**UrbanUniversity**

**Дипломная работа**

**На тему**

«Использование AI в поседневной жизни»

Выполнил: студент

Приходько Д.В.

Направление подготовки:

Python разработчик

Проверил(а):

Кирилл Батькович

Новороссийск

2024

#### 1. Введение

Обоснование выбора темы: В современных условиях активно развиваются технологии искусственного интеллекта, в частности, глубокое обучение. Одним из популярных направлений является обработка изображений, включая удаление нежелательных объектов. Данная работа посвящена разработке веб-приложения, которое позволяет пользователям удалять лишние объекты с изображений с использованием AI.

Определение цели и задач исследования: Цель дипломной работы — разработка платформы, которая предоставляет пользователям возможность удалять нежелательные объекты с изображений, используя предварительно обученные модели ИИ. Задачи:

1. Изучение существующих методов обработки изображений.
2. Разработка интерфейса для загрузки и обработки изображений.
3. Интеграция с моделью ИИ для удаления объектов.
4. Разработка системы регистрации и авторизации пользователей.

#### 2. Основные понятия и определения

1. Искусственный интеллект (AI).
2. Глубокое обучение.
3. Обработка изображений.
4. Django (веб-фреймворк).

#### 3. Методы и подходы к разработке

Выбор технологии: Для разработки веб-приложения был выбран фреймворк Django благодаря его гибкости, встроенной системе авторизации и возможностям расширяемости.

Архитектура веб-приложения: Приложение реализовано на Django с использованием базы данных для хранения изображений и информации о пользователях. Взаимодействие с моделью ИИ осуществляется через API.

Обеспечение безопасности: Важной частью проекта является безопасность данных пользователей, включая механизмы авторизации, аутентификации и защиту загруженных изображений.

#### 4. Обзор популярных инструментов для разработки веб-приложений на Python

Django: Django был выбран как основной фреймворк благодаря своей мощной системе ORM, возможностям быстрой разработки и встроенной поддержке авторизации.

Flask: Рассматривался как альтернатива для небольших приложений, но был отвергнут из-за меньшего функционала «из коробки».

FastAPI: Подходит для разработки высокопроизводительных API, но для работы с интерфейсом был менее удобен по сравнению с Django.

#### 5. Проектирование приложения

Планирование и анализ требований: Основные требования включают возможность загрузки изображений, их обработки с помощью модели ИИ и управление пользователями (регистрация, авторизация).

Основные требования:

1. Пользователи могут регистрироваться и авторизовываться.
2. Приложение должно позволять загружать изображения и применять к ним ИИ-модель для удаления объектов.

Технические требования:

1. Язык разработки: Python.
2. Веб-фреймворк: Django.
3. База данных: SQLite (для разработки) с возможностью использования других СУБД в продакшене.
4. Модель для удаления объектов IOPaint: предварительно обученная модель на основе глубокой свёрточной нейронной сети.

#### 6. Разработка в соответствии с созданной документацией

Планирование разработки:

Этапы разработки включают создание пользовательского интерфейса, реализацию системы авторизации и интеграцию модели AI.

Разработка:

1. Создание и настройка модели пользователя в Django.
2. Реализация системы авторизации и регистрации.
3. Интеграция модели ИИ для удаления объектов с изображений.
4. Возможность сменить пароль

#### 7. Анализ и интерпретация результатов

Сравнение производительности модели: Анализ времени обработки изображений, качественные оценки удаления объектов и обратная связь от пользователей.

Также были реализованы несколько форм для различных функциональных задач, таких как регистрация, авторизация и смена пароля. Django предоставляет удобные механизмы для работы с формами, что делает процесс их создания и обработки более удобным и безопасным.

Для регистрации пользователя в проекте были использованы два подхода. Первый — это стандартная Django форма UserCreationForm, которая предоставляет базовую логику для создания пользователя с проверкой полей пароля и его подтверждения. Этот подход упрощает реализацию стандартных функциональных требований.

Второй подход — создание кастомной формы с добавлением валидации и специфических требований проекта. В частности, для регистрации была добавлена проверка возраста: пользователь должен быть старше 18 лет, иначе форма выводит соответствующую ошибку. Также была добавлена валидация пароля — он должен содержать не менее 8 символов. При заполнении формы, если пользователь нарушает одно из этих требований, на экран выводится соответствующее уведомление об ошибке, что делает процесс регистрации интуитивно понятным.

Авторизация реализована с использованием встроенного механизма аутентификации Django через форму логина. Пользователь вводит свои данные (имя и пароль), и система проверяет корректность этих данных. Если данные введены верно, пользователь перенаправляется на главную страницу. В случае ошибки, система выводит уведомление, что введены неправильные имя пользователя или пароль.

Форма для смены пароля была создана с учетом дополнительных требований к безопасности. Пользователь должен ввести свой старый пароль для подтверждения права на изменение данных. Если старый пароль введен неверно, система выводит сообщение об ошибке. Новый пароль также проверяется на соответствие минимальным требованиям по длине и сложности.

При успешной смене пароля выводится сообщение о том, что пароль был изменен. Это помогает пользователю убедиться в успешном завершении операции и повысить уровень удобства использования системы.

#### Обработка ошибок и пользовательские подсказки

Для улучшения взаимодействия с пользователем были добавлены подсказки и сообщения об ошибках. При заполнении любой из форм, если были допущены ошибки (например, несовпадение паролей или несоответствие возраста), система предоставляет точную информацию о том, что нужно исправить. Это сделано для того, чтобы минимизировать фрустрацию пользователей при заполнении форм и ускорить процесс регистрации или смены пароля.

Рекомендации по дальнейшему развитию:

1. Реализация монетизации за использование ресурса
2. Добавление текстового чата
3. Реализация подобного рода еще сервисов для повышения индекса качества сайт

#### 8. Заключение

Обзор выполненной работы: В рамках дипломной работы был создан веб-сервис, который позволяет пользователям загружать изображения и удалять нежелательные объекты с помощью ИИ.

Дальнейшие планы:

1. Улучшение пользовательского интерфейса.
2. Оптимизация работы модели для обработки более сложных изображений.
3. Добавление более динамичных структур, такие как текстовый чат или афтооризация с помощью соц сетей

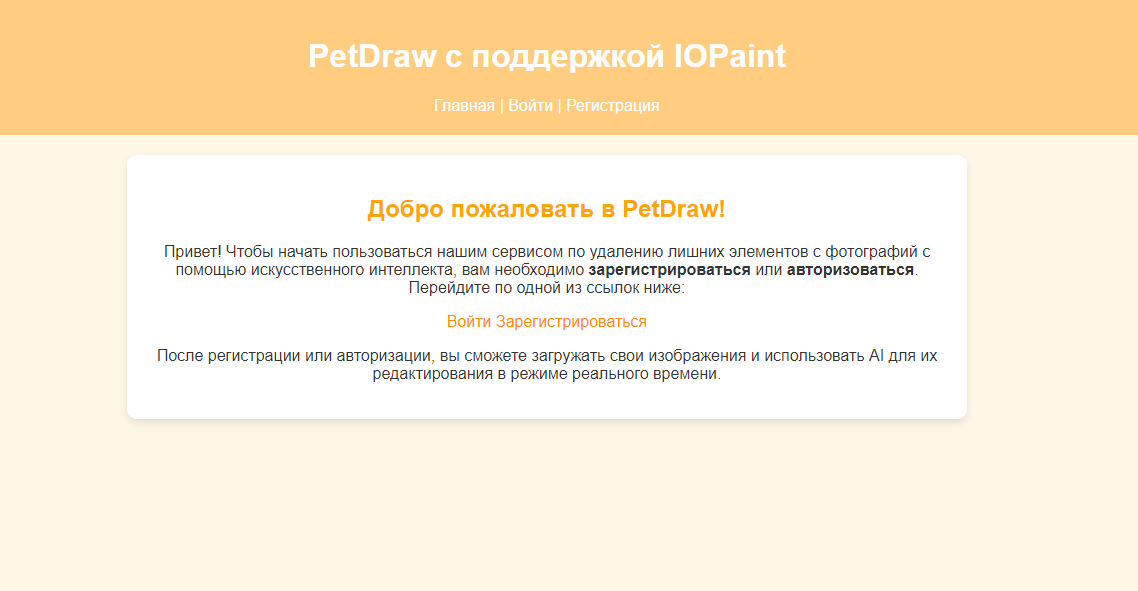
### Приложение: Django-приложение для удаления объектов с изображений

#### 1. Обзор проекта

Приложение представляет собой платформу для пользователей, где они могут загружать изображения и удалять нежелательные объекты с помощью модели ИИ. Сервис требует авторизации для загрузки и обработки изображений, а также содержит систему чата для обсуждения.

#### 2. Структура проекта

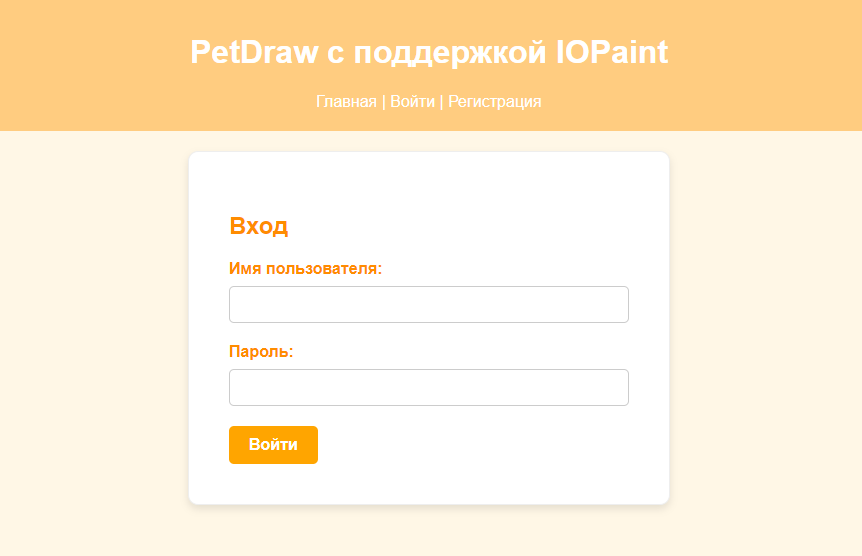
**Домашняя страница (Перед Аторизацией)**: Информация о необходимости регистрации/авторизации для работы с приложением.



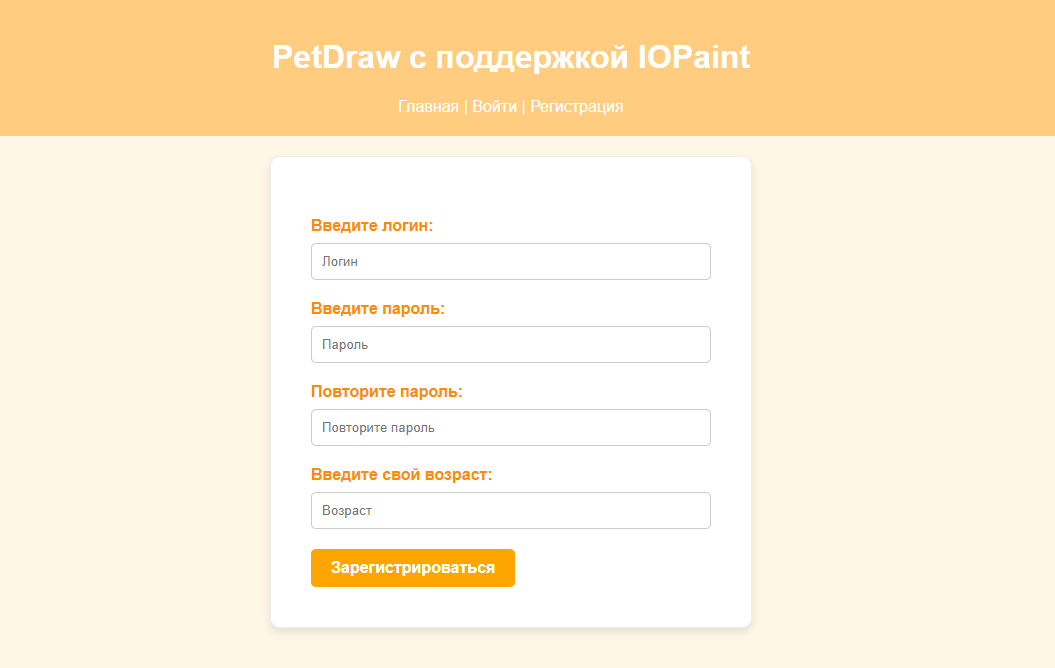
**Домашняя страница (ПослеАторизациии)**:



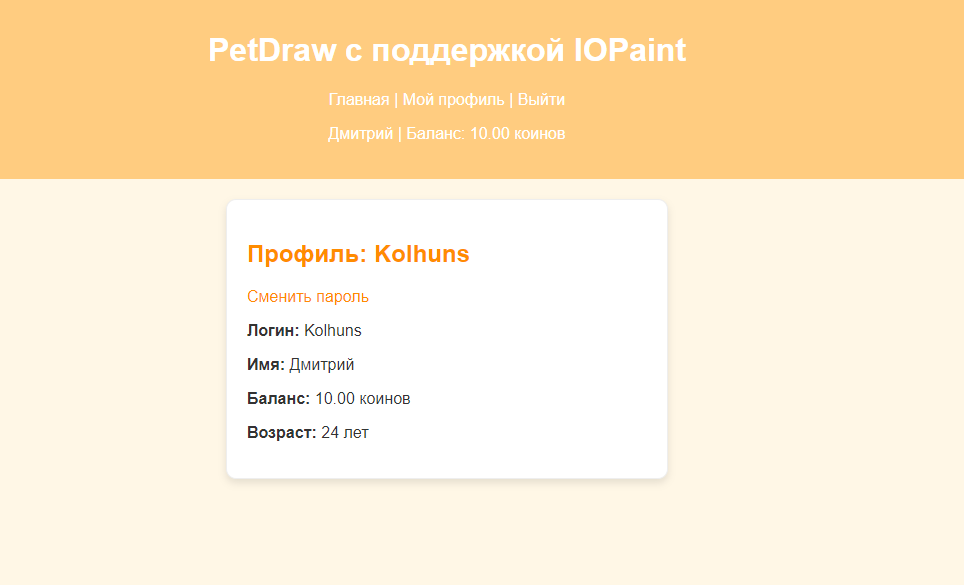
**Страница авторизации**: Пользователи могут войти в аккаунт.



**Страница авторизации**: Пользователи могут завести себе новый аккаунт.



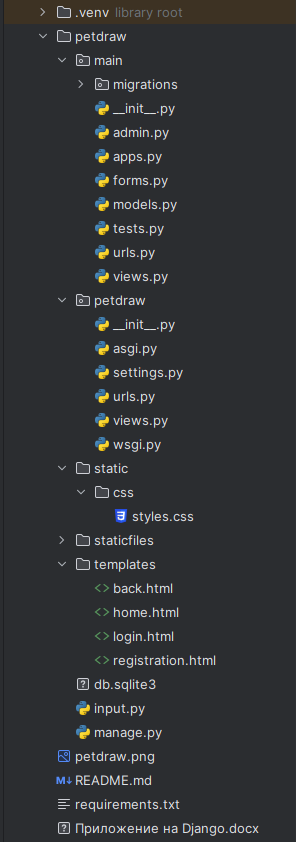
**Личный кабинет**: Пользователь видит свои обработанные изображения и может загружать новые.



**Смена пароля**: Пользователь может со свой страницы сменить свой пароль на новый

#### 

**[Приложение 1. Файловая структуры проекта](https://urban-university.ru/members/courses/course999421818026/null)**



**[Приложение 2.](https://urban-university.ru/members/courses/course999421818026/null)Список необходимых библиотек**

Django == 5.1.2  
IOPaint == 1.4.3