

Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут імені Ігоря Сікорського” Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Лабораторна робота №4

**Сучасні технології серверного програмування з**

**використанням Python (Django)**

**«Використання пакета FLASK з мовою програмування**

**Python 3 для побудови веб-сайту візитки»**

|  |  |
| --- | --- |
| Виконали студенти групи ІТ-03: | Перевірив: |
| Федяй Б.В.  Кобилинський В. О.  Кашталян Є. С. | Резніков С.А. |

Київ 2023

**Варіант 16:**

Сайт для автоматизації прийому замовлень ресторану. Можливість виконання замовлення, наявність і кількість інгредієнтів, їх вартість.

**Завдання:**

Розробити статичний сайт серверної частини інформаційної системи з використанням мови Python 3 та пакета Flask згідно № варіанту та додаткових умов (без підключення СУБД та ORM).

Сайт обов’язково повинен включати наступні сторінки:

1. Головна сторінка
2. Товари та послуги
3. Прайс-лист (ціни на товари / послуги)
4. Контакти (зворотний зв’язок) (Поштова та фактична адреси, номери телефонів, e-mail, адреса сайту, посилання на сторінку Facebook та Telegram.) Також навести додаткову інформацію або умови для клієнтів.

Для оформлення сайту використати файли CSS та шаблони Jinja2. Кількість позицій товарів або послуг не більше 8.

Використати тільки державну мову.

[Репозиторій із кодом лабораторної роботи](https://github.com/Django-Teams/Lab4)

1. **Бізнес-логіка та дані.**

В цьому проекті, в силу невеликої кількості коду для опису бізнес-логіки, всі роутери знаходяться в одному файлі – app.py. Більшість із функцій там повертають HTML-сторінки (рисунок 1.1), у той час як деякі інші, виконують дії над імпровізованим сховищем даних та або повертають відповідь про успішне виконання роботи (рисунок 1.2), або перенаправляють користувача на іншу адресу (рисунок 1.3).

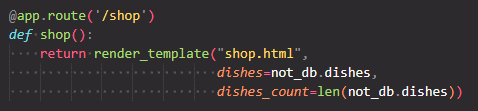


Рисунок 1.1 – функція shop()

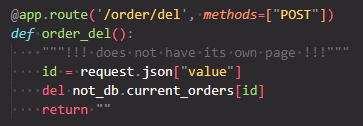


Рисунок 1.2 – функція order\_del()

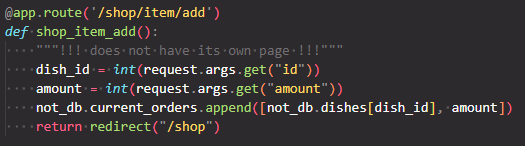


Рисунок 1.3 – функція shop\_item\_add()

Ситуація із даними в нас трохи цікавіша. Оскільки нам по завданню не можна було використовувати жодних СУБД, в тому числі й SQLite, що дозволяє зберігати дані в окремому файлі на диску, треба було щось зробити, аби можна було надавати динамічний функціонал веб-застосунку. Хоч в завданні і було сказано «розробити статичний сайт», без жодної можливості збереження даних реалізувати функціонал для «сайту для автоматизації прийому замовлень ресторану із можливістю виконання замовлення, наявності і кількості інгредієнтів, їх вартості» було б неможливо.

Було прийняте рішення зробити імпровізоване сховище даних у файлі clearly\_not\_a\_db.py (рисунок 1.4), котрий містить три масиви:

1. Кортеж dishes – зберігає дані про доступні у ресторані страви: назву, опис, зображення та потрібні для приготування інгрідієнти
2. Словник ingredients – зберігає дані про доступні у ресторані інгрідієнти: їх ціну та кількість на складі
3. Список current\_orders – зберігає дані про замовлені страви та їх кількість

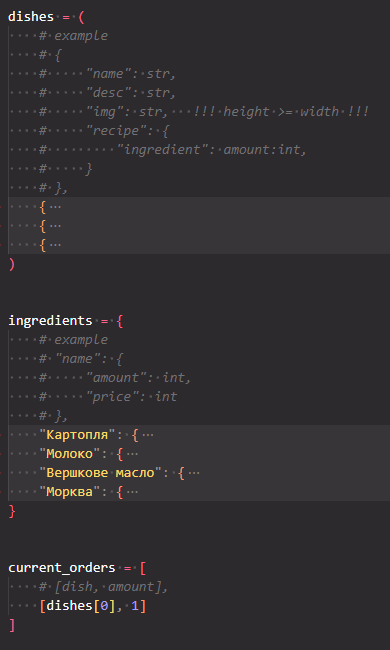


Рисунок 1.4 – вигляд файлу clearly\_not\_a\_db.py

1. **Шаблони та статика.**

У цьому проекті шаблони були розділені на дві групи: основні та допоміжні (рисунок 2.1). Завдяки тому, що модуль Flask підтримує використання шаблонів Jinja2, ми можемо виносити дублюючі частини html-розмітки в окремі файли та циклічно їх імпортувати на етапі подготовки відповіді користувачу.

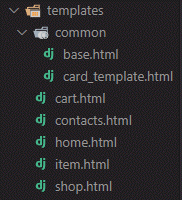


Рисунок 2.1 – шаблони створеного проекту

Легше буде почати опис із допоміжних шаблонів, оскільки усі основні базуються на одному із них – в файлі base.html описані хедер для навігації по сайту, уся метадата та деякі бібліотеки і скрипти, що усюди використовуються в проекті. Цей файл має у собі три блоки:

1. title – назва сторінки в браузері
2. content – контент сторінки
3. custom\_static – місце для додавання локальних статичних файлів

В противагу до base.html, котрий усі файли розширяють, також є і card\_template.html, котрий є розширенням функціоналу шаблонів. Він описує структуру «карток» страв, доступних у каталозі (рисунок 2.2).



Рисунок 2.2 – зовнішний вигляд сторінки «Каталог» із картками страв

До основних шаблонів відносяться усі ті сторінки, що є розширеннями до шаблону base.html. Приклад розмітки такої сторінки можна подивитися на рисунку 2.3. Це є:

1. cart – замовлення користувача
2. contacts – контактна інформація ресторану
3. home – головна сторінка сайту
4. item – сторінка будь-якої страви
5. shop – каталог страв



Рисунок 2.3 – текст шаблону shop.html

До статистичних файлів відносяться усі JS-скрипти та CSS-файли. Оскільки загалом у проекті стилів небагато, вони всі зберігаються в одному загальному файлі styles.css (рисунок 2.4). Щодо до скриптів, то вони усі прив’язані до якогось функціоналу, а не до графічних наворотів. Наприклад, скрипт item.js описує усю ту невелику клиєнтську логіку сторінки item.html і лише змінює лічильник загальної ціни на N замовлених страв (рисунок 2.5).

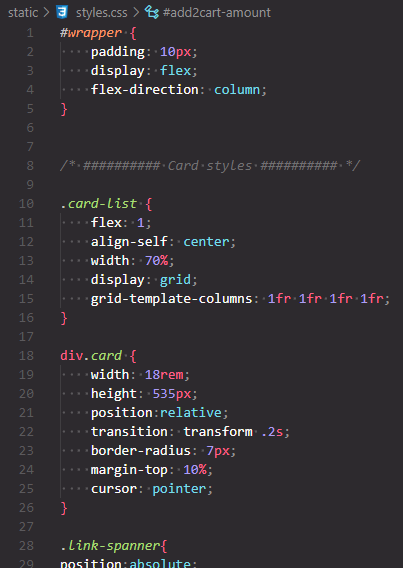


Рисунок 2.4 – фрагмент файлу styles.css

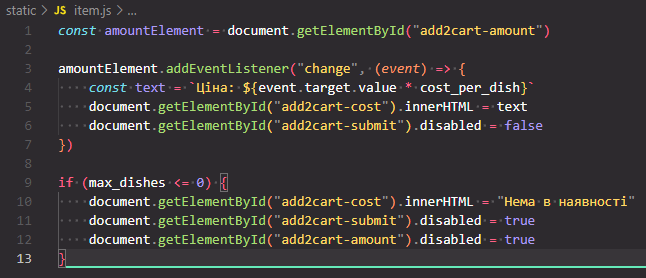


Рисунок 2.5 – код скрипту item.js

1. **Демонстрація проекту.**

Почнемо з головної сторінки (рисунок 3.1). До неї можна потрапити за адресою «/» або «/home» чи натиснувши на назву ресторану у лівому верхньому кутку на хедері. Ми не знали, що саме там можна розмістити, тому лишили її досить пустою, але із урахуванням того, що фронтенд-розробник міг туди додати що завгодно.

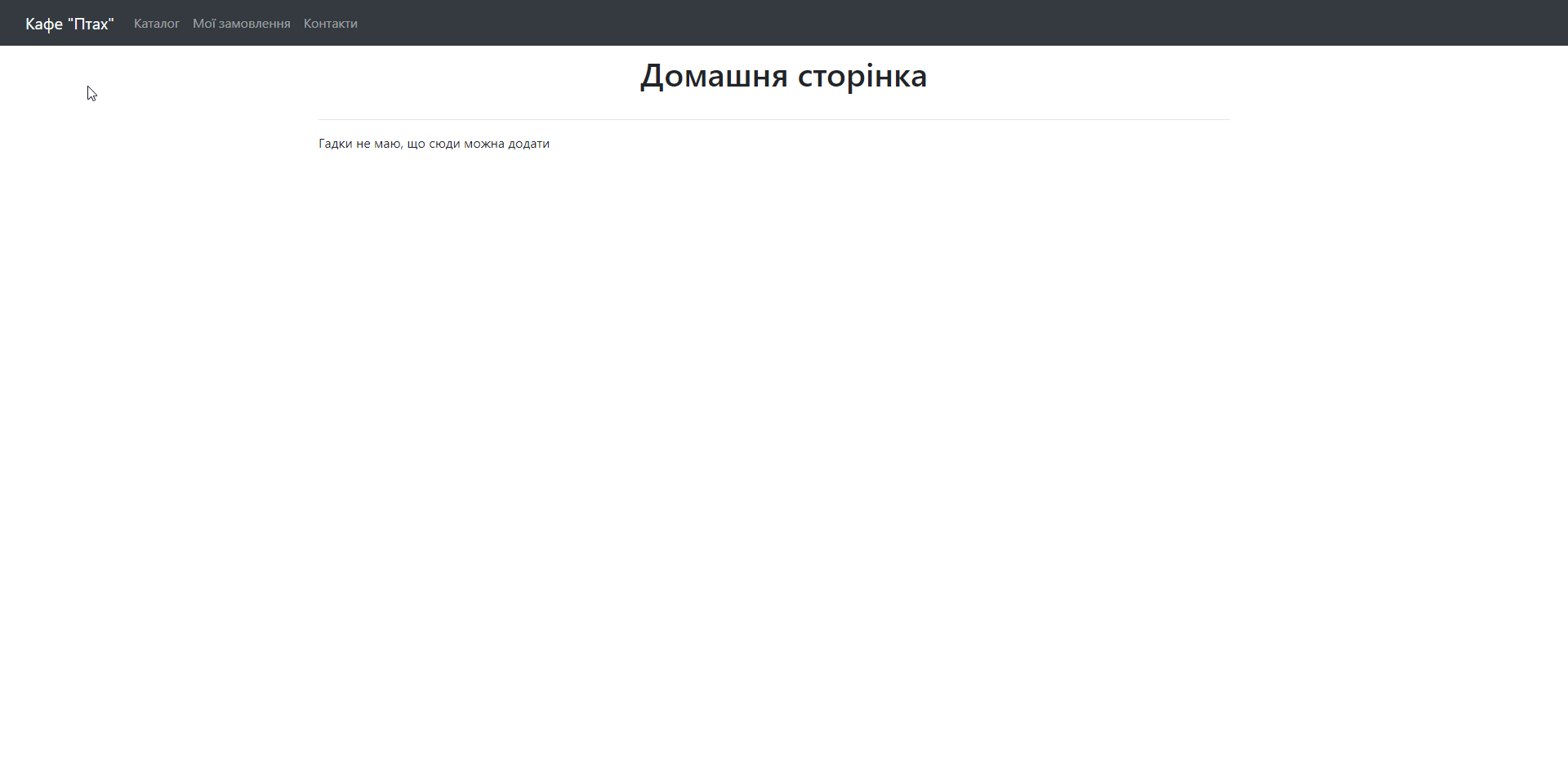


Рисунок 3.1 – головна сторінка

Сторінка контактних даних (рисунок 3.2). Заповнена імпровізованими даними та може бути розширеною, якщо треба.

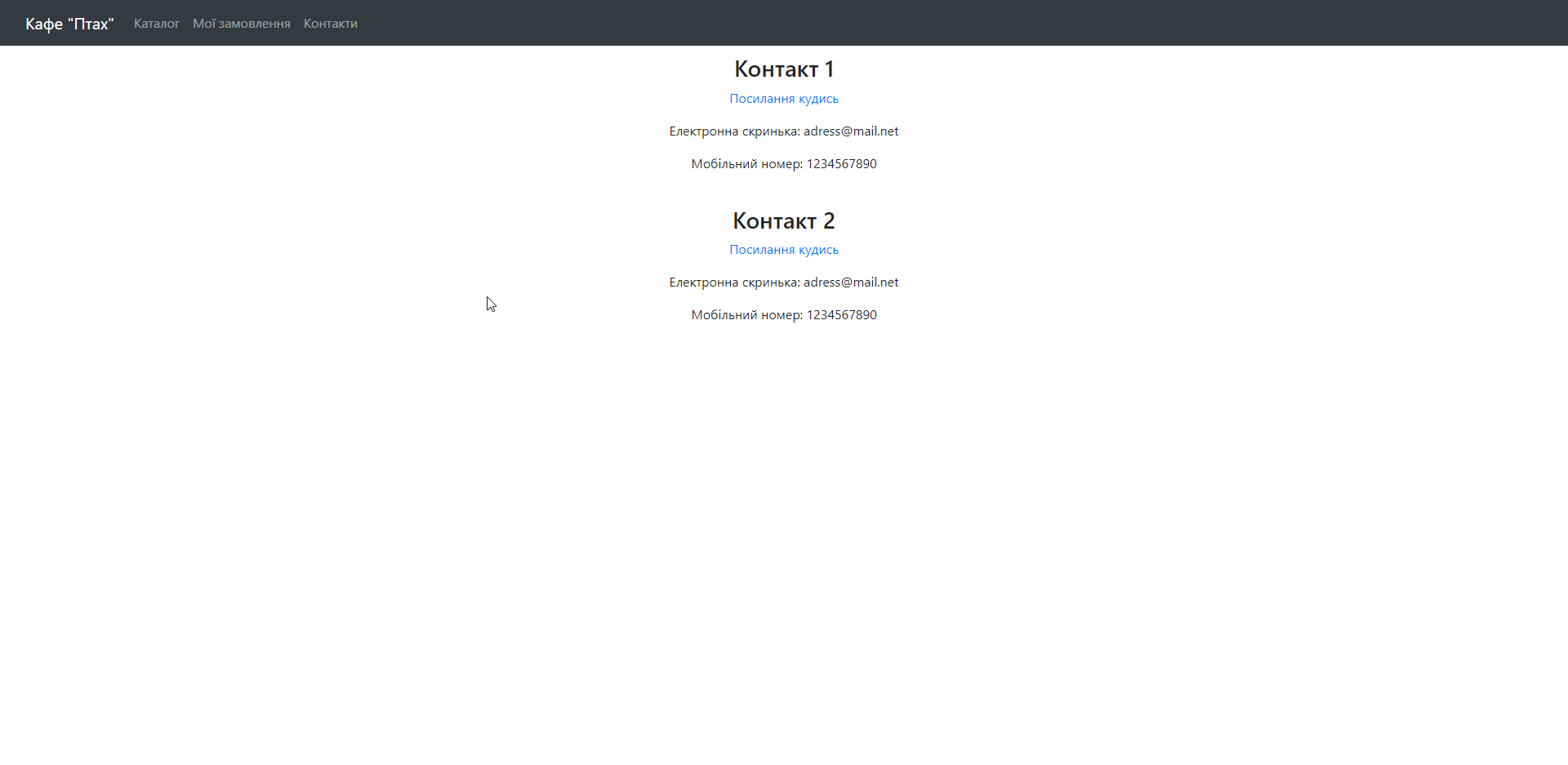


Рисунок 3.2 – сторінка контактних даних

Сторінку каталогу вже можна було побачити вище у звіті (рисунки 2.2, 3.3). На ній можна побачити картки зі стравами, на котрих можна побачити частину їх опису та зображення. Кожна із них кликабельна та веде до сторінки самої страви.



Рисунок 3.3 – сторінка каталогу

На сторінці страви (рисунок 3.4) можна переглянути повний опис страви та замовити деяку її кількість (скільки дозволяє склад інгрідієнтів). При замовленні страви, формується POST-запит та відправляється на адресу «/shop/item/add», звідки, в разі успіху, користувача перенаправляє на сторінку каталогу.

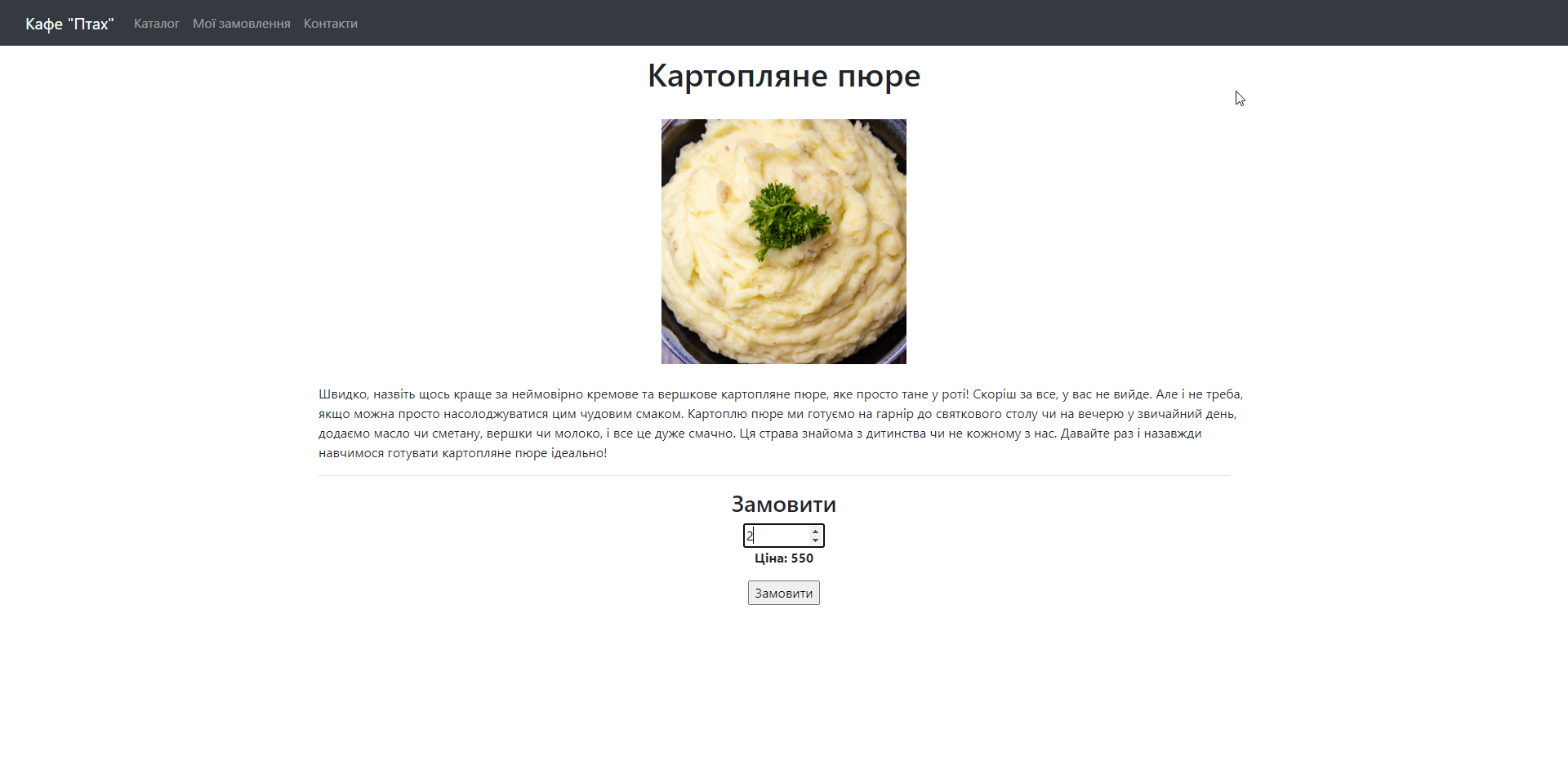


Рисунок 3.4 – сторінка страви

І в нас залишається лише сторінка із нашими замовленнями (рисунок 3.5). На ній відображається інформація про усі окремі замовлення страв та загальна ціна чеку. Тут також можна, натиснувши на хрестик, видалити страву з чеку.

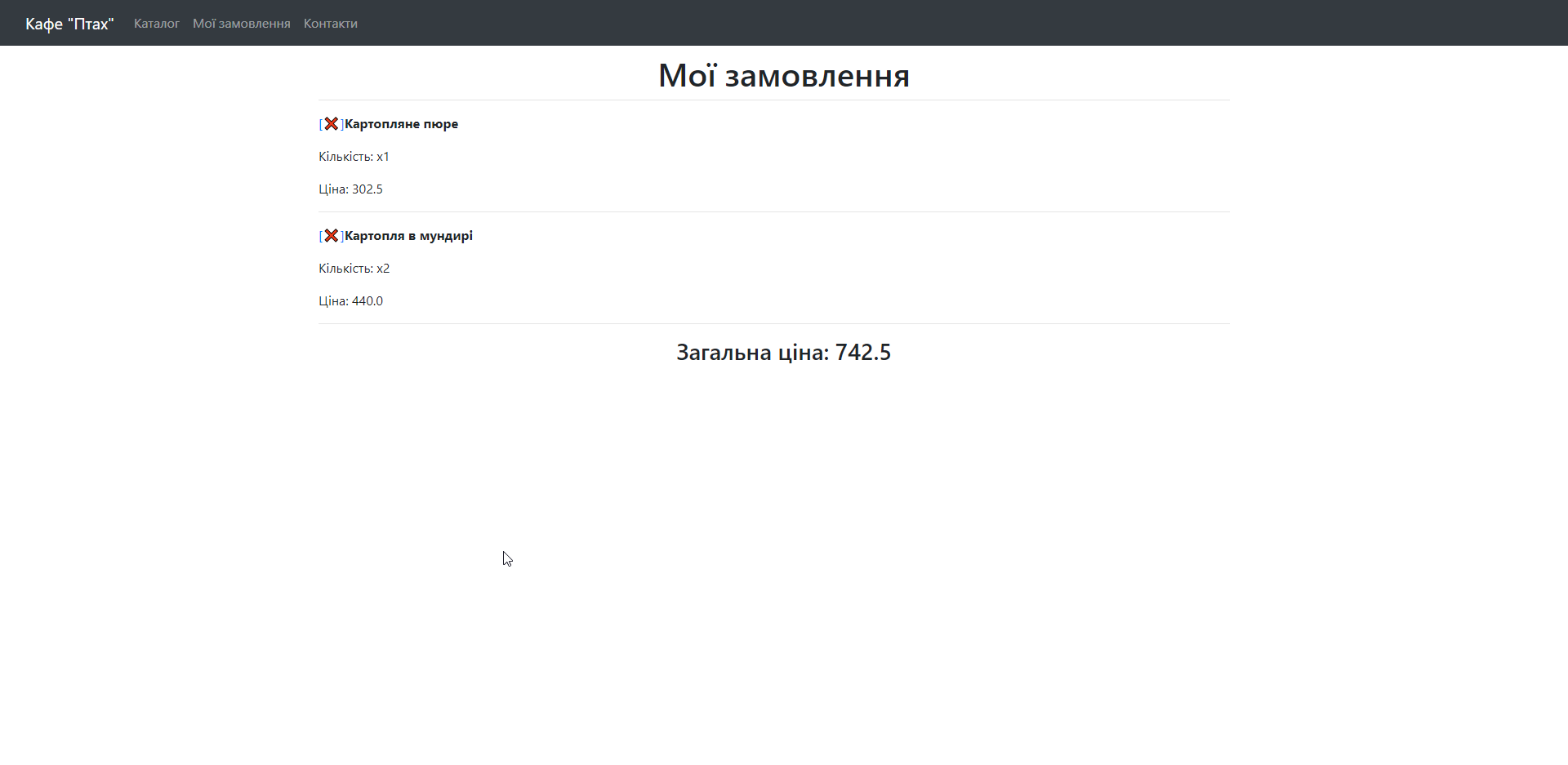


Рисунок 3.5 – сторінка замовлень

**Висновок:** під час виконання цієї лабораторної, було поновлено знання створення веб-застосунків на веб-фреймворку Flask з імпровізованим сховищем даних, не використовуючи жодних СУБД.