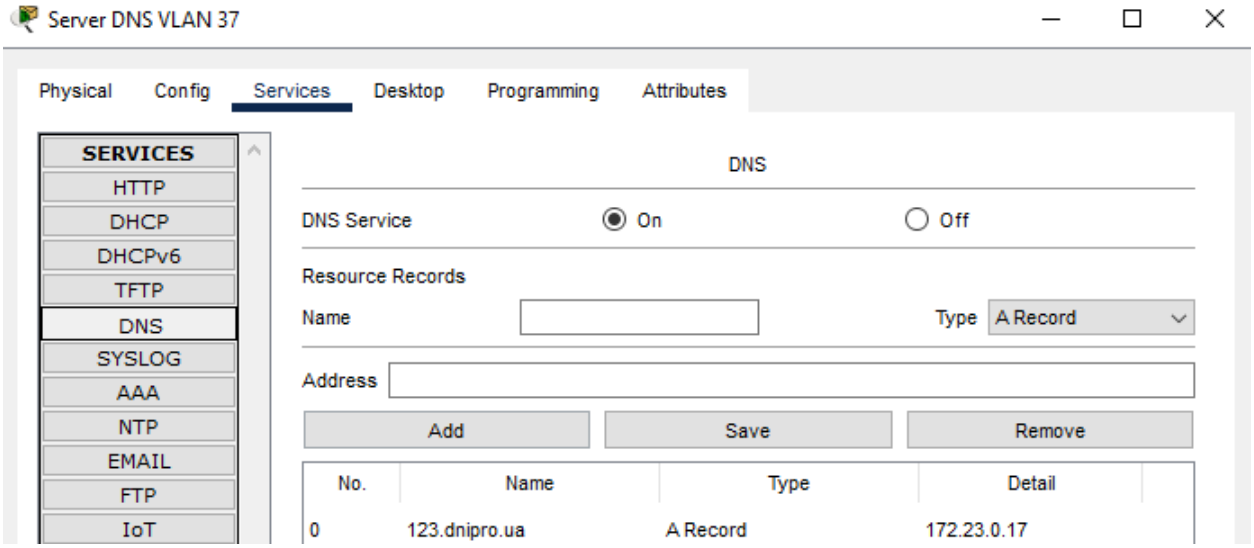

Рисунок 3.6 – Налаштування DNS на Server DNS VLAN 37

Також потрібно забезпечити трансляцію локальних адрес в глобальну з віддаленої мережі (LAN4). Для цього можна використати PAT. Він забезпечить трансляцію адрес з локальних LAN4 на глобальну маршрутизатора.

Далі наведено налаштування PAT на Kvasha_Router_0. Процес схожий з налаштуванням динамічного NAT. Основна різниця в останній команді, замість пулу адрес там надається інтерфейс, що буде overload для PAT.

```
Kvasha_Router_0(config)#ip access standard 1
Kvasha_Router_0(config-std-nacl)#permit 172.23.0.128 0.0.0.127
Kvasha_Router_0(config-std-nacl)#exit
Kvasha_Router_0(config)#int gig0/0/1
Kvasha_Router_0(config-if)#ip nat outside
Kvasha_Router_0(config-if)#int gig0/0/0
Kvasha_Router_0(config-if)#ip nat inside
Kvasha_Router_0(config-if)#exit
Kvasha_Router_0(config)#ip nat inside source list 1 int gig0/0/1 overload
```

3.5.6 Налаштування списків доступу ACL

Потрібно забезпечити захищеність мереж від доступу за локальними адресами з зовні. Це можна зробити за допомогою налаштувань ACL списків доступу. Це потрібно зробити для маршрутизаторів Kvasha_Router_0 та Kvasha_Router_3.

Далі наведено налаштування ACL для Kvasha_Router_0.

```
Kvasha_Router_0(config)#ip access extended 100
Kvasha_Router_0(config-ext-nacl)#permit ip any 10.7.7.2 0.0.0.255
Kvasha_Router_0(config-ext-nacl)#permit ospf any any
Kvasha_Router_0(config-ext-nacl)#exit
Kvasha_Router_0(config)#int gig0/0/1
Kvasha_Router_0(config-if)#ip access-group 100 in
```

В обох маршрутизаторах ці списки встановлено для зовнішніх інтерфейсів маршрутизаторів мереж.

Далі наведено налаштування ACL для Kvasha_Router_3.

```
Kvasha_Router_3(config)#ip access extended 100
Kvasha_Router_3(config-ext-nacl)#permit ip any 209.165.200.0 0.0.0.255
Kvasha_Router_3(config-ext-nacl)#permit ip 10.7.7.0 0.0.0.255 209.165.202.0 0.0.0.3
Kvasha_Router_3(config-ext-nacl)#permit ospf any any
Kvasha_Router_3(config-ext-nacl)#exit
Kvasha_Router_3(config)#int s0/1/1
Kvasha_Router_3(config-if)#ip access-group 100 in
```

3.5.7 Налаштування VPN-тунелю

Для встановлення VPN-тунелю між Kvasha_Router_3 та Kvasha_Router_0 потрібно створити ACL-список (так само як для динамічного NAT), створити та налаштувати crypto policy, налаштувати ipsec, створити crypto-map та застосувати її до зовнішнього інтерфейсу маршрутизатора.

Далі наведено налаштування Kvasha_Router_0.

Kvasha_Router_0(config)#ip access-list extended VPN

Kvasha_Router_0(config-ext-nacl)#permit ip 172.23.0.128 0.0.0.127 172.23.0.0 0.0.255.255

Kvasha_Router_0(config-ext-nacl)#exit

Kvasha_Router_0(config)#crypto isakmp policy 1

Kvasha_Router_0(config-isakmp)#encryption aes 256

Kvasha_Router_0(config-isakmp)#authentication pre-share

Kvasha_Router_0(config-isakmp)#group 1

Kvasha_Router_0(config-isakmp)#exit

Kvasha_Router_0(config)#crypto isakmp key cisco address 209.165.202.1

Kvasha_Router_0(config)#crypto ipsec transform-set VPN-IPSEC-SET esp-aes esp-sha-hmac

Kvasha_Router_0(config)#crypto map MAP 1 ipsec-isakmp

Kvasha_Router_0(config-crypto-map)#set peer 209.165.202.1

Kvasha_Router_0(config-crypto-map)#set transform-set VPN-IPSEC-SET

Kvasha_Router_0(config-crypto-map)#match address VPN

Kvasha_Router_0(config-crypto-map)#exit

Kvasha_Router_0(config)#interface GigabitEthernet0/0/1

Kvasha_Router_0(config-if)#crypto map MAP

Далі наведено налаштування Kvasha_Router_3.

Kvasha_Router_3(config)#ip access-list extended VPN

Kvasha_Router_3(config-ext-nacl)#permit ip 172.23.0.0 0.0.255.255 172.23.0.128 0.0.0.127

Kvasha_Router_3(config-ext-nacl)#exit

Kvasha_Router_3(config)#crypto isakmp policy 1

Kvasha_Router_3(config-isakmp)#encryption aes 256

Kvasha_Router_3(config-isakmp)#authentication pre-share

Kvasha_Router_3(config-isakmp)#group 1

Kvasha_Router_3(config-isakmp)#exit

Kvasha_Router_3(config)#crypto isakmp key cisco address 10.7.7.2

Kvasha_Router_3(config)#crypto ipsec transform-set VPN-IPSEC-SET esp-aes esp-sha-hmac

Kvasha_Router_3(config)#crypto map MAP 1 ipsec-isakmp

Kvasha_Router_3(config-crypto-map)#set peer 10.7.7.2

Kvasha_Router_3(config-crypto-map)#set transform-set VPN-IPSEC-SET

Kvasha_Router_3(config-crypto-map)#match address VPN

Kvasha_Router_3(config-crypto-map)#exit

Kvasha_Router_3(config)#interface Serial0/1/1

Kvasha_Router_3(config-if)#crypto map MAP

Окрім цього потрібно змінити списки доступу для NAT з урахуванням VPN.

Далі налаштування на Kvasha_Router_0.

Kvasha_Router_0(config)#ip access-list extended 101

Kvasha_Router_0(config-ext-nacl)#deny ip 172.23.0.128 0.0.0.127 172.23.0.0 0.0.255.255

Kvasha_Router_0(config-ext-nacl)#permit ip 172.23.0.128 0.0.0.127 any

Kvasha_Router_0(config-ext-nacl)#exit

Kvasha_Router_0(config)#no ip nat inside source list 1 interface GigabitEthernet0/0/1 overload

Kvasha_Router_0(config)#ip nat inside source list 101 interface GigabitEthernet0/0/1 overload

Так само і для Kvasha_Router_3.

Kvasha_Router_3(config)#ip access-list extended 101

Kvasha_Router_3(config-ext-nacl)#deny ip 172.23.0.0 0.0.255.255 172.23.0.128 0.0.0.127

Kvasha_Router_3(config-ext-nacl)#permit ip 172.23.0.0 0.0.255.255 any

Kvasha_Router_3(config-ext-nacl)#no ip nat inside source list 1 pool Internet

Kvasha_Router_3(config-ext-nacl)#ip nat inside source list 101 pool Internet

3.5.8 Налаштування служби AAA

Налаштування аутентифікації за допомогою служби AAA було виконано на всіх маршрутизаторах, окрім маршрутизатора у віддаленій мережі. Потрібно надати можливість аутентифікації за допомогою служби AAA по лінії консолі, якщо така аутентифікація недоступна, застосувати локальні credentials. Аутентифікація по лініях VTY повинна бути тільки по локальних credentials.

Для налаштування служби AAA потрібно на маршрутизаторі створити новий спосіб аутентифікації за AAA та спосіб з локальними даними за замовчанням. Застосувати ці способи до інтерфейсів line con 0 та VTY-ліній згідно попередньо описаної умови. Та налаштувати radius-сервер на маршрутизаторі: дані для аутентифікації на сервері.

Далі наведено налаштування служби AAA на Kvasha_Router_1.

```
Kvasha_Router_1(config)#aaa new-model
Kvasha_Router_1(config)#aaa authentication login default local
Kvasha_Router_1(config)#aaa authentication login radius-auth group radius local
Kvasha_Router_1(config)#line con 0
Kvasha_Router_1(config-line)#login authentication radius-auth
Kvasha_Router_1(config-line)#exit
Kvasha_Router_1(config)#line vty 0 15
Kvasha_Router_1(config-line)#login authentication default
Kvasha_Router_1(config-line)#exit
Kvasha_Router_1(config)#radius host 172.23.0.17 key radius123
```

Окрім налаштувань на маршрутизаторах, потрібно налаштувати сам radius-сервер з відповідними даними для аутентифікації (як на ньому так і облікові дані самих користувачів). В якості radius-серверу було обрано Server HTTP.

Далі можна побачити налаштування radius-серверу.

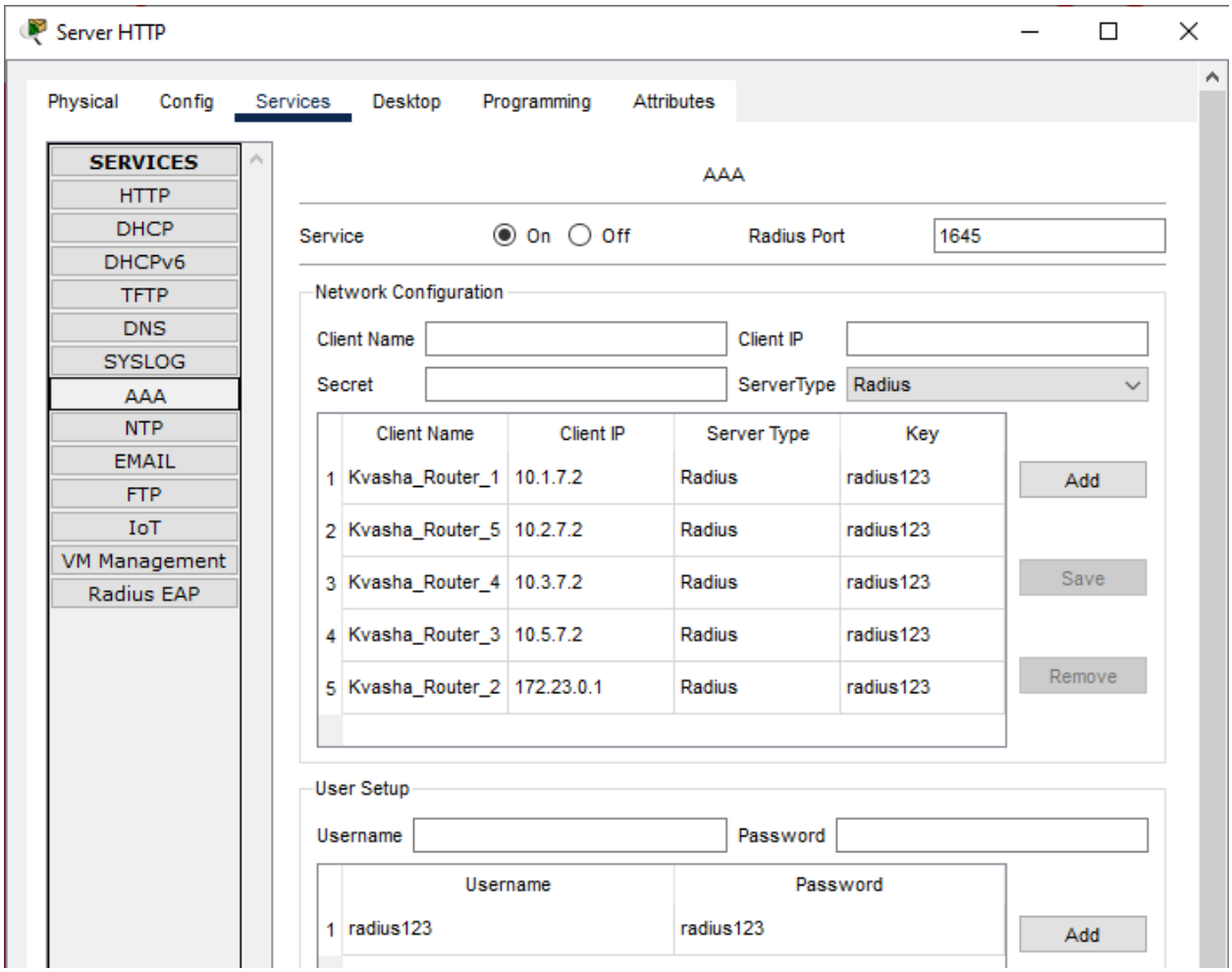

Рисунок 3.1 – Налаштування AAA на Server HTTP

На Рисунку 3.7 можна побачити всі маршрутизатори, що можуть увійти до серверу та облікові дані користувачів (пароль та username – radius123).

3.5.9 Налаштування безпеки комутаторів

Потрібно налаштувати захист портів на всіх комутаторах, що підключені до серверів (Kvasha_Switch_0, Kvasha_Switch_3, Kvasha_Switch_7). Комутатор повинен запам’ятовувати MAC-адреси і дозволити підключення максимум перших двох пристроїв з унікальними адресами. Якщо підключається третій прилад, потрібно виводити повідомлення про це і не дозволяти проходити трафіку (порт не виключати).

Далі можна почити налаштування Kvasha_Switch_0.

Kvasha_Switch_0(config)#interface FastEthernet0/15

Kvasha_Switch_0(config-if)#switchport port-security

Kvasha_Switch_0(config-if)#switchport port-security maximum 2

Kvasha_Switch_0(config-if)#switchport port-security mac-address sticky

Kvasha_Switch_0(config-if)#switchport port-security violation restrict

Kvasha_Switch_0(config-if)#do show port-security interface fa0/15

Так само були налаштовані наступні маршрутизатори, за виключенням портів, що підключені до серверів.

3.6 Перевірка роботи КС

Перше, що варто перевірити, це роботу OSPF. Це можна перевірити пінгуванням вузлів у різних мережах (розташованих не в межах одного маршрутизатора) та командою show ip ospf int brief. Ця команда покаже визначені шляхи на маршрутизаторі.

Далі можна побачити успішне знаходження шляхів на Kvasha_Router_3.

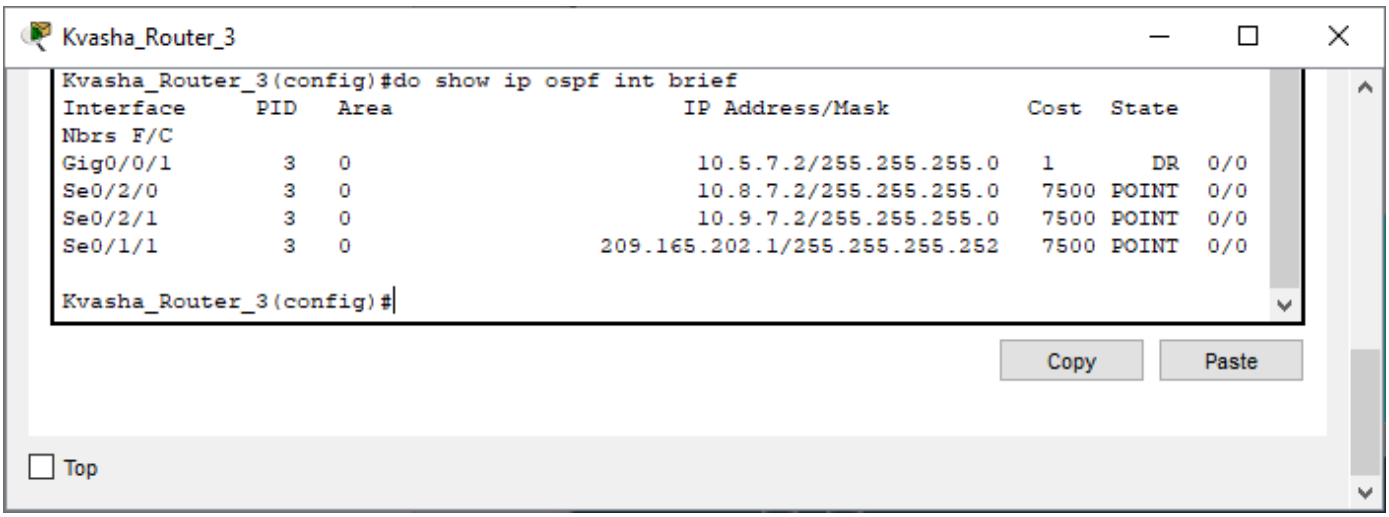

Рисунок 3.8 – Перевірка OSPF

Також можна перевірити VPN-з’єднання між Kvasha_Router_3 та Kvasha_Router_0 командою sh crypto isakmp sa. Як можна побачити на наступному рисунку, з’єднання є і воно активне.

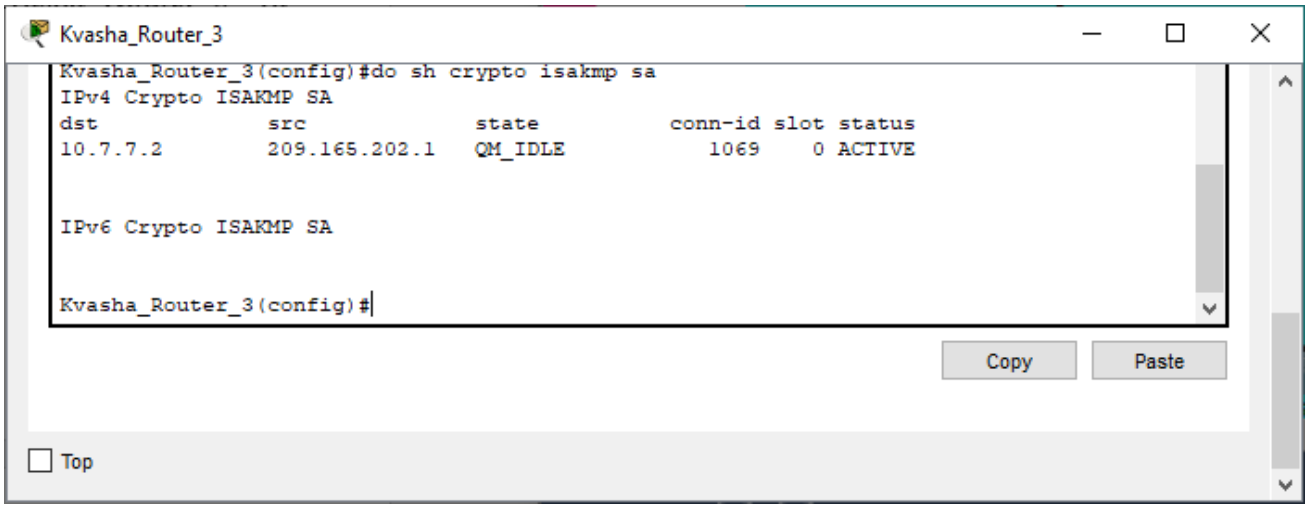

Рисунок 3.9 – Перевірка VPN

Port security можна перевірити за допомогою команди show port-security interface <назва підключеного інтерфейсу до сервера> (потрібно щоб попередньо сервер надіслав пакет через цей порт).

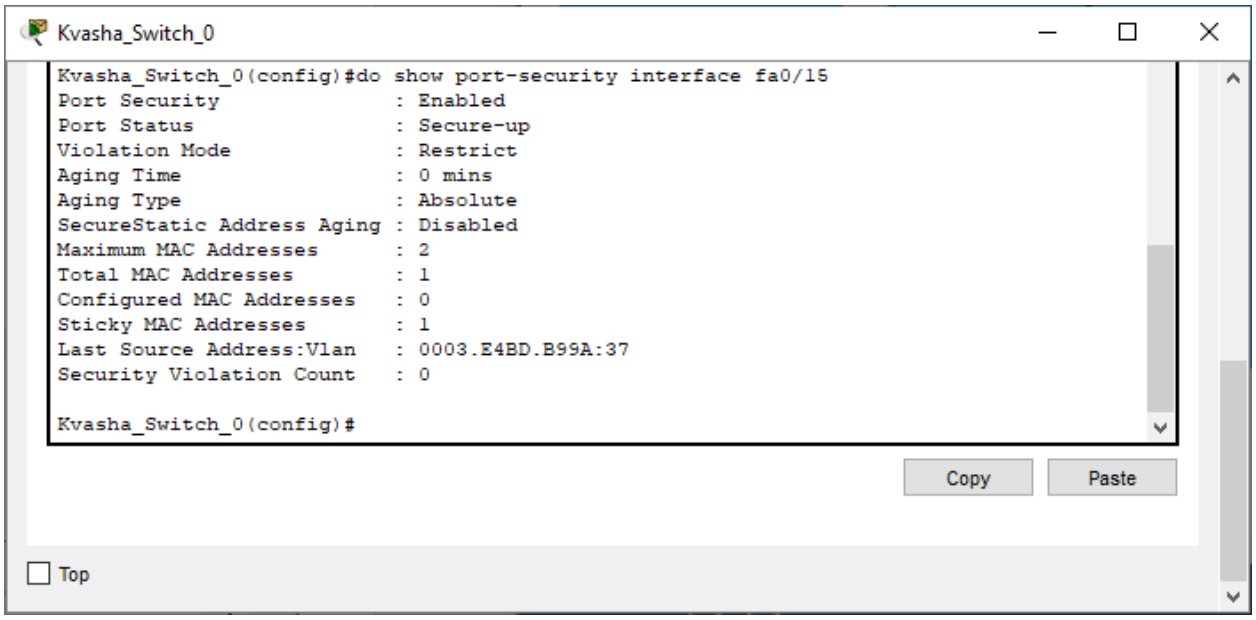

Рисунок 3.10 – Перевірка Port security

Як можна побачити (Рис. 3.10), комутатор запам’ятав MAC-адресу.

Роботу служби AAA можна перевірити за допомогою аутентифікації у маршрутизації за допомогою даних з серверу через консольну лінію.

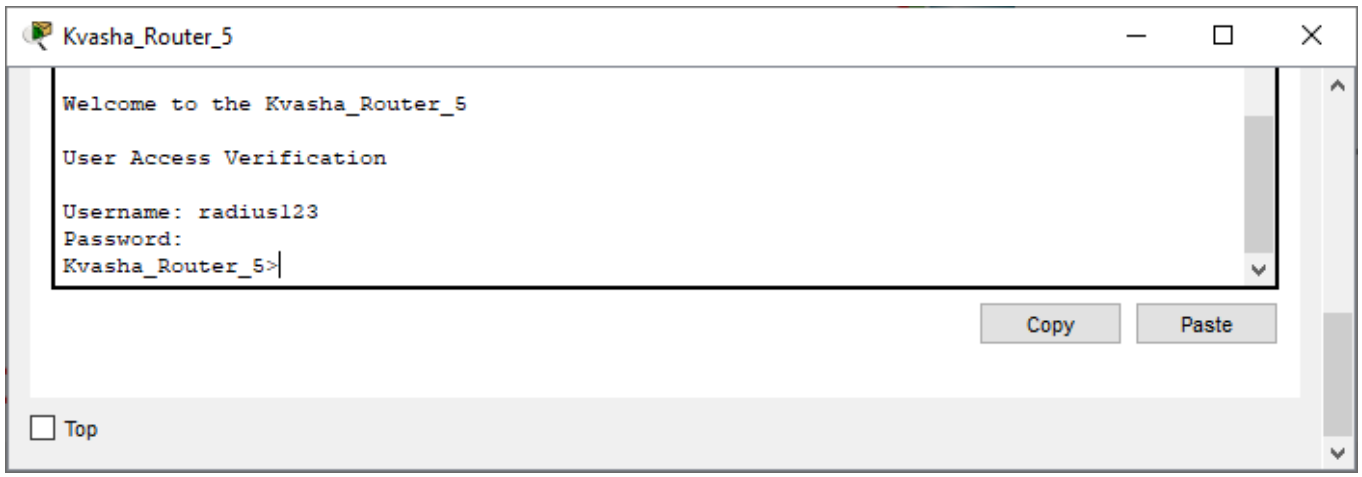

Рисунок 3.11 – Перевірка AAA

Як можна побачити аутентифікація успішно пройшла.

Далі потрібно перевірити зв’язок різних вузлів в різних підмережах.

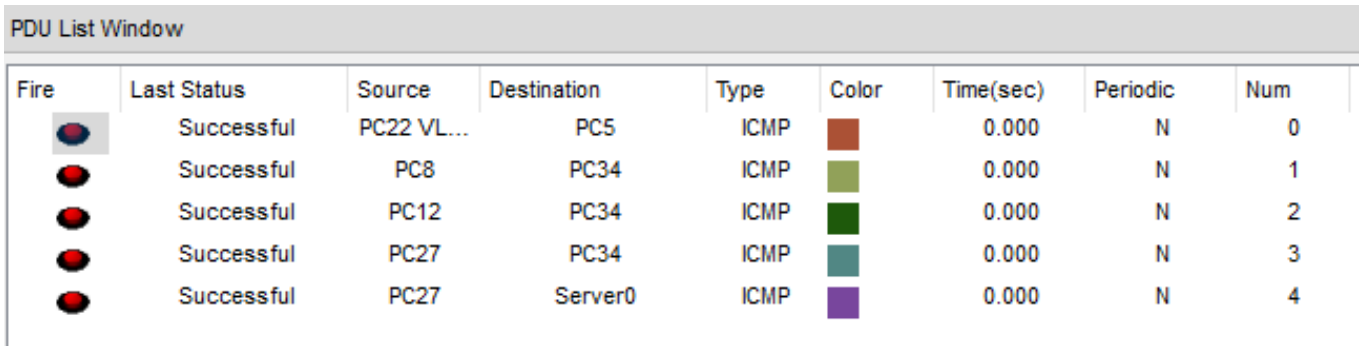

Рисунок 3.12 – Перевірка проходження трафіку

Як можна побачити на Рисунку 3.12, трафік проходить успішно між усіма підмережами.

Також можна перевірити роботу DNS-серверів та Server HTTP. Для цього потрібно відкрити HTML сторінку з будь-якого ПК за адресою.

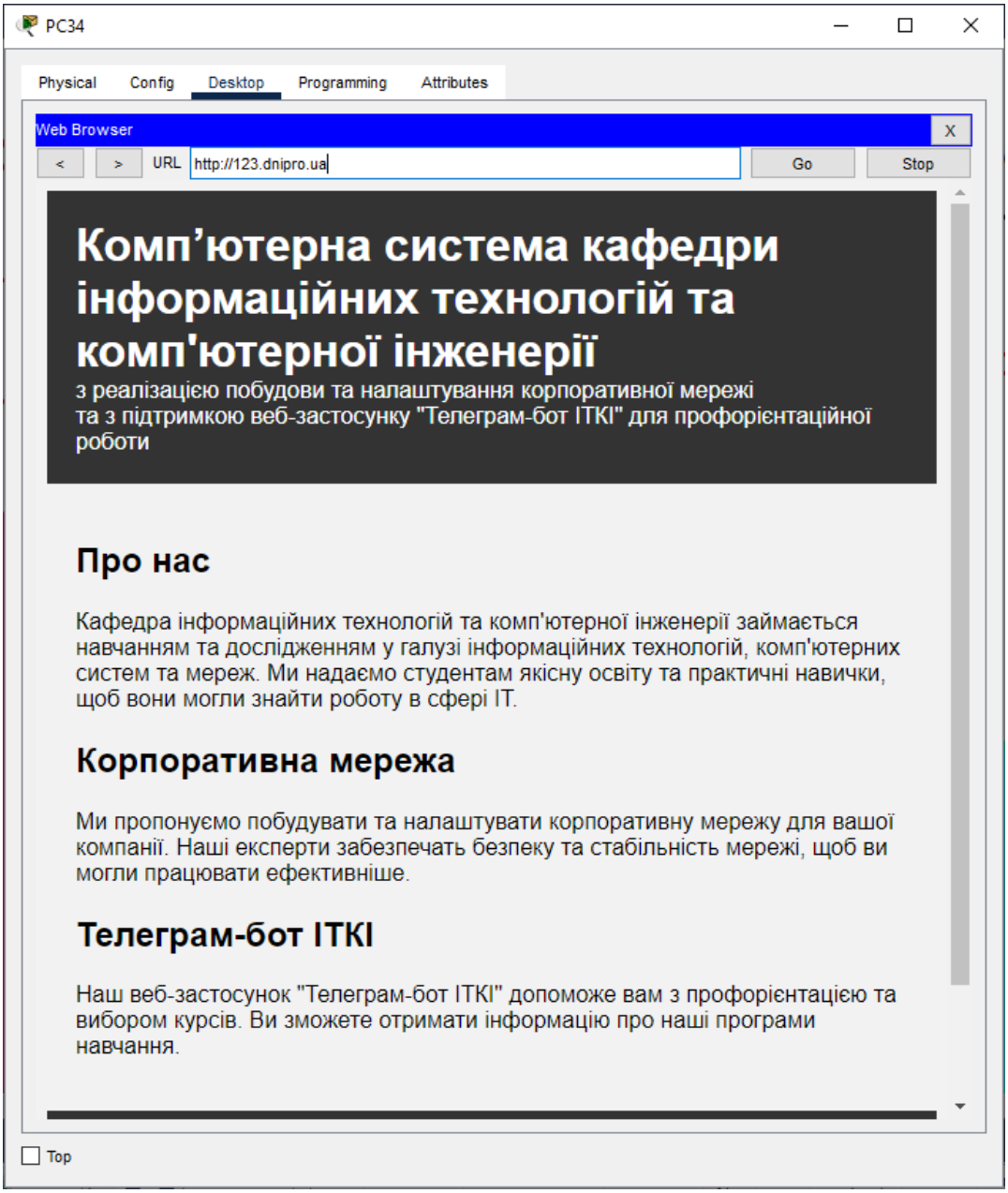

Рисунок 3.13 – Перевірка web-серверу

Як можна побачити (Рис. 3.13), всі сервери працюють коректно. Source code сторінки, надано в Додатку А.

4 РОЗРОБКА СТРУКТУРИ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

4.1 Призначення області застосування ПЗ

Розроблене програмне забезпечення призначене для виконання на серверах (комп'ютерах з білою адресою).

За рахунок контейнеризації (Docker), у розробленого додатку немає особливих вимог до системи на якій він буде розгортатися, окрім вимог до об'єму ресурсів системи (об'єму основної пам'яті, часу процесора та ін.).

Розгортання розробленого додатку розраховане на WEB-сервері з публічною адресою з локальної мережі LAN1 (розділ 1), але може бути замінене на AWS EC2, AWS ECS або схожі сервіси від інших IT-компаній (heroku, MS та ін.).

Також можливе розгортання без контейнеризації на кількох serverless ресурсах (AWS Lambda). Але такий підхід має значні недоліки такі як: перезапуск окремого сервіса, складність налаштування і оновлення додатку (потрібно підтримувати одразу кілька окремих сервісів).

При використанні одного серверу для розгортання додатку, рекомендується використовувати Kubernetes.

Kubernetes (або K8s) – це відкрите програмне забезпечення для автоматизації розгортання, масштабування та керування контейнерними додатками. В основі Kubernetes лежить ідея оркестрування контейнерів, що дозволяє автоматизувати процеси розгортання та керування додатками в контейнерах.

4.2 Обґрунтування технічних характеристик

Розроблений додаток, має обмеження по завантаженню файлів і тексту (розмір файлу – не більше 50МБ, розмір запиту з файлами – не більше 500МБ, розмір тексту – не більше 4000 символів).

Всі ці обмеження продиктовані Telegram Bot API.

Сервіс API у розробленому додатку, створений з використанням Java Spring Boot, що забезпечує велику відмовостійкість системи і Angular (самий продуктивний frontend framework). А за рахунок створення Stateless API, також можливе розгортання кількох веб додатків одночасно (наприклад для різних зон).

Для підвищення продуктивності Angular UI, бажано не використовувати development server, що йде за замовчанням з angular-cli, а спершу зібрати додаток у статичні файли верстки і розгорнути їх на окремому nginx server (вже реалізовано в розробленому додатку, команда для зібрання – «make build»).

NGINX – це веб-сервер та проксі-сервер з відкритим вихідним кодом. Він може виконувати роль веб-сервера, який обслуговує веб-сторінки та інші ресурси, а також проксі-сервера, який забезпечує пересилання запитів між різними серверами [10].

4.3 Опис розробленої програми

Розроблений додаток складається з кількох окремих сервісів, що взаємодіють між собою за протоколом HTTP (рисунок 1.3).

Java API було розроблене за REST методологією з використанням SOLID принципів. Воно складається з 68 кастомних класів і інтерфейсів, отримати доступ до діаграми класів можна за посиланням https://viewer.diagrams.net/?tags=%7B%7D&highlight=0000ff&edit=_blank&layers=1&nav=1&title=itki-bot-api.drawio#Uhttps%3A%2F%2Fdrive.google.com%2Fuc%3Fid%3D1HbwMwuLBnvjK1iIFojLOPI0-hdo321vC%26export%3Ddownload.

Java REST API разом з собою запускає систему міграції баз даних Liquibase. Liquibase створює таблиці і заповнює початкову інформацію (початкового користувача, ролі). Всі зміни реалізовані у вигляді changelog.

Далі можна переглянути ER-діаграму бази даних після змін.

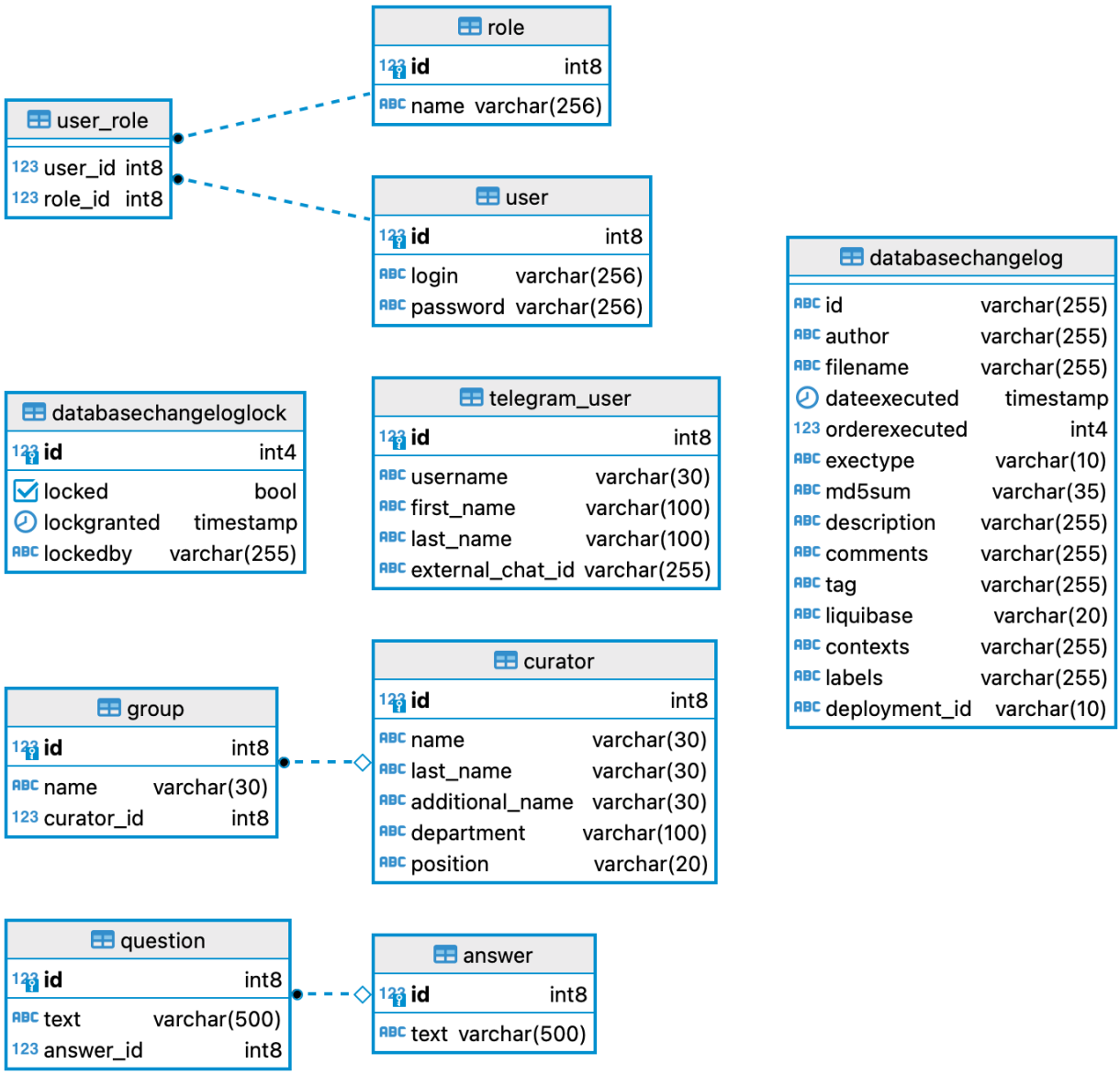

Рисунок 4.1 – ER-діаграма БД

На Рисунку 4.1 можна побачити окрім кастомних таблиць, ще дві стандартні, що створюються Liquibase: databasechangelog, databasechangelock.

REST (Representational State Transfer) – це архітектурний стиль для проектування розподілених систем, який використовується для створення веб-сервісів. REST вважається одним з найбільш популярних підходів до розробки веб-сервісів.

Основні принципи REST включають в себе:

– клієнт-серверна архітектура: система складається з клієнтів, які відправляють запити, та серверів, які обробляють ці запити та відповідають на них;

– безстанність: кожен запит від клієнта містить достатньо інформації для сервера, щоб обробити запит, тому сервер не зберігає стан клієнта між запитами;

– кешування: клієнти та сервери можуть кешувати відповіді на запити, щоб зменшити час відповіді на повторні запити.

– єдина точка входу: система повинна мати єдину точку входу для доступу до ресурсів;

– можливість розширення: Система повинна бути легко розширюваною, дозволяючи додавати нові функції та ресурси без необхідності зміни існуючих;

– використання стандартних протоколів: REST використовує стандартні протоколи, такі як HTTP та HTTPS, для обміну даними.

Stateless стан АПІ було забезпечено за рахунок використання аутентифікації через JWT токен. За рахунок цього, один сервер може прийняти користувача від іншого перевіривши токен підписаний іншим сервером.

JWT (або JSON Web Token) – це стандарт, який використовується для передачі даних між сторонами у вигляді JSON об'єктів. JWT складається з трьох частин: заголовка, корисного навантаження та підпису [8].

Заголовок містить тип токена та алгоритм підпису, який використовується для підпису корисного навантаження.

Корисне навантаження містить інформацію, яку потрібно передати між сторонами. Це може бути інформація про користувача, що аутентифікується, або будь-яка інша інформація, яку потрібно передати.

Підпис створюється за допомогою алгоритму підпису, вказаного у заголовку. Це додається для того, щоб сторони могли перевірити цілісність токена та переконатися, що дані не були змінені під час передачі.

Для взаємодії з БД, розроблене АПІ використовує фреймворк Hibernate і систему міграції БД Liqubase.

Liquibase – це інструмент для керування версіями баз даних. Він дозволяє розробникам контролювати структуру бази даних та зміни в схемі бази даних зі збереженням історії змін та можливістю відкату до попереднього стану.

Hibernate – це відкрите програмне забезпечення для об'єктно-реляційного відображення (ORM), яке дозволяє розробникам взаємодіяти з реляційною базою даних (RDBMS) у відповідності з об'єктно-орієнтованим підходом до програмування.

Hibernate забезпечує спрощений доступ до даних з бази даних і дозволяє розробникам працювати з об'єктами даних, замість того, щоб створювати складні запити SQL, забезпечуючи тим самим високий рівень абстракції.

Всі паролі користувачів зберігаються у захешованому вигляді (як на АПІ так і в Node-RED). Для хешування паролів, АПІ використовує Bcrypt хеш-функцію.

Bcrypt – це хеш-функція для шифрування паролів, яка є однією з найбільш безпечних на сьогодні. Його основна мета - захистити паролі користувачів від зламування та витоку даних.

Bcrypt генерує хеш шляхом застосування функції хешування до пароля з випадковою "сіллю" (salt), що додається до пароля перед хешуванням. Сіль - це випадкова рядок символів, який додається до пароля перед хешуванням, щоб ускладнити процес підбору паролю.

В АПІ реалізована авторизація за рахунок надання кожному користувачу ролі (ADMIN та USER).

Основним елементом взаємодії з кінцевим користувачем (не адміністратором бота) є Node-RED telegram bot.

Node-RED telegram bot приймає запити від користувачів (опитує телеграм АПІ) і надсилає відповіді на команди.

RED Bot – це розширення для Node-RED, яке надає можливість створювати та керувати chat-ботами з Node-RED. Він дає можливість створювати ботів для різних платформ, таких як Slack, Telegram, Facebook Messenger, інтерфейси голосових помічників і багато інших.

RED Bot дозволяє легко налаштувати ботів за допомогою візуального інтерфейсу Node-RED, де можна складати різні функції та обробки повідомлень. Він підтримує створення діалогових потоків, зберігання станів ботів та керування взаємодією з користувачами.

RED Bot має також розширену функціональність, яка дозволяє використовувати сервіси штучного інтелекту, такі як IBM Watson або Google Dialogflow, для збагачення функціональності ботів. Завдяки цьому RED Bot може бути потужним інструментом для створення чат-ботів різного рівня складності та функціональності.

Node-RED telegram bot отримує всі дані з розробленого REST API і веб-сайту університету. Всі данні бот підтягає за кастомним розкладом і кешує в основній пам’яті.

Основний інструмент кастомізації розробленого бота – це UI частина, написана на Angular.

При першому зверненні він вимагає аутентифікації користувача, після аутентифікації в розробленому АПІ він отримує два токени (token та refreshToken). Перший токен, надає можливість користуватися АПІ, другий – надає можливість подовжувати сесію, тобто користувач (адмін) може бути потенційно нескінченно в системі без необхідності перезаходити.

Кожен сервіс в розробленому додатку представляє собою окремий самостійний проект з власним стеком технологій.

Репозиторій розробленого додатку розташований за посиланням https://github.com/Noct2000/itki-bot

4.4 Робота з програмою

Для взаємодії з ботом є два основних способи: Перший це взаємодія як абітурієнт (або студент). Користувач заходить в телеграм додаток, переходить за посиланням створеного боту, для розробки було використано https://t.me/ITKI_bot. Після цього бот його вітає, і говорить як його можна використовувати. Основні функції користувача реалізовані за допомогою команд (рисунок 2.1), окрім функції допомоги.

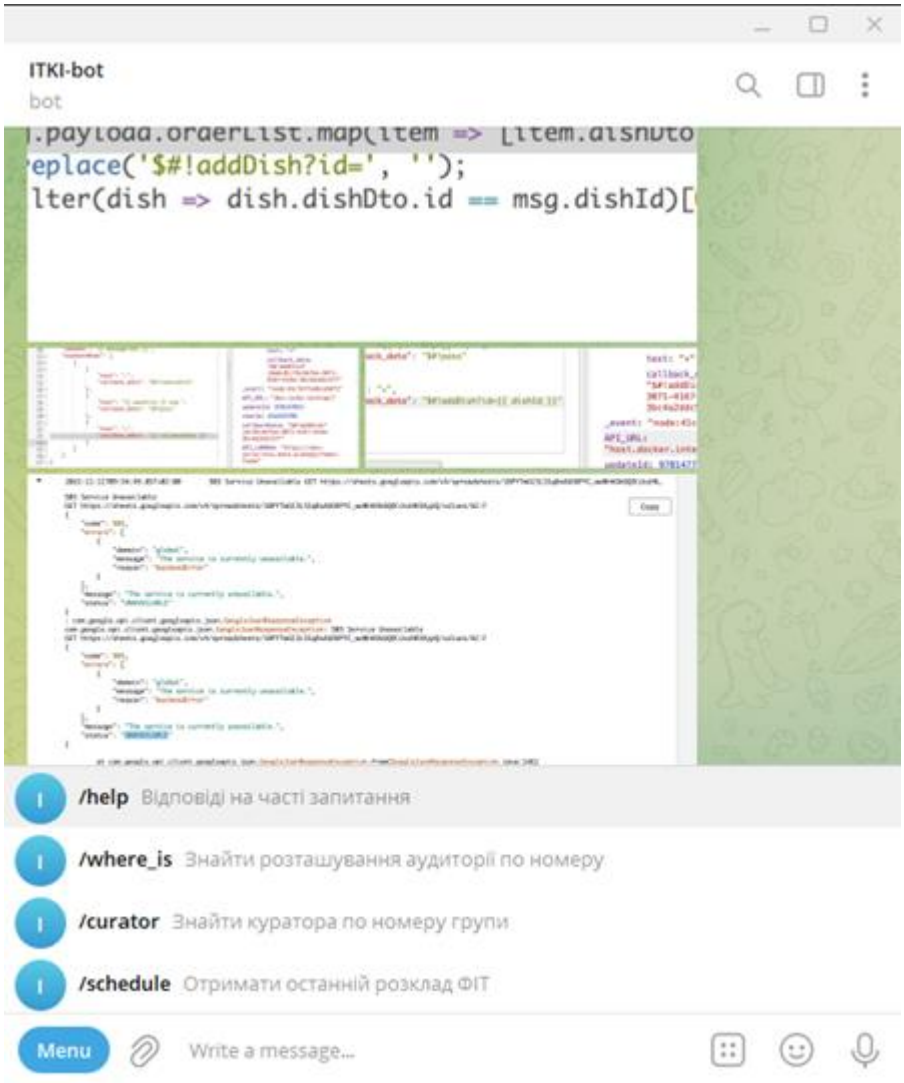

Рисунок 4.1 – Меню команд для бота ITKI

Для команд можливе автозаповнення для забезпечення більш user friendly інтерфейсу.

Функція допомоги, реалізована по іншому принципу (не команд). Вона реалізована через окрему клавіатуру, яку можна змінювати через інструменти адміністрування. Інформацію, в бота можна записати будь яку, він все оновить самостійно за своїм розкладом.

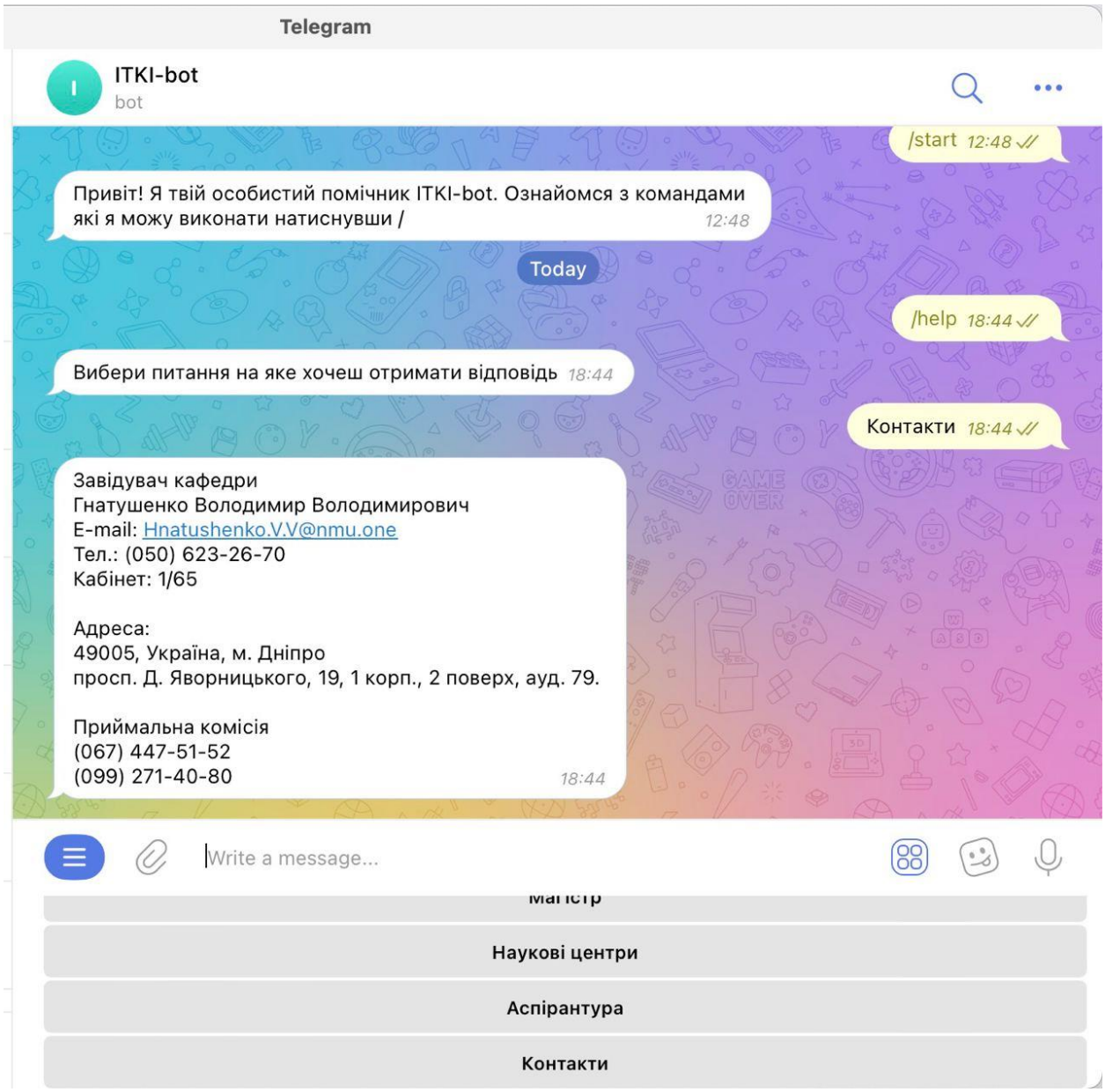

Рисунок 4.2 – Клавіатура запитань для бота ITKI

Клавіатура також має ще корисну особливість – при натисканні на кнопку, буде відправлятися повідомлення з текстом цієї кнопки до боту, тому користувач крім клавіатури, ще може ввести своє питання самостійно без переходу до клавіатури і виклику допомоги.

Окрім взаємодії в якості абітурієнту (або студенту), з ботом також можна взаємодіяти через власний UI, що було створено на Angular.

Для відрисовки елементів UI було використано графічну бібліотеку NG-Zorro, що забезпечує сучасний і красивий вигляд додатку.

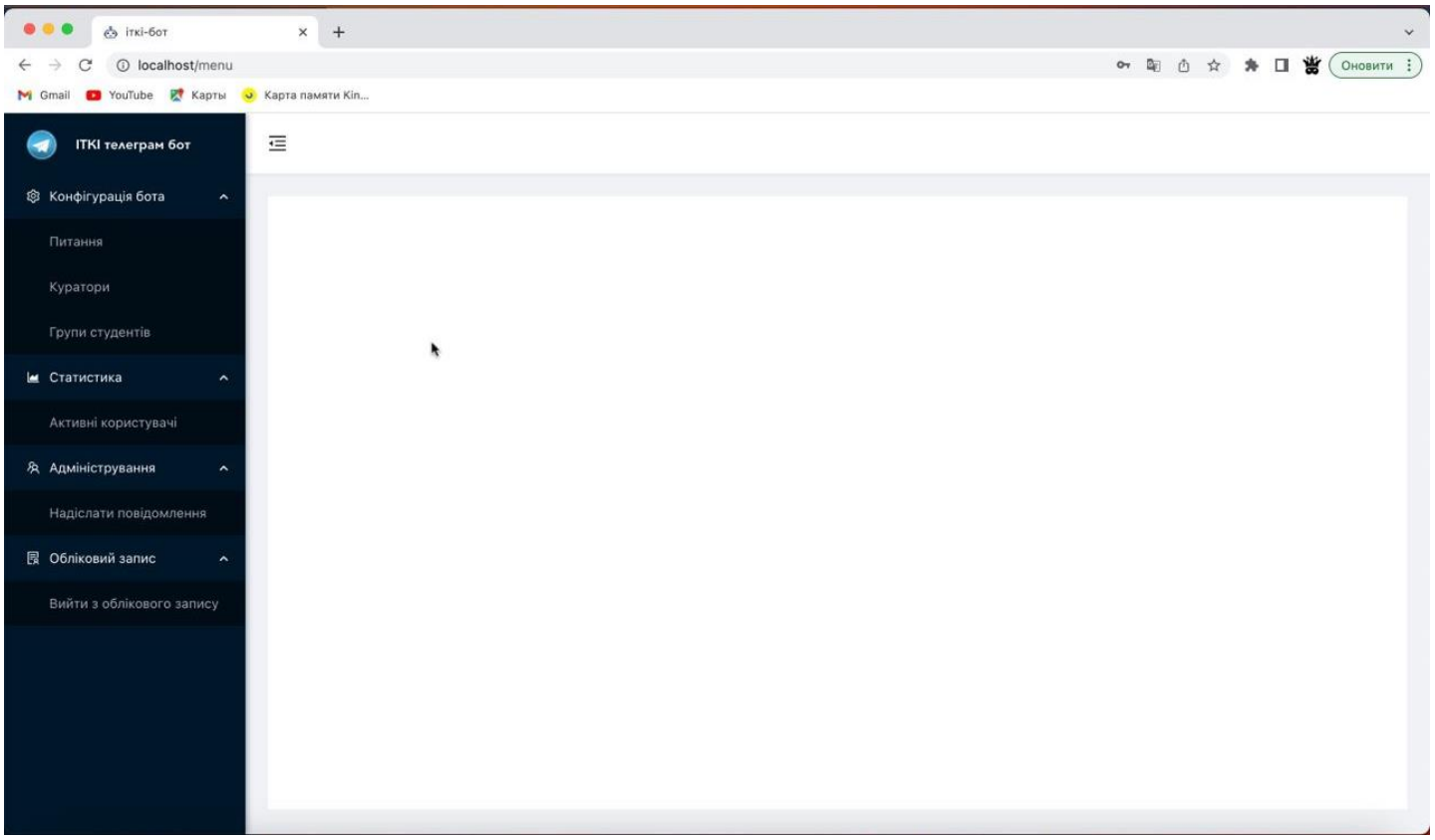

Рисунок 4.3 – Головна сторінка платформи для налаштування боту

Для того щоб перейти до цієї платформи (Рисунок 4.3) потрібно пройти аутентифікацію. Пароль – admin, логін – admin.

Меню платформи адміністрування складається з 4 груп опцій:

– Конфігурація бота;
– Статистика;
– Адміністрування;
– Обліковий запис.

Далі можна побачити схематичне пояснення структурних елементів UI.

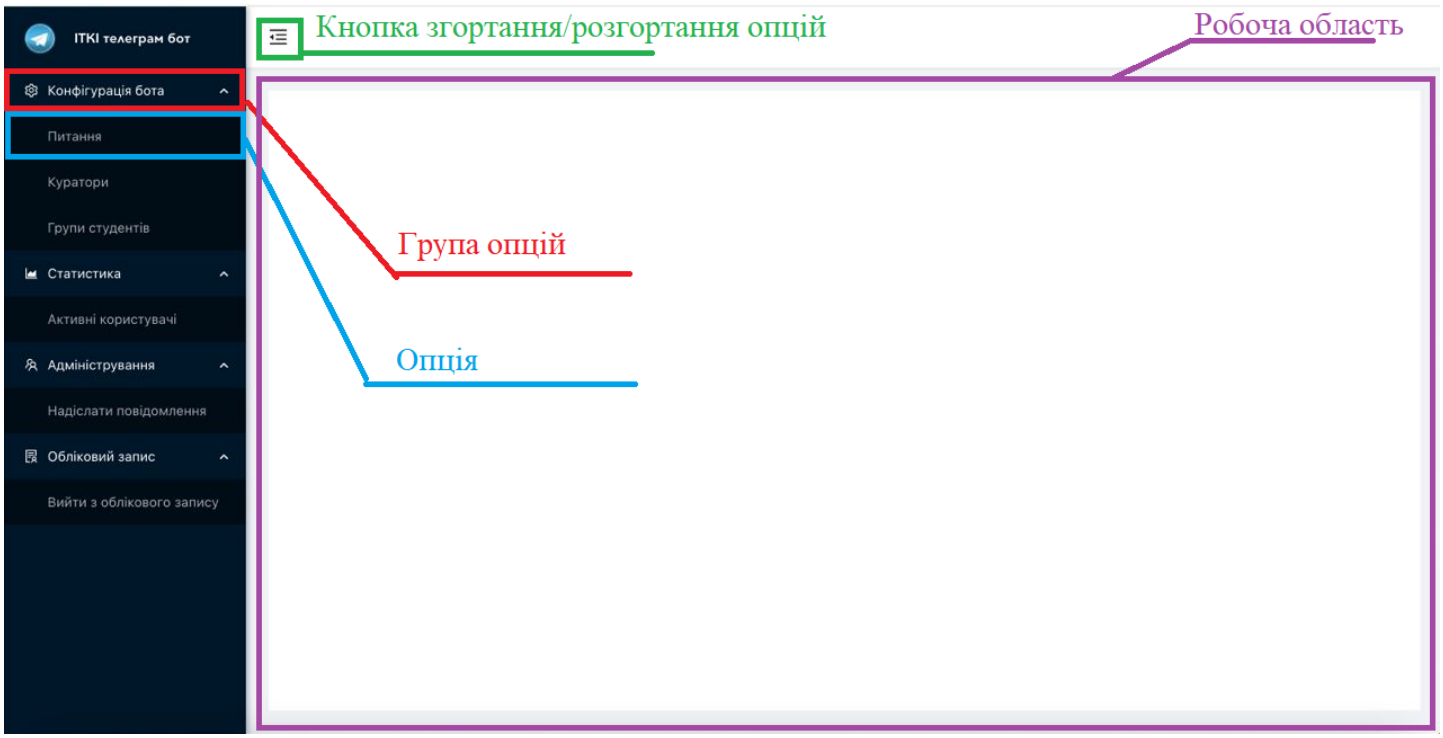

Рисунок 4.4 – Схема структури UI

Групи опцій слугують контейнерами для опцій додатку. Ці групи можна згортати і розгортати. При використанні кнопки згортання опцій, групи опцій будуть відображатися у вигляді випадаючого меню при наведенні (Рис 4.5).

При виборі опції, бот виконає redirect на відповідний endpoint (/questions, /curators та інші). При співпадінні посилання в опції та посилання на якому знаходиться користувач, відповідна опція буде підсвічена блакитним кольором.

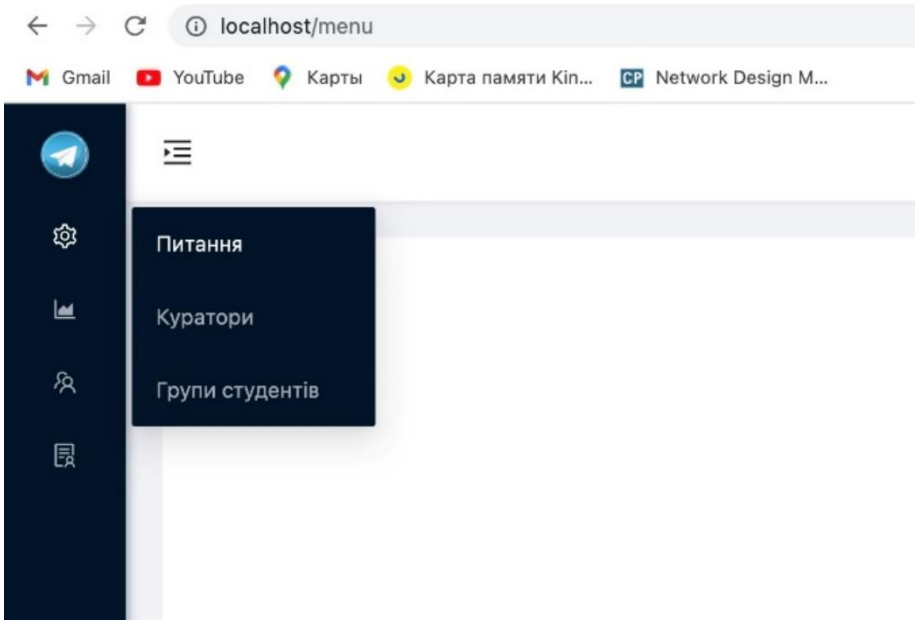

Рисунок 4.5 – Меню зі згорнутими опціями

Кожна опція меню відповідає за конкретну функціональність системи адміністрування бота.

Основні можливості адміністрування бота:

– зміна питань на які може відповідати бот;
– зміна кураторів і груп студентів;
– моніторинг користувачів, що увійшли до боту;
– надсилання повідомлення усім користувачам бота (підтримуються кілька типів повідомлень).

Переглянути і додавати питання можна через першу опцію меню (Рис. 4.3). Робоча область для цієї опції представляє собою список з пагінацією (максимум 5 елементів на сторінці), кнопку для додавання питань та кнопки на елементах для видалення (Рис. 4.6).

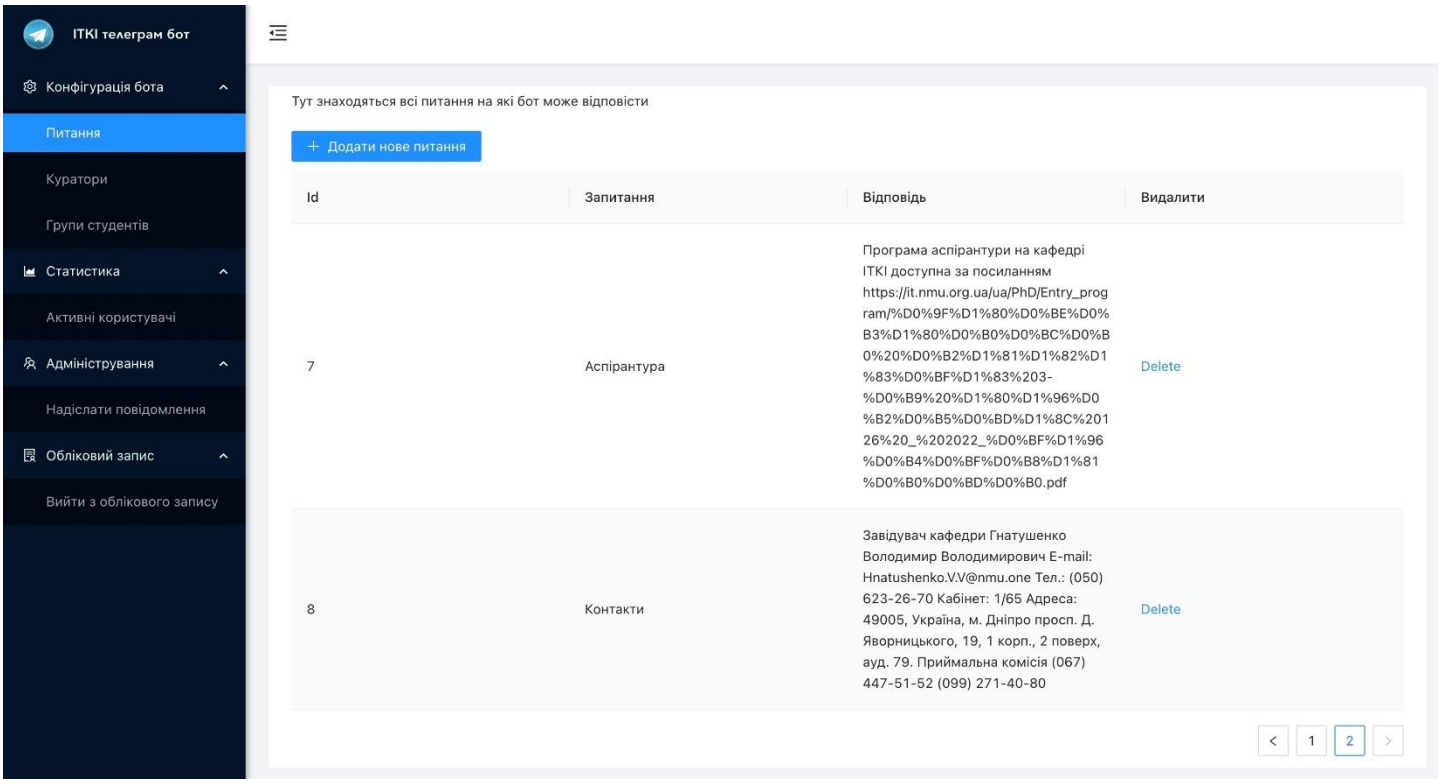

Рисунок 4.6 – Меню зі згорнутими опціями

Додавання питань реалізоване через модальне вікно, що з’являється при натисненні на відповідну кнопку (Рис. 4.7).

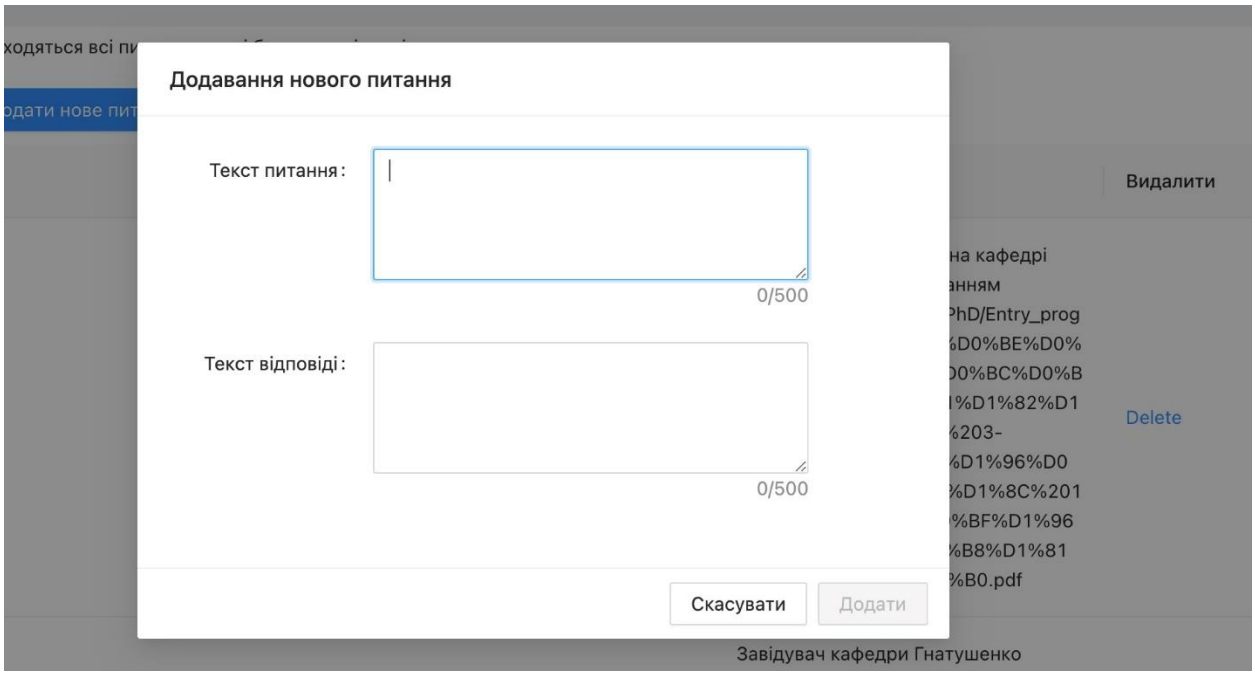

Рисунок 4.7 – Модальне вікно додавання питань

Всі додані питання будуть відображатися в клавіатурі питань бота (Рис 4.2).

Наступні дві опції це додавання кураторів та груп студентів. Ці дані використовуються для команди /curator (Рис. 4.1).

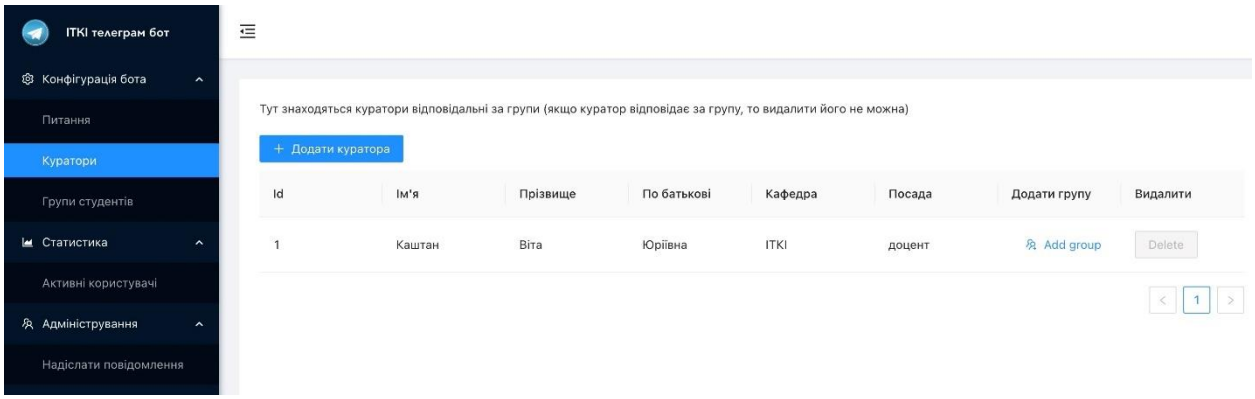

Рисунок 4.8 – Налаштування кураторів

Реалізація додавання кураторів виконана через модальне вікно як в попередньому прикладі. Кожна форма яку можна заповнити підлягає валідації і відображає помилки у відповідному полі.

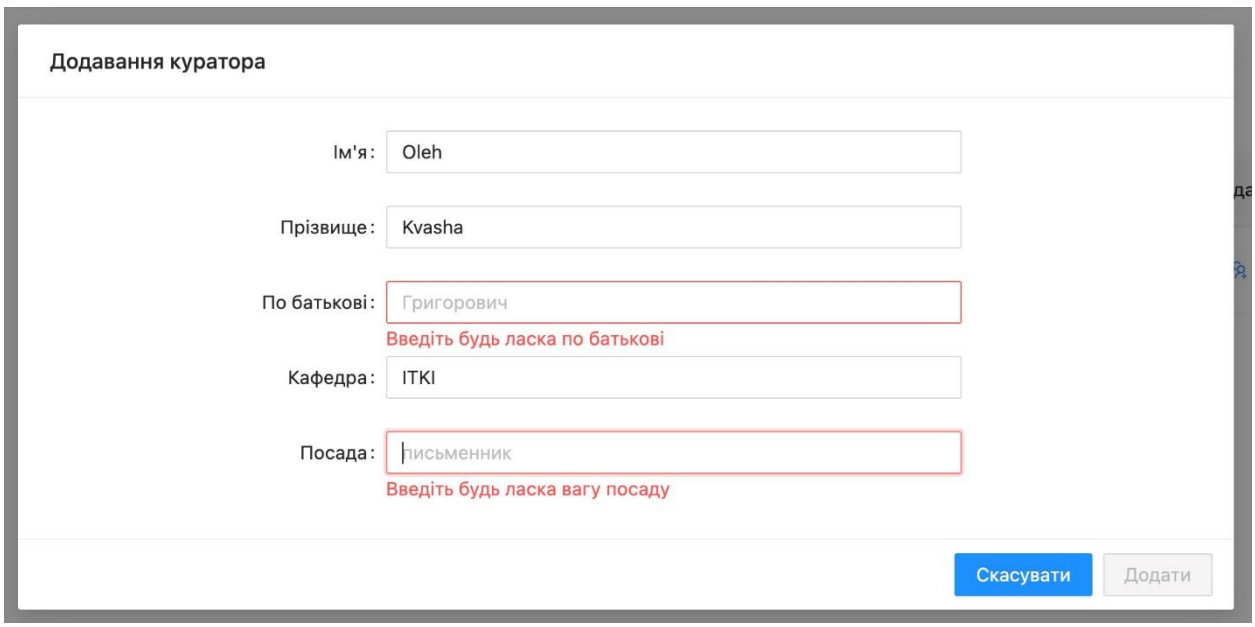

Рисунок 4.9 – Приклад валідації полів у формах

У списку, кураторів також можна побачити кнопку Add group. Це кнопка для додавання групи до відповідного куратора. При додаванні групи, в заголовку модального вікна буде відображатись ПІБ куратора до якого додають групу. Якщо куратор має під керівництвом хоча б одну групу, тоді видаляти його не можна.

Видалити групу можна в наступному вікні (Рис. 4.10).

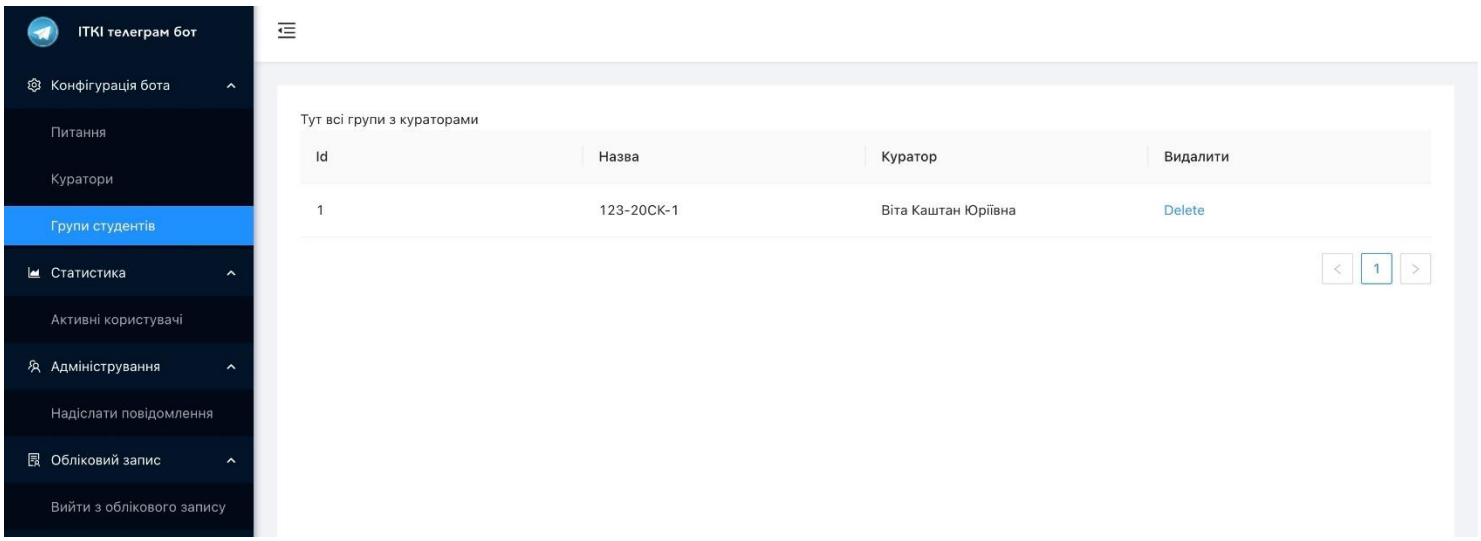

Рисунок 4.10 – Вікно перегляду груп студентів

Наступна опція відображає всіх користувачів, що почали розмову з розробленим ботом в телеграмі (Рис. 4.11).

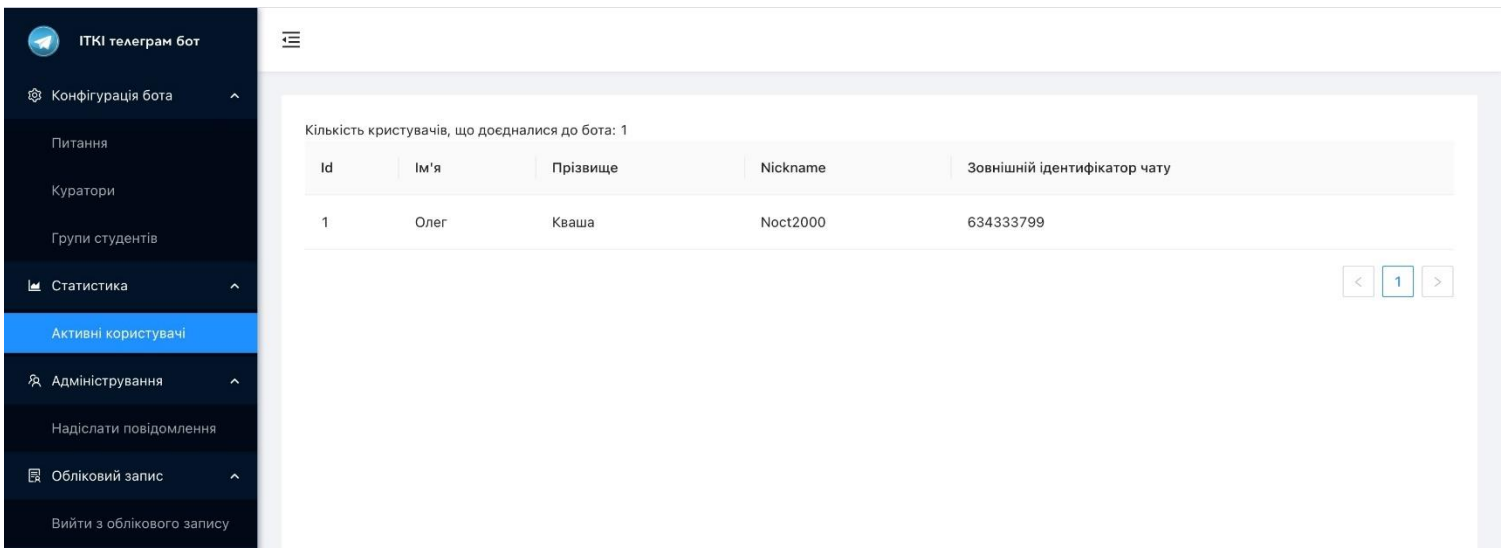

Рисунок 4.11 – Вікно перегляду телеграм-користувачів

В опції перегляду телеграм-користувачів можна побачити всю інформацію яку може надати телеграм про користувача. Ця інформація може бути не консистентна: може не бути інформація крім зовнішнього ідентифікатору і імені користувача.

Наступна опція – це опція надсилання повідомлень.

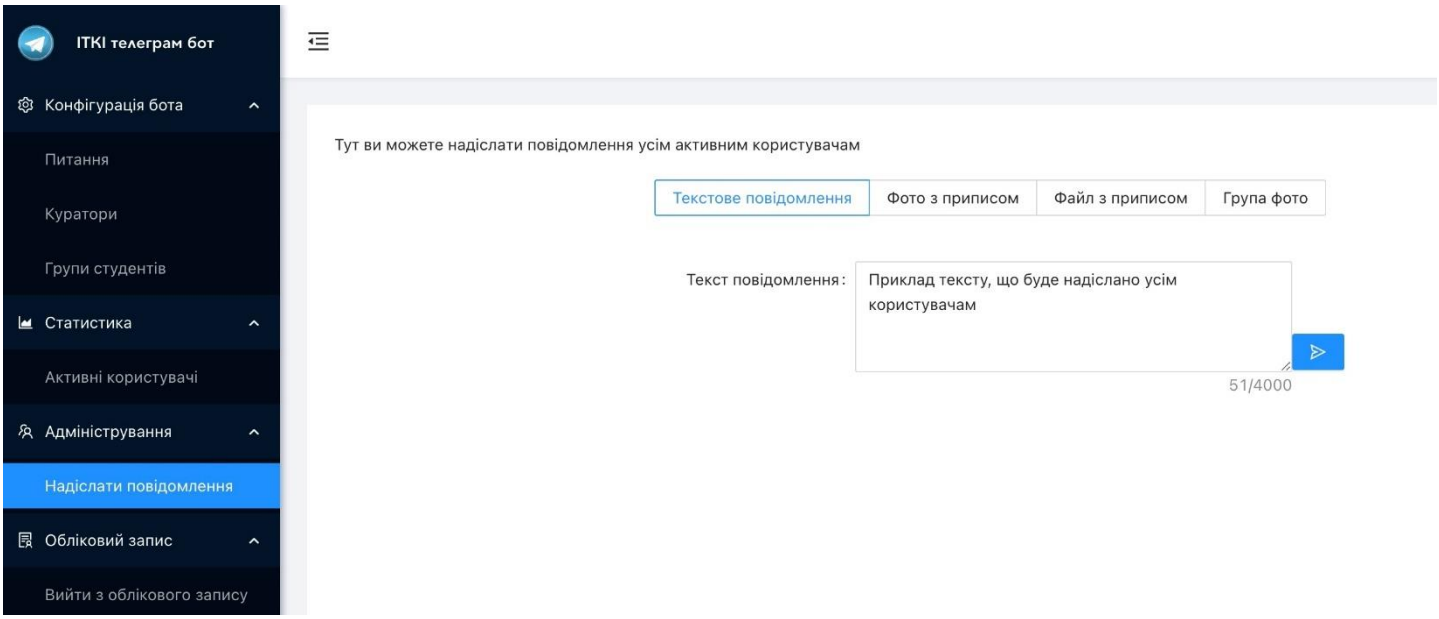

Рисунок 4.12 – Вікно надсилання повідомлень

Опція надсилання повідомлень підтримує такі типи повідомлень як текстове, фото з приписом, файл з приписом, група фото. Форма для надсилання повідомлення перемикається через radio button.

При надсиланні фото з приписом, припис заповнювати не обов’язково, фото можна завантажити лише одне. Воно буде стиснуто базовими механізмами телеграму. Для додавання фото, необхідно натиснути на область додавання і у діалоговому вікні браузера вибрати необхідний файл або просто перетягнути файл у межі області додавання файлів.

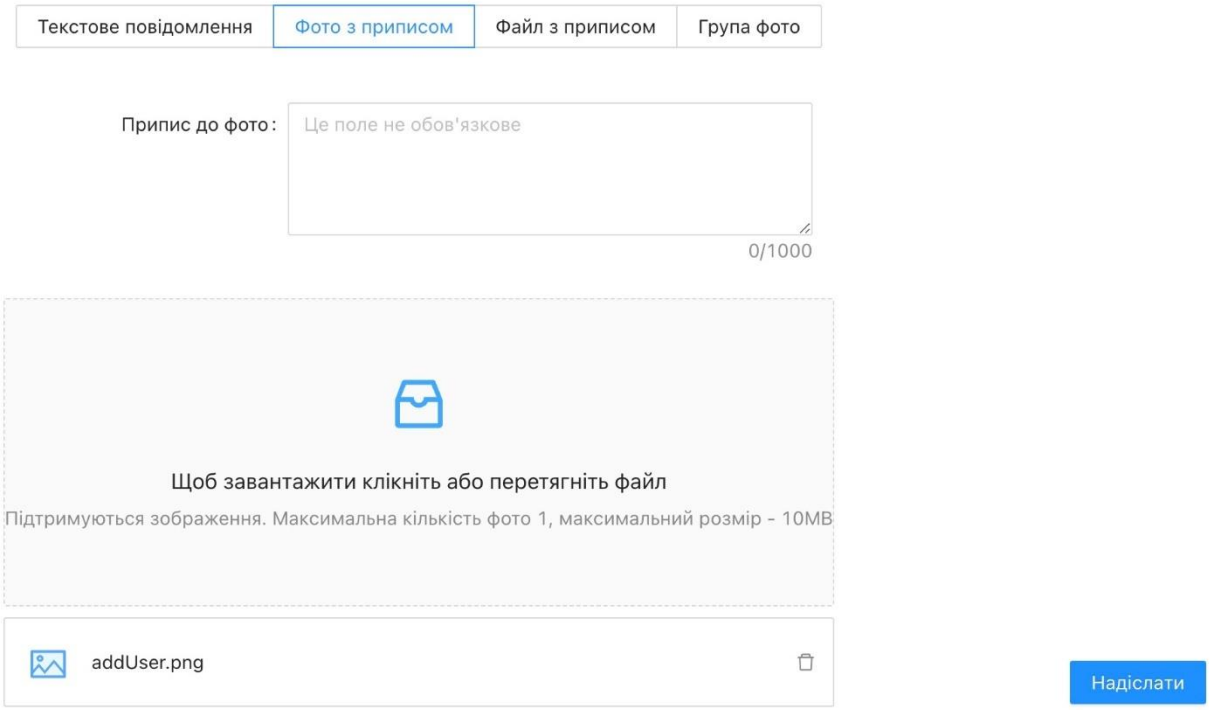

Рисунок 4.13 – Форма надсилання фото з приписом

Всі рисунки відображаються у вигляді списку з відповідною назвою файлу. При необхідності, цей файл можна прибрати натиснувши на кнопку корзини бля необхідного фото.

Такий самий механізм надсилання файлів з приписом. Основні відмінності в обробці файлів алгоритмами телеграму (стиснення і адаптація клієнтами телеграму) та обмеженнями на надсилання (об’єм файлу).

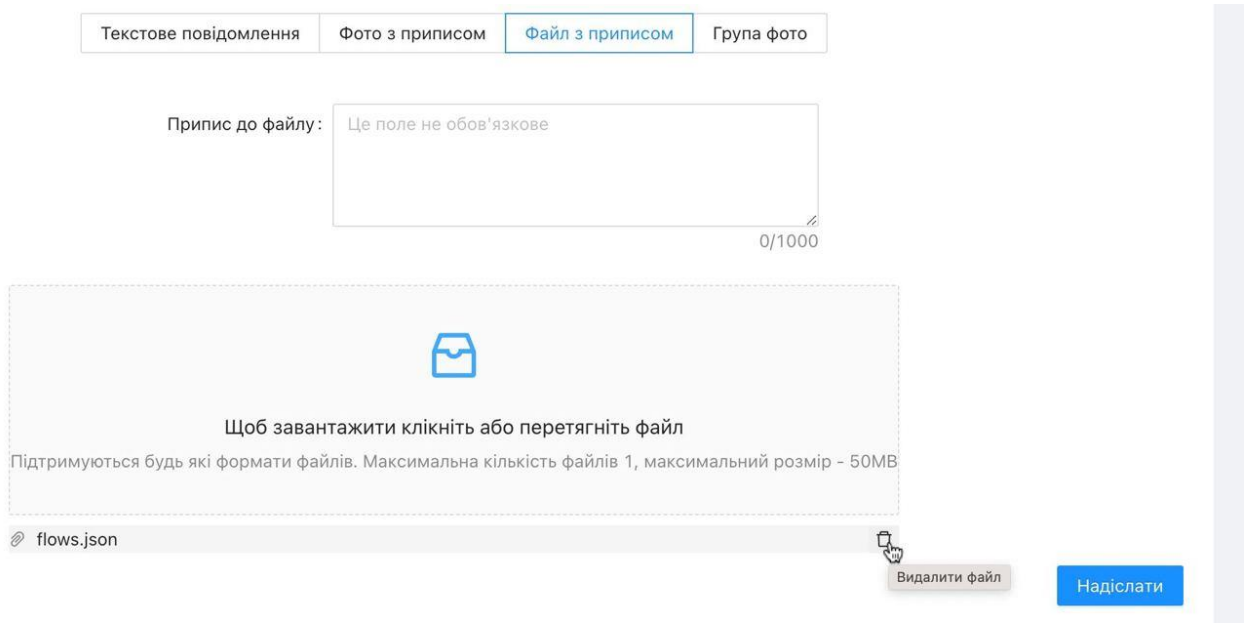

Рисунок 4.14 – Форма надсилання файлу з приписом

Надсилання групи фото, підтримує завантаження кількох файлів-зображень (файли, що браузер асоціює як зображення).

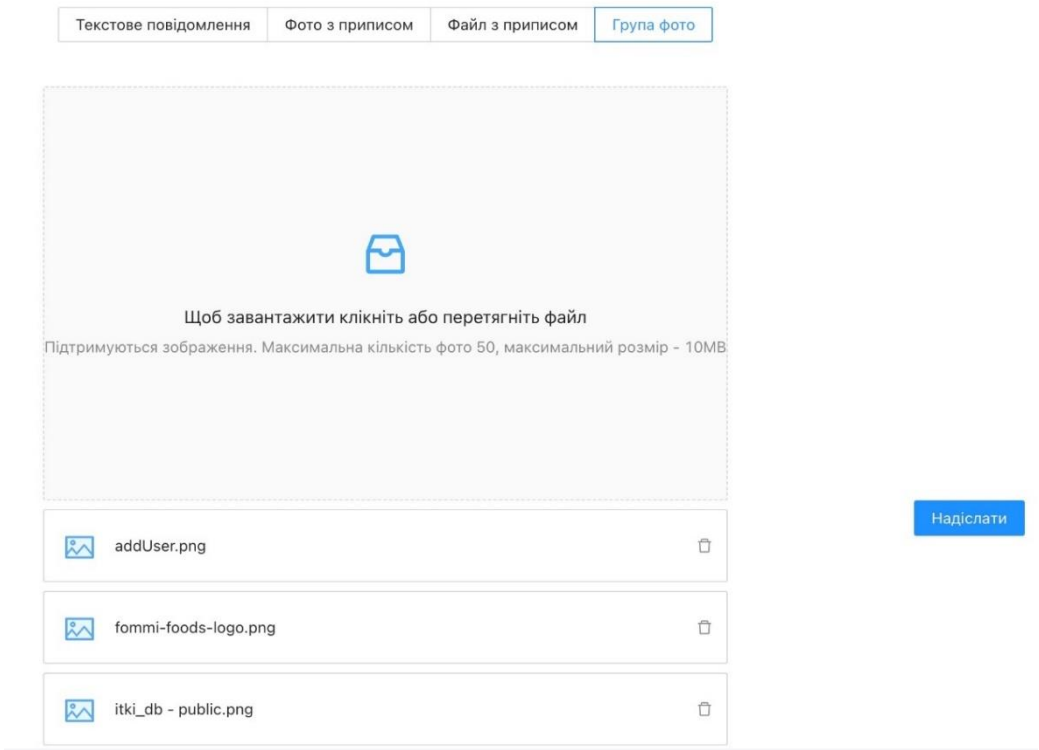

Рисунок 4.15 – Форма надсилання групи фото

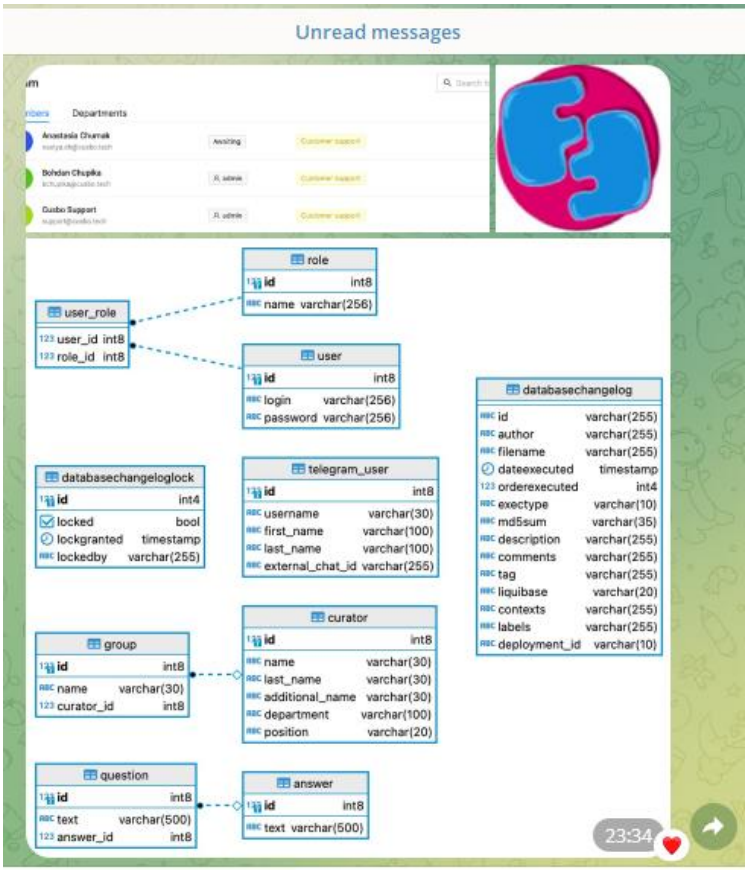

Рисунок 4.16 – Приклад надсилання групи фото

Група фото в телеграмі відображається як альбом з фото (Рис. 4.16).

У надсилання повідомлень є корисна особливість, файли і фото не надсилаються одразу на сервер. Сперш браузер їх додає на надсилання, а завантаження відбувається тільки при натисненні кнопки Надіслати.

Окрім, UI на Angular, поведінку бота можна змінювати через Node-RED.

Node-RED доступний за портом 1880. Також вимагає аутентифікацію (логін – admin, пароль – admin).

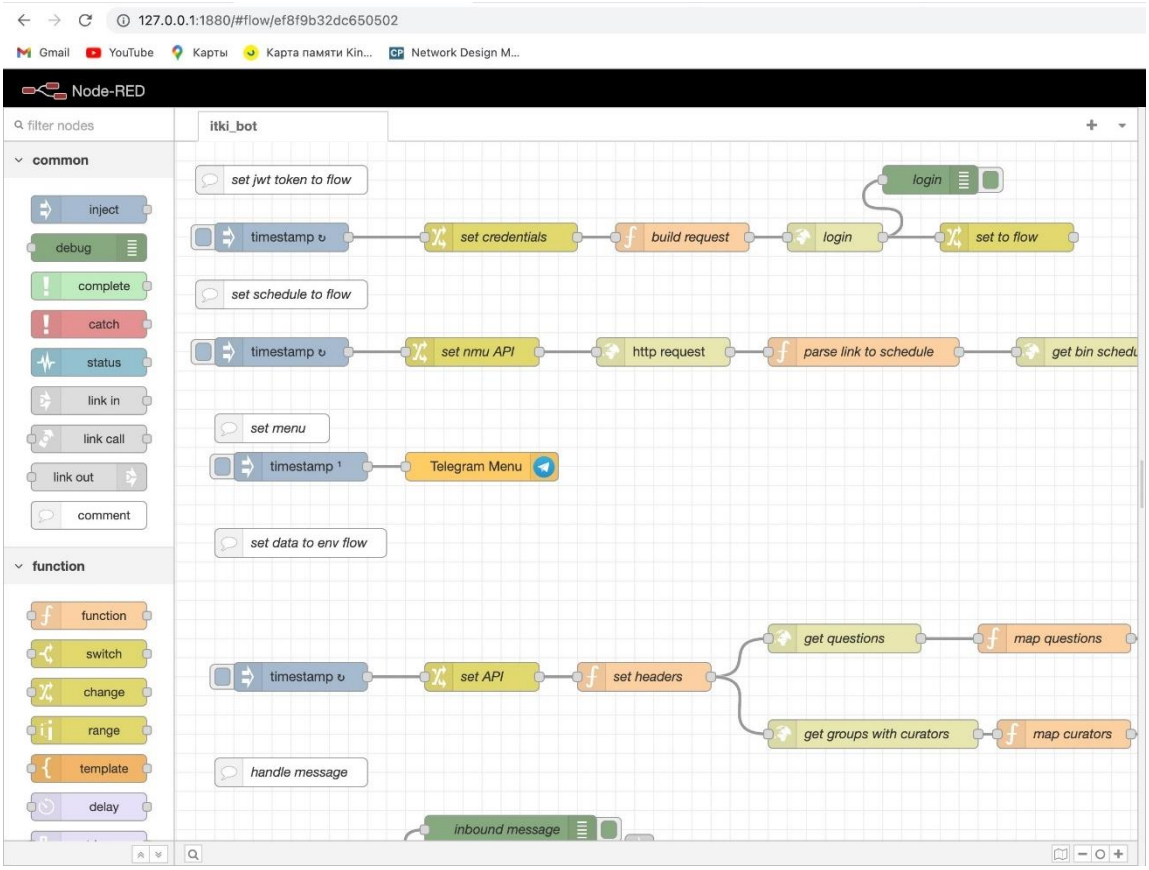

Рисунок 4.17 – Node-RED flow

В Node-RED можна керувати підтягненням даних з кастомного API та сайту НТУ «ДП». Також можна змінювати основну поведінку і команди які надсилає бот.

Розклад підтягання даних ботом: підтягання розкладу з сайту НТУ «ДП» – кожні 12 годин, підтягання даних з кастомного API – кожні 10 хвилин.

Далі можна побачити Use case UML діаграму для користувачів телеграму та адміністраторів додатку.

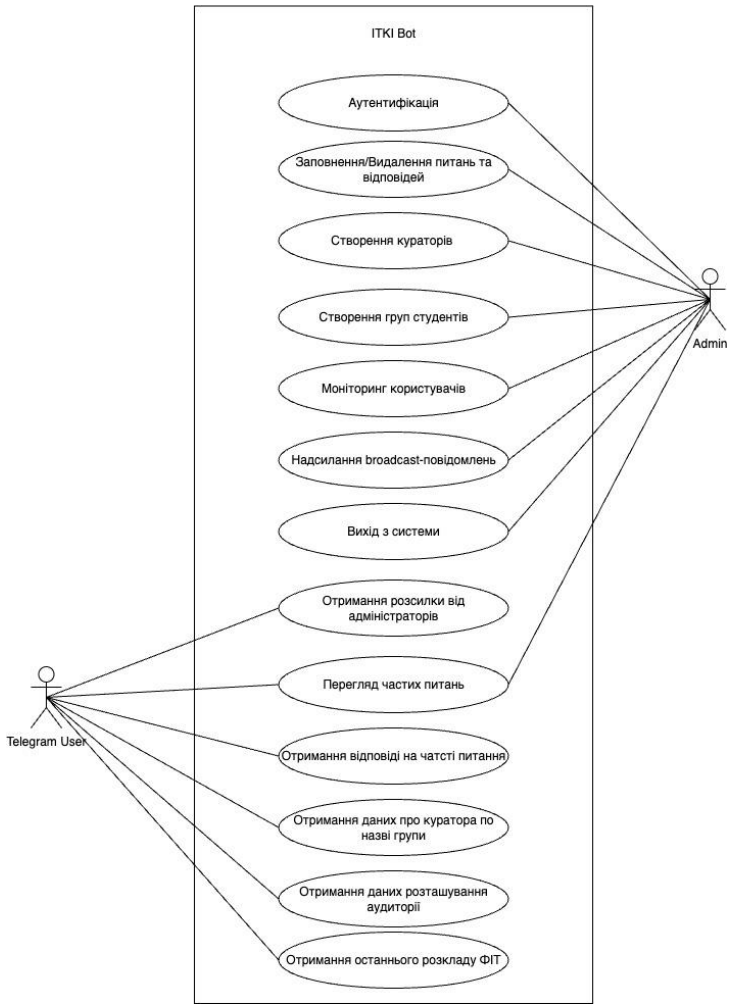

Рисунок 4.18 – Use case UML діаграма

Діаграма на рисунку 4.18 показує взаємодію з додатком (усіма сервісами розробленого додатку) для користувачів телеграму та адміністраторів бота.

4.5 Опис змінних програми

Увесь додаток (всі сервіси) використовують змінні зі штучного віртуального середовища, що створює Docker. Всі змінні можна подивитись у файлі .env_sample.

Далі наведено його вміст. Коментарі відділені позначкою #.

```
# Змінні для функціонування Node-RED
TZ=Europe/Kiev
API_HOST=host.docker.internal:8080 # Посилання на розроблене АПІ
API_LOGIN=admin # Логін для аутентифікації в АПІ
```

API_PASSWORD=admin # Пароль для аутентифікації в АПІ

add your telegram token and username here

TELEGRAM_TOKEN=_ # Токен через якого додаток буде заходити в телеграм бота

TELEGRAM_USERNAME=_ # Ім’я бота створеного в телеграмі

Посилання на розклад в університеті

NMU_API=https://www.nmu.org.ua/ua/content/student_life/students/schedule

Дані для входження до БД серверу

POSTGRES_USER=root

POSTGRES_PASSWORD=12345

POSTGRES_DB=itki_db

API

SPRING_LOCAL_PORT=8080 # Порт який АПІ буде використовувати на хості

SPRING_DOCKER_PORT=8080 # Порт в середині докер контейнеру (не рекомендується змінювати)

DEBUG_PORT=5005 # Порт для віддаленого відлагоджування

JWT_TOKEN_TTL=3600000 # Час який буде діяти виданий токен після аутентифікації (без оновлення), час задано у мілісекундах.

ВИСНОВОК

Розроблена корпоративна мережа, виконує всі покладені на неї завдання:

– швидкий та централізований доступ до глобальної мережі Інтернет;

– надання доступу до серверу бази даних тестових версій проектів;

– віддалений доступ до мережевих ресурсів з використанням VPN-тунелю.

Також до переваг даної комп’ютерної мережі можна віднести, її легке налаштування та масштабування. Кожна підмережа обладнана DHCP сервером. Таким чином можна буде забезпечити легке підключення нових робочих станцій та підвищити захищеність системи за рахунок обмеження кількості пристроїв в мережі пулом адрес.

Розроблена мережа має кілька способів захисту: port security, SSH, service AAA.

Розроблена мережа має високу надійність за рахунок використання повно зв’язних елементів, резервних шляхів, резервних маршрутизаторів, LACP.

Окрім цього було створено сучасний full-stack додаток з кількома способами взаємодії. Додаток було розроблено за клієнт-серверною архітектурою з використанням архітектурного стилю REST, принципів розробки SOLID та з використанням технологій віртуалізації (Docker).

При розробці були вибрані найсучасніші і найнадійніші технології у сфері web-розробки: Java, Liquibase, Angular, Docker.

Головної метою цього додатку є взаємодія зі абітурієнтами та діючими студентами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Методичні рекомендації до виконання кваліфікаційної роботи бакалавра студентами галузі знань 12 Інформаційні технології спеціальності 123 Комп'ютерна інженерія / Л.І. Цвіркун, С.М. Ткаченко, Я.В. Панферова, Д.О. Бешта, Л.В. Бешта. – Д.: НТУ «Дніпровська політехніка», 2022.
2. Методичні вказівки до виконання розділу „Охорона праці“ в дипломних проектах (роботах) бакалаврів інституту електроенергетики / В.І. Голінько, В.Ю. Фрундін, Ю.І. Чеберячко, М.Ю. Іконніков. – Д.: Державний ВНЗ «Національний гірничий університет», 2012. – 8 с
3. Комп'ютерні мережі. Методичні рекомендації до виконання лабораторних робіт студентами галузі знань 12 Інформаційні технології спеціальності 123 Комп'ютерна інженерія: у 2 ч. / Л.І. Цвіркун, Я.В. Панферова ; М-во освіти і науки України, НТУ «Дніпровська політехніка», 2018. – Ч. 1. – 41 с.
4. Кафедра Інформаційних технологій та комп'ютерної інженерії. Історія створення кафедри [Електронний ресурс] – Режим доступу до ресурсу: https://it.nmu.org.ua/ua/about_department_ist/history_department.php
5. Spring. Web applications [Електронний ресурс] – Режим доступу до ресурсу: https://spring.io/web-applications
6. Angular. What is Angular? [Електронний ресурс] – Режим доступу до ресурсу: https://angular.io/guide/what-is-angular
7. Node-RED. About [Електронний ресурс] – Режим доступу до ресурсу: https://nodered.org/about/
8. Introduction to JSON Web Tokens [Електронний ресурс] – Режим доступу до ресурсу: https://jwt.io/introduction
9. Docker. Get started. Overview [Електронний ресурс] – Режим доступу

до ресурсу: https://docs.docker.com/get-started/

10. NGINX. Beginner’s Guide [Електронний ресурс] – Режим доступу до ресурсу: https://nginx.org/en/docs/beginners_guide.html

Додаток А

Вихідний код html-сторінки з Cisco PT

```
<!DOCTYPE html>
<html>
<head>
    <title>Комп’ютерна система кафедри ІТКІ</title>
</head>
<body style="background-color: #f2f2f2; font-family: Arial, sans-serif;">
    <header style="background-color: #333; color: #fff; padding: 20px;">
        <h1 style="margin: 0;">Комп’ютерна система кафедри інформаційних технологій та комп'ютерної інженерії</h1>
        <p style="margin: 0;">з реалізацією побудови та налаштування корпоративної мережі</p>
        <p style="margin: 0;">та з підтримкою веб-застосунку "Телеграм-бот ІТКІ" для профорієнтаційної роботи</p>
    </header>
    <main style="padding: 20px;">
        <h2>Про нас</h2>
        <p>Кафедра інформаційних технологій та комп'ютерної інженерії займається навчанням та дослідженням у галузі інформаційних технологій, комп'ютерних систем та мереж. Ми надаємо студентам якісну освіту та практичні навички, щоб вони могли знайти роботу в сфері IT.</p>

        <h2>Корпоративна мережа</h2>
        <p>Ми пропонуємо побудувати та налаштувати корпоративну мережу для вашої компанії. Наші експерти забезпечать безпеку та стабільність мережі, щоб ви могли працювати ефективніше.</p>

        <h2>Телеграм-бот ІТКІ</h2>
        <p>Наш веб-застосунок "Телеграм-бот ІТКІ" допоможе вам з профорієнтацією та вибором курсів. Ви зможете отримати інформацію про наші програми навчання.</p>
    </main>

    <footer style="background-color: #333; color: #fff; padding: 20px; text-align:center;">
    <p>© Кафедра інформаційних технологій та комп'ютерної інженерії, 2023</p>
    </footer>
    </body>
    </html>
```