**Міністерство освіти і науки України**

**Національний університет «Запорізька політехніка»**

Кафедра програмних засобів

**ЗВІТ**

Дисципліна «Розробка прикладних програм»

Робота №3

Тема «Розроблення вебдодатків та реалізація доступу до систем керування базами даних через програмні інтерфейси»

**Виконав варіант 19**

Студент КНТ-122 Онищенко О. А.

**Прийняли**

Викладач Дейнега Л. Ю.

2024

Мета роботи

Навчитися використовувати програмні інтерфейси для доступу до баз даних. Навчитися розробляти вебдодатки за допомогою фреймворку Django.

Індивідуальне завдання

Вебдодаток керування фінансами, який забезпечує виконання одноразових та періодичних платежів клієнтами. Початкова кількість коштів на рахунку кожного клієнта та його кредитні ліміти визначаються менеджером, якому доступна вся інформаія про клієнтів. Клієнт може переглядати всю інформацію про власний рахунок (разом з історією транзакцій), вносити кошти, сплачувати за послуги, призначати періодичні платежі.

Тексти файлів

Результати виконання

Контрольні питання

Які існують способи збереження даних в програмах написаних мовою Python?

Кілька способів:

* Як локальні змінні: тоді воні зберігатимуться лише на час роботи програми;
* Як файли: тоді їх потрібно вручну читати та записувати при змінах. Поширені формати: JSON, CSV, TXT;
* У базі даних: для цього зазвичай використовують Систему керування базою даних. Поширені СКБД: MySQL, PostgreSQL, MongoDB.

Для чого призначена бібліотека MySQLdb?

Для підключення та взаємодії з СКБД MySQL.

Які існують високорівневі функції бібліотеки MySQLdb?

Найпоширеніші:

* connect: створює підключення з базою даних. Приймає як аргументи назву хоста, ім’я користувача, пароль, опціонально базу даних з якою працювати;
* query: формує та виконує запит до бази даних. Текст запиту пишеться мовою SQL для MySQL;
* store\_result: завантажує результати запиту та зберігає їх локально в повному обсязі;
* use\_result: завантажує результати запиту та зберігає їх на сервері, подає рядок за рядком;
* fetch\_row: показує рядок результатів запиту.

[Джерело](https://mysqlclient.readthedocs.io/user_guide.html#some-mysql-examples)

Яким чином виконати запит до бази даних та яким чином переглянути результати?

Через підключення до бази даних викликом методу connect, формування запиту викликом методу query з текстом запиту мовою SQL, збереження результатів віддалено викликом методу use\_result, та виведення результатів рядок за рядком викликом метду fetch\_row. АЛИЛУЯ

Що таке шоблон проєктування MVC?

Поділяє архітектуру програми на три рівні:

1. *model* (модель): містить всі інформаційні частини додатку;
2. *view* (представлення): описує логіку інтерфейсу та показує дані користувачеві;
3. *controller* (контроллер): інтерфейс між моделлю та представленням. Обробляє запити, модифікує дані, надсилає їх представленню.

[Джерело](https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm)

За допомогою яких команд виконується розроблення додатків у бібліотеці Django?

Важливі команди:

1. Додати теку де проєкт буде жити

mkdir server

1. Ініціалізувати сам проєкт

django-admin startproject runner server

1. Перейти у теку з проєктом

cd server

1. Зробити новий застосунок де будуть всі файли основні

py manage.py startapp app\_name

1. Коли зробили зміни до моделі реєструємо міграцію

py manage.py makemigrations app\_name

1. Тоді здійснюємо міграцію

py manage.py migrate

1. Коли треба до адмін панелі зайти, створити користувача

py manage.py createsuperuser

Вказати всі необхідні дані:

* імя користувача
* електронна пошта
* пароль

1. Запустити сервер аби побачити всі зміни на сайті

py manage.py runserver

З яких файлів складається проєкт Django?

При створенні містить таку структуру файлів:

project/

default\_app/

\_\_init\_\_.py

asgi.py

settings.py

urls.py

wsgi.py

new\_app/

migrations/

\_\_init\_\_.py

0001\_initial.py

\_\_init\_\_.py

admin.py

apps.py

models.py

tests.py

urls.py

views.py

manage.py

Яку структуру мають додатки Django?

Типова структура:

назва\_проєкту/

назва\_додатку/

назва\_додатку/

manage.py

Для чого необіхдні та яким чином реалізуються моделі Django?

Модель це таблиця даних для бази даних. Реалізується створенням класів моделей у файлі models.py потрібного додатку:

from django.db import models

class BibleBook(models.Model):

canonical\_order\_number=models.PositiveIntegerField()

name=models.CharField(max\_length=77)

number\_of\_chapters=models.PositiveIntegerField()

Яким чином визначаються представлення Django?

У теці додатку має міститися файл *views.py*. До нього можна додати функцію, яка прийматиме аргумент *request* і повертатиме відповідь типу *HttpResponse*. Приклад такої функції нижче:

from django.shortcuts import render

form django.http import HttpResponse

def get\_response(request):

return HttpResponse("**ІСУС ХРИСТОС – ГОСПОДЬ**")

[Джерело](https://www.w3schools.com/django/django_views.php)

Для чого необіхдні та яким чином визначаються і підключаються шаблони?

Для створення шаблону потрібно створити нову теку у теці додатку під назвою *templates*. До неї можна вносити файли шаблонів типу HTML. Сам файл шаблону може містити HTML код, а також спеціальні вирази для відображення динамічного вмісту.

Для підключення шаблону до представлення необхідно модифікувати функцію у файлі *views.py* таким чином:

from django.shortcuts import render

form django.template import loader

def get\_response(request):

template=loader.get\_template("template\_name.html")

return HttpResponse(template.render())

[Джерело](https://www.w3schools.com/django/django_templates.php)

Яким чином можна використати статичні ресури в Django?

Для додавання статичних файлів до проєкту Django потрібно створити теку static у теці додатку. До теки static можна додавати статичні файли, як от зображення чи таблиці стилів.

Після додавання необхідних статичних файлів для їх відображення у шаблоні можна модифікувати його так:

{% load static %}

<!DOCTYPE html>

<html>

<link rel="stylesheet" href="{% static 'file\_name.css' %}">

<body>

**ІСУС ХРИСТОС – ГОСПОДЬ**

</body>

</html>

[Джерело](https://www.w3schools.com/django/django_add_static_files.php)