## Документация за E-Commerce Проекта

**1. Въведение**

Проектът представлява **уеб-базирана e-commerce платформа** с две роли: **собственик** (owner) и **клиент** (customer).

* **Собственикът** може да управлява продукти, да разглежда поръчки и да анализира статистики за продажби.
* **Клиентът** може да разглежда продукти, да добавя артикули в кошницата и да прави поръчки с избор на начин на плащане.

**2. Функционалност**

**Общи функции:**

* Вход за **собственик** и **клиент**.
* Разделено съдържание за двете роли.
* Управление на продукти и поръчки.
* Разплащане с **кредитна карта** или **наложен платеж**.
* Допълнителни такси при определени условия (доставка и плащане в брой).

**Функции за собственика:**

* Вход с парола (12345).
* Добавяне, редактиране и премахване на продукти.
* Списък и сортиране на продукти (по име, цена, количество, ID).
* Разглеждане на поръчки с детайли.
* Преглед на най-продаваните продукти.

**Функции за клиента:**

* Вход само с име.
* Разглеждане на наличните продукти.
* Добавяне на продукти в кошницата.
* Изчисляване на крайната цена с допълнителни такси.
* Завършване на поръчката и избор на начин на плащане.

**3. Архитектура**

Проектът е разделен на **Back-end** (Go) и **Front-end** (HTML, CSS, JavaScript).

**Структура на проекта:**

/ecommerce

│── cmd/api/main.go # Главен файл за стартиране на сървъра

│── internal/database # База данни и модели

│── internal/product # Продукти

│── internal/order # Поръчки

│── internal/user # Потребители

│── internal/payment # Плащания

│── ui # Front-end файлове

│ ├── index.html # Основен интерфейс

│ ├── card.html # Форма за разплащане с карта

│ ├── app.js # Логика на клиента

│ ├── styles.css # Стилове

**4. Реализация**

**4.1. Back-end (Go + SQLite)**

**main.go (Главен файл за стартиране на сървъра)**

* Зарежда **Gin router**.
* Свързва се към **SQLite база данни**.
* Регистрира **API ендпойнти** за продукти, поръчки и плащания.

**database.go (Инициализация на базата)**

* Отговаря за отварянето на базата и създаването на таблици.
* Основните таблици са **products, orders, order\_items, users, payments**.

**product.go (Модел за продукти)**

* Дефинира структурата на продуктите с полета: ID, Name, Description, Price, Quantity.
* **Функции:**
  + CreateProduct
  + GetProductByID
  + UpdateProduct
  + DeleteProduct
  + ListProducts

**order.go (Модел за поръчки)**

* Записва поръчки с **клиентски данни, артикули, метод на плащане** и **финална сума**.
* **Функции:**
  + CreateOrder
  + GetOrderByID
  + ListOrders

**payment.go (Разплащания)**

* Записва информация за плащанията.
* Поддържа **кредитна карта** и **наложен платеж**.

**4.2. Front-end (HTML, CSS, JavaScript)**

**index.html (Главен интерфейс)**

* Съдържа три основни секции:
  + **Login Page** (вход за потребители).
  + **Owner Page** (таблици с продукти и поръчки).
  + **Customer Page** (магазин, кошница и разплащане).
* Използва **Bootstrap 5** за стилизиране.

**card.html (Плащане с карта)**

* Съдържа **форма за въвеждане на данни на карта**.
* Изпраща данните след натискане на „Pay“.

**app.js (Основен JavaScript файл)**

* Управлява динамичните действия на страницата.
* **Функционалност:**
  + Регистрира **event listeners** за всички бутони.
  + Обработва **вход за клиенти и собственици**.
  + Създава заявки към API за **листване, добавяне и премахване на продукти**.
  + Управлява кошницата и изчислява крайната сума с допълнителни такси.
  + При плащане с карта отваря card.html.

**5. Модел на данните**

**Таблица: Products**

| **Поле** | **Тип** |
| --- | --- |
| id | TEXT PRIMARY KEY |
| name | TEXT |
| description | TEXT |
| price | REAL |
| quantity | INTEGER |

**Таблица: Orders**

| **Поле** | **Тип** |
| --- | --- |
| id | TEXT PRIMARY KEY |
| customer\_name | TEXT |
| phone | TEXT |
| email | TEXT |
| address | TEXT |
| payment\_method | TEXT |
| final\_total | REAL |

**6. Конфигурация**

* Използва **Go Modules** за управление на зависимостите.
* SQLite база данни се зарежда в internal/database/database.go.
* API сървърът използва **Gin framework**.

**7. Използвани технологии и библиотеки**

* **Go** – за back-end логика.
* **Gin** – за REST API.
* **SQLite** – база данни.
* **Bootstrap 5** – за front-end дизайн.
* **Vanilla JavaScript** – за front-end логика.

**8. Проблеми и решения**

* **Презаписване на event listeners:** решено чрез DOMContentLoaded.
* **Проблеми със зареждането на кеширани файлове:** решено чрез **изчистване на кеша** или **инкогнито режим**.
* **Неуспешно записване на поръчки при плащане с карта:** решено чрез първо записване в базата и после отваряне на card.html.

**9. Референции**

* [Gin Framework](https://github.com/gin-gonic/gin)
* [SQLite Documentation](https://www.sqlite.org/index.html)
* [Bootstrap 5](https://getbootstrap.com/)
* [MDN Fetch API](https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API)
* [Go SQLite3](https://github.com/mattn/go-sqlite3)