

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования
«Московский политехнический университет»

Кафедра «Инфокогнитивные технологии»
Образовательная программа «Веб-технологии»

Отчет по курсовому проекту
по дисциплине «Инженерное проектирование»

Тема: «Портфолио студентов»

Выполнил:

Студент группы 191-322

Цалапов А.М.

подпись, дата

Принял:

Старший преподаватель

Даньшина М.В.

подпись, дата

Оглавление

| | |
|--|----|
| Введение | 4 |
| Основная часть | 5 |
| Анализ конкурентов..... | 5 |
| i-exam..... | 5 |
| 4portfolio..... | 5 |
| Проектирование..... | 6 |
| Выбор функционала | 6 |
| Кейсы использования приложения | 6 |
| Этапы разработки приложения..... | 7 |
| Создание макета на Figma..... | 8 |
| CRUD в React | 8 |
| Вывод агрегирующей информации | 8 |
| 3 типа пользователей..... | 9 |
| Административная часть | 9 |
| Реализована в Django, в ней можно менять всю информацию и в React, в ней можно менять информацию о студентах..... | 9 |
| Используются фреймворки Django, React, Bootstrap | 9 |
| Интеграция по RestAPI..... | 9 |
| Фильтрация и поиск реализованы в Django | 9 |
| Ссылки: | 9 |
| Вход для обычного пользователя: логин user, пароль useruser | 9 |
| Вход для пероснала: логин staff, пароль staffstaff | 9 |
| Вход для администратора: логин admin, пароль adminadmin | 9 |
| Django:..... | 9 |
| Данные для входа те же, что и для реакта | 9 |
| RestAPI:..... | 9 |
| (Для просмотра необходимо войти в Django через администратора)..... | 9 |
| Код проходит валидацию на w3.org..... | 9 |
| Адаптация под 3 устройства | 10 |
| Технологии ручного тестирования проекта..... | 11 |
| SPA и GraphQL | 11 |
| Заключение..... | 11 |

Введение

Для инженерного проекта по программе «Веб-технологии» был выдан проект по теме «Продажа билетов на мероприятия». Это веб-приложение с базой данных из 10 таблиц, разработан с использованием технологий HTML, CSS, Django, MYSQL, фреймворка Bootstrap, Rest API, Postman. Сайты аналогичных организаций:

- <https://4portfolio.ru/student.html>
- <https://i-exam.ru/node/291>
- <https://portfolio.hse.ru/?filter=1,1,N,10,N,2020,N,1,N,N,N>
- https://www.canva.com/ru_ru/?irgwc=1&utm_medium=affiliate&utm_source=IndoLeads.&clickId=wbPUbV26GxyLWLt0M-0V7wD0UkEUnmXplSq0Tg0

Основная часть

Анализ конкурентов

i-exam

Ссылка - <https://i-exam.ru/node/291>

Удобный, интуитивно понятный интерфейс, подробная информация о студентах, есть фильтры по статусу, дисциплинам, датам и т.п., что облегчает работу с интерфейсом. Подробная статистика с разными типами диаграмм, с её помощью можно узнать рейтинг, группу и т.п.

4portfolio

Ссылка - <https://4portfolio.ru/student.html>

Плюсы:

- Кастомизация
- Современный интерфейс
- Соответствие требованиям ФГОС

Минусы

- Узкий функционал

Проектирование

Выбор функционала

Веб-приложение для онлайн просмотра портфолио тысяч студентов. В приложении будет 2 типа пользователей – обычный пользователь и администратор. У администратора есть права ко всем таблицам, доступ к админ панели, он имеет полный контроль над сайтом. Также администратор имеет возможность импортировать/экспортировать данные в таблицы, что позволяет ему делать резервные копии или же переносить таблицы в другие сервисы

Кейсы использования приложения

- Награждение лучших студентов по данным рейтинга.
- Работодатель может выбрать лучших студентов на стажировку
- Авторизация
- Посещение страницы
- Регистрация новых пользователей в админке
- Размещение студентов
- Размещение новых проектов
- Редактирование информации о проектах, студентах
- Импорт данных из таблиц
- Экспорт отредактированных данных

Этапы разработки приложения

Веб-приложение состоит из страницы с предложениями, на ней же размещен вход в учетную запись.

1. Анализ аналогичных веб-сервисов. Рассмотрение функционала, дизайна и пользовательских сценариев с перечисленных аналогов. Обнаружение отличий между этими сервисами.
2. Создание макета страницы на figma
3. Изучить документацию по React, RestAPI
4. Разработать SPA на реакте
5. Протестировать запросы через Postman
6. Прописать запросы в React, чтобы связать её с Django
7. Исправление ошибок, неполадок, выявленных в следствии тестирования получившегося веб-приложения. Размещение приложения на сервере fit.mospolitech.
8. Окончательное тестирование получившегося веб-приложения.
9. Разработка и документирование кейсов взаимодействия обычного пользователя или администратора с получившимся веб-приложением.
10. Написание Readme для репозитория, дозаполнение документации. Выгрузка окончательного набора документации и файлов веб-приложения в репозиторий

Создание макета на Figma

<https://www.figma.com/file/R8uhDSITBUyJ0kas0ZkaEp/AllStudents?node-id=0%3A1>

CRUD в React

Список студентов

Добавить студента

| # | Имя | Фамилия | Статус | Курс | | |
|----|-----------|-----------|----------|------|---------|----------|
| 27 | Александр | Цалапов | Учится | 2 | Удалить | Изменить |
| 35 | Александр | Цалапофф2 | Учится | 3 | Удалить | Изменить |
| 37 | Qoter | Аширов | Решается | 3 | Удалить | Изменить |
| 40 | NoName | Gas | Учится | 1 | Удалить | Изменить |
| 41 | Айрат | Аширов | Отчислен | 2 | Удалить | Изменить |
| 42 | Иван | Аширов | Учится | 3 | Удалить | Изменить |
| 43 | Дмитрий | Аширов | Решается | 1 | Удалить | Изменить |
| 44 | Виктор | Аширов | Учится | 4 | Удалить | Изменить |
| 45 | Ашир | Аширов | Отчислен | 4 | Удалить | Изменить |
| 46 | Александр | Цалапов | Зачислен | 4 | Удалить | Изменить |

Добавить студента

Имя:

Фамилия:

Статус обучения:
(учится/отчислен/выпустился и т.п.)

Учится

Курс:

1

Сохранить

Назад

Вывод агрегирующей информации

Список студентов

#27: Александр Цалапов
Учится на 2 курсе

#35: Александр Цалапофф2
Учится на 3 курсе

#37: Qoter Аширов
Решается об обучении 3 курсе

#40: NoName Gas
Учится на 1 курсе

#41: Айрат Аширов
Отчислен после 2 курсе

#42: Иван Аширов
Учится на 3 курсе

#43: Дмитрий Аширов
Решается об обучении 1 курсе

#44: Виктор Аширов
Учится на 4 курсе

#45: Ашир Аширов
Отчислен после 4 курсе

3 типа пользователей.

У всех пользователей могут быть следующие права: обычные, персонала и администратора. Пользователь с обычными правами может просматривать агрегирующую информацию. Пользователь со статусом персонала может просматривать новости для персонала. Пользователь со статусом администратора может редактировать информацию о студентах. Права пользователей могут сочетаться.

Административная часть

Реализована в Django, в ней можно менять всю информацию и в React, в ней можно менять информацию о студентах

Используются фреймворки Django, React, Bootstrap

Интеграция по RestAPI

Фильтрация и поиск реализованы в Django

Ссылки:

React: <http://react.std-962.ist.mospolytech.ru/Main>

Вход для обычного пользователя: логин user, пароль useruser

Вход для персонала: логин staff, пароль staffstaff

Вход для администратора: логин admin, пароль adminadmin

Django: <http://project-4-sem.std-962.ist.mospolytech.ru/admin/>

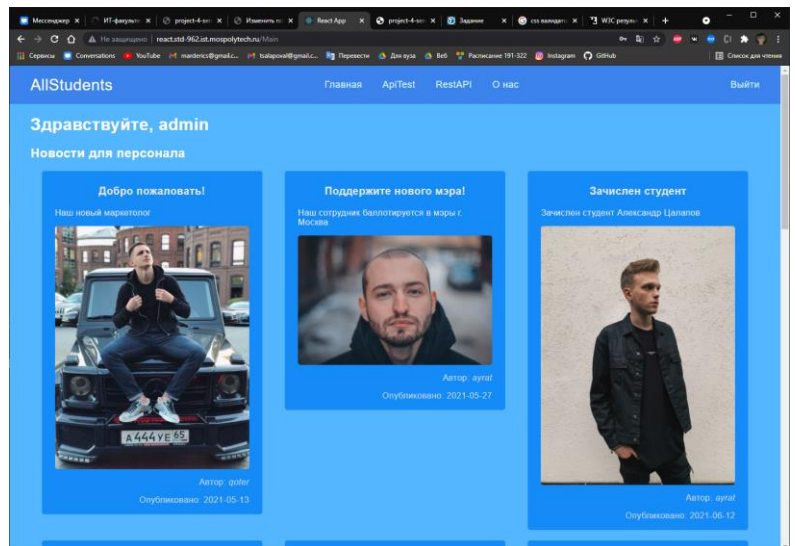
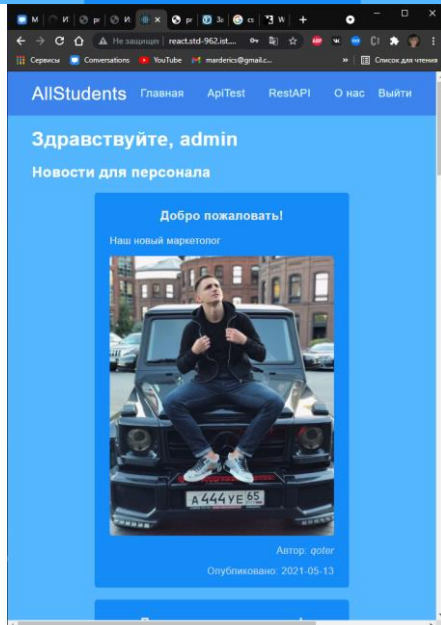
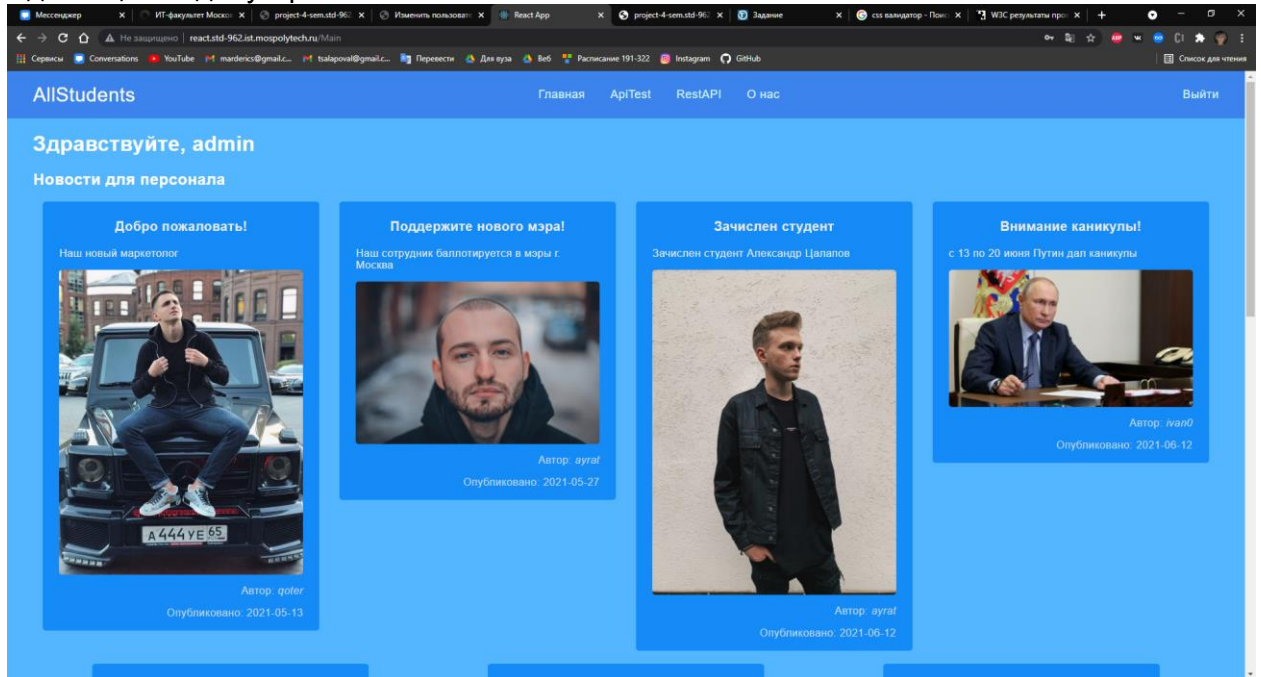
Данные для входа те же, что и для реакта

RestAPI: <http://project-4-sem.std-962.ist.mospolytech.ru/api/students/>

(Для просмотра необходимо войти в Django через администратора)

Код проходит валидацию на w3.org

Адаптация под 3 устройства



Технологии ручного тестирования проекта

Postman, через него я настраивал авторизацию и дальнейшие запросы CRUD, которые полсе тестирования реализовал в реакте.

SPA и GraphQL

SPA сделана на React, вместо GraphQL был использован RestFUL API

Заключение

В результате всей проделанной работы, получился готовый продукт, обладающий всеми нужными функциями для сосуществования и конкурирования со своими аналогами.

Сайт – <http://react.std-962.ist.mospolytech.ru/>

GITHUB репозиторий - <https://github.com/qoterwoter/project-4-sem>