Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Сибирский государственный автомобильно-дорожный университет (СибАДИ)»

Факультет Информационные системы в управлении

Направление 10.03.01Информационная безопасность

Кафедра Информационная безопасность

**ОТЧЕТ**

**по лабораторной работе№ 1**

**Тема: Реализация веб приложения**

по дисциплине «Защита Web-приложенений»

Выполнила студент группы БИб-21Э1 Мамагиев М.И

Преподаватель

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Брикман Я.К.

«\_3\_» \_сентября\_\_2024г

Омск 2024

**Лабораторная работа №1**

Реализовать веб-сервис, который:

- принимает запросы на создание объектов (с несколькими атрибутами);

- принимает запросы на получение объекта по идентификатору;

- отображает детальную информацию по объекту с его атрибутами;

Язык реализации любой, но без использования готовых больших фреймворков, закрывающих взаимодействие с БД / большим количеством middleware (т.к. мы хотим работать с запросами на более низком уровне - например с заголовками и писать свои обработчики).

Дальнейшие лабораторные будем реализовывать, основываясь на этом проекте.

Возможные задачи внутри команды:

- проектирование и реализация решения;

- анализ предметной области, формирование образа целевого результата;

постановка задач разработчикам;

- \*опционально\* - дизайн и реализация интерфейса;

- \*бонусно\* - упаковка решения в контейнер;

- презентация / демо;

**Реализация веб сервиса – Лотерейные билеты**

Для реализации веб приложения был использован Flask, SQLite для базы данных и добавлены стили для html.

Запустите приложение, выполнив команду:

python app.py

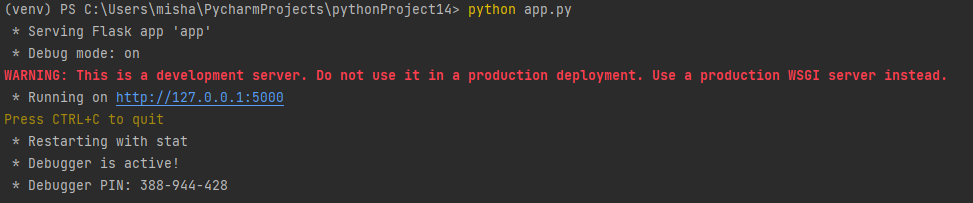


Рисунок 1. Запуск приложения

Переход по ссылке <http://127.0.0.1:5000/>

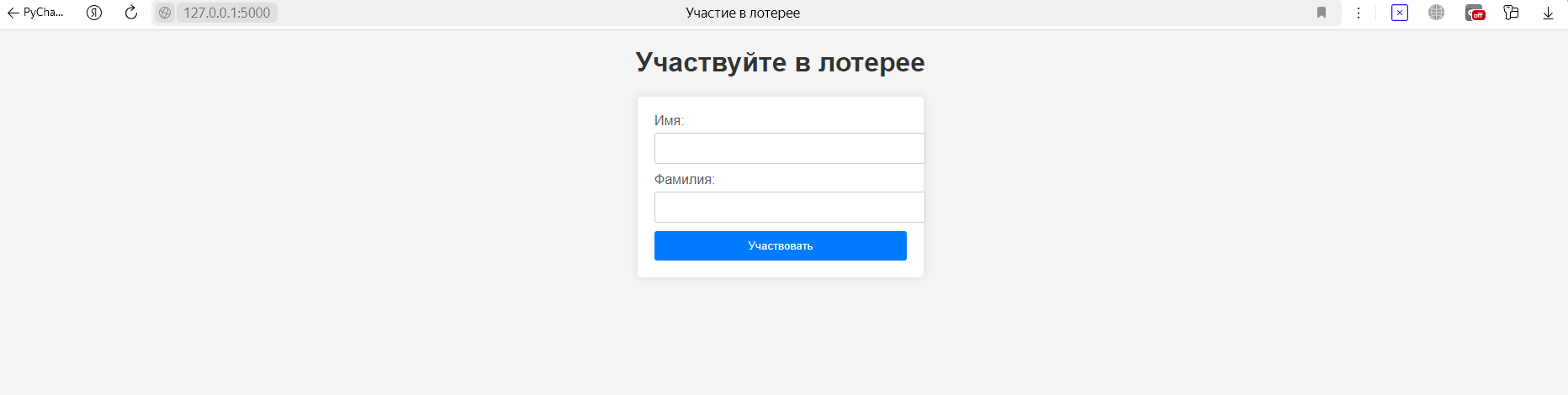


Рисунок 2. Работа приложения, участие в лотерее

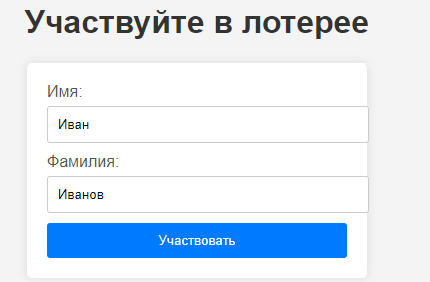


Рисунок 3. Заполнение данными

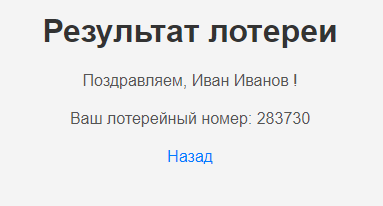


Рисунок 4. Результат лотереи

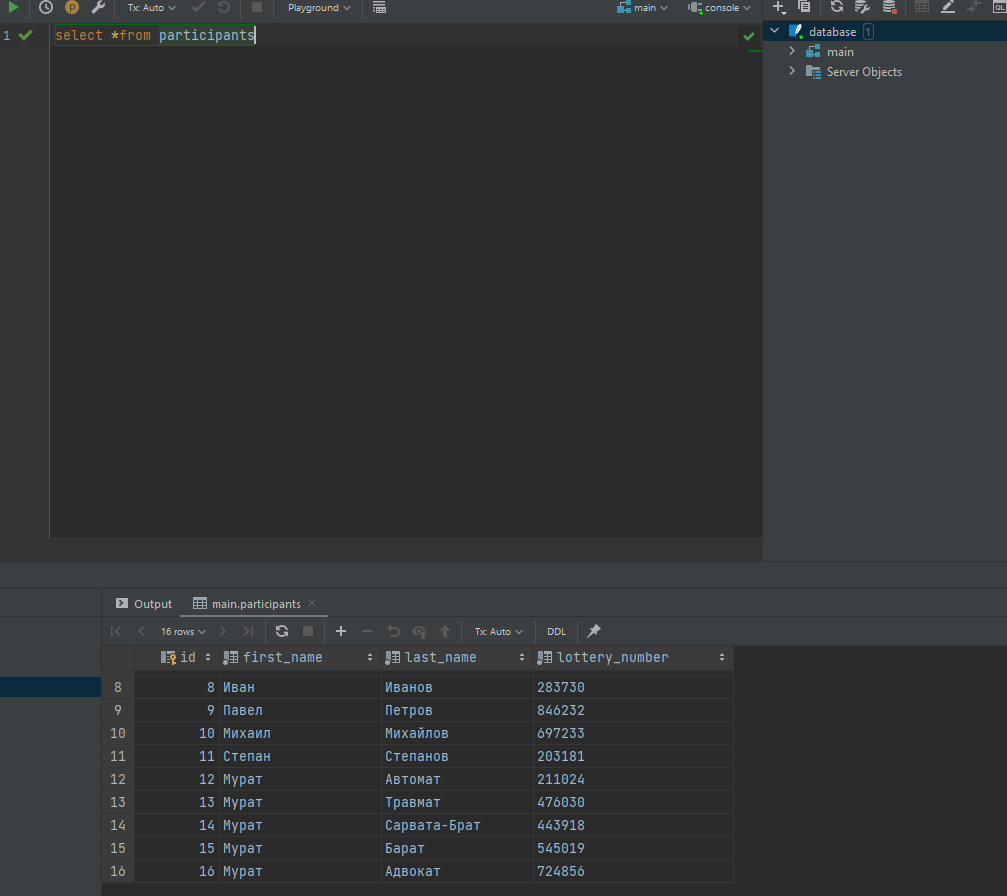


Рисунок 5. Сохранение данных в бд

**Объяснение кода:**

**Первым делом устанавливаем необходимые инструменты:**

- Установить питон с [официального сайта](https://www.python.org/downloads/)

- Установите Flask: Flask — это микрофреймворк для создания веб-приложений на Python.

pip install Flask

- Установите SQLite: SQLite обычно уже установлен с Python, но если нет, вы можете установить его отдельно.

**Создание структуры проекта:**

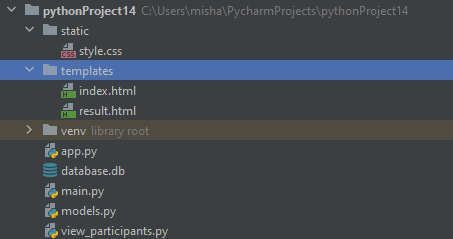


Рисунок 6. Структура проекта

**Настройка базы данных models.py:**

import sqlite3  
import random  
  
def init\_db():  
 conn = sqlite3.connect('database.db')  
 cursor = conn.cursor()  
  
 cursor.execute('''  
 CREATE TABLE IF NOT EXISTS participants (  
 id INTEGER PRIMARY KEY AUTOINCREMENT,  
 first\_name TEXT NOT NULL,  
 last\_name TEXT NOT NULL,  
 lottery\_number TEXT NOT NULL  
 )  
 ''')  
  
 conn.commit()  
 conn.close()  
  
def add\_participant(first\_name, last\_name):  
 lottery\_number = generate\_lottery\_number()  
 conn = sqlite3.connect('database.db')  
 cursor = conn.cursor()  
 cursor.execute('INSERT INTO participants (first\_name, last\_name, lottery\_number) VALUES (?, ?, ?)', (first\_name, last\_name, lottery\_number))  
 conn.commit()  
 conn.close()  
 return lottery\_number  
  
def generate\_lottery\_number():  
 return ''.join(random.choices('0123456789', k=6))  
  
def get\_participant\_by\_id(participant\_id):  
 conn = sqlite3.connect('database.db')  
 cursor = conn.cursor()  
 cursor.execute('SELECT \* FROM participants WHERE id = ?', (participant\_id,))  
 participant = cursor.fetchone()  
 conn.close()  
 return participant

**Создание веб-приложения app.py:**

from flask import Flask, request, render\_template, redirect, url\_for  
import models  
  
app = Flask(\_\_name\_\_)  
  
@app.route('/')  
def index():  
 return render\_template('index.html')  
  
@app.route('/participate', methods=['POST'])  
def participate():  
 first\_name = request.form['first\_name']  
 last\_name = request.form['last\_name']  
 lottery\_number = models.add\_participant(first\_name, last\_name)  
 participant\_id = models.get\_participant\_by\_id(lottery\_number)  
 return render\_template('result.html', first\_name=first\_name, last\_name=last\_name, lottery\_number=lottery\_number)  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 models.init\_db()  
 app.run(debug=True)

**Создание шаблонов templates/index.html:**

<!DOCTYPE html>  
<html lang="ru">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>Участие в лотерее</title>  
 <link rel="stylesheet" href="{{ url\_for('static', filename='style.css') }}">  
</head>  
<body>  
 <h1>Участвуйте в лотерее</h1>  
 <form action="/participate" method="post">  
 <label for="first\_name">Имя:</label>  
 <input type="text" id="first\_name" name="first\_name" required>  
 <label for="last\_name">Фамилия:</label>  
 <input type="text" id="last\_name" name="last\_name" required>  
 <button type="submit">Участвовать</button>  
 </form>  
</body>  
</html>

**templates/result.html**

<!DOCTYPE html>  
<html lang="ru">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>Результат лотереи</title>  
 <link rel="stylesheet" href="{{ url\_for('static', filename='style.css') }}">  
</head>  
<body>  
 <h1>Результат лотереи</h1>  
 <p>Поздравляем, {{ first\_name }} {{ last\_name }}!</p>  
 <p>Ваш лотерейный номер: {{ lottery\_number }}</p>  
 <a href="/">Назад</a>  
</body>  
</html>

**Создание CSS-файла static/style.css:**

body {  
 font-family: Arial, sans-serif;  
 margin: 0;  
 padding: 0;  
 background-color: #f4f4f4;  
}  
  
h1 {  
 text-align: center;  
 color: #333;  
}  
  
form {  
 width: 300px;  
 margin: 20px auto;  
 padding: 20px;  
 background-color: #fff;  
 border-radius: 5px;  
 box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
}  
  
label {  
 display: block;  
 margin-bottom: 5px;  
 color: #555;  
}  
  
input[type="text"] {  
 width: 100%;  
 padding: 10px;  
 margin-bottom: 10px;  
 border: 1px solid #ccc;  
 border-radius: 3px;  
}  
  
button {  
 width: 100%;  
 padding: 10px;  
 background-color: #007BFF;  
 color: #fff;  
 border: none;  
 border-radius: 3px;  
 cursor: pointer;  
}  
  
button:hover {  
 background-color: #0056b3;  
}  
  
p {  
 text-align: center;  
 margin: 20px 0;  
 color: #555;  
}  
  
a {  
 display: block;  
 text-align: center;  
 margin-top: 20px;  
 color: #007BFF;  
 text-decoration: none;  
}  
  
a:hover {  
 text-decoration: underline;  
}

**Запуск приложения:**

python app.py