

Міністерство освіти і науки України
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра комп'ютерної інженерії

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри,
канд. техн. наук, доцент

_____ Я. М. Крайник

« __ » _____ 2022 р.

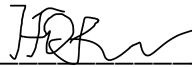
КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

Система веб-інтерфейсу для роботи з мережевими каталогами

Спеціальність 123 Комп'ютерна інженерія

123 – КР.1 – 405.21810504

Студент:

 Ю.О. Гончаренко
« __ » _____ 2022 р.

Керівник: канд. техн. наук, доцент

_____ Я.М. Крайник
« __ » _____ 2022 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра комп'ютерної інженерії

ЗАТВЕРДЖУЮ

Зав.кафедри _____ Я. М. Крайник

« _____ » _____ 2022_р.

ЗАВДАННЯ
На виконання кваліфікаційної роботи

Видано студенту групи 405 факультету комп'ютерних наук

Гончаренко Юрій Олегович

(прізвище, ім'я, по батькові студента)

1. Тема кваліфікаційної роботи

Система веб-інтерфейсу для роботи з мережевими каталогами

Затверджена наказом по ЧНУ від «21» _____ грудня _____ 2021 р. №298

2. Строк представлення кваліфікаційної роботи «27» червня _____ 2022 _____ р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні

Спроектований та побудований локальний сервер на основі DNS з можливістю налаштування локального сховища. Сервер побудований на базі OpenServer.

4. Перелік питань, що підлягають розробці Аналіз інформації стосовно апаратних та програмних рішень побудови локального сервера на базі OpenServer, практичне використання такого сервера, подальше можливе вдосконалення технологій побудови локального сервера. Безпека зберігання даних на локальному сервері.

5. Перелік графічних матеріалів

Макетна схема локального сервера

Принципова схема апаратно-програмного комплексу локального сервера

Блок-схема алгоритму роботи ПЗ локального сервера

6. Завдання до спеціальної частини

Визначення порядку побудови локального сервера

7. Консультанти:

Консультант	Кафедра(організація)	Частина роботи
А. О. Алексеєва	кафедра екології Медичного інституту	Спеціальна частина з охорони праці

Керівник роботи канд. тех. наук, доцент Крайник Ярослав Михайлович

(посада, прізвище, ім'я, по батькові)

(підпис)

Завдання прийнято до виконання

Гончаренко Юрій Олегович

(прізвище, ім'я, по батькові студента)



(підпис)


Дата видачі завдання « 23__ » листопада _____ 2021__ р.

КАЛЕНДАРНИЙ ПЛАН
Виконання кваліфікаційної роботи

Тема: Система веб-інтерфейсу для роботи з мережевими каталогами

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Розробка та затвердження завдання на виконання КР	18.11.21	19.11.21	Виконав
2.	Аналіз інформації стосовно технології DNS: особливості, особливості вибору системи, сфери застосування, переваги та недоліки	23.11.21	03.12.21	Виконав
3.	Порівняльний аналіз існуючих технічних рішень	07.12.21	09.12.21	Виконав
4.	Підбір необхідних для реалізації системи компонентів	15.12.21	25.12.21	Виконав
5.	Проектування прототипу локального сервера на основі OpenServer	10.01.22	22.01.22	Виконав
6.	Реалізація програмної частини локального сервера на основі OpenServer	05.02.22	09.03.22	Виконав
8.	Тестування програмної частини локального сервера на основі OpenServer	14.03.22	18.03.22	Виконав
9.	Фінальна збірка локального сервера	03.04.22	11.04.22	Виконав
10.	Оформлення роботи	08.05.22	16.05.22	Виконав
11.	Підготовка презентації	17.05.22	19.05.22	Виконав
12.	Попередній захист	10.06.22	10.06.22	Виконав
13.	Рецензування	15.06.22	15.06.22	Виконав
14.	Завершення оформлення КР та презентації	22.06.22	25.06.22	Виконав
15.	Захист кваліфікаційної роботи	27.06.22	27.06.22	Виконав

Розробив студент Гончаренко Юрій Олегович
(прізвище, ім'я, по батькові)


(підпис)

«__» 2022_p.

Керівник роботи доцент Крайник Ярослав Михайлович
(посада, прізвище, ім'я, по батькові)

(підпис)

«__» 2022_p.

АНОТАЦІЯ

до кваліфікаційної роботи

«Система веб-інтерфейсу для роботи з мережевими каталогами»

Студент _405__ гр.: Гончаренко Ю.О.

Керівник: канд. тех. наук, доцент Крайник Я. М.

Кваліфікаційна робота присвячена розробці локального сервера на основі OpenServer з кешуючим DNS-сервером. Актуальність цієї роботи обумовлена малою кількістю на ринку програмних локальних серверів для невеликої мережі. Метою кваліфікаційної роботи є вивчення апаратних та програмних засобів для встановлення локального сервера.

Практичним значенням результатів дослідження та розробки локального сервера є можливість побудови невеликої мережі замкнутого циклу.

Пояснювальна записка кваліфікаційної роботи складається зі вступу, трьох розділів та висновків. У вступі визначається актуальність теми, сформульовані мета, об'єкт, предмет та завдання роботи. У першому розділі розглядаються мережеві технології. Увага приділяється особливостям побудови, основним поняттям та зроблено особливий наголос на історії та еволюції мережевих технологій, оскільки це дає змогу зробити аналіз та спрогнозувати подальший їх розвиток.

У другому розділі представлено огляд такого поняття, як DNS, зроблено акцент на первинному та вторинному DNS-сервері, оскільки практична частина побудована саме на повноцінному DNS-сервері. Окремо приділяється увага на побудові DNS-сервера на базі Linux.

У третьому розділі безпосередньо описується практична побудова локального DNS-сервера на базі OpenServer, що є оптимальним варіантом для невеликої локальної мережі. Дається детальний опис побудови сторінки для локальної мережі, технологія локального сховища. На основі HTML, CSS, JavaScript пишеться програмний код для майбутньої сторінки.

Робота містить: 75 сторінок основного тексту, 38 малюнків, 32 використаних джерела.

Ключові слова: мережеві технології, локальні мережі, сайт, сервер, DNS-сервер, активна директорія, кеш.

ANNOTATION

to qualifying work

"Web interface system for working with network directories"

Student __405_ gr .: Honcharenko Yurii

Scientific Advisor: Ph.D. in Computer Systems and Components, Associate Professor
Y. M. Krainyk

Qualification work is devoted to the development of a local server based on OpenServer with a caching DNS server. The relevance of this work is due to the small number on the market of software local servers for a small network. The purpose of the qualification work is to study the hardware and software for installing a local server.

The practical significance of the results of research and development of a local server is the ability to build a small closed loop network.

The explanatory note of the qualification work consists of an introduction, three sections and conclusions. The introduction determines the relevance of the topic, formulates the purpose, object, subject and objectives of the work. The first section discusses network technologies. Attention is paid to the peculiarities of construction, the basic concepts and special emphasis is placed on the history and evolution of network technologies, as it allows analyzing and predicting their further development.

The second section provides an overview of the concept of DNS, with an emphasis on the primary and secondary DNS server, as the practical part is built on a full-fledged DNS server. Special attention is paid to building a DNS server based on Linux.

The third section directly describes the practical construction of a local DNS server based on OpenServer, which is the best option for a small local network. The detailed description of construction of the page for a local area network, technology of local storage is given. On the basis of HTML, CSS, JavaScript the program code for the future page is written.

The work contains: 75 pages of the main text, 38 pictures, 32 used sources.

Keywords: network technologies, local networks, site, server, DNS server, active directory, cache.

ЗМІСТ

Скорочення та умовні позначки	3
ВСТУП	4
РОЗДІЛ 1 ОГЛЯД МЕРЕЖЕВИХ КАТАЛОГІВ	7
1.1 Основні поняття	7
1.2 Основні характеристики мережеских ресурсів	8
1.3 Еволюція служб каталогів	15
РОЗДІЛ 2 СИСТЕМА DNS	20
2.1 Загальні поняття	20
2.2 Налаштування кешуючого DNS-сервера unbound	24
2.3 Налаштування повноцінного DNS-сервера	30
2.4 Налаштування вторинного DNS-сервера	33
2.5 DNS server BIND на Linux	35
Висновки до розділу 2	40
РОЗДІЛ 3 СТВОРЕННЯ WEB-ІНТЕРФЕЙСУ ДЛЯ МЕРЕЖЕВОГО КАТАЛОГУ	41
3.1 Розмітка сторінки, css	41
3.2 Сторінка services з вихідними тестовими файлами	59
3.3 Створення локального серверу	64
Висновки до розділу 3	73
ВИСНОВКИ	74
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ПОСИЛАНЬ	76

Скорочення та умовні позначки

IT	–	Information Technology. Інформаційні технології.
DNS	–	Domain Name System. Система доменних імен.
DS	–	Directory Service. Служба каталогів.
IRF	–	Inheritance Rights Filter. Фільтри успадкованих прав.
LDA	–	Lightweight Directory Access Protocol. Полегшений протокол доступу до каталогів
TCP/IP	–	Transmission Control Protocol / Internet Protocol. Протокол управління передачею/протокол Інтернету.
SSL	–	Secure Sockets Layer. Протоколу захищених сокетів.

ВСТУП

Корпоративні обчислювальні середовища мають потребу в централізованому зберіганні інформації, сховищі даних, щоб користувачі могли їх додавати, видаляти, змінювати та робити запит до додатків [1]. Збереженою інформацією можуть бути облікові записи користувачів, адреси електронної пошти, цифрові сертифікати, імена об'єктів компонентів, мережеві назви, принтери, групи тощо. В цьому необхідність доступу до цієї інформації як зсередини підприємства, так і ззовні [2]. Обсяг інформації, що зберігається, дуже залежить від клієнта. Ці дані зараз відомі як служба каталогів.

Служби каталогів не тільки дозволяють знайти ці ресурси та отримати доступ до них, а й дозволити керувати зв'язками між ними.

Як зразок можна взяти довідник. Для власного використання користувачі мають особистий довідник з адресами що в ньому зберігаються. Номери телефонів та іншу інформацію зберігається у найбільш зручному форматі. Але коли ми говоримо про підтримку глобальної служби каталогів в Інтернеті або в будь-якій організації, служба каталогів має відповідати таким критеріям:

- Бути досить гнучкими для зберігання різноманітних типів інформації
- Надавати безпечний доступ як з Інтернету, так і з інтранету
- Можливість масштабування від малого бізнесу до найбільшого підприємства.

Використання відкритого протоколу дає змогу клієнтам різних постачальників отримати доступ до інформації в службі каталогів. Служби каталогів від різних постачальників, які спілкуються за допомогою відкритого протоколу, можуть обмінюватися інформацією один з одним для створення зведених каталогів. Саме в обміні інформацією і полягає важливість служб каталогів.

Основні функції служби каталогів:

1. Керування користувачами та групами (створення/видалення, налаштування прав доступу).
2. Управління ресурсами (подання до загального доступу, встановлення обмежень, віддалене адміністрування тощо).

Також служби каталогів дозволяють робити:

- A. пошук ресурсів;
- B. поширення мережових політик [3].

Все це робить тему кваліфікаційної роботи дуже актуальною у наш час.

Об'єктом є локальна мережа.

Предметом є локальний сервер.

Метою кваліфікаційної роботи є вивчення апаратних та програмних засобів для встановлення локального сервера. Для реалізації цього завдання потрібно вирішити такі задачі:

- розглянути програмні засоби для побудови локальної мережі;
- провести огляд засобів для встановлення локального сервера;
- зробити вибір мережевого обладнання;
- зробити вибір програмного серверного обладнання.

Етапи проектування:

- аналіз варіантів побудови локальної мережі;
- підбір необхідного обладнання;

- створення корпоративного сайту;
- виведення корпоративного сайту у локальну мережу;
- створення бази даних сайту.

Практична значимість полягає у реалізації практичних заходів із проектуванню сайту, написанню його програмного коду та виведення сайту в локальну мережу.

Все це робить тему кваліфікаційної роботи дуже актуальною у наш час.

Об'єктом є локальна мережа.

Предметом є локальний сервер.

Метою кваліфікаційної роботи є вивчення апаратних та програмних засобів для встановлення локального сервера. Для реалізації цього завдання потрібно вирішити такі задачі:

- розглянути програмні засоби для побудови локальної мережі;
- провести огляд засобів для встановлення локального сервера;
- зробити вибір мережевого обладнання;
- зробити вибір програмного серверного обладнання.

При побудові локальних мереж важливим є вміння розбиратися в обладнанні, необхідному для організації мережі, архітектурі та існуючих технологіях мережі.

Етапи проектування:

- аналіз варіантів побудови локальної мережі;
- підбір необхідного обладнання;
- створення корпоративного сайту;
- виведення корпоративного сайту у локальну мережу;
- створення бази даних сайту.

Практична значимість полягає у реалізації практичних заходів із проектуванню сайту, написанню його програмного коду та виведення сайту в локальну мережу.

РОЗДІЛ 1 ОГЛЯД МЕРЕЖЕВИХ КАТАЛОГІВ

1.1 Основні поняття

Щоб засвоїти роботу каталогів, важливо зрозуміти декілька нових термінів.

Інформація в каталозі зберігається як об'єкти або записи спеціальних полів, які називаються атрибутами. Набір характеристик, їх правопис і вимоги заповнення вирішуються самим каталогом. Інформація каталогу може бути представлена у вигляді схожої на дерево структури. Це нагадує базис, який використовують багато файлових систем [4]. Кожен об'єкт в сховищі визначається певною характеристикою Dynamic Name (DN) (Видозмінене ім'я [5]). За аналогією з DN, вона показує, що об'єкт можливо знайти в дереві каталогів. Будь-який DN має закінчуватися суфіксом DN, що зветься суфіксом каталогу, та коренем дерева [6].

Кореневий об'єкт [7] є основним елементом в сховищі файлів. DN кореневого об'єкта цілком відповідає суфіксу кореня каталогу (видозмінене ім'я кореня), певному об'єкту, що дає опис адміністратора каталогу. Як правило, об'єкт такого типу не має суфікса.

Базове видозмінене ім'я [8] є об'єктом каталогу, з якого здійснюється пошук. Справа в тому, що не завжди потрібно шукати по всьому дереву сховища. Є можливість зменшити рівень запитів. Автоматично цей параметр відповідає суфіксу.

1.2 Основні характеристики мережевих ресурсів

Існує багато типів мережевих каталогів або серверів, включаючи веб-сервери, поштові сервери та віртуальні сервери.

Окрема система може надавати ресурси та використовувати їх з іншої системи одночасно. Це означає, що пристрій може бути і сервером, і клієнтом одночасно [9].

Спочатку такі сервери були підключені до клієнтів, відомих як термінали, які не виконували жодних фактичних обчислень. Ці термінали, звані «тупими» терміналами, існували просто для того, щоб приймати введення через клавіатуру або пристрій для зчитування карт і повертати результати будь-яких обчислень на екран дисплея або принтер. Фактичні обчислення були зроблені на сервері.

Пізніше сервери часто були поодинокими потужними комп'ютерами, підключеними через мережу до набору менш потужних клієнтських комп'ютерів. Цю архітектуру мережі часто називають моделлю клієнт-сервер, в якій і клієнтський комп'ютер, і сервер володіють обчислювальною потужністю, але певні завдання делегуються серверам. У попередніх обчислювальних моделях, таких як модель мейнфрейм-термінал, мейнфрейм дійсно виконував роль сервера, навіть якщо він не називався під цим ім'ям [10].

Оскільки технології розвивалися, разом з нею розвивалося й визначення сервера. У наші дні сервер може бути не що інше, як програмне забезпечення, що працює на одному або кількох фізичних комп'ютерних пристроях. Такі сервери часто називають віртуальними серверами. Спочатку віртуальні сервери використовувалися для збільшення кількості функцій сервера, які може виконувати один апаратний сервер.

Сервер може бути розроблений для виконання окремого завдання, наприклад, поштовий сервер, який приймає та зберігає електронну пошту, а потім

надає її клієнту, який запитує. Сервери також можуть виконувати кілька завдань, таких як сервер файлів і друку, який одночасно зберігає файли та приймає завдання друку від клієнтів, а потім надсилає їх на підключений до мережі принтер.

Щоб функціонувати як сервер, пристрій має бути налаштований на прослуховування запитів від клієнтів у мережевому з'єднанні. Ця функція може існувати як частина операційної системи як інстальована програма, роль або їх комбінація.

Наприклад, операційна система Microsoft Windows Server надає функціональні можливості для прослуховування запитів клієнтів і відповіді на них. Додатково встановлені ролі або служби збільшують кількість запитів клієнтів, на які сервер може реагувати. В іншому прикладі веб-сервер Apache відповідає на запити Інтернет-браузера через додаткову програму Apache, встановлену поверх операційної системи [11].

Коли клієнту потрібні дані або функціональні можливості від сервера, він надсилає запит по мережі. Сервер отримує цей запит і відповідає відповідною інформацією.

Типи серверів

Існує багато типів серверів, які виконують різні функції. Багато мереж містять один або кілька поширених типів серверів:

Файлові сервери зберігають і розповсюджують файли. Декілька клієнтів або користувачів можуть ділитися файлами, що зберігаються на сервері. Крім того, централізоване зберігання файлів пропонує простіші рішення щодо резервного копіювання або відмовостійкості, ніж спроби забезпечити безпеку та цілісність файлів на кожному пристрої в організації. Апаратне забезпечення файлового сервера може бути розроблене для максимальної швидкості читання та запису для підвищення продуктивності.

Сервери друку дозволяють керувати та розподіляти функціональні можливості друку. Замість того, щоб підключати принтер до кожної робочої станції, один сервер друку може відповідати на запити друку від численних клієнтів. Сьогодні деякі більші та високоякісні принтери мають власний вбудований сервер друку, який усуває потребу в додатковому комп'ютерному сервері друку. Цей внутрішній сервер друку також функціонує, відповідаючи на запити на друк від клієнта.

Сервери додатків запускають програми замість клієнтських комп'ютерів, на яких запущені програми локально. Сервери додатків часто запускають ресурсомісткі програми, які спільно використовують велика кількість користувачів. Це усуває потребу для кожного клієнта мати достатні ресурси для запуску програм. Це також усуває необхідність встановлення та обслуговування програмного забезпечення на багатьох машинах, а не лише на одному.

Сервери системи доменних імен (DNS) — це сервери додатків, які забезпечують розпізнавання імен клієнтським комп'ютерам шляхом перетворення імен, легко зрозумілих людям, у машиночитані IP-адреси. Система DNS — це широко поширена база даних імен та інших серверів DNS, кожен з яких можна використовувати для запиту невідомого імені комп'ютера. Коли клієнту потрібна адреса системи, він надсилає DNS-запит із назвою потрібного ресурсу на DNS-сервер. DNS-сервер відповідає з необхідною IP-адресою зі своєї таблиці імен.

Поштові сервери є дуже поширеним типом серверів додатків. Поштові сервери отримують електронні листи, надіслані користувачеві, і зберігають їх до тих пір, поки клієнт не запитає їх від імені зазначеного користувача. Наявність сервера електронної пошти дозволяє на одній машині бути належним чином налаштованим і підключеним до мережі в будь-який час. Тоді він готовий надсилати й отримувати повідомлення, а не вимагати, щоб кожна клієнтська машина постійно працювала власною підсистемою електронної пошти.

Одним з найбільш поширених типів серверів на сучасному ринку є веб-сервер. Веб-сервер — це особливий вид сервера додатків, на якому розміщуються програми та дані, запитувані користувачами через Інтернет або інтранет. Веб-сервери відповідають на запити веб-сторінок або інших веб-сервісів від браузерів, запущених на клієнтських комп'ютерах. Поширені веб-сервери включають веб-сервери Apache, сервери Microsoft Internet Information Services (IIS) і сервери Nginx .

Обсяг даних, які використовують компанії, користувачі та інші служби, вражає. Значна частина цих даних зберігається в базах даних. Бази даних повинні бути доступні кільком клієнтам у будь-який момент часу і можуть вимагати надзвичайної кількості дискового простору. Обидві ці потреби добре підходять для розташування таких баз даних на серверах. Сервери баз даних запускають програми баз даних і відповідають на численні запити клієнтів. Поширені програми сервера баз даних включають Oracle, Microsoft SQL Server, DB2 та Informix.

Віртуальні сервери захоплюють світ серверів штурмом. На відміну від традиційних серверів, які встановлюються як операційна система на апаратному забезпеченні машини, віртуальні сервери існують лише у тому вигляді, який визначено у спеціалізованому програмному забезпеченні, яке називається гіпервізором. Кожен гіпервізор може одночасно запускати сотні або навіть тисячі віртуальних серверів. Гіпервізор представляє серверу віртуальне обладнання так, ніби це реальне фізичне обладнання. Віртуальний сервер використовує віртуальне обладнання, як зазвичай, а гіпервізор передає фактичні потреби в обчисленні та зберіганні на реальне обладнання під ним, яке використовується спільно всіма іншими віртуальними серверами.

Проксі-сервер виступає як посередник між клієнтом і сервером. Часто використовується для ізоляції клієнтів або серверів з метою безпеки, проксі-

сервер приймає запит від клієнта. Замість того, щоб відповідати клієнту, він передає запит на інший сервер або процес. Проксі-сервер отримує відповідь від другого сервера, а потім відповідає оригінальному клієнту так, ніби він відповідає самостійно. Таким чином, ні клієнту, ні серверу, що відповідає, не потрібно безпосередньо підключатися один до одного.

Деякі сервери існують для моніторингу та керування іншими системами та клієнтами. Існує багато типів серверів моніторингу. Деякі з них прослуховують мережу і отримують кожен запит клієнта і відповідь сервера, але деякі самі не запитують і не відповідають на дані. Таким чином, сервер моніторингу може відстежувати весь трафік в мережі, а також запити та відповіді клієнтів і серверів, не заважаючи цим операціям. Сервер моніторингу відповідатиме на запити від клієнтів моніторингу, наприклад тих, якими керують адміністратори мережі, які стежать за станом мережі.

Концепція серверів майже така ж стара, як і сама мережа. Зрештою, суть мережі полягає в тому, щоб дозволити одному комп'ютеру спілкуватися з іншим комп'ютером і розподіляти роботу або ресурси. З тих пір комп'ютери еволюціонували, в результаті чого з'явилися кілька типів серверних структур та обладнання.

Наступна велика хвиля серверів включала комп'ютерні сервери. Багато в чому ці сервери були не чим іншим, як більшими, потужнішими настільними комп'ютерами. Такі сервери, як правило, були дорогими і містили набагато більше пам'яті та дискового простору, ніж більшість клієнтських комп'ютерів. Кожен сервер все ще був автономним блоком із власною материнською платою, процесором, пам'яттю, дисководом та джерелом живлення. Подібні сервери часто зберігалися в кондиціонованих приміщеннях, які називаються серверними, а пізніше були закріплені на стелажах для кращого зберігання та доступності.

Блейд-сервери

Оригінальне обладнання комп'ютерного сервера було великим і зберігалось в стелажах, які могли вмістити сотні фунтів. З часом, однак, швидші засоби підключення обладнання призвели до того, що частини сервера витягувалися з одного автономного пристрою. Видаляючи жорсткі диски, виключаючи внутрішнє охолодження та продовжуючи мініатюризувати комп'ютерні частини, сервери в кінцевому підсумку були зведені до єдиного тонкого сервера, відомого як блейд-сервер. Незважаючи на те, що блейд-сервери все ще зберігаються в стійках у серверних кімнатах, блейд-сервери менші і їх легше замінити.

Об'єднання серверів

Ще до віртуалізації сервери вилучалися зі стандартної моделі односерверної операційної системи, встановленої на апаратній машині. Технології, такі як мережеве сховище, позбавили сервера від необхідності мати власне сховище. Інші технології, такі як дзеркальне відображення та кластеризація, дозволили об'єднати частини обладнання в більші та потужніші сервери. Такий сервер може складатися з кількох блейд-серверів, кількох підключених пристроїв зберігання даних та зовнішнього джерела живлення, і кожен елемент можна замінити іншим, поки сервер ще працює.

Віртуальні сервери

Віртуальні сервери все ще потребують апаратного забезпечення, але тепер воно запускає інший процес, відомий як гіпервізор. У деяких випадках, як-от Microsoft Hyper-V, повна операційна система продовжує працювати на самому апаратному забезпеченні. В інших випадках так звані «голі» гіпервізори можуть бути встановлені безпосередньо на серверне обладнання. В обох випадках саме апаратне забезпечення часто розповсюджується на масиві блейд-серверів, мережевих сховищ і джерел живлення, що призводить до середовища, де неможливо визначити, де закінчується окремий сервер і починається інший [12].

Можна стверджувати, що Windows для робочих груп була першою серверною операційною системою Microsoft. У цій версії певні комп'ютери можна було налаштувати на спільне використання ресурсів і відповіді на запити від клієнтів, що робило їх серверами за визначенням. Першою справжньою серверною операційною системою Microsoft була Windows NT. Її версії 3.5 і 3.51 працювали в багатьох бізнес-мережах, поки Microsoft не випустила свою лінійку Windows Server, яка продовжує існувати і сьогодні. Найновіша версія Windows Server — Windows Server 2016. Ця версія підтримує численні програми та бази даних, а також гіпервізор, який дозволяє використовувати віртуальні сервери.

Сервери Linux / Unix

Іншим основним гравцем серверних операційних систем є сфера Linux/Unix. Існує кілька версій і різновидів Linux/Unix, включаючи Red Hat Enterprise Linux, Debian і CentOS . Як операційна система з відкритим вихідним кодом, Linux дуже популярний як веб-сервер, часто з встановленим сервером веб-додатків Apache.

NetWare

Незважаючи на те, що NetWare більше не випускається, NetWare був головним гравцем у сфері серверного програмного забезпечення, оскільки ера клієнт-сервер наростала. Згодом NetWare перемістила свою серверну операційну систему на ядро на базі Linux і назвала її Novell Open Enterprise Server (OES).

Хмарні сервери

Віртуальні сервери, розміщені на сторонній інфраструктурі у відкритій мережі, наприклад в Інтернеті, називаються хмарними серверами. Сьогодні існує безліч постачальників хмарних серверів, зокрема Cloud Platform від Google, Microsoft Azure та IBM Cloud. Однак головним піонером корпоративних хмарних обчислень стала платформа AWS від Amazon. Спочатку він почав використовувати вільні потужності власних

серверів і мереж Amazon, але тепер AWS дозволяє клієнтам створювати віртуальний сервер майже миттєво, а потім регулювати кількість ресурсів, які сервер може використовувати на льоту. Сьогодні сервер може бути не що інше, як дані фізичного обладнання, яке складається з кількох процесорів, дисководів, пам'яті та мережевих з'єднань. Але навіть зараз сервер — це лише система, яка відповідає на запит клієнта.

1.3 Еволюція служб каталогів

Служби каталогів у тій чи іншій формі існували від початку епохи комп'ютерів — вони призначалися для пошуку файлів й у виробничих мережевих реалізаціях. Служба каталогів надає детальну інформацію про користувачів та об'єкти мережі, приблизно так, як телефонна книга дозволяє знайти номер телефону за відомим прізвищем. Наприклад, об'єкт користувача у службі каталогів може містити номер телефону, адресу електронної пошти, назву підрозділу та ще стільки інших атрибутів, скільки забажає системний адміністратор.

Служби каталогів часто називають "білими сторінками" мережі. Вони забезпечують визначення та адміністрування користувачів та об'єктів.

Перші електронні каталоги були створені невдовзі після винаходу цифрових комп'ютерів та застосовувалися для аутентифікації користувачів та управління доступом до ресурсів. З розширенням мережі Інтернет та збільшенням спільного використання комп'ютерів до функцій каталогів було включено зберігання основної контактної інформації про користувачів. Прикладами ранніх каталогів можуть бути MVS PROFS (IBM), база реєстраційних даних Grapevine і WHOIS [13].

Незабаром з'явилися спеціалізовані служби каталогів для програм, призначені для спеціальної адресації, пошуку та введення контактної інформації

для кожного програмного продукту. Доступ до таких каталогів був можливий лише за допомогою спеціальних методів, а сфера їх застосування була обмеженою. Додатками, що використовують ці типи каталогів, були такі програми, як Novell GroupWise, Lotus Notes та файл /etc/aliases утиліти sendmail у UNIX [14].

Подальший розвиток великомасштабних служб каталогів для підприємств очолила компанія Novell, випустивши на початку 90-х років минулого століття службу каталогів Novell (Novell Directory Services - NDS). Вона була прийнята організаціями NetWare, а потім розширена за рахунок включення підтримки змішаних середовищ NetWare/NT. Громіздка лінійна структура доменів NT та недостатня синхронізація та взаємодія цих двох середовищ змусила багато організацій перейти на використання NDS як реалізацію служби каталогів. Саме ці недоліки NT були основною причиною випуску служби Microsoft AD DS [15].

Розробка полегшеного протоколу доступу до каталогів (Lightweight Directory Access Protocol — LDAP) була викликана розширенням Інтернету та необхідністю тісної взаємодії та суворої стандартизації. Цей загальноприйнятий метод доступу до інформації каталогів та її модифікації, що користується всіма можливостями протоколу TCP/IP, виявився надійним та функціональним, і для його застосування було розроблено нові реалізації служб каталогів. Сама служба AD DS розроблялася так, щоб відповідати стандарту LDAP [16].

У системі Exchange Server 5.5 існувала власна служба каталогів, яка запускала як частина середовища обміну електронними повідомленнями. Насправді багато ключових компонентів AD DS були запозичені саме з цієї початкової служби каталогів Exchange. Наприклад, у базі даних AD DS використовується той самий формат бази даних Jet, що й у Exchange 5.5, та й топологія реплікації сайтів багато в чому схожа [17].

Власні служби каталогів існували і в інших програмах Microsoft, наприклад, в Internet Information Server (Сервер інформації Інтернету) і Site Server (Сервер сайтів). Однак кожна з цих служб каталогів ніяк не співвідносилася з іншими, до того ж рівень інтеграції між різними реалізаціями був не дуже високим.

Центральну роль функціональності AD DS грають п'ять ключових компонентів. Вимоги сумісності нових служб каталогів зі стандартами Інтернету призвели до того, що в існуючі реалізації були внесені відповідні зміни та приділено більше уваги наведеним нижче областям.

Протокол LDAP був розроблений як стандартний Інтернет-протокол для доступу до каталогів. Він застосовується при оновленні та запитах даних, що зберігаються в каталогах. У AD DS протокол LDAP безпосередньо підтримується.

Підтримка доменних імен. Система доменних імен (Domain Name System - DNS) була створена для задоволення потреби в перетворенні спрощених імен, зрозумілих людям (таких як www.*.com), в IP-адреси, зрозумілі комп'ютерам (на кшталт 12.155.166.151). В AD DS вона не просто підтримується, а навіть потрібна для нормальної роботи.

Підтримка безпеки відповідно до стандартів Інтернету надзвичайно важлива для безперебійного функціонування середовища, яке, по суті, підключається до мільйонів комп'ютерів по всьому світу. Недостача надійних засобів захисту є своєрідним запрошенням для хакерів, тому в Windows Server 2008 R2 та AD DS можливості для забезпечення безпеки були значно розширені. Так, у Windows Server 2008 R2 і AD DS була вбудована безпосередня підтримка для IPSec, Kerberos, центрів сертифікації та шифрування за допомогою протоколу захищених сокетів (Secure Sockets Layer - SSL).

Зручне адміністрування. Хоча при реалізації потужних служб каталогів зручності адміністрування та конфігурування середовища часто не приділяється належної уваги, цей аспект дуже впливає на загальну вартість експлуатації

середовищ. В AD DS та Windows Server 2008 R2 було спеціально все продумано так, щоб ними було зручно користуватися, і щоб на освоєння нового середовища витрачалось якнайменше зусиль. У Windows Server 2008 R2 покращено можливості для адміністрування AD DS за рахунок додавання компонента Active Directory Administration Center (Центр адміністрування Active Directory), компонента Active Directory Web Services (Веб-служби Active Directory) та модуля для адміністрування Active Directory з оболонки Windows PowerShell .

AD DS вперше з'явилася в Windows 2000 Server як заміна для доменів Windows NT 4.0 (тоді називалася просто AD), технологія AD DS пізніше була значно покращена у Windows Server 2003 та Windows Server 2003 R2 Edition . Вона дуже швидко отримала широке визнання в промислових колах і зарекомендувала себе як надійна, масштабована і високопродуктивна система. Поява AD DS дозволила позбавитися деяких обмежень, властивих схемам з доменами NT 4.0, а також забезпечити можливість інтеграції майбутніх продуктів виробництва Microsoft та інших виробників в один спільний інтерфейс.

З виходом Windows Server 2008 R2 готовність середовища Microsoft до взаємодії з Інтернетом досягає нових рівнів функціональності завдяки доданню наступних поліпшень: відновлення віддалених об'єктів за допомогою кошика Active Directory (Active Directory Recycle Bin); приєднання до домену в автономному режимі (Offline Domain Join); застосування керованих облікових записів служб (Managed Service Accounts);

Висновки до розділу 1.

Розробляючи Windows Server 2000/2003, а потім і Windows Server 2008 R2, у Microsoft завжди прагнули до того, щоб усі їхні програмні продукти підтримували стандарти Інтернету. Ті стандарти, які раніше пропонувалися як додаткові або взагалі не підтримувалися, поступово влітали в програмне забезпечення і ставали основними методами для передачі та обробки даних. В результаті всі

програми та операційні системи перетворювалися на сумісні з TCP/IP, а спеціалізовані протоколи, такі як NetBEUI, видалялися. Подальше прийняття TCP/IP як стандартний Інтернет-протокол для обміну даними зробило його одним з лідерів у світі протоколів і, по суті, перетворило на обов'язковий протокол для операційних систем рівня підприємства.

РОЗДІЛ 2 СИСТЕМА DNS

2.1 Загальні поняття

Система DNS відповідає за дозвіл числових IP-адрес у символічні доменні імена [18]. Наприклад, сайту `www.google.com` відповідає IP-адреса `172.217.22.14`. У браузері можна ввести або символічне ім'я (`www.google.com`) або цю IP-адресу. Машинам простіше працювати з числами, людині – із символічними іменами.

Розглянемо процес дозволу доменного імені в IP-адресу. Коли ви вводите адресу вузла в браузер, браузер звертається до резолвера - це частина операційної системи, яка відповідає за дозвіл на доменні імена. Якщо доменне ім'я є в локальному кеші резолвера, він одразу повертає його браузеру. Браузер, отримавши IP-адресу вузла, надсилає йому запит, наприклад, `GET/` з метою отримання кореневої сторінки (яка зазвичай називається `index.*`).

Якщо в кеші резолвера немає доменного імені, тоді він звертається до DNS-сервера, який вказаний в налаштуваннях мережі або отриманий від DHCP-сервера під час автоматичного налаштування мережевого інтерфейсу. Як правило, це DNS сервер провайдера. Якщо в ньому буде потрібне доменне ім'я, він відправляє відповідну йому IP-адресу резолверу. Якщо ж ні, тоді DNS-сервер звертається до DNS-сервера кореневого домену `.com`. Швидше за все, в його кеші буде потрібна IP-адреса, якщо ж ні, тоді буде звернено до DNS-сервера домену `google.com` з метою отримати IP-адресу вузла `www`. Отримана IP-адреса буде по ланцюжку повернена вузлу, який запросив дозвіл доменного імені.

Схема дозволу доменного імені є рекурсивною, а наш запит – рекурсивним.

Існують різні види DNS-серверів. Навіть якщо у організації немає свого домену або віддається перевага, щоб делегуванням домену займався реєстратор або Інтернет-провайдер, все одно можна встановити кешуючий DNS-сервер. При цьому рекурсивними запитами займатимуться DNS-сервери провайдера, а серверу

потрібно лише кешувати результати запитів – так прискориться швидкість дозволу доменних імен – сторінки почнуть відкриватися швидше. Навіщо потрібен кеш DNS-сервер, якщо в системі є кеш резолвера? У кеші резолвера знаходяться лише ті імена, до яких ви зверталися. У кеші DNS-сервера, що кешує, знаходяться всі імена, до яких зверталися всі користувачі вашої мережі. Отже, чим більше користувачів у вашій мережі, тим більшою буде ефективність кешуючого DNS-сервера [19].

Є і звичайний сервер DNS - він зберігає інформацію про вашу доменну зону. Також він називається первинним DNS-сервером. На допомогу первинному налаштовують вторинний DNS-сервер - він оброблятиме DNS-запити, коли первинному серверу стало «погано». Далі в цьому розділі буде показано, як налаштувати вторинний DNS-сервер.

Налаштування DNS-сервера почнемо з найпростішого варіанта – з кешуючого DNS-сервера.

Основна концепція того, як DNS виконує свою роботу, досить проста: кожна адреса веб-сайту, введена у веб-браузер (наприклад, Chrome, Safari або Firefox), надсилається на DNS-сервер, який розуміє, як зіставити це ім'я на належну IP-адресу. Це IP-адреса, яку пристрої використовують для взаємодії один з одним, оскільки вони не можуть передавати і не передають інформацію за допомогою імені типу www.google.com, www.youtube.com і т. д. Ми просто вводимо просту назву цих веб-сайтів, тоді як DNS робить для нас усі пошуки, надаючи нам майже миттєвий доступ до належних IP-адрес, необхідних для відкриття потрібних нам сторінок. Знову ж таки, www.microsoft.com, www.tebapit.com, www.amazon.com, і будь-яка інша назва веб-сайту використовується лише для нашої зручності, оскільки набагато легше запам'ятати ці імена, ніж запам'ятати їх IP-адреси. Кореневі сервери відповідають за збереження IP-адрес для кожного домену верхнього рівня. Коли запитується веб-сайт, це кореневий сервер, який спочатку

обробляє цю інформацію, щоб визначити наступний крок у процесі пошуку. Потім доменне ім'я пересилається на вирішувач доменних імен (DNR), який знаходиться в межах Інтернет-провайдера, щоб визначити правильну IP-адресу. Нарешті, ця інформація надсилається назад на пристрій, у якого її вимагали.

Операційні системи, такі як Windows та інші, зберігатимуть IP-адреси та іншу інформацію про імена хостів локально, щоб отримати до них доступ швидше, ніж постійно звертаючись до DNS-сервера. Коли комп'ютер розуміє, що певне ім'я хосту є синонімом певної IP-адреси, цю інформацію можна зберігати або кешувати на пристрої. Запам'ятовувати інформацію про DNS корисно, але іноді вона може бути пошкодженою або застарілою. Зазвичай операційна система видаляє ці дані через певний проміжок часу, але якщо виникають проблеми з доступом до веб-сайту, і є підозра, що це пов'язано з проблемою DNS, першим кроком є примусове видалення цієї інформації, щоб звільнити місце для нових, оновлених записи DNS. Можна просто перезавантажити комп'ютер, якщо виникли проблеми з DNS, оскільки кеш DNS не зберігається під час перезавантаження. Однак очищення кешу вручну замість перезавантаження відбувається набагато швидше. Можна очистити DNS в Windows через командний рядок за допомогою `ipconfig / flushDNS` команди. Важливо пам'ятати, що залежно від того, як налаштований конкретний маршрутизатор, записи DNS також можуть зберігатися там. Якщо очищення кешу DNS на комп'ютері не вирішує проблему з DNS, слід спробувати перезапустити маршрутизатор, щоб очистити цей кеш DNS. Записи у файлі `hosts` не видаляються, коли кеш DNS очищається.

Враховуючи, що DNS відповідає за спрямування імен хостів на певні IP-адреси, очевидно, що це головна мета зловмисної діяльності. Хакери можуть перенаправити запит з нормально функціонуючого ресурсу на той, який є пасткою для збору паролів або обслуговування шкідливих програм.

Отруєння DNS і підробка DNS – це терміни, що використовуються для опису атаки на кеш-пам'ять розв'язувача DNS з метою перенаправлення імені хосту на іншу IP-адресу, відмінну від тієї, що істинно присвоєна цьому імені хосту, ефективно переспрямовуючи туди, куди ви збиралися перейти. Зазвичай це робиться для того, щоб перейти на веб-сайт, який заповнений шкідливими файлами, або виконати фішинг-атаку, щоб змусити отримати доступ до схожого веб-сайту, щоб викрасти облікові дані для входу. Більшість служб DNS забезпечують захист від таких типів атак. Ще один спосіб для зловмисників вплинути на записи DNS – це використання файлу hosts. Файл hosts – це локально збережений файл, який використовувався замість DNS до того, як DNS насправді став широко розповсюдженим інструментом для розпізнавання імен хостів, але файл все ще існує у популярних операційних системах. Записи, що зберігаються у цьому файлі, замінюють налаштування сервера DNS, тому це загальна мета для зловмисного програмного забезпечення. Простий спосіб захистити файл hosts від редагування – позначити його як файл лише для читання. У Windows просто перейдіть до папки, в якій є файл hosts: % Systemdrive% Windows System32 drivers etc Клацніть правою кнопкою миші або натисніть та утримуйте, виберіть Властивості, а потім поставте галочку в полі біля Лише для читання атрибут.

Інтернет-провайдер, який зараз обслуговує доступ до Інтернету, призначив DNS-сервери для використання пристроями (якщо є підключення до DHCP). Інші сервери можуть надавати функції реєстрації для відстеження відвідуваних веб-сайтів, блокування реклами, фільтри для дорослих веб-сайтів та інші функції. Незалежно від того, використовує комп'ютер DHCP для отримання IP-адреси, чи використовує статичну IP-адресу, все одно можна визначити власні DNS-сервери. Явні налаштування сервера DNS мають перевагу над неявними параметрами зверху вниз. Іншими словами, це налаштування DNS, найближчі до пристрою, який використовує пристрій. Наприклад, якщо змінили налаштування DNS-

сервера на своєму маршрутизаторі на щось конкретне, тоді всі пристрої, підключені до згаданого маршрутизатора, також використовуватимуть ці DNS-сервери. Однак якщо потім буде змінено налаштування DNS-сервера на ПК на щось інший, цей комп'ютер використовуватиме DNS-сервери, що відрізняються від усіх інших пристроїв, підключених до одного маршрутизатора. Це є причиною того, що пошкоджений кеш DNS на комп'ютері може перешкоджати завантаженню веб-сайтів, навіть якщо ті самі зазвичай відкриваються на іншому комп'ютері в одній мережі.

2.2 Налаштування кешуючого DNS-сервера unbound

Для налаштування кешуючого DNS-сервера раніше було прийнято використовувати BIND9 - той самий пакет, який використовується для налаштування повноцінного DNS-сервера. Але зараз замість нього прийнято використати пакет unbound [20].

Сервер Unbound розповсюджується під ліцензією BSD, має модульну структуру і може працювати, як у рекурсивному, так і режимі кеші. Ми ж будемо використовувати Unbound суто в режимі кеш.

Основний конфігураційний файл називається `/etc/unbound/unbound.conf`. За замовчуванням він практично порожній, а повний приклад з усіма можливими опціями можна знайти у каталозі `/usr/share/doc/unbound/examples`.

У лістингу представлений листинг `/etc/unbound/unbound.conf`, що настроює Unbound працювати в кешуючому режимі.

Лістинг. Файл `/etc/unbound/unbound.conf`

```
server:
```

```
# Порт, на якому наш сервер «слухатиме» запити port: 53
```

Описуємо інтерфейси, на яких ми слухатимемо запити # 192.168.1.1 - сервер нашої локальної мережі, на якому встановлено Unbound interface: 127.0.0.1 interface: 192.168.1.1

Вихідний інтерфейс (WAN)

outgoing-interface: xxx.xx.xx.xx

Мережа, якій дозволено доступ до нашого сервера access-control: 192.168.1.0/24 allow

Дозволяємо IPv4 TCP/UDP, забороняємо IPv6 do-ip4: yes do-ip6: do-udp: yes do-tcp: yes

Користувач, від імені якого запускатиметься сервер username: unbound

Вказуємо файл журналу та відключаємо використання syslog logfile: "unbound. log" use-syslog: no

Шлях до PID-файлу

pidfile: "/var/run/local_unbound.pid"

Приховуємо версію софту hide-version: yes

Рівень журналювання – 0 (тільки помилки) verbosity: 0

Наступний рядок настроює Unbound на здійснення криптографічної

валідації DNSSEC, використовуючи кореневий ключ auto-trust-anchor-file: "/var/lib/unbound/root.key"

Тепер перевіримо конфігурацію сервера:

unbound-checkconf

Якщо помилок немає, ви отримаєте повідомлення:

unbound-checkconf: no errors в /etc/unbound/unbound.conf

Залишилося лише перезапустити Unbound:

service unbound restart

Залишилося налаштувати DHCP-сервер, щоб він повідомляв усім локальним вузлам нову IP-адресу DNS-сервера. У нашому випадку – це 192.168.1.1.

Все, що нам потрібно зробити - це додати IP-адреси DNS-сервера провайдера до блоку експедиторів. Вони будуть виконувати роботу розв'язання доменних імен, а наш сервер буде кешувати тільки результати запиту.

Перед блоком `forwarders` можна вказати параметр `forward`, який може набувати значення `only` або `first`. У першому випадку наш сервер взагалі не намагатиметься обробити запит самостійно. У другому випадку сервер спробує обробити запит самостійно, якщо не отримає відповіді від серверів, описаних у блоці `forwarders`. Друге значення більш переважно:

```
forward first;
```

```
forwarders {
```

```
8.8.8.8;
```

```
8.8.8.4;
```

Перезапускаємо сервер:

```
# перезапуск служби bind9
```

Перегляньте файл журналу:

```
# tail /var/log/daemon.log
```

Ви повинні побачити повідомлення, що сервер запущений як:

```
Nov 29 09:41:42 debian named [6921]: running
```

Перевіримо, чи працює наш DNS-сервер. У `/etc/resolv.conf` на DNS-сервері додайте рядок:

```
Сервер імен 127.0.0.1
```

Якщо ви з якихось причин не відключили `NetworkManager`, він при наступному перезавантаженні перезапише цей файл. Зрозуміло, що `NetworkManager` отримає IP-адресу DNS-сервера від DHCP-сервера, але поки ви ще не налаштували DHCP-сервер, тоді можете заборонити зміну файлу `/etc/resolv.conf`:

```
# chattr +i /etc/resolv.conf
```

Потім перезавантажте мережу або комп'ютер. Після цього введіть команду:

```
ns lookup google.com
```

Висновок буде таким:

```
Server:127.0.0.1
```

```
Address: 127.0.0.1153
```

```
Non-authoritative answer:
```

```
Name: google.com
```

```
Address: 172.217.22.15
```

```
Name: google.com
```

```
Address: 172.217.22.16
```

```
Name: google.com
```

```
Address: 94.100.100.201
```

```
Name: google.com
```

```
Address: 94.100.180.200
```

Відповідь надійшла від нашого сервера 127.0.0.1. Ми переконалися, що наш DNS-сервер працює, отже, можна налаштувати DHCP-сервер, щоб він «роздавав» усім нашим клієнтам IP-адресу щойно налаштованого DNS-сервера. Звичайно, прописувати в налаштуваннях DHCP-сервера потрібно не IP-адреса 127.0.0.1, а IP-адреса сервера, яку ви можете отримати командою `ifconfig`, запущеною на DNS-сервері.

Люди часто отримують помилки DNS, коли підключення до Інтернету встановлено неправильно. Обов'язково перевірте кожен маршрутизатор, модеми чи дроти, якими ви користуєтесь, і переконайтеся, що вони у робочому стані та належним чином підключені до відповідних портів.

Якщо все виглядає добре, відкрийте панель управління. Опинившись там, виберіть „Мережа та Інтернет”, а на наступному екрані виберіть „Центр мереж та спільного доступу”. У центрі мережі та обміну даними ви зможете побачити

детальну інформацію про підключення до Інтернету. Обов'язково перевірте, чи вказано тип доступу „Інтернет” або „Місцевий та Інтернет”. Якщо все гаразд, перейдіть до наступного кроку, і якщо ви виявите щось інше, ніж те, що ми згадали, зверніться до свого постачальника послуг Інтернету (ISP).

Запуск діагностики мережі Windows

Починаючи з центру мережі та діагностики, виберіть „Вирішення проблем”. Для старих версій Windows це має відкрити сторінку усунення несправностей і дасть вам можливість вибрати, з якими проблемами ви стикаєтесь. Запустіть засіб усунення несправностей.

Клацніть правою кнопкою миші на меню запуску Windows 10, виберіть налаштування,

Далі натисніть Мережа та Інтернет,

Прокрутіть вниз і знайдіть засіб усунення несправностей мережі, натисніть на нього,

Дотримуйтесь інструкцій на екрані, щоб розпочати процес діагностики.

Засіб усунення несправностей мережі.

Оновлення драйверів мережевих адаптерів

Ми почнемо вирішувати цю проблему шляхом оновлення драйверів мережевих адаптерів.

Натисніть клавішу Windows + X, виберіть диспетчер пристроїв.

Шукайте мережеві адаптери в довгому списку, наданому менеджером пристроїв.

Клацніть правою кнопкою миші на мережевому адаптері та виберіть «Видалити». Це видалить поточний драйвер мережевого адаптера.

У верхній частині екрана в диспетчері пристроїв натисніть Дія та натисніть Сканувати на зміну обладнання. Клацніть на ньому.

Тепер знову знайдіть мережевий адаптер і клацніть правою кнопкою миші.

Виберіть «Оновити програмне забезпечення драйвера».

Коли з'явиться відповідний запит, виберіть „Автоматично шукати оновлене програмне забезпечення драйверів“.

Тепер Windows встановить відповідні та оновлені драйвери для вашого пристрою.

Після завершення встановлення перезапустіть пристрій.

Перевірка наявності шкідливих програм та надмірно активних антивірусів

У деяких рідкісних випадках проблеми з DNS можуть бути спричинені шкідливим програмним забезпеченням, яке може потрапити у вашу систему через Інтернет або через пошкоджений файл, який ви вручну запустили у своїй системі.

У наш час антивіруси не лише захищають від вірусів у нашій системі, але також захищають від шкідливих веб-сайтів. Зараз інше може бути проблемою, оскільки антивіруси, як правило, блокують з'єднання з Інтернетом, як тільки виявляють щось підозріле. Часто деякі оновлення програм, що відбуваються у фоновому режимі вашого пристрою, можуть викликати помилкову тривогу для антивіруса, і це може примусово відключити всі підключення до Інтернету.

Цю проблему також може спричинити непрацездатне програмне забезпечення третьої сторони.

Натисніть клавішу Windows + R.

Напишіть «msconfig» і натисніть Enter.

Відкриється нове вікно під назвою «Конфігурація системи».

Поставте прапорець біля пункту 'сховати всі служби Microsoft' унизу та натисніть 'Вимкнути всі'.

Натисніть Застосувати, щоб зберегти ці варіанти, і натисніть ОК, щоб закрити вікно.

Перезавантажте ПК.

Якщо після перезапуску ви не зіткнулися з проблемою, ви можете бути впевнені, що це спричинено сторонніми програмами.

Повторіть процедуру і дозвольте лише 1 сторонній програмі одночасно ідентифікувати винуватця.

Якщо вам вдалося ідентифікувати програму, негайно видаліть її.

DNS-сервери є невід'ємною частиною сучасних мережесистем, і їх складність є причиною частих помилок. Але безсумнівно, що більшість найпоширеніших питань можна вирішити за допомогою вищезазначених кроків. Якщо після випробування всіх цих рішень ви все ще стикаєтеся з тією ж проблемою, не соромтеся звертатися до виробника ПК за допомогою та обслуговуванням.

2.3 Налаштування повноцінного DNS-сервера

Налаштування повноцінного DNS-сервера складніше за налаштування кешуючого DNS-сервера. Адже нам доведеться налаштувати одну чи кілька зон, які обслуговуватиме наш DNS-сервер. Домен – це не зона. Зона – це частина домену, яка керується певним DNS-сервером. Що стосується невеликих доменів, то зона - домен. Але іноді буває так, що частина піддоменів одного домену обслуговується одним сервером DNS, а інша частина - іншим сервером DNS. Так ось ця частина, яка обслуговується певним DNS-сервером, є зона.

Припустимо, що ми маємо домен `example.com`, який представляє мережу `192.168.1.0`. У конфігурацію BIND потрібно додати такі рядки:

```
zone "example.com" {
type master;
файл "example.com";
notify no;
b-
```

```
zone "0.1.168.192.in-addr.arpa" {
type master;
file "192.168.1.0";
notify yes;
}
```

Дані рядки потрібно додати або у файл `/etc/bind/named.conf.local` або прямо у файл `/etc/bind/named.conf` після тих рядків, які вже є.

Файл `example.com`, описаний у першому блоці `zone`, містить конфігурацію прямого перетворення, тобто використовується для перетворення доменних імен на IP-адресу.

Лістинг. Файл `/etc/bind/example.com`

```
0 IN SOA ns.example.com. admin.example.com. (1 ; серійний номер
```

```
60480 ; оновлення кожні 60480 секунд
```

```
86400 ; повтор кожні 86400 секунд
```

```
2419200; час зберігання інформації – 672 години 86400 ; TTL записи)
```

```
IN NS ns.example.com.
```

```
IN A 192.168.1.1
```

```
IN MX 100 mail.example.com.
```

```
www IN CNAME ns.example.com.
```

```
mail IN A 192.168.1.3
```

```
ftp IN A 192.168.1.2
```

```
локальний. IN A 127.0.0.1
```

Якщо в кінці доменного імені ставиться крапка, то сервер не дописуватиме ім'я домену `example.com` до імені. Якщо ім'я вказано без крапки, до нього буде дописано ім'я домену `example.com`.

Запис `SOA` описує початок мандата. Перша назва після того, як `SOA` є ім'я цього комп'ютера `ns.example.com`. Тоді адреса електронної пошти адміністратора

сервера вказується. Символ @ зарезервований, тому перша точка вважається @. Виявляється, що адреса адміністратора є admin@example.com. Інша частина запису SOA закоментований в перерахуванні.

Запис NS визначає ім'я DNS-сервера імен, у нашому випадку це ns.example.com, запис A задає його IP-адресу.

Запис MX визначає адресу та пріоритет (100) поштового сервера. Ви можете мати кілька поштових серверів, тоді ви можете вказати кілька записів MX з різним пріоритетом. Чим нижче значення пріоритету, тим вище пріоритет сервера, наприклад:

```
IN MX 100 mail.example.com
```

```
IN MX 200 mail2.example.com.
```

Далі записи CNAME створюють псевдонім для імені www. Це означає, що веб-сервер теж запущений на цьому комп'ютері і коли хтось вкаже ім'я www.example.com, запит прийде до вузла з IP-адресою 192.168.1.1.

Запис A використовується для перекладу доменного імені до відповідного IP-адреса. Ми встановлюємо IP-адреси для комп'ютерів імені пошти, FTP і локальну.

Інші комп'ютери, як правило, є звичайними робочими станціями та призначенням IP-адрес ним займається DHCP-сервер. Зазвичай не потрібно дозволяти імена цих комп'ютерів в IP-адреси, оскільки жодних з'єднань із ними встановлювати не планується.

Тепер розглянемо файл зворотного перетворення – /etc/bind9/192.168.1.0

Лістинг. Файл /etc/bind 9/192.168.1.0

```
0 IN SOA ns.example.com.admin.example.com. (
```

```
1 ; серійний номер
```

```
60480 ; оновлювати кожні 60480 секунд
```

```
86400 ; повтор кожні 86400 секунд
```

```
2419200; Час зберігання інформації - 672 годин
```

86400 ; TTL записи

)

0 B NS ns.example.com

1 IN PTR ns.example.com

2 IN PTR ftp.example.com

3 .IN PTR mail.example.com

SGENERATE 5-104 \$ PTR ф-192-168-1. - \$ Приклад .com

З лістингу видно, що IP-адреса 192.168.1.1 належить вузлу ns.example.com, 192.168.1.2 – вузлу ftp.example.com та адресу 192.168.1.3 – вузлу mail.example.com. Останній запис не обов'язковий. Він каже, що вузли IP-адресами від 192.168.1.5 до 192.168.1.104 будуть відповідати іменам ip-192-168-IN-example.com, де N – останнє число IP-адреси. Цей запис потрібний лише, якщо ви дбаєте про перетворення IP-адрес, виданих вашим DHCP-сервером, на доменні імена.

У цьому файлі ви можете не вказувати IP-адреси повністю, але якщо ви це робите, їх потрібно вказувати в зворотному порядку, наприклад:

1.1.168.192 IN PTR ns.example.com

Крапки наприкінці доменного імені також не потрібні. Для налаштування віддаленого керування сервером необхідно підготувати блоки key та controls. Найпростіше це зробити за допомогою команди:

```
# /usr/sbin/rndc-confgen>remote.conf.
```

2.4 Налаштування вторинного DNS-сервера

У великих мережах важливо налаштувати вторинний DNS-сервер, який буде обслуговувати запити клієнтів у разі відмови первинного сервера [21].

Вторинний сервер налаштовується, як і первинний, тільки тип зони задається як підлегла (slave), а в блоці masters вказуються первинні DNS-сервери (у нашому випадку тільки один):

```
файл "example.com";
masters {192.168.1.1;};
};
```

На первинному сервері в блоці options потрібно додати блок allow-transfer, в якому вказують IP-адресу вторинного DNS-сервера:

```
options (
allow-transfer {192.168.1.2;
```

При виникненні проблем із відкриттям сайтів, таких як ERR_NAME_NOT_RESOLVED та багатьох інших одна з перших рекомендованих дій змінити DNS сервер у властивостях підключення Windows. Використання іншого сервера може працювати швидше (і точніше), ніж DNS сервер вашого провайдера, прискорюючи тим самим завантаження сторінок. Також деякі сервери надають функції фільтрації небажаного трафіку.

Натисніть клавіші Win+R на клавіатурі, введіть ncpa.cpl у вікно "Виконати" та натисніть Enter.

У вікні, натисніть правою кнопкою миші по підключенню, що використовується для доступу в Інтернет і виберіть пункт «Властивості» в контекстному меню .

У списку компонентів підключення виберіть «IP версії 4» або «TCP/IPv4» і натисніть кнопку «Властивості» . Відкрити властивості протоколу IP версії 4

наступні адреси DNS-серверів» і вкажіть потрібні адреси .

Натисніть кнопку Ок.

За необхідності (зазвичай така відсутня) змініть аналогічно DNS для IP версії 6.

Після зміни налаштувань DNS не обов'язково, але бажано скинути кеш DNS.

На цьому процес буде завершено, а при відкритті сайтів в Інтернеті у вас буде використовуватися вказана вами адреса DNS-сервера.

Ще один спосіб змінити DNS сервер у Windows 10.

У Windows 10 є додатковий метод зміни DNS-сервера для підключення до Інтернету:

Зайдіть до Установки — Мережа та Інтернет, ліворуч виберіть тип підключення (Ethernet, Wi-Fi), для якого потрібно змінити.

Натисніть на ім'я активної мережі Властивості мережі у параметрах Windows 10. Перейдіть до наступної сторінки вниз до розділу «Параметри IP» та натисніть кнопку «Редагувати» .Редагувати параметри IP

Замість "Автоматично" встановіть "Вручну".

Увімкніть IPv4, перейдіть вниз і встановіть бажані параметри бажаного та додаткового DNS сервера, збережіть налаштування.

За потреби задайте DNS для IPv6 (зазвичай не потрібно).

Так само, як і в попередньому випадку, бажано очистити кеш DNS після застосування налаштувань.

Адреси популярних DNS-серверів

Багато відомих Інтернет-компаній надають доступ до своїх DNS-серверів: ви можете ввести їх у параметрах, і вони будуть використовуватися вашим підключенням. Серед популярних DNS-серверів:

Google – 8.8.8.8 та 8.8.4.4 (для IP версії 4), 2001:4860:4860::8888 та 2001:4860:4860::8844 (IP версії 6).

Cloudflare — 1.1.1.1 та 1.0.0.1 (IPv4), 2606:4700:4700::1111 та 2606:4700:4700::1001 (IPv6).

2.5 DNS server BIND на Linux

Named - це демон, що входить до складу пакету bind9 і є сервером доменних імен. Демон named може реалізовувати функції серверів будь-якого типу: master, slave, cache. Бінарний файл, який виконує основну роботу, розташований в / usr /

`sbin / named`. Він бере налаштування з основного конфігураційного файлу, який називається `named.conf` і розташований в каталозі `/ etc / bind`. В основному конфігураційному файлі описується робочий каталог сервера, найчастіше це каталог `/ var / cache / bind`, в якому лежать файли опису зон та інші службові файли. Відповідність назви зони і файлу опису зони задає розділ `zone` з параметром `file`. Розділ `zone` так само задає тип відповідальності даного сервера за зону (`master`, `slave` і ін.), А так само визначає особливі параметри для поточної зони (наприклад, на якому інтерфейсі обробляти запити для поточної зони). У файлах опису зон містяться параметри зон і записи ресурсів (шляху, зазначені в цьому абзаці можуть відрізнятися, це залежить від дистрибутива Linux або параметрів збірки сервера з початкових кодів).

Формат файлу конфігурації для 4-ої версії програми відрізняється від того, який застосовується в восьмою і дев'ятою версіях BIND.

Вихідні дані

Для коректної роботи DNS ньому необхідно мати налаштовану мережу. DNS налаштований на дистрибутиві Debian, особливості інших дистрибутивів теж будуть відзначені. Конфігураційний файл мережі стану наступний:

```
DNS: ~ # cat / etc / network / interfaces auto lo iface lo inet loopback auto eth0
iface eth0 inet static address 10.0.0.152 netmask 255.255.255.0 gateway 10.0.0.254 auto
eth1 iface eth1 inet static address 192.168.1.1 netmask 255.255.255.0
```

де `10.0.0.152/24` - зовнішній інтерфейс (підмережа, виділена провайдером), `192.168.1.1/24` - внутрішній (Локальна мережа). Налаштовувана зона матиме ім'я `example.com`. У прикладі зі `slave` сервером, вторинний сервер буде розташований на IP `10.0.0.191`.

Установка BIND9

Для роботи DNS сервера необхідно встановити пакет `bind9` (в деяких дистрибутивах - `bind`). Як зазначено - основним конфігураційним файлом BIND є

файл `named.conf` (даний файл може бути розміщений в каталозі / `etc`, іноді в / `etc` / `bind`).

Параметри (синтаксис) `named.conf`

Синтаксис файлу `named.conf` дотримується наступних правил:

IP-адреси - список IP повинен бути розділений символом ";" , Можливо вказувати підмережа в форматі `192.168.1.1/24` або `192.168.1.1/255.255.255.0`, (для виключення IP перед ним потрібно поставити знак!), Може свідчити імена "any", "none", "localhost" в подвійних лапках.

Коментарі - рядки починаються на #, // і укладені в / * і * / вважаються коментарями.

У файлах опису зон - символ @ є "змінної" зберігає ім'я зони, зазначеної в файлі конфігурації `named.conf` або в директиві @ \$ ORIGIN поточного опису зони.

Кожна завершена рядок параметрів повинна завершуватися символом;

розділ Acl

Acl (access control list) - дозволяє задати іменованний список мереж. Формат розділу: `acl "ім'я_сеті" {ip; ip; ip; };`

розділ Options

Розділ Options задає глобальні параметри конфігураційного файлу, керівники всіма зонами. Даний розділ має формат: `options {оператори розділу_Options};` Options може бути "вкладений" в розділ Zone, при цьому він перевизначає глобальні параметри. Часто використовувані оператори options:

`allow-query {список_ip}` - Дозволяє відповіді на запити тільки з список_ip. При відсутності - сервер відповідає на всі запити.
`allow-recursion {список_ip}` - На запити з списку ip будуть виконуватися рекурсивні запити. Для інших - ітеративні. Якщо не заданий параметр, то сервер виконує рекурсивні запити для всіх мереж.
`allow-transfer {список_ip}` - Вказує список серверів, яким дозволено брати зону з сервера (в основному тут вказують slave сервера)
`directory / path / to /`

`work / dir` - вказує абсолютний шлях до робочого каталогу сервера. Цей оператор допустимо тільки в розділі `options.forwarders {ip порт, ip порт ...}` - вказує адреси хостів і якщо потрібно порти, куди переадресовувати запити (зазвичай тут вказуються DNS провайдерів ISP). `forward ONLY` або `forward FIRST` - параметр `first` вказує, DNS-сервера намагатися вирішувати імена за допомогою DNS-серверів, зазначених в параметрі `forwarders`, і лише в разі, якщо дозволити ім'я за допомогою даних серверів не вдалося, то буде здійснювати спроби дозволу імені самостійно. `notify YES | NO` - YES - повідомляти slave сервера про зміни в зоні, NO - не повідомляють. `recursion YES | NO` - YES - виконувати рекурсивні запити, якщо просить клієнт, NO - не виконувати (тільки ітеративні запити). Якщо відповідь знайдена в кеші, то повертається з кешу. (Може використовуватися тільки в розділі Options)

розділ Zone

Визначає опис зон (и). Формат розділу: `zone {оператори_розділу_zone};`
Оператори, які найбільш часто використовуються:

`allow-update {список_ip}` - вказує системи, яким дозволено динамічно оновлювати дану зону. `file "назва_файла"` - вказує шлях до файлу параметрів зони (повинен бути розташований в каталозі, визначеному в розділі `options` оператором `directory`) `masters {список_ip}` - зазначає список майстер-серверів. (Припустимо тільки в підлеглих зонах) `type "тип_зони"` - вказує тип зони, описуваної в поточному розділі, `тип_зони` може набувати таких значень:

`forward` - вказує зону переадресації, яка переадресовує запити, які прийшли в цю зону. `hint` - вказує допоміжну зону (даний тип містить інформацію про кореневих серверах, до яких сервер буде звертатися в разі неможливості знайти відповідь в кеші) `master` - вказує працювати в якості майстер сервера для поточної зони. `slave` - вказує працювати в якості підлеглого сервера для поточної зони.

Розширені можливості пошуку конфігурації

Значення часу в файлах зон за замовчуванням вказується в секундах, якщо за ними не стоїть одна з наступних букв: S - секунди, M - хвилини, H- годинник, D - дні, W - тижні. Відповідно, запис 2h20m5s матиме значення 2 години 20 хвилин 5 секунд і відповідати 8405 секунд.

Будь-яке ім'я хоста / записи, які не закінчуються точкою вважається не FQDN ім'ям і буде доповнено ім'ям поточної зони. Наприклад, запис `domain` в файлі зони `example.com` буде розгорнуто в FQDN-им'я `domain.example.com`.

У конфігураційних файлах BIND можуть застосовуватися такі директиви:

\$ TTL - визначає TTL за замовчуванням для всіх записів в поточній зоні.
 \$ ORIGIN - змінює ім'я зони з зазначеного в файлі `named.conf`. При цьому, область дії даної директиви не поширюється "вище" (тобто якщо файл включений директивою \$ INCLUDE, то область дії \$ ORIGIN не поширюється на батьківський)\$ INCLUDE - включає вказаний файл як частина файлу зони.

Для того щоб локальний Резолвер сервера теж використовував локальний DNS, необхідно привести файл `resolv.conf` до наступного вигляду:

```
DNS: ~ # cat /etc/resolv.conf nameserver 127.0.0.1
```

Якщо в імені ресурсної записи зустрічається символ "*", то він означає що замість нього можна мати на увазі будь-яку дозволена послідовність символів. Такий запис називають "wildcard запис". Однак, символ "*" не може бути використаний будь-де. Це може бути тільки перший символ в поле Name поточного домену, відокремлений від решти символом "."

Налаштування кешуючого DNS сервера

Після установки `bind`, він повністю готовий працювати як кешуючий DNS сервер без додаткового налаштування. Єдиний недолік - він обробляє запити на всіх інтерфейсах.

Висновки до розділу 2

Хоча URL-адреси, які ми зазвичай вводимо в наші веб-браузери, – це легко запам'ятовувані імена, ви можете замість цього використовувати IP-адресу, на яку вказує ім'я хосту, наприклад <https://151.101.1.121>), щоб отримати доступ до того самого веб-сайту. Це пов'язано з тим, що ви все одно отримуєте доступ до того самого сервера в будь-якому випадку – один метод (з використанням імені) просто легше запам'ятати. Зверніть увагу, що якщо у вас коли-небудь виникає проблема з контактом вашого пристрою з DNS-сервером, ви завжди можете обійти це, ввівши IP-адресу в адресний рядок замість імені хосту. Більшість людей не зберігають локальний список IP-адрес, які відповідають іменам хостів, оскільки, врешті-решт, у цьому полягає вся мета використання DNS-сервера. Це працює не з кожним веб-сайтом та IP-адресою, оскільки деякі веб-сервери налаштували спільний хостинг, а це означає, що доступ до IP-адреси сервера через веб-браузер не описує, яку саме сторінку слід відкрити. Пошук “телефонної книги”, який визначає IP-адресу на основі імені хосту, називається а прямий пошук DNS. Навпаки, а зворотний пошук DNS, це щось інше, що можна зробити за допомогою DNS-серверів. Це коли ім'я хосту ідентифікується за його IP-адресою. Цей тип пошуку спирається на ідею, що IP-адреса, пов'язана з цим конкретним іменем хосту, є статичною IP-адресою. Бази даних DNS зберігають багато речей, крім IP-адрес та імен хостів.

РОЗДІЛ 3 СТВОРЕННЯ WEB-ІНТЕРФЕЙСУ ДЛЯ МЕРЕЖЕВОГО КАТАЛОГУ

3.1 Розмітка сторінки, css

Для створення повноцінної сторінки будемо використовувати три компоненти:

1. HTML – розмітка сторінки.
2. CSS - формальна мова опису зовнішнього вигляду документа, написаного з використанням мови розмітки.
3. JavaScript – мова програмування, яка дає можливість реалізовувати складну поведінку веб-сторінки [22].

Плагін під назвою Emmet для Notepad++ дозволяє прискорити процес введення гіпертекстових рядків через поєднання клавіш.

Завантажуємо і встановлюємо цей плагін (рис. 3.1).



Рис. 3.1. Сторінка завантаження плагіну Emmet

На першому етапі створюємо гіпертекстову розмітку майбутньої сторінки. Вводимо «html:5» і отримуємо стандартну розмітку

Даємо заголовок майбутній сторінці

```
<title>Мережеве сховище</title>
```

Зберігаємо в папці, вказавши розширення файлу (Рис. 3.2).

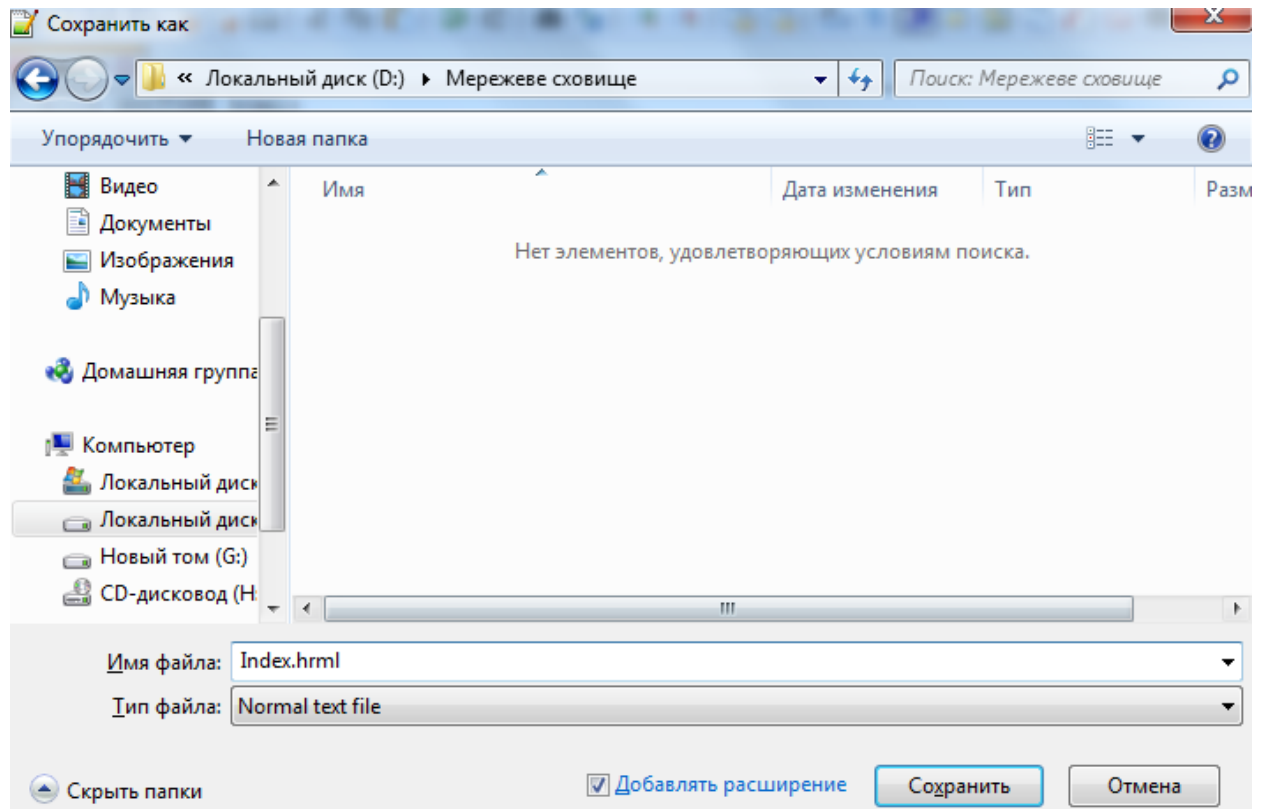


Рис. 3.2. Директорія розміщення сторінки

У цій папці створюємо каталог assets, де будуть зберігатися допоміжні файли – зображення, текстові документи. А також файли для CSS і JavaScript.

У папці assets створюємо каталог images та поміщаємо туди графічний файл під назвою fav з розширенням .jpg – це буде іконка сайту. Для її відображення прописуємо рядок після тега title:

```
<link rel="shortcut icon" href="assets/images/fav.jpg"> (Рис.3.3.)
```

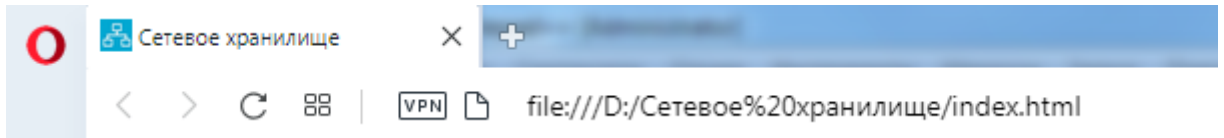


Рис. 3.3. Сторінка з приєднаним графічним файлом-іконкою

Підключення зовнішніх файлів відбувається через "link rel", після чого потрібно вказати відносний або абсолютний шлях до файлу.

Далі прописуємо рядок, який приєднає до сторінки шрифти.

```
<link href="https://fonts.googleapis.com/css?family=Source+Sans+Pro"
rel="stylesheet">
```

У папці assets створюємо каталог CSS і в ньому розміщуємо файл під назвою bootstrap.min та розширенням .css

Підключаємо файл до сторінки.

Аналогічно чинимо з fontawsom-all.min.css. Шрифт Awesome — це колекція векторних іконок, що масштабуються. Іконки можна формувати за допомогою CSS-властивостей, встановлювати для них колір, розмір, тінь та багато іншого [25].

```
<link rel="stylesheet" href="assets/css/fontawsom-all.min.css">
```

До сторінки потрібно підключити ще два файли, які необхідно завантажити: owl.carousel.min.css і owl.theme.min.css.

Owl.js це jQuery плагін, який дозволяє створювати ефект каруселі [26].

Їх ми вкладаємо в окремий каталог, який має таку структуру:

```
assets/plugins/testimonial/css/
```

На цьому етапі нам ще необхідно створити файл під назвою style з розширенням .css. Його ми вкладаємо в папку assets/css

Перший елемент сторінки виглядає таким чином (Рис. 3.4):


```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1,
6      shrink-to-fit=no">
7      <title>Сетевое хранилище</title>
8
9      <link rel="shortcut icon" href="assets/images/fav.jpg">
10     <link href="https://fonts.googleapis.com/css?family=Source+Sans+Pro" rel=
11     "stylesheet">
12     <link rel="stylesheet" href="assets/css/bootstrap.min.css">
13     <link rel="stylesheet" href="assets/css/fontawsom-all.min.css">
14     <link rel="stylesheet" href="assets/plugins/testimonial/css/owl.carousel.min.css">
15     <link rel="stylesheet" href="assets/plugins/testimonial/css/owl.theme.min.css">
16     <link rel="stylesheet" type="text/css" href="assets/css/style.css" />
17 </head>
18 <body>
19 </body>
20 </html>

```

Рис. 3.4. Розмітка сторінки в текстовому редакторі Notepad++

На наступному етапі створюємо універсальний контейнер для потокового контенту (<div>) і називаємо його header-top. Це буде контейнер, який відповідає за верхню частину сторінки

Прописуємо <header> та вкладаємо в нього header-top

У файлі style.css прописуємо властивості контейнера header та відповідних підкласів. Тут ми вказали властивості контейнера: його позицію, ширину, положення елемента та нижче розташованих елементів по осі z та колір тла. CSS-властивість box-shadow дозволяє додавати елементу одну або кілька тіней, вказуючи їх параметри через кому. Властивість тіні є описані зміщення по x і y від елемента, радіус розмиття і його радіус, а також колір, border-bottom-color встановлює колір кордону, значення може бути у будь-якому допустимому для CSS форматі. У нашому випадку – 1 піксель. Solid означає «суцільний».

Прописуємо ще один підклас `<div class = "col-lg-7 col-md-12 left-item">`

Властивість `float` CSS вказує, що елемент повинен бути взятий з нормального потоку і поміщений вздовж лівої або правої сторони контейнера (у нашому випадку—з лівої).

Властивість `padding` встановлює внутрішні відступи/поля з усіх боків елемента. `Font-size` визначає розмір шрифту елемента.

Поміщаємо сюди список

```
<ul>
```

```
<li><i class="fas fa-envelope-square"></i>
```

І вказуємо електронну адресу гіпотетичної компанії, яка надає послуги серверного сховища

```
egotsv03@gmail.com
```

Сюди ж поміщаємо гіпотетичний телефон. Сторінка на даному етапі виглядає так (Рис. 3.5)

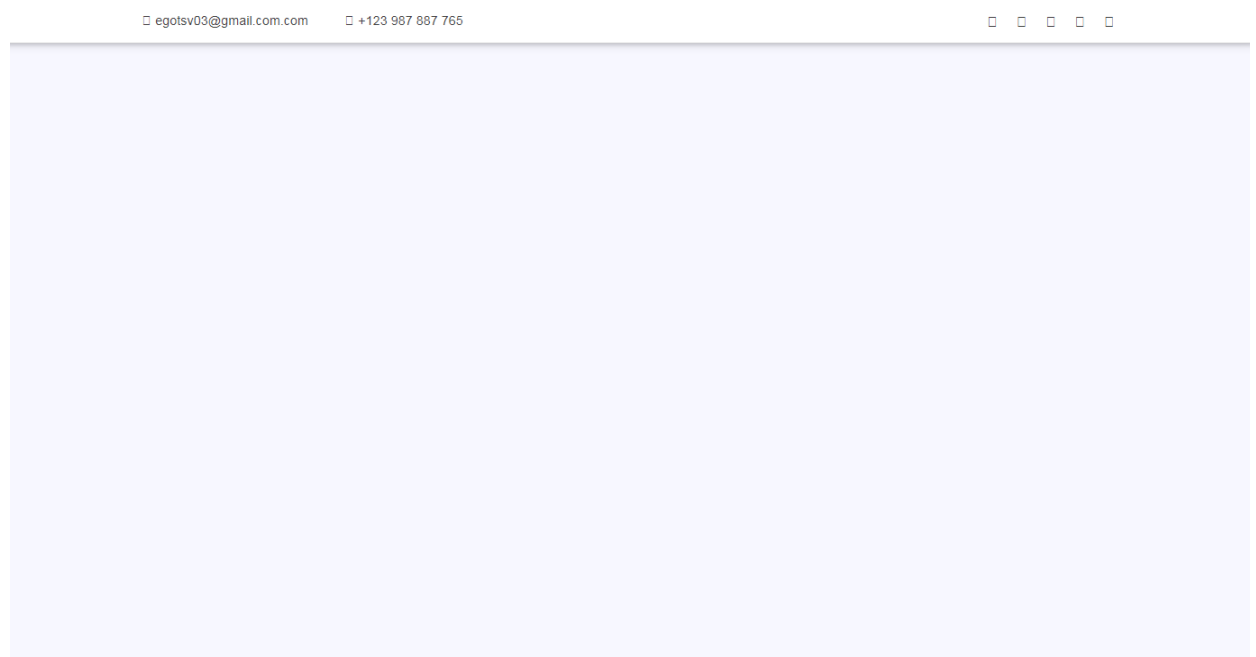


Рис. 3.5. Вигляд сторінки після проведення першої розмітки

Створюємо каталог webfonts у папці assets. Тут будуть розміщені майбутні файли іконок та шрифти, що відповідають за верхнє меню сторінки. Їх можна створити через онлайнвий інструмент webfonts-generator, або скачати готові графічні об'єкти. Спростимо процедуру і завантажимо шрифти, вони розміщуються у вільному доступі (Рис. 3.6)

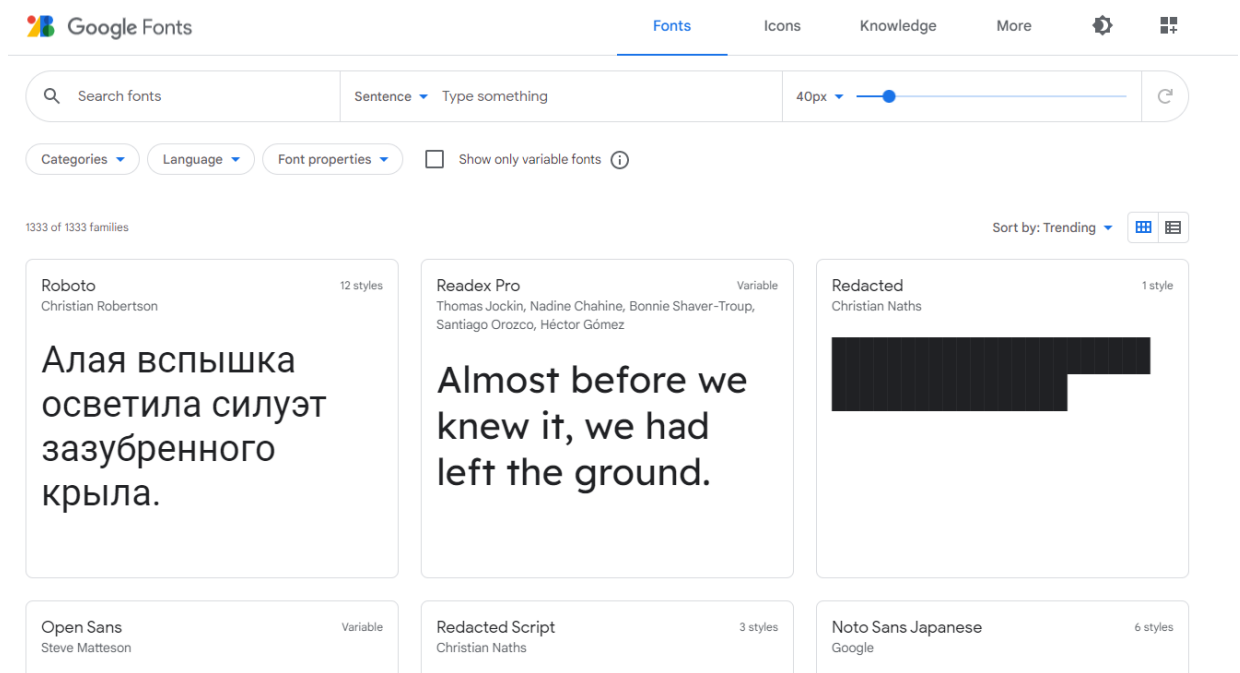


Рис. 3.6. Сторінка завантаження шрифтів Google Fonts

У рядку `<i class="fas fa-facebook-alt"></i>` вказуємо назву графічного об'єкта, який нам підходить. Безкоштовна бібліотека таких об'єктів розміщена в Інтернеті (Рис. 3.7. [27]):



Рис. 3.7. Верхній блок сторінки після приєднання і class="fas"

Робимо заміну тега «a» у списку на «a href=""» та вписуємо посилання на ресурс:

`<i class="fab fa-facebook-f"></i>`

(Рис. 3.8.)

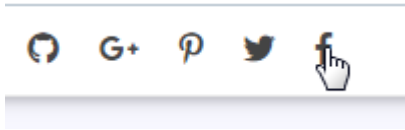


Рис. 3.8. Активна іконка-посилання

Іконка стала активною. Аналогічно чинимо з іншими іконками

Створюємо логотип сторінки сховища мережі. Як зображення можна взяти готове фото або створити його самостійно. Скористаємося графічним редактором (Рис. 3.9.).

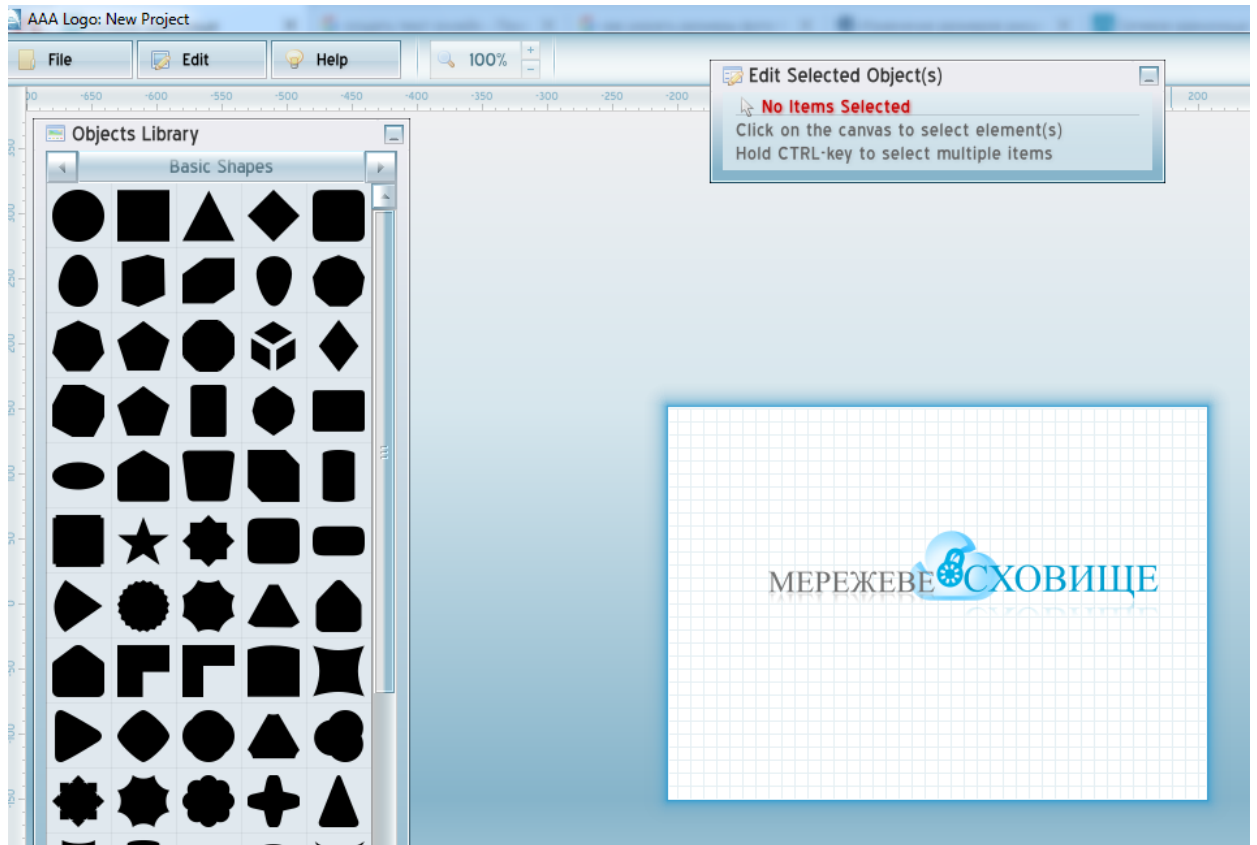


Рис. 3.9. Створення логотипу сторінки в програмі AAA Logo

Експортуємо зображення до каталогу images.

Створюємо новий контейнер та даємо йому ідентифікатор `nav-head`, клас `header-nav`. Вкладаємо в нього ще два контейнери `container` та `row`.

Прописуємо характеристики цих контейнерів у файлі `style.css` (Рис. 3.10.)

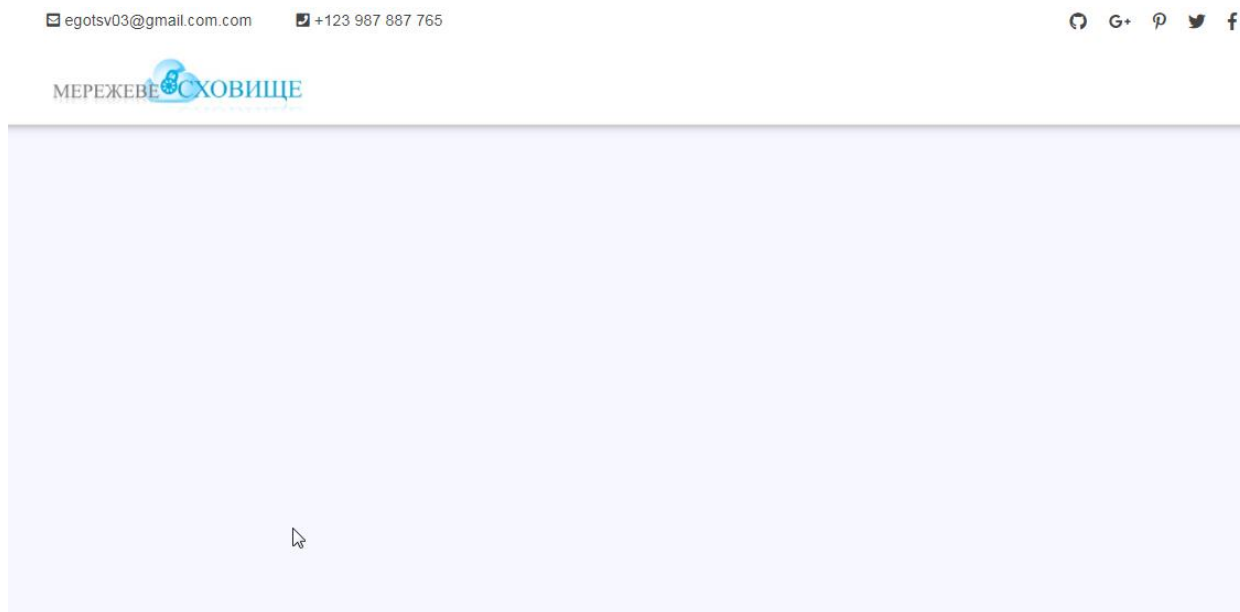


Рис. 3.10. Вигляд сторінки з логотипом

Прописуємо пункти меню. Створюємо ідентифікатор `.menu` та прописуємо його характеристики. У теці, що розкривається, а `href` –прописуємо посилання кожного пункту меню на відповідні сторінки. На даному етапі логічно створити ці сторінки, але залишити їх порожніми, окрім сторінки «сховища», де розміщуватимуться тестові файли для завантаження (Рис. 3.11.).



Рис. 3.11. Вигляд сторінки з прописаними пунктами меню

Створюємо центральний слоган сторінки. Для цього створюємо контейнер `slider`. Прописуємо характеристики контейнера `slider` (Рис. 3.12.)

```

/* ===== Slider CSS ===== */
.slider {
  height: 600px;
  overflow: hidden;
  background-color: #0098ef;
  padding-top: 150px; }
.slider::before {
  right: 65%;
  padding-right: 220px;
  content: '';
  background-position: left bottom;
  background-image: url(../images/clouds-light-left.png);
  position: absolute;
  z-index: 2;
  margin-top: 272px;
  width: 1046px;
  height: 178px;
  background-size: 1046px 178px;
  background-repeat: no-repeat; }
.slider::after {
  left: 66%;
  padding-right: 320px;
  content: '';
  background-position: left bottom;
  background-image: url(../images/clouds-light-left.png);
  position: absolute;
  z-index: 2;
  margin-top: -202px;
  width: 1046px;
  height: 178px;
  background-size: 1046px 178px;
  background-repeat: no-repeat; }
.slider .slider-titl {
  color: #FFF;
  text-align: center;
  padding-top: 40px;
  width: 700px;
  margin: auto; }
@media screen and (max-width: 767px) {
  .slider .slider-titl {
    width: 100%; } }
.slider .slider-titl img {
  margin-top: 50px;
  width: 76%; }

```

Рис. 3.12. Прописані CSS-властивості контейнера slider текстовому редакторі Notepad++

Пишемо заголовок і вказуємо його розмір

<h1>Найдешевші пропозиції веб-хостингу SSD</h1>

Пишемо підзаголовок:

<p>Розгорніть свою інфраструктуру послуг на нашій повністю резервованій, високопродуктивній хмарній платформі та скористайтеся перевагами її високої надійності, безпеки та набору корпоративних функцій.</p> (Рис. 3.13.)

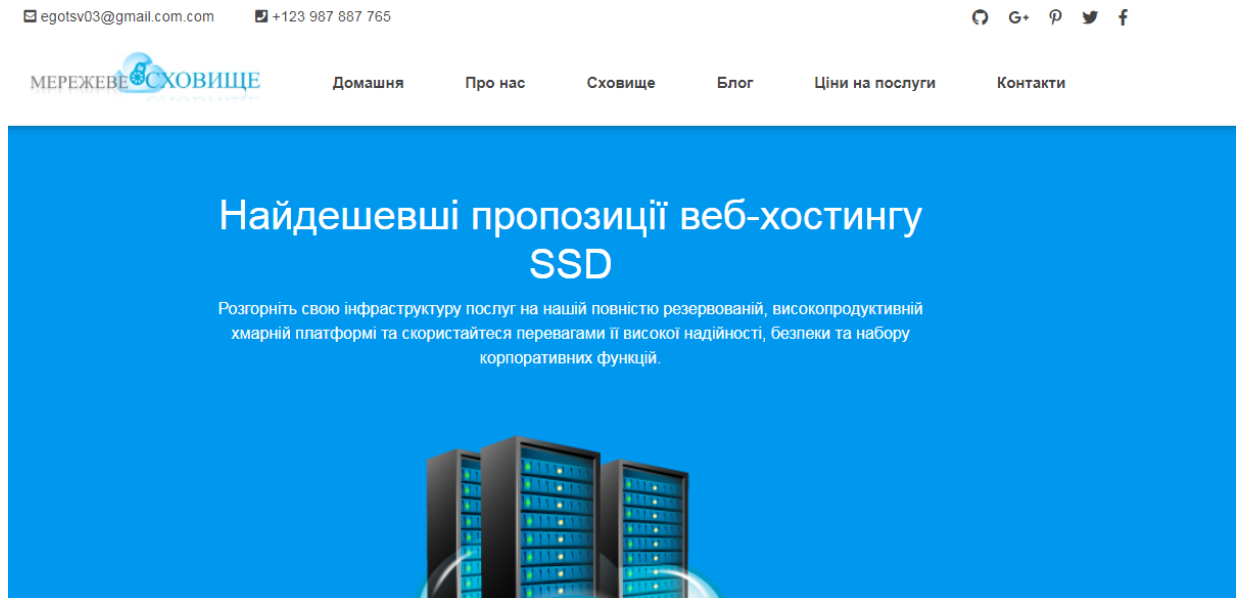


Рис. 3.13. Центральний вигляд сторінки

Створюємо ще один контейнер для текстової інформації key-features та прописуємо характеристики (Рис. 3.14.)

```

/* ===== Key Features CSS
===== */
.key-features {
  padding: 50px;
  background-color: #FFF; }
@media screen and (max-width: 940px) {
  .key-features {
    padding: 50px 10px; } }
.key-features .single-key {
  background-color: #FFF;
  margin-bottom: 30px;
  text-align: center;
  padding: 30px; }
.key-features .single-key h5 {
  font-family: "mouse-300", Arial, Helvetica, sans-serif;
  font-size: 1.5rem; }
.key-features .single-key i {
  background-color: #0098ef;
  padding: 20px;
  font-size: 42px;
  color: #FFF;
  margin-bottom: 30px;
  border-radius: 50px; }
.key-features .single-key:hover {
  box-shadow: 0 2px 5px 0 rgba(0, 0, 0, 0.16), 0 2px 10px 0 rgba(0, 0,
0, 0.12); }
.key-features .single-key p {
  font-family: "mouse-300", Arial, Helvetica, sans-serif;
  font-size: 15px; }

```

Рис. 3.14. Прописані CSS-властивості контейнера key-features текстовому редакторі Notepad++

Створюємо контейнери для опису ключових можливостей. Заповнюємо їх текстом (Рис. 3.15)

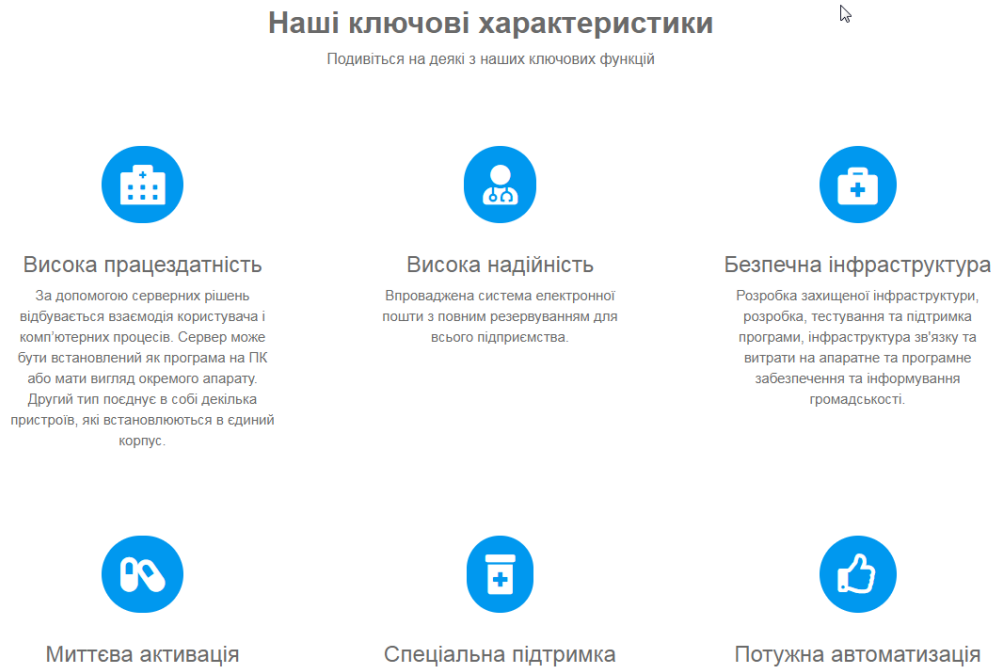


Рис. 3.15. Вигляд блоку «ключові характеристики»

Якщо йдеться про комерційну організацію, логічно вставити блок із цінами на послуги хостингу. Для корпоративного веб-сайту цей етап можна пропустити.

Створюємо ще один контейнер і називаємо його packages.

```
<div class="container">
```

```
<div class="inner-title"> (Рис. 3.16.)
```

```

[ ] .packages {
[ ]   padding: 50px; }
[ ] @media screen and (max-width: 940px) {
[ ]   [ ] .packages {
[ ]     [ ] padding: 50px 10px; } }
[ ]   [ ] .packages .product-box {
[ ]     background-color: #FFF;
[ ]     border-radius: 20px;
[ ]     box-shadow: 0 2px 5px 0 rgba(0, 0, 0, 0.16), 0 2px 10px 0 rgba(0, 0, 0,
[ ]     0.12); }
[ ]   [ ] .packages .product-box .product-header {
[ ]     padding: 20px;
[ ]     text-align: center;
[ ]     border-bottom: 1px solid #CCC; }
[ ]     [ ] .packages .product-box .product-header h4 {
[ ]       margin-bottom: 0px; }
[ ]     [ ] .packages .product-box .product-header p {
[ ]       font-size: .9rem; }
[ ]   [ ] .packages .product-box .product-popular {
[ ]     padding: 16px;
[ ]     color: #FFF;
[ ]     background-color: #0098ef;
[ ]     font-family: "mouse-500", Arial, Helvetica, sans-serif;
[ ]     border-top-left-radius: 15px;
[ ]     border-top-right-radius: 15px;
[ ]     text-align: center; }
[ ]   [ ] .packages .product-box .product-price {
[ ]     padding: 20px;
[ ]     text-align: center;
[ ]     font-size: 28px;
[ ]     font-weight: 600; }
[ ]   [ ] .packages .product-box .product-features {
[ ]     text-align: center; }
[ ]     [ ] .packages .product-box .product-features ul li {
[ ]       padding: 3px;
[ ]       color: #444;
[ ]       font-family: "mouse-300", Arial, Helvetica, sans-serif; }

```

Рис. 3.16. Прописані CSS-властивості контейнера packages текстовому редактору Notepad++

Заповнюємо секції текстом. Отримуємо три варіанти для замовлення (Рис. 3.17.)

Наші ключові характеристики

Подивіться на деякі з наших ключових функцій

Початковий рівень Середнє навантаження	Найбільш популярний тариф Професійний рівень Максимальне навантаження	Бізнес-рівень Максимальне навантаження
\$9 / щомісяця	\$19 / щомісяця	\$49 / щомісяця
10 Розміщених доменів 25 GB об'єм сховища 200 GB Пропускна здатність	20 Розміщених доменів 50 GB Об'єм сховища 400 GB Пропускна здатність	50 Розміщених доменів 1000 GB Об'єм сховища 400 GB Пропускна здатність
Виділена IP-адреса Угода про рівень безвідмовної роботи на 90,00%. Вихідний канал 1000 Мбіт	Виділена IP-адреса Угода про рівень безвідмовної роботи на 95,00%. Вихідний канал 2000 Мбіт	Виділена IP-адреса Угода про рівень безвідмовної роботи на 100%. Вихідний канал 4000 Мбіт
Замовити зараз	Замовити зараз	Замовити зараз

Рис. 3.17. Блок «Наші ключові характеристики»

У властивості CSS вказуємо, що при наведенні на курсор пункт меню змінює свій колір. Створюємо контейнер для точки входу/реєстрації. Прописуємо властивості CSS. Заповнюємо текстом (Рис. 3.18.)

10 Розміщених доменів 25 GB об'єм сховища 200 GB Пропускна здатність	20 Розміщених доменів 50 GB Об'єм сховища 400 GB Пропускна здатність	50 Розміщених доменів 1000 GB Об'єм сховища 400 GB Пропускна здатність
Виділена IP-адреса Угода про рівень безвідмовної роботи на 90,00%. Вихідний канал 1000 Мбіт	Виділена IP-адреса Угода про рівень безвідмовної роботи на 95,00%. Вихідний канал 2000 Мбіт	Виділена IP-адреса Угода про рівень безвідмовної роботи на 100%. Вихідний канал 4000 Мбіт
Замовити зараз	Замовити зараз	Замовити зараз

Увійдіть для вибору пропозиції

Наші технічні спеціалісти допоможуть зробити найкращий вибір

[Увійти](#)

Рис. 3.18. Блок входу/реєстрації

При наведенні на кнопку «Увійти» вона змінює свій колір. Ці властивості прописані у CSS.

Створюємо контейнер для відгуків. Прописуємо заголовок та підзаголовок.

```
<h2>Відгуки</h2>
```

```
<p>Подивіться, що люди говорять про нас</p>
```

Розділ із відгуками буде реалізований за допомогою слайдера: зміна відгуку при натисканні на бічну стрілку. Тут буде реалізована зміна слайдера при наведенні на стрілку з певною затримкою в кілька секунд (Рис. 3.19.)

```

    <h2>Відгуки</h2>
    <p>Подивіться, що люди говорять про нас</p>
</div>
<div class="row">
  <div class="col-md-offset-2 float-auto col-md-10">
    <div id="testimonial-slider" class="owl-carousel">
      <div class="testimonial">
        <div class="pic">
          
        </div>
        <p class="description">
          Єдиний зручний спосіб для зберігання файлів, який
          приходить в голову - це хмарне сховище.
          Але яке ж вибрати серед багатьох найкраще? Адже не секрет,
          що за свої послуги вони теж беруть гроші і не малі, а
          безкоштовно надають лише дуже маленьку частину простору.
          Отже, користуючись не одним сховищем, я вибрав собі єдине.
        </p>
        <h3 class="title">Іван
          <span class="post"> - Розробник</span>
        </h3>
      </div>
      <div class="testimonial">
        <div class="pic">
          
        </div>
        <p class="description">
          Чим дорожить кожен із нас? Фотографіями, файлами з відео,
          музикою, документами ..., взагалі пам'яттю та інформацією.
          Зрозуміло, найнеобхідніше і найцінніше, ми намагаємося
          зберегти на "століття", мати до цих даних простий і
          швидкий доступ. Безпечно та надійно зберігати цінну
          інформацію. Карту пам'яті легко втратити, диск із даними
          пошкодити чи зламати. Тут і на допомогу приходять хмарні
          сховища.
        </p>
        <h3 class="title">Дмитро
          <span class="post"> - Інженер</span>
        </h3>
      </div>
    </div>
  </div>
</div>

```

Рис. 3.19. Розмітка блоку «Відгуки»

На цьому етапі візуально блок із відгуками поки що не реалізований. Тут потрібно буде підключати Javascript.

Створимо в папці assets створюємо каталог js. Тут розташовуватимуться файли з розширенням .js і кодом JavaScript.

Завантажуємо з Інтернету файл `bootstrap.min.js` (в архіві) та поміщаємо його вміст у цей каталог.

Завантажуємо з Інтернету готову бібліотеку jQuery - набір функцій JavaScript, що фокусується на взаємодії JavaScript та HTML, поміщаємо його в той же каталог.

Імпортуємо `popper.js` – невеликий плагін підказок для сайту. Для економії часу імпортуємо більш ранню версію (1.x), тому що в наступній версії просте підключення до HTML відсутнє [28]. Поміщаємо цей файл у вищезгаданий каталог.

Створюємо текстовий файл, перейменовуємо його в `script.js` – тут буде розміщено основний код JavaScript для нашої сторінки [30].

Приєднуємо файли JavaScript до сторінки

Додаємо фото до «автора» відгуку. Створимо каталог `testimonial` (обговорення) і поміщаємо його в папку `images` (Рис. 3.20.)

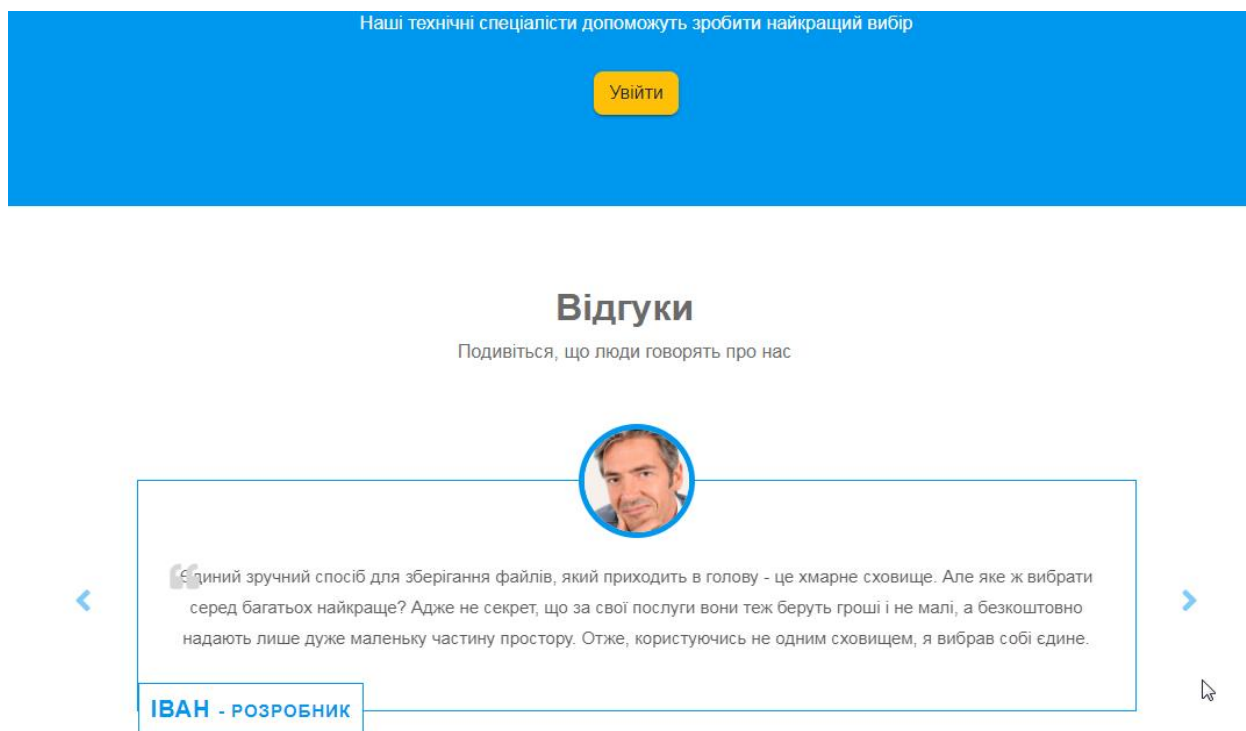


Рис. 3.20. Блок «Відгуки» в готовому вигляді

Створюємо блок footer – розділ, розташований під основним текстом. У ньому будуть прописані основні дані про власника.

Прописуємо текст (Рис. 3.21.)

```

<h2>Про нас</h2>
<p>
    Хмарне середовище – провідний постачальник послуг з
    інформаційних технологій, консалтингу та
    бізнес-процесів. Наші віддані своїй справі
    співробітники пропонують стратегічні ідеї,
    технологічні знання та досвід у галузі.
</p>
<p>Ми зосереджуємось на технологіях, які обіцяють
    знизити витрати, оптимізувати процеси та прискорити
    вихід на ринок, завдяки нашим високоякісним процесам та
    багатим досвідом управління. </p>
</div>
<div class="col-md-4 col-sm-12">
    <h2>Корисні посилання</h2>
    <ul class="list-unstyled link-list">
        <li><a ui-sref="about" href="#/about">Про нас</a><i
            class="fa fa-angle-right"></i></li>
        <li><a ui-sref="portfolio" href="#/portfolio">Сховище
            </a><i class="fa fa-angle-right"></i></li>
        <li><a ui-sref="products" href="#/products">Останні
            напрацювання</a><i class="fa fa-angle-right"></i></li>
        <li><a ui-sref="gallery" href="#/gallery">Ціни</a><i
            class="fa fa-angle-right"></i></li>
        <li><a ui-sref="contact" href="#/contact">Наші
            контакти</a><i class="fa fa-angle-right"></i></li>
    </ul>
</div>
<div class="col-md-4 col-sm-12 map-img">
    <h2>Наші контакти</h2>
    <address class="md-margin-bottom-40">
        Вулиця Південна<br>
        Миколаїв<br>
        Україна <br>
        Телефон: +38 098 000 0000</br>
        Email: <a href="mailto:egotsv03@gmail.com" class="">
            <u>mailto:egotsv03@gmail.com</u></a><br>
        <a href="https://yuri.goncharenko.com">
            <u>https://yuri.goncharenko.com</u></a><br>
        Web: <a href="https://yuri.goncharenko.com/" class="">
            <u>https://yuri.goncharenko.com/</u></a>
    </address>
</div>
</div>

```

Рис. 3.21. Розмітка блоку «Footer»

Створюємо ще один контейнер, де буде вказано автора сайту сторінки та продубльовані посилання на гіпотетичні соціальні мережі (Рис. 3.22.)

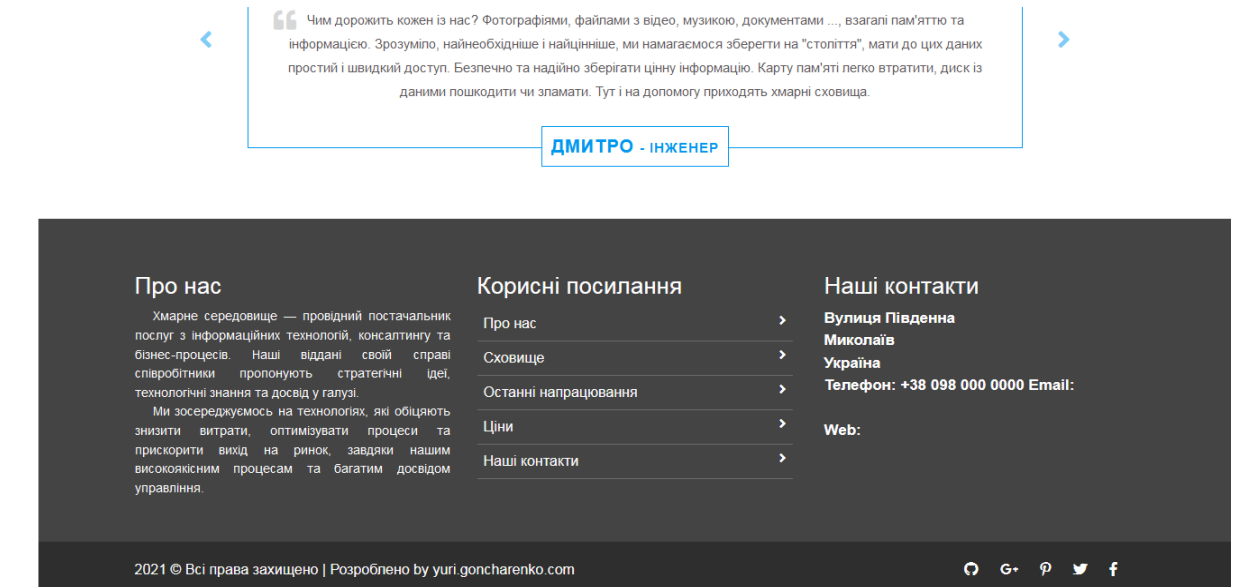


Рис. 3.22. Блок «Footer» в готовому вигляді

3.2 Сторінка services з вихідними тестовими файлами

Створюємо ще один файл із розширенням .html і називаємо його services, поміщаємо в кореневу папку поруч із початковим файлом index.html. Це буде друга сторінка, де будуть розміщені посилання на тестові файли. Реалізована вона буде в тому ж стилі, що й основна сторінка.

Створюємо блок із заголовком з посиланням на домашню сторінку.

Називаємо його «Наше мережеве сховище».

Змінюємо рядок

```
<li><a href="#"><i class="fas fa-home"></i>Домашня</a></li>
```

і додаємо до нього посилання на домашню сторінку (index.html)

`<i class="fas fa-home"></i>Домашня` (Рис. 3.23.)

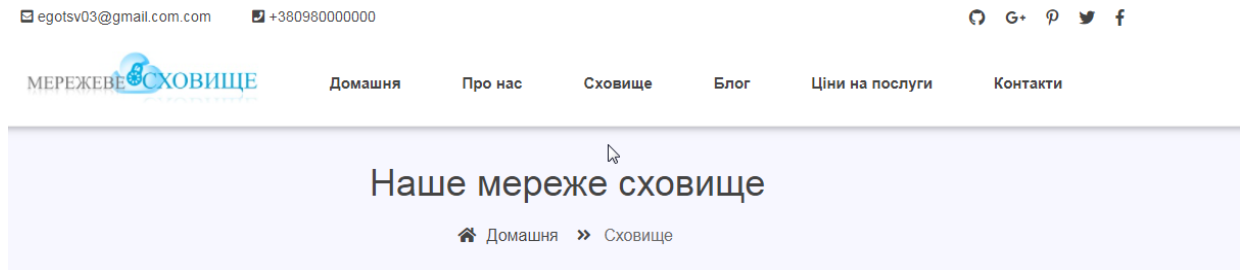


Рис. 3.23. Верхній блок сторінки «Наше мережеве сховище»

Створюємо контейнер `key-features`, його характеристики вже прописані в CSS. Тут будуть розміщені посилання на тестові файли.

Пишемо заголовок та підзаголовок

Створюємо блок `row` (аналогічно до стартової сторінки), де будуть розміщені посилання на тестові файли.

У рядку `<i class="fas fa-file-alt"></i>` вказуємо назву графічного об'єкта, який нам підходить. Безкоштовна бібліотека таких об'єктів розміщена в Інтернеті.

Прописуємо заголовок та підзаголовок (Рис. 3.24.)



Файли загального типу

Сховище

Рис. 3.24. Пункт «Сховище»

Аналогічно - з іншими вкладеними блоками (Рис. 3.25)

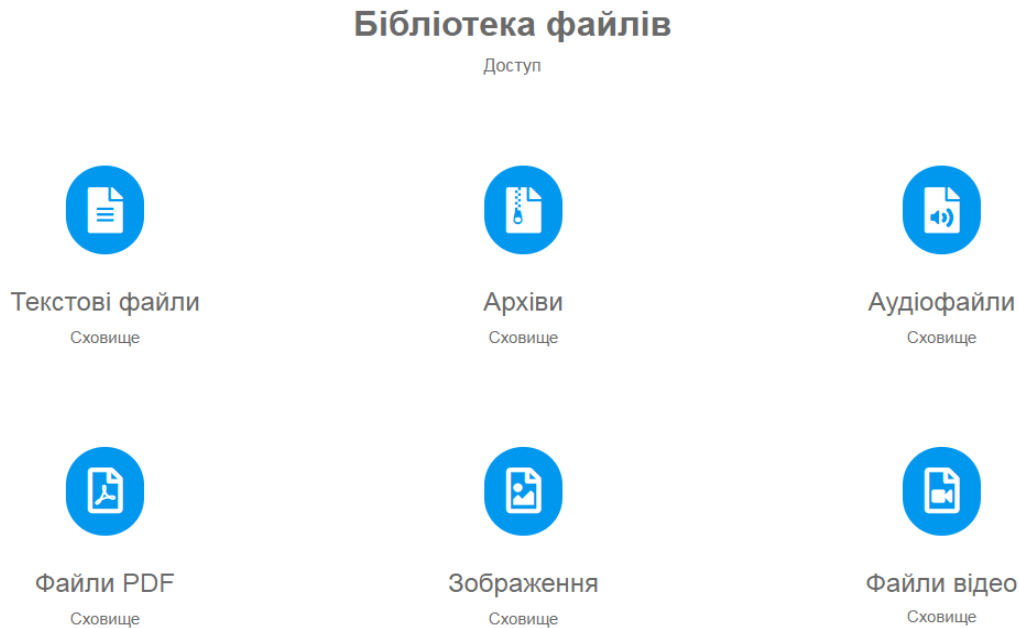


Рис. 3.25. Блок «Бібліотека файлів»

Реалізуємо завантаження тестового файлу при натисканні посилання.

Створюємо таку структуру каталогів: `assets/files/text/` та вкладаємо в папку `text` документ `test` з розширенням `.doc`

У рядку `<p>Сховище</p>`, що відноситься до пункту `<h5>Текстові файли</h5>` перед словом «сховище» пропишемо тег, що відкривається, а `href=""` і вкажемо шлях до файлу. Сюди ж додаємо атрибут `download`.

```
<p><a href="assets/files/text/test.doc" download>Сховище</a></p>
```

Слово "сховище" стало посиланням, при натисканні на неї відбувається процес завантаження текстового файлу (Рис. 3.26.)

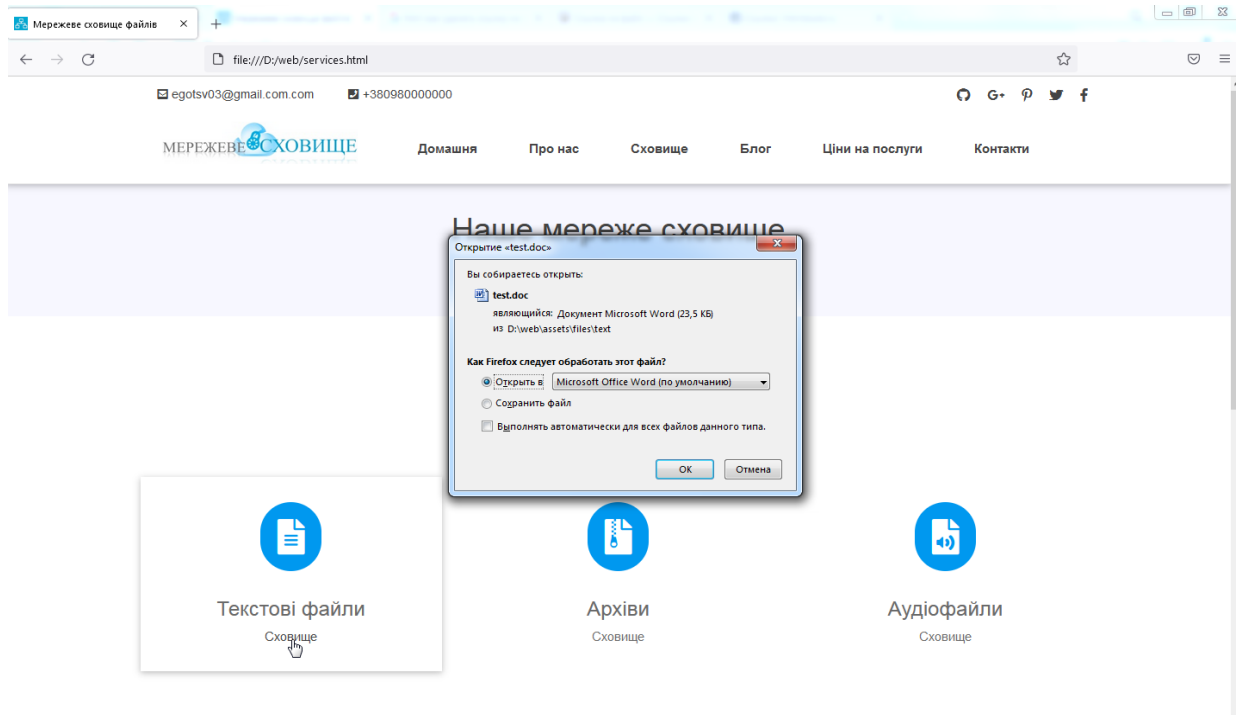


Рис. 3.26. Завантаження тестового файлу

Залежно від браузера, буде можливість або зберегти тестовий файл, або відразу його відкрити.

Додамо можливість завантаження файлу за кнопкою, яка є посиланням. Використовуємо `div class btn btn-warning`, який ми використовували під час роботи з попередньою сторінкою. Його властивості вже прописані у CSS.

`<div class="btn btn-warning">Сховище</div></p>` (Рис. 3.27.)

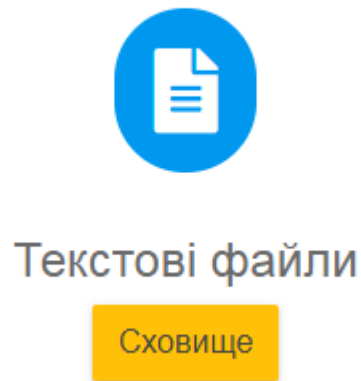


Рис. 3.27. Реалізована кнопка завантаження файлу

Робимо аналогічним чином з іншими блоками даного контейнера. Створюємо структури каталогів у папці assets відповідно до завантажуваного типу файлів (Рис. 3.28.)

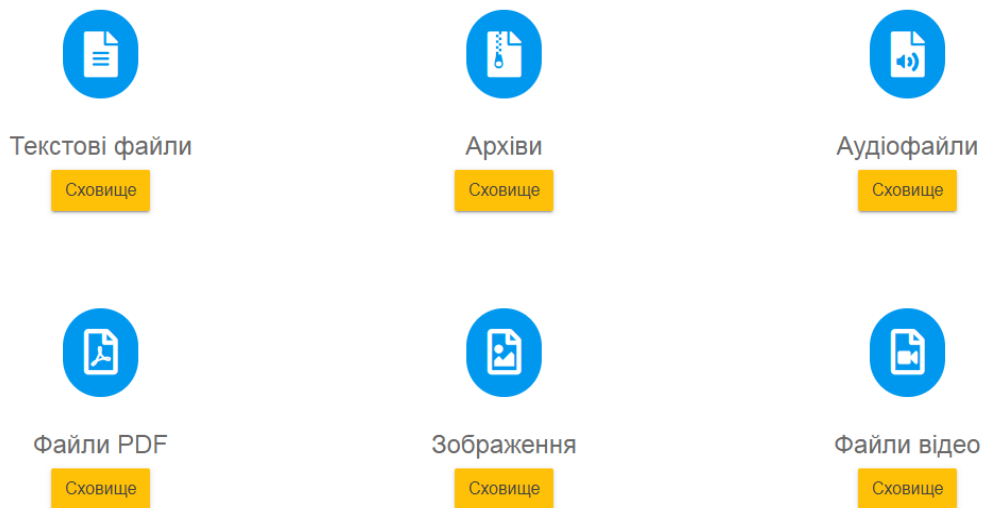


Рис. 3.28. Блок «Завантаження файлів» в готовому вигляді.

Копіюємо всі елементи з блоків footer та col-md-4 col-sm-12 map-imgr із файлу index.html. Також підключаємо JavaScript.

3.3 Створення локального серверу

Корпоративний сайт може бути або в загальному доступі до мережі Інтернет, або перебувати всередині замкнутої мережі однієї організації.

У першому випадку необхідно скористатися послугами хостингової компанії, провайдера тощо: замовити покупку доменного імені, виділене місце на серверному сховищі. Автор сайту отримує адміністративні права, логін, пароль. Щороку всі ці послуги потрібно сплачувати.

Другий варіант передбачає створення сервера на машині, яка знаходиться в межах корпоративної мережі. Доступ до сайту буде надано лише членам мережі. У рамках цієї роботи розглядатиметься саме цей варіант.

Однак є й проміжний варіант, коли до локального сервера можна отримати доступ з Інтернету. У такому випадку необхідно придбати у провайдера статичну IP-адресу (відомо, що більшість провайдерів виділяють своїм клієнтам динамічні IP-адреси) і проводити налаштування сервера. Налаштувати сервер будемо на базі OpenServer. Завантажити можна безкоштовно з офіційного сайту команди-розробників [31].

Перед вибором версії OpenServer важливо визначитися із розрядністю своєї операційної системи. Для цього в меню Пуск достатньо клікнути правою кнопкою миші на "Мій комп'ютер", викликавши пункт "Властивості". У вікні буде вказано розрядність системи.

Після завантаження та початку встановлення програми важливо вибрати директорію для встановлення (Рис. 3.29.)

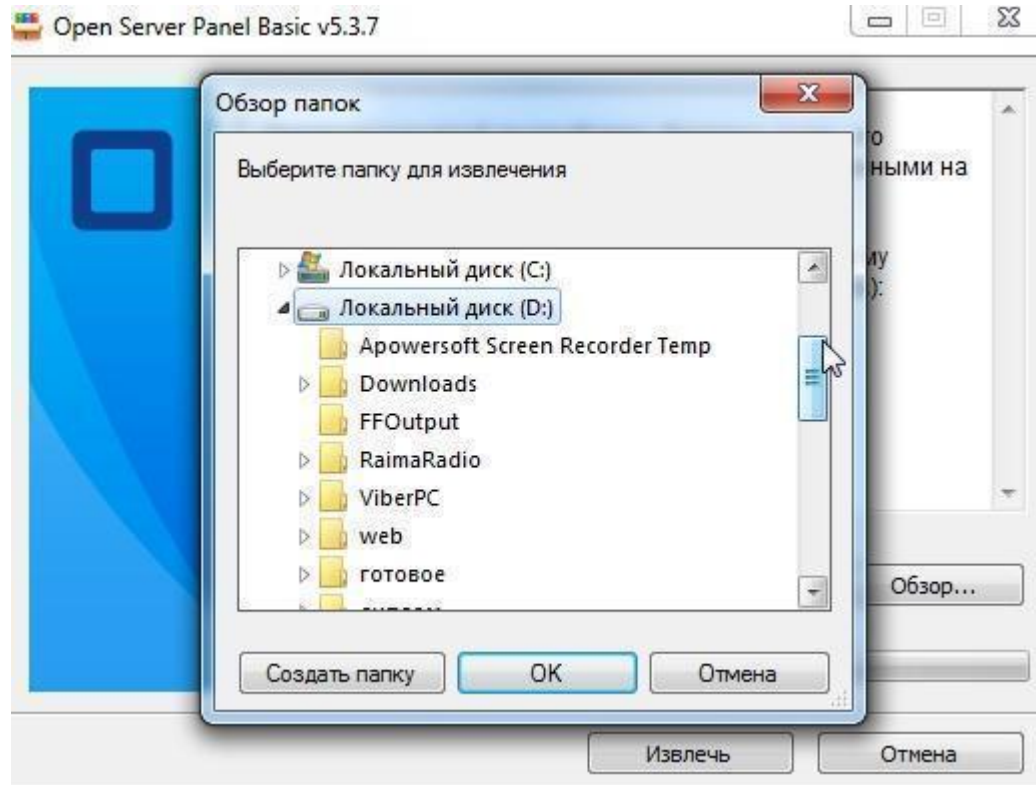


Рис. 3.29. Вибір директорії для інсталяції локального серверу

Рекомендується проводити встановлення не на системний диск, щоб не допустити втрату даних у разі збою операційної системи.

Після закінчення установки буде запропоновано вибрати мову інтерфейсу.

Переходимо в каталог із встановленою програмою та знаходимо папку `domains`, де буде розміщуватися сайт. Копіюємо туди каталог із раніше створеними файлами сторінками корпоративного сайту. Даємо ім'я цієї директорії (`webhost.com`). У цій папці вже знаходиться каталог `localhost`. Перейменовуємо його. `localhost`-це "ім'я", яке вказує на IP-адресу `127.0.0.1`, яка повинна існувати на будь-якому комп'ютері і відноситься до самої машини. (Будь-яка IP-адреса, починаючи з `127.x.x.x` відноситься до локальної машини). В ідеалі не рекомендується використовувати "`localhost`" як ім'я хоста, так як це викликатиме проблеми з деякими програмами, у тому числі і `OpenServer`. Додаємо до імені цього каталогу одиницю.

Запускаємо версію OpenServer відповідно до розрядності системи (Рис. 3.30.)

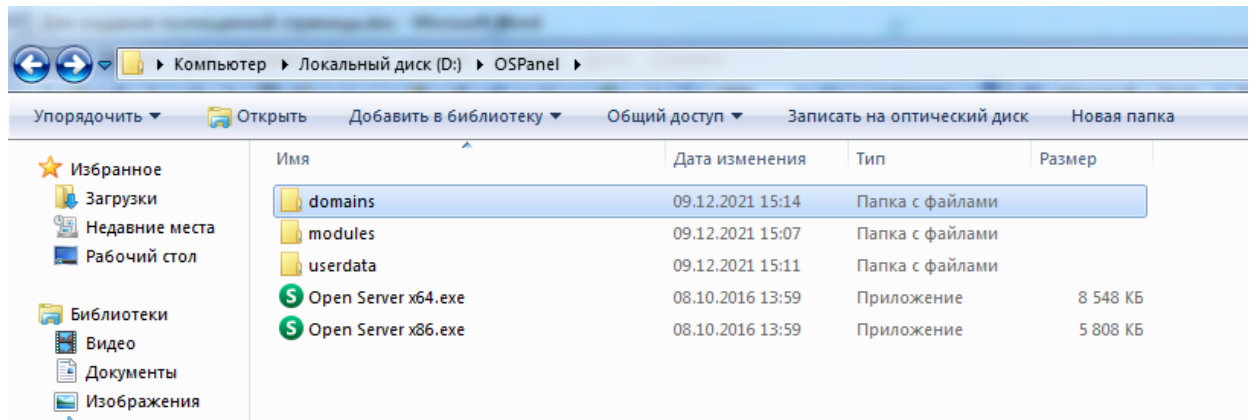


Рис. 3.30. Директорія локального серверу після інсталяції

На панелі інструментів з'являється значок з прапором. Зараз вона неактивна. Правою кнопкою миші клікаємо по цій іконці та викликаємо пункт «Налаштування». Потрапляємо до загального меню програми. Переходимо в пункт «Сервер» (Рис. 3.31.)

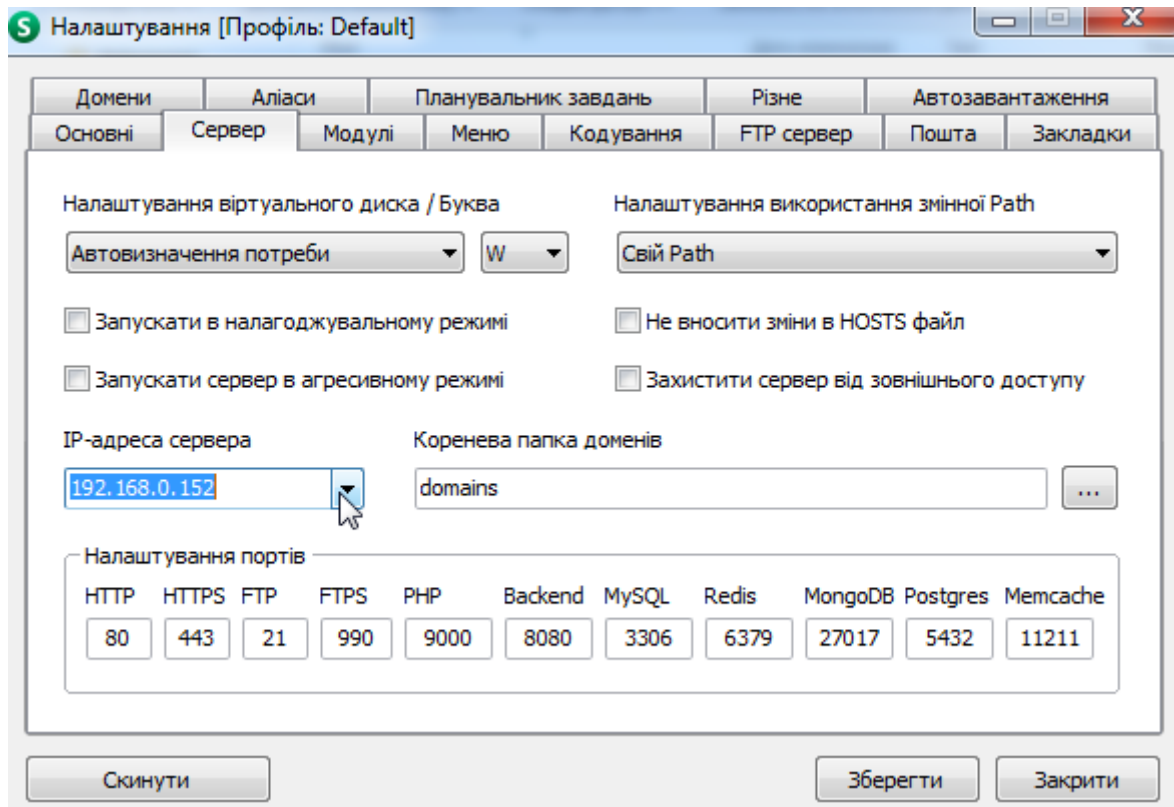


Рис. 3.31. Головне меню OpenServer

Якщо ми настроюємо локальний сервер, нам необхідно вказати IP-адресу машини, на якій розташовується директорія з сайтом.

Щоб дізнатися свою IP-адресу в мережі, необхідно викликати командний рядок (клавіші win+R та введення CMD або через пуск/всі програми/стандартний/командний рядок). У командному рядку вводимо ipconfig та

отримуємо інформацію про свій IP (Рис. 3.32.)

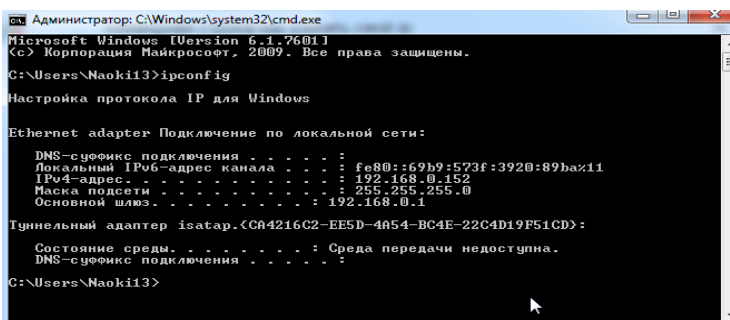


Рис. 3.32. Інформація про локальну IP-адресу в командному рядку

IPv4-адреса. У нашому випадку – 192.168.0.152

DNS сервер має бути надійним. Наприклад, від компанії Google (Рис. 3.33.)

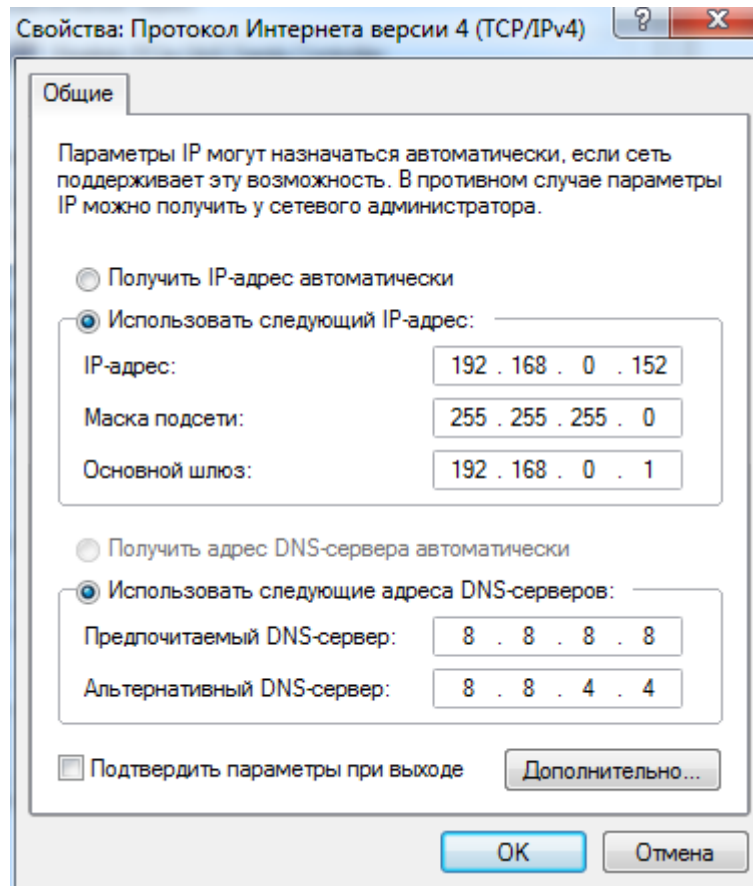


Рис. 3.33. Зміна локальної адреси в ручному режимі

Повертаємось у налаштування програми OpenServer. Коренева папка доменів повинна співпадати за назвою з відповідною директорією.

Переходимо до розділу «Аліаси» та вказуємо оригінальний домен. Це буде IP-адреса машини, на якій розміщується сайт. Кінцевий домен збігається на ім'я з директорією, в якій знаходяться файли сайту. Зберігаємо та закриваємо налаштування (Рис. 3.34.)

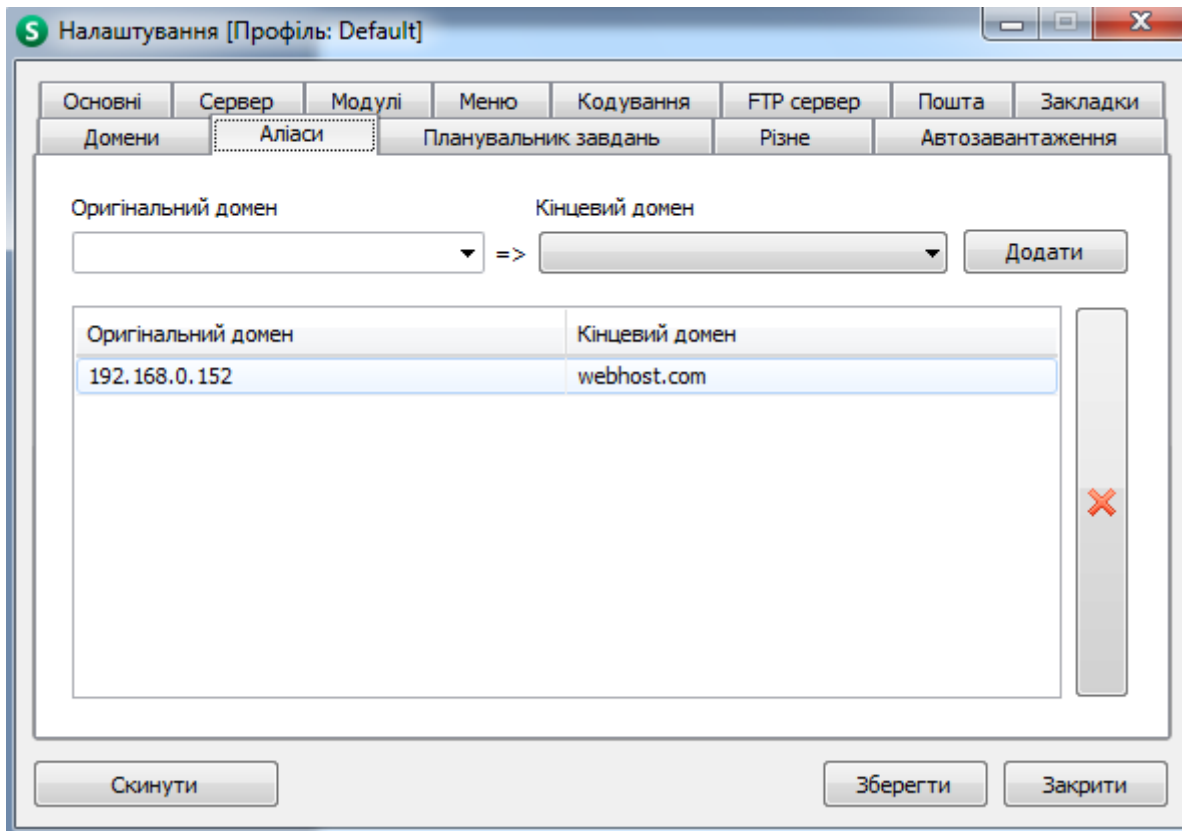


Рис. 3.34. Пункт «Аліаси» в головному меню OpenServer

Запускаємо OpenServer. У вкладці "Мої сайти" перевіряємо наявність сайту.

Тепер при переході в мережі з іншої машини на адресу 192.168.0.152 ми відкриваємо наш сайт (Рис. 3.35.)



Рис. 3.35. Сторінка сайту при переході з мобільного пристрою

Для зручності можна встановити перехід не за мережевою адресою, а за доменним іменем (webhost.com). Для цього переходимо в каталог, який має таку кореневу структуру: C:\Windows\System32\drivers\etc (за умови, що диск C є системним). Знаходимо файл hosts, робимо його копію (на випадок, якщо доведеться робити відкат до попередніх налаштувань). Відкриваємо за допомогою блокнота та додаємо такий рядок після localhost: 192.168.0.152 webhost.com (Рис. 3.36.)

```

1 # Start Open Server f578989365e751ea6bd588207f6a6257_hosts
2 192.168.0.152 webhost.com
3 192.168.0.152 webhost
4 # End Open Server f578989365e751ea6bd588207f6a6257_hosts
5 # Copyright (c) 1993-2009 Microsoft Corp.
6 #
7 # This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
8 #
9 # This file contains the mappings of IP addresses to host names. Each
10 # entry should be kept on an individual line. The IP address should
11 # be placed in the first column followed by the corresponding host name.
12 # The IP address and the host name should be separated by at least one
13 # space.
14 #
15 # Additionally, comments (such as these) may be inserted on individual
16 # lines or following the machine name denoted by a '#' symbol.
17 #
18 # For example:
19 #
20 #     102.54.94.97       rhino.acme.com          # source server
21 #     38.25.63.10      x.acme.com              # x client host
22 # localhost name resolution is handled within DNS itself.
23 #   127.0.0.1         localhost
24 #   ::1              localhost
25 192.168.0.152      webhost.com
26

```

Рис. 3.36. Редагування файлу etc.

Аналогічну процедуру повторюємо на іншому комп'ютері у мережі. Тепер під час введення у рядок пошуку запиту webhost.com відкривається сторінка з корпоративним сайтом (Рис. 3.37.; Рис. 3.38.)

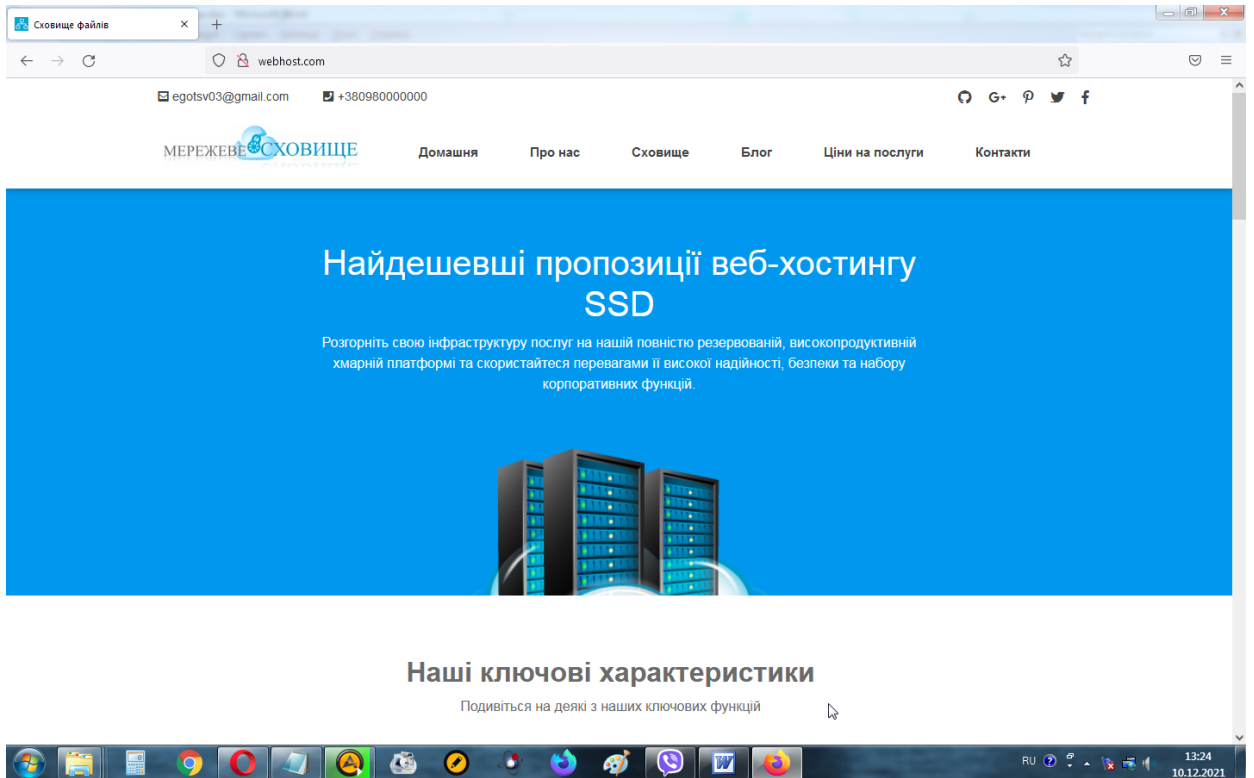


Рис. 3.37. Вигляд сайту при відкритті з головної машини

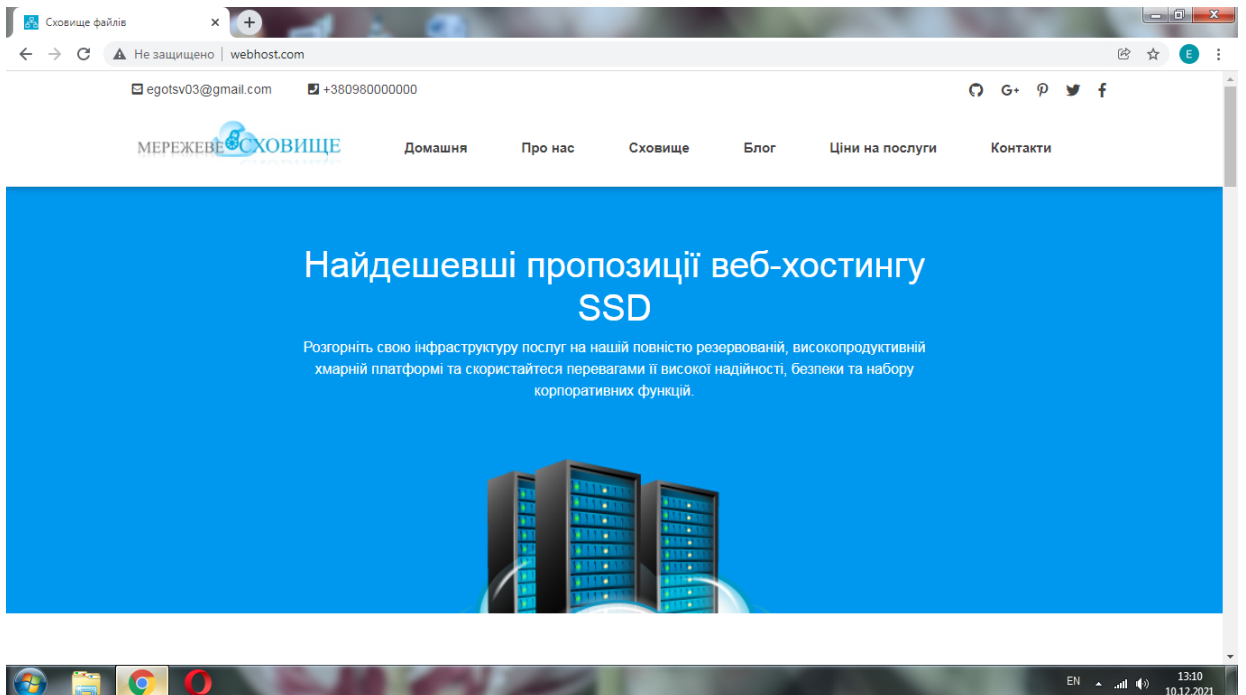


Рис. 3.38. Вигляд сайту при відкритті з допоміжної машини.

Висновки до розділу 3

Так як дана робота є демонстраційною, автор вважає, що достатньо буде показати налаштування сервера, який працюватиме в межах корпоративної мережі.

Для організації такої роботи нам знадобиться спеціальне програмне забезпечення під назвою OpenServer: портативне програмне середовище, створене спеціально для веб-розробників. Це програмне середовище має ряд функцій, серед яких ми будемо використовувати створення локального сервера. Цей програмний комплекс є безкоштовним і в порівнянні з аналогічними комплексами (Denwer, WampServer, XAMPP) має головну перевагу – регулярне оновлення.

Цю процедуру можна повторити і на пристроях, що працюють на OS Android, проте редагування файлу hosts вимагає наявності root-прав, що дещо виходить за рамки даної роботи.

При великій кількості корпоративних машин у мережі, має сенс зробити переадресацію на вказану IP-адресу в налаштуваннях роутера. Цією процедурою займається системний адміністратор.

ВИСНОВКИ

Метою кваліфікаційної роботи була розробка сайту для локальної комп'ютерної мережі. Для досягнення поставленої мети було вирішено наступні завдання:

1. обрано мережеву архітектуру для комп'ютерної мережі, метод доступу, топологію, тип кабельної системи;
2. вибрано спосіб керування мережею;
3. підібрані конфігурації мережевого обладнання;
4. розроблено рекомендації з управління мережевими ресурсами та користувачами мережі;
5. вивчено питання безпеки мережі;
6. спроектовано та реалізовано сайт для внутрішнього користування;
7. встановлено та налаштовано локальний сервер.

Було розроблено раціональну, гнучку структурну схему мережі організації, передбачено режими швидкого оновлення оперативної інформації на сервері.

На сьогоднішній день розробка та впровадження локальних інформаційних мереж є одним із найцікавіших та найважливіших завдань у галузі інформаційних технологій. Виникає потреба у використанні нових технологій передачі. Інтенсивне використання інформаційних технологій вже зараз є найсильнішим аргументом у конкурентній боротьбі, що розгорнулася на світовому ринку.

Реалізовано технологію Fast Ethernet 100 Base TX (як середовище передачі використовується неекранована кручена пара категорії 5). Для зручності прокладання кабелю та його структуризації використовується структурована кабельна система. Є можливість розширення мережі, в комутаторі залишаються не задіяні порти. За потреби можна передбачити додаткові місця підключення робочих станцій (додаткові розетки), так що підключення робочих станцій до

мережі визначатиметься часом налаштування мережевого програмного забезпечення.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ПОСИЛАНЬ

1. Власов К.А. Файловая система: термины и понятия // Total Commander: эффективная работа с файлами и архивами. — БХВ-Петербург, 2012. — 752 с.
2. Когаловский М. Р. Энциклопедия технологий баз данных. — М.: Финансы и статистика, 2002. — 800 с.
3. "Everything is a File". Behavior Genetics Association. Archived from the original on March 10, 2012. Retrieved April 30, 2021.
4. "Close-To-Open Cache Consistency in the Linux NFS Client". Citi.umich.edu. Retrieved 19 November 2014.
5. Network Working Group RFC 4511". IETF.org. 2006-06-01. Retrieved 2014-04-04
6. Arkills, B (2003). LDAP Directories Explained: An Introduction and Analysis. Addison-Wesley Professional.
7. "DNSSEC on all root servers". The H. 2010-05-06
8. Carter, G (2003). LDAP System Administration. O'Reilly Media
9. Carter, Gerald (2003). LDAP System Administration. O'Reilly Media
10. <https://support.microfocus.com/kb/doc.php?id=7007292#>
11. Дейт К. Дж. Введение в системы баз данных = Introduction to Database Systems. — 8-е изд. — М.:Вильямс, 2005. — 1328 с
12. Коннолли Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика = DatabaseSystems: A Practical Approach to Design, Implementation, and Management. — 3-е изд. — М.:Вильямс, 2003. — 1436 с.

13. "Everything is a File". Behavior Genetics Association. Archived from the original on March 10, 2012. Retrieved April 30, 202
14. Close-To-Open Cache Consistency in the Linux NFS Client". Citi.umich.edu. Retrieved 19 November 2014
15. Surfing the High Tech Wave: A story of Novell's early years, 1980–1990 – Roger Bourke White Jr.'s self-published history
16. Rhoton, J (1999). Programmer's Guide to Internet Mail: SMTP, POP, IMAP, and LDAP. Elsevier
17. Exchange Server build numbers and release dates". Microsoft. Retrieved February 19, 2018
18. J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman, and B. Weihl. "Globally Distributed Content Delivery, IEEE Internet Computing, September/October 2002, pp. 50-58"
19. Elizabeth Feinler, IEEE Annals, 3B2-9 man2011030074.3d 29/7/011 11:54 Page 74
20. Geoff Huston (October 2015). "Happy Eyeballs for the DNS, (see slide 37)"
21. Lindsay, David (2007). International Domain Name Law: ICANN and the UDRP. Bloomsbury Publishing. p. 8.
22. Фримен Эрик, Фримен Элизабет. Изучаем HTML, XHTML и CSS = Head First HTML with CSS & XHTML. — П.: «Питер», 2010. — 656 с.
23. <http://www.ibm.com/developerworks/ru/library/wa-bootstrap/>
24. <https://trends.builtwith.com/cdn/BootstrapCDN>
25. <https://fontawesome.com>
26. <https://owlcarousel2.github.io/OwlCarousel2/>
27. <https://fonts.google.com>
28. <https://coderoad.ru/52824839/Почему-popper-js-не-работает-в-laravel>

29. Dirksen, Jos (2013). Learning Three.js: The JavaScript 3D Library for WebGL. UK: Packt Publishing.
30. "Right-click "protection"? Forget about it". June 17, 2008
31. https://ru.wikipedia.org/wiki/Xinuos_OpenServer
32. Кузнецов С. Д. Основы баз данных. — 2-е изд. — М.: Интернет-университет информационных технологий; БИНОМ. Лаборатория знаний, 2007. — 484 с.