

Студ. Д.Ю. Егоров, В.А. Баранов
Науч. рук. зав. кафедрой Д.М. Романенко
(кафедра информатики и веб-дизайна, БГТУ)

ОСОБЕННОСТИ РАЗРАБОТКИ ОНЛАЙН-ПЛАТФОРМЫ «Е-КАФЕДРА»

В настоящее время работники кафедр ВУЗов вынуждены решать множество различных задач, связанных с учебным процессом, функционированием кафедры и т. д. Так, например, в течении учебного года преподаватель должен составлять отчёты о выполнении учебной нагрузке. Выполнение данных действий вручную не несёт в себе высокой эффективности и отнимает большое количество времени. Также свою роль имеет человеческий фактор, что увеличивает вероятность появления ошибок, а также затраты времени на их исправление.

Решением выше описанных проблем является онлан-платформа «Е-Кафедра» – сервис для преподавательского состава кафедры ВУЗа, целью которого является автоматизация рутинных процессов, возложенных на сотрудников в виде учёта своей нагрузки, эффективного планирования и предоставление средства для удобного формирования отчёта о списанной нагрузке, введение электронного журнала для отслеживания успеваемости студентов и т.д. Отметим, что подобная платформа должна иметь модульную структуру и быть легко расширяемой. В качестве основной технологии для создания онлайн-платформы был выбран NodeJS и язык программирования TypeScript. NodeJS – программная платформа, основанная на движке V8 (транслирующем JavaScript в машинный код), превращающая JavaScript из узкоспециализированного языка в язык общего назначения.

Для реализация пользовательского интерфейса онлайн-платформы были использованы следующие средства: библиотека React, библиотека для управления состоянием Redux и библиотеки MaterialUI, содержащая готовые UI-компоненты, что позволяет ускорить разработку приложения [1]. Серверная часть также разработана с помощью фреймворка NestJS. NestJS – это тот фреймворк, созданный для облегчения жизни разработчика, использующий правильные архитектурные подходы [2].

Сервис предоставляет разделение пользователей на роли: пользователь, менеджер, администратор. *Пользователю* доступны такие возможности как регистрация, авторизация, заполнения расписания, управление своей учебной нагрузкой, генерация отчётов о списанной нагрузке. *Менеджер* имеет возможность управлять данными, используемые для взаимо-

действия с платформой (расписание звонков, дисциплины), активировать/деактивировать пользователя, управлять нагрузкой как своей, так и других пользователей, редактировать как своё, так и расписание других пользователей. Пользователь с ролью *Администратор* имеет вышеперечисленные возможности для ролей *Пользователь* и *Менеджер*, а также изменений ролей пользователя и верификация нагрузки, созданной другим преподавателем (рисунок 1).



Рисунок 1– Диаграмма вариантов использования

Важным в разрабатываемом модуле учета выполнения нагрузки является максимальная автоматизация процесса верификации списания нагрузки исходя из заполненного преподавателем расписания, хранящейся в БД плановой нагрузки, календаре рабочих и выходных дней и т. д., а также реализация подсистемы учета проведения занятий другим преподавателем, например, в виду болезни и т. д.

ЛИТЕРАТУРА

1. React. JavaScript-библиотека для создания пользовательских интерфейсов [Электронный ресурс] / Веб-сайт с документацией библиотеки – Калифорния, 2021. – Режим доступа: <https://ru.reactjs.org/> – Дата доступа: 25.03.2022.
2. Фреймворк для разработки масштабируемых веб-приложений [Электронный ресурс] / Веб-сайт с документацией фреймворка – Калифорния, 2017. – Режим доступа: <https://docs.nestjs.com/> – Дата доступа: 28.03.2022.