

Харківський національний університет імені Н. В. Каразіна
Факультет комп'ютерних наук

ЗВІТ
З ЛАБОРАТОРНОЇ РОБОТИ №3
«Spring boot та робота з базою даних»
Дисципліна: «ПРОЕКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ»

Виконав: студент групи КС31
Єрмоменко Володимир
Володимирович

Перевірив: ст. викл. О.Ю.
Товстокоренко

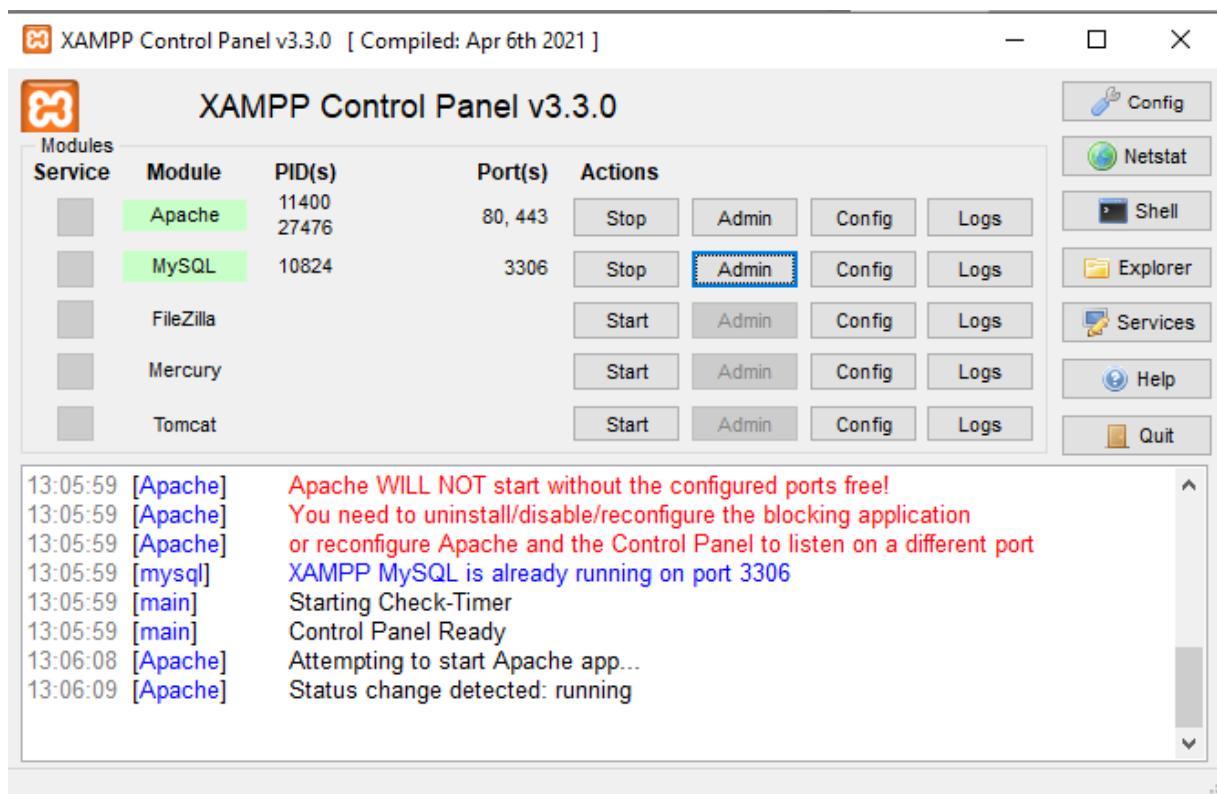
Харків
2022

Завдання:

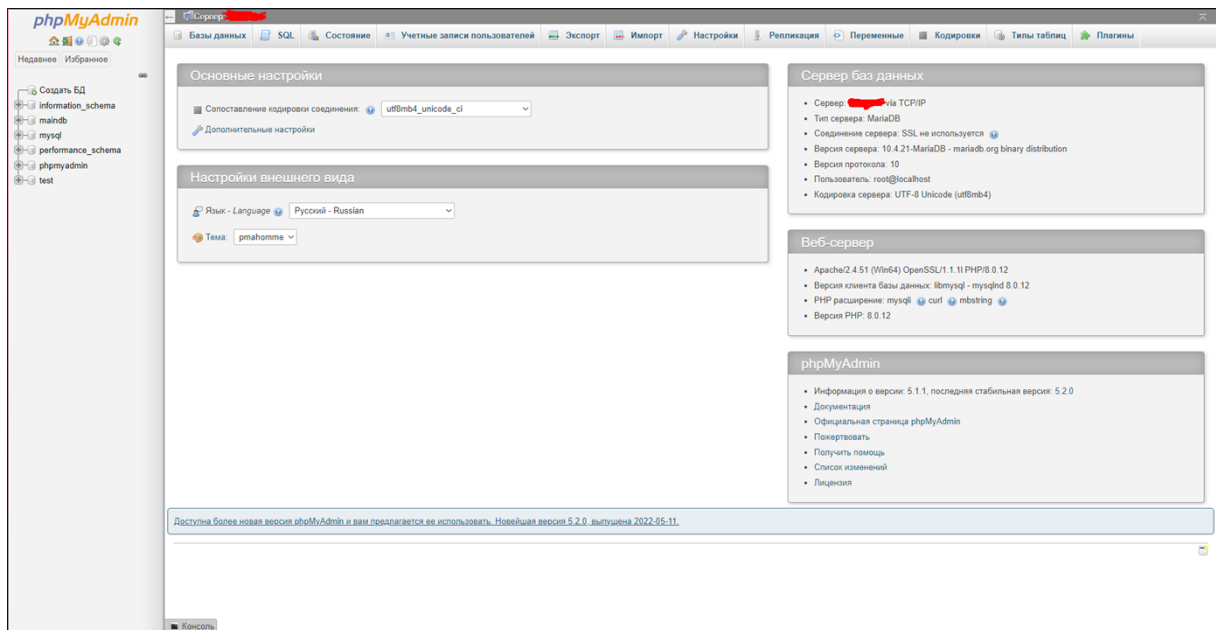
Створити свою базу даних MySQL обраної тематики через Java Spring, додати туди не менше 4 таблиць, та зв'язок один до одного та зв'язок багато до одного. Після запуску програми, перевірити чи все працює шляхом механічного внесення інформації до БД через phpMyAdmin.

Результати виконання лабораторної роботи №2 представлені:

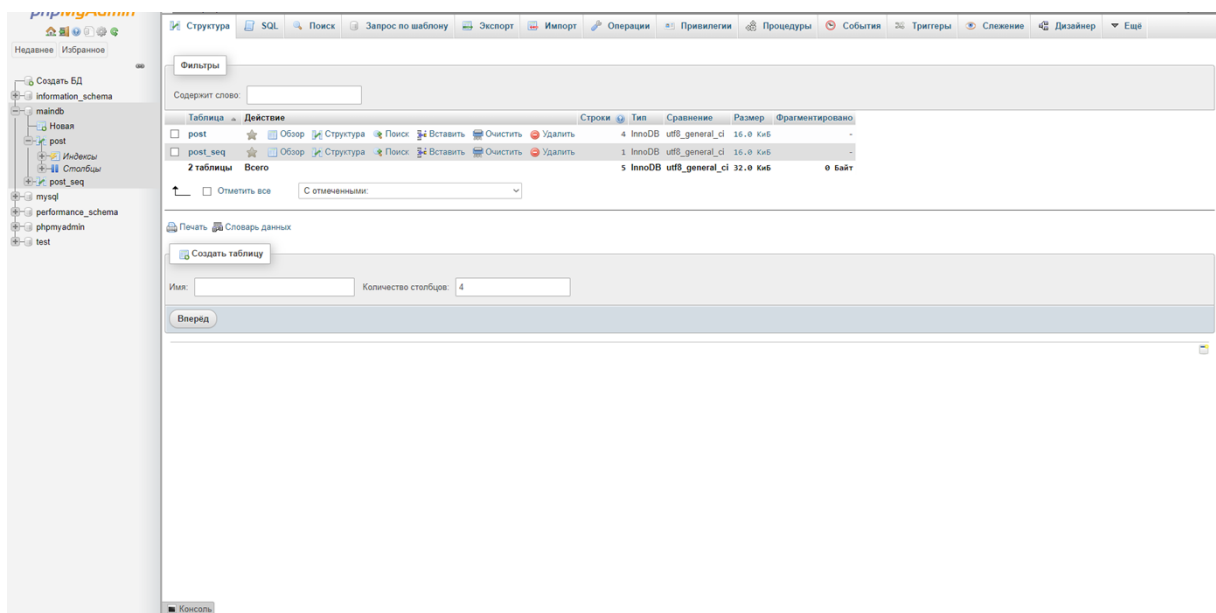
1. На скріншотах 1-5 – результати виконання роботи.
2. В додатку 1 – Всі класи програми



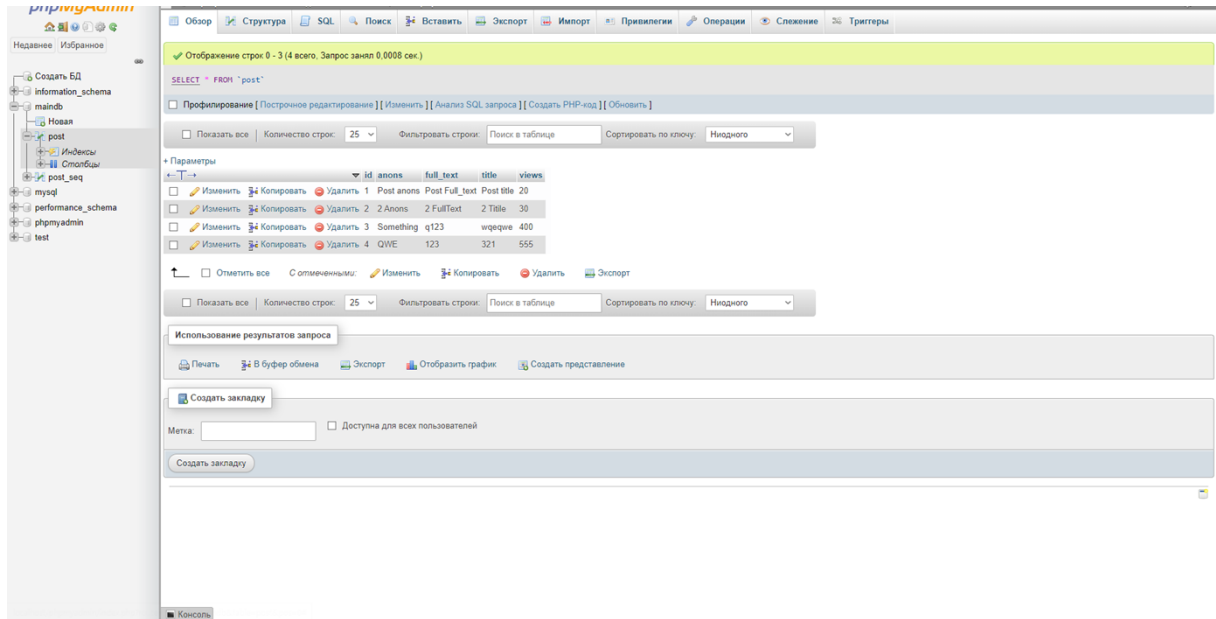
Скріншот 1 – Для віртуального серверу був використан додаток XAMPP



Скриншот 2 – Встановлений та запусканий PhpMyAdmin



Скриншот 3 – База данных maindb, котра була створена виконанням коду.



Скриншот 4 – Доданні до бази даних данні



Скриншот 5 – Вивод даних з бази даних за допомогою html сторінки

ДОДАТОК 1

Класс Post

```
package com.laboratiry3.lab.models;

import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;

@Entity
public class Post
{
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;
    private String title, anons, full_text;
    private int views;

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public String getAnons() {
        return anons;
    }

    public void setAnons(String anons) {
```

```

        this.anons = anons;
    }

    public String getFull_text() {
        return full_text;
    }

    public void setFull_text(String full_text) {
        this.full_text = full_text;
    }

    public int getViews() {
        return views;
    }

    public void setViews(int views) {
        this.views = views;
    }
}

```

Интерфейс PostRepository

```

package com.laboratiry3.lab.repo;

import com.laboratiry3.lab.models.Post;
import org.springframework.data.repository.CrudRepository;

public interface PostRepository extends CrudRepository<Post, Long>
{
}

```

Класс BlogController

```

package com.laboratiry3.lab.controllers;

import com.laboratiry3.lab.models.Post;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import com.laboratiry3.lab.repo.PostRepository;

@Controller
public class BlogController
{
    @Autowired
    private PostRepository postRepository;

    @GetMapping("/blog")
    public String blogmain(Model model){
        Iterable<Post> posts = postRepository.findAll();
        model.addAttribute("posts", posts);
        return "blog-main";
    }
}

```

```
}
}
```

HTML сторінка для відображення інформації

```
<!DOCTYPE HTML>
<html xmlns:th="http://www.thymeleaf.org">
<head>
  <title>Web Blog</title>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>
<body>

<div>
<h1>Web Blog</h1>
  <div th:each="el : ${posts}">
    <p th:text="ID: ' + ${el.id}" />
    <p th:text="Anons: ' + ${el.anons}" />
    <p th:text="Title: ' + ${el.title}" />
    <p th:text="FullText: ' + ${el.full_text}" />
    <a href="/blog/ + ${el.id}">Detail</a>
  </div>
</div>
</body>
</html>
```

Клас LabApplication

```
package com.laboratiry3.lab;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class LabApplication {

    public static void main(String[] args) {
        SpringApplication.run(LabApplication.class, args);
    }

}
```

Файл application.properties

```
server.port= 8081

spring.jpa.hibernate.ddl-auto=update
spring.datasource.url=jdbc:mysql://localhost:3306/maindb?enabledTLSProtocols=TLSv1.2
spring.datasource.username=root
spring.datasource.password=
```