

Міністерство освіти і науки України
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій

Методичні вказівки до виконання лабораторних робіт
з дисципліни
«Організація серверів глобальних мереж»

для студентів
спеціальності 123 «Комп'ютерна інженерія»

Тернопіль 2025

Мельник Г.М. Методичні вказівки до проведення лабораторних занять з дисципліни «Організація серверів глобальних мереж» для студентів спеціальності 123 «Комп'ютерна інженерія». Тернопіль: ЗУНУ, 2025. 28 с.

Укладач: Мельник Г.М., к.т.н., доцент кафедри комп'ютерної інженерії

Відповідальний за випуск: Дубчак Л.О., к.т.н., доцент, завідувач кафедри комп'ютерної інженерії

Рецензенти:

Микитишин А. Г. доц., к. т. н., завідувач кафедри комп'ютерно-інтегрованих технологій ТНТУ ім. Івана Пулюя

Манжула В.І. доц., д.т.н, доцент кафедри комп'ютерних наук
Західноукраїнського національного університету

Методичні рекомендації розглянуто та рекомендовано до друку на засіданні кафедри комп'ютерної інженерії
протокол № 7 від 28 лютого 2025 р.

ЗМІСТ

Лабораторна робота №1 Створення середовища розробки сайтів	4
1. Теоретичні відомості.	4
2. Хід роботи	5
3. Зміст звіту	10
4. Контрольні питання.....	10
5. Довідкова інформація.....	10
Лабораторна робота №2. Тестування вебсерверів	12
1. Теоретичні відомості.	12
2 Хід роботи	14
3 Зміст звіту	18
4 Контрольні питання.....	18
Лабораторна робота 3. Розгортання системи дистанційного навчання.....	19
1. Теоретичні відомості.	19
2 Хід роботи	21
3 Зміст звіту	23
4 Контрольні питання.....	23
Лабораторна робота 4. Розгортання сервера інтеграції даних IoT.....	24
1. Теоретичні відомості.	24
2 Хід роботи	26
3. Зміст звіту	28
4. Контрольні питання.....	28

Лабораторна робота №1

Створення середовища розробки сайтів

Мета заняття: Розгорнути середовище розробки сайтів.

Отримати навички налаштування середовищ розробки сайтів

1. ТЕОРЕТИЧНІ ВІДОМОСТІ.

1. Основні вимоги до запуску веб-проєкту

Для розгортання веб-сайту необхідно мати **веб-хостинг** та **доменне ім'я**, що зв'язує сайт із відповідною IP-адресою веб-сервера. Якщо веб-сайт є **динамічним**, а не статичним HTML-документом, сервер має підтримувати **мову серверного програмування** (PHP, Python, Java, .NET) та **систему управління базами даних (СУБД)** (MySQL, PostgreSQL, Oracle, Microsoft SQL Server).

Зазвичай для запуску веб-проєктів використовують готові серверні збірки, які поєднують ключові компоненти в єдине середовище. Найбільш популярною серед них є **LAMP**—набір відкритого ПЗ, що включає:

- Linux операційну систему,
- Apache веб-сервер,
- MySQL/MariaDB СУБД,
- PHP/Python/Perl інтерпретатор мови програмування.

Аналогічні збірки існують і для інших платформ: **WAMP** (Windows), **MAMP** (Mac OS), **XAMPP** (кросплатформова версія).

2. Віртуальний хостинг та серверне середовище

Віртуальний хостинг дозволяє одному фізичному серверу обслуговувати кілька веб-сайтів, кожен із власним доменним іменем. Це досягається завдяки відповідним налаштуванням веб-серверів, таких як Apache, NGINX або Microsoft IIS.

Окрім традиційних серверних стеків, сьогодні все частіше використовують контейнеризацію (наприклад, Docker), що забезпечує ізольоване середовище для кожного веб-застосунку.

3. XAMPP – кросплатформова збірка веб-сервера

XAMPP — це популярна серверна платформа для розробників, яка містить всі необхідні компоненти для запуску локального веб-сервера. Вона включає:

- Apache HTTP Server – для обслуговування статичних та динамічних сторінок.
- MySQL/MariaDB – для роботи з базами даних.
- PHP та Perl – серверні мови програмування.
- FTP-сервер – для управління файлами.
- phpMyAdmin – для адміністрування баз даних через веб-інтерфейс.
- Apache Tomcat – для підтримки Java-додатків (JSP/Servlets).

XAMPP забезпечує швидке розгортання локального сервера без необхідності налаштовувати кожен компонент окремо.

4. Системи керування вмістом (CMS)

Система керування вмістом (CMS, Content Management System) – це програмне забезпечення для створення, управління та оновлення веб-сайтів без потреби в глибоких знаннях програмування.

Види CMS:

1. Системи керування веб-контентом (WCMS) – для управління веб-сайтами, блогами, форумами (WordPress, Joomla, Drupal).
2. Транзакційні CMS – для роботи з електронною комерцією (Magento, OpenCart, PrestaShop).
3. Документоорієнтовані CMS – для управління корпоративними документами (SharePoint, Alfresco).
4. Електронні бібліотеки – для керування мультимедійними ресурсами (DAM-системи).
5. Освітні CMS – для онлайн-курсів (Moodle, Open edX).

Популярні CMS з українською локалізацією:

- Moodle – використовується в навчальних закладах.
- MaxSite CMS – легка CMS на CodeIgniter.
- Joomla 5.2 – гнучка платформа для розробки веб-додатків.

Корпоративні CMS (Enterprise CMS) це системи для великих організацій, які дозволяють керувати внутрішньою та зовнішньою документацією, автоматизувати бізнес-процеси (наприклад, OpenText, IBM FileNet, Microsoft SharePoint).

Для розгортання та налаштування веб-сервера необхідно встановити відповідне серверне середовище, наприклад, XAMPP, відповідно до заданого варіанта конфігурації. Якщо ж метою є розгортання комплексного серверного середовища з підтримкою різних сервісів, рекомендується встановити всі необхідні програмні продукти та серверні компоненти.

Для створення повноцінного серверного середовища можна застосовувати технології віртуалізації, такі як VirtualBox, VMware, або засоби контейнеризації, наприклад, Docker та Kubernetes. Це дозволить гнучко керувати ресурсами, ізолювати сервіси та забезпечити масштабованість системи.

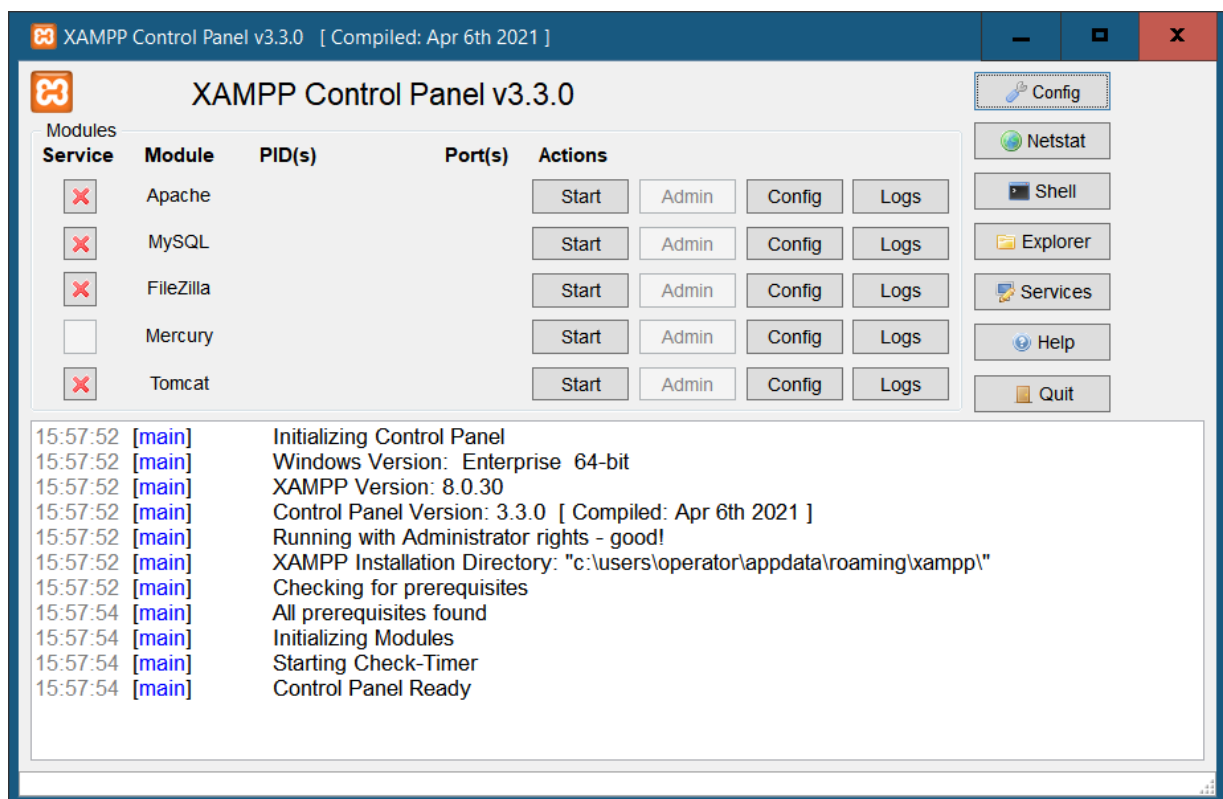
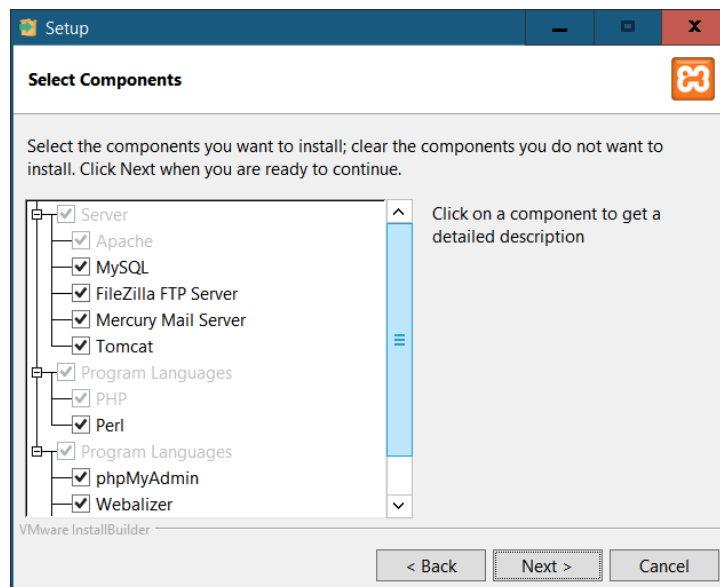
Окрім встановлення самого серверного середовища, важливо забезпечити налаштування баз даних (MySQL, PostgreSQL), веб-серверів (Apache, Nginx), засобів безпеки (SSL, брандмауери) та механізмів автоматизації (CI/CD, скрипти розгортання). Це дозволить не лише розгорнути сервер для розробки та тестування, а й створити стабільну та безпечну інфраструктуру для продуктивного використання.

2. ХІД РОБОТИ

Завдання 1. Встановити середовище веб-сервера

Налаштування XAMPP

1. Завантаження XAMPP [Apache Friends](#)
2. Вибір компонентів: залиште всі компоненти за замовчуванням або оберіть ті, які вам потрібні (наприклад, Apache, MySQL, PHP, phpMyAdmin). Натисніть "Next"



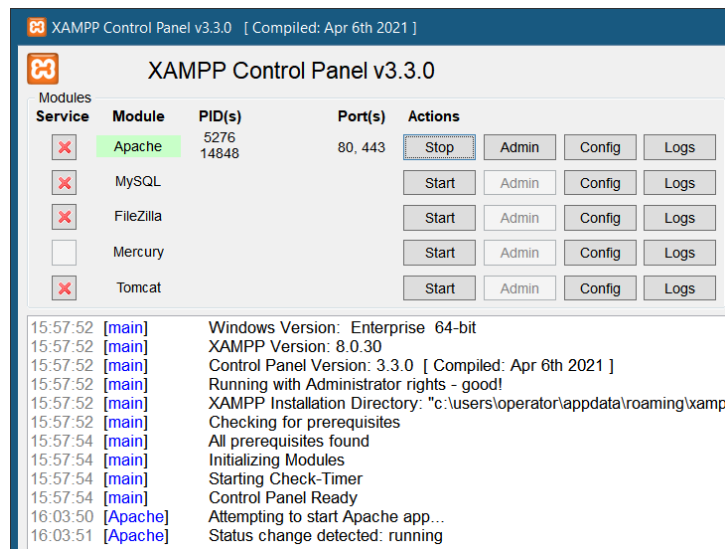
Первинна конфігурація XAMPP

1. Запуск XAMPP Control Panel:

- Якщо панель керування XAMPP не запустилася автоматично, відкрийте її вручну, запустивши `xampp-control.exe` з директорії інсталяції (наприклад, `C:\xampp`).

2. Запуск служб:

- У панелі керування натисніть кнопку "Start" навпроти Apache та MySQL, щоб запустити ці служби. У колонці "Module" повинні з'явитися зелені маркери, що свідчать про успішний запуск. Також ОС запитає дозволу замінити налаштування брандмауера.



3. Перевірка роботи Apache: Відкрити веб-браузер і ввести в адресному рядку `http://localhost`. Ми повинні побачити вітальну сторінку XAMPP.



Welcome to XAMPP for Windows 8.0.30

You have successfully installed XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components. You can find more info in the [FAQs](#) section or check the [HOW-TO Guides](#) for getting started with PHP applications.

4. Перевірка роботи MySQL:
- У панелі керування XAMPP натисніть кнопку "Admin" праворуч від MySQL. Відкриється phpMyAdmin - веб-інтерфейс для управління базами даних MySQL.
5. Налаштування безпеки:
- Рекомендується налаштувати паролі для користувача root у MySQL та інших служб для забезпечення безпеки.
 - Перейдіть у phpMyAdmin, виберіть "User accounts", знайдіть користувача root і змініть пароль.

Додаткові налаштування:

- Файловий менеджер:
 - Всі веб-файли повинні зберігатися у папці `htdocs` в директорії інсталяції XAMPP (наприклад, `C:\xampp\htdocs`).
- Конфігураційні файли:

- Основні конфігураційні файли для Apache (`httpd.conf`), PHP (`php.ini`) та MySQL (`my.ini`) знаходяться в папці `xampp`.

3. Запуск і зупинка служб:

- використовується панель керування XAMPP.

Встановлення WordPress

Welcome to WordPress. Before getting started, you will need to know the following items.

1. Назва бази даних
2. Ім'я користувача бази даних
3. Пароль бази даних
4. Сервер бази даних
5. Табличний префікс (якщо ви хочете запустити більше ніж один WordPress сайт на одній базі даних)

This information is being used to create a `wp-config.php` file. **If for any reason this automatic file creation does not work, do not worry. All this does is fill in the database information to a configuration file. You may also simply open `wp-config-sample.php` in a text editor, fill in your information, and save it as `wp-config.php`.** Need more help? [Read the support article on wp-config.php](#).

In all likelihood, these items were supplied to you by your web host. If you do not have this information, then you will need to contact them before you can continue. If you are ready...

[Вперед!](#)

Встановлення WordPress

Табличний префікс (Table Prefix) — це префікс, який додається до назв всіх таблиць бази даних, створених під час встановлення WordPress. Він потрібен, щоб уникнути конфліктів з іншими таблицями в тій же базі даних, якщо в ній є кілька екземплярів WordPress або інших додатків. Якщо ви вказати префікс `wp_`, то назви таблиць будуть виглядати так: `wp_posts`, `wp_users`, `wp_options`.

Нижче ви маєте ввести свої деталі підключення до бази даних. Якщо ви не впевнені, зв'яжіться зі своїм хостинг-провайдером.

Назва бази даних	wordpress	Назва бази даних, яку ви хочете використовувати з WordPress.
Ім'я користувача	root	Ваше ім'я користувача бази даних.
Пароль	пароль Показати	Ваш пароль бази даних.
Хост бази даних	localhost	You should be able to get this info from your web host, if localhost does not work.
Табличний префікс	wp_	Якщо ви хочете мати одну базу даних для кількох інсталяцій WordPress, змініть це.

[Відправити](#)

Встановлення WordPress

Необхідна інформація

Будь ласка, надайте наступну інформацію. Не хвилюйтесь, ви завжди зможете змінити ці налаштування пізніше.

Назва сайту

Ім'я користувача
Імена користувачів можуть містити тільки букви, цифри, пробіли, нижні лінії, дефіси, крапки, та символ @.

Пароль
Дуже слабкий [Show](#)

Важливо: Вам буде потрібен цей пароль, щоб увійти. Будь ласка, зберігайте його в безпечному місці.

Підтвердіть пароль ☒ Дозволити використання слабкого паролю.

Ваш e-mail
Двічі перевірте свою e-mail адресу перед тим, як продовжити.

Видимість для пошукових систем ☒ Запропонувати пошуковим системам не індексувати цей сайт
Пошукові системи можуть ігнорувати цей запит.

[Встановити WordPress](#)

Вітаємо!

WordPress встановлено. Насолоджуйтесь!

Ім'я користувача first

Пароль *Ваш вказаний пароль.*

[Увійти](#)

Ім'я користувача або Email адреса

first

Пароль

••••

☐ Запам'ятати мене

[Увійти](#)

Втратили свій пароль?

[← Назад до myfirstsite](#)



Українська

[Змінити](#)

localhost/wordpress/wp-admin/

myfirstsite 5 0 + Додати

Майстерня

Головна
Оновлення 1

Записи
Медіафайли
Сторінки
Коментарі
Вигляд
Плагіни
Користувачі
Інструменти
Налаштування
Згорнути меню

Майстерня

Ласкаво просимо до WordPress!

[Дізнайтесь більше про версію 6.5.3.](#)

Author rich content with blocks and patterns

Block patterns are pre-configured block layouts. Use them to get inspired or create new pages in a flash.

[Додати нову сторінку](#)

Customize your entire themes

Design everything on your site - the footer, all using blocks and

[Відкрити редактор сайту](#)

Встановлення WordPress

Завдання 2. Створення контенту

З допомогою системи керування вмістом створити статті про академічну групу і власний блог.

3. ЗМІСТ ЗВІТУ

1. Титульний лист, тема та мета роботи
3. Скріншоти послідовності роботи
4. Конфігураційні файли серверів
- 5 Висновки.

4. КОНТРОЛЬНІ ПИТАННЯ

1. Як працює HTTP протокол
2. Яке призначення та структура LAMP;
3. Основні компоненти дистрибутиву XAMPP;
4. Призначення систем керування вмістом

5. ДОВІДКОВА ІНФОРМАЦІЯ

Таблиця Опис призначення каталогів і файлів у XAMPP 8 версії

Каталог/Файл	Призначення
anonymous\	Каталог для зберігання анонімних FTP файлів.
apache\	Файли конфігурації, модулі та інші Apache.
cgi-bin\	Каталог для зберігання CGI скриптів.
contrib\	Додаткові файли та модулі для XAMPP.
FileZillaFTP\	Файли та конфігурації FTP-сервера FileZilla.
htdocs\	Головний каталог для веб-документів.
img\	Зображення, що використовуються в інтерфейсі XAMPP.
install\	Файли, необхідні для встановлення XAMPP.
licenses\	Ліцензійні угоди для програмного забезпечення XAMPP.
locale\	Файли локалізації для підтримки різних мов.
mailoutput\	Каталог для зберігання вихідних електронних листів.
mailtodisk\	Каталог для зберігання електронних листів на диску.
MercuryMail\	Файли поштового сервера Mercury Mail.
mysql\	Файли MySQL бази даних та конфігураційні файли.
perl\	Файли Perl інтерпретатора та модулів.
php\	Файли PHP інтерпретатора та розширень.
phpMyAdmin\	Веб-інструмент для управління MySQL базами даних.
src\	Вихідні коди для програмних компонентів XAMPP.
tmp\	Тимчасові файли.
tomcat\	Файли Apache Tomcat сервера, конфігураційні
webdav\	Каталог для WebDAV ресурсів.
apache_start.bat	Скрипт для запуску Apache веб-сервера.
apache_stop.bat	Скрипт для зупинки Apache веб-сервера.
catalina_service.bat	для управління Apache Tomcat сервером як сервісом.
catalina_start.bat	Скрипт для запуску Apache Tomcat сервера.
catalina_stop.bat	Скрипт для зупинки Apache Tomcat сервера.

ctlscript.bat	Загальний скрипт для управління всіма серверами в ХАМРР.
filezilla_setup.bat	Скрипт для встановлення FileZilla FTP сервера.
filezilla_start.bat	Скрипт для запуску FileZilla FTP сервера.
filezilla_stop.bat	Скрипт для зупинки FileZilla FTP сервера.
killprocess.bat	Скрипт для завершення процесів, які блокують запуск серверів.
mercury_start.bat	Скрипт для запуску Mercury Mail сервера.
mercury_stop.bat	Скрипт для зупинки Mercury Mail сервера.
mysql_start.bat	Скрипт для запуску MySQL сервера.
mysql_stop.bat	Скрипт для зупинки MySQL сервера.
passwords.txt	Файл для зберігання паролів (для тестових цілей).
properties.ini	Конфігураційний файл для налаштування ХАМРР.
readme_de.txt	Інструкції для користувачів німецькою мовою.
readme_en.txt	Інструкції для користувачів англійською мовою.
service.exe	Виконуваний файл для управління ХАМРР як Windows сервісом.
setup_xampp.bat	Скрипт для первинного налаштування ХАМРР.
test_php.bat	Скрипт для тестування PHP.
uninstall.dat	Файл для видалення ХАМРР.
uninstall.exe	Виконуваний файл для видалення ХАМРР.
xampp-control.exe	Виконуваний файл для запуску ХАМРР Control Panel.
xampp-control.ini	Конфігураційний файл для ХАМРР Control Panel.
xampp_shell.bat	Скрипт для запуску командного рядка ХАМРР.
xampp_start.exe	Виконуваний файл для запуску всіх серверів ХАМРР.
xampp_stop.exe	Виконуваний файл для зупинки всіх серверів ХАМРР.

Лабораторна робота №2.

Тестування вебсерверів

Мета роботи: Аналіз продуктивності вебсерверів.

Набуття навичок навантажувального тестування

1. ТЕОРЕТИЧНІ ВІДОМОСТІ.

Навантажувальне тестування застосовується для оцінки продуктивності інформаційних систем під різними рівнями навантаження. Воно імітує роботу певної кількості користувачів, які одночасно взаємодіють із системою, що дозволяє визначити її стійкість та продуктивність.

Основні поняття

Віртуальний користувач (Virtual User) – ключове поняття тестування, що визначає рівень навантаження на систему. Кожен віртуальний користувач виконує типові дії, відправляючи серверу запити відповідно до протоколів зв'язку (HTTP, ODBC, NCA тощо).

Крива деградації (Degradation Curve) – графік, що відображає залежність продуктивності системи (наприклад, часу відповіді) від зростання навантаження (кількості віртуальних користувачів).

Основні показники продуктивності

Час відповіді (Response Time) – час, необхідний для виконання операції.

Пропускна здатність (TPS – Transactions Per Second) – кількість транзакцій, які система може обробити за секунду.

Результати навантажувального тестування:

- Визначення продуктивності системи.
- Виявлення вузьких місць та їх усунення.
- Оптимізація конфігурації серверів і програмного забезпечення.

Apache JMeter — це інструмент для навантажувального тестування веб-додатків, API, баз даних та інших систем. Він дозволяє створювати тестові сценарії, що імітують поведінку користувачів та аналізують продуктивність сервера.

Створення тестового сценарію JMeter.

1 Додавання Thread Group (Групи потоків)

Thread Group визначає, скільки віртуальних користувачів буде виконувати запити та як часто.

1. Додати новий тестовий план:
 - File → New.
2. Додати Thread Group:
 - Test Plan → Add → Threads (Users) → Thread Group.

Налаштування Thread Group:

- Number of Threads (users) – кількість одночасних користувачів (наприклад, 50).

- Ramp-Up Period (seconds) – час, протягом якого всі користувачі будуть активовані (наприклад, 10 секунд).
- Loop Count – кількість повторів кожного користувача (наприклад, 100 разів).
- Scheduler (опціонально) – для задання часу тестування.

Приклад: 50 користувачів будуть активовані за 10 секунд і кожен виконає 100 запитів.

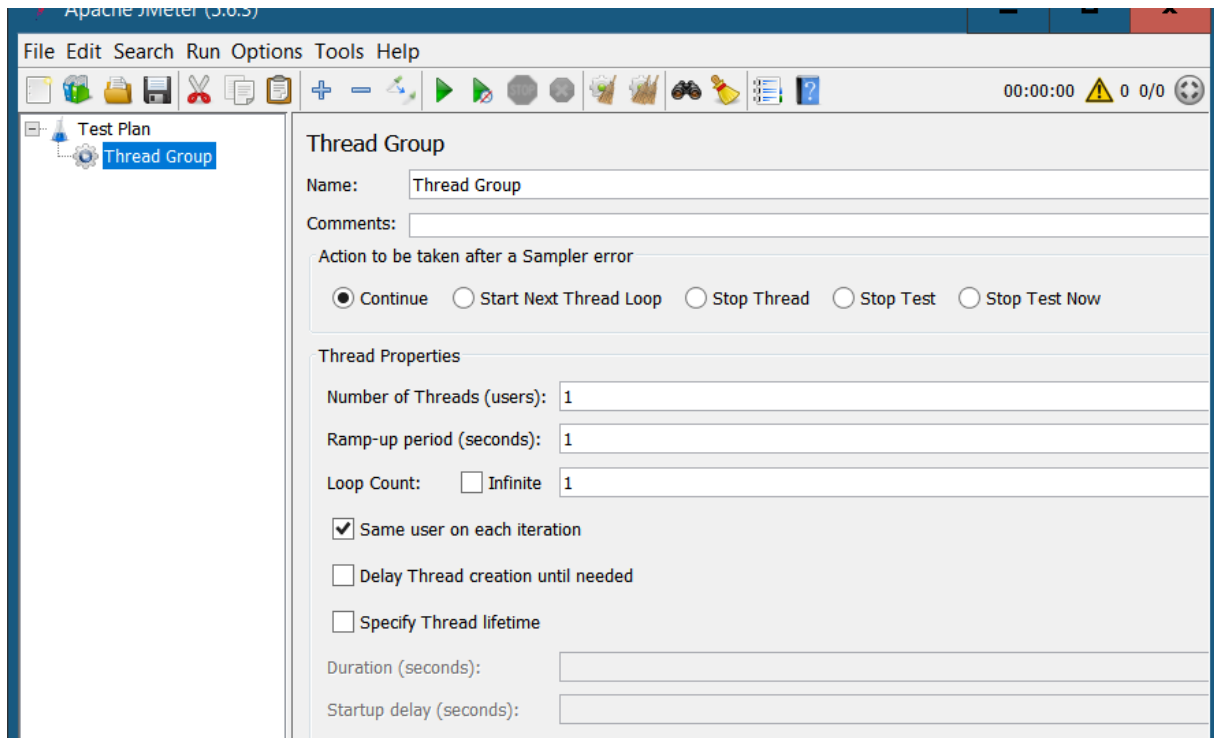


Рисунок 2.1 - Створення «Групи потоків»

2 Додавання HTTP Request Sampler (Запити до сервера)

HTTP Request Sampler відповідає за відправку HTTP-запитів до веб-сервера або API.

1. Додати HTTP-запит:
 - Thread Group → Add → Sampler → HTTP Request.
2. Налаштувати параметри HTTP-запиту:
 - Server Name or IP – адреса сервера (server-ip або example.com).
 - Protocol – вибір http або https.
 - Path – шлях до ресурсу (наприклад, /index.html або /api/users).
 - Method – тип HTTP-запиту (GET, POST, PUT, DELETE).
 - Parameters – параметри запиту (наприклад, id=123).
 - Body Data – якщо POST або PUT, тут можна вказати JSON або форму.

Приклад: GET-запит до `http://server-ip/get_grades.php` для отримання JSON-даних.

3 Додавання Graph Results Listener (Графічне відображення результатів)

Graph Results Listener дозволяє візуалізувати результати тестування у вигляді графіків.

1. Додати Graph Results: Thread Group → Add → Listener → Graph Results.
2. Запустити тест: Натиснути "Start" (зелена кнопка).

3. Аналізувати результати:

- **Average** – середній час відповіді сервера.
- **Median** – медіанний час відповіді.
- **Deviation** – відхилення від середнього значення.
- **Throughput** – кількість запитів у секунду.

Оптимізація: Якщо сервер перевантажений, можна збільшити Ramp-Up Period або зменшити Number of Threads.

Аналіз результатів

- Час відповіді сервера (Response Time) – чи сервер швидко обробляє запити
- Пропускна здатність (Throughput) – скільки запитів обробляється за секунду
- Кількість помилок (Errors) – чи є помилки під час навантаження

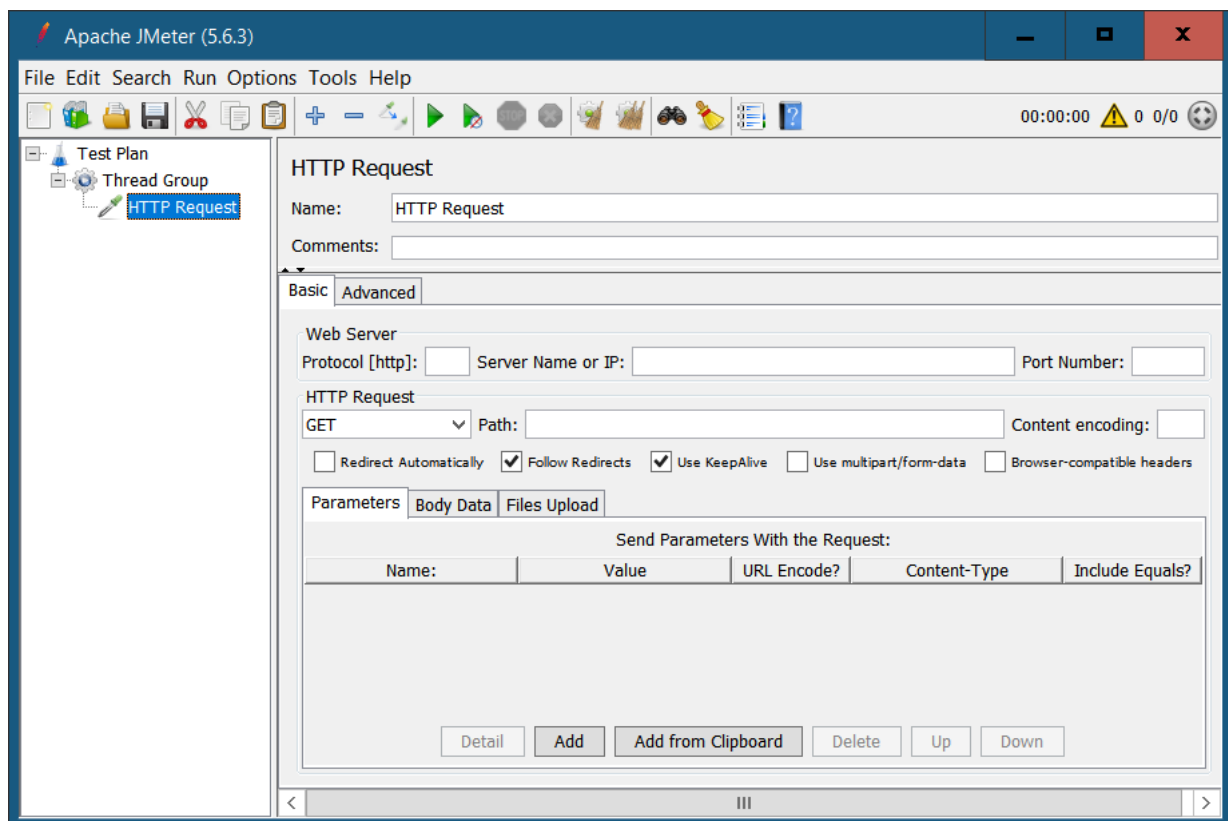


Рисунок 2.2 - Додавання HTTP request

2 ХІД РОБОТИ

Крок 1: розгортання статичного PHP-сайту

1.1 Налаштування веб-сервера Linux (Apache)

Запустіть такий сценарій на сервері Linux, щоб установити та налаштувати Apache:

```
#!/bin/bash
# Install Apache and required utilities
echo "Installing Apache and system utilities..."
sudo apt update
sudo apt install -y apache2
```

```
# Enable Apache and allow firewall rules
echo "Enabling Apache service..."
sudo systemctl enable --now apache2
sudo ufw allow in "Apache"

# Create a basic static webpage
echo "Creating static website..."
echo "<h1>Welcome to the Static Web Server</h1>" | sudo tee
/var/www/html/index.html

# Restart Apache to apply changes
sudo systemctl restart apache2

echo "Static website is available at http://$(hostname -I | awk '{print $1}')"
```

Перевірка

Відкрийте `http://server-ip/` у веб-браузері, щоб переконатися, що статичний сайт працює.

Крок 2. Розгортання динамічного PHP-сайту

Для тестового сайту згенеруємо базу даних оцінок студентів.

2.1 Встановіть Apache, PHP і MySQL

Запустіть такий сценарій, щоб налаштувати Apache, PHP і MySQL:

```
#!/bin/bash
# Встановити Apache, PHP, and MySQL server
echo "Installing Apache, PHP, and MySQL..."
sudo apt update
sudo apt install -y apache2 php php-mysql mariadb-server

# Enable services
echo "Enabling Apache and MySQL..."
sudo systemctl enable --now apache2 mariadb

# Secure MySQL installation (manual step recommended)
echo "Run 'sudo mysql_secure_installation' to set up MySQL security."

echo "Installation complete!"
```

2.2 Налаштувати базу даних оцінок студентів

Запустіть такий сценарій SQL, щоб створити базу даних, користувачів і таблиці:

```
-- Create database
CREATE DATABASE student_db;

-- Create user and grant privileges
CREATE USER 'student_user'@'localhost' IDENTIFIED BY 'password123';
GRANT ALL PRIVILEGES ON student_db.* TO 'student_user'@'localhost';
FLUSH PRIVILEGES;
```

```
-- Create table for student grades
USE student_db;
CREATE TABLE grades (
    id INT AUTO_INCREMENT PRIMARY KEY,
    student_name VARCHAR(100) NOT NULL,
    subject VARCHAR(100) NOT NULL,
    score INT CHECK (score BETWEEN 0 AND 100),
    exam_date DATE
);

-- Insert test data
INSERT INTO grades (student_name, subject, score, exam_date) VALUES
('Оксана Петренко', 'Математика', 85, '2024-05-10'),
('Сергій Коваль', 'Фізика', 78, '2024-05-12'),
('Петро Палилюлько', 'Хімія', 92, '2024-05-14'),
('Петро Мишин', 'Біологія', 67, '2024-05-15');
```

Щоб виконати цей сценарій на сервері:

```
mysql -u root -p < student_grades.sql
```

2.3 Створіть PHP API для отримання оцінок студентів

Розгорніть такий сценарій PHP (/var/www/html/get_grades.php):

```
<?php
$servername = "localhost";
$username = "student_user";
$password = "password123";
$dbname = "student_db";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Query to fetch student grades
$sql = "SELECT student_name, subject, score, exam_date FROM grades";
$result = $conn->query($sql);

$grades = array();
if ($result->num_rows > 0) {
    while ($row = $result->fetch_assoc()) {
        $grades[] = $row;
    }
}

// Output JSON response
```

```
header('Content-Type: application/json');
echo json_encode($grades);

$conn->close();
?>
```

Перевірка:

Відкрийте `http://server-ip/get_grades.php` і переконайтеся, що результат має формат JSON.

Крок 3: Завантажте тестування з клієнта Windows

3.1 Встановіть засоби тестування в Windows

1. Встановіть Apache Benchmark (ab)
 - о Якщо у вас встановлено Apache, `ab.exe` включено в каталог `bin/`.
 - о Крім того, використовуйте WSL (підсистема Windows для Linux).
2. Встановіть JMeter
 - о Завантажте з [Apache JMeter](#)
 - о Розпакуйте та запустіть `jmeter.bat`.

3.2 Тестування навантаження статичної сторінки за допомогою Apache Benchmark

1. Запустіть простий тест із 1000 запитами та 50 одночасними користувачами:
 2. `ab -n 1000 -c 50 http://server-ip/index.html`
 3. Проаналізуйте результат: подивіться на час відповіді, частоту відмов і кількість запитів за секунду.

3.3 Навантажувальне тестування Dynamic PHP API

Перевірте продуктивність PHP API на базі даних:

```
ab -n 500 -c 20 http://server-ip/get_grades.php
```

Проаналізуйте результат, щоб порівняти час відповіді зі статичною сторінкою.

3.4 Тестування навантаження за допомогою JMeter (необов'язково)

1. Відкрийте JMeter.
2. Створіть групу потоків “**Thread Group**” із:
 - о Кількість користувачів: 50, Кількість петель (Loop): 100
3. Додайте вибірку запитів HTTP “**HTTP Request Sampler**”:
 - о Метод: GET
 - о URL: `http://server-ip/get_grades.php`
4. Додайте прослуховувач результатів графіка (Graph Results Listener).
5. Запустіть тест і проаналізуйте час відповіді.

Крок 4: Моніторинг продуктивності сервера

4.1 Відстежуйте використання процесора та пам'яті

- Основна статистика (Linux Server) команда `top`

- Більш детальний моніторинг `htop`
- Продуктивність диска та введення/виведення
`iostat -x 1`

4.2 Моніторинг журналів Apache

- Перегляд запитів у реальному часі:
`tail -f /var/log/apache2/access.log`

Крок 5: Аналіз і звіт про результати

1. Порівняйте продуктивність статичного та динамічного вмісту.
2. Визначте максимальну кількість одночасних користувачів, які може обслуговувати сервер.
3. Проаналізуйте використання процесора, оперативної пам'яті та диска під різними навантаженнями.

3 ЗМІСТ ЗВІТУ

1. Тема та мета лабораторної роботи;
2. Хід роботи із копіями екрану;
3. Висновки по роботі.

4 КОНТРОЛЬНІ ПИТАННЯ

1. Що таке час відповіді сервера (Response Time) ?
2. Що таке пропускна здатність (Throughput) ?
3. Що таке кількість помилок (Errors) ?

Лабораторна робота 3.

Розгортання системи дистанційного навчання

Мета: Встановити систему дистанційного навчання

1. ТЕОРЕТИЧНІ ВІДОМОСТІ.

Moodle (Modular Object-Oriented Dynamic Learning Environment) – це одна з найпопулярніших платформ для дистанційного навчання (LMS – Learning Management System). Вона дозволяє викладачам створювати курси, завдання, форуми, опитування та інші навчальні матеріали, а студентам – взаємодіяти з ними онлайн.

Основні особливості Moodle:

- Відкрите програмне забезпечення (ліцензія GNU GPL), що дозволяє безкоштовно використовувати і модифікувати систему.
- Гнучкість – підтримка модулів і плагінів.
- Підтримка різних методів автентифікації (LDAP, OAuth, Google, Microsoft та інші).
- Можливість масштабування – використання для малих та великих навчальних закладів.
- Інтеграція з іншими сервісами (наприклад, Google Drive, Office 365).

Формати курсів Moodle:

- структура (навчальні модулі без прив'язки до календаря)
- календар (навчальні модулі з прив'язкою до календаря)

Варіюючи поєднання різних елементів курсу, викладач організує вивчення матеріалу таким чином, щоб форми навчання відповідали цілям та завданням конкретних занять.

Рисунк 3.1 – Сторінка курсу в режимі редагування

Елементи Moodle-курсу можна розділити на дві частини:.

- Resource/Ресурси – це теоретичні матеріали.
- Activity/Діяльність/Активності – засоби організації самостійної роботи

До ресурсів належать: Текстова сторінка, Веб сторінка, Посилання на файл або веб-сторінку, Посилання на каталог, Пояснення.

Інтерактивні елементи курсу: Лекція, Завдання, Тест, Робочий зошит, Глосарій, Форум, Чат, Опитування, Анкета.

Додати діяльність або ресурс

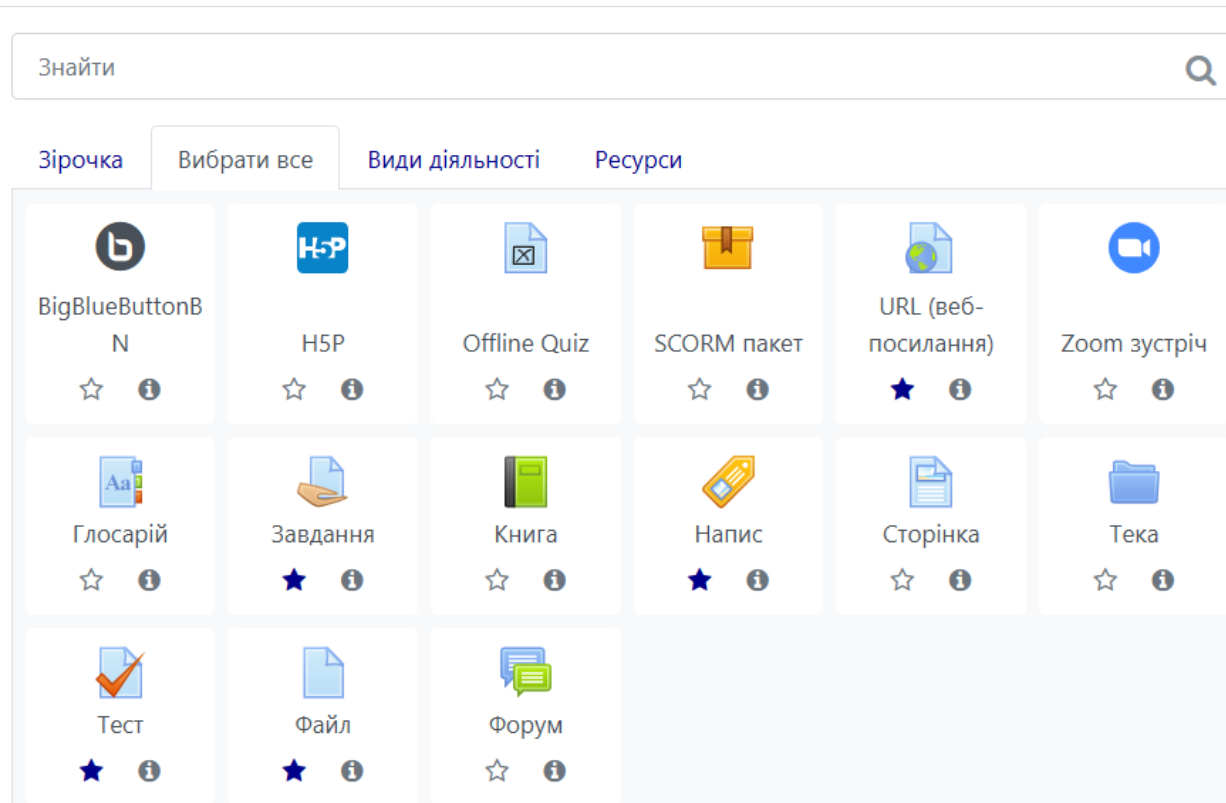


Рисунок 3.2 – Додавання діяльності або ресурсу

Вимоги для розгортання Moodle

Мінімальні системні вимоги:

- Операційна система: Ubuntu 20.04+.
- Веб-сервер: Apache або Nginx.
- База даних: MySQL/MariaDB або PostgreSQL.
- PHP: версія 7.4 або вище (рекомендована – 8.0+).
- Додаткові бібліотеки: PHP-розширення (gd, intl, xml, zip, soap, mbstring, curl, json).

Архітектура розгортання:

1. Веб-сервер (Apache/Nginx) – обробляє HTTP-запити користувачів.
2. PHP – виконує серверний код Moodle.
3. СУБД (MySQL, MariaDB, PostgreSQL) – зберігає курси, користувачів та контент.
4. Файлова система – зберігає завантажені матеріали.
5. SSL-сертифікат – для захищеного з'єднання HTTPS.

Окрім традиційного встановлення на сервері Ubuntu, можливі такі варіанти розгортання:

- Docker-контейнери – запуск Moodle у контейнеризованому середовищі.
- Cloud-платформи (AWS, Google Cloud, Azure) – встановлення через керовані сервіси.
- Автоматизоване розгортання через Ansible – для масштабованих середовищ.

2 ХІД РОБОТИ

Завдання 1. Створення середовища

Вивчити системні вимоги поточної версії системи Moodle до дистрибутивів: PHP, MariaDB 10.6.7, MySQL, Postgres, MSSQL 2017, Oracle.

Встановлення необхідних пакетів: Apache, MySQL/MariaDB, PHP

Перед встановленням необхідно оновити пакети та систему:

```
sudo apt update && sudo apt upgrade -y
```

Встановлення веб-сервера Apache

```
sudo apt install apache2 -y
```

Запуск і перевірка статусу сервера:

```
sudo systemctl enable apache2 --now  
sudo systemctl status apache2
```

Встановлення MySQL/MariaDB

```
sudo apt install mysql-server -y
```

Безпечне налаштування MySQL:

```
sudo mysql_secure_installation
```

Включення та перевірка статусу:

```
sudo systemctl enable mysql --now  
sudo systemctl status mysql
```

Встановлення PHP та необхідних модулів

```
sudo apt install php php-mysql php-xml php-curl php-gd php-intl php-mbstring  
php-zip php-soap -y
```

Перевірка версії PHP:

```
php -v
```

Завдання 2. Створення та налаштування бази даних

Вхід до MySQL

```
sudo mysql -u root -p
```

Створення бази даних Moodle

```
CREATE DATABASE moodle DEFAULT CHARACTER SET utf8mb4 COLLATE  
utf8mb4_unicode_ci;
```

Створення користувача та надання прав

```
CREATE USER 'moodleuser'@'localhost' IDENTIFIED BY 'securepassword';  
GRANT ALL PRIVILEGES ON moodle.* TO 'moodleuser'@'localhost' WITH GRANT  
OPTION;
```

```
FLUSH PRIVILEGES;  
EXIT;
```

Завдання 3. Завантаження Moodle

Завантаження Moodle

```
cd /var/www/html  
sudo wget https://download.moodle.org/download.php/stable402/moodle-latest-402.tgz
```

Розпакування та налаштування прав

```
sudo tar -xvzf moodle-latest-402.tgz  
sudo mv moodle /var/www/html/  
sudo mkdir /var/www/moodledata  
sudo chown -R www-data:www-data /var/www/html/moodle /var/www/moodledata  
sudo chmod -R 775 /var/www/html/moodle /var/www/moodledata
```

Завдання 4. Налаштування веб-сервера Apache

Створення нового конфігураційного файлу

```
sudo nano /etc/apache2/sites-available/moodle.conf
```

Додавання конфігурації

```
<VirtualHost *:80>  
    ServerAdmin admin@example.com  
    DocumentRoot /var/www/html/moodle  
    ServerName moodle.example.com  
  
    <Directory /var/www/html/moodle>  
        AllowOverride All  
        Require all granted  
    </Directory>  
  
    ErrorLog ${APACHE_LOG_DIR}/error.log  
    CustomLog ${APACHE_LOG_DIR}/access.log combined  
</VirtualHost>
```

Активація конфігурації та перезапуск Apache

```
sudo a2ensite moodle.conf  
sudo a2enmod rewrite  
sudo systemctl restart apache2
```

Виконання початкової конфігурації через веб-інтерфейс

1. Відкрити у браузері:

`http://server-ip` або `http://moodle.example.com`

2. Пройти кроки встановлення:

- Вибрати тип бази даних (MariaDB/MySQL).
- Ввести облікові дані користувача moodleuser.
- Вибрати каталог для moodledata.
- Завершити установку.

3. Створити адміністративний обліковий запис.

Перевірка коректної роботи системи

1. Авторизуватись у Moodle як адміністратор.
2. Перейти в **Site Administration** → **Notifications**.
3. Переконавшись, що всі компоненти встановлені правильно.
4. Створити тестовий курс і додати користувачів.
5. Налаштувати HTTPS (опціонально):

```
sudo apt install certbot python3-certbot-apache -y  
sudo certbot --apache -d moodle.example.com
```

Завдання 5 Додати власний курс у Moodle

Створити сторінку курсу та завантажити декілька активностей та ресурсів, включаючи тест.

3 ЗМІСТ ЗВІТУ

1. Тема та мета лабораторної роботи;
2. Хід роботи із копіями екрану;
3. Висновки .

4 КОНТРОЛЬНІ ПИТАННЯ

1. Які основні компоненти необхідні для роботи Moodle?
2. Які бази даних підтримуються Moodle?
3. Як налаштувати веб-сервер для роботи з Moodle?
4. Які розширення PHP необхідні для Moodle?
5. Які методи масштабування Moodle ви знаєте?

Лабораторна робота 4.

Розгортання сервера інтеграції даних IoT

Мета роботи: Встановити сервер інтеграції даних IoT

Ознайомитися з платформою для інтеграції та керування IoT-даними.
Вивчити основи налаштування та конфігурації сервера IoT на базі Ubuntu.
Виконати встановлення та запуск ThingsBoard, підключення до бази даних.
Дослідити механізми отримання, обробки та візуалізації IoT-даних.

1. ТЕОРЕТИЧНІ ВІДОМОСТІ.

ThingsBoard – це відкрита платформа IoT для збирання, обробки, збереження та візуалізації даних із пристроїв. Вона підтримує основні протоколи передачі IoT-даних, такі як MQTT, HTTP, CoAP, і надає можливості керування пристроями через панель адміністратора. Особливості:

- Підтримка різних протоколів IoT (MQTT, CoAP, HTTP).
- Збереження IoT-даних у базах даних (PostgreSQL, Cassandra, HSQLDB).
- Гнучкі механізми обробки та маршрутизації даних.
- Вбудована система дашбордів для візуалізації.
- Розмежування ролей користувачів (адміністратори, клієнти, пристрої).
- Масштабованість для великих IoT-мереж.

Сервер ThingsBoard працює у багатокомпонентній архітектурі:

1. IoT-пристрої – сенсори та контролери, що надсилають дані через MQTT, CoAP або HTTP.

2. Message Queue (Kafka, RabbitMQ) – використовується для обробки великих обсягів даних.

3. База даних (PostgreSQL, Cassandra) – зберігає телеметричні дані та метадані.

4. Обробник правил (Rule Engine) – система автоматичного аналізу та обробки даних.

5. Панель керування – веб-інтерфейс для адміністраторів та клієнтів.

Dashboard (панель управління) – це візуальний інтерфейс для моніторингу, аналізу та керування IoT-даними. Він дозволяє користувачам отримувати реальний стан підключених пристроїв у вигляді графіків, таблиць, віджетів, карт та аналітичних звітів. Призначення :

– Візуалізація IoT-даних – графічне відображення даних сенсорів у реальному часі.

– Моніторинг стану пристроїв – можливість відстежувати показники (температура, вологість, тиск, рівень заряду тощо).

– Керування пристроями – можливість надсилати команди та змінювати параметри.

– Аналіз та звітність – використання історичних даних для аналітики.

– Кастомізація інтерфейсу – створення динамічних та інтерактивних віджетів для різних груп користувачів.

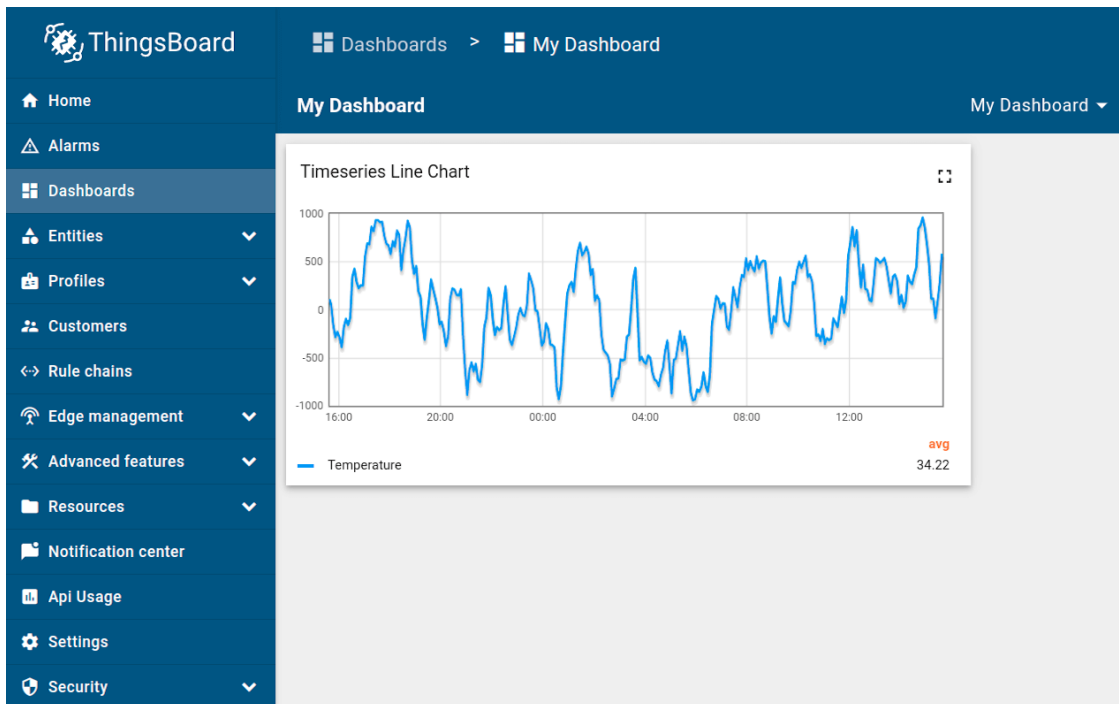


Рисунок 4.1 – Панель керування Dashboard

Основні функції Dashboard

1. Збір та відображення IoT-даних

- Підключення до пристроїв через MQTT, CoAP, HTTP.
- Автоматичне отримання телеметрії у режимі реального часу.
- Відображення значень у вигляді графіків, таблиць, карт, індикаторів.

2. Налаштування взаємодії з пристроями

- Надсилання команд управління на пристрої.
- Використання двостороннього зв'язку для зміни налаштувань у реальному часі.
- Створення кнопок та слайдерів для зміни параметрів.

3. Аналіз історичних даних

- Доступ до історичних даних у базі PostgreSQL/Cassandra.
- Фільтрація та вибір періоду часу для аналізу.
- Використання статистичних алгоритмів.

У ThingsBoard віджети поділяються на категорії:

- Charts – графіки (лінійні, стовпчикові, секторні).
- Cards – карткові інтерфейси для ключових показників.
- Map – інтеграція з OpenStreetMap або Google Maps.
- Control – кнопки, слайдери для управління пристроями.
- Tables – табличні представлення даних.
- Alarm – відображення повідомлень та аварійних ситуацій.

Сутності (Entities) – це пристрої, датчики, користувачі або групи пристроїв.

Псевдоніми (Aliases) – використання груп пристроїв для спрощення налаштувань.

Triggers – механізми автоматичних дій у відповідь на зміну параметрів.

Actions – можливість надсилати команди при зміні параметрів.

Мінімальні вимоги для встановлення ThingsBoard на Ubuntu:

- ОС: Ubuntu 20.04 або новіша версія.
- Java: OpenJDK 11 або вище.
- База даних: PostgreSQL 12+ (або Cassandra).
- RAM: мінімум 2 GB (рекомендовано 4+ GB).
- CPU: 2 ядра.

Основні етапи розгортання ThingsBoard

1. Оновлення системи та встановлення необхідних пакетів.
2. Встановлення та налаштування PostgreSQL для збереження IoT-даних.
3. Завантаження та встановлення ThingsBoard.
4. Конфігурація сервера та запуск служби.
5. Перевірка роботи через веб-інтерфейс.
6. Підключення IoT-пристроїв і тестування передачі даних.

2 ХІД РОБОТИ

1. Оновлення системи та встановлення необхідних пакетів

```
sudo apt update && sudo apt upgrade -y
```

Встановлення необхідного програмного забезпечення:

```
sudo apt install openjdk-11-jdk wget curl -y
```

Перевірка версії Java:

```
java -version
```

2. Встановлення та налаштування PostgreSQL для збереження IoT-даних

2.1 Встановлення PostgreSQL

```
sudo apt install postgresql postgresql-contrib -y
```

Запуск та активація PostgreSQL:

```
sudo systemctl enable postgresql --now  
sudo systemctl status postgresql
```

2.2 Створення бази даних для ThingsBoard

```
sudo -u postgres psql
```

Виконати наступні SQL-команди:

```
CREATE DATABASE thingsboard;  
CREATE USER thingsboard WITH PASSWORD 'securepassword';  
ALTER ROLE thingsboard SET client_encoding TO 'utf8';  
ALTER ROLE thingsboard SET default_transaction_isolation TO 'read  
committed';  
ALTER ROLE thingsboard SET timezone TO 'UTC';  
GRANT ALL PRIVILEGES ON DATABASE thingsboard TO thingsboard;  
\q
```

Переконайтеся, що база даних створена:

```
sudo -u postgres psql -c "\l"
```

Завантаження та встановлення ThingsBoard

Завантаження інсталяційного файлу

wget

```
https://github.com/thingsboard/thingsboard/releases/download/v3.4.2/thingsboard-3.4.2.deb
```

Встановлення пакета

```
sudo dpkg -i thingsboard-3.4.2.deb
```

Налаштування конфігураційного файлу

Редагування файлу налаштувань:

```
sudo nano /etc/thingsboard/conf/thingsboard.yml
```

Додати конфігурацію підключення до PostgreSQL:

datasource:

url: jdbc:postgresql://localhost:5432/thingsboard

username: thingsboard

password: securepassword

Збережіть зміни (Ctrl + X, Y, Enter).

Ініціалізація бази даних

```
sudo /usr/share/thingsboard/bin/install/install.sh --loadDemo
```

Запуск сервісу ThingsBoard

```
sudo systemctl enable thingsboard --now
```

```
sudo systemctl status thingsboard
```

Переконатися, що сервер запущено.

Перевірка роботи через веб-інтерфейс

Відкрити браузер і перейти за адресою:

http://<IP-адреса-сервера>:8080

Використати стандартні облікові дані:

- Логін: sysadmin@thingsboard.org
- Пароль: sysadmin

Завдання 2 Підключення IoT-пристроїв і тестування передачі даних

Створення нового пристрою

1. Перейти у вкладку «Devices».
2. Натиснути «Add new device» та задати ім'я.
3. Після створення перейти у вкладку «Details» та отримати «Access Token».

Відправка тестових даних через MQTT

Встановлення MQTT-клієнта:

```
sudo apt install mosquitto-clients -y
```

Відправка тестових даних:

```
mosquitto_pub -d -h <IP-адреса-сервера> -p 1883 -t v1/devices/me/telemetry -u '<ACCESS_TOKEN>' -m '{"temperature":25}'
```

Перевірка отриманих даних у ThingsBoard.
Створити дашборд для відображення отриманих даних.

3. ЗМІСТ ЗВІТУ

1. Тема та мета лабораторної роботи;
2. Хід роботи із копіями екрану;
3. Висновки .

4. КОНТРОЛЬНІ ПИТАННЯ

1. Які основні компоненти архітектури ThingsBoard?
2. Які протоколи передачі IoT-даних підтримує ThingsBoard?
3. Як налаштувати базу даних PostgreSQL для ThingsBoard?
4. Як підключити IoT-пристрій через MQTT?
5. Як створити власний дашборд для візуалізації даних?

Підписано до друку 28.02.2025.
Формат 60x 84/16. Гарнітура Times New Roman.
Папір офсетний 70 г/м². Друк електрографічний.
Умов.-друк. арк. 1,63. Обл.-вид. арк. 0,84
Тираж 35 примірників. Замовлення № 02/25/2-1.

Видавець та виготувач:

ФОП Осадца Ю. В.
м. Тернопіль, вул. 15 Квітня, 2Д/10
тел. (097) 988-53-23

*Свідоцтво про внесення суб'єкта видавничої справи до державного
реєстру видавців, виготівників і розповсюджувачів видавничої продукції
серія ТР № 46 від 07 березня 2013 р.*