

Функционально приложение решает следующие задачи: реализация ролей пользователя и администратора; обеспечение поиска и фильтрации мероприятий по виду досуга; обеспечение возможности регистрации и авторизации; обеспечение пользователя возможностью создания, редактирования и удаления заявок на мероприятие; обеспечение пользователя возможностью откликнуться на мероприятие другого пользователя и добавление мероприятия в Избранное; обеспечение пользователя возможностью просматривать список откликнувшихся пользователей, а также просматривать список заявок, на которые он сам откликнулся; обеспечение пользователя возможностью принятия отклика на его мероприятие и возможностью переписки с участниками мероприятия; обеспечение администратора возможностью блокировать профили пользователей, которые публикуют нежелательный контент, а также активировать ранее заблокированные профили.

В данном проекте использовались следующие технологии:

- MySQL в качестве базы данных приложения;
- Node.js для разработки сервера приложения;
- пакет sequelize для связи базы данных с сервером приложения;
- React.js и MaterialUI для разработки пользовательского интерфейса;
- пакет Axios для отправки запросов на сервер;
- Socket.io для реализации обмена сообщениями в режиме реального времени.

УДК 004.415.2

Студ. А.Э. Шиманель
Науч. рук. зав. кафедрой Н.В. Пацей
(кафедра программной инженерии, БГТУ)

ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ОНЛАЙН КОНСУЛЬТАЦИИ С ВРАЧАМИ

По данным доклада Всемирного банка и ВОЗ, 50% населения планеты не имеет доступа к медицинским услугам [1-2]. Современные технологии могут кардинально изменить здравоохранение и предоставить всем пациентам доступ к помощи квалифицированных врачей. Разработка медицинских приложений сейчас актуальна и востребована. Целью данного проекта было создание веб-приложения для проведения онлайн консультаций с врачами.

Разработанный проект решает такие задачи, как возможность пользователям общаться между собой, обмениваться изображениями,

просматривать список публикаций врачей, выполнять подписку на определенных специалистов, находить врачей по различным параметрам, выполнять видеозвонки (рисунок 1).

The screenshot shows the 'HealthApp' web interface. On the left is a dark sidebar with navigation links: 'Мой профиль', 'Лента новостей', 'Пользователи', 'Найти врача', and 'Сообщения'. The main content area features a user profile for 'Апалония Греческая' with a profile picture and a bio section containing the text 'test' and a 'Редактировать' button. To the right is a health questionnaire titled 'О своем состоянии здоровья сообщайте следующее:'. The questionnaire consists of multiple sections with radio button options for 'Да' (Yes) and 'Нет' (No):

- Заболела ли вы:**
 - Заболели ли сердца: Да Нет
 - Заболели ли легких: Да Нет
 - Заболели ли глаз: Да Нет
 - Заболели ли почки: Да Нет
 - Заболели ли кожи: Да Нет
 - Заболели ли пор органов: Да Нет
 - Инфекционные заболевания: Да Нет
 - Заболели ли кишечника: Да Нет
- Аллергические реакции:**
 - На антибиотики: Да Нет
 - На гормоны: Да Нет
 - На пыльцу и растения: Да Нет
 - На пищевые продукты: Да Нет
 - На шерсть животных: Да Нет
 - На препараты с йодом: Да Нет
- Также сообщите, что ранее было:**
 - Травмы: Да Нет
 - Операции: Да Нет
 - Переливания крови: Да Нет
 - Сотрясения мозга: Да Нет
 - Химио или лучевая терапия: Да Нет
 - Курение: Да Нет
- Для пациентов:**
 - Беременны ли вы?: Да Нет
 - Являетесь ли кормящей матерью?: Да Нет
 - Принимаете ли противозачаточные препараты?: Да Нет

At the bottom of the questionnaire, there are buttons for 'Отменить' (Cancel) and 'Сохранить' (Save), along with a note: 'Данная информация будет видна врачам' (This information will be visible to doctors) with a 'Да' (Yes) radio button selected.

Рисунок 1 – Веб-приложение онлайн консультаций

Серверная часть приложения реализована на платформе NodeJS с использованием языка TypeScript и фреймворка Express. В качестве хранилища данных была выбрана MongoDB, доступ к которой осуществляется с помощью ODM библиотеки Mongoose. Для обмена данными в режиме реального времени была выбрана библиотека Socket.io имеющая интерфейсы для клиентской и серверной части приложения. Для организации видеозвонков использован протокол WebRTC. Клиентская часть приложения разработана в виде одностраничного приложения, с использованием библиотеки ReactJS.

ЛИТЕРАТУРА

1. GlobalWebIndex [Электронный ресурс] – Режим доступа: <https://www.gwi.com/coronavirus> – Дата доступа: 13.04.2022.
2. ВОЗ [Электронный ресурс] – Режим доступа: <https://www.who.int/news/item/13-12-2017-world-bank-and-who-half-the-world-lacks-access-to-essential-health-services-100-million-still-pushed-into-extreme-poverty-because-of-health-expenses> – Дата доступа: 13.04.2022.