Санкт-Петербургский государственный университет

Математическое обеспечение и администрирование информационных систем

Семенов Александр Сергеевич

Сервис курсовых HwProj Coursework

Отчёт по учебной практике

Научный руководитель:
Кандидат технических наук, доцент, Ю.В. Литвинов

Санкт-Петербург
2020

Оглавление

Введение

В современном мире информационные технологии занимают важную часть жизни общества, они помогают людям общаться, развлекаться, а также учиться. Сложно представить себе современного человека, не пользующегося ими.

Технологии помогают оптимизировать множество процессов как в бизнесе, так и в учебе. Одной из сфер учебной деятельности, которая несомненно нуждается в оптимизации, являются процессы, связанные с курсовыми работами. И в этой сфере, как и во многих других, есть множество моментов, которые могли бы быть лучше, если их оптимизировать при помощи современных технологий. Самой явной проблемой этой сферы является отсутствие централизованной системы, в которой преподаватели могут дистанционно информировать студентов о свободных темах курсовых работ, а студенты могли бы выбрать курсовую из множества свободных, предварительно ознакомившись со всем требованиями.

Процесс курсовых работ заключается в следующем: преподаватели размещают темы своих курсовых, а студенты могут их просматривать и подавать заявки со своим резюме на выполнение работы. Затем преподаватель просматривает заявки и одобряет или отклоняет их. Потом студент может прикрепить файл с отчетом, презентацию и добавить ссылку на свою работу. Также есть роль куратора курсовых работ, в чьи задачи входит установление дедлайнов, подтверждение защиты курсовой студентом и назначение рецензентов. Так как в СПбГУ отсутствует система поддержки работы с курсовыми, студенты, преподаватели и кураторы сталкиваются с некоторыми трудностями в процессе выполнения курсовых работ.

Таким образом, создание системы поддержки работ с курсовыми, в которой можно было бы выполнять все процессы, связанные с курсовыми работами как со стороны преподавателя, так и со стороны студента, является необходимым условием оптимизации этого вида деятельности.

1. Постановка задачи

Целью работы является создание системы курсовых работ, которая включает с себя поддержку действий со стороны преподавателя, студента и куратора курсовых работ. Помимо этого, система должна быть реализована так, чтобы в дальнейшем в неё можно было вносить улучшения, например, систему уведомлений. Проект является совместным и выполнялся группой студентов.

Задачи:

1. Разработать серверную часть приложения.
2. Реализовать серверную часть приложения и методы взаимодействия с клиентом.
3. Разработать прототип интерфейса.
4. Реализовать клиентскую часть приложения и наладить связь с сервером.

Эти задачи были распределены в группе, задачей данной работы были пункты 3 и 4, а именно работа с интерфейсом пользователя.

2. Обзор существующих решений

Аналоги:

1. Сайт easychair.org – система для организации конференций и организации подачи и рецензирования статей. Это хороший вариант решения, так как там есть всё необходимое для организации процесса курсовых работ, но у него есть определенные минусы, а именно англоязычный интерфейс, который может вызывать затруднения у людей, не достаточно хорошо владеющих английским языком. Также основная роль этого сайта – организация конференций, что не подходит для задач организации процесса курсовых работ.

2. Сайт bb.spbu.ru – сайт системы Blackboard, предназначенный для отслеживания своих курсов и оценок студентами, а также для выполнения тестовых работ. Этот сайт тоже не в полной мере подходит для организации процесса курсовых работ, так как он имеет достаточно специфический интерфейс, что может вызывать определенные трудности у студентов и преподавателей.

3. Обзор используемых технологий

Проанализировав аналогичные решения, были составлены требования к будущей системе. А именно: русскоязычный и понятный интерфейс, который не будет захламлен ненужными элементами, система должна быть интерактивной и быстро работать.

Сперва был выбор формата системы, а именно выбор между веб-сайтом и веб-приложением. Так как сервис курсовых работ предполагает интерактивное пользование, а также обработку вводимой пользователем информации, то выбор пал именно на веб-приложение, потому что для таких задач оно более подходит, чем статический веб-сайт.

Для более точного понимания интерфейса, а также упрощения его разработки, было принято решение сперва разработать прототип веб-приложения [3]. Выбор производился между онлайн-сервисом Figma, программой Adobe Photoshop и графическим редактором Sketch. Основные критерии выбора – бесплатность, возможность создания переиспользуемых компонентов, а также больше предпочтения отдавалось облачным решениям. В итоге был выбран онлайн-сервис разработки интерфейсов Figma, так как он бесплатный, в отличии от других предендентов, позволяет редактировать прототип с любого устройства, так как это облачный продукт и можно работать из браузера, и лёгок в презентации прототипа другим членам команды.

Язык программирования был выбран Typescript, так как это строго типизированный язык разработки веб-приложений, расширяющий возможности самого популярного языка веб-разработки Javascript. Выбор пал именно на язык статической типизации, так как из-за того, что проверка типов в нем происходит один раз – на этапе компиляции, то скорость его работы выше, а также повышается вероятность на этапе компиляции обнаружить ошибки и повышается скорость разработки, так как среда разработки сразу выделяет ошибки, связанные с несовместимостью типов [4].

Далее предстоял выбор фреймворка. Выбор осуществлялся между React и Angular, так как это самые популярные фреймворки, и они

поддерживают разработку на Typescript. При выборе рассматривались несколько пунктов: простота в изучении, быстрота работы и возможность создания легко тестируемого и многократно используемого кода. Окончательный выбор пал именно на React, так как из-за разнообразия различных структур Angular сложнее в изучении, к тому же React работает быстрее за счет реализации React Virtual DOM и различным оптимизациям рендеринга. Также преимуществом React перед Angular является поддержка концепции функционального программирования, что позволяет легко писать многократно переиспользуемый код [2].

Для запросов на сервер была выбрана библиотека Axios, так как она самая распространенная, и по сравнению со встроенной библиотекой Fetch она не требует дополнительного действия для получения данных из ответа. К тому же Fetch не отлавливает все ошибки, которые могут возникнуть, что может привести к ошибке всего приложения [1].

4. Описание решения

Веб-приложение работает по следующему принципу: сперва пользователь регистрируется или входит в свою учетную запись, где за каждым пользователем закреплена его роль: студент, преподаватель или куратор. Затем в зависимости от роли пользователя ему рендерится соответствующий интерфейс. Студент может просматривать свободные курсовые, а также свою текущую и завершенные курсовые. Также он может прикрепить нужные файлы к текущей курсовой, подать заявку на новую курсовую, а также изъявить желание быть рецензентом, чтобы оценивать другие работы.

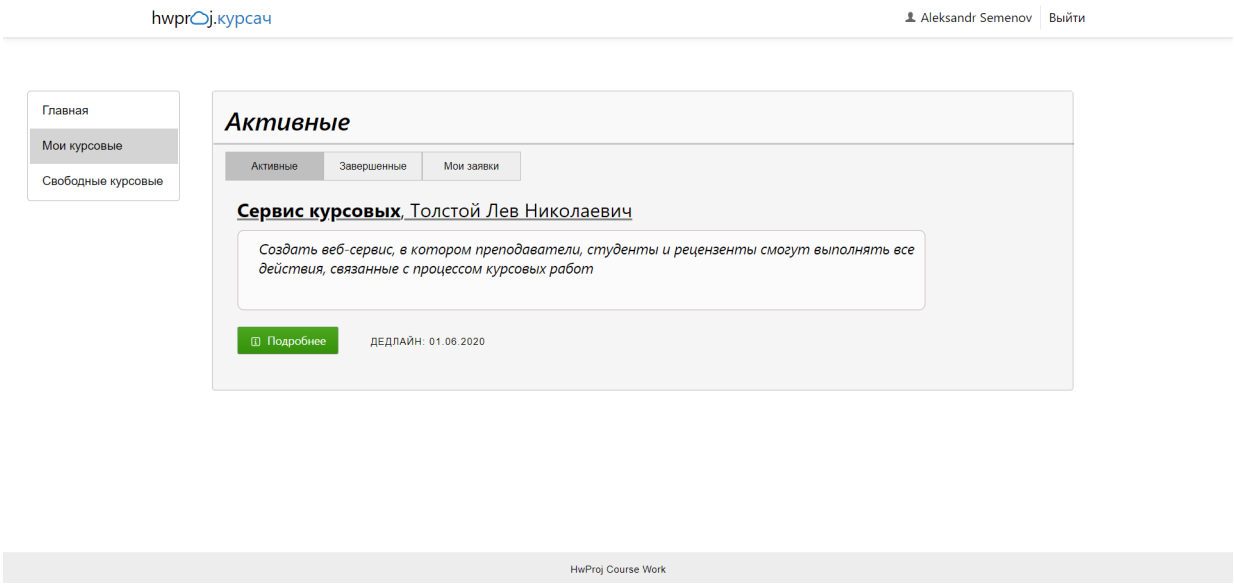

Рис. 1: Активная курсовая работа

Преподаватель может также просматривать свободные курсовые, добавлять новые темы, принимать и отклонять заявки на свои работы, предварительно просмотрев резюме студента. Также как и студент, преподаватель может быть рецензентом. Что касается куратора, то он может назначать дедлайны, выбирать, какие рецензенты могут рецензировать работы, которые относятся к этому куратору, а также он может согласиться или же изменить результаты биддинга – специального механизма назначения курсовой работе рецензента. Роль куратора немного сложнее остальных, так как помимо чисто кураторских дей-

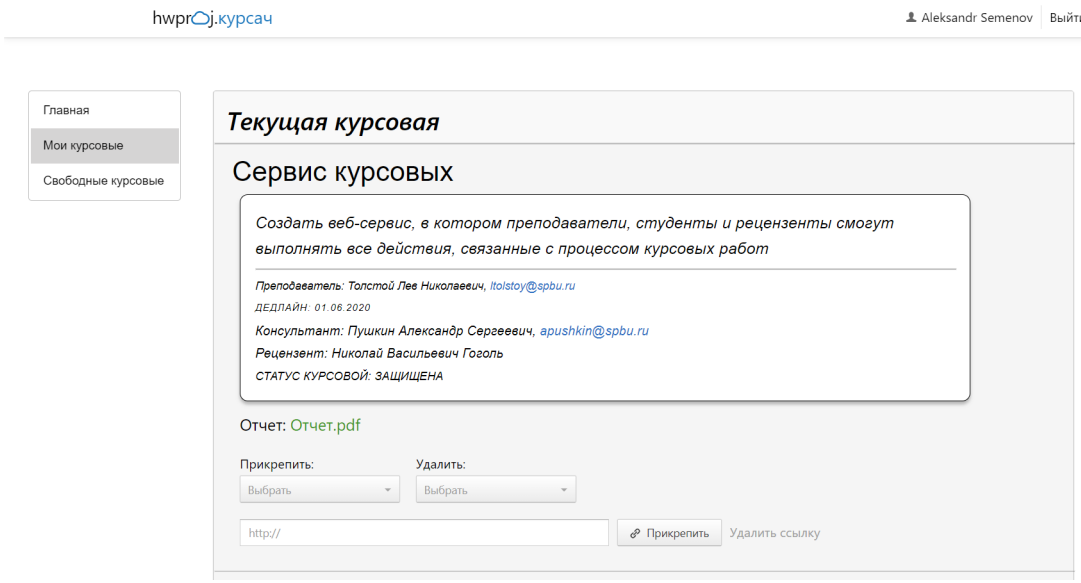

Рис. 2: Страница текущей курсовой

ствий, он может выступать в роли преподавателя с присущими ему функциями.

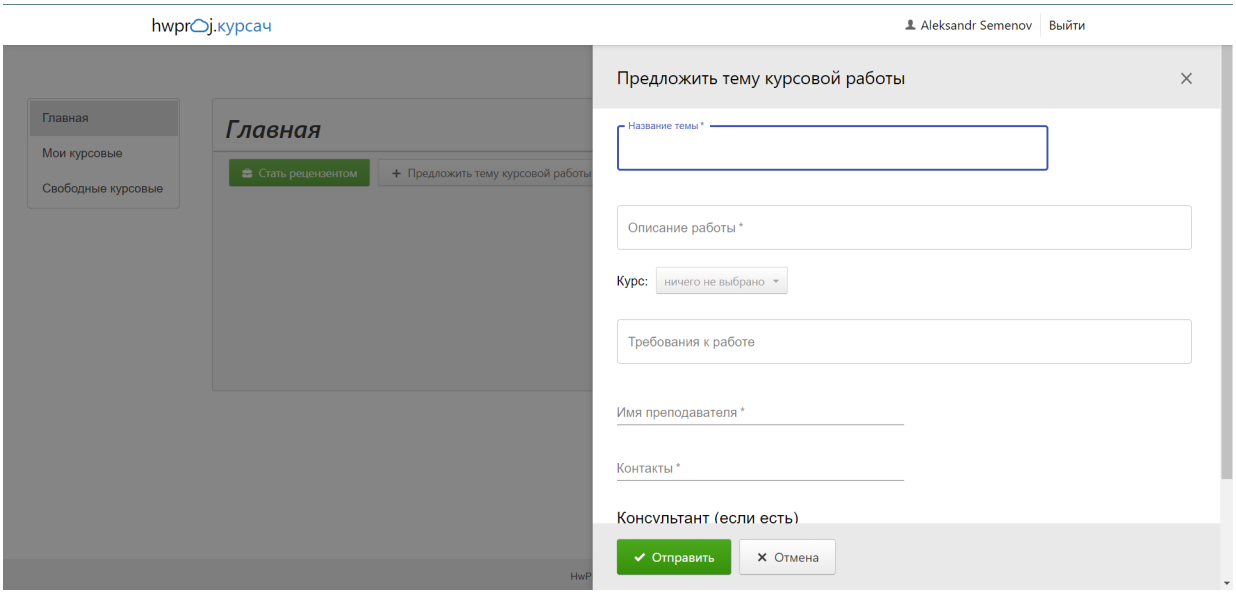

Рис. 3: Добавление новой курсовой преподавателем

Эта идея реализована при помощи так называемых компонентов, которые меняются при переходе на другую страницу приложения. В каждом компоненте есть метод render, который отвечает за вывод информации на экран. Функции, отвечающие за отрисовку компонента, возвращают JSX элемент – расширение синтаксиса Typescript, позво-

ляющее совместить его синтаксис с языком разметки. Благодаря этому в той части компонента, которая отвечает за внешний вид, можно писать код, который очень походит на HTML с применением CSS, что позволяет стилизовать компоненты. У приложения есть глобальная переменная page, которая меняется при изменении страницы, что и позволяет определить, в какой момент какой компонент выводить. Перед непосредственным рендером компонента, делается запрос на сервер при помощи библиотеки Axios, затем данные сохраняются во внутреннее состояние компонента, а потом выводятся на экран.

Когда пользователь регистрируется или входит в свою учетную запись, его данные посылаются на сервер, который возвращает JSON Web Token, затем он дешифруется, и из него извлекается вся нужная информация, например, имя и роль пользователя в системе. Эта технология используется для безопасной аутентификации пользователей.

Заключение

В результате совместной работы был разработан сервис курсовых работ, позволяющий студентам, преподавателям и кураторам курсовых работ централизованно совершать все действия, связанные с процессом курсовых работ.

Результаты:

1. Был разработан прототип интерфейса
2. Реализована клиентская часть приложения и налажена связь с сервером.

Исходный код: https://github.com/alechh/coursework.front

Список литературы

[1] Sayfan Gigi. Получение данных в вашем React приложении // Envato Tuts+, публикации. — 2018. — URL: https://code.tutsplus.com/ru/tutorials/fetching-data-in-your-react-application--cms-30670 (дата обращения: 26.05.2020).

[2] @yalatan Наталья. React или Angular или Vue.js — что выбрать? // Habrahabr, публикации. — 2019. — URL: https://habr.com/ru/post/476312/ (дата обращения: 26.05.2020).

[3] Батурин Андрей. Зачем надо делать прототип сайта // Evalution, серия статей про проектирование. — 2015. — URL: https://webevolution.ru/blog/sajti/zachem-nado-delat-prototip-sajta/ (дата обращения: 26.05.2020).

[4] Груздев Денис. Ликбез по типизации в языках программирования // Habrahabr, публикации. — 2012. — URL: https://habr.com/ru/post/161205/ (дата обращения: 26.05.2020).