Лабораторная работа

Тема: Разработка веб-приложения с использованием Python и SQLite

Введение

Современные веб-приложения играют важную роль в автоматизации бизнес-процессов, предоставлении сервисов пользователям и управлении данными.
Python — один из наиболее популярных языков программирования для разработки веб-приложений, благодаря своей простоте и обширной экосистеме библиотек.
В рамках данной лабораторной работы изучаются основы создания веб-приложений с использованием фреймворка FastAPI и базы данных SQLite.

Цель и задачи

Цель работы:
Разработать простое веб-приложение с функциональностью взаимодействия с базой данных.

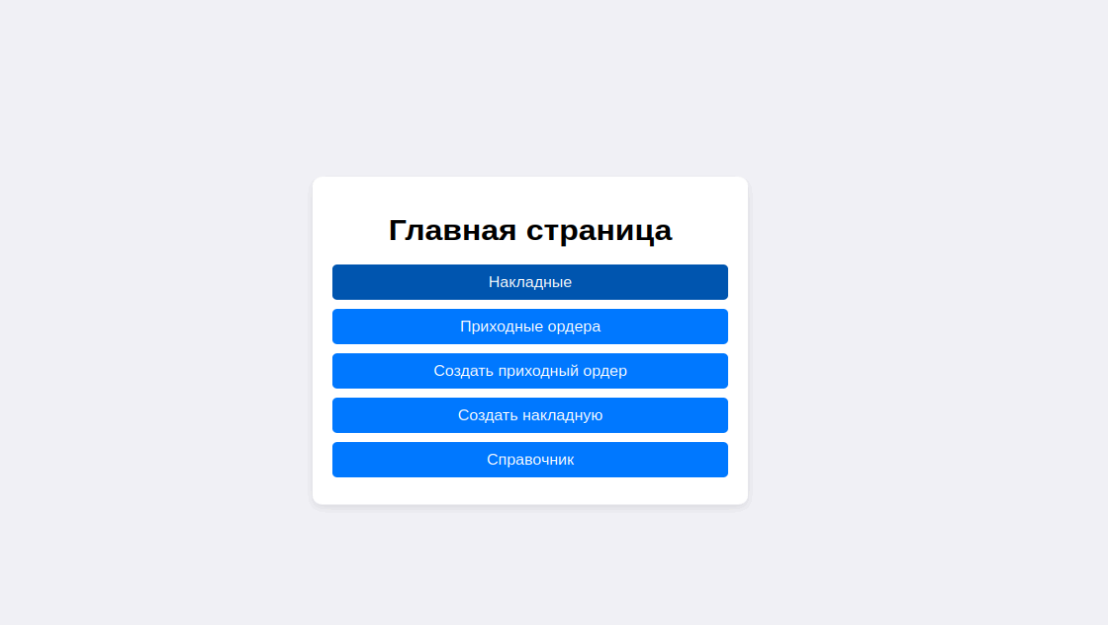
Задачи:

1. Изучить основы работы с FastAPI.
2. Настроить базу данных SQLite и создать модели данных.
3. Разработать маршруты для взаимодействия с пользователем.
4. Реализовать HTML-шаблоны для отображения данных.
5. Проверить функциональность приложения.

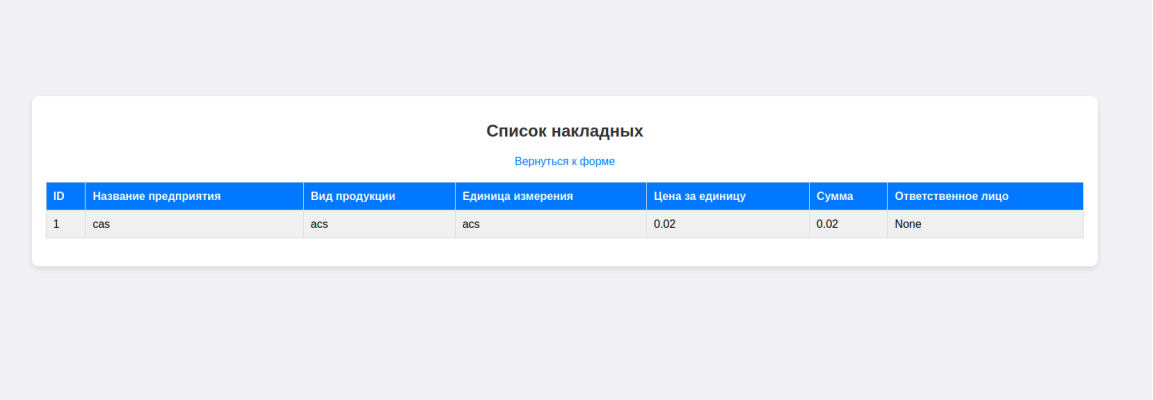
Структура проекта

Проект состоит из следующих файлов и директорий:
- `main.py`: основной скрипт приложения, содержит маршруты и логику работы сервера.
- `models.py`: файл с определением моделей данных для базы данных.
- `req.txt`: список зависимостей, необходимых для работы проекта.
- `templates`: директория с HTML-шаблонами для визуализации данных.
- `test.db`: база данных SQLite для хранения данных приложения.

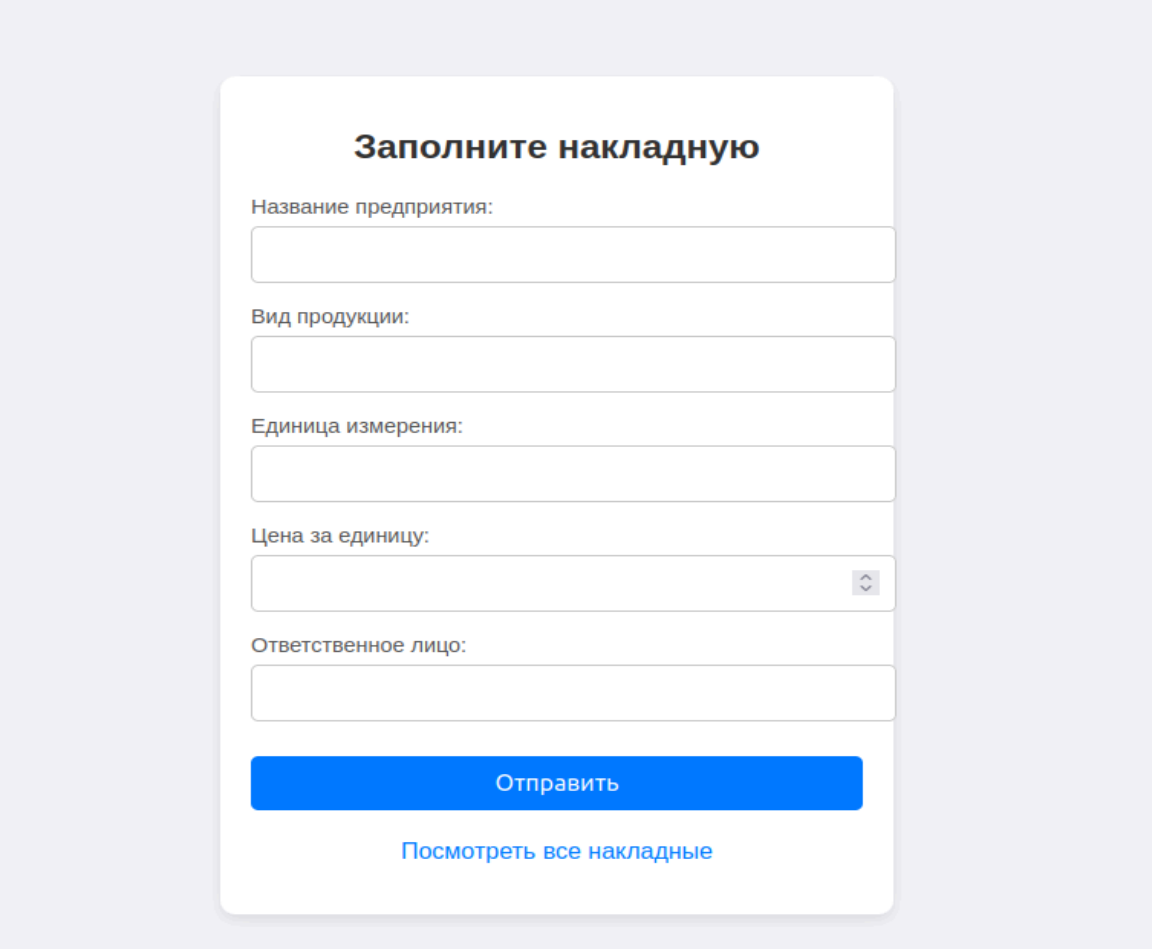
Главная страница



Накладные



Создание накладных



Практическая часть

Основной файл приложения `main.py` содержит ключевые маршруты, такие как:

- `/`: отображение главной страницы.
- `/add`: маршрут для добавления данных.
- `/view`: отображение данных из базы.

```
from sqlalchemy import Column, Integer, String, Float, create_engine, Date, DateTime

from sqlalchemy.ext.declarative import declarative_base

from sqlalchemy.orm import sessionmaker

from sqlalchemy import func

Base = declarative_base()

class Invoice(Base):

    __tablename__ = "invoices"

    id = Column(Integer, primary_key=True, index=True)

    company_name = Column(String)

    product_type = Column(String)

    unit_of_measurement = Column(String)

    price_per_unit = Column(Float)

    total_sum = Column(Float)

    responsible_person = Column(String)

class IncomingOrder(Base):

    __tablename__ = "incoming_orders"


    id = Column(Integer, primary_key=True, index=True)

    company_name = Column(String, nullable=False)  # Название предприятия

    product_type = Column(String, nullable=False)  # Вид продукции

    storage = Column(String, nullable=False)  # Склад
```

```
    registration_number = Column(String, nullable=False)
    unit_of_measurement = Column(String, nullable=False)  # Единица измерения
    price_per_unit = Column(Float, nullable=False)  # Цена за единицу
    total_sum = Column(Float, nullable=True)  # Сумма (можно рассчитать)
    date_created = Column(DateTime, nullable=False, server_default=func.now())
    chief_accountant = Column(String, nullable=True)  # Главный бухгалтер
    cashier = Column(String, nullable=True)  # Кассир


class Directory(Base):
    __tablename__ = "directory"
    id = Column(Integer, primary_key=True, index=True)
    company_name = Column(String, nullable=False)
    product_type = Column(String, nullable=False)
    code = Column(String, nullable=False)
    storage_code = Column(String, nullable=False)
    responsible_person = Column(String, nullable=True)


# Database setup
DATABASE_URL = "sqlite:///./test.db"
engine = create_engine(DATABASE_URL)
Base.metadata.create_all(bind=engine)
SessionLocal = sessionmaker(autocommit=False, autoflush=False, bind=engine)
```

В файле `models.py` описаны классы моделей с использованием SQLAlchemy. Пример модели:

Описание файла main.py

Файл `main.py` выполняет роль основного приложения. Он содержит маршруты для взаимодействия с пользователем и обработки данных. Используется фреймворк FastAPI для создания веб-сервера. Пример ключевых маршрутов:
- `/`: отображение главной страницы.
- Другие маршруты (например, для добавления, изменения данных) могут быть добавлены по мере необходимости.

Описание файла models.py

Файл `models.py` содержит описание моделей данных для базы данных SQLite. Для взаимодействия с базой данных используется библиотека SQLAlchemy. Пример структуры модели данных:
- `User`: модель для хранения данных о пользователях.
- `Task`: модель для работы с заданиями (если такая имеется).

Работа с шаблонами и базой данных

Шаблоны, расположенные в папке `templates`, используются для динамической генерации HTML-страниц. Они позволяют отображать данные из базы данных в удобочитаемой форме. База данных `test.db` служит для хранения информации, связанной с пользователями и другими сущностями.

Вывод

В ходе выполнения лабораторной работы был разработан базовый прототип веб-приложения. Проект включает в себя настройку базы данных, создание моделей и маршрутов, а также использование HTML-шаблонов для представления данных пользователям.