**Федеральное государственное автономное образовательное учреждение высшего образования   
«Национальный исследовательский университет   
“Высшая школа экономики”**

**Московский институт электроники и математики им. А.Н. Тихонова НИУ ВШЭ**

**Департамент компьютерной инженерии**

Группа БИВ 233

Бригада номер 3

Артемьев Антон Андреевич, Голуб Владислав Витальевич, Зубарев Егор Вячеславович

Оглавление

[Введение 2](#_Toc169215543)

[Описание проекта 2](#_Toc169215544)

[Цели проекта 2](#_Toc169215545)

[Функциональные возможности 2](#_Toc169215546)

[Установка и запуск 2](#_Toc169215547)

[Системные требования 2](#_Toc169215548)

[Установка и запуск программы 3](#_Toc169215549)

[Скачивание Anaconda 3](#_Toc169215550)

[Установка Anaconda на Windows 3](#_Toc169215551)

[Распаковка архива проекта 3](#_Toc169215552)

[Запуск программы 3](#_Toc169215553)

[Описание скриптов 3](#_Toc169215554)

[Описание используемых модулей 6](#_Toc169215555)

[Архитектура системы 6](#_Toc169215556)

[Взаимодействие компонентов 7](#_Toc169215557)

[Заключение 7](#_Toc169215558)

# Введение

## Описание проекта

Наш проект представляет собой интернет-магазин, разработанный для упрощения процесса покупок и управления товарами для пользователей. Интернет-магазин позволяет пользователям просматривать, искать и покупать товары через удобный веб-интерфейс, а администраторам - управлять ассортиментом, отслеживать заказы и обрабатывать транзакции.

## Цели проекта

Основной целью проекта является создание платформы, которая:

1. **Обеспечивает удобный и интуитивно понятный интерфейс для покупателей**:
   * Пользователи могут легко выбирать и покупать товары.
2. **Облегчает управление магазином для администраторов**:
   * Администраторы могут добавлять, редактировать и удалять товары из каталога.
   * Система позволяет отслеживать заказы и управлять запасами на складе.
   * Включает функции обработки транзакций.
3. **Обеспечивает безопасность и надежность**:
   * Защита данных пользователей и обработка платежей через безопасные каналы.
   * Хеширование паролей.

## Функциональные возможности

1. **Каталог товаров**:
   * Просмотр списка товаров, их описания и цен.
2. **Оформление заказа**:
   * Выбор товара и его количества.
   * Оплата через платежный сервис.
3. **Управление аккаунтом**:
   * Регистрация и авторизация пользователей.
4. **Административная панель**:
   * Управление товарами (добавление, редактирование, удаление).
   * Обработка заказов.
5. **Интеграция с платежными системами**:
   * Поддержка различных способов оплаты (кредитные карты, электронные кошельки и так далее…).

# Установка и запуск

## Системные требования

* **Операционная система: Windows**
* **Язык программирования: Python 3.x (дистрибуив anaconda)**
* **Аппаратные требования:**
  + **Процессор: Современный процессор с минимальной тактовой частотой 1 ГГц**
  + **Оперативная память: 2 ГБ и выше**
  + **Дисковое пространство: Не менее 300 МБ свободного места**

## Установка и запуск программы

### **Скачивание Anaconda**

1. **Откройте браузер и перейдите на официальный сайт Anaconda: *https://www.anaconda.com/products/distribution.***
2. **Выберите версию для Windows.**
3. **Нажмите кнопку "Download" для начала загрузки установочного файла.**

### **Установка Anaconda на Windows**

1. **После завершения загрузки запустите установочный файл (обычно это файл с расширением .exe).**
2. **Появится окно приветствия, нажмите "Next".**
3. **Примите лицензионное соглашение, выбрав "I Agree".**
4. **Выберите вариант установки "Just Me" и нажмите "Next".**
5. **Выберите место для установки Anaconda и нажмите "Next". Рекомендуется оставить путь по умолчанию.**
6. **Очень важно: отметьте опцию "Add Anaconda to my PATH environment variable" и "Register Anaconda as my default Python 3.9" (если у вас другая версия Python, она будет указана там). Затем нажмите "Install".**
7. **После завершения установки нажмите "Next", а затем "Finish".**

## Распаковка архива проекта

1. **Скачайте архив с проектом.**
2. **Распакуйте архив в желаемую директорию.**

## Запуск программы

**Запустите скрипт *"run.bat"***

# Описание скриптов

 **db\_session.py** Скрипт для инициализации и создания сессий базы данных.

* global\_init(db\_file)
  + Вход: db\_file (строка, путь к файлу базы данных).
  + Выход: Создаёт и инициализирует базу данных.
* create\_session()
  + Вход: Нет входных параметров.
  + Выход: Создаёт и возвращает новую сессию базы данных.

 **items.py** Скрипт для описания модели товаров и формы ввода данных о товарах.

* Класс Item
  + Модель таблицы item с колонками id, name, description, price, img, quantity.
* Класс ItemsForm
  + Форма для добавления и редактирования товаров с полями name, description, price, quantity, img.

 **orders.py** Скрипт для описания модели заказов и формы ввода данных о заказах.

* Класс Orders
  + Модель таблицы order с колонками id, buy, buyer, quantity, condition, paid\_for.
* Класс OrderForm
  + Форма для создания заказа с полем quantity.

 **sent\_email.py** Скрипт для отправки писем с информацией о заказе.

* send\_email\_order(mail, id, name, des, money, img=False)
  + Вход: mail (электронная почта получателя), id (идентификатор товара), name (название товара), des (описание товара), money (сумма к оплате), img (опционально, путь к изображению).
  + Выход: Отправляет письмо с информацией о заказе.

 **users.py** Скрипт для описания модели пользователей и форм авторизации и регистрации.

* Класс User
  + Модель таблицы user с колонками id, surname, name, age, address, email, moder, hashed\_password.
  + Методы: check\_password(password) - проверяет пароль.
* Класс LoginForm
  + Форма для авторизации с полями email, password, remember\_me.
* Класс RegisterForm
  + Форма для регистрации с полями surname, name, age, address, email, password.
* Класс ProfForm
  + Форма для редактирования профиля с полями surname, name, age, address, email.

**main.py**

* base()
  + Отображает главную страницу с товарами.
* load\_user(user\_id)
  + Загружает пользователя по идентификатору.
* logout()
  + Выход из текущего аккаунта и перенаправление на главную страницу.
* login()
  + Отображает страницу авторизации и обрабатывает авторизацию. При успешной авторизации перенаправляет на главную страницу, иначе отображает сообщение об ошибке.
* reg()
  + Отображает страницу регистрации и обрабатывает регистрацию нового пользователя. При успешной регистрации перенаправляет на главную страницу, иначе отображает сообщение об ошибке.
* add\_item()
  + Доступно только администраторам. Отображает страницу добавления товара и обрабатывает добавление нового товара. При успешном добавлении перенаправляет на главную страницу.
* delete\_jobs(id)
  + Доступно только администраторам. Удаляет товар по идентификатору и перенаправляет на главную страницу. Если товар не найден, вызывает ошибку 404.
* edit\_jobs(id)
  + Доступно только администраторам. Отображает страницу редактирования товара и обрабатывает изменения. При успешном редактировании перенаправляет на главную страницу.
* profile()
  + Отображает страницу редактирования профиля текущего пользователя и обрабатывает изменения. При успешном редактировании остаётся на странице профиля.
* make\_order(id)
  + Отображает страницу оформления заказа для товара по идентификатору и обрабатывает заказ. При успешном заказе открывает страницу оплаты и перенаправляет на главную страницу.
* view\_orders()
  + Отображает страницу со списком заказов текущего пользователя.
* admin\_orders()
  + Доступно только администраторам. Отображает страницу со списком всех заказов.
* edit\_status(id, st)
  + Доступно только администраторам. Обновляет статус заказа по идентификатору и перенаправляет на страницу со списком всех заказов. Если заказ не найден или статус некорректен, вызывает ошибку 404.
* delete\_order(id)
  + Доступно только администраторам. Удаляет заказ по идентификатору и возвращает количество товара на склад, после чего перенаправляет на страницу со списком всех заказов. Если заказ не найден, вызывает ошибку 404.
* not\_found(error)
  + Возвращает JSON с сообщением об ошибке 404.

**Также присутствует скрипт для работы через API:**

 **items\_resource.py**

* abort\_if\_news\_not\_found(user\_id)
  + Вход: user\_id (идентификатор товара).
  + Выход: Проверяет наличие товара и вызывает ошибку 404, если товар не найден.
* get\_item(items\_id)
  + Вход: items\_id (идентификатор товара).
  + Выход: Возвращает JSON с данными о товаре.
* get\_items()
  + Вход: Нет входных параметров.
  + Выход: Возвращает JSON со списком всех товаров.

# Описание используемых модулей

**1. os**

Модуль для взаимодействия с операционной системой, например, для управления файловой системой.

**2. sys**

Модуль для доступа к некоторым переменным и функциям, которые сильно взаимодействуют с интерпретатором Python.

**3. flask**

Основной фреймворк для создания веб-приложений.

**4. flask\_sqlalchemy**

Расширение Flask для интеграции с базой данных SQLAlchemy.

**5. flask\_migrate**

Расширение Flask для управления миграциями базы данных.

**6. flask\_login**

Расширение Flask для управления аутентификацией пользователей.

**7. werkzeug.security**

Модуль для обеспечения безопасности, например, для хэширования паролей.

# Архитектура системы

Архитектура проекта состоит из следующих компонентов:

1. **Frontend**:
   * HTML/CSS: Используются для создания пользовательского интерфейса.
2. **Backend**:
   * Flask: Веб-фреймворк для обработки HTTP-запросов и генерации ответов.
   * SQLAlchemy: ORM для работы с базой данных.
   * Модули для управления пользователями, обработкой заказов и других бизнес-логик.
3. **База данных**:
   * Используется для хранения информации о пользователях, товарах, заказах и других данных.
   * SQLite для разработки и продакции

# Взаимодействие компонентов

1. **Пользовательский интерфейс**:
   * Пользователь взаимодействует с веб-интерфейсом, отправляя запросы (например, просмотр товаров, добавление в корзину, оформление заказа).
2. **Контроллеры Flask**:
   * Flask принимает запросы от пользователя, обрабатывает их в контроллерах и вызывает соответствующие функции.
3. **Модели базы данных**:
   * Контроллеры взаимодействуют с моделями, чтобы получить или сохранить данные в базе данных.
4. **Отправка ответов**:
   * После обработки запроса данные отправляются обратно пользователю через сгенерированные HTML-страницы или JSON-ответы.
5. **Административная панель**:
   * Администраторы имеют доступ к отдельным маршрутам для управления товарами и заказами.
   * Панель позволяет добавлять новые товары, редактировать существующие и обрабатывать заказы.

# Заключение

Проект предоставляет удобную платформу для создания интернет-магазина с широкими возможностями для пользователей и администраторов. Архитектура системы обеспечивает гибкость и масштабируемость, а также легкость в обслуживании и расширении функциональности.