Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра обчислювальної техніки

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Комп'ютерна мережа компанії з віддаленими офісами»

08-54.БКР.030.00.000 ПЗ

Виконав: студент 4 курсу, групи 2СП-21б

спеціальності 123 — Системне програмування

_____ Кубенко Д.М.

(підпис)

Керівник: к.т.н., проф. каф. ОТ

_____Захарченко С. М.

(підпис)

«_____» _____2025 р.

Рецензент: доктор філософії, доцент каф. МБІС

_____Салієва О. В.

(підпис)

«_____» _____2025 р.

Допущено до захисту

Завідувач кафедри ОТ

д.т.н., проф. Азаров О. Д.

«17» 06 2025 р.

Вінниця ВНТУ — 2025 рік

ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра обчислювальної техніки

Освітньо-кваліфікаційний рівень бакалавр

Галузь знань — 12 «Інформаційні технології»

Спеціальність — 123 «Комп'ютерна інженерія»

Освітньо-професійна програма «Системне програмування

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ

д.т.н., проф. Азаров О. Д.

«21» березня 2025 р.

З А В Д А Н Н Я

НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Кубенко Денису Михайловичу

1 Тема роботи: «Комп'ютерна мережа компанії з віддаленими офісами», керівник роботи к.т.н., проф. каф. ОТ Захарченко С.М., затверджена наказом вищого навчального закладу від 20 березня 2025 р. №97.

2 Термін подання студентом роботи 13 травня 2025 р.

3 Вихідні дані до роботи: призначення — проєктування комп'ютерних мереж; основні підтримувані функції — передача пакетів між робочими ПК.

4 Зміст розрахунково-пояснювальної записки: вступ, аналіз предметної області, проєктування мережі, конфігурація та тестування мережі і створення веб-сервісу.

5 Перелік ілюстративної частини (з точним зазначенням обов'язкових креслень): логічна топологія мережі компанії з віддаленими офісами, структурна схема мережі компанії з віддаленими офісами.

6 Консультанти розділів роботи приведені в таблиці 1.

Таблиця 1 — Консультанти роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата завдання видав	Підпис, дата завдання прийн
1-3	к.т.н., проф. каф. ОТ Захарченко С. М.	21.03.25	13.05
Нормоконтроль	ас. каф. ОТ Швець С. І.	13.05.25	30.05

7 Дата видачі завдання 21 березня 2025 р.

8 Календарний план виконання БКР приведений в таблиці 2.

Таблиця 2 — Календарний план виконання БКР

№	Назва етапу	Термін виконання	Підпис
1	Постановка задачі	21.03.25	
2	Аналіз технологій для створення комп'ютерних мереж і веб-сервісу	21.03.25— 30.03.25	
3	Аналіз та обгрунтування вибору технологій	21.03.25 — 30.03.25	
4	Реалізація комп'ютерної мережі і веб сервісу	31.03.25— 13.04.25	
5	Тестування комп'ютерної мережі	14.04.25— 27.04.25	
6	Підготовка тезових матеріалів для публікації	21.04.25— 28.04.25	
7	Оформлення пояснювальної записки та презентації для доповіді	29.04.25— 12.05.25	
8	Підготовка супровідної документації, перевірка відповідності нормам та проходження тесту на плагіат	13.05.25	

Студент Денис КУБЕНКО

Керівник роботи к.т.н., проф. Сергій ЗАХАРЧЕНКО

АНОТАЦІЯ

УДК 004.7

Кубенко Д.М. Комп’ютерна мережа компанії з віддаленими офісами. Бакалаврська кваліфікаційна робота зі спеціальності 123 «Комп’ютерна інженерія», освітня програма «Системне програмування». Вінниця: ВНТУ, 2025. 110 с.

На укр. мові. Бібліогр.: 20 назв, рис.: 41 табл. 11.

У бакалаврській дипломній роботі розроблено проєкт комп'ютерної мережі компанії з віддаленими офісами на основі сучасних мережевих технологій. Для забезпечення динамічної маршрутизації між двома відділеннями та серверною використано протокол OSPF. Реалізовано IPSec VPN-з’єднання для обміну даними між офісами.

Особливу увагу приділено питанням безпеки мережі: налаштовано ACL для обмеження SSH-доступу лише з визначених IP-адрес на трьох маршрутизаторах, встановлено паролі для консольного доступу та загального доступу до обладнання. Для захищеного з'єднання між мережами впроваджено протокол IPSec. Централізовано налаштовано DHCP-сервер в серверній з використанням DHCP Relay для автоматичного призначення IP-адрес у всіх мережевих сегментах. Додатково розгорнуто DNS-сервер та HTTP-сервер для забезпечення необхідних мережевих сервісів.

Окремо реалізовано сервер бази даних MariaDB на платформі Windows та розроблено веб-сервіс з використанням мови програмування PHP, який взаємодіє з базою даних для забезпечення необхідного функціоналу.

Розроблений проєкт забезпечує надійне та безпечне функціонування корпоративної мережі з ефективним управлінням віддаленими офісами.

Ключові слова: комп'ютерна мережа, OSPF, VPN, IPSec, DHCP, ACL, SSH, DNS, HTTP, MariaDB, PHP.

ABSTRACT

Kubenko D.M. Computer Network of a Company with Remote Offices. Bachelor Thesis in the specialty 123 “Computer Engineering”, educational program “System Programming". Vinnytsia: VNTU, 2025. 110 p.

In Ukrainian. Bibliography: 20 titles, figs.: 41, tables: 11

The present work presents the development of a computer network project for a company with remote offices based on modern networking technologies. The OSPF protocol was implemented to ensure dynamic routing between two departments and the server room. IPSec was deployed VPN connections were established for secure data exchange between offices.

Special attention was paid to network security issues: ACL was configured to restrict SSH access only from specified IP addresses on three routers, passwords were established for console access and general equipment access. The IPSec protocol was implemented for secure inter-network connections. A DHCP server was centrally configured in the server room using DHCP Relay for automatic IP address assignment across all network segments. Additionally, DNS server and HTTP server were deployed to provide essential network services.

A separate MariaDB database server was implemented on the Windows platform and a web application was developed using PHP programming language, which interacts with the database to provide the required functionality.

The developed project ensures reliable and secure operation of the corporate network with efficient management of remote offices.

Keywords: computer network, OSPF, VPN, IPSec, DHCP, ACL, SSH, DNS, HTTP, MariaDB, PHP.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

OSPF — Open Shortest Path First

VPN — Virtual Private Network

IPSec — Internet Protocol Security

DHCP — Dynamic Host Configuration Protocol

ACL — Access Control List

SSH — Secure Shell

DNS — Domain Name System

HTTP — HyperText Transfer Protocol

TCP/IP — Transmission Control Protocol/Internet Protocol

IP — Internet Protocol

PHP — PHP: Hypertext Preprocessor

LAN — Local Area Network

Wi-Fi — Wireless Fidelity

ЗМІСТ

ВСТУП

В сучасному світі комп’ютерні мережі є одними з основних частин діяльності багатьох організацій, підприємств та установ. Комп’ютерна мережа —це система, яка дозволяє двом або більше пристроям взаємодіяти один з одним з метою обміну даними, спільного використання ресурсів та централізованого управління інформацією. З розвитком технологій розміри та складність комп’ютерних мереж значно зросли, починаючи від локальних мереж (LAN) і переходячи до глобальних мереж (WAN), які охоплюють цілі континенти та регіони.

Актуальність розробки полягає у необхідності впровадження надійних мережевих рішень для компаній з віддаленими офісами через зростання обсягів переданої інформації та складності корпоративних мережевих операцій. Розробка інтегрованої мережевої системи дасть можливість оптимізувати управління бізнес-процесами, забезпечити високий рівень безпеки даних та ефективну комунікацію між віддаленими підрозділами.

У наукових публікаціях та дослідженнях наголошується на значущості застосування сучасних технологій, як протоколи з’єднання віддалених мереж, включаючи IPSec.

Мета бакалаврської кваліфікаційної роботи полягає у проєктуванні, побудові та налаштуванні розподіленої комп'ютерної мережі компанії з віддаленими офісами з використанням сучасних технологій динамічної маршрутизації, забезпеченні високого рівня інформаційної безпеки через впровадження IPSec, розробці політик доступу (ACL), організації захищеного віддаленого адміністрування SSH та налаштуванні централізованих мережевих сервісів.

Завдання для досягнення поставленої мети БКР:

— здійснити аналіз сучасних технологій побудови розподілених комп'ютерних мереж;

— розробити архітектуру мережевої інфраструктури для компанії з віддаленими офісами;

— дослідити та обрати оптимальні протоколи динамічної маршрутизації та засоби мережевої безпеки;

— виконати планування IP-адресного простору та підбір необхідного мережевого обладнання;

— створити веб-додаток з інтеграцією бази даних;

— реалізувати перевірку функціональності створеної мережевої системи.

Об'єкт дослідження — це процес проєктування та налаштування комп'ютерної мережі компанії з віддаленими офісами.

Предметом дослідження є технології проєктування, налаштування та забезпечення безпеки розподілених комп'ютерних мереж з використанням набору протоколів IPSec для забезпечення захисту даних, які будуть передаватись по міжмережевому протоколу IP, протокол для динамічної маршрутизації OSPF, DHCP та SSH.

Новизна результатів полягає у комплексному підході до створення мережевої архітектури для компаній з географічно розподіленими підрозділами, що поєднує використання протоколу OSPF для динамічної маршрутизації, технології IPSec для захищеного міжофісного зв'язку та централізованого управління мережевими сервісами через DHCP Relay, що забезпечує підвищену надійність та масштабованість корпоративної мережевої інфраструктури.

Матеріали роботи були представлені та **апробовані**[1]:

Технології і методи з’єднання віддалених мереж. / Д.М. Кубенко, С. М. Захарченко // Матеріали конференції LIV Всеукраїнської науково-технічної конференції факультету інформаційних технологій та комп'ютерної інженерії. Вінниця 2025 р. [Електронний ресурс]. Режим доступу: https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2025/paper/view/24510

1 АНАЛІЗ СУЧАСНИХ ТЕХНОЛОГІЙ ПРОЄКТУВАННЯ КОМП’ЮТЕРНИХ МЕРЕЖ

1.1 Основи комп’ютерних мереж

Комп’ютерна мережа являє собою взаємопов’язану систему двох або більше комп'ютерів та периферійних пристроїв, які можуть обмінюватися даними та спільно використовувати ресурси. Сучасні комп'ютерні мережі є основою цифрової інфраструктури підприємств, організацій та глобального інформаційного простору, забезпечуючи можливість передачі інформації на різні відстані з високою швидкістю та надійністю.

Базова архітектура комп’ютерних мереж ґрунтується на принципі пакетної комутації, де дані розбиваються на невеликі фрагменти, які називаються пакетами, та передаються через мережу незалежно один від одного. Кожен пакет містить не лише частину передаваних даних, але й службову інформацію, яка включає адреси відправника та одержувача, номер пакета в послідовності, контрольні суми для перевірки цілісності даних та інші метадані, необхідні для правильної доставки та відновлення повідомлення на стороні отримувача.

Фундаментальними компонентами будь-якої комп’ютерної мережі є кінцеві пристрої, мережеве обладнання та канали зв'язку. Кінцеві пристрої включають:

— персональні комп'ютери;
— сервери;
— смартфони;
— планшети;
— принтери;
— сканери;
— IP-камери;

Персональні комп'ютери функціонують як основні робочі станції користувачів для виконання повсякденних завдань та доступу до мережевих ресурсів.

Сервери забезпечують централізоване зберігання даних, виконання додатків

та надання різноманітних мережевих сервісів іншим пристроям.

Смартфони та планшети представляють мобільні пристрої, які дозволяють користувачам отримувати доступ до мережевих ресурсів у будь-якому місці та в будь-який час.

Принтери та сканери забезпечують можливість спільного використання периферійних пристроїв кількома користувачами мережі.

IP-камери надають можливості відеоспостереження та безпеки з віддаленим доступом через мережу.

Мережеве обладнання в більшості випадків складається з [2]:

— комутаторів (Switch);
— маршрутизаторів (Router);
— концентраторів (Hub);
— точок доступу(Access Point);
— брандмауерів (Firewall).

Комутатори працюють на другому рівні моделі OSI та забезпечують інтелектуальну комутацію кадрів на основі MAC-адрес, створюючи окремі домени колізій для кожного порту та підвищуючи загальну продуктивність мережі.

Маршрутизатори функціонують на третьому рівні OSI та відповідають за пересилання пакетів між різними мережевими сегментами на основі IP-адрес, виконуючи алгоритми маршрутизації для визначення оптимального шляху доставки даних.

Концентратори являють собою найпростіші мережеві пристрої, які працюють на фізичному рівні та просто повторюють отримані сигнали на всі свої порти.

Точки доступу забезпечують бездротовий зв'язок між мобільними пристроями чи пристроями з WiFi доступом та дротовою мережею.

Брандмауери забезпечують захист мережі від несанкціонованого доступу та шкідливого трафіку через фільтрацію пакетів відповідно до встановлених правила безпеки.

Типи каналів провідного зв’язку представлені [3]:

— вита пара (Twisted pair cable);

— оптоволоконий кабель (Fiber optic cable);

— коаксіальний кабель (Coaxial cable).

Вита пара — це кабель, в якому два ізольованих мідних дроти скручені один навколо одного для зменшення перешкод або перехресних завад. Він використовує роз'єми RJ-45. Кабель цього типу зазвичай містить два або більше струмопровідних дроти, екранованих або не екранованих ізолятором, і поділяється на два типи: екрановану виту пару і неекрановану виту пару [4].

Також скручені пари дротів мають покриття для захисту від будь-яких пошкоджень. В таблиці 1.1 показані стандарти мідних кабелів [5].

Таблиця 1.1 – Стандарти мідних кабелів

Швидкість	Назва	IEEE стандарт	Інформаційна назва	Максимальна довжина
10 Мбіт	Ethernet	802.3i	10BASE-T	100м
100 Мбіт	Fast Ethernet	802.3u	100BASE-T	100м
1 Гбіт	Gigabit Eternet	802.3ab	1000BASE-T	100м
10 Гбіт	10 Gig Ethernet	802.3an	10GBASE-T	100м

Оптоволоконні кабелі використовують оптичні волокна, які виготовляються зі скляних сердечників, оточених декількома шарами покривного матеріалу, як правило, з ПВХ або тефлону [6]. Дані передаються у вигляді світлових сигналів, завдяки чому у волоконній оптиці немає проблем з перешкодами. Волоконна оптика може передавати сигнали на дуже великі відстані в порівнянні з витими парами або коаксіальними кабелями. Також опловолоконні кабелі здатні передавати інформацію з великою швидкістю.

В таблиці 1.2 показані стандарти оптоволоконних кабелів [7].

Таблиця 1.2 – Стандарти оптоволоконних кабелів

Швидкість	Тип кабелю	IEEE стандарт	Інформаційна назва	Максимальна довжина
1 Гбіт	Багатомодові або одномодові	802.3z	1000BASE-LX	550м (MM) 5 км (SM)
10 Гбіт	Багатомодові	802.3ae	10GBASE-SR	400м
10 Гбіт	Одномодові	802.3ae	10GBASE-LR	10км
10 Гбіт	Одномодові	802.3ae	10GBASE-ER	30км

Коаксіальний кабель використовується для передачі високочастотних електричних сигналів з низькими втратами. Він має мідний провідник посередині, який оточений діелектричним ізолятором, зазвичай виготовленим з ПВХ або тефлону. Зазвичай використовується в телефонних системах чи кабельному телебаченні. Основні типи кабелів зображені на рисунку 1.1.

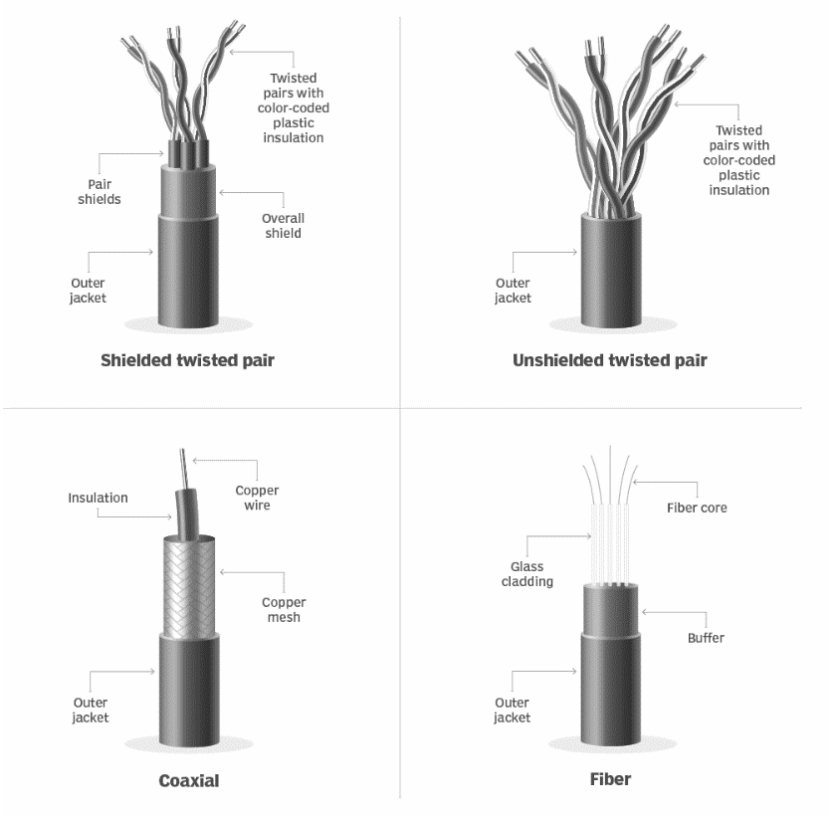


Рисунок 1.1 — Основні типи кабелів

Топологія мережі визначає спосіб фізичного та логічного з'єднання мережевих компонентів. Основні типи топологій включають [8]:

— шина (bus);

— зірка (star);

— кільце (ring).

Шинна топологія, також відома як топологія магістральної мережі. Ця конфігурація з'єднує всі пристрої з головним кабелем за допомогою відгалужувальних ліній [9]. Переваги цієї топології полягають у її простоті, оскільки потрібно менше кабелю, ніж в альтернативних топологіях, що полегшує встановлення.

Зіркоподібна топологія відома тим, що це найпоширеніша мережева топологія, яка з'єднує кожен пристрій у мережі з центральним концентратором, і пристрої можуть взаємодіяти один з одним лише опосередковано через центральний концентратор.

Кільцева топологія передбачає, що два виділені канали "точка-точка" з'єднують пристрій з двома пристроями, розташованими по обидва боки від нього, створюючи кільце пристроїв, через яке дані пересилаються через ретранслятори, поки не досягнуть цільового пристрою. На рисунку 1.2 зображено схему основних типів топологій.

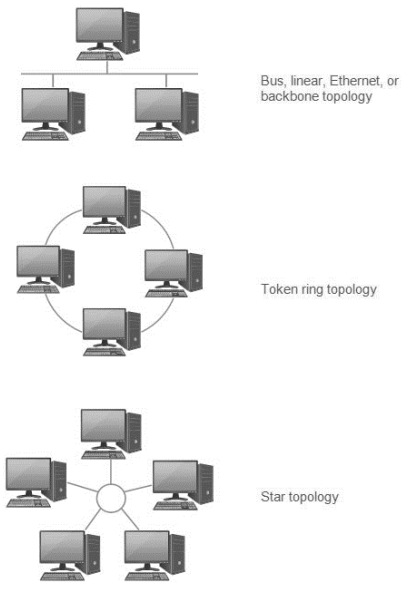


Рисунок 1.2 — Основні типи топології

Також існують і додаткові топології [10]:

— повне з’єднання;

— лінійна (ланцюгова);

— дерево;

— решітка;

— сніжинка;

— подвійне кільце.

Додаткові топології в більшості є похідними від основних.

Класифікація комп'ютерних мереж за географічним охопленням включає [11]:

— персональні мережі (PAN);

— локальні мережі (LAN);

— міські мережі (MAN);

— глобальні мережі (WAN).

Персональні мережі об’єднують персональні технологічні пристрої для зв'язку на невеликій відстані. Вони охоплюють лише менше 10 метрів площі. PAN має менше користувачів порівняно з іншими мережами, такими як LAN, WAN тощо. PAN зазвичай використовує певну форму бездротової технології. PAN передбачає передачу даних між інформаційними пристроями, такими як смартфони, персональні комп’ютери, планшети тощо.

Локальні мережі охоплюють територію одного будинку, офісу або кампусу. Тобто, вона з’єднує мережеві пристрої таким чином, що персональні комп’ютери та робочі станції можуть обмінюватися даними, інструментами та програмами. Група комп’ютерів і пристроїв з’єднана між собою комутатором або стеком комутаторів, використовуючи схему приватної адресації, визначену протоколом TCP/IP. Приватні адреси є унікальними по відношенню до інших комп'ютерів у локальній мережі. На межі локальної мережі знаходяться маршрутизатори, які з’єднують її з глобальною мережею.

Міська мережа охоплює більшу територію, ніж локальна мережа, і меншу територію порівняно з глобальною мережею Радіус дії MAN становить 5-50 км.

Вона з’єднує два або більше комп'ютерів, які знаходяться на відстані один від одного, але в одному або різних містах. Охоплює велику географічну територію і може служити в якості ISP (інтернет-провайдера).

Глобальна мережа — це комп'ютерна мережа, яка має велике охоплення, хоч вона може бути обмежена межами країни. WAN має радіус дії понад 50 км. Вона може являти собою з’єднання локальної мережі з іншими локальними мережами за допомогою телефонних ліній і радіохвиль і може бути обмеженою підприємством або доступною для громадськості. На рисунку 1.3 зображено позиції основних типів мереж за географічним розташуванням.

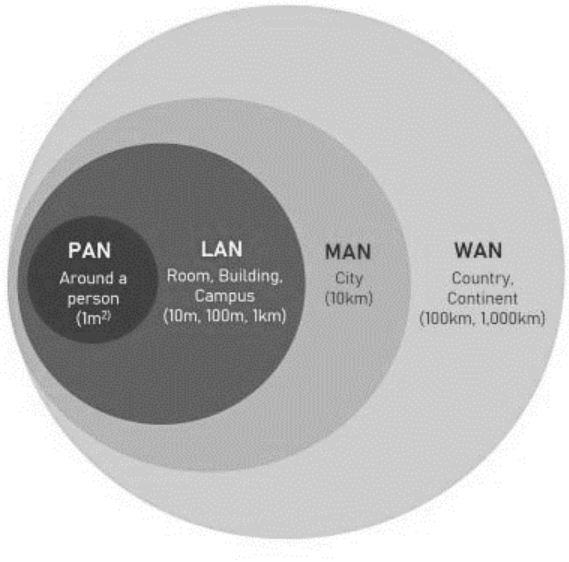


Рисунок 1.3 — Позиції основних типів мереж

1.2 Мережеві моделі OSI та TCP/IP

Передача даних — це процес чи дія, за допомогою якої ми можемо надсилати або отримувати дані. Для передачі даних існують дві моделі:

— модель OSI (Open Systems Interconnection);

— модель TCP/IP (Transmission Control Protocol/Internet Protocol).

Модель OSI — це набір правил, які пояснюють, як різні комп'ютерні системи взаємодіють через мережу. Модель забезпечує чітку структуру для передачі даних та управління мережевими проблемами, і також вона широко використовується як довідник для розуміння того, як функціонують мережеві системи.

Модель OSI складається з 7 рівнів:

— фізичний рівень (Physical Layer);
— канальний рівень (Data Link Layer););
— мережевий рівень (Network Layer);
— транспортний рівень (Transport Layer);
— сеансовий рівень (Session Layer);
— рівень представлення (Presentation Layer);
— рівень додатків (Application Layer).

Фізичний рівень є найнижчим рівнем моделі OSI, який відповідає за фізичне з'єднання між пристроями. Він містить інформацію у вигляді бітів та відповідає за передачу окремих бітів від одного вузла до іншого. При отриманні даних цей рівень отримує отриманий сигнал, перетворює його в 0 та 1 і надсилає на канальний рівень, який збирає кадр назад. Найпоширенішими пристроями цього рівня є концентратор і кабелі [12].

Канальний рівень відповідає за доставку повідомлення від вузла до вузла. Основною функцією цього рівня є забезпечення безпомилкової передачі даних від одного до іншого вузла через фізичний рівень. Пакет на канальному рівні називається кадром. Комутатори є поширеними пристроями канального рівня.

Мережевий рівень пов'язаний на передачі даних від одного до іншого пристрою, які знаходяться в різних мережах, і в тому числі він відповідає за маршрутизацію пакетів, вибір найкоротшого шляху для передачі пакета. Також IP-адреси відправника і одержувача розміщуються в заголовку на мережевому рівні. Сегмент на мережевому рівні називається пакетом. Поширеними пристроями мережевого рівня є маршрутизатори і комутатори.

Транспортний рівень надає послуги прикладному рівню і приймає послуги від мережевого рівня. Дані на транспортному рівні називаються сегментами. Цей рівень відповідає за повну доставку повного повідомлення, він також забезпечує підтвердження успішної передачі даних і повторно передає дані при виявленні помилки.

Сеансовий рівень відповідає за встановлення з'єднань, управління

з’єднаннями та завершення сеансів між двома пристроями.

Рівень представлення відповідає за обробку даних з прикладного рівня відповідно до необхідного формату для передачі мережею.

Найвищим на стеці моделі OSI знаходиться прикладний рівень, який реалізується мережевими додатками, які створюють дані для передачі через мережу. Цей рівень є вікном для доступу прикладних сервісів до мережі і для відображення інформації користувачу.

На рисунку 1.4 зображено структуру моделі OSI.

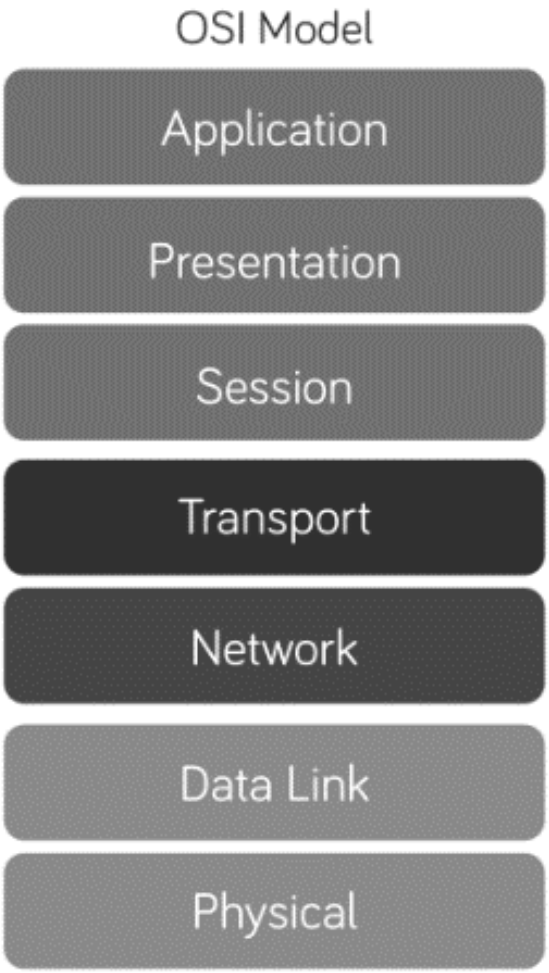


Рисунок 1.4 — Структура моделі OSI

Модель TCP/IP — це мережева структура, яка має 4 рівні, які забезпечують зв’язок між пристроям, та надає стандартизований набір протоколів для передачі даних через мережі, забезпечуючи ефективну та безпомилкову доставку. Вона

складається з чотирьох взаємопов’язаних рівнів, на відміну від семи рівнів у моделі OSI. Кожен рівень виконує певне завдання щодо даних, які передаються мережевим каналом, і дані рухаються від одного рівня до іншого.

Рівні моделі TCP/IP:

— прикладний рівень (Application Layer);

— транспортний рівень (Transport Layer);

— мережевий рівень (Network/Internet Layer);

— рівень доступу до мережі (Link Layer).

Прикладний рівень є найближчим рівнем до кінцевого користувача і є місцем, де знаходяться програми та користувацькі інтерфейси.

Транспортний рівень забезпечує надійну передачу даних між пристроями в правильному порядку.

Мережевий рівень керує маршрутизацією пакетів даних через мережі. Він використовує Internet Protocol (IP) для присвоєння унікальних IP-адрес пристроям і визначення найкращого в плані ефективності шляху доставки даних до місця призначення.

Рівень доступу до мережі є найнижчим в моделі TCP/IP і відповідає за фізичне з’єднання між пристроями в одній мережі.

На рисунку 1.5 зображено порівняння рівнів моделі OSI та моделі TCP/IP [13].

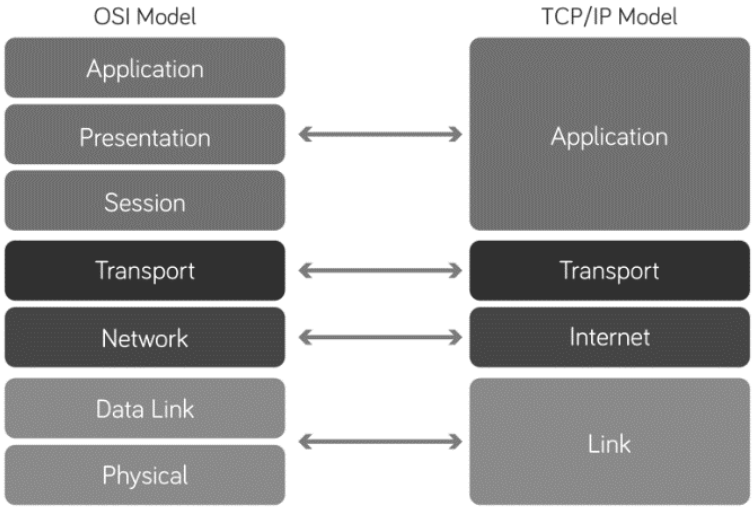


Рисунок 1.5 — Порівняння рівнів моделі OSI і моделі TCP/IP

1.3 IP-адресація та CIDR

Кожен пристрій, який підключений до мережі, повинен мати ідентифікатор. IP-адреси — це числові адреси, які використовуються для ідентифікації конкретного обладнання, підключеного до мережі.

Існує два типи версій IP:

— IPv4;

— IPv6.

IPv4-адреса складається з 32 бітів, що записуються у вигляді чотирьох десяткових чисел, розділених крапками, наприклад, 192.168.10.1, що в двійковому представленні буде виглядати як 11000000.10101000.00001010.00000001.

Адресний простір IPv4 поділяється на класи:

— клас A (1.0.0.0 - 126.255.255.255);

— клас B (128.0.0.0 - 191.255.255.255);

— клас C (192.0.0.0 - 223.255.255.255).

Також виділяються окремо приватні адреси 10.0.0.0/8, 172.16.0.0/12 та 192.168.0.0/16, які використовуються в приватних мережах.

IPv6-адреса складається з 128 бітів, що зазвичай записуються у вигляді восьми груп чотирьох шістнадцяткових чисел, розділених двокрапками, наприклад, 2001:0db8:85a3:0000:0000:8a2e:0370:7334.

Адресний простір в ній організований інакшим чином, і тому IPv6 не має чітких класів, але він використовує безкласову адресацію (CIDR).

Сучасний підхід до планування адресного простору базується на безкласовій міжмережевій маршрутизації (Classless Inter-Domain Routing), яка дозволяє гнучкіше використовувати IP-адреси. Вона використовує нотацію з префіксом, наприклад, 192.168.10.0/24, де число після слешу вказує кількість бітів мережевої частини адреси, а решта 8 біт є ідентифікатором хоста. Маска підмережі дозволяє оптимально розподілити адресний простір та забезпечити необхідну кількість адрес для кожного сегмента мережі. Процес поділу мережі на підмережі (subnetting) та об'єднання підмереж (supernetting) є ключовими інструментами для ефективного використання IP-адрес [14].

1.4 Основна інформація маршрутизацію

Маршрутизація — це процедура прийняття рішень, під час якої маршрутизатор обирає найкращий шлях для передачі даних від джерела до місця призначення.

Динамічна маршрутизація — це метод пошуку найкращого шляху для проходження даних через мережу. В цьому процесі маршрутизатор може передавати дані різними маршрутами і досягати місця призначення, виходячи зі стану каналів зв'язку на той момент.

Протоколи динамічної маршрутизації класифікуються за різними критеріями:

— алгоритм маршрутизації;

— область застосування;

— методи розповсюдження маршрутної інформації.

За алгоритмом маршрутизації протоколи поділяють:

— дистанційно-векторні (Distance Vector);

— протоколи стану каналу (Linked State);

— гібридні протоколи (Hybrid).

Алгоритм дистанційно-векторної маршрутизації використовується маршрутизаторами для пошуку найкращого шляху для передачі даних через мережу. Кожен маршрутизатор зберігає таблицю, яка показує найкоротшу відстань до кожного іншого маршрутизатора, виходячи з кількості хопів, необхідних для їх досягнення. Маршрутизатори обмінюються цією інформацією зі своїми сусідами, що дозволяє їм оновлювати свої таблиці і знаходити найефективніші маршрути. Використовує алгоритм Беллмана-Форда.

Протоколи стану каналу використовується маршрутизатори за станом зв'язку для обміну повідомленнями, які дозволяють кожному маршрутизатору вивчити всю топологію мережі, і на основі цієї вже вивченої топології кожен маршрутизатор може обчислити свою таблицю маршрутизації, використовуючи обчислення найкоротшого шляху. Для знаходження найкоротшого шляху використовується алгоритм Дейкстри.

Гібридні протоколи маршрутизації використовують елементи як протоколів маршрутизації за станом зв'язку, так і протоколів за вектором шляху. Такі протоколи використовують механізми обміну інформацією про маршрути між сусідніми маршрутизаторами, як у протоколах за вектором шляху, а також можуть будувати топологічну карту мережі в окремих областях, подібно до протоколів за станом зв'язку. Ці особливості дозволяють цьому протоколу швидше реагувати на зміни у топології та забезпечувати більш ефективне використання ресурсів мережі. Він використовує комбінацію алгоритмів Беллмана-Форда та Дейкстри для обчислення маршрутів [15].

За областю застосування протоколи маршрутизації поділяють:

— протокол внутрішнього шлюзу;

— протокол зовнішнього шлюзу.

Протоколи внутрішнього шлюзу використовуються для маршрутизації в межах однієї автономної системи, забезпечуючи швидку конвергенцію та ефективне використання ресурсів мережі.

Протоколи зовнішнього шлюзу призначені для обміну маршрутною інформацією між різними автономними системами.

На рисунку 1.6 зображено класифікацію протоколів маршрутизації.

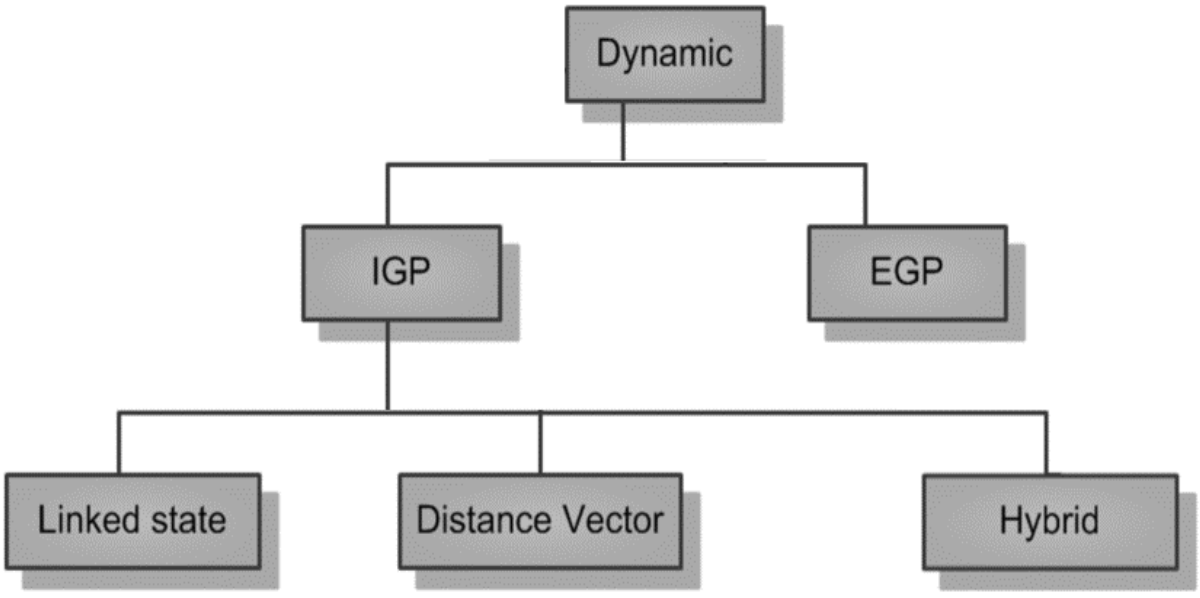


Рисунок 1.6 – Класифікація протоколів маршрутизації

1.5 OSPF (Open Shortest Path First)

Протокол OSPF — протокол маршрутизації стану каналу, який використовує алгоритм найкоротшого шляху Дейкстри для розрахунку оптимальних маршрутів на основі детальної інформації про топологію мережі. Протокол забезпечує доволі швидку конвергенцію, також він підтримує ієрархічну організацію мережі через концепцію областей та ефективно масштабується для великих корпоративних мереж та мереж інтернет-провайдерів.

Переваги протоколу OSPF:

— відкрита специфікація;

— швидка конвергенція;

— ефективна робота у великих мережах;

Недоліки протоколу OSPF:

— потреба ретельного планування дизайну мережі;

— складність у великих мережах через потребу у продуманій ієрархії.

Робота протоколу OSPF базується на створенні та підтримці бази даних стану каналів, яка містить повну інформацію про топологію мережевої області.

Кожен маршрутизатор OSPF збирає інформацію про стан своїх безпосередньо підключених каналів та поширює її всім іншим маршрутизаторам в області через повідомлення Link State Advertisement. На основі отриманої інформації кожен маршрутизатор будує повну карту мережі та розраховує найкоротші шляхи до всіх призначень. Процес роботи OSPF включає кілька етапів, починаючи з встановлення відносин сусідства між безпосередньо підключеними маршрутизаторами через обмін Hello-повідомленнями. Сусідські відносини встановлюються лише між маршрутизаторами, які знаходяться в одній області, мають однакові параметри Hello-інтервалу та Dead-інтервалу, та узгоджені налаштування автентифікації.

Після встановлення сусідства маршрутизатори синхронізують свої бази даних стану каналів через процес обміну Database Description повідомленнями, Link State Request та Link State Update повідомленнями [16].

1.6 Алгоритм Дейкстри в OSPF та побудова дерева найкоротших шляхів

Алгоритм Дейкстри є серцем протоколу OSPF. Він забезпечує розрахунок найкоротшого шляху від кожного маршрутизатора до всіх інших призначень у мережі на основі повної інформації про топологію області. Він працює ітеративно, на кожному кроці вибираючи вершину з найменшою відстанню від початкової точки та оновлюючи відстані до всіх сусідніх вершин.

Процес виконання алгоритму Дейкстри в OSPF починається з ініціалізації, коли маршрутизатор встановлює відстань до самого себе рівною нулю, а відстані до всіх інших вершин рівною нескінченності, далі маршрутизатор створює два множини вершин: множину необроблених вершин, яка з самого початку містить всі вершини мережі, і створює множину оброблених вершин, яка порожня на початку, і на кожній ітерації алгоритм вибирає з множини необроблених вершин ту, що має найменшу відстань від початкової точки, переміщує її до множини оброблених вершин та оновлює відстані до всіх її сусідів. Метрика OSPF базується на вартості каналу, яка розраховується як співвідношення еталонної пропускної здатності до фактичної пропускної здатності інтерфейсу, за формулою 1.1:

$$Cost = \frac{Reference\ Bandwidth}{Interface\ Bandwidth}, \tag{1.1}$$

де $Reference\ Bandwidth$ — еталонна пропускна здатність,

$Interface\ Bandwidth$ — фактична пропускна здатність інтерфейсу.

Еталонна пропускна здатність за замовчуванням становить 100 Мбіт/с (100,000,000 біт/с).

Результатом виконання алгоритму Дейкстри є дерево найкоротших шляхів з коренем у маршрутизаторі, який виконує розрахунок. Це дерево містить найкоротші шляхи до всіх призначень у мережі та використовується для побудови таблиці маршрутизації.

Маршрутизатор виконує алгоритм Дейкстри кожного разу, коли

відбуваються зміни в базі даних стану каналів, щоразу забезпечуючи актуальність маршрутної інформації.

1.7 Мережеві сервіси і служби

Мережеві сервіси та служби є важливими компонентами сучасних корпоративних мереж, які забезпечують автоматизацію конфігурації, розв'язання імен, веб-сервіси та інші критично важливі функції. Вони дозволяють централізовано керувати мережевими ресурсами, спрощувати адміністрування та підвищувати ефективність роботи мережевої інфраструктури.

1.7.1 Служба DHCP та DHCP Relay

Протокол DHCP — мережевий протокол, який використовується для автоматизації процесу призначення IP-адрес та інших параметрів мережевої конфігурації пристроям в мережі.

Замість ручного конфігурування IP-адреси та інших параметрів для кожного пристрою, DHCP дозволяє пристроям підключатися до мережі та отримувати всю необхідну мережеву інформацію автоматично від DHCP-сервера:

— IP-адреса;

— маска підмережі;

— шлюз за замовчуванням;

— адреси DNS-серверів.

Процес отримання IP-адреси через DHCP відбувається у чотири етапи:

— Discover;

— Offer;

— Request;

— Acknowledge.

Спочатку клієнт надсилає широкомовне повідомлення DHCP Discover для пошуку доступних DHCP-серверів, потім сервери відповідають повідомленням DHCP Offer, пропонуючи доступні IP-адреси та параметри конфігурації, після отримання повідомлення DHCP Offer клієнт вибирає одну з пропозицій та надсилає

DHCP Request, підтверджуючи вибір конкретного сервера, після якого обраний сервер надсилає DHCP Acknowledge, підтверджуючи призначення IP-адреси і передаючи всі необхідні параметри конфігурації [17].

DHCP Relay Agent вирішує проблему роботи DHCP у сегментованих мережах, де DHCP-сервер знаходиться в іншому сегменті від клієнтів. Оскільки DHCP використовує широкомовні повідомлення, які не проходять через маршрутизатори, DHCP Relay Agent перехоплює широкомовні DHCP-запити клієнтів та пересилає їх як unicast-повідомлення до віддаленого DHCP-сервера, і після отримання відповіді від сервера, Relay Agent пересилає її назад до клієнта.

Переваги DHCP:

— централізоване керування IP-адресами;

— автоматизація конфігурації мережевих параметрів;

— спрощення адміністрування великих мереж.

DHCP також підтримує резервування адрес для конкретних пристроїв за MAC-адресою та керування часом оренди адрес.

1.7.2 DNS-сервіси

DNS (Domain Name System) — система перетворення доменних імен у IP-адреси, яка забезпечує зручний спосіб доступу до мережевих ресурсів без необхідності запам’ятовування числових адрес. Вона функціонує як ієрархічна база даних, організована у вигляді дерева доменів з кореневими серверами на вершині ієрархії.

DNS включає:

— кореневі сервери;

— сервери доменів верхнього рівня;

— авторитетні сервери доменів.

Кореневі сервери містять інформацію про розташування серверів TLD, таких як .com, .org, .net та національних доменів.

Процес розв’язання DNS-імен може відбуватися рекурсивно або ітеративно. При рекурсивному розв’язанні DNS-сервер повністю обробляє запит клієнта,

звертаючись до інших серверів за необхідністю та повертаючи готову відповідь. При ітеративному розв’язанні сервер повертає посилання на інші сервери, залишаючи клієнту продовжувати процес розв'язання самостійно. Приклад роботи DNS показано на рисунку 1.7.

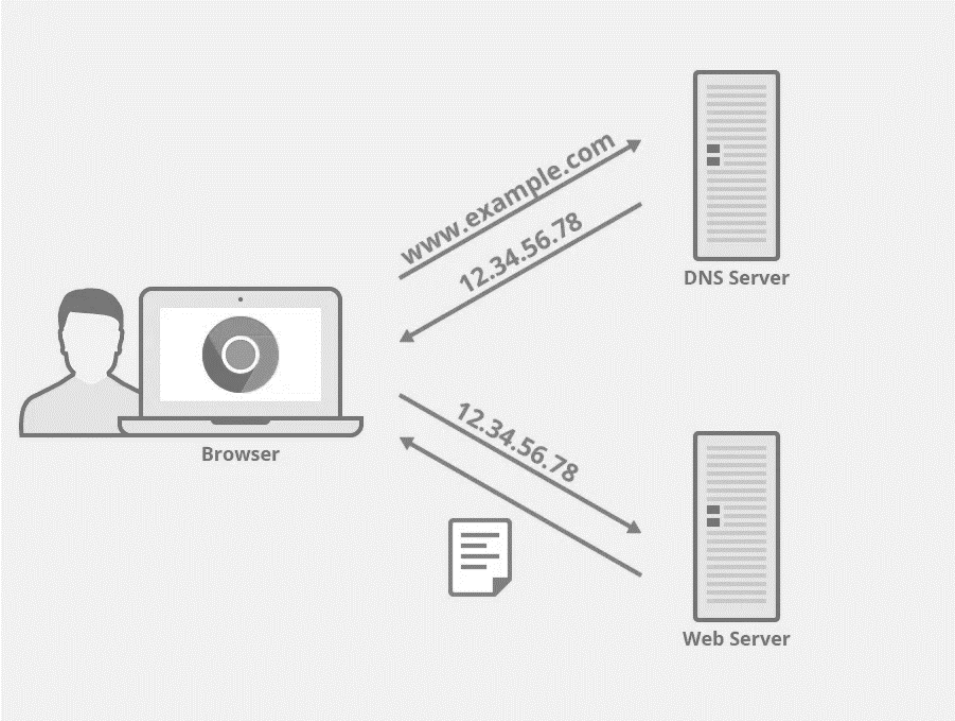


Рисунок 1.7 — Ілюстрація роботи DNS

DNS-кешування значно підвищує ефективність системи, зберігаючи результати попередніх запитів на локальних DNS-серверах та клієнтах. Час життя записів у кеші контролюється значенням TTL (Time To Live), що дозволяє балансувати між продуктивністю та актуальністю інформації. Корпоративні мережі зазвичай використовують внутрішні DNS-сервери для розв’язання локальних імен та пересилання зовнішніх запитів до публічних DNS-серверів.

1.7.3 HTTP-сервери

HTTP-сервери забезпечують доставку веб-контенту клієнтам.

Вони обробляють такі типи запитів:

— GET;

— POST;

— PUT;

— DELETE.

І після вони повертають відповіді з веб-сторінками та ресурсами.

Сучасні сервери підтримують:

— віртуальні хости для розміщення кількох сайтів на одному сервері;

— SSL/TLS шифрування для HTTPS-з’єднань;

— кешування і стиснення.

Також є HTTPS, який захищає трафік через SSL/TLS шифрування, а HTTP/2 та HTTP/3 протоколи оптимізують швидкість завантаження через мультиплексування та стиснення заголовків.

1.8 Технології забезпечення мережевої безпеки

Мережева безпека є дуже важливою в сучасних мережевих інфраструктурах. Технології мережевої безпеки включають:

— контроль доступу;

— шифрування трафіку;

— аутентифікацію користувачів;

— захист мережевого обладнання від несанкціонованого доступу.

1.8.1 Списки контролю доступу (Access Control Lists)

ACL (Access Control List) – це набір правил, визначених для контролю мережевого трафіку та зменшення кількості мережевих атак. Він використовується для фільтрації трафіку на основі набору правил, визначених для вхідного або вихідного трафіку.

Існують два основні типи ACL:

— стандартні;

— розширені.

Стандартні ACL фільтрують трафік тільки на основі IP-адреси джерела, що обмежує їх гнучкість, але спрощує конфігурацію для базових сценаріїв фільтрації.

Розширені ж ACL надають значно більше можливостей, оскільки вони

дозволяють фільтрувати трафік за такими параметрами:

— IP-адреса джерела та призначення;

— протоколи транспортного рівня;

— номери портів;

— прапорці TCP;

— інші параметри пакетів.

ACL можуть застосовуватися у двох напрямках:

— вхідний трафік;

— вихідний трафік.

Вхідний трафік фільтрується перед обробкою маршрутизатором, а вихідний трафік фільтрується після прийняття рішення про маршрутизацію.

Правила ACL обробляються послідовно зверху донизу, при цьому перше правило, що відповідає пакету, визначає дію, наприклад, дозволити або заборонити.

В кінці кожного ACL також неявно присутнє правило «deny all», яке блокує весь трафік, що не відповідає жодному з явно визначених правил.

Переваги ACL:

— підвищення продуктивності через фільтрацію небажаного трафіку;

— покращення безпеки через блокування небажаних з’єднань;

— контроль тррафіку.

1.8.2 Протокол SSH

SSH (Secure Shell) — це криптографічний мережевий протокол, який забезпечує безпечний доступ до віддалених систем через незахищені мережі.

Він заміняє небезпечні з точки зору безпеки протоколи Telnet та rlogin, надаючи шифрування всього трафіку, в тому числі і паролі, команди та передані дані, що запобігає витоку конфіденційної інформації.

SSH функціонує на основі архітектури клієнт-сервер, де SSH-сервер очікує з’єднання на стандартному порту 22.

Протокол забезпечує три основні сервіси безпеки:

— аутентифікацію для підтвердження ідентичності сторін;

— шифрування для захисту даних;

— перевірку цілісності.

Він також підтримує декілька методів аутентифікації:

— парольна аутентифікація;

— аутентифікація за допомогою відкритих ключів;

— багатофакторна аутентифікація.

Аутентифікація за допомогою відкритих ключів є найбільш безпечним методом, де користувач генерує пару криптографічних ключів: приватний ключ зберігається на клієнтській системі, а відкритий ключ розміщується на сервері, і під час аутентифікації сервер надсилає виклик, який клієнт підписує своїм приватним ключем, а сервер перевіряє підпис за допомогою відкритого ключа. Цей метод усуває необхідність передачі паролів через мережу і забезпечує набагато кращий рівень безпеки.

SSH також підтримує тунелювання, яке дозволяє безпечно передавати трафік інших протоколів через SSH-з’єднання.

1.8.3 Консольний режим

Консольний доступ забезпечує прямий локальний доступ до мережевого обладнання через фізичний консольний порт, що є критично важливим для початкової конфігурації, відновлення після збоїв та безпечного керування пристроями.

Консольне з’єднання не залежить від мережевої конфігурації та залишається доступним навіть у разі проблем з мережевою зв’язністю або неправильних налаштувань IP.

Фізичний доступ до консольного порту надає повний контроль над пристроєм, включаючи можливість скидання паролів, зміни конфігурації та перезавантаження системи.

На рисунку 1.8 зображено консольний порт на комутаторі Cisco Catalyst 2960-S Series 10G.

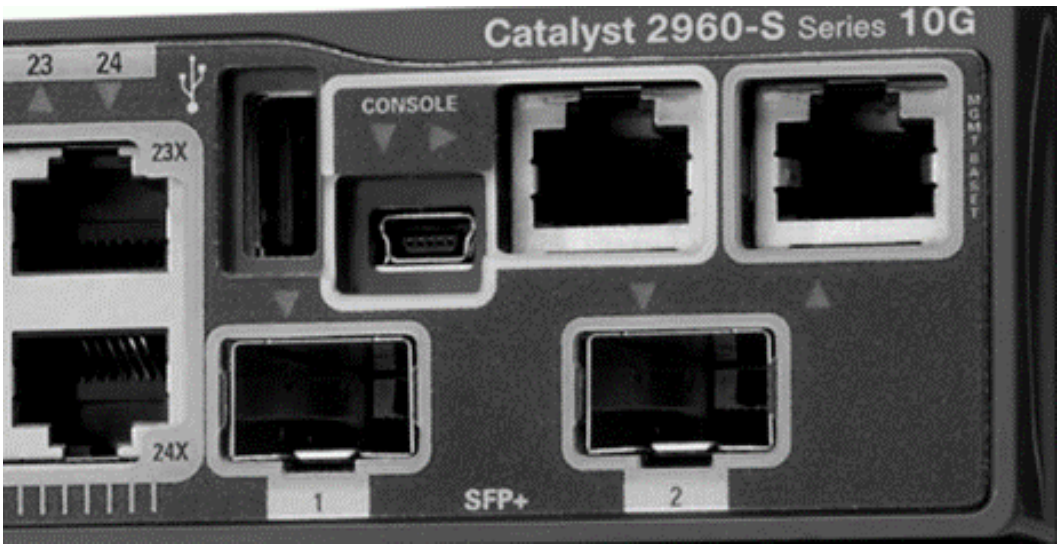


Рисунок 1.8 – Консольний порт на комутаторі Cisco Catalyst 2960-S

Конфігурація консольного доступу включає встановлення паролів для консольного порту, налаштування тайм-ауту сесії для автоматичного відключення неактивних з'єднань та обмеження кількості одночасних консольних сесій.

1.9 Технології з’єднання віддалених мереж

Важливим аспектом з’єднання віддалених мереж є забезпечення комунікації між географічно віддаленими точками. У сучасному світі організації часто мають декілька різних відділів чи офісів в різних регіонах, тому з’являється необхідність використання різноманітних технологій, які ефективно, а головне безпечно, поєднувати ці мережі з можливістю масштабування.

1.9.1 MPLS (Multiprotocol Label Switching)

MPLS — це технологія, що застосовується в телекомунікаційних мережах з високою пропускною здатністю з метою оптимізації маршрутизації потоків даних. На відміну від традиційних мережевих підходів, які базуються на використанні IP-адрес для визначення маршруту передавання, MPLS здійснює комутацію на основі коротких міток, що прикріплюються до пакетів даних. Ці мітки функціонують як ідентифікатори маршрутів, забезпечуючи швидке й упорядковане переміщення інформації через мережу. Застосування даного механізму сприяє підвищенню ефективності, стабільності та швидкодії в наданні таких послуг, як

широкосмуговий доступ до Інтернету, IP-телефонія, а також у розгортанні масштабованих VPN [18].

Переваги технології MLPS:

— швидкий трафік;

— менша затримка;

— внеможливлення перезавантаження обчисленнями;

— полегшений контроль якості обслуговування (QoS).

Недоліки технології MLPS:

— складність в управлінні;

— висока вартість впровадження і підтримки.

1.9.2 SD-WAN (Software Defined networking in a Wide Area Network)

SD-WAN — це технологія, яка дозволяє керувати мережами через програмне забезпечення, та має можливість використовувати різні комбінації з’єднання, щоб створити більш економічно вигідне з’єднання.

Переваги технології SD-WAN:

— адаптивність;

— включення в собі багатьох важливих функцій безпек;

— централізоване управління;

— спрощення налаштування;

— спрощення моніторингу.

Недоліки технології SD-WAN:

— залежність від Інтернет-з’єднання;

— складність у забезпеченні високої продуктивності в разі поганого налаштування.

1.9.3 IPSec (IP Security)

IPSec є технологією з’єднання віддалених мереж, яка забезпечує безпечне туннелювання трафіку через незахищені публічні мережі. На відміну від традиційних методів з'єднання віддалених локацій через виділені канали або frame

relay, IPSec створює віртуальні приватні мережі поверх існуючої IP-інфраструктури, дозволяючи організаціям економічно ефективно об’єднувати географічно розподілені офіси, філії та віддалених користувачів. Технологія забезпечує захищені тунелі між мережевими шлюзами різних локацій, гарантуючи конфіденційність, цілісність та автентичність переданих даних при проходженні через потенційно небезпечні мережеві сегменти.

Архітектура IPSec для з’єднання віддалених мереж базується на тунельному режимі роботи, де весь оригінальний IP-пакет інкапсулюється в новий IPSec пакет з заголовками мережевих шлюзів як відправника та одержувача, що дозволяє створювати Site-to-Site VPN з’єднання між корпоративними мережами, забезпечуючи прозорий доступ користувачів однієї мережі до ресурсів іншої, ніби вони знаходяться в єдиній локальній мережі [19].

Переваги технології IPSec:

краща масштабованість;

— прозорість для користувачів і додатків;

— високий рівень безпеки завдяки сучасним криптографічним алгоритмам;

— стандартизованість;

— сумісність між обладнанням різних виробників.

Недоліки технології IPSec:

— залежність від якості Інтернет-з’єднання;

— складність налаштування;

— потенційні проблеми з NAT і брандмауерами;

— складність моніторингу та діагностики мережевого трафіку;

— необхідність координації налаштувань між віддаленими локаціями.

Вибір технології для з’єднання віддалених мереж є критичним рішенням, яке визначає продуктивність, безпеку, масштабованість та економічну ефективність корпоративної мережевої інфраструктури.

Порівняння технологій з’єднання віддалених мереж чи точок показано в таблиці 1.3

Таблиця 1.3 – Порівняльна таблиця технологій

Критерій	MPLS	SD-WAN	IPSec
Принцип роботи	Комутація на основі коротких міток	Програмне управління з різними каналами зв'язку	Шифровані VPN тунелі через IP-інфраструктуру
Швидкість і затримка	Висока швидкість, мінімальна затримка	Залежить від налаштування та каналів	Залежить від якості Інтернет-з'єднання
Безпека	Приватна мережа провайдера	Вбудовані функції безпеки	Високий рівень криптографічного захисту
Масштабованість	Обмежена інфраструктурою провайдера	Висока адаптивність	Краща масштабованість
Складність управління	Складне управління і налаштування	Централізоване управління	Складність координації
Вартість	Висока вартість	Середня вартість	Низька вартість
Залежність від Інтернету	Приватна мережа	Повна залежність від Інтернет-з'єднання	Залежність від якості Інтернет-з'єднання
Сумісність	Залежить від провайдера	Може вимагати специфічного обладнання	Висока сумісність між виробниками
Гнучкість каналів	Тільки мережа провайдера	Множинні канали	Переважно Інтернет-канали

Різноманітні технології з’єднання віддалених мереж, включаючи MPLS, SD-WAN та IPSec, представляють унікальні підходи до вирішення завдань корпоративного зв'язку, кожен з яких оптимізований для специфічних сценаріїв використання та організаційних потреб. Наприклад, MPLS забезпечує гарантовану якість обслуговування та передбачувану продуктивність через приватну мережу провайдера, але вимагає значних капітальних інвестицій та багатьох годин впровадження. IPSec ж пропонує високий рівень криптографічного захисту та гнучкість використання існуючих Інтернет-каналів, проте може страждати від обмеженої пропускної здатності публічних мереж.

SD-WAN представляє програмний підхід до управління глобальними мережами, який дозволяє централізовано контролювати трафік через множинні канали зв'язку, включаючи MPLS, широкосмуговий Інтернет, LTE та супутникові з’єднання, і незважаючи на залежність від якості базових транспортних мереж, SD-WAN надає організаціям можливість динамічно оптимізувати використання доступних каналів, автоматично переключати трафік при погіршенні якості з'єднання та значно знижувати операційні витрати через спрощення управління мережею та зменшення залежності від дорогих MPLS каналів.

Процес вибору оптимальної технології з’єднання повинен враховувати специфічні вимоги організації до пропускної здатності, латентності, доступності, безпеки, географічного покриття та загальної вартості володіння. Гібридні підходи, які поєднують переваги різних технологій, часто представляють найбільш збалансоване рішення для складних корпоративних середовищ з різноманітними вимогами до зв’язку.

1.10 Веб-технології для розробки веб-сервісів

Сучасна веб-розробка вимагає комплексного підходу до вибору технологій, які забезпечують створення функціональних, надійних та масштабованих веб-додатків.

Ефективний веб-додаток складається з таких компонентів:

— серверна частини для обробки логіки додатку;

— система управління базами даних для зберігання та обробки інформації;

— інструмент для адміністрування та підтримки бази даних.

1.10.1 PHP для розробки веб-додатків

PHP (PHP: Hypertext Preprocessor) – це серверна мова програмування, яка має загальне призначення і була спеціально розроблена для веб-розробки. Вона виконується на сервері, генерує динамічний HTML-контент і взаємодіє з базами даних, що робить її ідеальним вибором для створення інтерактивних веб-додатків.

Переваги PHP:

— відкритий вихідний код;

— велика спільнота розробників;

— обширна документація;

— простота мови;

— підтримка об’єктно-орієнтованого програмування.

PHP широко використовується для створення динамічних веб-сайтів, систем управління контентом, платформ електронної комерції та веб-API.

1.10.2 Система управління базами даних MariaDB.

MariaDB — це відкрита реляційна система управління базами даних, яка була створена як форк MySQL з метою забезпечення сумісності та розширення функціональності. Також MariaDB зберігає повну бінарну сумісність з MySQL, що дозволяє легко мігрувати існуючі проєкти.

MariaDB має високу продуктивність через оптимізовані алгоритми індексації та кешування, які забезпечують швидкий доступ до даних при великих обсягах інформації. База даних є доволі легкомасштабованою, має можливість горизонтального і вертикального масштабування для обробки навантажень, і також високий рівень безпеки завдяки вбудованим механізмам аутентифікації, авторизації та шифрування даних.

Переваги MariaDB:

— повна сумісність з MySQL,

— регулярні оновлення і виправлення багів;

— покращена продуктивність порівняно з MySQL;

— відкритий вихідний код.

1.10.3 Інструменти адміністрування баз даних phpMyAdmin

phpMyAdmin — це веб-додаток з відкритим вихідним кодом, який написаний на PHP. Він надає зручний графічний інтерфейс для адміністрування MySQL та MariaDB серверів через веб-браузер. Веб-додаток дозволяє виконувати всі операції

з базами даних без необхідності майже без використання командного рядка.

Основний функціонал phpMyAdmin включає управління базами даних:

— створення, видалення та модифікація баз даних;

— імпорт та експорт структури і даних;

— копіювання баз даних між серверами.

Робота з таблицями включає:

— створення та редагування структури таблиць;

— додавання, модифікація та видалення записів;

— встановлення індексів та зовнішніх ключів;

— оптимізація таблиць.

Система забезпечує виконання SQL-запитів через вбудований редактор SQL з підсвічуванням синтаксису, також присутнє збереження та повторне використання запитів, додатково є аналіз продуктивності запитів.

Адміністрування користувачів включає:

— створення та управління обліковими записами;

— налаштування привілеїв та прав доступу;

— моніторинг активних з’єднань.

Переваги phpMyAdmin:

— інтуїтивно зрозумілий веб-інтерфейс;

— підтримка багатьох мов;

— можливість роботи з декількома серверами одночасно.

2 АНАЛІЗ ТА ОБҐРУНТУВАННЯ ВИБОРУ ТЕХНОЛОГІЙ ДЛЯ КОРПОРАТИВНОЇ МЕРЕЖІ

2.1 Аналіз структури та потреб компанії з віддаленими офісами

Сучасна корпоративна мережа повинна забезпечувати ефективну взаємодію між усіма структурними підрозділами компанії, незалежно від їх географічного розташування.

Аналізована компанія представляє собою типову модель сучасного підприємства з централізованим управлінням та розподіленою структурою обслуговування клієнтів.

Архітектура компанії базується на трирівневій моделі:

— головний офіс;

— віддалений офіс;

— серверна інфраструктура.

Головний офіс є основним відділом компанії, він є основним центом управління і координації в компанії. Віддалений офіс слугує як географічно віддалений відділ компанії. Також є серверна у вигляді географічно віддаленої серверної інфраструктури, яка географічно віддаленна від відділів компанії. Подібна структура дозволяє оптимізувати витрати на IT-інфраструктуру.

Серед важливих факторів, яке впливає на проєктування мережевої інфраструктури компанії є:

— географічна розподіленість;

— кількість пристроїв;

— безпека;

— масштабованість.

Географічна розподіленість визначає те, як далеко розташовані підрозділи компанії, що повністю впливає на мережу, на тип цієї мережі, на технологію з’єднання віддалених відділів.

Кількість пристроїв в мережі визначає те, наскільки навантажена мережа, визначає її складність, і відповідно, визначає тип самого мережевого обладнання.

Безпека є важливим фактором в кожній мережі, оскільки мережа має бути захищена від несанкціонованого доступу в її компоненти, також вона повинна бути захищена від зовнішніх і внутрішніх загроз.

Масштабованість враховує здатність мережі адаптуватись до зростання компанії, зростання її мережевої інфраструктури, тому вона повинна бути спроєктована таким чином, щоб потім після її розширення вона не потребувала повного перероблення, а тільки модернізації.

Функціональні вимоги до мережі включають забезпечення надійного зв’язку без затримки між всіма відділами та централізоване управління користувачами і ресурсами.

Технічні вимоги до мережі включають:

— висока масштабованість мережевої інфраструктури для підключення додаткових відділів;

— забезпечення пропускної здатності не менше 100 Мбіт/с для кожного відділу в компанії;

— захист від загроз;

Організаційна структура компанії передбачає розподіл між різними відділами, де головний офіс виконує переважно управлінські і фінансові функції наряду з обслуговуванням клієнтів, віддалений офіс забезпечує обслуговування клієнтів, а серверна інфраструктура обслуговує технологічні потреби всіх відділів компанії.

Враховуючи організаційну структуру компанії, де буде відбуватись інтенсивний обмін даними, важлива ставка буде робитись на ефективне і надійне з’єднаннями між відділеннями.

2.1.1 Аналіз структури і потреб головного офісу

Головний офіс має 2 поверхи. Перший поверх призначений для обслуговування клієнтів, а на другому поверсі відбувається адміністративна та фінансова діяльність. На рисунку 2.1 наведено структурну схему головного офісу компанії.

Рисунок 2.1 – Структурна схема головного офісу

Перший поверх має реєстратуру, де відбувається прийом і ідентифікація відвідувачів та направлення їх до спеціалістів, кабінет колл-центру, де відбувається телефонне обслуговування клієнтів, 5 кабінетів персоналу, де відбувається обслуговування клієнтів та гостьова кімната, де відбувається очікування клієнтів.

В кожному кабінеті потрібно в середньому 1-2 ПК та 1 принтер.

Для першого поверху критично важлива надійна і стабільна робота системи електронного документообігу та робота інформаційної системи для автоматизації процесів. В таблиці 2.1 показано потреби першого поверху головного офісу.

Таблиця 2.1 – Потреби першого поверху головного офісу

Тип кабінету	Кількість приміщень	Середня кількість ПК	Середня кількість принтерів	Всього ПК	Всього принтерів
Реєстратура	1	2	1	2	1
Кабінет колл-центру	1	2	1	4	1
Кабінети персоналу	5	1	1	6	5
Гостьова кімната	1	–	–	–	–
Разом	8	–	–	13	4

Другий поверх зосереджений на технічній підтримці, адміністративних і фінансових функціях компанії, тут присутня бухгалтерія, де ведеться фінансова діяльність, кабінет директора, в якому здійснюється адміністративне управління, кабінет персоналу для обслуговування клієнтів, та кабінет системного адміністратора, де відбувається технічна підтримка компанії, забезпечується централізоване управління мережею та підтримка і безпека інформаційної системи.

В кожному кабінеті потрібно в середньому 2-4 ПК та 1-2 принтери.

Для другого поверху важлива стабільна робота серверної файлової системи і системи електронного документообігу. В таблиці 2.2 показано потреби другого поверху головного офісу

Таблиця 2.2 – Потреби другого поверху головного офісу

Тип кабінету	Кількість приміщень	Середня кількість ПК	Середня кількість принтерів	Всього ПК	Всього принтерів
Бухгалтерія	1	4	1	4	1
Кабінет директора	1	1	1	1	1
Кабінети персоналу	1	2	1	2	1
Кабінет системного адміністратора	1	2	2	2	2
Разом	4	–	–	9	5

2.1.2 Аналіз структури і потреб віддаленого офісу

Віддалений офіс є географічно відокремленим відділом компанії, який функціонує як самостійний відділ компанії, але при цьому зберігаючи зв’язок із головним офісом.

Віддалений офіс розташований на 1 поверсі, та призначений в основному для обслуговування клієнтів.

Структурну схему віддаленого офісу компанії зображено рисунку 2.2.

Рисунок 2.2 – Структурна схема віддаленого офісу

Віддалений офіс має реєстратуру, де відбувається прийом, ідентифікація та направлення відвідувачів до відповідних спеціалістів, має 6 кабінетів персоналу для безпосередньо обслуговування клієнтів та зона очікування, де відбувається комфортне очікування клієнтів на прийом.

В кожному кабінеті у середньому потрібно 2 ПК та 1 принтер.

Особливістю віддаленого офісу є те, що там дуже критично важливе забезпечення постійного і безперебійного з’єднання з серверною та головним офісом, важлива стабільна робота системи електронного документообігу.

В таблиці 2.3 показано потреби другого поверху головного офісу.

Таблиця 2.3 – Потреби віддаленого офісу

Тип кабінету	Кількість приміщень	Середня кількість ПК	Середня кількість принтерів	Всього ПК	Всього принтерів
Бухгалтерія	1	4	1	4	1
Кабінет директора	1	1	1	1	1
Кабінети персоналу	1	2	1	2	1
Кабінет системного адміністратора	1	2	2	2	2
Разом	4	–	–	9	5

Забезпечення локального резервного копіювання важливих даних у віддаленому офісі також має вирішальне значення для забезпечення неперервності роботи у випадку тимчасових проблем зі зв’язком.

2.1.3 Аналіз структури і потреб серверної

Серверна інфраструктура — централізований технологічний центр компанії, який географічно відокремлений від основних офісів. Така архітектура дозволяє підвищити рівень безпеки та значно оптимізувати витрати на IT-обладнання, оскільки не потрібно для кожного офісу на місці ставити потрібне серверне обладнання.

Серверна інфраструктура має включати в себе:

— серверну стійку з основними серверами;

— систему зберігання даних;

— системи безперебійного живлення (UPS);

— мережеве обладнання для забезпечення зв’язку;

Також до серверної входить робоче місце системного адміністратора для управління серверною інфраструктурою.

Серверна стійка з основними серверами включає в себе файловий сервер, сервер баз даних та веб-сервер.

Система зберігання даних забезпечує централізоване зберігання даних з підтримкою технології RAID та різних протоколів для сумісності з різним обладнанням.

Системи безперебійного живлення забезпечать роботу серверної інфраструктури при відключенні електроенергії.

Мережеве обладнання включає маршрутизатори, комутатори, та інше обладнання, яке використовується для забезпечення зв’язку з різними відділами компанії.

Робоче місце системного адміністратора включає робочу станцію і систем иуправління серверами та моніторингу інфраструктури.

Потреби серверної зображено в таблиці 2.4.

Таблиця 2.4 – Потреби віддаленого офісу

Тип обладнання	Кількість	Призначення
Система зберігання даних	1	Файлове сховище та резервні копії
Веб-сервер	1	Хостинг внутрішнього веб-сервісу
DHCP-сервер	1	Динамічний розподіл адрес в мережі
DNS-сервер	1	Локальна доменна система
Сервер баз даних	1	Зберігання баз данних
Маршрутизатор	1	Інтернет, VPN
Комутатор	2	З’єднання серверів і ПК
Резервний сервер	1	Резервні копії всіх систем

2.2 Вибір і планування розподілу адресного простору

Для створення мережі компанії була вибрана мережа 192.168.0.0/16. Вона належить до діапазону приватних мереж (Клас C) IPv4, які зарезервувала Організація з присвоєння номерів в інтернеті (IANA) для використання на підприємствах без реєстрування і узгодження з Організацією з присвоєння номерів в інтернеті, реєстром Інтернету чи провайдером. Адреса мережі має маску підмережі 255.255.0.0 або префікс /16 в нотації CIDR, який означає, що мережна частина становить 16 бітів, а інші 16 бітів залишаються для адресації хостів.

Кількість доступних адрес в цій мережі дорівнює 65534 IP-адрес, з яких широкомовна і адреса мережі зарезервовані, а діапазон використовуваних адрес охоплює адреси від 192.168.0.1 до 192.168.255.255. Така велика кількість адрес виправдана тим, що в майбутньому планується масштабування мережі, тому вибір мережі 192.168.0.0/16 є гнучким і технічно правильним вибором.

Розподіл адресного простору відбувається за географічним принципом, що передбачає виділення окремих підмереж для кожного відділу компанії по фізичному місцю розташування. Також ми враховуємо те, що різні відділи компанії в майбутньому будуть планувати розширюватись, відповідно мережа теж буде

масштабуватись, тому потрібно виділяти набагато більше адрес, ніж та кількість, яка планує використовуватись. Потрібно враховувати і те, що компанія буде створювати нові офіси і відділи, тому потрібно буде резервувати окремі підмережі для майбутніх відділів компанії.

Також потрібно стандартизувати нумерацію, щоб використовувати її принципи однаково в кожному відділі компанії. Планується, що на кожен відділ чи офіс компанії буде виділятись одна підмережа з маскою 255.255.255.0 або префіксом /24 в нотації CIDR. Цій підмережі буде присвоєно значення в 3 октеті, наприклад, для підмережі серверної буде присвоєно адресу 192.168.50.x. Для кожного відділення резервується 9 підмереж для додаткових підвідділів чи поверхів. Виняток буде становити головний офіс, в якому 1 і 2 поверх матиме 192.168.10.x і 192.168.20.x адреси відповідно. Стандарт нумерації підмереж компанії показано в таблиці 2.5.

Таблиця 2.5 – Таблиця нумерації підмереж компанії

Відділ	Опис	Адреса підмережі
Резерв	Адреси, які зарезервовані на майбутнє	192.168.1.x
Головний офіс (1 поверх)	Адреси для 1 поверху головного офісу	192.168.10.x
Головний офіс (2 поверх)	Адреси для 2 поверху головного офісу	192.168.20.x
Віддалений офіс	Адреси для віддаленого офісу	192.168.30.x
Резерв	Адреси, які зарезервовані на майбутнє	192.168.40.x
Серверна	Адреси для серверної	192.168.50.x
Резерв	Адреси, які зарезервовані на майбутнє	192.168.60.x – 192.168.255.x

Після стандартизації нумерації підмереж, також потрібно стандартизувати принципи розподілу адрес в підмережі, щоб певний діапазон з цієї підмережі займали певні пристрої, от як наприклад, з діапазону 192.168.1.0 по 192.168.1.254 комп’ютери будуть займати адреси тільки з 192.168.1.1 по 192.168.1.50.

Стандарт нумерації адрес в підмережі показано в таблиці 2.6.

Таблиця 2.6 — Таблиця стандартизації розподілу нумерації IP-адрес в підмережі

Категорія	IP-діапазон	Призначення
Мережеве обладнання	192.168.x.1 – 192.168.x.30	Адреси длоя шлюзів роутеів, комутаторів
Резерв	192.168.x.33 – 192.168.x.62	Адреси для резервування
Комп’ютери	192.168.x.65 – 192.168.x.126	Адреси для ПК
Резерв	192.168.x.129 – 192.168.x.158	Адреси для резервування
Принтери	192.168.x.161 – 192.168.x.190	Адреси для принтерів
IP-телефони	192.168.x.193 – 192.168.x.222	Адреси для IP-телефонів
Резерв	192.168.x.224 – 192.168.x.254	Адреси для резервування

В кожній підмережі передбачається встановлення центральний маршрутизатора, забезпечуватиме з'єднання підмережі з іншими підмережами в мережі компанії. З’єднання в кабінетах буде виконуватись через локальні комутатори, які будуть з’єднуватись з магістральними комутаторами, а ті будуть мати з'єднання з центральним роутером відділу чи офісу. На рисунку 2.3 зображено структурну схему мережі компанії.

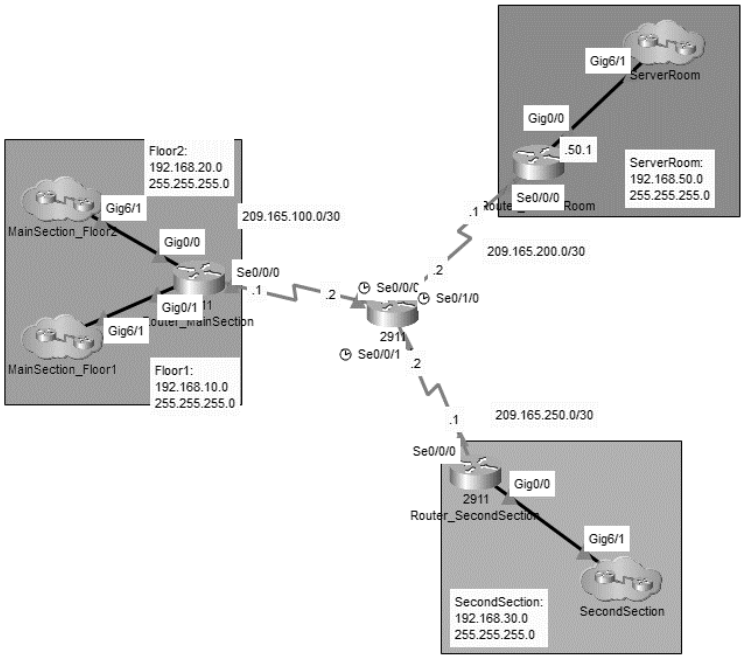


Рисунок 2.3 — Структурна схема мережі компанії

2.3 Обґрунтування вибору протоколу маршрутизації

На основі проведеного аналізу вимог обрано протокол OSPF версії 2 як основний протокол маршрутизації для мережі компанії. Використання алгоритму Дейкстри запобігає петлям маршрутизації, також обирається оптимальний маршрут на основі метрики cost.

Плюсом в сторону використання OSPF в мережі компанії є те, що в цей протокол маршрутизації забезпечує швидку збіжність завдяки Hello-пакетам, які надсилаються кожні 10 секунд для пошуку збоїв в сусідів, пакетам Dead, в яких інтервал 40 секунд для оголошення недоступності сусіднього маршрутизатора, що дозволяє оперативно і швидко виправити помилку в разі її появи і запобігти критичним випадкам. Серед плюсів також є те, що в OSPF є миттєве розповсюдження LSA при зміні топології та обчислення для великих мереж.

Ієрархічне масштабування через структуру областей дозволяє підтримувати до 65,535 маршрутизаторів в одній області і мати велику кількість областей.

Також тут присутнє зменшення розміру таблиць маршрутизації через сумаризацію шляхів та є підтримка маскування підмережі змінної довжини (VLSM), яке забезпечує ефективне використання IP-адрес.

Протокол підтримує різні типи мереж:

— точка-точка для WAN-з’єднань між відділенями;

— Frame Relay;

— MPLS;

— віртуальні канали (VPN) для маршрутизації між відділеннями через Інтернет.

В порівнянні з протоколами, які альтернативні OSPF, він показує значні переваги. RIP має критичні технічні обмеження:

— максимальна відстань 15 хопів, яка не відповідає гарній масштабованості;

— метрика hop count не враховує пропускну здатність каналів;

— повільна збіжність до 3 хвилин є критичною для компанії;

— передача повної таблиці маршрутизації кожні 30 секунд створює

надмірне навантаження.

На відміну від RIP, OSPF забезпечує необмежену кількість хопів, метрику на основі пропускної здатності, збіжність, яка відбувається за секунди, та передачу тільки змін топології.

EIGRP має певні переваги:

— швидшу збіжність через алгоритм DUAL;

— менше споживання CPU і пам’яті;

— підтримку нерівномірного балансування навантаження;

— автоматичне резюмування маршрутів.

Однак для компанії EIGRP має критичні недоліки:

— обмежений вибір мережевого обладнання;

— вищі ліцензійні витрати;

— складність інтеграції з обладнанням інших виробників.

А OSPF як відкритий стандарт — підтримується всіма основними виробниками, забезпечує більшу гнучкість у виборі обладнання та має широку базу знань та документації без додаткових ліцензійних витрат.

Приклад налаштування OSPF показано на лістингу 2.1:

Лістинг 2.1 — Приклад налаштування OSPF

```
en
conf t
router ospf 1
network 192.168.20.0 0.0.0.255 area 0
network 192.168.10.0 0.0.0.255 area 0
network 209.165.100.0 0.0.0.3 area 0
```

2.4 Обґрунтування вибору технології з’єднання

В процесі проєктування мережі компанії дуже велику увагу було приділено вибору технології з’єднання, яка буде забезпечувати високий рівень захищеності переданої інформації та буде забезпечувати стабільність роботи мережі навіть у випадках використання ненадійних каналів зв’язку.

Зважаючи на сучасні вимоги до безпеки сучасних мереж, найбільш доцільним рішенням є впровадження IPSec, який є універсальним та

масштабованим протоколом, що дозволяє реалізовувати захищені тунельні з'єднання між віддаленими підмережами.

Основною перевагою IPSec є можливість гнучкої інтеграції з існуючою мережею без потреби в спеціалізованому обладнанні, що робить IPSec дуже актуальним для компаній, що використовують мережу Інтернет для організації з'єднання між офісами чи відділенямми. Також використання цього протоколу дозволяє знизити витрати на організацію захищених каналів зв'язку, забезпечуючи при цьому високий рівень конфіденційності. Захищеність даних у IPSec досягається завдяки шифруванню пакетів даних, забезпечуючи конфіденційність переданої інформації і запобігає перехопленню даних. Окрім того, IPSec гарантує цілісність даних шляхом перевірки кожного пакета, що дозволяє вчасно виявляти спроби модифікації трафіку.

В мережі компанії планується застосовувати тунельний режим IPSec, який буде дозволяти ефективно організувати VPN-з'єднання між центральним, віддаленими офісами з серверною через мережу Інтернет.

IPSec має високу сумісність із сучасними протоколами маршрутизації, наприклад, OSPF, який викоористовується в мережі компанії. Ця сумісність особливо важлива для мережі компанії, де в майбутньому буде передбачено масштабування мережі та підключення нових офісів або відділень.

Але не рахуючи всі переваги, окрім IPSec в сучасних мережах також існують і альтернативи у вигляді SD-WAN і MPLS.

MPLS має:

— високий рівень стабільності;

— високу якість обслуговування;

— підтримка передачі даних з низькою затримкоюю.

Але для використання MPLS потрібно мати контракт з провайдером, що може бути набагато дорожче, також ймовірно що гнучкість подібного типу з'єднання буде гірше ніж в IPSec.

SD-WAN може використовувати IPSec як частину своєї архітектури для створення тунелів із захистом, але вона вимагає додаткових витрат на програмне

забезпечення.

Тому враховуючи всі плюси та недоліки, технологія IPSec була обрана як технологією для з’єднання віддалених мереж в компанії.

2.5 Обгрунтування використання DHCP та DHCP Relay

Для мережі компанії важливим завданням є автоматизація налаштування IP-адресів для всіх пристроїв у мережі, тому для цього було обрано протокол DHCP, який буде автоматично призначати IP-адреси, маски підмережі та DNS-сервер, що значно зменшить ризик помилки в налаштуванні майбутніх робочих станцій, принтерів чи інших пристроїв.

Оскільки в мережі компанії використовується кілька відокремлених підмереж, то важливо забезпечити коректну роботу DHCP для всіх цих сегментів, і тому було впроваджено механізм DHCP Relay, який матиме змогу передавати DHCP-запити та відповіді між підмережами без потреби розгортання окремого DHCP-сервера у кожній з цих підмереж, що дозволить централізоване управління налаштуванням IP-адрес, спростить адміністрування мережі і також дозволить скоротити витрати на IT-інфраструктуру. І до того ж, використання DHCP Relay спрощує масштабування мережі.

Приклад налаштування DHCP Relay показано на лістингу 2.1:

Лістинг 2.1 — Приклад налаштування DHCP Relay на маршрутизаторі

```
en
conf
interface Gig0/0
ip helper-address 192.168.50.11
interface Gig0/1
ip helper-address 192.168.50.12
```

2.6 Вибір мережевого обладнання

Було обрано обладнання, яке забезпечує стабільну роботу та підтримку необхідних мережевих протоколів і сервісів, зокрема OSPF, DHCP, ACL, SSH, IPSec та інших.

Основним маршрутизатором в мережі є Cisco 2911, який відповідає за

маршрутизацію між підмережами через протокол OSPF, і також який виконує функції DHCP Relay.

На ньому також налаштовані ACL для контролю доступу, та SSH для захищеного віддаленого доступу.

Для підключень в кімнатах використовуються комутатори Cisco Catalyst 2960, які відповідають за передачу даних і забезпечують швидку і надійну комутацію. Ці комутатори з'єднуються через проміжні комутатори Cisco Catalyst 2950T, які об'єднують трафік та підключаються до основного маршрутизатора Cisco 2911 в підмережі.

В таблиці 2.7 показано кількість мережевого обладнання у відділеннях компанії.

Таблиця 2.7 — Кількість мережевого обладнання у відділеннях компанії

Відділення	Cisco 2911	Cisco Catalyst 2950T	Cisco Catalyst 2960	DHCP-сервер	HTTP-сервер
Головний офіс (1 поверх)	1	4	7	–	–
Головний офіс (2 поверх)	–	2	4	–	–
Віддалений офіс	1	4	7	–	–
Серверна	1	1	1	1	1
Суммарна кількість	3	11	19	1	1

Обране обладнання та протоколи відповідають всім потребам, враховуючи резервні потреби, що дозволяє в майбутньому додавати нові робочі станції, принтери і інші пристрої в мережу без кардинальних чи масштабних змін в кмоп'ютерніймережі.

В таблиці 2.8 показано загальну кількість і ціни мережевого обладнання, яке потрібне для мережі компанії.

Таблиця 2.8 — Загальна кількість і ціни мережевого обладнання

Пристрій	Кількість	Ціна за одиницю	Суммарна ціна
Cisco 2911	3	70000 грн.	210000 грн.
Cisco Catalyst 2950T	11	3000 грн.	33000 грн.
Cisco Catalyst 2960	19	25000 грн	475000 грн
DHCP-сервер	1	30000 грн.	30000 грн.
HTTP-сервер	1	40000 грн.	40000 грн.
Суммарна ціна	–	–	788000 грн.

2.7 Використання PHP як основної мови програмування

В додаток до комп'ютерної мережі для прикладу було реалізовано веб-сервіс з базою даних на медичну тему для роботи з клієнтами, яка матиме наступні функції та можливості:

— додавання нових пацієнтів;

— додавання, видалення препаратів і змінення їх кількості;

— створення нових прийомів до лікарів і подальша зміна його статусу;

— додавання призначень до поточного прийому;

— різні ролі з різними можливостями.

Для розробки цього веб-сервісу було обрано мову програмування PHP, основними перевагами якої є:

— простий синтаксис;

— велика кількість бібліотек і фреймворків;

— гарна інтеграція з вебсервером Apache.

На відміну від таких мов як Python чи Node.js, PHP не потребує встановлення додаткових серверів чи складних налаштувань для запуску базового вебсервера, оскільки він працює в середовищі XAMP. До того ж, на відміну від ASP.NET, який орієнтований тільки на Windows, обрана мова розробки є кросплатформенною – вона легко встановлюється на Linux і має відкритий вихідний код.

2.8 Обгрунтування використання бази даних MariaDB

Для зберігання даних було обрано СУБД MariaDB, основними перевагами якої є:

— відкритий вихідний код;

— висока продуктивність і стабільність роботи при обробці великих обсягів даних;

— сумісність з MySQL.

На відміну від комерційних СУБД на кшталт Microsoft SQL Server чи Oracle Database, вибрана база даних є повністю відкритою. Також, в порівнянні з PostgreSQL, вибрана база даних має простіший механізм адміністрування і ширшу підтримку в спільноті, що робить її зручною для розгортання веб-сервісів.

2.9 Обгрунтування використання phpMyAdmin як додатку для адміністрування бази даних

Для керування обраною базою даних у вигляді MariaDB було використано phpMyAdmin, чиїми основними перевагами є:

— інтуїтивно зрозумілий інтерфейс;

— підтримка всіх основних операцій з базою даних;

— можливість керування базою даних з будь-якого пристрою через браузер.

На відміну від застосунків, таких як MySQL Workbench, обраний сервіс окрім сервера не потребує встановлення додаткового програмного забезпечення на комп’ютер, оскільки він працює безпосередньо у браузері. Також якщо порівнювати з більш простим Adminer, обраний сервіс має набагато ширший функціонал, що є плюсом при збільшенні обсягів даних.

3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ КОМП'ЮТЕРНОЇ МЕРЕЖІ

3.1 Проєктування логічної і фізичної топології мережі

Мережна інфраструктура компанії була спроєктована з врахуванням сучасних вимог до корпоративних мереж.

Фізична топологія мережі компанії побудована на базі зіркоподібної архітектури з чітким розподілом на три основні сегменти:

— головний офіс;

— віддалений офіс;

— серверна інфраструктура.

В головному офісі на першому поверсі встановлено 7 комутаторів Cisco Catalyst 2960 для підключення 13 ПК і 4 принтерів, які розподілені між реєстратурою, колл-центром та кабінетами персоналу.

Також використовується 4 магістальних комутатора Cisco Catalyst 2950T, які підключаються до центрального маршрутизатора Cisco 2911. На рисунку 3.1 зображено топологію мережі першого поверху головного офісу.

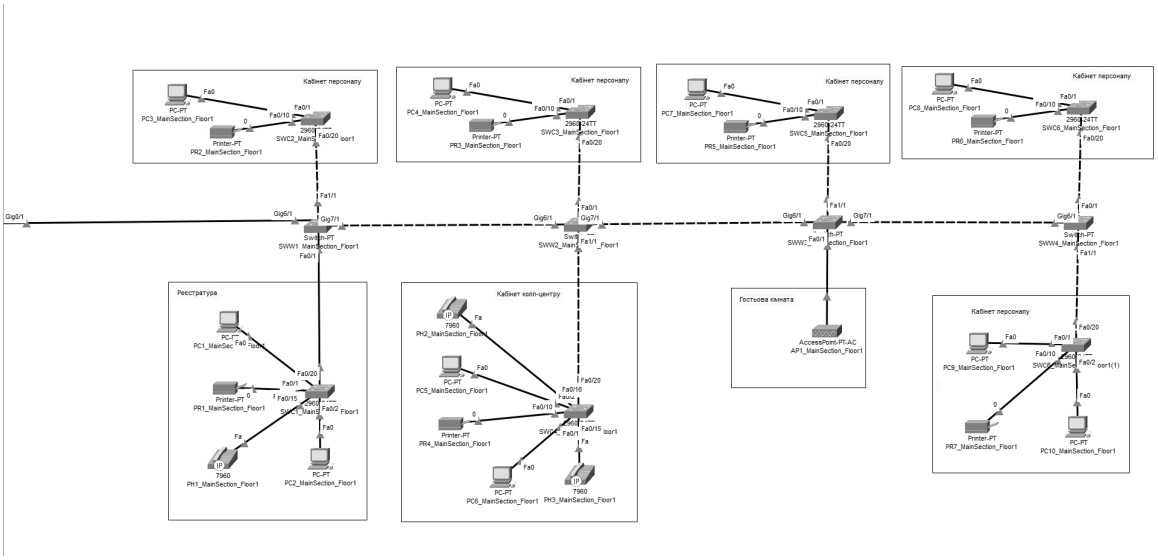


Рисунок 3.1 — Топологія мережі першого поверху головного офісу

Другий поверх головного офісу спроєктований з 4 комутаторами того ж типу, які обслуговують 9 комп'ютерів та 5 принтерів в бухгалтерії, кабінеті директора, в кабінеті системних адміністраторів та кабінету персоналу, а 2 магістральні

комутатори Cisco Catalyst 2950T об’єднують трафік і та забезпечують зв’язок із центральним маршрутизатором Cisco 2911. Топологію мережі другого поверху головного офісу зображено на рисунку 3.2.

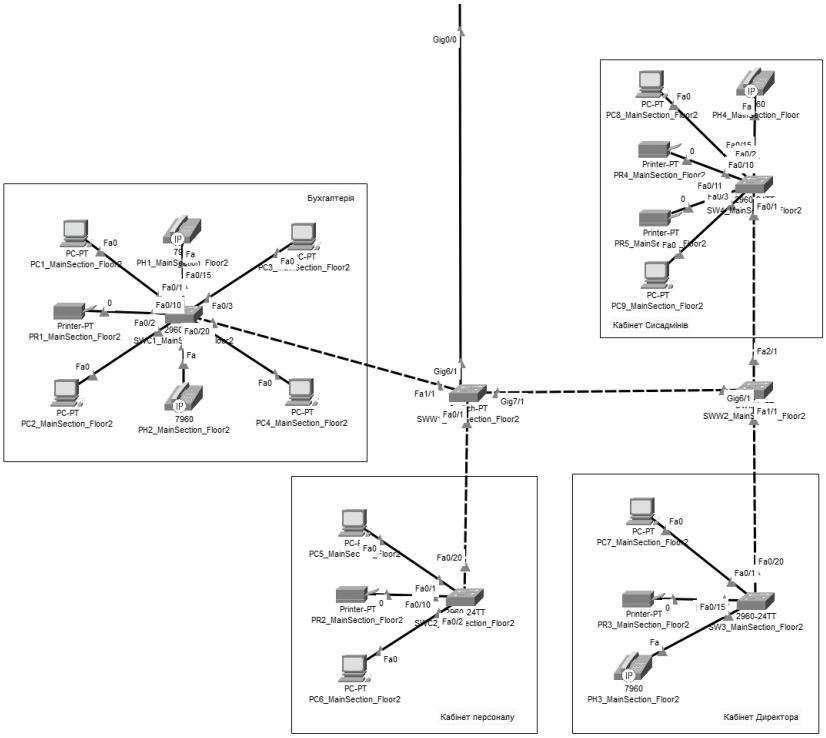


Рисунок 3.2 — Топологія мережі другого поверху головного офісу

Віддалений офіс має дуже схожу, але більш спрощену структуру, він використовує 7 комутаторів Cisco Catalyst 2960, які в певних кімнатах обслуговує взагалом 10 ПК та 7 принтерів (рис.3.2).

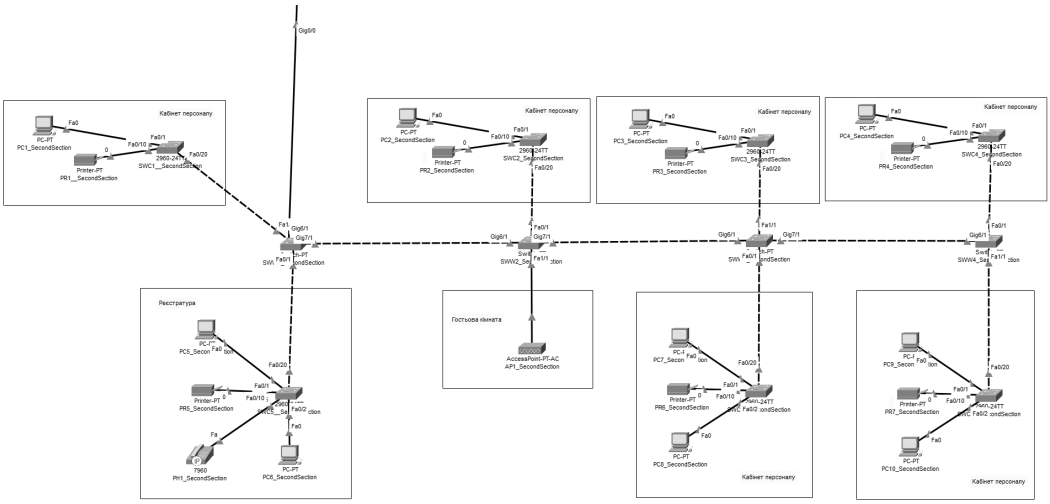


Рисунок 3.3 — Топологія мережі віддаленого офісу

Серверна інфраструктура, яка розміщена в серверній, включає маршрутизатор Cisco 2911, комутатори Cisco Catalyst 2950T та Cisco Catalyst 2960 відпоідно, та має набір серверів у вигляді серверів DHCP, DNS, та HTTP, що забезпечують стабільну роботу мережі. Вигляд топології мережі серверної зображено на рисунку 3.4.

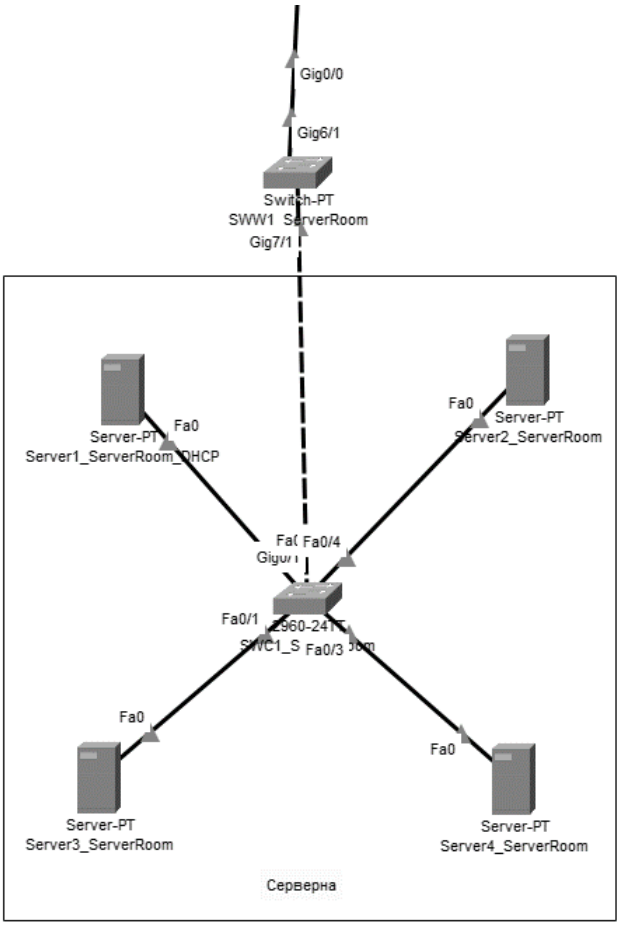


Рисунок 3.4 — Топологія мережі серверної

3.2 Налаштування і тестування мережевого обладнання

На всіх маршрутизаторах Cisco 2911 було вибрано певний інтерфейс через команду interface та налаштовано для кожного з’єднання, також були присвоєні перші адреси в кожній потрібній підмережі за допомогою команди ip address, де вказують потрібну IP-адресу, яку потрібно присвоїти, і маску підмережі. На лістингу 3.1 показано налаштування і призначення адрес інтерфейсам маршрутизатора головного офісу. Подібні налаштування проводимо також і з іншими маршрутизаторами.

Лістинг 3.1 — Налаштування маршрутизатора головного офісу

```
en
conf t
interface Gig0/0
ip address 192.168.20.1 255.255.255.0
ex
interface Gig0/1
ip address 192.168.10.1 255.255.255.0
```

Також є можливість налаштувати інтерфейсам машрутизатора адреси без використання команд через вбудовані налаштування Cisco Packet Tracer (рис. 3.5).

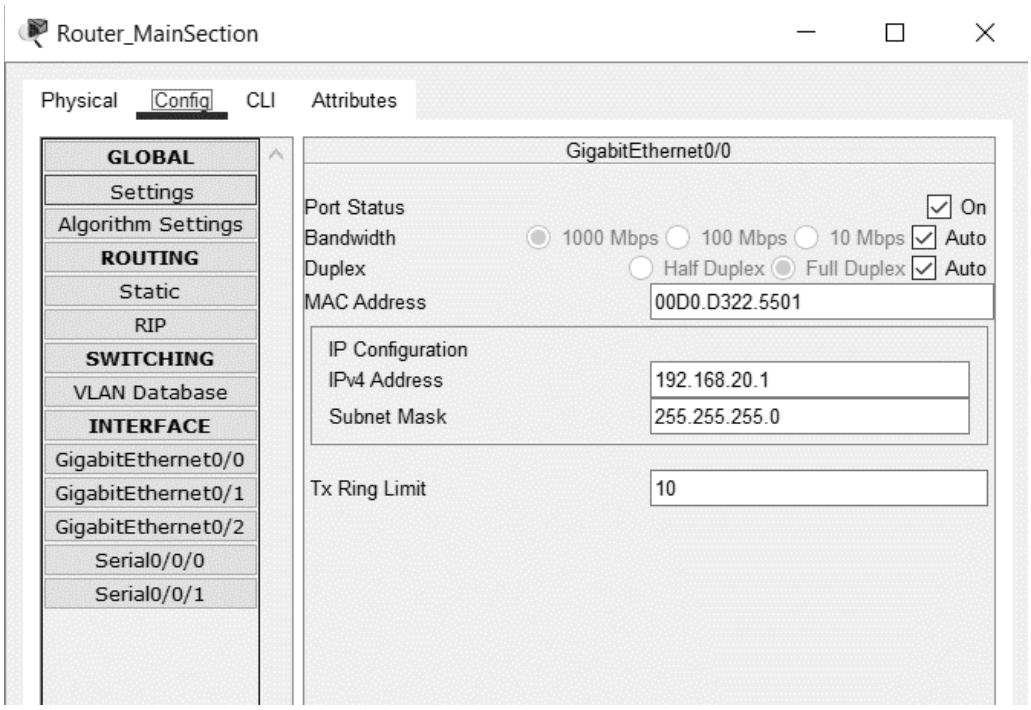


Рисунок 3.5 — Налаштування інтерфейсу маршрутизатора без використання команд

3.2.1 Налаштування і тестування протоколу OSPF

Після встановлення IP-адрес відповідним інтерфейсам, на всіх роутерах було запущено процес OSPF з номером 1 за допомогою команди router ospf 1. Після запущення процесів OSPF було оголошено мережі, які були підключені до машрутизаторів напряму через команду network з вказанням адреси підключеної напряму підмережі, інвертованої маски та області 0. Налаштування маршрутизатора центрального офісу показано на лістингу 3.2.

Лістинг 3.2 — Налаштування OSPF на маршрутизаторі головного офісу

```
en
conf t
router ospf 1
network 192.168.20.0 0.0.0.255 area 0
ex
network 192.168.10.0 0.0.0.255 area 0
ex
network 209.165.100.0 0.0.0.3 area 0
```

Після реалізації динамічної маршрутизації за допомогою протоколу OSPF на всіх маршрутизаторах в мережі потрібно провести тестування – перевірити їх корректну роботу шляхом перевірки записів таблиці маршрутизації шляхом введення команди show ip route. Також потрібно зробити перевірку мережі на правильність з'єднання шляхом відправки пакетів з ПК однієї мережі до ПК іншої.

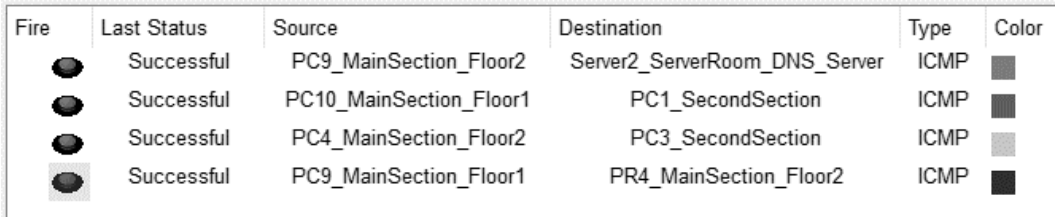
На рисунку 3.6 зображено таблицю маршрутизції маршрутизатора головного офіса, де на таблиці показано, що маршрутизатор успішно з'єднаний з мережею першого і другого поверху через протокол OSPF.

```
MainSection_Router#sh ip ro
MainSection_Router#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter ar
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.10.0/24 is directly connected, GigabitEthernet0/1
L       192.168.10.1/32 is directly connected, GigabitEthernet0/1
     192.168.20.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.20.0/24 is directly connected, GigabitEthernet0/0
L       192.168.20.1/32 is directly connected, GigabitEthernet0/0
O    192.168.30.0/24 [110/129] via 209.165.100.2, 00:25:56, Serial0/0/0
O    192.168.50.0/24 [110/129] via 209.165.100.2, 00:25:56, Serial0/0/0
     209.165.100.0/24 is variably subnetted, 2 subnets, 2 masks
C       209.165.100.0/30 is directly connected, Serial0/0/0
L       209.165.100.1/32 is directly connected, Serial0/0/0
     209.165.200.0/30 is subnetted, 1 subnets
O       209.165.200.0/30 [110/128] via 209.165.100.2, 00:25:56, Serial0/0/0
     209.165.250.0/30 is subnetted, 1 subnets
O       209.165.250.0/30 [110/128] via 209.165.100.2, 00:25:56, Serial0/0/0

MainSection_Router#
```

Рисунок 3.6 — Перевірка таблиці маршрутизації на маршрутизаторі головного офісу

Також в середовищі симуляції комп'ютерних мереж Cisco Packet Tracer є спосіб надіслати швидкий пакет для перевірки з'єднання між вдома пристроями, який зображується на рисунку 3.7. Він показує, що з'єднання між пристроями в

різних мережах успішне — пристрої приймають пакети даних та відповідають на них.



Fire	Last Status	Source	Destination	Type	Color
	Successful	PC9_MainSection_Floor2	Server2_ServerRoom_DNS_Server	ICMP	
	Successful	PC10_MainSection_Floor1	PC1_SecondSection	ICMP	
	Successful	PC4_MainSection_Floor2	PC3_SecondSection	ICMP	
	Successful	PC9_MainSection_Floor1	PR4_MainSection_Floor2	ICMP	

Рисунок 3.7 — Надсилання пакетів від одного пристрою на інший

3.2.2 Налаштування і тестування DNS, DHCP і HTTP серверів та DHCP Relay

Для автоматичного надання мережевих параметрів робочим станціям та іншим пристроям в мережі було реалізовано сервер DHCP, який був розміщений на сервері в серверній кімнаті головного офісу.

DHCP-сервер автоматично призначає IP-адресу, маску підмережі, шлюз за замовчуванням та адресу DNS-сервера.

Оскільки сервер розміщується лише в одній підмережі, для пристроїв з інших підмереж було налаштовано DHCP Relay на маршрутизаторі серверної кімнати, що дозволило пересилати DHCP-запити, які були отримані з інших інтерфейсів, до сервера, що обробляє запити централізовано.

Для цього було використано команду ip helper-address, яка вказує IP-адресу DHCP-сервера.

На лістингу 3.3 показано використання команди ip helper address на маршрутизаторі головного офісу.

Лістинг 3.3 — Налаштування IP Helper на

```
en
conf t
interface GigabitEthernet0/1
ip helper-address 192.168.10.2
exit
```

Рисунок інтерфейсу налаштування DHCP-сервера показано на рисунку 3.8 та результат тестування DHCP зображено на рисунку 3.9

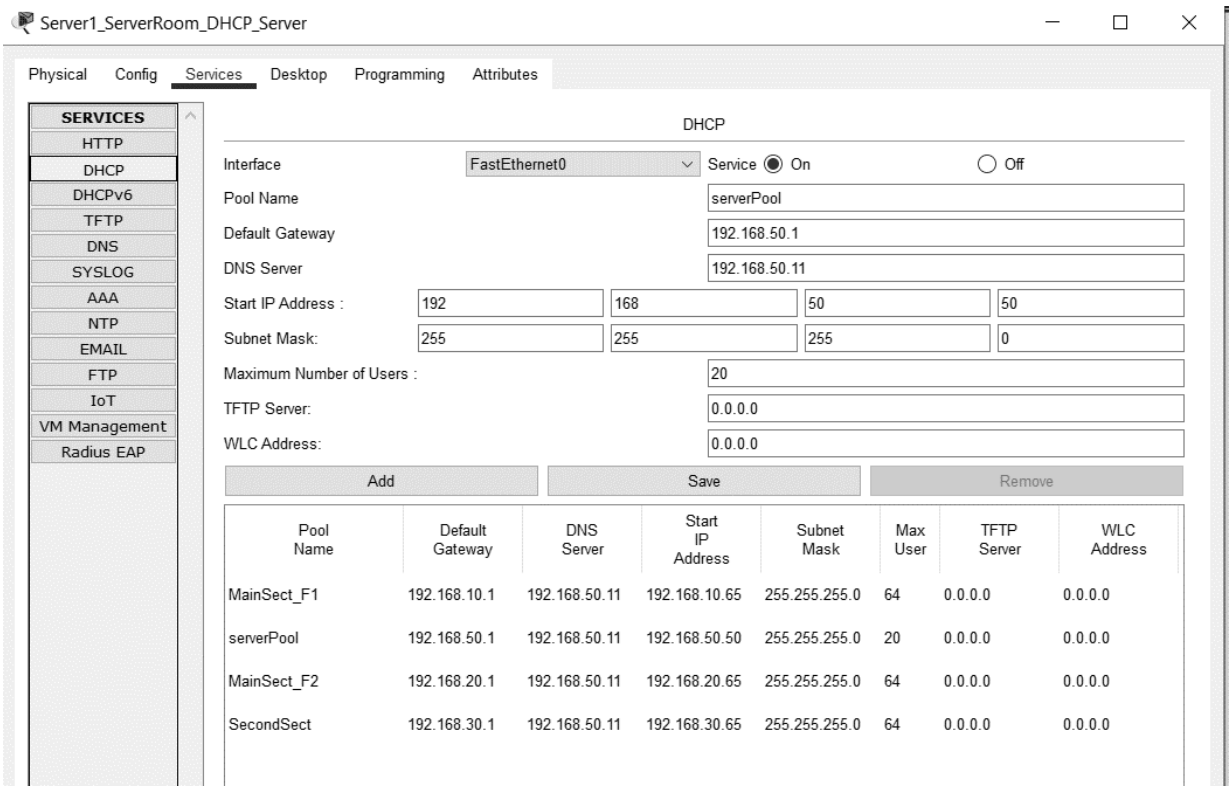


Рисунок 3.8 — Інтерфейс налаштування DHCP

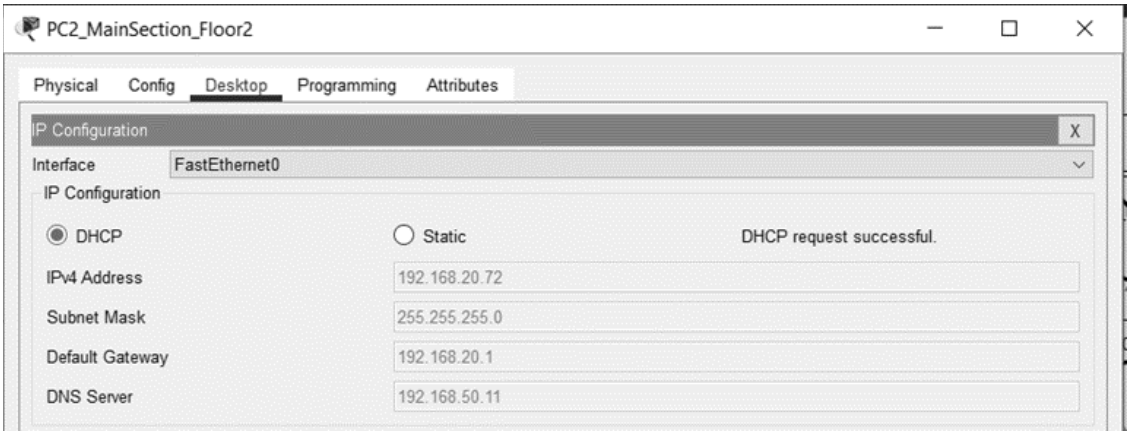


Рисунок 3.9 — Успішна робота DHCP

Також в мережі було налаштовано службу DNS, яка дозволяє клієнтам здійснювати перетворення доменних імен у відповідні IP-адреси (рис. 3.10).

До цього ж, адресу DNS-сервера було вказано у відповідних DHCP-параметрах, тому вона автоматично призначається клієнтам разом з іншими налаштуваннями, які надає їй DHCP-сервер та DHCP Relay Agent на маршрутизаторах.

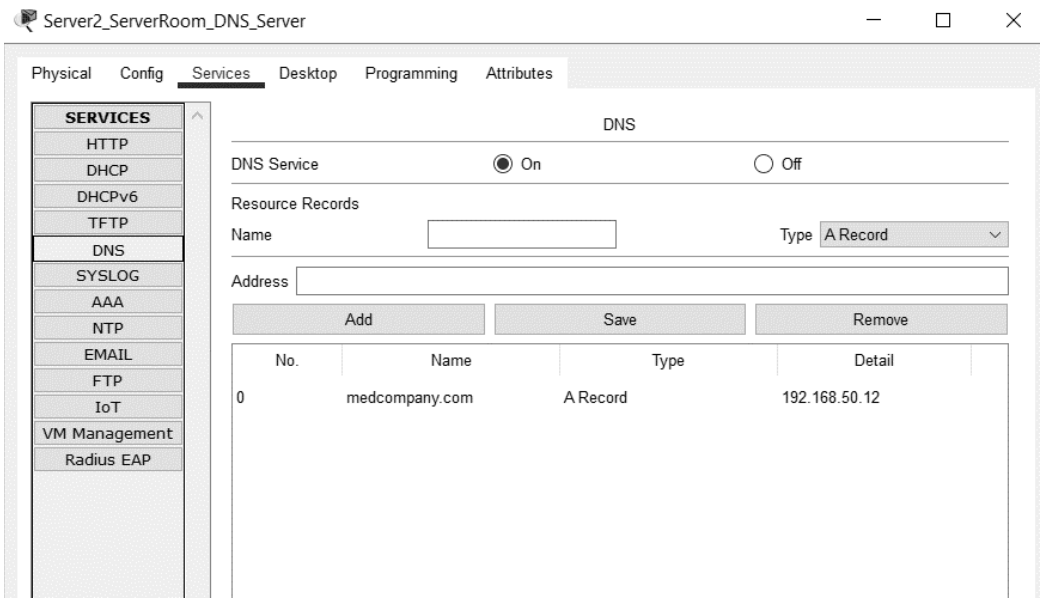


Рисунок 3.10 – Інтерфейс налаштування DNS

Також було налаштовано HTTP-сервер, на адресу якого буде переадресовуватись доменне ім’я medcompany.com. Налаштування HTTP-сервера показано на рисунку 3.11.

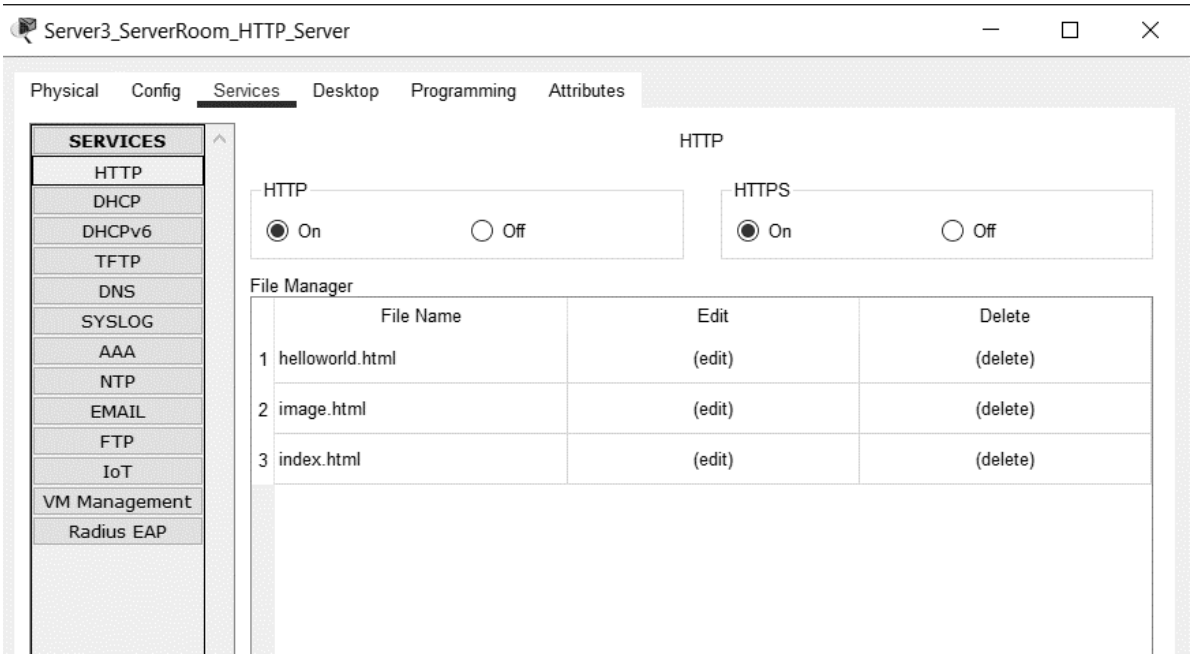


Рисунок 3.11 – Налаштування HTTP-сервера

Після налаштування DNS та HTTP-сервера було проведено перевірку роботи налаштованих сервісів, яка завершилась успіхом. Результат перевірки зображено на рисунку 3.12.

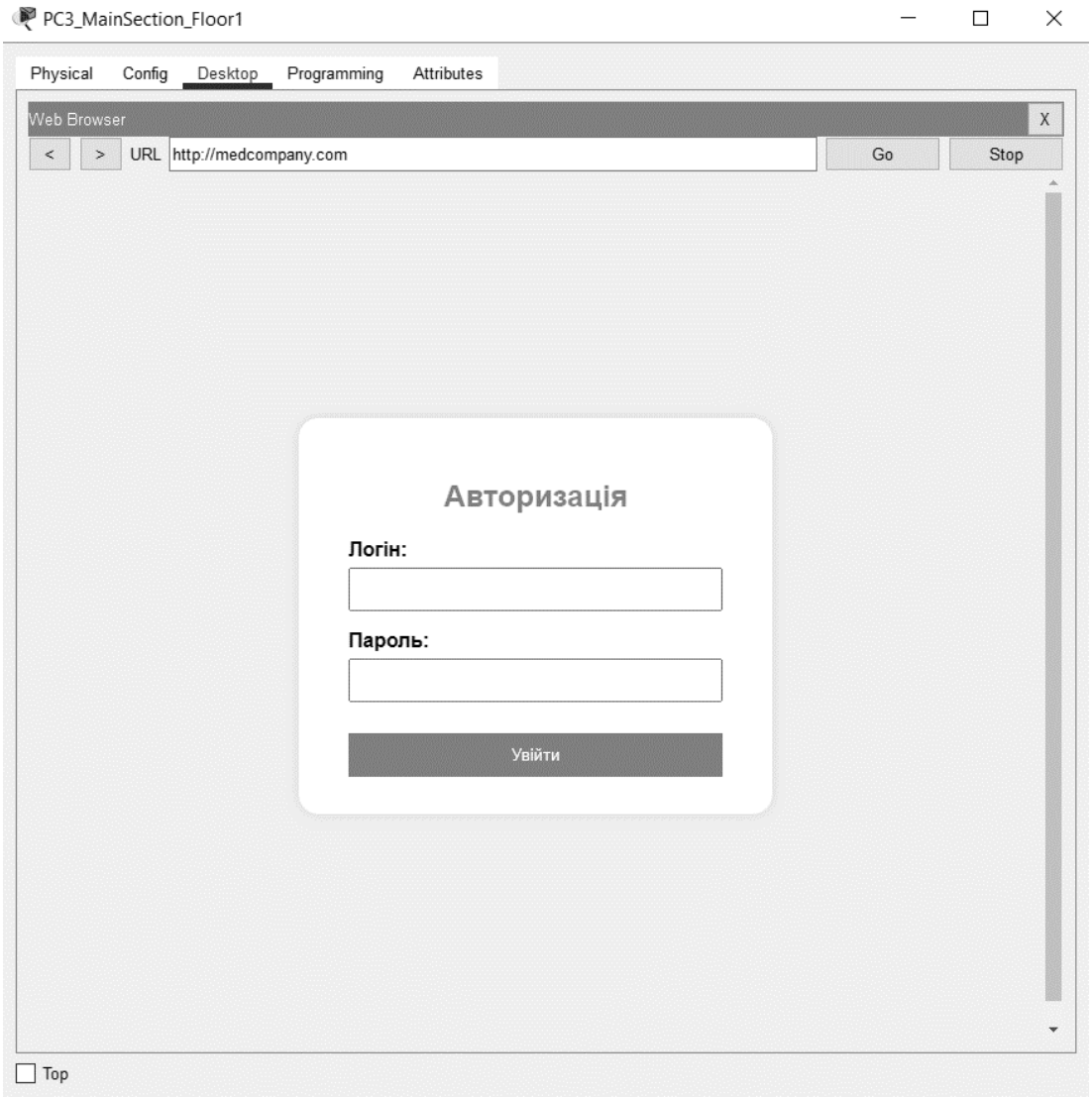


Рисунок 3.12 — Успішна перевірка DNS та HTTP-сервера

3.2.3 Налаштування і тестування ACL та SSH

Для того, щоб підвищити безпеку доступу до маршрутизаторів, було реалізовано захищений доступ до них за допомогою SSH. Налаштування SSH-доступу на маршрутизаторах показано в лістингу 3.3

Лістинг 3.3 — Налаштування SSH

```
ip ssh version 2
ip ssh authentication-retries 5
ip ssh time-out 100
username admin privilege 15 secret denus111
```

Тестування доступу приєднанню по SSH до маршрутизатора головного офісу показало, що з’єднання по SSH було налаштовано правильно, з’єднання з

маршрутизатором успішне, що показується на рисунку 3.13.

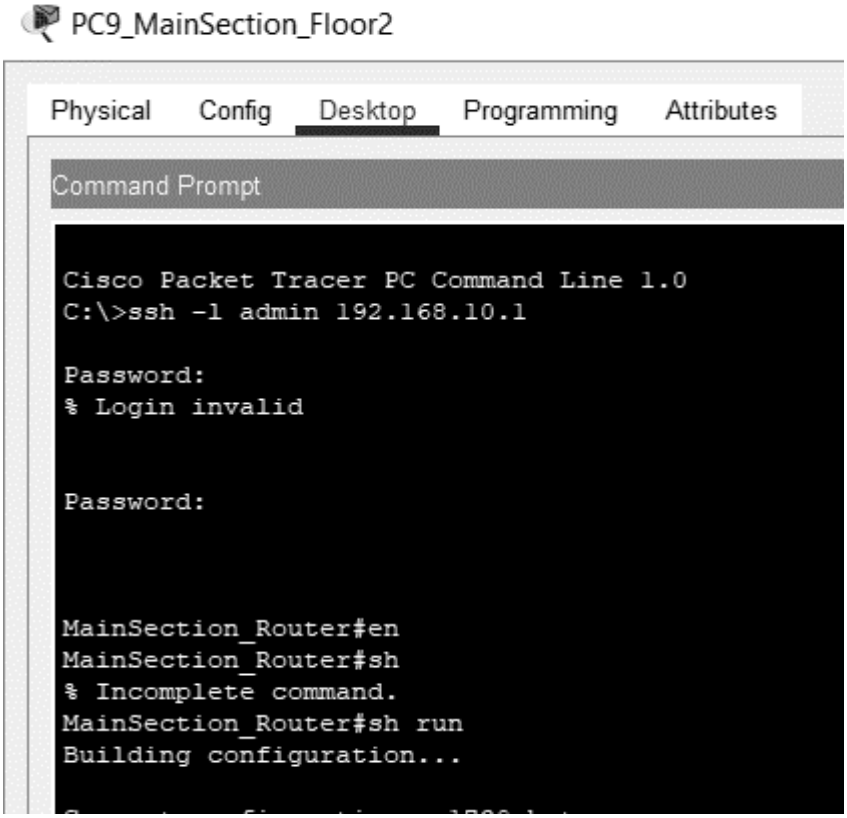


Рисунок 3.13 — Успішне з’єднання до маршрутизатора через SSH-доступ

Також було обмежено доступ до маршрутизатора лише для системних адміністраторів допомогою стандартного ACL. На кожному маршрутизаторі були виконані подібні команди. Виконання команд показано в лістингу 3.4.

Лістинг 3.4 — Створення списків доступу та обмеження доступу

```
access-list 50 permit host 192.168.20.10
access-list 50 permit host 192.168.20.11
line vty 0 4
access-class 50 in
login local
transport input ssh
```

Для перевірки обмеження було здійснено спробу входу до маршрутизатора по SSH з ПК, IP-адреси якого немає в списку доступу.

Спроба входу на рисунку 3.14 виявилась невдалою, що свідчить про успішну роботу налаштування списку доступу та успішну роботу захисту доступу до маршрутизаторів по протоколу SSH.

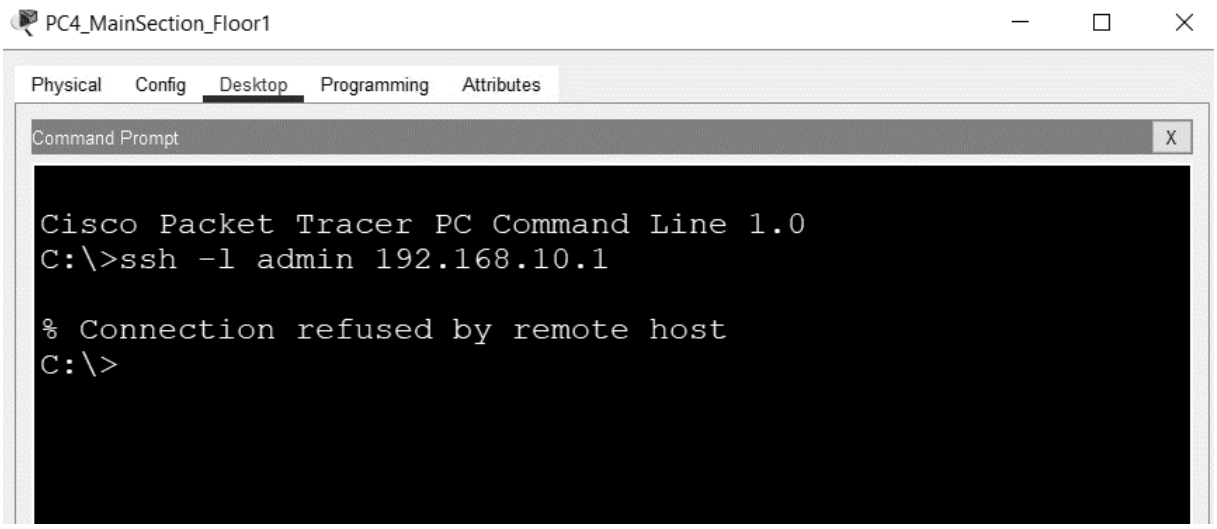


Рисунок 3.14 — Успішна робота списків доступу

3.2.4 Налаштування і тестування IPSec

Для захисту даних між головним офісом, серверною кімнатою та віддаленим офісом було реалізовано VPN-з’єднання на основі IPSec. Для початку використання IPSec на маршрутизаторі головного офісу активовано ліцензію securityk9, що показано на лістингу 3.5. Цю дію було виконано на всіх маршрутизаторах.

Лістинг 3.5 — Активування ліцензії securityK9

```
license udi pid CISCO2911/K9 sn FTX15245IU8-
license boot module c2900 technology-package securityk9
```

Правильність активування показано на рисунку 3.14 через команду показу версії маршрутизатора.

```
License Info:

License UDI:

-------------------------------------------------
Device#   PID                 SN
-------------------------------------------------
*0        CISCO2911/K9        FTX15245IU8-


Technology Package License Information for Module:'c2900'

-----------------------------------------------------------------
Technology    Technology-package          Technology-package
              Current       Type          Next reboot
-----------------------------------------------------------------
ipbase        ipbasek9      Permanent     ipbasek9
security      securityk9    Evaluation    securityk9
uc            disable       None          None
data          disable       None          None

Configuration register is 0x2102
```

Рисунок 3.14 — Перевірка активування ліцензії securitK9

Після активування ліцензії було налаштовано політику ISAKMP під номером 1 з шифруванням AES-256, автентифікацією через попередньо узгоджений ключ і використанням групи обміну ключами 5. Для цього створено ключ denus111 з вказанням IP-адрес VPN Інтернету — 209.165.200.1 та 209.165.250.1. Подібна операція була здійснена в інших маршрутизаторів по відношенню до сусідніх. Порядок і виконання команд показано в лістингу 3.6.

Лістинг 3.6 — Налаштування політики ISAKMP

```
crypto isakmp policy 1
encr aes 256
authentication pre-share
group 5
crypto isakmp key denus111 address 209.165.200.1
crypto isakmp key denus111 address 209.165.250.1
```

Далі створено набір перетворень VPN_Set, який задає, яким чином IPSec буде шифрувати й перевіряти трафік. А потім на основі цього набору було створено дві криптографічні карти для тунелювання:

— одна для з’єднання з серверною кімнатою;

— інша для з’єднання з віддаленим офісом.

Подібну операцію з налаштуванням було зроблено і на боці маршрутизатора серверної по відношенню до маршрутизатора головного офісу та до маршрутизатора віддаленого офісу, також подібна операція була здійснена на маршрутизаторі віддаленого офіса по відношенню дл маршрутизатора головного офісу і маршрутизатора серверної. Обидві карти використовують різні ACL які визначають дозволений VPN-трафік. Виконання команд показано на лістингу 3.7.

Лістинг 3.7 — Налаштування політики ISAKMP

```
crypto ipsec transform-set VPN_Set esp-aes esp-sha-hmac
crypto map VPN_Set 1 ipsec-isakmp
description vpn conns site second section
set peer 209.165.250.1
set transform-set VPN_Set
match address 100
crypto map VPN_Set 20 ipsec-isakmp
description vpn tunnel to server room
set peer 209.165.200.1
set transform-set VPN_Set
match address 100
```

На лістингу 3.7 ці карти були застосовані до серійного інтерфейсу маршрутизатора головного офісу, через який відбувається з'єднання з іншими вузлами.

Після налаштування IPSec на всіх маршрутизаторах, і створення тунелів один до одного — було виконано перевірку коректної роботи IPSec, і для цього потрібно виконати команду ping з IP-адреси однієї підмережі, на IP-адресу іншої підмережі, та перевірити кількість пакетів, отриманих маршрутизатором. На рисунку 3.15 зображено початкову кількість зашифрованих та дешифрованих маршрутизатором пакетів.

```
   local  ident (addr/mask/prot/port):
(192.168.20.0/255.255.255.0/0/0)
   remote  ident (addr/mask/prot/port):
(192.168.30.0/255.255.255.0/0/0)
   current_peer 209.165.250.1 port 500
    PERMIT, flags={origin_is_acl,}
   #pkts encaps: 3, #pkts encrypt: 3, #pkts digest: 0
   #pkts decaps: 2, #pkts decrypt: 2, #pkts verify: 0
   #pkts compressed: 0, #pkts decompressed: 0
   #pkts not compressed: 0, #pkts compr. failed: 0
   #pkts not decompressed: 0, #pkts decompress failed: 0
   #send errors 1, #recv errors 0

     local crypto endpt.: 209.165.100.1, remote crypto
endpt.:209.165.250.1
     path mtu 1500, ip mtu 1500, ip mtu idb Serial0/0/0
     current outbound spi: 0xC77C1CEC(3346799852)

     inbound esp sas:
      spi: 0x9BF444B7(2616476855)
        transform: esp-aes esp-sha-hmac ,
        in use settings ={Tunnel, }
        conn id: 2002, flow_id: FPGA:1, crypto map:
VPN_Set
        sa timing: remaining key lifetime (k/sec):
(4525504/3576)
        IV size: 16 bytes
```

Рисунок 3.15 — Початкова кількість пакетів

Після виконання команди пінгування IP-адреси іншої підмережі на рисунку 3.16 було показано, що кількість зашифрованих і розшифрованих пакетів збільшилась, що свідчить про успішне налаштування і про успішну роботу протоколу IPSec.

```
   local  ident (addr/mask/prot/port):
(192.168.20.0/255.255.255.0/0/0)
   remote  ident (addr/mask/prot/port):
(192.168.30.0/255.255.255.0/0/0)
   current_peer 209.165.250.1 port 500
    PERMIT, flags={origin_is_acl,}
   #pkts encaps: 11, #pkts encrypt: 11, #pkts digest: 0
   #pkts decaps: 10, #pkts decrypt: 10, #pkts verify: 0
   #pkts compressed: 0, #pkts decompressed: 0
   #pkts not compressed: 0, #pkts compr. failed: 0
   #pkts not decompressed: 0, #pkts decompress failed: 0
   #send errors 1, #recv errors 0

     local crypto endpt.: 209.165.100.1, remote crypto
endpt.:209.165.250.1
     path mtu 1500, ip mtu 1500, ip mtu idb Serial0/0/0
     current outbound spi: 0xC77C1CEC(3346799852)

     inbound esp sas:
      spi: 0x9BF444B7(2616476855)
        transform: esp-aes esp-sha-hmac ,
        in use settings ={Tunnel, }
        conn id: 2002, flow_id: FPGA:1, crypto map:
VPN_Set
        sa timing: remaining key lifetime (k/sec):
(4525504/2999)
        IV size: 16 bytes
        replay detection support: N
        Status: ACTIVE
```

Рисунок 3.16 — Кількість пакетів після виконання пінгування

3.2.5 Тестування комп’ютерної мережі

Для тестування мережі було обрано випадкові робочі станції, на яких було здійснено відправлення пакетів даних на інші робочі станції як в тій же мережі, так і поза нею. Успішний результат перевірки надсилання пакетів даних зображено на рисунку 3.17.

Fire	Last Status	Source	Destination	Type	Color
	Successful	PC2_MainSection_Floor2	PC9_MainSection_Floor1	ICMP	
	Successful	PC9_MainSection_Floor1	PC2_SecondSection	ICMP	
	Successful	PC1_MainSection_Floor2	PR7_SecondSection	ICMP	
	Successful	PC1_MainSection_Floor2	Server1_ServerRoom_DHCP_Server	ICMP	
	Successful	PC9_MainSection_Floor2	Server2_ServerRoom_DNS_Server	ICMP	
	Successful	PC3_SecondSection	Server2_ServerRoom_DNS_Server	ICMP	
	Successful	PC8_SecondSection	PC5_MainSection_Floor1	ICMP	
	Successful	PR2_SecondSection	PC9_MainSection_Floor2	ICMP	
	Successful	PC10_MainSection_Floor1	PR5_MainSection_Floor2	ICMP	

Рисунок 3.17 — Перевірка надсилання пакетів на випадкові пристрої

Також було здійснено перевірку на доступ до маршрутизаторів по SSH. Для цього було здійснено доступ до маршрутизатора з ПК, який не має IP-адреси ПК адміністратора, та ПК адміністратора. Результат успішної роботи показано на рисунку 3.18 від ПК зі звичайної адреси і результат на рисунку 3.19 від ПК з IP-адресою системного адміністратора.

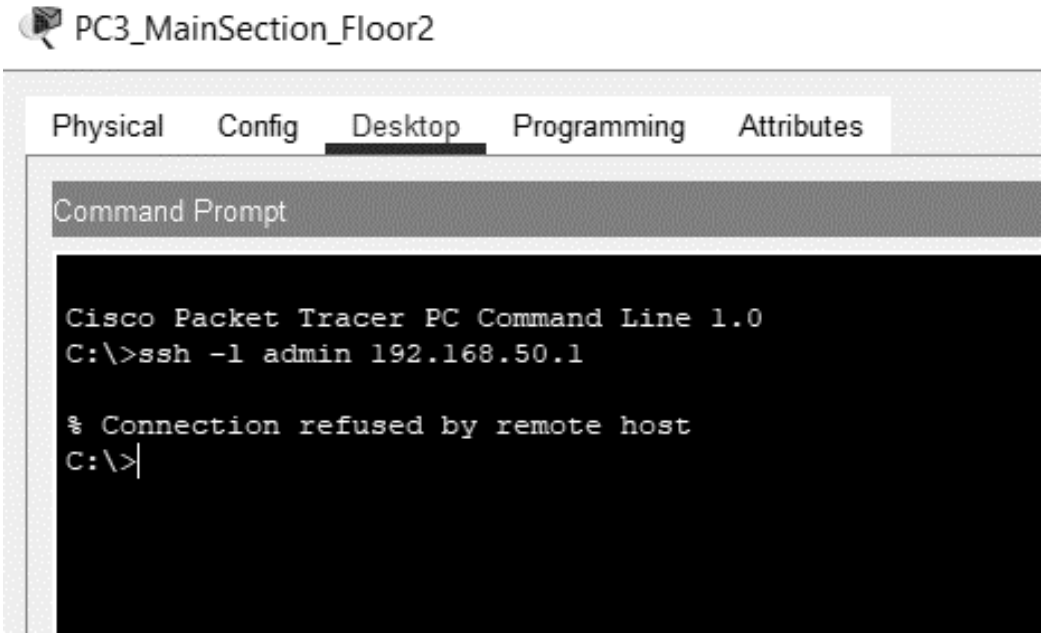


Рисунок 3.18 — Перевірка доступу до маршрутизатора зі звичайного ПК

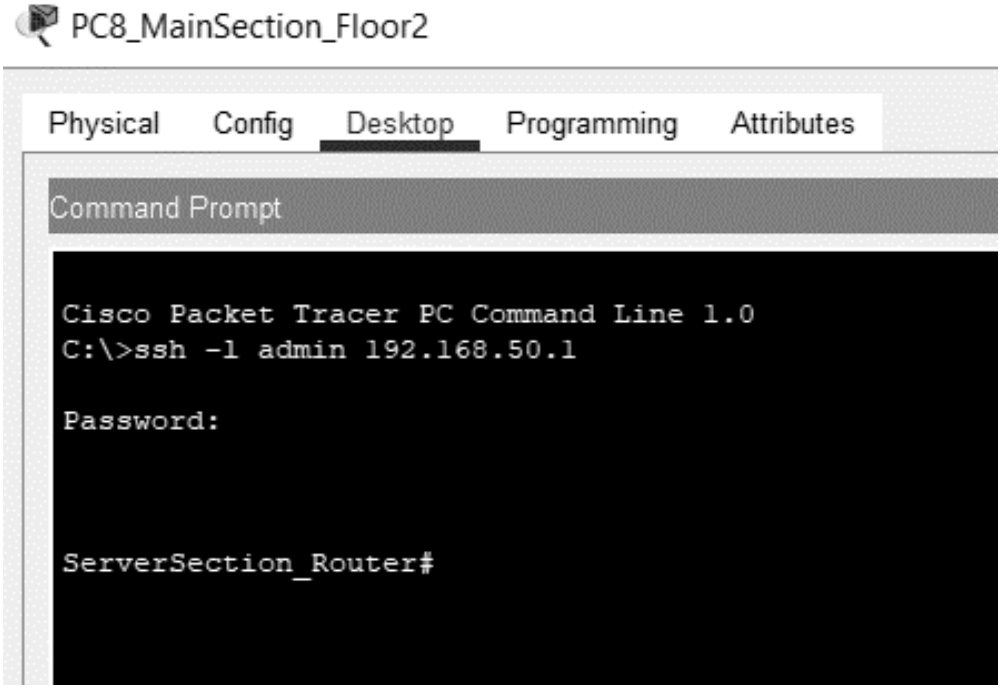


Рисунок 3.19 — Перевірка доступу з ПК системного адміністратора

Також потрібно здійснити тестування DNS та HTTP-сервера, ввівши в випадковий ПК адресу локального сайту medcompany.com, яка має IP-адрес 192.168.50.12. Після відображення сторінки авторизації можна вважати роботу DNS та HTTP-сервера успішним. Результат роботи зображено на рисунку 3.20.

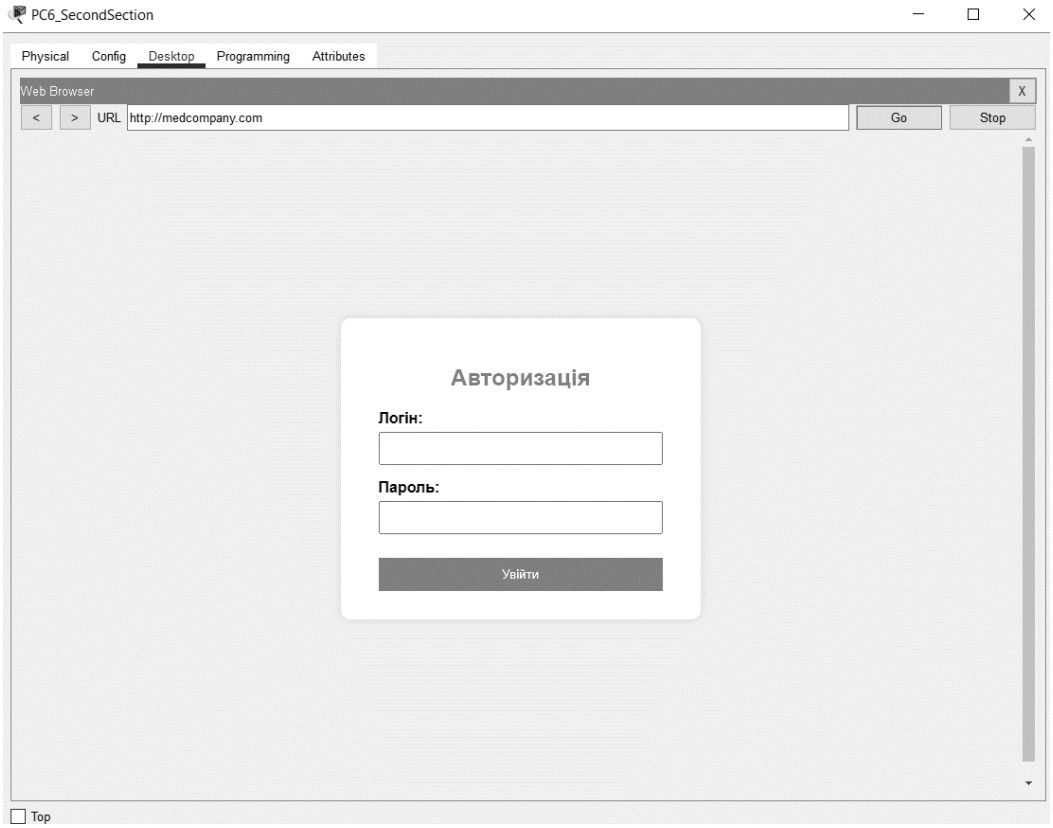


Рисунок 3.20 — Головна сторінка локального сайту

3.3 Розробка вебсервісу

Для реалізації веб-сервісу медичної системи обслуговування клієнтів було використано серверне середовище XAMPP з веб-сервером Apache, сервером баз даних MariaDB та інтерпретатором PHP (рис.3.21).

В розробленому веб-сервісі доступно:

— авторизація з різними ролями і доступами;

— управління графіками прийомів;

— перегляд і додавання пацієнтів;

— призначення лікування;

— додавання і змінення списку медикаментів.

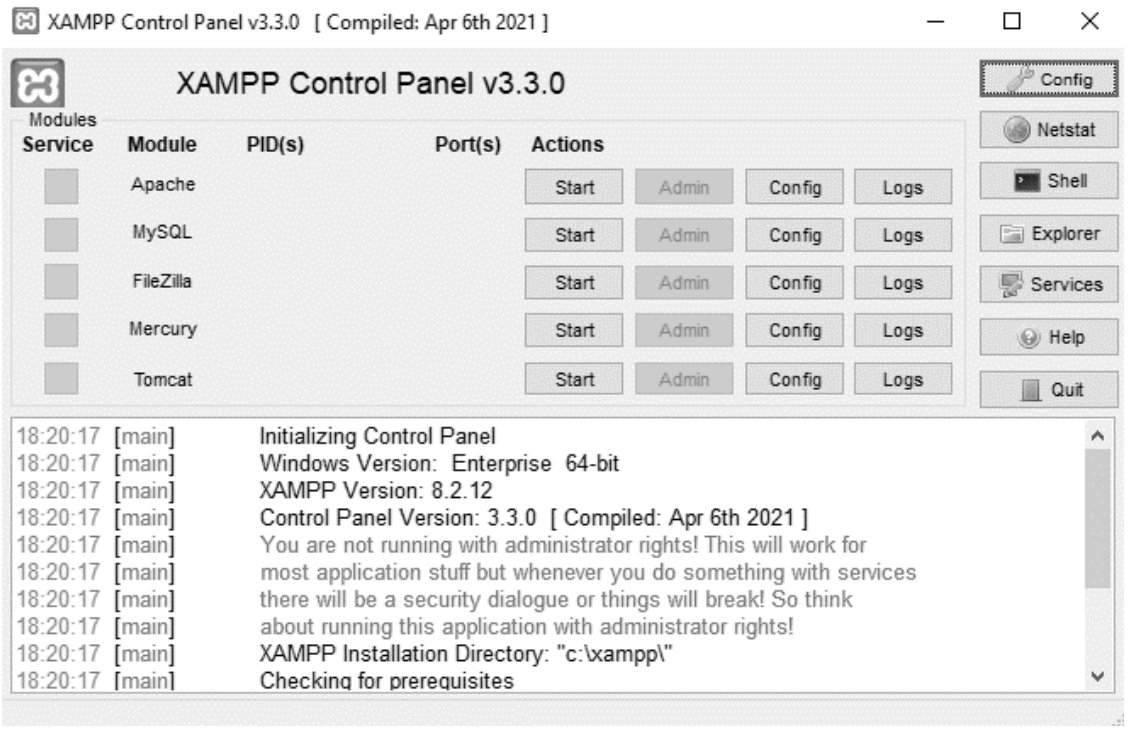


Рисунок 3.21 — Серверне середовище XAMPP

3.3.1 Використання phpMyAdmin для адміністрування бази даних та створення бази даних MariaDB

Для того, щоб реалізувати серверну частину веб-сервісу було використано базу даних MariaDB.

Адміністрування бази даних, створення таблиць та їх перегляд здійснюються через графічний інтерфейс phpMyAdmin, через якого була створена база даних medical_system_db, яка містить 5 таблиць:

— users;

— patients;

— appointments;

— treatments;

— medications.

SQL-команда створення таблиці users показано в лістингу 3.8.

Інші таблиці в процесі були створені за аналогічною логікою, в певних моментах змінюючи чи додаючи нові додаткові параметри та варіації записів в залежності від потрібної інформації.

Лістинг 3.7 — SQL-команда для створення таблиці users

```
CREATE TABLE `users` (
`id` int(11) NOT NULL AUTO_INCREMENT,
`username` varchar(50) NOT NULL,
`password_hash` varchar(255) NOT NULL,
`role` enum('admin','doctor','reception') NOT NULL,
PRIMARY KEY (`id`),
UNIQUE KEY `username` (`username`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
```

Таблиця patients містить дані про пацієнтів: ПІБ, контактну інформацію та дату народження.

Таблиця appointments використовується для записів на прийоми з даними клієнта, лікаря, дати, часу і статусу прийому, який автоматично записується і змінюється.

Таблиця treatments призначена для зберігання інформації про лікування клієнта, медикаменти, дозування та інструкції.

Таблиця medications містить перелік доступних препаратів і їх кількість.

Графічне зображення перелічених таблиць в phpMyAdmin показано на рисунку 3.21.

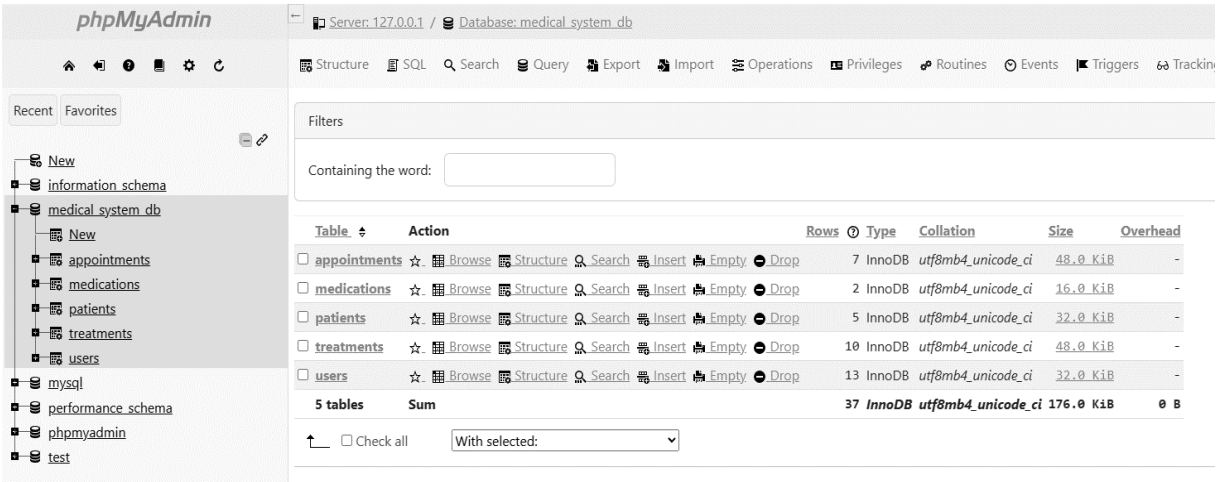


Рисунок 3.21 — phpMyAdmin з таблицями бази даних

В таблицю users було додано записи, які мають різний логін, пароль та роль, яка суттєво впливає на доступний функціонал.

Таблицю користувачів медичного інформаційного веб-сервісу показано в phpMyAdmin на рисунку 3.22.

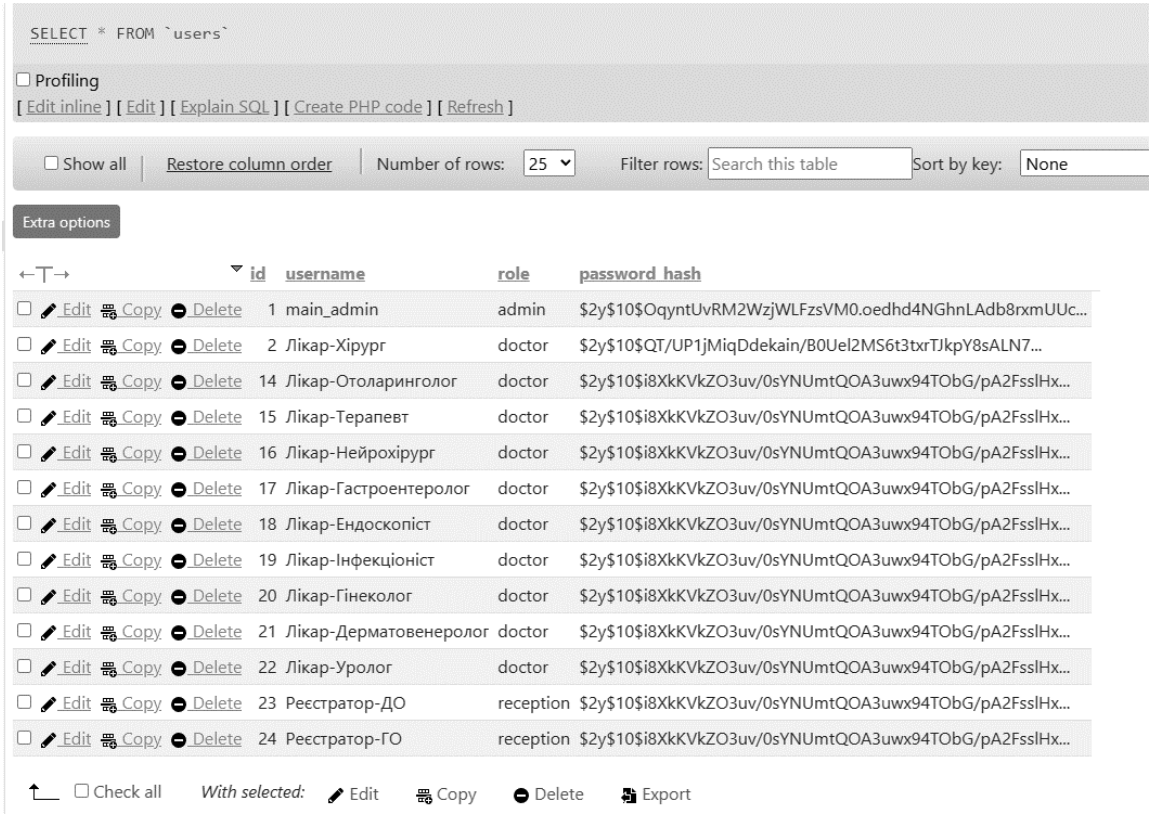


Рисунок 3.22 — Записи таблиці users в phpMyAdmin

3.3.2 Розробка сайту з використанням PHP і бази даних MariaDB

Реалізація веб-сервісу медичної інформаційної системи відбувається мовою PHP з використанням бази даних MariaDB. Взаємодія з базою даних відбувається за об’єктно-орієнтованим підходом з використанням PDO (PHP Data Objects), що забезпечує безпечний доступ до даних, і це також запобігає SQL-ін’єкціям та підвищує надійну роботу веб-сервісу.

Перша сторінка веб-сервісу — сторінка авторизації, з якого відбувається вхід у веб-сервіс медичної інформаційної системи. В ній відбувається перевірка даних, введених користувачем та після успішної ідентифікації в базі даних — надання доступу до веб-сервісу. Було створено PHP скрипт, який отримує логін і пароль з форми, виконує пошук подібних значень, і якщо дані співпадають, то створюється сесія для збереження авторизованого стану користувача.

Додавання нових користувачів цієї системи відбувається шляхом додавання нових записів вручну в базу даних MariaDB через консоль або через phpMyAdmin.

На рисунку 3.23 показано головну сторінку веб-сервісу.

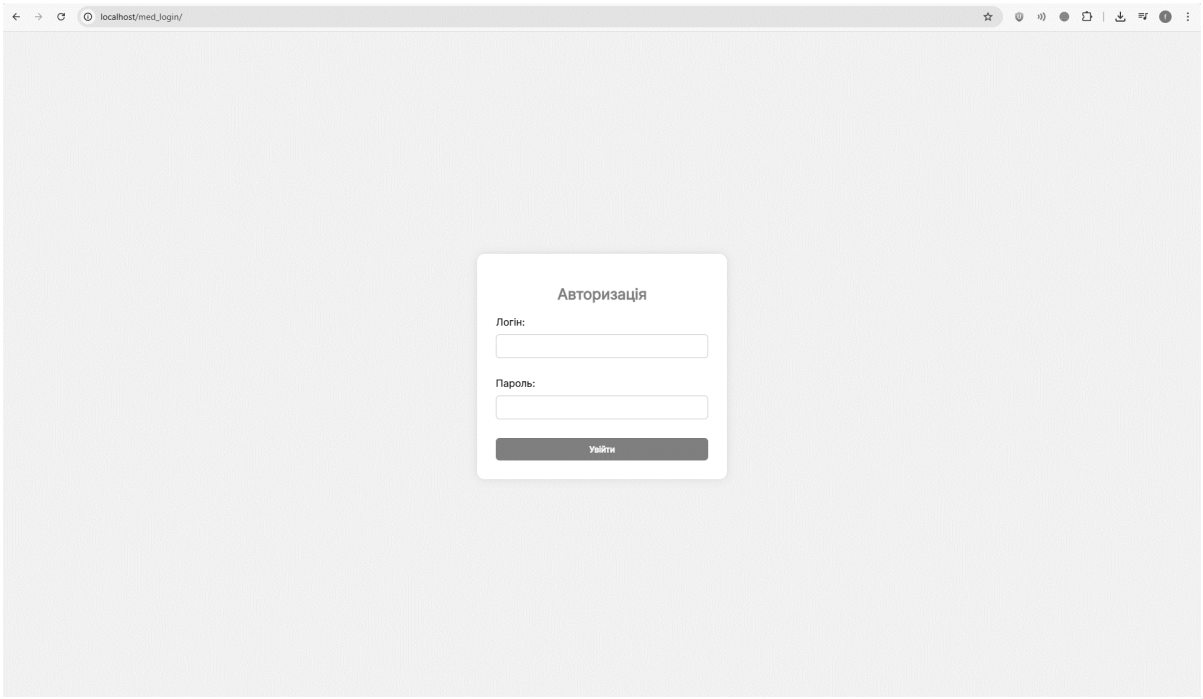


Рисунок 3.23 — Головна сторінка сайту

Код авторизації користувачів з базою даних показаний на лістингу 3.8.

Лістинг 3.8 — Код з SQL-запитами до авторизації користувача

```
if ($username && $password) {
$stmt = $pdo->prepare("SELECT * FROM users WHERE username = ?");
$stmt->execute([$username]);
$user = $stmt->fetch();
if ($user && password_verify($password, $user['password_hash'])) {
$_SESSION['username'] = $user['username'];  $_SESSION['role']
=$user['role'];
header('Location: ../dashboard.php');
```

Після успішної авторизації нас переводять на функціональну сторінку dashboard.php, де знаходиться основний функціонал веб-сервісу. На даний момент я зайшов у веб-сервіс на сторінку main_admin, яка знаходиться під роллю admin, якій доступний весь функціонал сервісу.

Також існує роль doctor, якій доступний тільки графік прийомів, список препаратів та призначення лікування. Графік прийомів доступний тільки обмежено, редагування статусу прийому заборонене, а закриття прийому відбувається в разі успішного призначення лікування клієнту.

Останньою роллю буде reception, якій доступна тільки зміна і додавання прийомів та додавання нових клієнтів.

Функціональна сторінка веб-ресурсу зображена на рисунку 3.34.

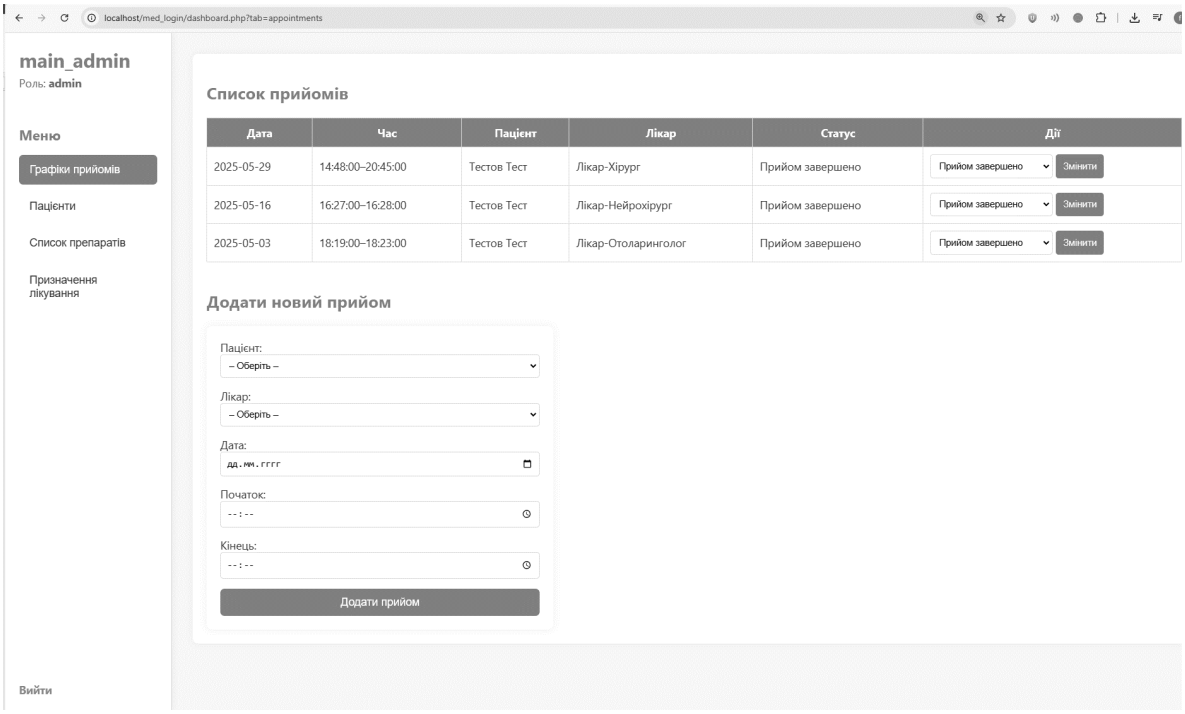


Рисунок 3.34 — Функціональна сторінка веб-сервісу

Функціонал веб-сервісу під ролями doctor та reception зображено на рисунку 3.35 та на рисунку 3.36.

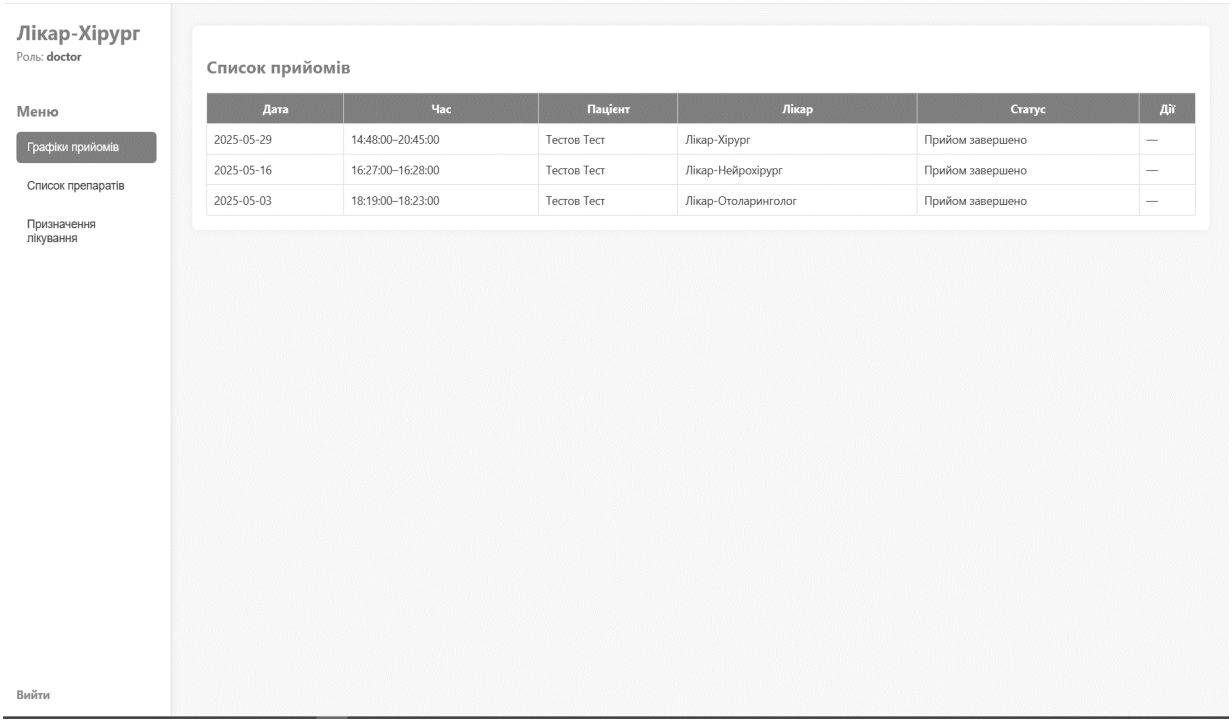


Рисунок 3.35 — Веб-сервіс під роллю doctor

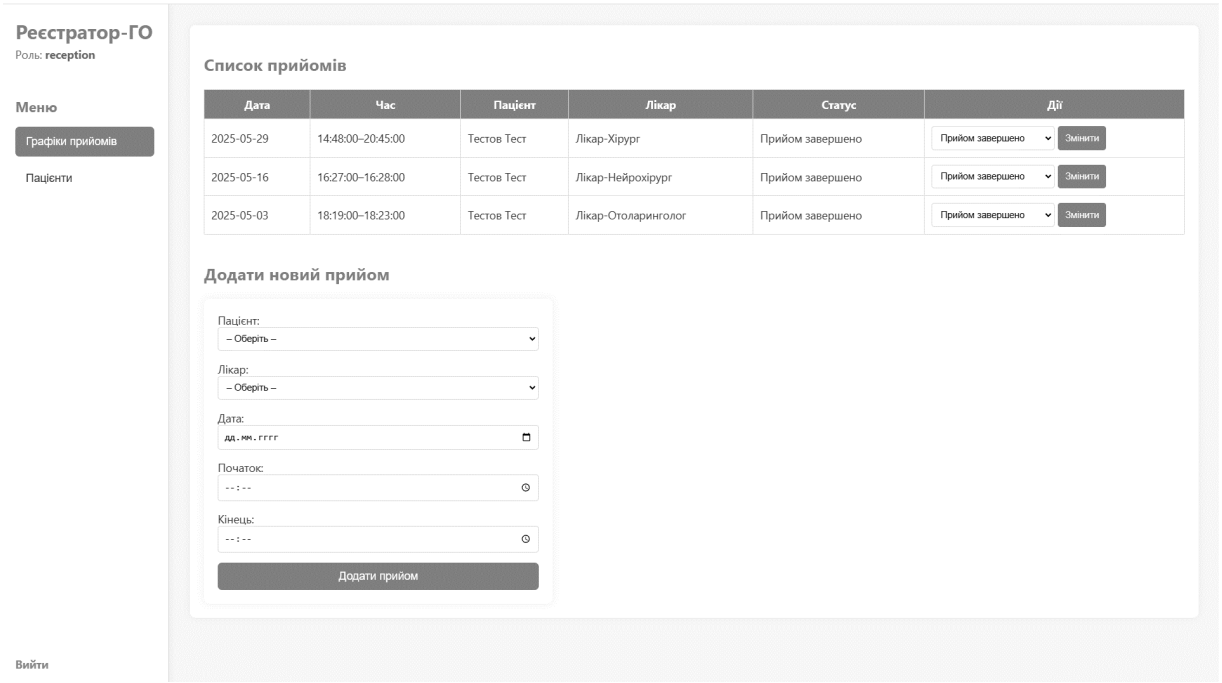


Рисунок 3.36 — Веб-сервіс під роллю reception

Для додавання нових пацієнтів до бази даних було на сторінці реалізовано форму, яка приймає обов’язкові поля ім’я та прізвище, і може не приймати дані в додаткові поля по-батькові, email та дата народження.

Код додавання пацієнта до бази даних показано в лістингу 3.9.

Лістинг 3.9 — Код з SQL-запитами до додавання пацієнта

```
$added = false;
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['add_patient'])) {
$first_name  = trim($_POST['first_name']  ?? '');
$last_name   = trim($_POST['last_name']   ?? '');
$middle_name = trim($_POST['middle_name'] ?? '');
$phone       = trim($_POST['phone']       ?? '');
$email       = trim($_POST['email']       ?? '');
$birth_date  = $_POST['birth_date'] ?? null;
if ($first_name !== '' && $last_name !== '') {
$stmt = $pdo->prepare("
INSERT INTO patients
(first_name, last_name, middle_name, phone, email, birth_date)
VALUES (?, ?, ?, ?, ?, ?)");
$stmt->execute([ $first_name, $last_name,
$middle_name !== '' ? $middle_name : null,
$phone       !== '' ? $phone       : null,
$email       !== '' ? $email       : null,
$birth_date  !== '' ? $birth_date  : null, ]);
$add_message = "Пацієнта успішно додано!";
$added = true;
} else {
$add_message = "Будь ласка, заповніть обов'язкові поля."; }
```

Форма з додаванням пацієнтів зображена на рисунку 3.37.

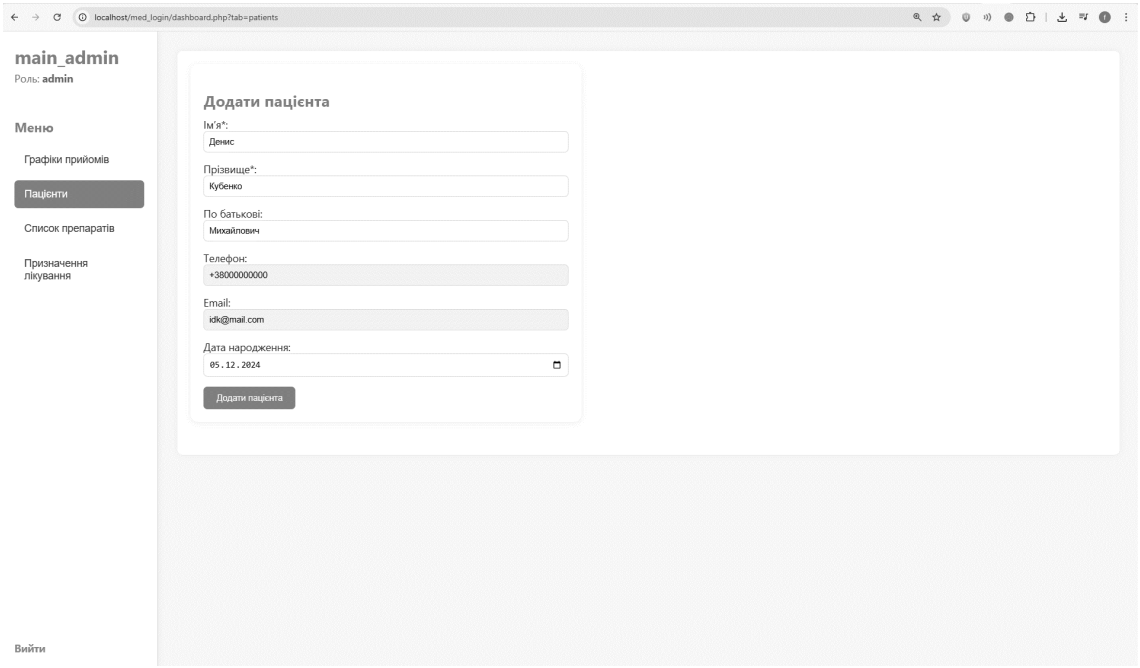


Рисунок 3.37 — Форма з додаванням пацієнта

Для додавання препаратів було реалізовано сторінку з таблицею із назвами, кількістю і діями до редагування та видалення препаратів. Запис до бази даних аналогічний до запису додавання пацієнта. Реалізація таблиці зображена на рисунку 3.38.

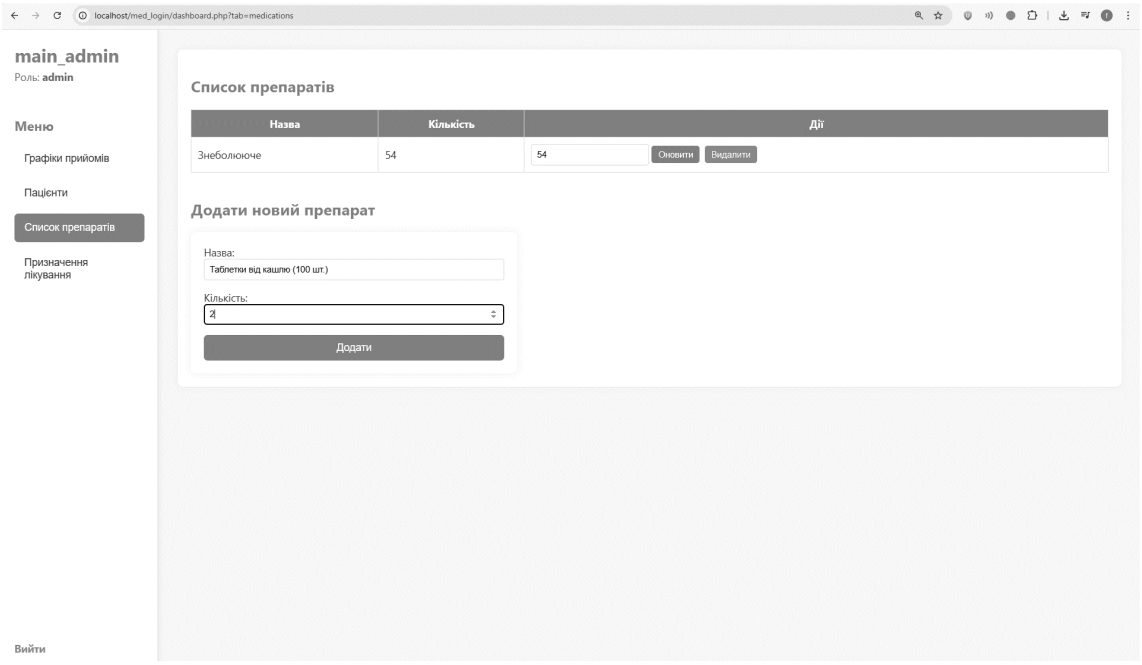


Рисунок 3.38 — Список препаратів і його можливості

Також було реалізовано сторінку з формою і таблицею, де є можливість обрати поточний прийом, обрати препарат і прописати його дозування, також потрібно написати інструкції до лікування. Код реалізації форми і таблиці призначення лікування клієнта показано в лістингу 3.10.

Лістинг 3.10 — Код з SQL-запитами до форми призначення лікування

```
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['add_treatment']))
{ $appointment_id = intval($_POST['appointment_id'] ?? 0);
$medication_id  = intval($_POST['medication_id'] ?? 0);
$dosage         = trim($_POST['dosage'] ?? '');
$instructions   = trim($_POST['instructions'] ?? '');
if ($appointment_id > 0 && $medication_id > 0 && $dosage !== '' &&
$instructions !== '') {
$stmt = $pdo->prepare("INSERT INTO treatments (appointment_id,
medication_id, dosage, instructions) VALUES (?, ?, ?, ?)");
$stmt->execute([$appointment_id, $medication_id, $dosage, $instructions]);
$stmt2 = $pdo->prepare("UPDATE appointments SET status = 'Прийом завершено'
WHERE id = ?");
$stmt2->execute([$appointment_id]);
$success = "Призначення збережено! Статус прийому оновлено на 'Прийом
завершено'.";
$added = true;
} else {
$error = "Будь ласка, заповніть усі поля.";}
```

Сторінка з реалізацією форми і таблиці з призначеним лікуванням показана на рисунку 3.39.

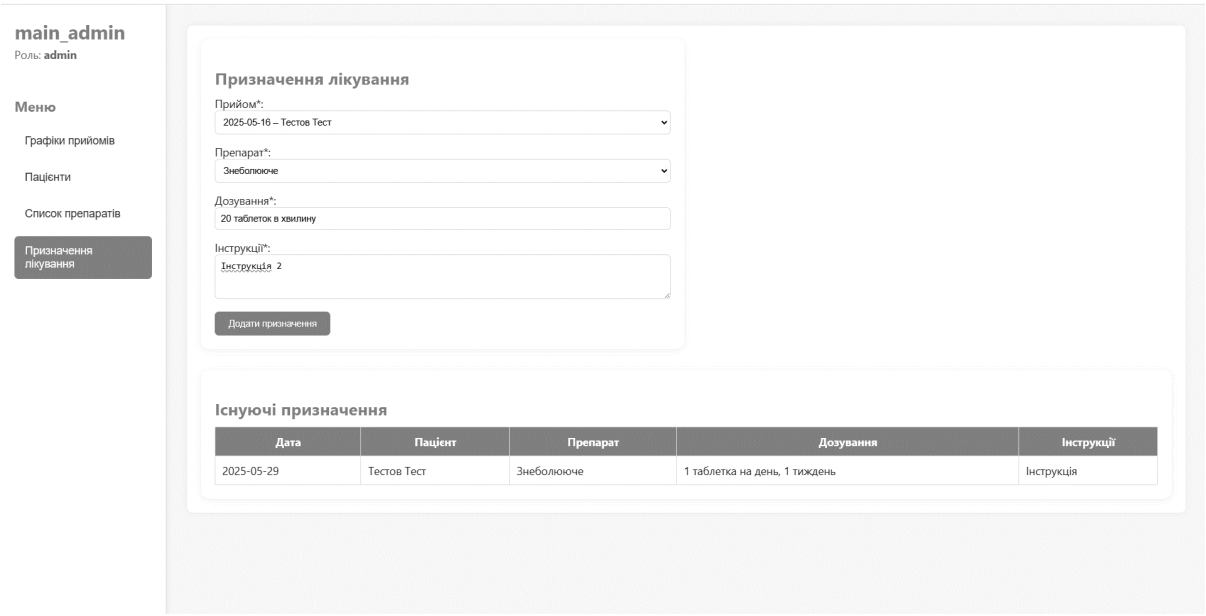


Рисунок 3.39 — Форма і таблиця з призначеними лікуваннями

ВИСНОВКИ

У результаті виконання бакалаврської кваліфікаційної роботи було спроєктовано, реалізовано та протестовано комп’ютерну мережу компанії з головним та віддаленими офісами на основі сучасних технологій маршрутизації, безпеки та адміністрування.

Розроблений проєкт враховує актуальні вимоги до захищеності, масштабованості та ефективного управління інформаційними потоками в умовах географічно розподіленої інфраструктури.

В процесі реалізації були досягнуті поставлені завдання:

— проведено аналіз архітектурних рішень для побудови розподілених комп’ютерних мереж;

— спроєктовано топологію мережі з чітким розподілом ролей між головним офісом, філіями та серверною зоною;

— впроваджено динамічну маршрутизацію OSPF, що забезпечила ефективне та адаптивне управління маршрутами між вузлами;

— реалізовано VPN-з’єднання на базі IPSec для захищеної передачі даних між офісами;

— налаштовано централізовану видачу IP-адрес через DHCP Relay, а також фільтрацію доступу до обладнання за допомогою ACL;

— забезпечено захищене адміністрування мережі через SSH-доступ до маршрутизаторів;

— реалізовано DNS і HTTP-сервери для підтримки внутрішньої інфраструктури;

— розгорнуто сервер бази даних MariaDB і створено веб-додаток на PHP, який забезпечує взаємодію з базою даних для зберігання та обробки інформації;

— протестовано всі компоненти мережі, що підтвердило коректність налаштувань, стабільність з’єднань і відповідність системи поставленим вимогам.

Запропонована мережа дозволяє забезпечити:

— безпечний та зашифрований обмін даними між офісами;

— централізоване управління сервісами та ресурсами;

— гнучке адміністрування з мінімальними витратами часу;

— масштабованість мережі для подальшого розширення бізнесу.

Новизна проєкту полягає в комплексному використанні протоколів і технологій, що дозволяють забезпечити повноцінне функціонування мережі з віддаленими офісами на високому рівні безпеки та надійності. Практична цінність полягає в можливості безпосереднього впровадження такої інфраструктури у реальному бізнес-середовищі для оптимізації інформаційних потоків та підвищення ефективності організаційної діяльності.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Технології і методи з’єднання віддалених мереж. / Д.М. Кубенко, С. М. Захарченко // Матеріали конференції LIV Всеукраїнської науково-технічної конференції факультету інформаційних технологій та комп’ютерної інженерії. Вінниця 2025 р. [Електронний ресурс]. Режим доступу: https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2025/paper/view/24510

2. Common Types of Network Devices and Their Functions. Режим доступу: https://blog.netwrix.com/network-devices-explained

3. What Are the 4 Types of Network Cables?. Режим доступу: https://www.tevelec.com/what-are-the-4-types-of-network-cables/

4. Twisted pair — Wikipedia. Режим доступу: https://en.m.wikipedia.org/wiki/Twisted_pair

5. Twisted Pair Cable: Types, Shielding, and Applications. Режим доступу: https://www.gcabling.com/what-is-twisted-pair-cable-and-types/

6. Fiber-optic cable — Wikipedia. Режим доступу: https://en.m.wikipedia.org/wiki/Fiber-optic_cable

7. Specifications For Fiber Optic Networks. Режим доступу: https://www.thefoa.org/tech/Linkspec.htm

8. Types of Network topology. Режим доступу: https://www.geeksforgeeks.org/types-of-network-topology/

9. Comer D. E. Computer Networks and Internets. 6th ed. 2020. 672 p.

10. Топологія мереж. Режим доступу: https://uk.m.wikipedia.org/wiki/%D0%A2%D0%BE%D0%BF%D0%BE%D0%BB%D0%BE%D0%B3%D1%96%D1%8F_%D0%BC%D0%B5%D1%80%D0%B5%D0%B6

11. Types of computer network. Режим доступу: https://www.geeksforgeeks.org/types-of-computer-networks/

12. Комп'ютерні мережі: підручник / Азаров О. Д., Захарченко С. М., Кадук О. В., Орлова М. М., Тарасенко В. П. - Вінниця: ВНТУ, 2020. 378 с

13. Коробейнікова Т. І., Захарченко С. М. Комп'ютерні мережі. Львів:

Видавництво Львівської політехніки, 2022. 228 с.

14. What is CIDR. Режим доступу: https://aws.amazon.com/what-is/cidr/

15. Dynamic Routing Protocols. Cisco Community. Режим доступу: https://community.cisco.com/t5/networking-knowledge-base/dynamic-routing-protocols-ospf-eigrp-ripv2-is-is-bgp/ta-p/4511577

16. Open Shortest Path First (OSPF) Protocol States. Режим доступу: https://www.geeksforgeeks.org/open-shortest-path-first-ospf-protocol-states/

17. Dynamic Host Configuration Protocol – Wikipedia. Режим доступу: https://en.m.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol

18. MPLS | What Is Multiprotocol Label Switching. Режим доступу: https://www.paloaltonetworks.com/cyberpedia/mpls-what-is-multiprotocol-label-switching

19. IPSec (Internet Protocol Security). Режим доступу: https://networklessons.com/security/ipsec-internet-protocol-security

20. MariaDB — Вікіпедія. Режим доступу: https://uk.m.wikipedia.org/wiki/MariaDB

ДОДАТОК А

Технічне завдання

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп’ютерної інженерії
Кафедра обчислювальної техніки

ЗАТВЕРДЖУЮ
Завідувач кафедри ОТ
проф., д.т.н.. Азаров О.Д.
21 березня 2025 р.

ТЕХНІЧНЕ ЗАВДАННЯ
на виконання бакалаврської кваліфікаційної роботи
«Комп’ютерна мережа компанії з віддаленими офісами»

08-54.БКР.030.00.000 ПЗ

Науковий керівник: к.т.н., проф. каф. ОТ
_____Захарченко С. М.

Студент групи 2СП-21б
_____Кубенко Д.М.

м. Вінниця — 2025

1 Підстава для виконання бакалаврської кваліфікаційної роботи (БКР)

1.1 Актуальність проєкту полягає в необхідності розробки безпечної, масштабованої та ефективної комп’ютерної мережі для компанії з віддаленими офісами. Сучасні організації мають потребу в централізованому управлінні даними, безпечному міжофісному зв’язку, динамічній маршрутизації та контролі доступу до ресурсів. З огляду на збільшення обсягів переданої інформації та складність мережевих операцій, впровадження IPSec, OSPF, DHCP Relay, ACL і SSH є ключовими для забезпечення безпеки та надійності корпоративної інфраструктури.

1.2 Завданням проєкту є побудова розподіленої корпоративної мережі з використанням динамічної маршрутизації (OSPF), захищених VPN-з’єднань (IPSec), механізмів автоматичної адресації (DHCP із Relay Agent), трансляції адрес (NAT), інструментів безпеки (ACL, SSH), а також реалізація вебзастосунку, що взаємодіє з централізованою базою даних (MariaDB).

1.3 Наказ про затвердження теми бакалаврської дипломної роботи від 21 березня 2025 р. №97.

2 Мета і призначення БКР

2.1 Метою роботи є проєктування, побудова та налаштування безпечної та ефективної розподіленої комп’ютерної мережі для компанії з географічно віддаленими офісами із використанням сучасних технологій маршрутизації, захищених каналів зв’язку та централізованих мережевих сервісів.

2.2 Призначення розробки полягає у створенні технічно обґрунтованого рішення з побудови корпоративної мережі, яка дозволяє об’єднати офіси компанії в єдину інформаційну систему з гнучким управлінням, високою безпекою та можливістю масштабування.

3 Вихідні дані для виконання БКР

3.1 Бізнес-модель компанії з центральним офісом та кількома віддаленими

філіями, які мають бути об’єднані захищеними VPN-з’єднаннями через публічну IP-мережу.

3.2 Аналіз та вибір сучасних мережевих технологій: протокол OSPF для динамічної маршрутизації, IPSec для захищених VPN, ACL для контролю доступу, SSH для адміністрування, DHCP Relay для автоматичної IP-конфігурації в підмережах, DNS для іменування, та HTTP/HTTPS для доступу до вебінтерфейсу.

3.3 Підбір маршрутизаторів, комутаторів, серверного обладнання, інсталяція ОС, сервер MariaDB, засоби адміністрування (phpMyAdmin, SSH).

3.4 Проєктування мережевої топології з врахуванням адресного планування, реалізація тунелів IPSec, реалізація ACL-фільтрації, маршрутизація через OSPF.

3.5 Налаштування централізованого DHCP-сервера в головному офісі з Relay Agent для віддалених мереж..

3.6 Розробка вебдодатку на PHP, який взаємодіє з базою даних MariaDB, для доступу співробітників до інформації згідно з ролями.

4 Вимоги до виконання БКР

Проєкт має передбачати:

— побудову мережі з динамічною маршрутизацією OSPF;

— налаштування VPN-з’єднання між офісами з використанням IPSec;

— використання DHCP Relay для автоматичної видачі IP-адрес у всіх офісах;

— впровадження ACL для обмеження доступу до ресурсів та SSH для безпечного адміністрування;

— інсталяцію DNS і HTTP серверів;

— створення бази даних MariaDB та вебдодатку на PHP з інтеграцією;

— тестування всіх сервісів, включно з VPN, DHCP, доступом до БД, авторизацією користувачів.

5 Етапи БКР та очікувані результати

Етапи розробки БКР та очікувані результати наведено у Таблиці А.1

Таблиця А.1 — Етапи БКР

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		Початок	Кінець	
1	Постановка задачі	21.03.25	21.03.25	Розділ 1
2	Аналіз технологій для створення комп'ютерних мереж	21.03.25	30.03.25	Розділ 1
3	Аналіз та обґрунтування вибору технологій	21.03.25	30.04.25	Розділ 2
4	Реалізація комп'ютерної мережі	31.03.25	13.04.25	Розділ 3
5	Тестування комп'ютерної мережі	14.04.25	27.04.25	Розділ 4
6	Підготовка тезових матеріалів для публікації	21.04.25	28.04.25	Тези
7	Оформлення пояснювальної записки та демонстраційної презентації для доповіді	29.04.25	12.05.25	ПЗ, презентація
8	Підготовка супровідної документації, перевірка відповідності нормам та проходження тесту на плагіат	13.05.25	13.05.25	ПЗ, презентація

6 Матеріали, що подаються до захисту БКР

До захисту подаються: пояснювальна записка БКР, ілюстративні матеріали, протокол попереднього захисту БКР на кафедрі, відгук наукового керівника, анотації до БКР українською та іноземною мовами, довідка про відповідність оформлення БКР чинним вимогам.

7 Порядок контролю виконання та захисту БКР

Виконання етапів графічної та розрахункової документації БКР контролюється науковим керівником к.т.н., проф. каф. ОТ Захарченком С. М. згідно зі встановленими термінами. Захист БКР відбувається на засіданні Екзаменаційної комісії, затвердженої наказом ректора ВНТУ.

8 Вимоги до оформлювання та порядок виконання БКР

8.1 При оформлюванні БКР використовуються:

8.1.1 ДСТУ 3008 : 2015 «Звіти в сфері науки і техніки. Структура та правила оформлювання»;

8.1.2 ДСТУ 8302 : 2015 «Бібліографічні посилання. Загальні положення та правила складання»;

8.1.3 методичні вказівки до виконання бакалаврських кваліфікаційних робіт для студентів спеціальності 123 «Комп'ютерна інженерія» (освітня програма «Системне програмування») — кафедра обчислювальної техніки, ВНТУ 2023.

8.2 Порядок виконання БКР викладено в «Положення про кваліфікаційні роботи на першому (бакалаврському) рівні вищої освіти СУЯ ВНТУ-03.02.02-П.001.01:21»

ДОДАТОК Б

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: ____Комп'ютерна мережа компанії з віддаленими офісами_____

Тип роботи: _____бакалаврська кваліфікаційна робота_____
(бакалаврська кваліфікаційна робота / магістерська кваліфікаційна робота)

Підрозділ _____кафедра обчислювальної техніки_____
(кафедра, факультет, навчальна група)

Коефіцієнт подібності текстових запозичень, виявлених у роботі системою StrikePlagiarism ___4,5____%

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

✓ Запозичення, виявлені у роботі, є законними і не містять ознак плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту

☐ У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.

☐ У роботі виявлено ознаки плагіату та/або текстових маніпуляцій як спроб укриття плагіату, фабрикації, фальсифікації, що суперечить вимогам законодавства та нормам академічної доброчесності. Робота до захисту не приймається.

Експертна комісія:

_____Азаров О.Д., зав. каф. ОТ _____ _____
(прізвище, ініціали, посада) (підпис)

_____Тарновський М.Г., доц. каф ОТ_____ _____
(прізвище, ініціали, посада) (підпис)

Особа, відповідальна за перевірку _____ ___Захарченко С.М.____
(підпис) (прізвище, ініціали)

З висновком експертної комісії ознайомлений(-на)

Керівник _____ _____Захарченко С.М._____
(підпис) (прізвище, ініціали, посада)

Здобувач _____ _____Кубенко Д.М._____
(підпис) (прізвище, ініціали)

ДОДАТОК В

ІЛЮСТРАТИВНА ЧАСТИНА

КОМП’ЮТЕРНА МЕРЕЖА КОМПАНІЇ З ВІДДАЛЕНИМИ ОФІСАМИ

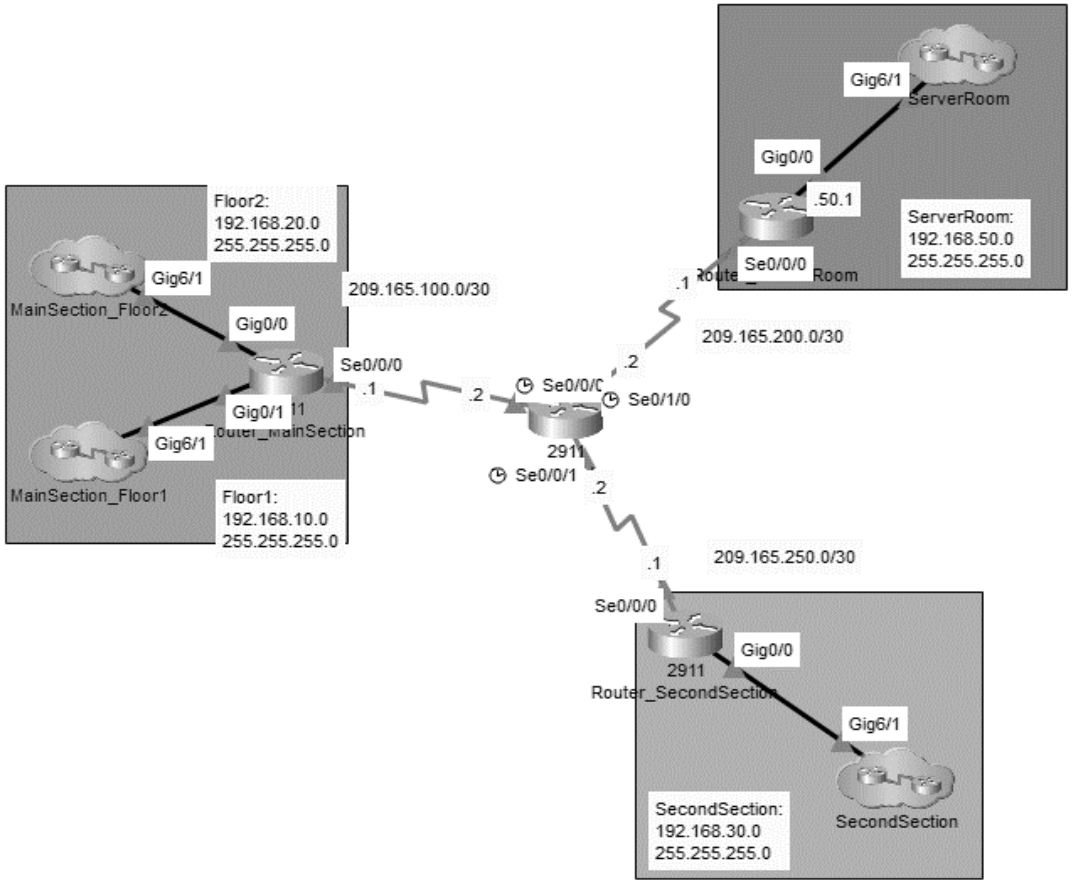


Рисунок В.1 – Логічна топологія мережі компанії з віддаленими офісами

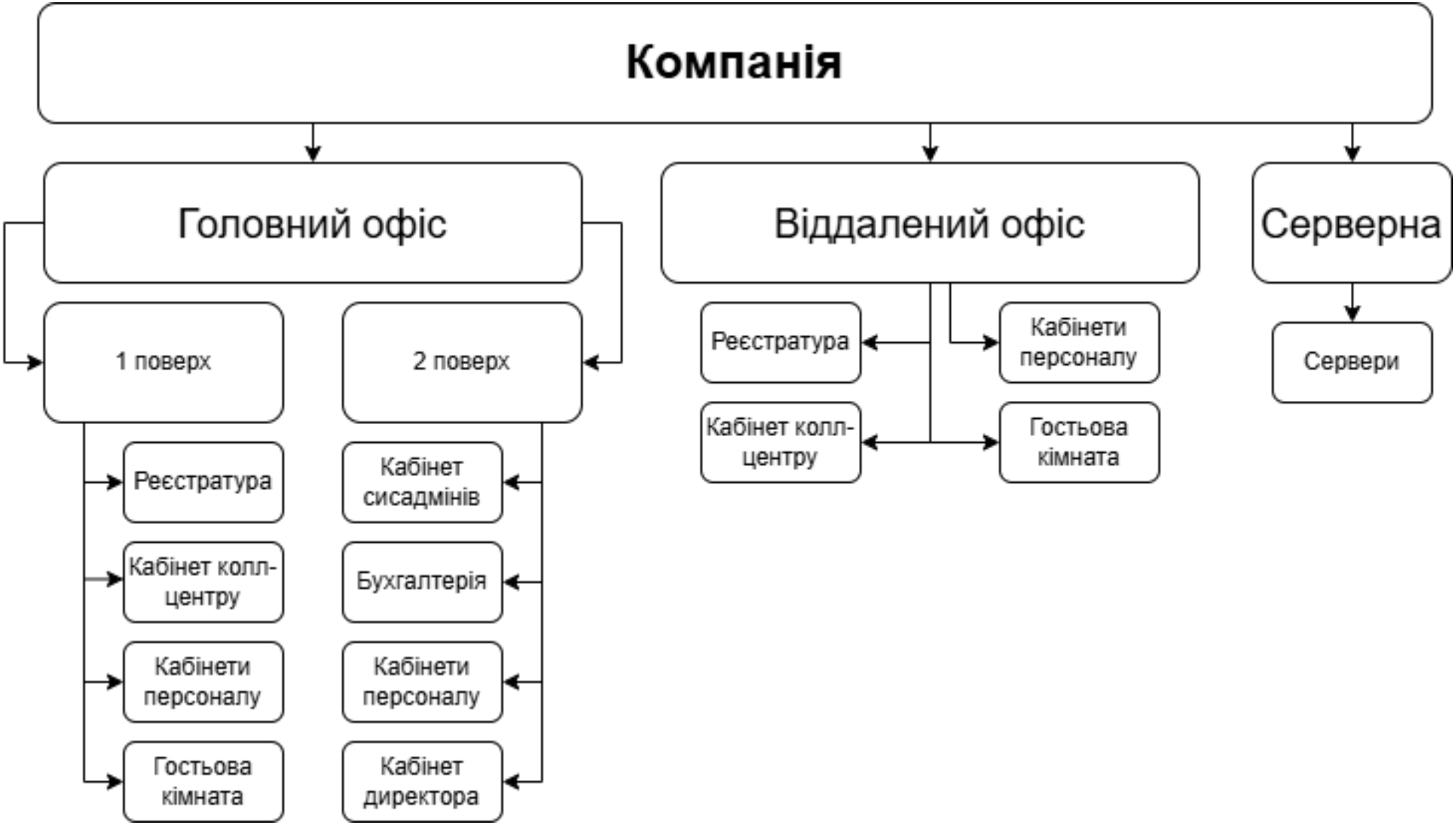


Рисунок В.2 — Структурна схема мережі компанії з віддаленими офісами

ДОДАТОК Г

Лістинги конфігураційних файлів

Лістинг Г.1 – Конфігураційний файл маршрутизатора головного офісу

```
Router_MainSection#sh run
Building configuration...
Current configuration : 1903 bytes
version 15.1
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
hostname Router_MainSection
no ip cef
no ipv6 cef
license udi pid CISCO2911/K9 sn FTX15245IU8-
license boot module c2900 technology-package securityk9
crypto isakmp policy 1
encr aes 256
authentication pre-share
group 5
crypto isakmp key denus111 address 209.165.200.1
crypto isakmp key denus111 address 209.165.250.1
crypto ipsec transform-set VPN_Set esp-aes esp-sha-hmac
crypto map VPN_Set 1 ipsec-isakmp
description vpn conns site second section
set peer 209.165.250.1
set transform-set VPN_Set
match address 100
crypto map VPN_Set 20 ipsec-isakmp
description vpn tunnel to server room
set peer 209.165.200.1
set transform-set VPN_Set
match address 100
spanning-tree mode pvst
interface GigabitEthernet0/0
ip address 192.168.20.1 255.255.255.0
ip helper-address 192.168.50.10
duplex auto
speed auto
interface GigabitEthernet0/1
ip address 192.168.10.1 255.255.255.0
ip helper-address 192.168.50.10
duplex auto
speed auto
interface GigabitEthernet0/2
no ip address
duplex auto
speed auto
shutdown
interface Serial0/0/0
ip address 209.165.100.1 255.255.255.252
crypto map VPN_Set
interface Serial0/0/1
```

```
no ip address
clock rate 2000000
shutdown
interface Vlan1
no ip address
shutdown
router ospf 1
log-adjacency-changes
network 192.168.20.0 0.0.0.255 area 0
network 192.168.10.0 0.0.0.255 area 0
network 209.165.100.0 0.0.0.3 area 0
ip classless
ip flow-export version 9
access-list 100 permit ip 192.168.10.0 0.0.0.255 192.168.30.0 0.0.0.255
access-list 100 permit ip 192.168.20.0 0.0.0.255 192.168.30.0 0.0.0.255
access-list 100 permit ip 192.168.10.0 0.0.0.255 192.168.50.0 0.0.0.255
access-list 100 permit ip 192.168.20.0 0.0.0.255 192.168.50.0 0.0.0.255
access-list 50 permit host 192.168.20.10
access-list 50 permit host 192.168.20.11
line con 0
line aux 0
line vty 0 4
login
end
```

Лістинг Г.2 – Конфігураційний файл маршрутизатора віддаленого офісу

```
Router_SecondSection#sh run
Building configuration...
Current configuration : 1778 bytes
version 15.1
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
hostname Router_SecondSection
no ip cef
no ipv6 cef
license udi pid CISCO2911/K9 sn FTX152471P6-
license boot module c2900 technology-package securityk9
crypto isakmp policy 1
encr aes 256
authentication pre-share
group 5
crypto isakmp key denus111 address 209.165.100.1
crypto isakmp key denus111 address 209.165.200.1
crypto ipsec transform-set VPN_Set esp-aes esp-sha-hmac
crypto map VPN_Set 10 ipsec-isakmp
description vpn conns site main section
set peer 209.165.100.1
set transform-set VPN_Set
match address 100
crypto map VPN_Set 20 ipsec-isakmp
description vpn conns site server room
set peer 209.165.200.1
set transform-set VPN_Set
```

```
match address 100
spanning-tree mode pvst
interface GigabitEthernet0/0
ip address 192.168.30.1 255.255.255.0
ip helper-address 192.168.50.10
duplex auto
speed auto
interface GigabitEthernet0/1
no ip address
duplex auto
speed auto
shutdown
interface GigabitEthernet0/2
no ip address
duplex auto
speed auto
shutdown
interface Serial0/0/0
ip address 209.165.250.1 255.255.255.252
crypto map VPN_Set
interface Serial0/0/1
no ip address
clock rate 2000000
shutdown
interface Vlan1
no ip address
shutdown
router ospf 1
log-adjacency-changes
network 209.165.250.0 0.0.0.3 area 0
network 192.168.30.0 0.0.0.255 area 0
ip classless
ip flow-export version 9
access-list 100 permit ip 192.168.30.0 0.0.0.255 192.168.10.0 0.0.0.255
access-list 100 permit ip 192.168.30.0 0.0.0.255 192.168.20.0 0.0.0.255
access-list 100 permit ip 192.168.30.0 0.0.0.255 192.168.50.0 0.0.0.255
line con 0
line aux 0
line vty 0 4
login
end
```

Лістинг Г.3 – Конфігураційний файл маршрутизатора серверної

```
Router_ServerRoom>en
Router#sh run
Building configuration…
Current configuration : 1792 bytes
version 15.1
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
hostname Router_ServerRoom
no ip cef
no ipv6 cef
```

```
license udi pid CISCO2911/K9 sn FTX1524H69K-
license boot module c2900 technology-package securityk9
crypto isakmp policy 1
encr aes 256
authentication pre-share
group 5
crypto isakmp key denus111 address 209.165.100.1
crypto isakmp key denus111 address 209.165.250.1
crypto ipsec transform-set VPN_Set esp-aes esp-sha-hmac
crypto map VPN_Set 20 ipsec-isakmp
description vpn tunnel to main sect router
set peer 209.165.100.1
set transform-set VPN_Set
match address 100
crypto map VPN_Set 30 ipsec-isakmp
description description vpn tunnel to second sect
set peer 209.165.250.1
set transform-set VPN_Set
match address 100
spanning-tree mode pvst
interface GigabitEthernet0/0
ip address 192.168.50.1 255.255.255.0
duplex auto
speed auto
interface GigabitEthernet0/1
no ip address
duplex auto
speed auto
shutdown
interface GigabitEthernet0/2
no ip address
duplex auto
speed auto
shutdown
interface Serial0/0/0
ip address 209.165.200.1 255.255.255.252
crypto map VPN_Set
interface Serial0/0/1
no ip address
clock rate 2000000
shutdown
interface Vlan1
no ip address
shutdown
router ospf 1
log-adjacency-changes
network 192.168.50.0 0.0.0.255 area 0
network 209.165.200.0 0.0.0.3 area 0
ip classless
ip flow-export version 9
access-list 100 permit ip 192.168.50.0 0.0.0.255 192.168.10.0 0.0.0.255
access-list 100 permit ip 192.168.50.0 0.0.0.255 192.168.20.0 0.0.0.255
access-list 100 permit ip 192.168.50.0 0.0.0.255 192.168.30.0 0.0.0.255
line con 0
line aux 0
line vty 0 4
```

Лістинг Г.4 – Конфігураційний файл маршрутизатора ISP

```
ISP#en
ISP#sh run
Building configuration...
Current configuration : 1169 bytes
version 15.1
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
hostname ISP
no ip cef
no ipv6 cef
license udi pid CISCO2911/K9 sn FTX1524EJU2-
spanning-tree mode pvst
interface GigabitEthernet0/0
no ip address
duplex auto
speed auto
shutdown
interface GigabitEthernet0/1
no ip address
duplex auto
speed auto
shutdown
interface GigabitEthernet0/2
no ip address
duplex auto
speed auto
shutdown
interface Serial0/0/0
ip address 209.165.100.2 255.255.255.252
clock rate 2000000
interface Serial0/0/1
ip address 209.165.250.2 255.255.255.252
clock rate 2000000
interface Serial0/1/0
ip address 209.165.200.2 255.255.255.252
clock rate 2000000
interface Serial0/1/1
no ip address
clock rate 2000000
shutdown
interface Vlan1
no ip address
shutdown
router ospf 1
log-adjacency-changes
network 209.165.100.0 0.0.0.3 area 0
network 209.165.200.0 0.0.0.3 area 0
network 209.165.250.0 0.0.0.3 area 0
ip classless
ip flow-export version 9
line con 0
line aux 0
```

```
line vty 0 4
login
end
```

Лістинг Г.5 – Файл index.php

```
<?php
session_start();
if (isset($_SESSION['username'])) {
    header('Location: dashboard.php');
    exit;
}
$error = $_GET['error'] ?? '';
?>
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <title>Вхід</title>
  <link href="https://fonts.googleapis.com/css2?family=Inter&display=swap"
rel="stylesheet">
</head>
<body>
  <div class="login-container">
    <h2>Авторизація</h2>
    <?php if ($error): ?>
      <div class="error"><?= htmlspecialchars($error) ?></div>
    <?php endif; ?>
    <form method="POST" action="actions/login.php">
      <label>Логін:</label><br>
      <input type="text" name="username" required><br><br>
      <label>Пароль:</label><br>
      <input type="password" name="password" required><br><br>
      <button type="submit">Увійти</button>
    </form>
  </div>
</body>
</html>
```

Лістинг Г.6 – Файл login.php

```
<?php
session_start();
require_once '../db.php';
$username = $_POST['username'] ?? '';
$password = $_POST['password'] ?? '';
if ($username && $password) {
    $stmt = $pdo->prepare("SELECT * FROM users WHERE username = ?");
    $stmt->execute([$username]);
    $user = $stmt->fetch();
    if ($user && password_verify($password, $user['password_hash'])) {
        $_SESSION['username'] = $user['username'];
        $_SESSION['role'] = $user['role'];
        header('Location: ../dashboard.php');
        exit;
```

```
    } else {
        header('Location: ../index.php?error=Невірний логін або пароль');
        exit;
    }
} else {
    header('Location: ../index.php?error=Заповніть всі поля');
    exit;
}
```

Лістинг Г.7 – Файл dashboard.php

```
<?php
session_start();
if (!isset($_SESSION['username'])) {
    header('Location: index.php');
    exit;
}
$username = $_SESSION['username'];
$role = $_SESSION['role'];
?>
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8" />
  <title>Дашборд</title>
</head>
<body>
<div class="dashboard">
  <aside class="sidebar">
    <div class="sidebar-top">
      <div class="user-info">
        <h1> <?= htmlspecialchars($username) ?></h3>
        <p>Роль: <b><?= htmlspecialchars($role) ?></b></p>
      </div>
      <h2>Меню</h2>
      <?php if (in_array($role, ['admin','reception','doctor'], true)): ?>
        <button data-tab="appointments">Графіки прийомів</button>
      <?php endif; ?>
      <?php if (in_array($role, ['admin','reception'], true)): ?>
        <button data-tab="patients">Пацієнти</button>
      <?php endif; ?>
      <?php if (in_array($role, ['admin','doctor'], true)): ?>
        <button data-tab="medications">Список препаратів</button>
        <button data-tab="treatments">Призначення лікування</button>
      <?php endif; ?>
    </div>
    <div>
      <a class="logout" href="actions/logout.php">Вийти</a>
    </div>
  </aside>
  <main class="main">
    <?php if (in_array($role, ['admin','reception','doctor'], true)): ?>
      <div id="appointments" class="tab-content">
        <?php include 'views/appointments.php'; ?>
      </div>
```

```
    <?php endif; ?>
    <?php if (in_array($role, ['admin','reception'], true)): ?>
      <div id="patients" class="tab-content">
        <?php include 'views/patients.php'; ?>
      </div>
    <?php endif; ?>
    <?php if (in_array($role, ['admin','doctor'], true)): ?>
      <div id="medications" class="tab-content">
        <?php include 'views/medications.php'; ?>
      </div>
      <div id="treatments" class="tab-content">
        <?php include 'views/treatments.php'; ?>
      </div>
    <?php endif; ?>
  </main>
</div>
<script>
  const buttons = document.querySelectorAll('.sidebar button');
  const tabs = document.querySelectorAll('.tab-content');
  function showTab(id) {
    tabs.forEach(t => t.classList.toggle('active', t.id === id));
    buttons.forEach(b => b.classList.toggle('active', b.getAttribute('data-
tab') === id));
  }
  const params = new URLSearchParams(window.location.search);
  const tab = params.get('tab');
  if (tab && document.getElementById(tab)) {
    showTab(tab);
  } else if (buttons.length) {
    showTab(buttons[0].getAttribute('data-tab'));
  }
  buttons.forEach(btn => {
    btn.addEventListener('click', () => {
      const t = btn.getAttribute('data-tab');
      showTab(t);
      history.replaceState(null, '', '?tab=' + t);
    });
  });
</script>
</body>
</html>
```

Лістинг Г.8 – Файл db.php

```
<?php
$host = 'localhost';
$db   = 'medical_system_db';
$user = 'root';
$pass = '';
$charset = 'utf8mb4';
$dsn = "mysql:host=$host;dbname=$db;charset=$charset";
$options = [
    PDO::ATTR_ERRMODE            => PDO::ERRMODE_EXCEPTION,
    PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
```

```
];
try {
    $pdo = new PDO($dsn, $user, $pass, $options);
} catch (\PDOException $e) {
    die('Помилка підключення до бази: ' . $e->getMessage());
}
```

Лістинг Г.9– Файл appointments.php

```
<?php
require_once __DIR__ . '/../db.php';
?>
<?php
$error   = '';
$success = '';
if ($_SERVER['REQUEST_METHOD'] === 'POST' &&
isset($_POST['add_appointment'])) {
    if ($_SESSION['role'] === 'admin' || $_SESSION['role'] === 'reception')
{
        $patient_id       = intval($_POST['patient_id'] ?? 0);
        $doctor_id        = intval($_POST['doctor_id']  ?? 0);
        $appointment_date = $_POST['appointment_date'] ?? '';
        $time_start       = $_POST['time_start'] ?? '';
        $time_end         = $_POST['time_end'] ?? '';
        if ($patient_id > 0 && $doctor_id > 0 && $appointment_date &&
$time_start && $time_end) {
            $stmt = $pdo->prepare("INSERT INTO appointments
                (patient_id, doctor_id, appointment_date, time_start,
time_end, status)
                VALUES (?, ?, ?, ?, ?, ?)");
            $stmt->execute([$patient_id, $doctor_id, $appointment_date,
$time_start, $time_end, 'Заплановано']);
            $success = "Прийом успішно додано!";
        } else {
            $error = "Будь ласка, заповніть всі поля правильно.";
        }
    } else {
        $error = "У вас немає прав додавати прийоми.";
    }
}
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['update_status']))
{
    if ($_SESSION['role'] === 'admin' || $_SESSION['role'] === 'reception')
{
        $appt_id = intval($_POST['appointment_id'] ?? 0);
        $new_status = $_POST['status'] ?? '';
        $allowed = ['Заплановано','Прийшов','Запізнюється','Прийом
розпочато','Прийом завершено'];
        if ($appt_id > 0 && in_array($new_status, $allowed, true)) {
            $pdo->prepare("UPDATE appointments SET status = ? WHERE id = ?")
                ->execute([$new_status, $appt_id]);
            $success = "Статус оновлено.";
        } else {
```

```
            $error = "Невірний статус чи ID.";
        }
    } else {
        $error = "У вас немає прав змінювати статус.";
    }
}
$roles_with_access = ['admin', 'reception', 'doctor'];
$appointments = [];
if (in_array($_SESSION['role'], $roles_with_access, true)) {
    $appointments = $pdo->query("
        SELECT a.*, p.last_name, p.first_name, u.username AS doctor
        FROM appointments a
        JOIN patients p ON a.patient_id = p.id
        JOIN users u    ON a.doctor_id   = u.id
        ORDER BY a.appointment_date DESC, a.time_start DESC
    ")->fetchAll();
}
$patients = $pdo->query("SELECT id, last_name, first_name FROM patients
ORDER BY last_name")->fetchAll();
$doctorsStmt  = $pdo->prepare("SELECT id, username FROM users WHERE role =
?");
$doctorsStmt->execute(['doctor']);
$doctors = $doctorsStmt->fetchAll();
$statuses = ['Заплановано','Прийшов','Запізнюється','Прийом
розпочато','Прийом завершено'];
?>
<div style="margin-bottom: 20px;">
  <h2 style="margin-bottom: 10px; color: var(--primary-color);">Список
прийомів</h2>
  <?php if ($error): ?>
    <div style="color: var(--error-text); background: var(--error-bg);
padding: 10px; border-radius: 6px;"><?= htmlspecialchars($error) ?></div>
  <?php endif; ?>
  <?php if ($success): ?>
    <div style="color: var(--success-text); background: var(--success-bg);
padding: 10px; border-radius: 6px;"><?= htmlspecialchars($success) ?></div>
  <?php endif; ?>
</div>
<div style="overflow-x: auto;">
  <table style="width:100%; border-collapse: collapse; background: var(--bg-
white); border-radius: 2px; overflow: hidden; box-shadow: 0 0 10px
rgba(0,0,0,0.05);">
    <thead style="background-color: var(--primary-color); color: var(--bg-
white);">
      <tr>
      <th style="padding: 10px; border: 1px solid var(--border-
color);">Дата</th>
      <th style="padding: 10px; border: 1px solid var(--border-
color);">Час</th>
      <th style="padding: 10px; border: 1px solid var(--border-
color);">Пацієнт</th>
      <th style="padding: 10px; border: 1px solid var(--border-
color);">Лікар</th>
      <th style="padding: 10px; border: 1px solid var(--border-
color);">Статус</th>
```

```
      <th style="padding: 10px; border: 1px solid var(--border-
color);">Дії</th>
      </tr>
    </thead>
    <tbody>
      <?php foreach ($appointments as $a): ?>
      <tr style="border-top: 1px solid #eee;">
        <td style="padding: 10px; border: 1px solid var(--border-
color);"><?= htmlspecialchars($a['appointment_date']) ?></td>
        <td style="padding: 10px; border: 1px solid var(--border-
color);"><?= htmlspecialchars($a['time_start'] . '–' . $a['time_end'])
?></td>
        <td style="padding: 10px; border: 1px solid var(--border-
color);"><?= htmlspecialchars($a['last_name'] . ' ' . $a['first_name'])
?></td>
        <td style="padding: 10px; border: 1px solid var(--border-
color);"><?= htmlspecialchars($a['doctor']) ?></td>
        <td style="padding: 10px; border: 1px solid var(--border-
color);"><?= htmlspecialchars($a['status']) ?></td>
      <td style="padding: 10px; border: 1px solid var(--border-color);">
          <?php if ($_SESSION['role'] === 'admin' || $_SESSION['role'] ===
'reception'): ?>
            <form method="post" style="display:flex; gap: 5px;">
              <input type="hidden" name="appointment_id" value="<?= $a['id']
?>">
              <select name="status" required style="width: 50%; padding:
8px; border:1px solid var(--border-color); border-radius:4px;">
                <?php foreach ($statuses as $st): ?>
                  <option value="<?= htmlspecialchars($st) ?>" <?= $st ===
$a['status'] ? 'selected' : '' ?>>
                    <?= htmlspecialchars($st) ?>
                  </option>
                <?php endforeach; ?>
              </select>
              <button type="submit" name="update_status" style="background:
var(--primary-color); color: var(--bg-white); border: none; padding: 6px
10px; border-radius: 4px; cursor: pointer;">Змінити</button>
            </form>
          <?php else: ?>
            –
          <?php endif; ?>
        </td>
      </tr>
      <?php endforeach; ?>
    </tbody>
  </table>
</div>
<?php if ($_SESSION['role'] === 'admin' || $_SESSION['role'] ===
'reception'): ?>
  <div style="margin-top: 40px;">
    <h2 style="color: var(--primary-color);">Додати новий прийом</h2>
    <form method="post" style="background: var(--bg-white); padding: 20px;
border-radius: 8px; box-shadow: 0 0 10px rgba(0,0,0,0.05); display: flex;
flex-direction: column; gap: 15px; max-width: 500px;">
      <label>Пацієнт:
```

```
        <select name="patient_id" required style="width: 100%; padding: 8px;
border:1px solid var(--border-color); border-radius:4px;">
          <option value="">– Оберіть –</option>
          <?php foreach ($patients as $p): ?>
            <option value="<?= htmlspecialchars($p['id']) ?>"><?=
htmlspecialchars($p['last_name'] . ' ' . $p['first_name']) ?></option>
          <?php endforeach; ?>
        </select>
      </label>
      <label>Лікар:
        <select name="doctor_id" required style="width: 100%; padding: 8px;
border:1px solid var(--border-color); border-radius:4px;">
          <option value="">– Оберіть –</option>
          <?php foreach ($doctors as $d): ?>
            <option value="<?= htmlspecialchars($d['id']) ?>"><?=
htmlspecialchars($d['username']) ?></option>
          <?php endforeach; ?>
        </select>
      </label>
      <label>Дата:<input type="date" name="appointment_date" required
style="width: 100%; padding: 8px; border:1px solid var(--border-color);
border-radius:4px;"></label>
      <label>Початок:<input type="time" name="time_start" required
style="width: 100%; padding: 8px; border:1px solid var(--border-color);
border-radius:4px;"></label>
      <label>Кінець:<input type="time" name="time_end" required
style="width: 100%; padding: 8px; border:1px solid var(--border-color);
border-radius:4px;"></label>

      <button type="submit" name="add_appointment" style="background: var(--
primary-color); color: var(--bg-white); border: none; padding: 10px; border-
radius: 6px; font-size: 16px; cursor: pointer;">Додати прийом</button>
    </form>
  </div>
<?php endif; ?>
```

Лістинг Г.10 – Файл medications.php

```
<?php
require_once __DIR__ . '/../db.php';
$error = '';
$success = '';
$updated = '';
$deleted = '';
$added = '';
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    if (isset($_POST['add_medication'])) {
        $name     = trim($_POST['name'] ?? '');
        $quantity = intval($_POST['quantity'] ?? 0);

        if ($name !== '' && $quantity >= 0) {
            $pdo->prepare("INSERT INTO medications (NAME, quantity) VALUES
(?, ?)")
                ->execute([$name, $quantity]);
```

```
            $success = "Препарат успішно додано!";
        } else {
            $error = "Введіть правильні дані.";
        }
    }
    if (isset($_POST['update_quantity'])) {
        $id       = intval($_POST['medication_id'] ?? 0);
        $quantity = intval($_POST['quantity'] ?? 0);

        if ($id > 0 && $quantity >= 0) {
            $pdo->prepare("UPDATE medications SET quantity = ? WHERE id =
?")
                ->execute([$quantity, $id]);
            $success = "Кількість успішно оновлено!";
        } else {
            $error = "Невірні дані для оновлення.";
        }
    }
    if (isset($_POST['delete_medication'])) {
        $id = intval($_POST['medication_id'] ?? 0);

        if ($id > 0) {
            $pdo->prepare("DELETE FROM medications WHERE id = ?")-
>execute([$id]);
            $success = "Препарат успішно видалено!";
        } else {
            $error = "Невірний ID препарату.";
        }
    }
}
$medications = $pdo->query("SELECT * FROM medications ORDER BY NAME")-
>fetchAll();
?>
<div style="margin-bottom: 20px;">
  <h2 style="margin-bottom: 10px; color: var(--primary-color);">Список
препаратів</h2>
  <?php if ($error): ?>
    <div style="color: var(--error-text); background: var(--error-bg);
padding: 10px; border-radius: 6px;"><?= htmlspecialchars($error) ?></div>
  <?php elseif ($added): ?>
    <div style="color: var(--success-text); background: var(--success-bg);
padding: 10px; border-radius: 6px;">Препарат успішно додано!</div>
  <?php elseif ($updated): ?>
    <div style="color: var(--success-text); background: var(--success-bg);
padding: 10px; border-radius: 6px;">Кількість успішно оновлено!</div>
  <?php elseif ($deleted): ?>
    <div style="color: var(--success-text); background: var(--success-bg);
padding: 10px; border-radius: 6px;">Препарат успішно видалено!</div>
  <?php endif; ?>
</div>
<div style="overflow-x: auto;">
  <table style="width:100%; border-collapse: collapse; background: var(--bg-
white); border-radius: 2px; overflow: hidden; box-shadow: 0 0 10px
rgba(0,0,0,0.05);">
    <thead style="background-color: var(--primary-color); color: var(--bg-
white);">
```

```
      <tr>
        <th style="padding: 10px; border: 1px solid var(--border-
color);">Назва</th>
        <th style="padding: 10px; border: 1px solid var(--border-
color);">Кількість</th>
        <th style="padding: 10px; border: 1px solid var(--border-
color);">Дії</th>
      </tr>
    </thead>
    <tbody>
      <?php foreach ($medications as $med): ?>
<tr>
  <td style="padding:10px; border:1px solid var(--border-color);"><?=
htmlspecialchars($med['NAME']) ?></td>
  <td style="padding:10px; border:1px solid var(--border-color);"><?=
$med['quantity'] ?></td>
  <td style="padding:10px; border:1px solid var(--border-color);">
    <div style="display:flex; gap:8px; align-items:center;">
      <form method="post" style="display:flex; gap:4px; align-
items:center;">
        <input type="hidden" name="medication_id" value="<?= $med['id'] ?>">
        <input type="number" name="quantity" value="<?= $med['quantity'] ?>"
min="0" required style="width: 100%; padding: 8px; border:1px solid var(--
border-color); border-radius:4px;">
        <button type="submit" name="update_quantity" style="background:var(-
-primary-color); color:var(--bg-white); border:none; padding:6px 10px;
border-radius:4px; cursor:pointer;">
          Оновити
        </button>
      </form>
      <form method="post" style="display:flex; gap:4px; align-items:center;"
onsubmit="return confirm('Видалити препарат?');">
        <input type="hidden" name="medication_id" value="<?= $med['id'] ?>">
        <button type="submit" name="delete_medication"
style="background:var(--error-text); color:var(--bg-white); border:none;
padding:6px 10px; border-radius:4px; cursor:pointer;">
          Видалити
        </button>
      </form>
    </div>
  </td>
</tr>
<?php endforeach; ?>
    </tbody>
  </table>
</div>
<div style="margin-top: 40px;">
  <h2 style="color: var(--primary-color);">Додати новий препарат</h2>
  <form method="post" style="background: var(--bg-white); padding:20px;
border-radius:8px; box-shadow:0 0 10px rgba(0,0,0,0.05); display:flex; flex-
direction:column; gap:15px; max-width:500px;">
    <label>Назва:<br>
      <input type="text" name="name" required style="width:100%;
padding:8px; border:1px solid var(--border-color); border-radius:4px;">
    </label>
    <label>Кількість:<br>
```

```
      <input type="number" name="quantity" min="0" value="0" required
style="width:100%; padding:8px; border:1px solid var(--border-color);
border-radius:4px;">
    </label>
    <button type="submit" name="add_medication" style="background: var(--
primary-color); color: var(--bg-white); border:none; padding:10px; border-
radius:6px; font-size:16px; cursor:pointer;">
      Додати
    </button>
  </form>
</div>
<?php if ($success): ?>
<script>
  document.addEventListener('DOMContentLoaded', () => {
    document.querySelectorAll('.med-form, .inline-form').forEach(form =>
form.reset && form.reset());
    if (history.replaceState) {
      const url = new URL(window.location);
      const tab = url.searchParams.get('tab');
      const newUrl = window.location.pathname + (tab ? `?tab=${tab}` : '');
      history.replaceState(null, '', newUrl);
    }
  });
</script>
<?php endif; ?>
```

Лістинг Г.11 – Файл patients.php

```
<?php
require_once __DIR__ . '/../db.php';
$add_message = '';
$added = false;
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['add_patient'])) {
   $first_name  = trim($_POST['first_name']  ?? '');
    $last_name   = trim($_POST['last_name']   ?? '');
    $middle_name = trim($_POST['middle_name'] ?? '');
    $phone       = trim($_POST['phone']       ?? '');
    $email       = trim($_POST['email']       ?? '');
    $birth_date  = $_POST['birth_date'] ?? null;
    if ($first_name !== '' && $last_name !== '') {
        $stmt = $pdo->prepare("
            INSERT INTO patients
              (first_name, last_name, middle_name, phone, email, birth_date)
            VALUES (?, ?, ?, ?, ?, ?)
        ");
        $stmt->execute([
            $first_name,
            $last_name,
            $middle_name !== '' ? $middle_name : null,
            $phone       !== '' ? $phone       : null,
            $email       !== '' ? $email       : null,
            $birth_date  !== '' ? $birth_date  : null,
        ]);
```

```
        $add_message = "Пацієнта успішно додано!";
        $added = true;
    } else {
        $add_message = "Будь ласка, заповніть обов'язкові поля.";
    }
}
?>
<div style="background: var(--bg-white); padding: 20px; border-radius: 12px;
box-shadow: 0 2px 8px rgba(0,0,0,0.05); max-width: 600px; margin-bottom:
30px; border: none; color: var(--text-color);">
  <h2 style="margin-bottom: 10px; color: var(--primary-color);">Додати
пацієнта</h2>
  <?php if ($add_message): ?>
    <p style=" color: <?= $added ? 'var(--success-text)' : 'var(--error-
text)' ?>; background-color: <?= $added ? 'var(--success-bg)' : 'var(--
error-bg)' ?>; padding: 10px; border-radius: 6px; margin-bottom: 15px;
border: 1px solid <?= $added ? 'var(--success-text)' : 'var(--error-text)'
?>;">
      <?= htmlspecialchars($add_message) ?>
    </p>
  <?php endif; ?>
  <form id="patient-form" method="post" style="display: flex; flex-
direction: column; gap: 15px;">
    <label>
      Ім’я*:<br>
      <input type="text" name="first_name" required style="border: 1px solid
var(--border-color); padding: 8px; border-radius: 6px; width: 100%;">
    </label>
    <label>
      Прізвище*:<br>
      <input type="text" name="last_name" required style="border: 1px solid
var(--border-color); padding: 8px; border-radius: 6px; width: 100%;">
    </label>
    <label>
      По батькові:<br>
      <input type="text" name="middle_name" style="border: 1px solid var(--
border-color); padding: 8px; border-radius: 6px; width: 100%;">
    </label>
    <label>
      Телефон:<br>
      <input type="text" name="phone" style="border: 1px solid var(--border-
color); padding: 8px; border-radius: 6px; width: 100%;">
    </label>
    <label>
      Email:<br>
      <input type="email" name="email" style="border: 1px solid var(--
border-color); padding: 8px; border-radius: 6px; width: 100%;">
    </label>
    <label>
      Дата народження:<br>
      <input type="date" name="birth_date" style="border: 1px solid var(--
border-color); padding: 8px; border-radius: 6px; width: 100%;">
    </label>
    <button type="submit" name="add_patient" style="padding: 10px 20px;
background-color: var(--primary-color); border: none; color: white; border-
radius: 6px; cursor: pointer; width: fit-content;">
```

```
      Додати пацієнта
    </button>
  </form>
</div>
<?php if ($added): ?>
<script>
document.addEventListener('DOMContentLoaded', () => {
  const form = document.getElementById('patient-form');
  if (form) form.reset();
  if (history.replaceState) {
    const url = new URL(window.location);
    const tab = url.searchParams.get('tab');
    const newUrl = window.location.pathname + (tab ? `?tab=${tab}` : '');
    history.replaceState(null, '', newUrl);
  }
});
</script>
<?php endif; ?>
```

Лістинг Г.12 – Файл treatments.php

```
<?php
require_once __DIR__ . '/../db.php';
$error = '';
$success = '';
$added = false;
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['add_treatment']))
{
    $appointment_id = intval($_POST['appointment_id'] ?? 0);
    $medication_id  = intval($_POST['medication_id'] ?? 0);
    $dosage         = trim($_POST['dosage'] ?? '');
    $instructions   = trim($_POST['instructions'] ?? '');
    if ($appointment_id > 0 && $medication_id > 0 && $dosage !== '' &&
$instructions !== '') {
        $stmt = $pdo->prepare("INSERT INTO treatments (appointment_id,
medication_id, dosage, instructions) VALUES (?, ?, ?, ?)");
        $stmt->execute([$appointment_id, $medication_id, $dosage,
$instructions]);
        $stmt2 = $pdo->prepare("UPDATE appointments SET status = 'Прийом
завершено' WHERE id = ?");
        $stmt2->execute([$appointment_id]);
        $success = "Призначення збережено! Статус прийому оновлено на
'Прийом завершено'.";
        $added = true;
    } else {
        $error = "Будь ласка, заповніть усі поля.";
    }
}
$appointments = $pdo->query("
    SELECT a.id, CONCAT(p.last_name, ' ', p.first_name) AS patient,
a.appointment_date
    FROM appointments a
    JOIN patients p ON a.patient_id = p.id
    ORDER BY a.appointment_date DESC
")->fetchAll();
```

```
$medications = $pdo->query("
    SELECT id, NAME AS name
    FROM medications
    ORDER BY NAME ASC
")->fetchAll();
$treatments = $pdo->query("
    SELECT t.id, a.appointment_date, CONCAT(p.last_name, ' ', p.first_name)
AS patient,
           m.NAME AS medication, t.dosage, t.instructions
    FROM treatments t
    JOIN appointments a ON t.appointment_id = a.id
    JOIN patients p ON a.patient_id = p.id
    JOIN medications m ON t.medication_id = m.id
    ORDER BY a.appointment_date DESC, t.id DESC
")->fetchAll();
?>
<div style="background: var(--bg-white); padding: 20px; border-radius: 12px;
box-shadow: 0 2px 8px rgba(0,0,0,0.05); margin-bottom: 30px; max-width:
700px; color: var(--text-color);">
  <h2 style="margin-bottom: 10px; color: var(--primary-color);">Призначення
лікування</h2>
  <?php if ($error): ?>
    <p style="color: var(--error-text); background: var(--error-bg);
padding: 10px; border-radius: 6px; margin-bottom: 15px;">
      <?= htmlspecialchars($error) ?>
    </p>
  <?php elseif ($success): ?>
    <p style="color: var(--success-text); background: var(--success-bg);
padding: 10px; border-radius: 6px; margin-bottom: 15px;">
      <?= htmlspecialchars($success) ?>
    </p>
  <?php endif; ?>
  <form id="treatment-form" method="post" style="display: flex; flex-
direction: column; gap: 15px;">
    <input type="hidden" name="add_treatment" value="1">
    <label>
      Прийом*:
      <select name="appointment_id" required style="width: 100%; padding:
8px; border: 1px solid var(--border-color); border-radius: 6px;">
        <option value="">– Оберіть –</option>
        <?php foreach ($appointments as $a): ?>
          <option value="<?= $a['id'] ?>"><?=
htmlspecialchars($a['appointment_date'] . ' – ' . $a['patient']) ?></option>
        <?php endforeach; ?>
      </select>
    </label>
    <label>
      Препарат*:
      <select name="medication_id" required style="width: 100%; padding:
8px; border: 1px solid var(--border-color); border-radius: 6px;">
        <option value="">– Оберіть –</option>
        <?php foreach ($medications as $m): ?>
          <option value="<?= $m['id'] ?>"><?= htmlspecialchars($m['name'])
?></option>
        <?php endforeach; ?>
      </select>
```

```
    </label>
    <label>
      Дозування*:
      <input type="text" name="dosage" required style="width: 100%; padding:
8px; border: 1px solid var(--border-color); border-radius: 6px;">
    </label>
    <label>
      Інструкції*:
      <textarea name="instructions" rows="3" required style="width: 100%;
padding: 8px; border: 1px solid var(--border-color); border-radius: 6px;
resize: vertical;"></textarea>
    </label>
    <button type="submit" style="background: var(--primary-color); color:
white; padding: 10px 20px; border: none; border-radius: 6px; cursor:
pointer; width: fit-content;">
      Додати призначення
    </button>
  </form>
</div>
<?php if ($added): ?>
<script>
document.addEventListener('DOMContentLoaded', () => {
  const form = document.getElementById('treatment-form');
  if (form) form.reset();
  if (history.replaceState) {
    const url = new URL(window.location);
    const tab = url.searchParams.get('tab') || 'treatments';
    history.replaceState(null, '', window.location.pathname + '?tab=' +
tab);
  }
});
</script>
<?php endif; ?>
<div style="background: var(--bg-white); padding: 20px; border-radius: 12px;
box-shadow: 0 2px 8px rgba(0,0,0,0.05); color: var(--text-color);">
  <h2 style="margin-bottom: 10px; color: var(--primary-color);">Існуючі
призначення</h2>
  <?php if (count($treatments) === 0): ?>
    <p>Призначень ще немає.</p>
  <?php endif; ?>
</div>
```