Работа и структура приложения основана на базе данных MS SQL Server. Интерфейс серверной части представлен в виде инструмента Swagger, где расположены все маршруты сервера.

В результате работы над приложением была спроектирована база данных, состоящая из 14 таблиц, и разработана серверная часть для благотворительной онлайн платформы с внедрением библиотеки React и Redux.

Литература

1. Прайс, М. Дж. C# 10 и .NET 6. Современная кросс-платформенная разработка / М. Дж. Прайс. – СПб : ООО «Питер Мейл», 2023. – 848 с.

Е. М. Морозова
(ГГУ имени Ф. Скорины, Гомель)
Науч. рук. **А. В. Семченко**, канд. физ.-мат. наук, доцент

РАЗРАБОТКА WEB-ПРИЛОЖЕНИЯ ДЛЯ ПЛАНИРОВАНИЯ ЗАДАЧ И ЦЕЛЕЙ

Человек, живя в современном мире не может держать все под своим контролем, и поэтому важно уметь планировать свой день и правильно расставлять приоритеты. Чтобы не забывать о предстоящих событиях, повседневных делах, работе, учебе и успевать все и всегда существует множество различных приложений и каждый выбирают, что ему подходит.

Разрабатываемое приложение позволяет составлять списки задач, расставлять приоритеты для каждой задачи, устанавливать время для выполнения выбранной задачи.

Задача – это составляющее цели. Например, в жизни нас интересует цель «устроиться на работу», и в некоторых случаях это не так просто сделать. Приложение позволяет нам поставить цели и расписать какие задачи нужно выполнить, чтобы достичь желаемого.

Однако еще одной проблемой может быть отсутствие мотивации. Для этого в приложении можно самому себе назначать награды, тем самым, мотивируя себя выполнять свои задачи для достижения цели.

Для создания макета сайта был выбран графический редактор Figma. Дизайн части страницы со списком целей и соответствующих задач и наград представлен на рисунке 1.

Архитектурно приложение разбито на две части: клиентскую (frontend) и серверную (backend).

В основу клиентской части лег фреймворк React.js, отвечающий за создание пользовательских интерфейсов. Также используются следующие модули:

– Redux Toolkit предоставляет инструменты для настройки глобального хранилища;

– Axios – клиентская библиотека HTTP, которая позволяет отправлять запросы к заданной конечной точке;

– React Router – это библиотека маршрутизации в React, которая дает возможность беспрепятственно перемещаться с одной страницы на другую без необходимости перезагружать браузер в одностраничном приложении;

– Date Time Picker. Этот элемент используется для выбора даты и времени, а также их отображение в выбранном формате;

– Sass – это язык для описания другого языка, который упрощает и ускоряет написание CSS-кода;

– CSS-модуль – это достаточно популярная система для модульности и компоновки CSS.

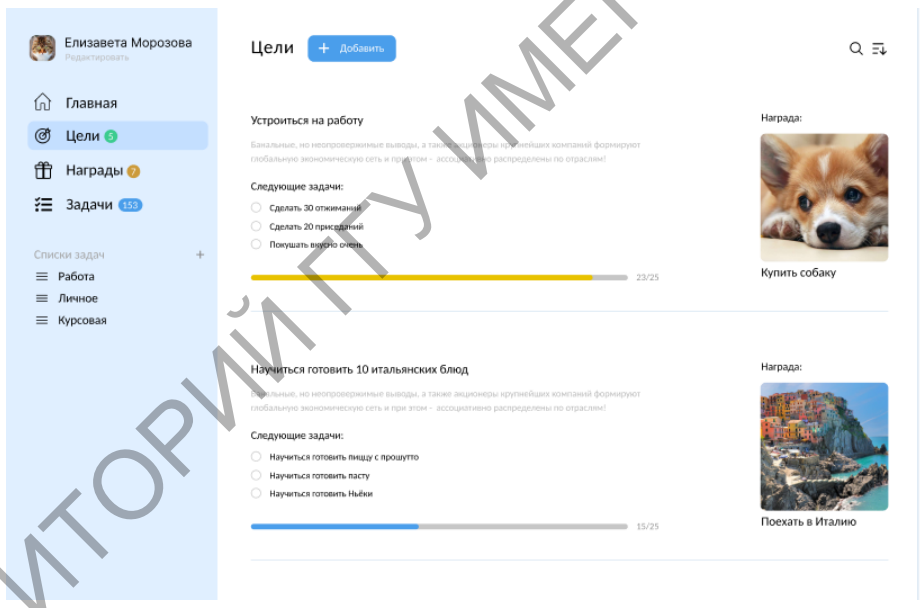

Рисунок 1 – Дизайн части страницы со списком целей и соответствующих задач и наград

Серверная часть приложение реализована на фреймворке Express.js, который позволяет нам удобно описывать маршруты, а также обеспечивать безопасность. Для удобства разработки выбран шаблон MVC.

Базой данных приложения является SQLite. Sqlite – это достаточно быстрая встраиваемая СУБД, которая позволяет хранить данные на одном устройстве.

Для удобства взаимодействия с базой данных и использование механизма ORM была выбрана библиотека sequelize, которая позволяет нам описывать модель данных прямо в исходном коде, а также делать автоматические миграции изменений.

Сборка приложения производится при помощи инструмента сборки vite.

П. С. Нагорский, Е. В. Рафалова
(ГГУ имени Ф. Скорины, Гомель)
Науч. рук. **В. Н. Леванцов**, ст. преподаватель

ДИНАМИЧЕСКОЕ ОБНОВЛЕНИЕ КОНТЕНТА ВЕБ-ПРИЛОЖЕНИЯ

Рассматриваемая проблема возникла в ходе создания клиент-серверного приложения «Карта достопримечательностей Беларуси». Суть приложения в том, что пользователь веб-сайта видит перед собой интерактивную карту Беларуси с нанесенными на неё маркерами достопримечательностей и возможностью кликнув на маркер – открыть соответствующую статью с подробным описанием места. Авторизированный гид или администратор веб-сайта может создавать новые статьи, однако у пользователей, которые в момент создания новой статьи уже просматривают веб-сайт, не отображается на карте новая локация, для обновления контента необходимо перезагрузить страницу. Этот нюанс может быть достаточно весомой проблемой при условии большой аудитории пользователей и высокой активности гидов или администраторов.

Основным протоколом передачи данных между классическим REST-API сервером и клиентским приложением, написанным на языке Javascript -является протокол HTTP/HTTPS, данные передаются в JSON формате. Данный протокол работает по схеме запрос-ответ. То есть, когда пользователь посещает веб-сайт – клиентский код посылает на сервер HTTP GET запрос и в ответ получает все данные о достопримечательностях, необходимые для их последующей визуализации пользователю. Аналогичным образом, при заполнении гидом на веб-сайте формы информацией о новой достопримечательности – на сервер посылается HTTP POST запрос с этой информацией, сервер проводит