

**ЛАБОРАТОРНА РОБОТА №5.**  
**РОЗГОРТАННЯ ПРОГРАМНОЇ СИСТЕМИ ТА ДЕМОНСТРАЦІЯ ЇЇ**  
**РОБОТИ**

**Система моніторингу опалення у домогосподарствах**

Версія 1.0, затверджена

Підготовлено Макогоном Б.О.

ПЗПІ-22-6

30.12.2024

Розгортання системи на сервері передбачає налаштування середовища для роботи бази даних, бекенду та взаємодії з модулями Arduino. Ось покроковий опис, як це реалізувати.

## **1. Вибір серверної платформи**

Для розгортання системи можна використовувати такі сервери:

- Хмарні платформи: AWS, Google Cloud, Azure, DigitalOcean.
- Віртуальні приватні сервери (VPS): Hetzner, Linode.
- Фізичні сервери: Розміщені у власній інфраструктурі.

## **2. Налаштування бази даних**

### **1. Вибір бази даних:**

- У проекті передбачена робота з MySQL. Її потрібно встановити на сервері.

### **2. Встановлення MySQL:**

- Для Linux (Ubuntu):

```
```bash
sudo apt update
sudo apt install mysql-server
...`
```

- Запустіть MySQL:

```
```bash
sudo systemctl start mysql
...`
```

### **3. Налаштування бази даних:**

- Увійдіть у MySQL:

```
```bash
mysql -u root -p`
```

```
'''
```

- Створіть базу даних і користувача:

```
'''sql
```

```
CREATE DATABASE sensor_system;
```

```
CREATE USER 'sensor_user'@'%' IDENTIFIED BY 'strongpassword';
```

```
GRANT ALL PRIVILEGES ON sensor_system.* TO 'sensor_user'@'%';
```

```
FLUSH PRIVILEGES;
```

```
'''
```

#### 4. Підключення до бази в Spring Boot:

- Налаштуйте файл `application.properties` або `application.yml`:

```
'''properties
```

```
spring.datasource.url=jdbc:mysql://your-server-ip:3306/sensor_system
```

```
spring.datasource.username=sensor_user
```

```
spring.datasource.password=strongpassword
```

```
spring.jpa.hibernate.ddl-auto=update
```

```
'''
```

### 3. Розгортання бекенду

#### 1. Підготовка серверного середовища:

- Встановіть Java (якщо не встановлено):

```
'''bash
```

```
sudo apt update
```

```
sudo apt install openjdk-17-jdk
```

```
'''
```

- Перевірте версію:

```
'''bash
```

```
java -version
```

...

## 2. Збірка Spring Boot проекту:

- Виконайте збірку за допомогою Maven:

```
```bash  
  
mvn clean package  
```
```

- У результаті отримаєте файл `.jar` у папці `target`.

## 3. Запуск Spring Boot на сервері:

- Скопіюйте `.jar` файл на сервер:

```
```bash  
  
scp target/your-application.jar user@your-server-ip:/path/to/deploy  
```
```

- Запустіть програму:

```
```bash  
  
java -jar your-application.jar  
```
```

## 4. Автоматизація запуску:

- Додайте Spring Boot до служби `systemd` для автоматичного запуску:

```
```bash  
  
sudo nano /etc/systemd/system/sensor-system.service  
```
```

Вміст файлу:

...

[Unit]

Description=Sensor System Backend

After=network.target

[Service]

User=your-user

ExecStart=/usr/bin/java -jar /path/to/deploy/your-application.jar

Restart=always

[Install]

WantedBy=multi-user.target

...

- Активуйте службу:

```
```bash
```

```
sudo systemctl daemon-reload
```

```
sudo systemctl enable sensor-system
```

```
sudo systemctl start sensor-system
```

...

## 4. Взаємодія з Arduino

### 1. Підключення Arduino до мережі:

- Arduino (з Ethernet Shield або модулем ESP8266/ESP32) повинен мати доступ до серверного API через публічну IP-адресу або доменне ім'я.

### 2. Доступ до API:

- У Arduino-кодi вказується URL сервера, наприклад:

```
```cpp
```

```
const char* serverUrl = "http://your-server-ip/api/sensors/{sensorId}/data";
```

...

### 3. Захист API:

- Додайте авторизацію для запитів Arduino, наприклад, через токен:

```
```java
```

```
@PostMapping("/{sensorId}/data")
```

```

public ResponseEntity<SensorData> addSensorData(
    @PathVariable Long sensorId,
    @RequestBody SensorDataDTO sensorDataDTO,
    @RequestHeader("Authorization") String token) {
    if (!authService.isValidToken(token)) {
        return ResponseEntity.status(HttpStatus.UNAUTHORIZED).build();
    }

    SensorData sensorData = sensorService.addSensorData(sensorId,
sensorDataDTO);

    return ResponseEntity.ok(sensorData);
}
'''

```

#### 4. Відкриття доступу до API:

- Якщо сервер знаходиться за NAT, використовуйте портфорвардинг для відкриття публічного доступу до API.

### 5. Налаштування зворотного проксі (NGINX)

Для покращення безпеки та оптимізації роботи можна використовувати NGINX як зворотний проксі.

#### 1. Встановлення NGINX:

```

'''bash

sudo apt update

sudo apt install nginx

'''

```

#### 2. Налаштування проксі:

- Редагуйте конфігурацію:

```

'''bash

sudo nano /etc/nginx/sites-available/sensor-system

```

...

Вміст файлу:

...

```
server {  
    listen 80;  
    server_name your-server-domain;  
    location / {  
        proxy_pass http://127.0.0.1:8080;  
        proxy_set_header Host $host;  
        proxy_set_header X-Real-IP $remote_addr;  
    }  
}
```

- Активуйте конфігурацію:

```
```bash  
sudo ln -s /etc/nginx/sites-available/sensor-system /etc/nginx/sites-enabled/  
sudo nginx -t  
sudo systemctl restart nginx  
```
```

## **6. Розгортання фронтенду (якщо є)**

Якщо є фронтенд (наприклад, сторінки на Thymeleaf):

- Файли фронтенду вже інтегровані в Spring Boot і доступні через те ж API.
- Фронтенд доступний за адресою сервера.

## **7. Тестування**

1. Перевірка бази даних:

- Виконайте запити до MySQL, щоб переконатися, що база працює.

## 2. Перевірка API:

- Використовуйте Postman для тестування основних ендпоїнтів.

## 3. Перевірка Arduino:

- Запустіть пристрій, переконайтеся, що дані надсилаються до сервера, і перевірте, чи зберігаються вони в базі.

# 8. Автоматизація та підтримка

## 1. Моніторинг:

- Встановіть інструменти моніторингу, такі як Prometheus чи Grafana, для збору метрик системи.

## 2. Резервне копіювання:

- Налаштуйте регулярне резервне копіювання бази даних.

## 3. Оновлення системи:

- Використовуйте CI/CD для автоматизації процесів деплою (наприклад, GitHub Actions або Jenkins).

Ці кроки забезпечать надійне та безпечне розгортання системи на сервері з інтеграцією Arduino для збору реальних даних.

Посилання на відео демонстрації роботи:

<https://www.youtube.com/watch?v=yn-cDbsklyE>