**Министерство науки и высшего образования Российской Федерации**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**(Университет ИТМО)**

Факультет **Прикладной информатики**

Направление подготовки **09.03.03 Прикладная информатика**

Образовательная программа **Мобильные и сетевые технологии**

**К У Р С О В О Й   П Р О Е К Т**

Тема: «Разработка веб-приложения для обмена видео-потоками (ВКС)»

Обучающийся: Янин Егор Вячеславович K3141

Санкт-Петербург 2024

СОДЕРЖАНИЕ

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ3

ВВЕДЕНИЕ4

1 Приложения для видеоконференций6

2 Стек технологий и разработка8

3 Результаты работы11

ЗАКЛЮЧЕНИЕ13

СПИСОК ИСПОЛЬЗОВАННЫХ МАТЕРИАЛОВ15

ПРИЛОЖЕНИЕ16

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

Frontend (фронтенд) – клиентская часть продукта.

Backend (бэкенд) – серверная часть продукта.

Fullstack (фуллстек-разработчик) – разработчик, который отвечает как за клиентскую, так и за серверную часть.

WebRTC – браузерная технология, предназначенная для передачи потоковых данных между браузерами или приложениями с использованием технологии двухточечной передачи.

Javascript – мультипарадигменный язык программирования.

Typescrript – язык программирования, средство разработки веб-приложений на Javascript.

React – библиотека с открытым исходным кодом JavaScript. Служит для создания пользовательского интерфейса из маленьких изолированных компонентов.

Vite – инструмент для сборки веб-приложений.

Express – простой и минималистичный web-framework для NodeJS. Служит для создания веб приложений, а также для создания API.

NodeJS – среда выполнения JavaScript построенная на движке Chrome V8. Служит для работы с серверной частью. С помощью NodeJS можно писать полноценные веб приложения или использовать его как веб-сервер.

PostgreSQL – свободная объектно-реляционная система управления базами данных.

Эндпоинт – это конечная точка веб-сервиса, к которой клиентское приложение обращается для выполнения определённых операций или получения данных.

ВВЕДЕНИЕ

Десятки миллионов людей в нашей стране испытывают потребность в общении со своими родственниками, друзьями, знакомыми, которые зачастую находятся в других городах, странах и даже на других континентах. Для обеспечения комфортного и безопасного соединения они используют огромное количество иностранных приложений, предоставляющих возможность осуществлять коммуникации посредством переписки или звонков, например, Zoom, Google Meets, WhatsApp, Skype. Однако в условиях нестабильной экономической и политической ситуации доступ пользователей к таким программам по тем или иным причинам может быть заблокирован, в связи с чем возникает острая необходимость в изобретении и реализации отечественного программного обеспечении, которое составит достойную конкуренцию зарубежному и будет способно заменить его на внутрироссийском рынке. На актуальность темы курсового проекта указывают такие факторы, как потенциальный уход с российского рынка иностранных компаний по разработке ПО для создания видеоконференций, отсутствие на российском рынке богатого разнообразия приложений для реализации возможности создания видеоконференций, а также необходимость поддержания главного принципа рыночной экономики и, как следствие, комфортного образа жизни людей - конкуренции.

Цель курсовой работы - создать приложение, предоставляющее возможность пользователям обмениваться видеопотоками с использованием технологии WebRTC.

В качестве основных задач в ходе работы над данным проектом являются создание клиентской части приложения, которое можно разложить на реализацию сервисной логики фронтального приложения и его инициализацию, реализацию логики обмена видео-потоками во фронтальном приложении, создание сервисной части приложения, подразумевающее реализацию сигнального, серверного приложений, а также создание базы данных для записи логинов и прочих данных пользователей, создание дизайна пользовательских сценариев, посредством которых пользователи смогут видеть результат их взаимодействия с программой, а также написание автоматических тестов для проверки работоспособности приложения.

1 Приложения для видеоконференций

Наша команда решила создать веб-приложение, которое позволит пользователю делиться изображением с видеокамеры и звуком микрофона, а также слышать и видеть других пользователей. Пользователи смогут создавать свои конференции и присоединяться к другим. Чтобы различать друг друга, пользователи смогут авторизироваться и установить себе никнейм.

Конечно, на рынке уже существует множество подобных продуктов. Из популярных приложений для видеоконференций можно выделить Zoom, Discord, Яндекс Телемост. Потребители у этих приложений отличаются: Zoom используется для корпоративной связи и образовательных целей во многих странах мира, Discord используют геймеры, а Яндекс Телемост – пользователи из России. Однако пользовательские данные так или иначе проходят через сервера компаний – зарубежных или отечественных. Из-за этого данные пользователей могут быть украдены при взломе серверов или с целью их продажи маркетологам для таргетированной рекламы.

Решить проблему с конфиденциальностью данных пользователей можно, если их данные будут проходить не через сервер, а напрямую от пользователя к пользователю. Такой способ подключения называется peer-to-peer [1]. Чтобы установить соединение между устройствами, они должны «найти» друг друга. Для этого необходим сигнальный сервер. Благодаря прямому подключению передача данных происходит быстро и эффективно. Технология WebRTC не использует сторонние сервисы, где можно было бы реализовать защиту данных. Однако она работает через проверенные сетевые протоколы, которые обеспечивают безопасность передачи данных.

Тем не менее, данный подход не лишён недостатков. С увеличением числа участников конференции растёт количество связей между ними. Если n – число участников конференции, то на каждого пользователя приходится n-1 соединений. Следовательно, увеличивается нагрузка на устройство. Это можно исправить, подключившись к сторонним серверам. Если на WebRTC разрабатывается крупный проект, в котором будет установлено соединение между сотнями пользователей, то придется использовать дополнительный сервер, чтобы уменьшить нагрузку на браузер. Также возникают проблемы из-за разных браузеров, интернет-провайдеров, скорости соединения и местоположения у пользователей. Из-за этого проще организовать один сервер для конференции. Несмотря на это, мы решили воспользоваться технологией WebRTC с прямым подключением между пользователями, так как наш проект небольшой и не предназначен для крупных видеоконференций на сотни участников.

2 Стек технологий и разработка

Наше веб-приложение представляет собой «single page application» − одностраничное приложение [2]. Это значит, что страница не перезагружается при переходе между разделами. В SPA-приложении все данные загружаются один раз, когда открывается приложение. Это делает приложение более быстрым и удобным для использования. Особенность SPA − использование JavaScript для формирования страницы у клиента. От сервера клиенту один раз приходит полный макет сайта, а дальше данные обновляются при запросах. Это значит, что большая часть логики и интерфейса находится на стороне пользователя, а сервер предоставляет только необходимые данные.

Преимущества SPA-приложений:

− высокая скорость загрузки,

– снижение нагрузки на сервер,

– удобство использования,

– простота разработки,

– совместимость, адаптивность.

Для разработки SPA-приложения нужен некоторый стек технологий. Мы использовали базу данных PostgreSQL, среду выполнения NodeJS, библиотеку ReactJS и фреймворк Express для NodeJS. Весь стек реализуется на Javascript, прост в использовании и позволяет создать полноценное веб-приложение. Мы выбрали именно этот стек, так как он прост в освоении и широко распространён в веб-разработке [3].

Для того чтобы эффективно выполнить работу, мы распределили обязанности. Было выделено 5 ролей: фронтенд-разработчик на React, бэкенд-разработчик (Express и PostgreSQL) для работы с серверной частью приложения, фуллстек разработчик, дизайнер на Figma и тестировщик на Cypress. Результат работы команды зависит от всех её членов и необходимо кооперироваться, чтобы разработка шла планомерно. Так фронтенд получает интерфейс от дизайнера, взаимодействует с бэкендом, чтобы создать работающие эндпоинты [4], и результаты работы уходят к тестировщику.

В результате распределения ролей и первой онлайн-встречи нашей команды передо мной была поставлена задача реализовать пользовательский интерфейс фронтального приложения, разработанный дизайнером проекта. Я отвечал за реализацию страниц авторизации, создания и подключения к конференции. Для этого я воспользовался такими инструментами как Typescript, React, Vite. Выбор этих технологий объясняется их популярностью и доступностью, а также тем, что руководитель проекта – фронтенд разработчик, знаком с ними. Его опыт помог нам во время реализации проекта.

Так как моя задача – сверстать страницы веб-приложения, в основном я работал с библиотекой React. Она позволяет создавать компоненты – отдельные небольшие куски кода, из которых затем составляется сложный пользовательский интерфейс [5]. Благодаря этому можно использовать один компонент, например кнопку, в нескольких местах. Кроме этого, я ознакомился с библиотекой для создания форм React Hook Form и использовал её при реализации страницы авторизации. Библиотека React Router Dom помогла настроить маршрутизацию страниц приложения.

Так как до проекта у меня не было опыта в разработке веб-приложений, необходимо было получить соответствующие знания. Чтобы узнать основы, я посмотрел обучающие видео, в том числе те, которые записал руководитель проекта. Они дали мне теоретический минимум, который был необходим для дальнейшей работы. Я освоил основы веб-разработки, общие подходы и принципы, которые используются в ней. Также я установил необходимое программное обеспечение и попрактиковался в вёрстке с помощью обучающих видео.

В это время мой коллега Александр создавал дизайн веб-приложения. Он представил первый вариант дизайна, и на онлайн-встрече мы обсудили уго работу, высказали свое мнение. Проблематично было согласовывать время наших встреч из-за разного расписания у участников проекта и занятости руководителя на работе. Также было непросто совмещать выполнение работы по проекту с другими заданиями и лабораторными работами. Из-за этого на создание интерфейса, правки и другие вопросы ушло много времени, так что у меня была неделя на то, чтобы выполнить работу.

Когда дизайн был готов, я приступил к вёрстке страниц авторизации и подключения к конференции. Трудности вызвала реализация многочисленных сценариев, на которые интерфейс должен выдать ответ, например предупреждения о неправильном логине или пароле. Необходимо было прописать все вариации, которые рассмотрел Александр. Также было тяжело создать работающие эндпоинты, так как постоянно возникали проблемы с запросами: они не доходили до сервера. Со всем этим я столкнулся впервые, но мы смогли определить, в чём была проблема, и решили её.

3 Результат работы

Я смог реализовать интерфейс, созданный моим коллегой, а именно страницу авторизации, создания конференции и подключения к ней (см. рис 1 и 2).

Изображение выглядит как текст, снимок экрана, розовый, дизайн

Автоматически созданное описание

Рисунок 1 – страница авторизации

Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание

Рисунок 2 – страница подключения к конференции

На странице авторизации пользователь может войти в аккаунт или создать новый. После авторизации пользователь попадает на главную страницу, где он может создать новую конференцию или подключиться к существующей. Пользовательский интерфейс отзывчивый − пользователь может увидеть последствия своих действий в виде предупреждений, изменения цвета элементов.

При попытке авторизации на сервер отправляется HTTP-запрос, если ответ положительный, происходит переадресация на главную страницу, иначе выводится сообщен2ие об ошибке. Если пользователь уже авторизован, он сразу же перенаправляется на главную страницу.

На главной странице пользователь может создать конференцию, и тогда он направится на страницу конференции, либо может подключиться по id, который можно получить на странице конференции.

Интерфейс адаптивный – с разных устройств веб-приложение будет работать корректно (см. рис 3).

Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание

Рисунок 3 – пользовательский интерфейс на мобильном устройстве

ЗАКЛЮЧЕНИЕ

Цель нашего проекта – создать приложение, предоставляющее возможность пользователям обмениваться видеопотоками с использованием технологии WebRTC – была достигнута. Мы успешно реализовали приложение, протестировали и запустили его. Все задачи были выполнены. Кроме того, остались возможности для улучшения проекта и дальнейшей разработки, внедрения новых функций, технологий и улучшения существующих.

Во время работы над проектом я впервые попробовал себя в веб-разработке, ознакомился с основами разработки фуллстек Javascript приложений, изучил библиотеку React и реализовал пользовательский интерфейс с помощью её функционала. Я узнал об архитектуре веб-приложений, технологии WebRTC и обмене видео и аудио потоками в реальном времени. Также я получил первый опыт взаимодействия с командой и распределения обязанностей в ней.

Руководитель проекта координировал нас, планировал созвоны, записывал обучающие видео для работы по проекту. Его опыт работы как фронтенд-разработчика понадобился мне и нашей команде во время разработки. Без него этот проект не был бы реализован. Команда работала планомерно, выполняла все задачи в срок, разногласий во время выполнения работы не было. В случае возникновения проблем или вопросов коллеги были готовы помочь.

Мой вклад в проект можно оценить неоднозначно. С одной стороны, мои обязанности мог выполнить наш фуллстек разработчик Владимир, так как он реализовал страницу конференции и обладает соответствующими компетенциями. Но с другой, в условиях дефицита времени такое распределение обязанностей позволило наиболее эффективно выполнить работу и достигнуть поставленных целей. Без моего участия в проекте качество результата бы ухудшилось, так что я считаю, что мой вклад в проект существенный.

Я приобрёл опыт участия в разработке полноценного функционального веб-приложение, который несомненно пригодится мне в будущей профессиональной деятельности. Было интересно участвовать в таком проекте на первом семестре первого курса. Несмотря на некоторые трудности, эта работа дала мне неоценимый опыт работы в команде, управления дедлайнами и распределения времени.

Я считаю, что ошибки и проблемы, с которыми я столкнулся во время работы, важнее, чем непосредственно результат работы, ведь именно на них мы учимся и избегаем их впоследствии. Один и тот же результат может быть получен разными путями, но вот ошибки уникальны и дают опыт, который невозможно получить по обучающим материалам и лекциям. Именно этот практический опыт наиболее ценен для меня, и его я получил благодаря этому проекту.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Немного о WebRTC: что где использовать и случай из практики [Электронный ресурс]: статья. URL: <https://habr.com/ru/articles/501416/> (дата обращения 02.01.2025)
2. Single Page Application [Электронный ресурс]: статья. URL: <https://ru.hexlet.io/courses/web-testing-basics/lessons/spa/theory_unit> (дата обращения 02.01.2025)
3. Стек MERN. Что, как и почему? [Электронный ресурс]: статья. URL: <https://habr.com/ru/articles/653981/> (дата обращения 03.01.2025)
4. Endpoint: что такое и как использовать [Электронный ресурс]: статья. URL: <https://tquality.ru/blog/kak-pisat-endpointy/> (дата обращения 04.01.2025)
5. React – JavaScript-библиотека для создания пользовательских интерфейсов [Электронный ресурс]: статья. URL: <https://ru.legacy.reactjs.org> (дата обращения 05.01.2025)

ПРИЛОЖЕНИЕ

Техническое задание

1 Название проекта

“Разработка веб-приложения для обмена видео-потоками (ВКС)”

2 Цель проекта

Создать приложение, предоставляющее возможность пользователям обмениваться видео-потоками с использованием технологии WebRTC.

3 Сроки выполнения

Начало –10.11.2024

Окончание – 20.12.2024

4 Руководитель проекта

Константинов Николай Юрьевич

5 Функциональные требования к проекту:

– наличие возможности видеть и слышать других пользователей,

– наличие возможности создать конференцию и пригласить других пользователей,

– наличие возможности у пользователей авторизоваться, если у них уже есть аккаунт,

– наличие у пользователя возможности делиться изображением со своей веб-камеры с другими пользователями (трансляции экрана).