МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ имени академика С.П.КОРОЛЕВА»

ИНСТИТУТ ИНФОРМАТИКИ И КИБЕРНЕТИКИ

КАФЕДРА ТЕХНИЧЕСКОЙ КИБЕРНЕТИКИ

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3

Тема: «Авторизация»

Дисциплина: «Технологии сетевого программирования»

Выполнил: АшугянТ.А., гр.6303

Проверил: Кашапов А.И.

Самара 2025

СОДЕРЖАНИЕ

[Введение 3](#_Toc199594136)

[1 Аутентификация и управление пользователем 4](#_Toc199594137)

[Приложение А 5](#_Toc199594138)

[Ссылка на GitHub с проектом 5](#_Toc199594139)

[Приложение Б 6](#_Toc199594140)

[Код файла “middleware.py” 6](#_Toc199594141)

Введение

AutoService – приложение для записи на услуги в автосервис. Стек технологий приложения: Python, Django, PostgreSQL. Все компоненты конечной системы контейнеризованны с помощью Docker для обеспечения изоляции окружения, удобства развертывания и масштабирования.

Ссылка на GitHub страницу со всеми файлами проекта можно найти в приложении А.

1. Аутентификация и управление пользователем

API методы, отвечающие за аутентификацию и управление пользователем, включая методы, URL, параметры, форматы запросов и ответов, описана в таблице 1.

Таблица 1 – Описание API

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Действие | Метод | URL | Параметры | Формат запроса | Формат ответа |
| Регистрация | POST | register | name,  phone,  login,  password |  | {“message”: “registration successful”} |
| Авторизация | POST | login | login,  password |  | {“message”: “login successful”} |
| Смена пароля | POST | change-password | old\_password,  new-password |  | {“message”: “password changed successful”} |
| Выход из системы | POST | logout |  |  | {“message”: “logout successful”} |

Согласно таблице, были реализованы методы регистрации пользователя, авторизации пользователя, изменения пароля и выхода из системы. Для проверки валидности JWT токенов и обновления access токена была написана функция, код которой представлен в приложении Б. JWT токены в API запросах записываются в HTTPOnly Cookie файлы. Для ограничения доступа к определенным эндпоинтам были добавлено разрешение isAuthenticated, позволяющее доступ только аутентифицированным пользователям.

Приложение А

Ссылка на GitHub с проектом

https://github.com/Tigr195/AutoService

Приложение Б

Код файла “middleware.py”

def check\_and\_refresh\_token(request):

access\_token = request.session.get('access\_token')

refresh\_token = request.session.get('refresh\_token')

if not access\_token or not refresh\_token:

logout(request)

return False

try:

access = AccessToken(access\_token)

if access['exp'] < datetime.now(timezone.utc).timestamp():

raise TokenError("Access token expired")

return True

except TokenError:

try:

refresh = RefreshToken(refresh\_token)

new\_access = refresh.access\_token

request.session['access\_token'] = str(new\_access)

return True

except TokenError:

logout(request)

return False