**Отчет о выполнении лабораторной работы №4**

Якухин Иван, 6303-010302D

*Идея: приложение с карточками для запоминания английских слов*

1. **Разработка пользовательского интерфейса**

С помощью программы Photoshop были нарисованы макеты отображения главной страницы, страницы колод, страницы выбора карточек, страницы обучения, карточки, страницы профиля.

Затем макеты были переведены в формат HTML/CSS для дальнейшей разработки frontend части приложения.

Пример:

<!DOCTYPE html>  
<html lang="en">  
 <head>  
 <meta charset="UTF-8" />  
 <meta name="viewport" content="width=device-width, initial-scale=1.0" />  
 <title>CardioGram</title>  
 <link rel="stylesheet" href="static/cards-style.css" />  
 </head>  
 <body style="background-image: url(static/card-pick-background2.png)">  
 <header class="header">  
 <div class="logo">CardioGram</div>  
 <nav class="nav">  
 <a href="#">Колоды</a>  
 <!-- <button class="login-button">Войти</button> -->  
 <button class="button">  
 <img src="static/profile.png" alt="" class="profile-img" />  
 </button>  
 </nav>  
 </header>  
 <div class="card-space">  
 <div class="card-back">  
 <div>  
 <p class="title">Forest</p>  
 <p class="translation">Лес</p>  
 </div>  
 <div class="image"></div>  
 <div>  
 <p class="desc">The forest is home to many animals</p>  
 <p class="translation">(Лес является домом для многих животных.)</p>  
 </div>  
 </div>  
 </div>  
 </body>  
</html>

На основе html фалов были написаны файлы react приложения добавляющие взаимодействие с сервером через API запросы.

import React, { useState, useEffect } from 'react';  
import { useParams } from 'react-router-dom';  
import Header from '../components/Header';  
import CardBack from '../components/CardBack';  
import LeftRightButtons from '../components/LeftRightButtons';  
import bg from '../assets/static/card-pick-background2.png';  
import axios from 'axios';  
import styles from '../assets/static/card.module.css';  
  
const Cards = () => {  
 const { deckId } = useParams();  
 const [cards, setCards] = useState([]);  
 const [currentIndex, setCurrentIndex] = useState(*0*);  
  
 useEffect(() => {  
 axios  
 .get(`http://127.0.0.1:8000/api/deck/${deckId}/`, { withCredentials: *true* })  
 .then((res) => {  
 setCards(res.data);  
 })  
 .catch((err) => {  
 console.error(err);  
 });  
 }, [deckId]);  
  
 const handleLeft = () => {  
 if (currentIndex !== -*1* && cards.length) {  
 axios  
 .post('http://127.0.0.1:8000/api/progress/', { card: cards[currentIndex].id }, { withCredentials: *true* })  
 .then(() => {})  
 .catch((err) => {  
 console.error(err);  
 });  
 }  
  
 if (currentIndex < cards.length - *1* && currentIndex !== -*1*) {  
 setCurrentIndex(currentIndex + *1*);  
 } else {  
 setCurrentIndex(-*1*);  
 }  
 };  
  
 const handleRight = () => {  
 if (currentIndex < cards.length - *1* && currentIndex !== -*1*) {  
 setCurrentIndex(currentIndex + *1*);  
 } else {  
 setCurrentIndex(-*1*);  
 }  
 };  
  
 return (  
 <div className={styles.cardPage} style={{ backgroundImage: `url(${bg})` }}>  
 <Header />  
 <LeftRightButtons leftFunc={handleLeft} rightFunc={handleRight} />  
 {currentIndex !== -*1* && cards.length ? (  
 <CardBack  
 title={cards[currentIndex]?.front\_text}  
 translation={cards[currentIndex]?.back\_text}  
 description={cards[currentIndex]?.example\_usage}  
 deckId = {cards[currentIndex]?.deck}  
 showBack = {*true*}  
 isSmall = {*false*}  
 />  
 ) : (  
 <p>Карточки в колоде закончились! Можете уже пойти поучиться)</p>  
 )}  
 </div>  
 );  
};  
  
export default Cards;

1. **Настройка взаимодействия с сервером**

В React приложении для взаимодействия с сервером была использована библиотека axios, позволяющая делать http запросы, с ее помощью страницы были наполнены динамическим контентом.

1. **Работа с аутентификацией**

Были реализованы формы для регистрации, входа в систему, смены пароля, редактирования личных данных. Все запросы посылаются на сервер с Cookie файлами, обеспечивающими работу JWT авторизация.

import React, { useState } from 'react';  
import { Link, useNavigate } from 'react-router-dom';  
import axios from 'axios';  
import '../assets/static/login.css';  
  
const Login = () => {  
 const [formData, setFormData] = useState({  
 username: '',  
 password: ''  
 });  
  
 const navigate = useNavigate();  
  
 const [error, setError] = useState('');  
 const [loading, setLoading] = useState(*false*);  
  
 const handleChange = (e) => {  
 setFormData({  
 ...formData,  
 [e.target.name]: e.target.value  
 });  
 };  
  
 const handleSubmit = (e) => {  
 e.preventDefault();  
 setLoading(*true*);  
 setError('');  
  
 axios  
 .post('http://127.0.0.1:8000/api/auth/login/', formData, {withCredentials: *true*})  
 .then((res) => { navigate('/'); })  
 .catch((err) => {  
 setError('Ошибка входа. Неверные логин или пароль.');  
 setLoading(*false*);  
 });  
 };  
  
 return (  
 <div className="login-page">  
 <h2>Вход</h2>  
 <form onSubmit={handleSubmit} className="login-form">  
 <input  
 type="text"  
 name="username"  
 placeholder="Логин"  
 value={formData.username}  
 onChange={handleChange}  
 required  
 className="login-input"  
 />  
 <input  
 type="password"  
 name="password"  
 placeholder="Пароль"  
 value={formData.password}  
 onChange={handleChange}  
 required  
 className="login-input"  
 />  
 {error && <p className="login-error">{error}</p>}  
 <button type="submit" disabled={loading} className="login-button">  
 {loading ? 'Загрузка...' : 'Войти'}  
 </button>  
 </form>  
  
 <p className="login-register-prompt">  
 Нет аккаунта?{' '}  
 <Link to="/signup">Зарегистрируйтесь</Link>  
 </p>  
 </div>  
 );  
};  
  
export default Login;