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 №4

Тема: «Разработка пользовательского интерфейса и взаимодействие с API»

Дисциплина: «Технологии сетевого программирования»

Выполнил: Якухин И.В., гр.6303

Проверил: Кашапов А.И.

Самара 2025

СОДЕРЖАНИЕ

[Введение 3](#_Toc199594716)

[1 Разработка пользовательского интерфейса 4](#_Toc199594717)

[2 Настройка взаимодействия с сервером 5](#_Toc199594718)

[3 Работа с аутентификацией 6](#_Toc199594719)

[Приложение А 7](#_Toc199594720)

[Ссылка на GitHub с проектом 7](#_Toc199594721)

[Приложение Б 8](#_Toc199594722)

[Страница регистрации 8](#_Toc199594723)

Введение

Cardiogram – приложение с увлекательной игрой для запоминания английских слов с помощью карточек. Стек технологий приложения: Python, Django, React, PostgreSQL. Все компоненты конечной системы контенеризованы с помощью Docker для обеспечения изоляции окружения, удобства развертывания и масштабирования.

Учебный проект разработан с целью демонстрации навыков работы с базой данных и инструментами серверной и клиентской разработки, проектирования структуры приложений и API, контейнеризации.

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

1. Разработка пользовательского интерфейса

Для создания пользовательского интерфейса были написаны шаблоны в формате html/css для последующей интеграции.



Рисунок 1 – Страница регистрации

Пример страницы регистрации в формате html приведен в приложении Б.

1. Настройка взаимодействия с сервером

После создания всех html шаблонов были преобразованы представления, для работы с этими шаблонами. Были настроены переходы между адресами в шаблонах. Данные с сервера попадают в клиентское приложение и динамически отрисовываются на страницах.

1. Работа с аутентификацией

Когда пользователь регистрируется в приложении или входит в свой аккаунт, сервер создает для него JWT токены и сохраняет их в Cookie файлы

Запросы используют Cookie файлы созданные сервером для получения доступа к защищенным эндпоинтам.

Приложение А

Ссылка на GitHub с проектом

https://github.com/Tigr195/AutoService

Приложение Б

Страница регистрации

{% extends 'base.html' %}

{% block title %}Регистрация{% endblock %}

{% block content %}

<h2>Регистрация клиента</h2>

{% if error %}

<p style="color: red;">{{ error }}</p>

{% endif %}

<form method="post">

{% csrf\_token %}

<label for="name">Имя:</label><br>

<input type="text" id="name" name="name" required><br><br>

<label for="phone">Телефон:</label><br>

<input type="text" id="phone" name="phone" required><br><br>

<label for="login">Логин:</label><br>

<input type="text" id="login" name="login" required><br><br>

<label for="password">Пароль:</label><br>

<input type="password" id="password" name="password" required><br><br>

<button type="submit">Зарегистрироваться</button>

</form>

{% endblock %}