Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки інформації та управління

**Звіт до комп’ютерного практикуму №1**

**З дисципліни «Основи Back-end технологій»**

Виконав:

Студент 3 курсу, гр. ІС-12

Кардаш Єгор

**2024 р.**

**Варіант 14.**

Спроектувати базу даних оголошень про квартири: вид оголошення

(здам / продам / зніму / куплю), адреса, кількість кімнат, дата, ціна

**Завдання.**

1. Виконати установку набору дистритутиву XAMPP(Denver або ін.).

2. Засобами php створити нову базу даних MySQL, відповідно до свого

варіанту.

3. Написати php-скрипт, який додає нового користувача в БД.

4. За допомогою php створити нову таблицю БД.

5. За допомогою утиліти phpMyAdmin створити другу таблицю БД.

6. За допомогою утиліти phpMyAdmin встановити зв’язок між створеними

таблицями.

7. Засобами PHP і SQL виконати виведення даних з таблиці бази даних на

WEB-сторінку.

8. Створити ще одну сторінку сайту і здійснити обмін даними між сторінками,

використовуючи гіперпосилання.

1.

За допомогою docker-compose запускаємо mysql та phpmyadmin. Також ця конфігурація створює користувача mysql

version: '3.8'

services:

mysql:

image: mysql:latest

restart: always

environment:

MYSQL\_USER: user

MYSQL\_PASSWORD: password

MYSQL\_ROOT\_PASSWORD: root\_password

MYSQL\_DATABASE: mydatabase

ports:

- "3306:3306"

volumes:

- mysql\_data:/var/lib/mysql

phpmyadmin:

image: phpmyadmin/phpmyadmin:latest

restart: always

environment:

PMA\_HOST: mysql

PMA\_PORT: 3306

MYSQL\_ROOT\_PASSWORD: root\_password

ports:

- "8080:80"

volumes:

mysql\_data:

2. За допомогою бібліотеки JS Prisma створюємо схему бази даних

generator client {

provider = "prisma-client-js"

}

datasource db {

provider = "mysql"

url = env("DATABASE\_URL")

}

model Flat {

id Int @id @default(autoincrement())

area String

floor Int

rooms Int

price Decimal

owner Owner @relation(fields: [ownerId], references: [id])

ownerId Int

}

model Owner {

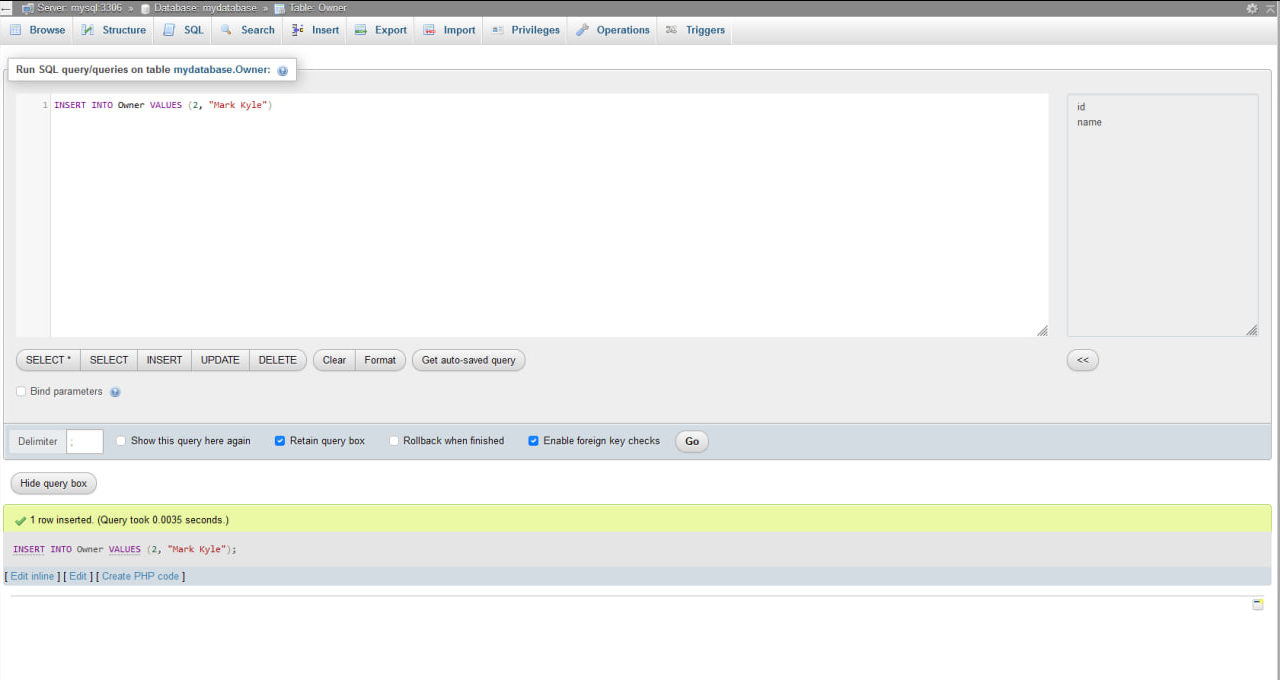
id Int @id @default(autoincrement())

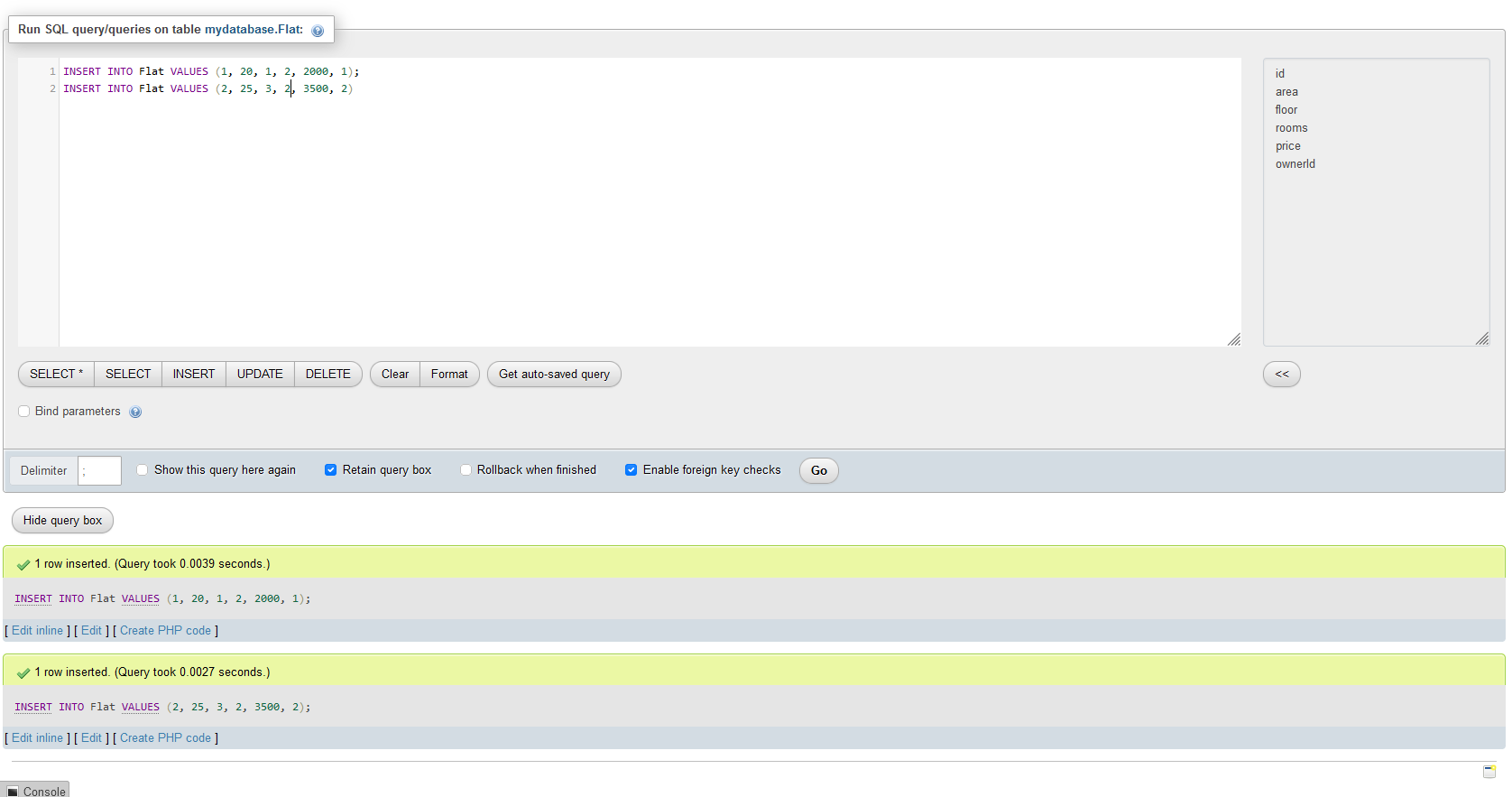
name String

flats Flat[]

}

3. Засобами PhpMyAdmin створюємо сутності





4. Запускаємо сервер, який повертає html сторінку з даними

import { PrismaClient } from '@prisma/client';

import express from 'express';

import path from 'path';

const prisma = new PrismaClient();

const app = express();

const port = 4000;

// Set Pug as the view engine

app.set('view engine', 'pug');

app.set('views', path.join(\_\_dirname, '..', 'public'));

app.get('/', async (req, res) => {

return 'Hello world';

});

app.get('/flats', async (req, res) => {

const flats = await prisma.flat.findMany({

include: { owner: true }

});

res.render('flat', { flats });

});

app.get('/owner/:id', async (req, res) => {

const { id } = req.params;

const owner = await prisma.owner.findFirst({

where: { id: +id }

});

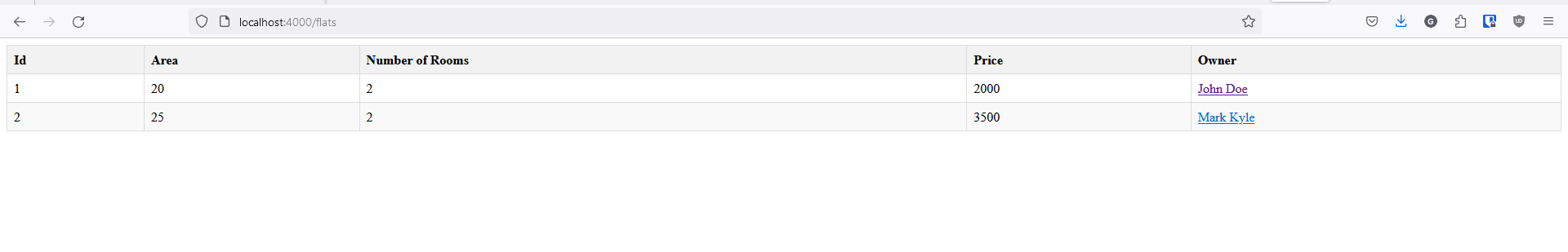
res.render('owner', owner);

});

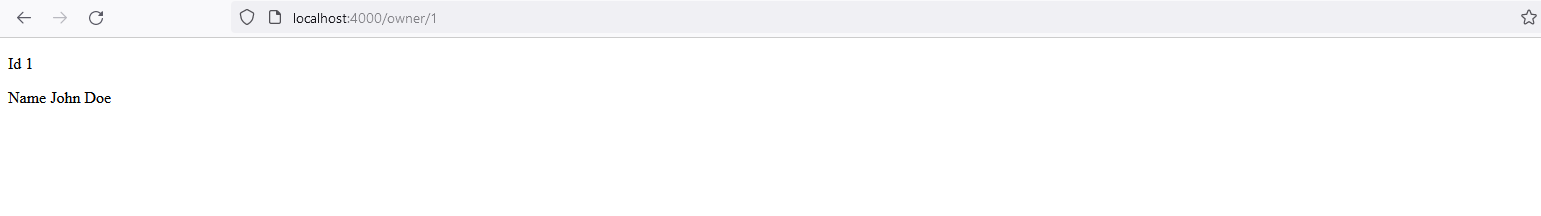
app.listen(port, () => {

console.log(`Server is listening on port ${port}`);

});



При переході за посиланням бачимо іншу сторінку



**Висновки**

Створив базу даних, таблиці, заповнив їх даними, вивів дані на WEB-сторінку, здійснити обмін даними між сторінками, використовуючи гіперпосилання.