

Создание web-приложения-органайзера с интеграцией расписания занятий для обучающихся

А.В. Аникин, М.С. Митрофанова, Е.О. Стукалин, Е.В. Сибряев, А.В. Благоев

Самарский национальный исследовательский университет имени академика С.П. Королева, Самара, Россия

Обоснование. Планирование распорядка дня является для каждого обучающегося важной задачей. Согласно проведенному в Самарском университете опросу среди студентов, из 107 опрошенных 44 человека пользуются планерами, также существует потребность в функциональном приложении, совмещающем в себе функции планера и расписания, чтобы была возможность структурировать свое личное расписание, совмещать учебу и иные виды деятельности (95 % среди тех, кто пользуется существующими аналогами) [1].

В настоящий момент существует ряд подобных приложений, однако у них есть несколько минусов. К примеру, Google Calendar — в формате календаря, необходимо вручную вносить данные; Studify — иностранное, нельзя использовать в России; «Расписание» — с сентября 2022 года больше не работает; telegram-bot (созданный студентами Самарского университета) — выдает расписание на один конкретный день, имеет неудобный интерфейс и не имеет возможности редактирования.

Цель — создать web-приложение-органайзер с интеграцией расписания занятий для обучающихся.

Методы. Для разработки были изучены следующие технологии: HTML и CSS для создания браузерных страниц [2]; Node.js для написания серверной части [3]; React для визуальной части [4, 5], для упрощения работы с технологией PWA; MySQL для создания базы данных; технология PWA для трансформации браузерной страницы в функционирующее на телефоне приложение. Отдельное внимание было уделено изучению PWA технологии, также были выделены следующие ее преимущества: более быстрая работа по сравнению с сайтом, отсутствует необходимость скачивания из магазинов приложений, маленький вес, работа без интернета.

Результаты. Предложен алгоритм считывания данных с сайта, осуществляющий поиск по структуре html страницы и объединяющий полученную информацию для последующего ее хранения в базе данных приложения. Схема работы алгоритма представлена на рис. 1.

Ссылка на ресурс для считывания и обновления информации находится в базе данных. Первое считывание происходит в момент первого запроса от пользователя к определенной группе, после чего расписание загружается в базу данных и при повторном обращении оттуда загружается. Обновление данных происходит в момент, когда пользователь, который выбрал это расписание занятий, заходит в приложение.

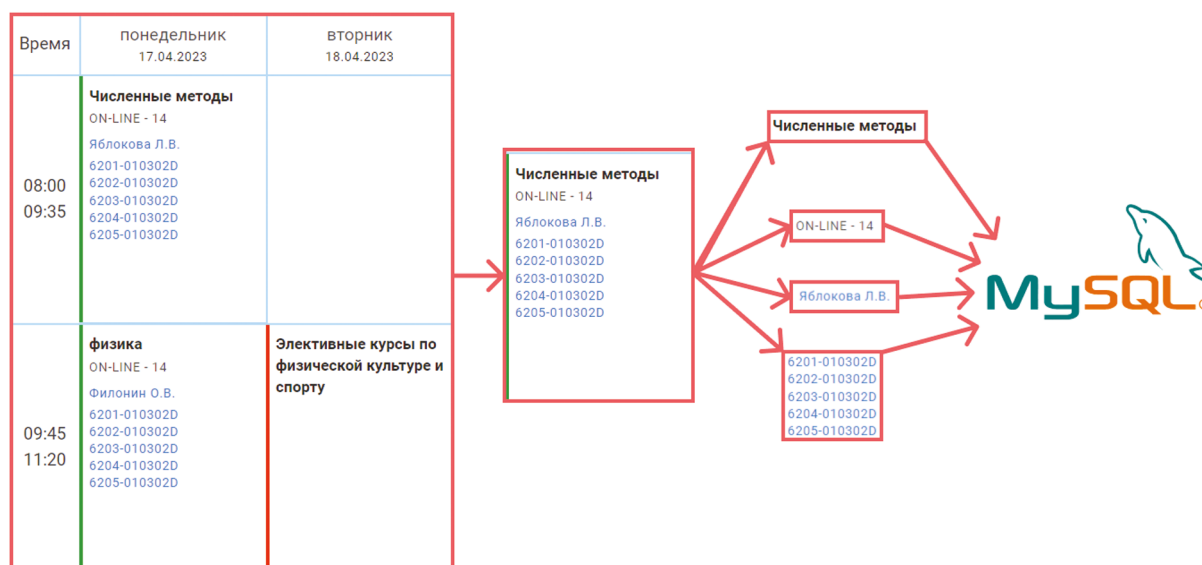


Рис. 1. Визуальная схема алгоритма парсинга страницы расписания

Авторами был создан прототип приложения в программе Figma, полностью отображающий функционал, такой как авторизация пользователей; создание личного кабинета; возможность добавления, редактирования, удаления задач; отображение задач на неделю; просмотр календаря с отображаемыми задачами. С помощью возможностей программы в прототипе реализованы переходы по ссылкам, нажатия на кнопки, скроллинг страницы, открытие меню, редактирование задач, профиля пользователя.

Также был продуман дизайн и выбраны специальные цвета, корректно отображаемые на всех видах устройств. Был проведен анализ различных шрифтов, выбраны самые легко читаемые, быстро воспринимаемые и подходящие по стилю.

Создана страница задач пользователя с использованием JavaScript-библиотеки React и серверной частью с использованием платформы Node.js. На странице реализовано отображение задач в соответствии с ID пользователя.

Выводы. Удобные приложения-органайзеры пользуются популярностью среди студентов. Интеграция подобных приложений с расписанием занятий — актуальная для студентов проблема. Используя созданный нами алгоритм парсинга и прототип приложения в дальнейшем мы планируем создать приложение, которое сможет предоставить для студентов необходимый функционал.

Ключевые слова: приложения-органайзеры; технология PWA; технология парсинга; библиотека React; веб-программирование.

Список литературы

1. docs.google.com [Электронный ресурс]. Результаты опроса [дата обращения: 08.05.23]. Доступ по: https://docs.google.com/forms/d/1CRdD5gFLI8SFemcyY4I2N7_wFkqHzclm1w_u3bWEDq0/edit#responses
2. Duckett J. HTML and CSS: Design and build websites. John Wiley and Sons, 2011. 490 p.
3. Herron D. Node.js web development: Server-side web development made easy with Node 14 using practical examples, 5th Edit. Packt Publishing, 2020. 762 p.
4. Wieruch R. The road to react: Your journey to master plain yet pragmatic React.js. Independently published, 2018. 292 p.
5. Banks A., Porcello E. Learning react: Functional Web Development with react and redux. O'Reilly Media, 2017. 350 p.

Сведения об авторах:

Арсений Васильевич Аникин — студент, группа 6205-010302D, институт информатики и кибернетики; Самарский национальный исследовательский университет имени академика С.П. Королева, Самара, Россия. E-mail: ars.anikin.2003@gmail.com

Мария Сергеевна Митрофанова — студентка, группа 6205-010302D, институт информатики и кибернетики; Самарский национальный исследовательский университет имени академика С.П. Королева, Самара, Россия. E-mail: mashamitr6556@gmail.com

Егор Олегович Стукалкин — студент, группа 6205-010302D, институт информатики и кибернетики; Самарский национальный исследовательский университет имени академика С.П. Королева, Самара, Россия. E-mail: egorstukalkin@yandex.ru

Егор Витальевич Сибряев — студент, группа 6205-010302D, институт информатики и кибернетики; Самарский национальный исследовательский университет имени академика С.П. Королева, Самара, Россия. E-mail: yegor.sibryayev@mail.com

Александр Владимирович Благов — научный руководитель авторов, кандидат технических наук, доцент; Самарский национальный исследовательский университет имени академика С.П. Королева, Самара, Россия. E-mail: blagov@ssau.ru