

## **1 ПРОГРАММА С TIME-LIMITED И START-LIMITED**

Я создал web-приложение с базой данных о пользователях и информации о них, в котором пользователь после авторизации может ознакомиться с контентом, а также добавлять информацию, которая будет размещена только на его личной странице. Другие пользователи могут посетить его страницу и посмотреть его профиль (будучи авторизованными). Данное приложение является пробной версией в его использовании и имеет платную подписку. Оно имеет ограничение по количеству авторизаций пользователя, которое определяется по его ip-адресу (часть (а) из задания), а также на количество времени, которое провел пользователь в приложении, будучи авторизованным (часть (б) из задания). Данное приложение после достижения лимитов перебрасывает на страницу с ценами на платную подписку, где говорится, что было достигнуто ограничение использования. Пользователь может оформить подписку и продолжить использование приложения без ограничений\*.

\*Оформление подписки а также функционирование приложения после подключения подписки не реализовано.

### **1.1 Задание**

Разработать простейшую программу, запрашивающую логин пользователя. Если такой логин, то выдавать об этом сообщение. После ввода информации программа должна завершать работу и сообщать пользователю о лимитах ее использования (временного или количества запусков). По достижении лимита запусков программа должна предложить пользователю приобрести ее полную версию. При повторной установке программы, она должна сообщать о своем предыдущем нахождении и сверяться с прошлыми лимитами пользования (т.е. не давать их суммарно превысить).

Выполняются две версии программы (можно сочетать в одной):

- а) Start-limited (ограничение на количество запусков – 5);
- б) Time-limited (ограничение по времени сделать не более 100 секунд).

## 1.2 Ход работы

### Вводная часть работы.

Создал WEB-приложение на фреймворк для web-приложений на языке Python.

Создаем проект test1 при помощи команды (Рисунок 1):

```
PS C:\Users\Пользователь\PycharmProjects\pythonProject> django-admin startproject test1
```

Рисунок 1 – Создание нового проекта в Django

Переходим в каталог test1 и создаем приложение main при помощи команды (Рисунок 2):

```
PS C:\Users\Пользователь\PycharmProjects\firstProject\test1> python manage.py startapp main
```

Рисунок 2 – Создание нового приложения внутри проекта

Затем же сразу создаем супер-пользователя, чтобы управлять базой данной и вносить туда изменения при помощи команды (Рисунок 3):

```
PS C:\Users\Пользователь\PycharmProjects\firstProject\test1> python manage.py createsuperuser
```

Рисунок 3 – Создание супер-пользователя

После создания базы данных, создаем миграции при помощи команды (Рисунок 4):

```
PS C:\Users\Пользователь\PycharmProjects\firstProject\test1> python manage.py makemigrations
```

Рисунок 4 – Проведение миграций для создания базы данных

Удалим всех пользователей в административной панели, которых я создавал для проверки работоспособности приложения (Рисунок 5):

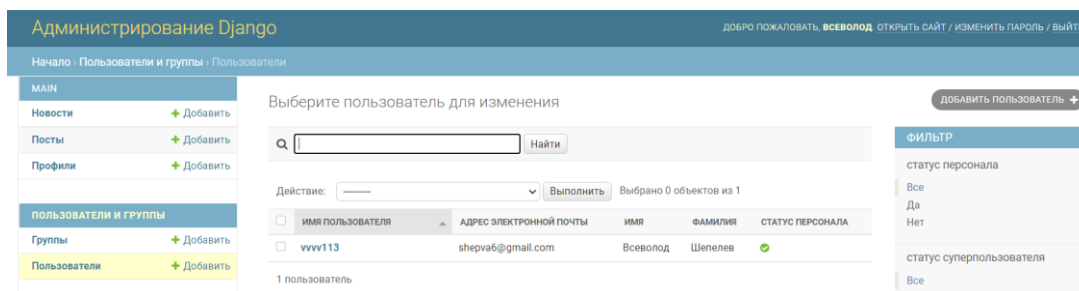


Рисунок 5 – Административная панель

Перейдем на главную страницу приложения (Рисунок 6):

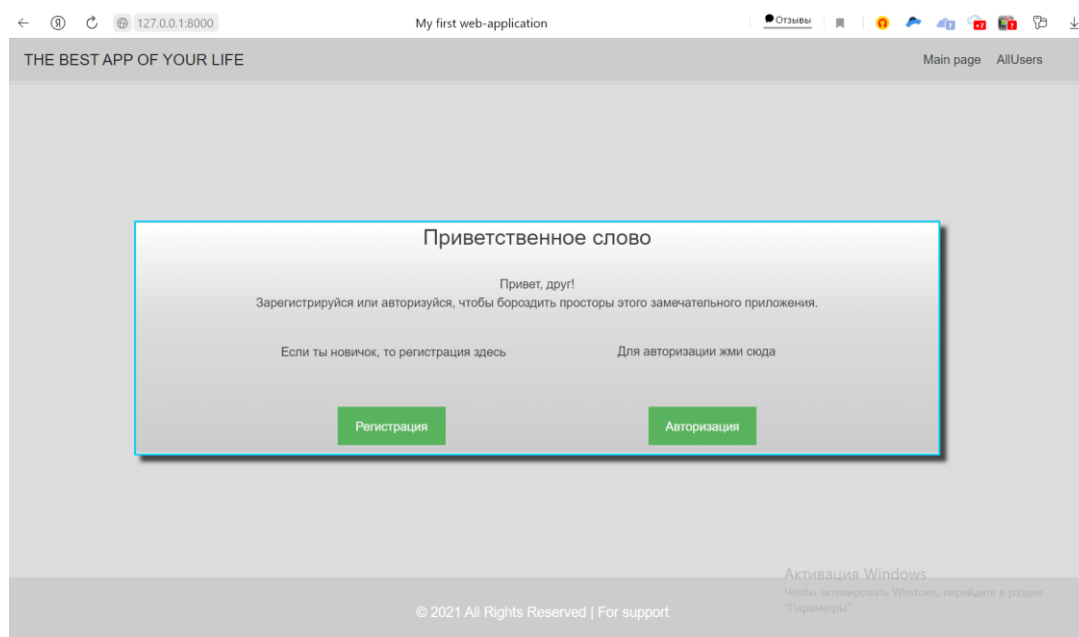


Рисунок 6 – Главная страница приложения

Перейдем на страницу регистрации и создадим первый профиль (Рисунок 7):

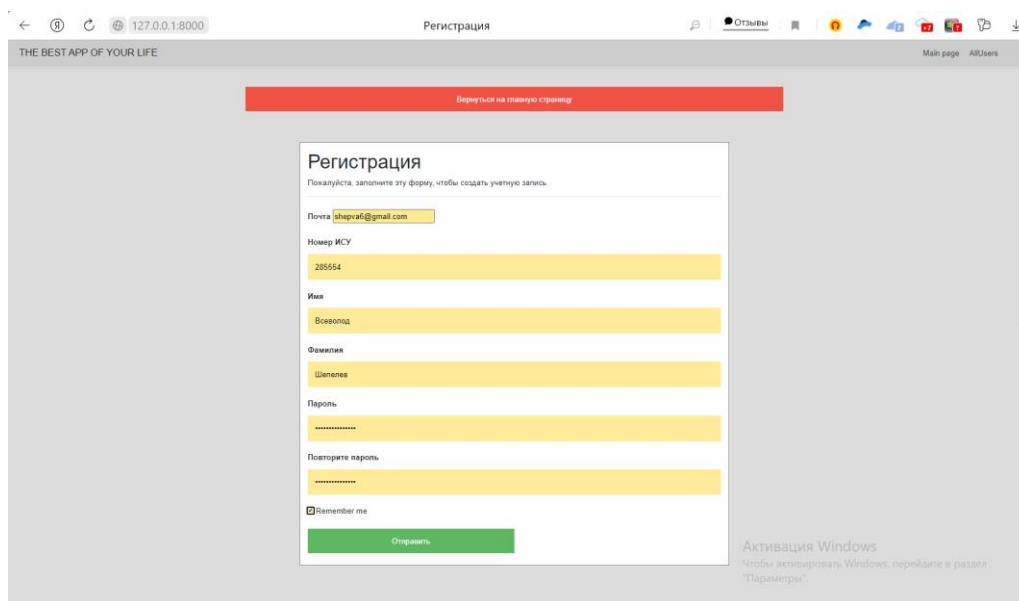


Рисунок 7 – Создание нового профиля

После чего авторизуемся для того, чтобы зайти в свой профиль (Рисунок 8):

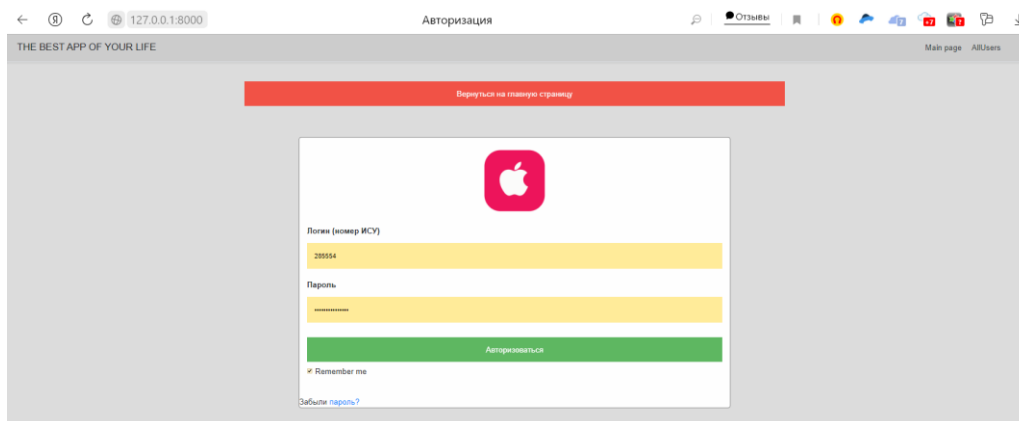


Рисунок 8 – Авторизация пользователя

После авторизации на главной странице профиля мы можем наблюдать его профиль на главной странице пользователя (Рисунок 9):

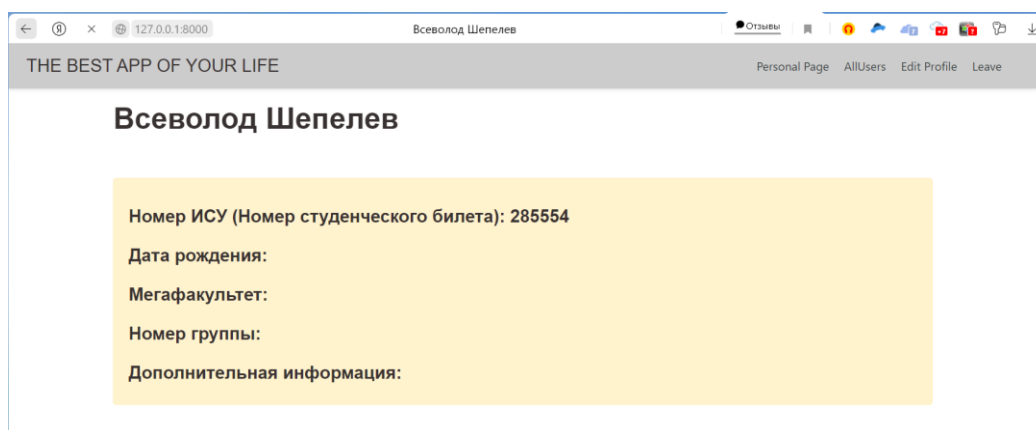


Рисунок 9 – Главная страница профиля пользователя

Перейдем на страницу редактирования профиля (Edit Profile), чтобы добавить информацию о себе (Рисунок 10):

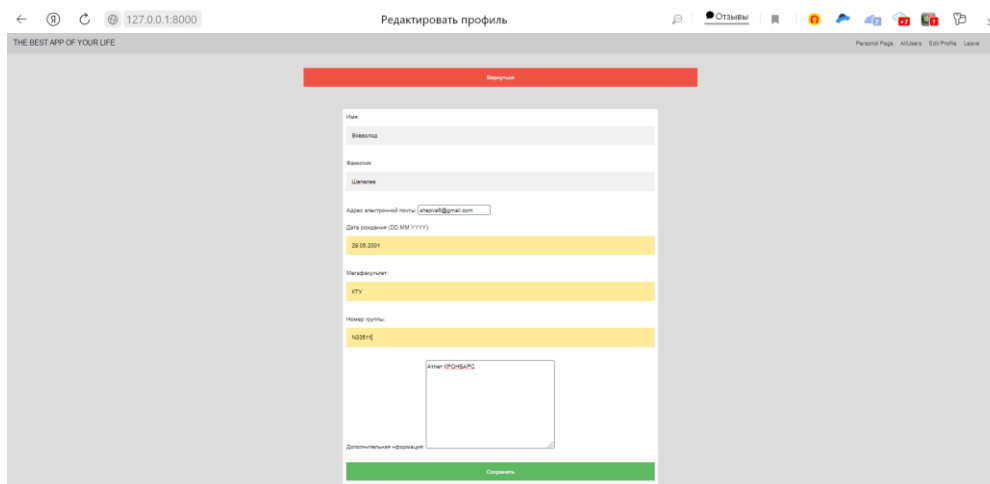


Рисунок 10 – Редактирование информации о себе в профиле

Теперь главная страница профиля стала выглядеть лучше (Рисунок 11):

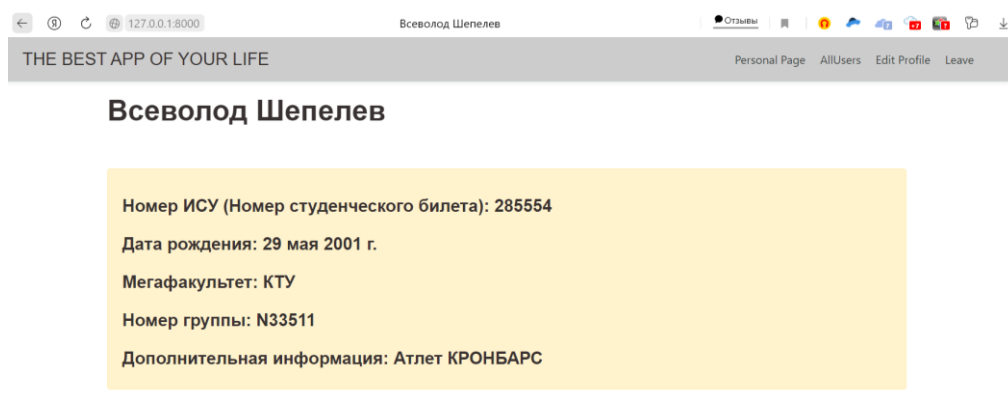


Рисунок 11 – Обновленная информация в профиле после его редактирования

Помимо этого на главной странице можно наблюдать режим работы Нестоловой на Кронверкском (Рисунок 12):

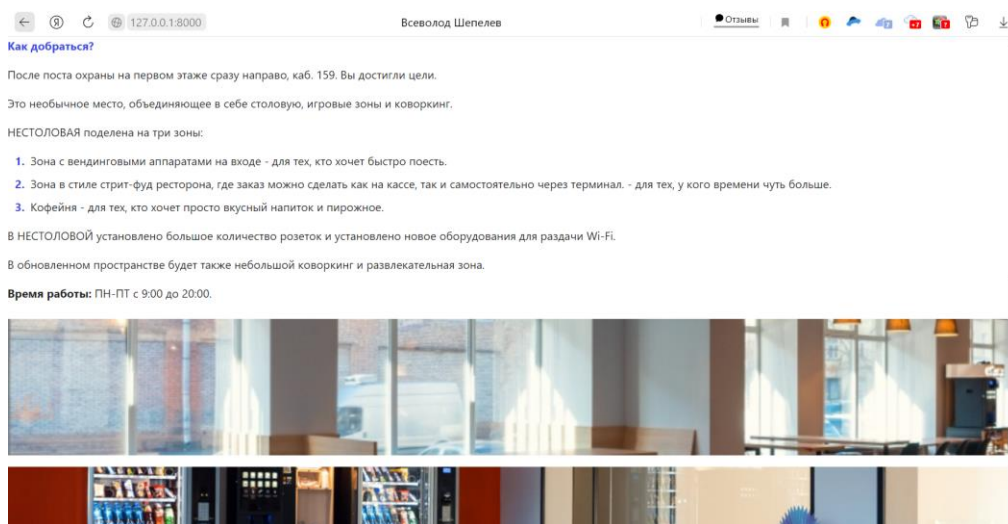


Рисунок 12 – Информация на странице пользователя

После выхода с профиля нас перебрасывает на главную страницу приложения для авторизации или регистрации.

### Часть а.

Теперь проверим Start-limited (ограничение на количество запусков, у меня установлено максимальное количество авторизаций – 5)

После 5 раз авторизаций, приложение не дает авторизоваться бой раз и перебрасывает нас на страницу, где говорится о том, что превышен один из лимитов, и предлагается оформить платную подписку приложения (Рисунок 13).

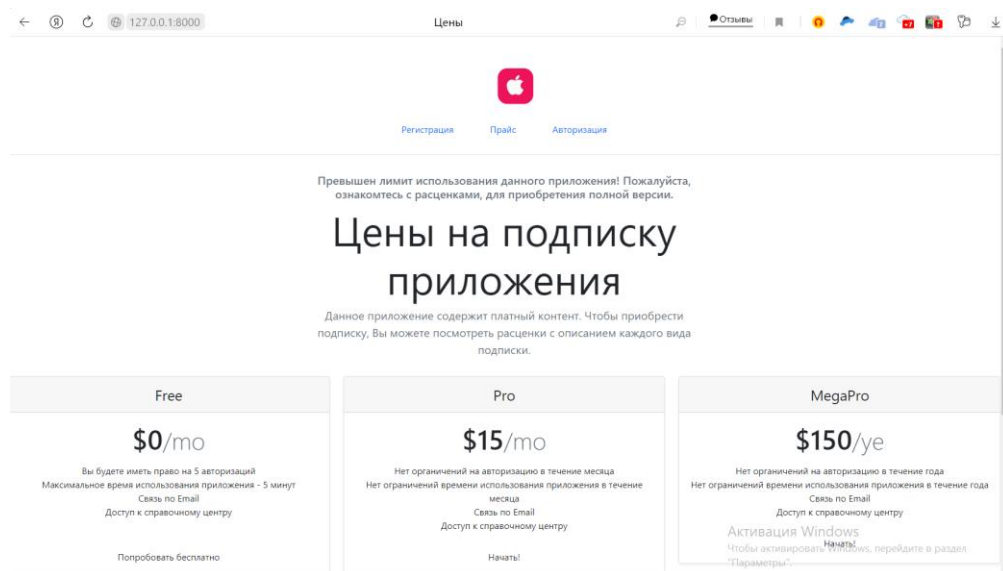


Рисунок 13 – Страница с предложением подписки

Посмотрим, что произойдет, если нажать на «Попробовать бесплатно» (Рисунок 14):

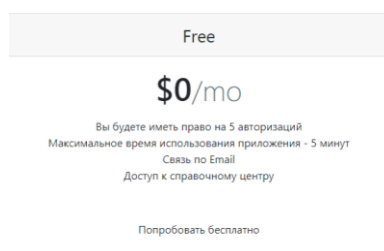


Рисунок 14 – Бесплатная версия приложения

Нас перебрасывает на страницу с регистрацией (Рисунок 15):

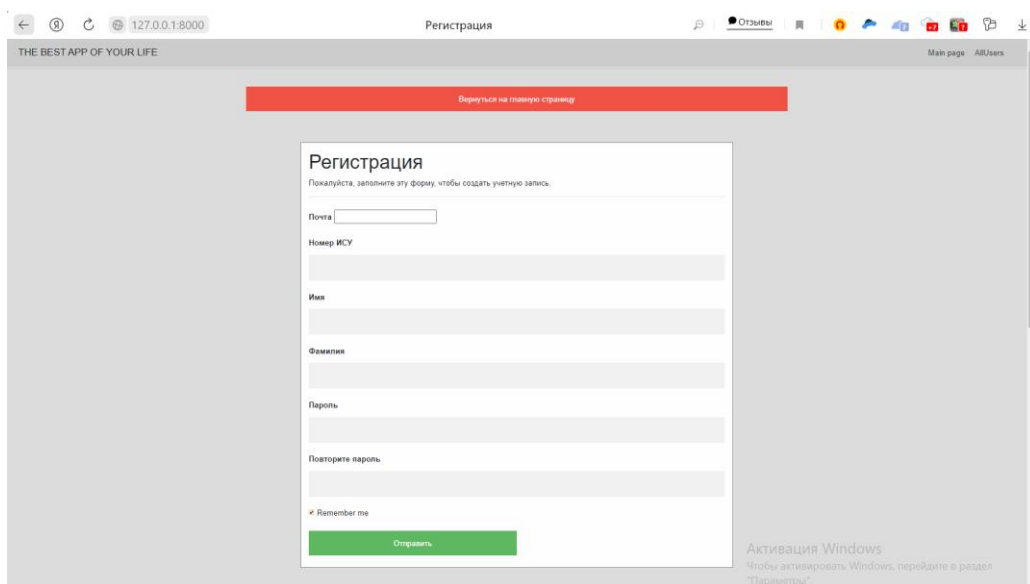


Рисунок 15 – Страница с регистрацией

Попробуем создать новую учетную запись и отправить запрос (Рисунок 16):

Рисунок 16 – Создание новой учетной записи после достижения лимитов

После отправки запроса нас снова перекидывает на страницу, где информируется что был превышен лимит (Рисунок 13).

Попробуем нажать «Начать», чтобы оформить подписку на месяц (Рисунок 17):

Рисунок 17 – Платная версия приложения

Нас перекидывает на страницу, где нужно внести данные о карте, чтобы подписка стала действительной (Рисунок 18):

Рисунок 18 – Оформление подписки на приложение

## Часть 6.

Удалим информацию (обнулим) о прошлых входах в систему в приложение в проекте Django, где я его реализовывал и проверим второй тип лимитов, а именно временного лимита.

Собираем данные о количестве времени, которые пользователь провел на своем профиле. Зайдем первый раз на 64 секунды.

В моем проекте произошла запись о том, сколько времени пользователь был авторизован всего (Рисунок 19):

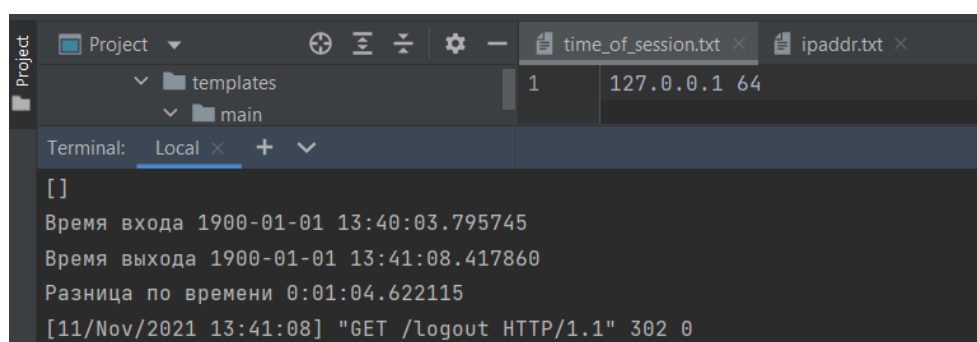


Рисунок 19 – Запись информации о длительности сессий

Теперь авторизуемся второй раз, только уже на 15 секунд. Снова данные о количестве общей авторизации сохраняются (Рисунок 20):

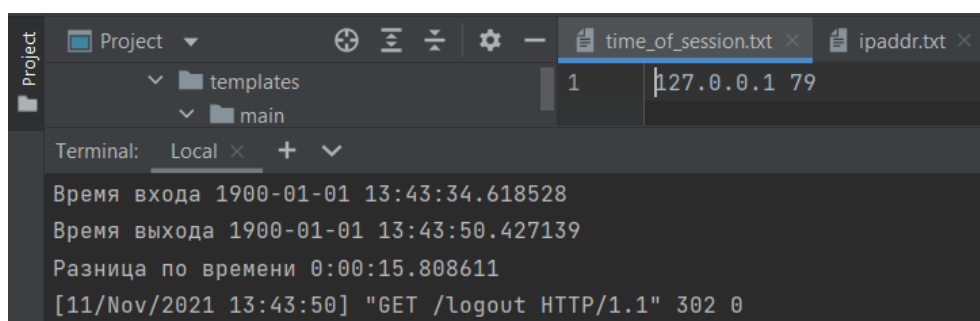


Рисунок 20 – Обновленная информация о длительности сессии (суммарное)

Зайдем третий раз на 38 секунд. Тогда общее время стало 117 секунд (время ограничения 100 секунд). При попытке пользователя зайти повторно, то есть в нашем случае, 4 раз, его перебрасывает на страницу, где информируется о том, что у него превышен лимит пользования приложением (Рисунок 21).



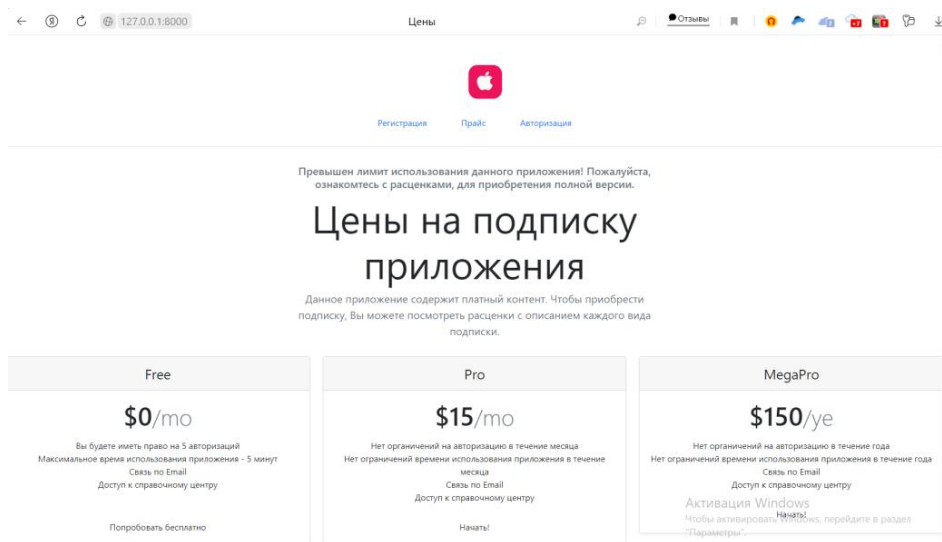


Рисунок 21 – Страница с предложением подписки

Также в настройках приложения (settings.py) я настроил время авторизации на сессии 120 секунд и после превышения этого лимита, на какую страницу пользователь не хотел бы перейти, его автоматически перебросит на страницу, где предлагается приобрести платную подписку (Рисунок 22). Данное средство, реализованное мною, сработает сразу же, если пользователь зашел в профиль единожды и провел на странице больше 120 секунд.

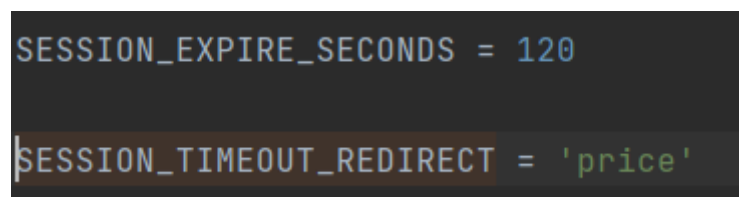


Рисунок 22 – Ограничение времени длительности сессии пользователя

## Заключительная часть выполнения работы.

Перейдем снова в панель администратора и посмотрим все существующие профили приложения (Рисунок 23):

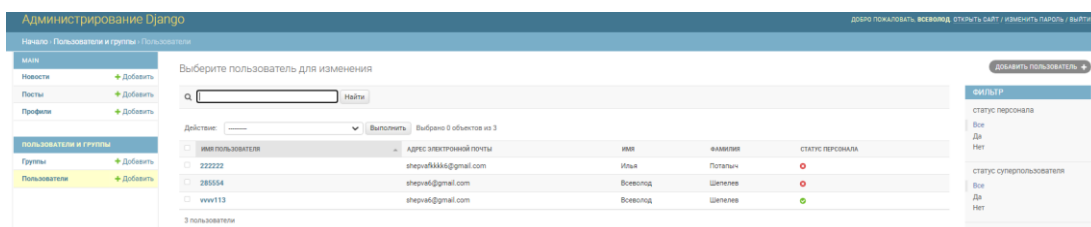


Рисунок 23 – Все профили созданные в приложении

Попробуем создать новый профиль, но уже с существующим логином (Рисунок 24):

Регистрация

Почта test@gmail.com

Пароль 332

Remember me

Отправить

Рисунок 24 – Создание профиля с уже существующим логином

Получаем ошибку о том, что профиль с таким логином уже существует (Рисунок 25):

Пользователь с таким логином уже существует.  
Попробуйте создать профиль, используя другой логин.

Регистрация

Рисунок 25 – Ошибка создания профиля

## **ЗАКЛЮЧЕНИЕ**

В ходе выполнения данной лабораторной работы было создано web-приложение и реализовано в нем оба варианта лимитов, предлагаемых в задании (ограничение на количество входов в систему с одного устройства start-limited и ограничение по времени использования приложения time-limited), а также реализована проверка на валидность данных введенных при создании профиля, не позволяющая создавать профиль при неправильных введенных данных или при введенном логине, уже существующим в базе данных. Я получил навыки создания приложений в framework Django с использованием базы данных и авторизацией и регистрацией пользователей в системе.