**Министерство науки и высшего образования Российской Федерации**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**(Университет ИТМО)**

Факультет **Инфокоммуникационных технологий**

Образовательная программа **Мобильные и облачные технологии**

Направление подготовки (специальность) **09.03.03 Прикладная информатика**

**К У Р С О В О Й   П Р О Е К Т**

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

Обучающийся: Буй Тхук Хуен, К3139

Санкт-Петербург 2024

**СОДЕРЖАНИЕ**

[**ВВЕДЕНИЕ 3**](#_tefj0ukik3v3)

[1 Описание проекта 5](#_kqddgk7gxsbz)

[2 Работа над проектом 7](#_dwkfgp1000zo)

[2.1 Задачи команды 7](#_z5t15k7gzpp)

[2.2 Задачи, поставленные передо мной 7](#_37qo8s6yuq4x)

[2.3 Ход работы 7](#_57xbj141k6fz)

[3 Анализ выполненной работы 10](#_wqaqgtguwgl5)

[3.1 Успехи в проекте 10](#_3evxvxdpcg8r)

[3.2 Проблемы и трудности 11](#_uhavupda713a)

[3.3 Планомерная работа 11](#_kamsgg3a5kxw)

[3.4 Извлеченные уроки 11](#_5vddn19p2cmv)

[4 Взаимодействие с командой 11](#_6v0jz45mn7bm)

[5 Взаимодействие с руководителем проекта 12](#_3ux2at4eyudo)

[6 Оценка работы руководителя 13](#_wqld03p2lr6f)

[**ЗАКЛЮЧЕНИЕ 15**](#_9745gqtr7w7n)

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 17**](#_1mopynrlv50)

[**ПРИЛОЖЕНИЕ 18**](#_ggr8cmprmnam)

# **ВВЕДЕНИЕ**

В России миллионы людей испытывают необходимость в общении с родственниками, друзьями и знакомыми, которые зачастую находятся в других городах, странах или даже на различных континентах. В условиях глобализации и развития технологий, такие зарубежные приложения, как Zoom, Google Meet, WhatsApp и Skype, стали неотъемлемой частью повседневной жизни, позволяя пользователям обмениваться сообщениями и совершать видеозвонки. Однако текущая нестабильная экономическая и политическая ситуация создает риски ограничения доступа к этим сервисам, что может существенно повлиять на возможность общения.

Таким образом, задача разработки отечественного программного обеспечения становится особенно актуальной. Необходимо создать решения, которые смогут не только конкурировать с иностранными аналогами, но и эффективно заменять их на российском рынке. Это связано с несколькими важными аспектами. Во-первых, существует реальный риск ухода иностранных компаний, занимающихся разработкой программ для видеоконференций, с российского рынка. Это может привести к нехватке доступных инструментов для общения и совместной работы. Во-вторых, в России наблюдается дефицит разнообразия приложений для организации видеосвязи, что ограничивает выбор пользователей и создает потребность в новых решениях.

Кроме того, поддержание конкуренции является одним из ключевых принципов рыночной экономики и необходимо для обеспечения комфортной жизни граждан. Конкуренция способствует инновациям, улучшению качества услуг и снижению цен, что в конечном итоге выгодно для потребителей. В связи с этим разработка и внедрение отечественных платформ для видеосвязи не только отвечает на вызовы времени, но и создает основу для устойчивого развития цифровой инфраструктуры страны.

Цель курсовой работы заключается в создании приложения, предоставляющего возможность пользователям обмениваться видеопотоками с использованием технологии WebRTC. Для реализации этой цели предусмотрены следующие этапы:

1. Разработка клиентской части приложения, включая реализацию сервисной логики фронтенда и его инициализацию.
2. Реализация логики обмена видеопотоками в пользовательском интерфейсе.
3. Создание серверной части приложения, включающей разработку сигнального и серверного компонентов.
4. Проектирование и внедрение базы данных для хранения данных пользователей, таких как логины.
5. Разработка дизайна пользовательских сценариев, позволяющих визуализировать результаты взаимодействия с программой.
6. Написание автоматический тестов для проверки функциональности приложения.

# 

# **1 Описание проекта**

Проект представляет собой разработку веб-приложения для обмена видео-потоками, основанного на технологии WebRTC. Основной целью работы стало создание системы, которая обеспечивала бы удобную и безопасную видеосвязь для пользователей в реальном времени.

Для успешной реализации данного проекта в нашу команду были привлечены специалисты различных направлений: разработчики backend и frontend, которые отвечают за серверную и клиентскую части приложения соответственно, дизайнер, занимающийся созданием визуальных элементов пользовательского интерфейса, а также тестировщик, чья основная задача заключается в создании тестовой модели, охватывающей все возможные сценарии работы программы для оценки её функциональности. Каждая из этих ролей требует специфических навыков и знаний, необходимых для эффективного выполнения поставленных задач. Это создает необходимость для каждого участника команды развивать и углублять свои профессиональные компетенции в своей области ответственности. проект не ограничивается лишь разработкой приложения для видеоконференций; он также представляет собой возможность для участников развивать как профессиональные, так и личные навыки, а также приобретать опыт командной работы. Это позволяет каждому члену команды не только внести свой вклад в общий успех проекта, но и значительно расширить свои горизонты в сфере разработки программного обеспечения.

В основе серверной части лежит платформа Node.js, усиленная возможностями фреймворка Express, что позволило организовать гибкую маршрутизацию запросов и эффективное взаимодействие между клиентом и сервером. Для обеспечения безопасности хранения данных в системе была внедрена библиотека bcrypt, которая отвечает за надежное хэширование пользовательских паролей, значительно снижая риски утечки конфиденциальной информации.

Фронтальная часть приложения разработана с использованием современного стека технологий, включающего React, TypeScript и Vite. Это позволило создать интуитивно понятный и высоко производительный пользовательский интерфейс, включающий ключевые страницы: авторизацию, главную страницу и интерфейс видеоконференций. Приложение поддерживает прямую передачу видео- и аудиоданных через WebRTC, что обеспечивает минимальные задержки, улучшая качество связи и делая взаимодействие пользователей максимально комфортным.

Дизайн интерфейса был проработан с помощью инструмента Figma, который помог создать отзывчивую и визуально привлекательную систему, способную адаптироваться к различным устройствам. Это обеспечило простоту взаимодействия пользователей с приложением и поддержку на десктопах, планшетах и смартфонах.

Для повышения надежности и стабильности работы приложения было проведено автоматизированное тестирование с использованием Cypress, что позволило исключить возможные ошибки и гарантировать стабильность функционала.

Итоговый продукт представляет собой производительное, безопасное и простое в использовании решение, которое может успешно применяться в реальных условиях, отвечая требованиям пользователей к качеству и надежности современных видеосервисов.

## 

# **2 Работа над проектом**

## **2.1 Задачи команды**

В начале работы над проектом командой был составлен план разработки:

1. Создание и редактирование технического задания
2. Распределение обязанностей и установка сроков выполнения задач
3. Выполнение индивидуальных задач
4. Подготовка к защите проекта
5. Защита проекта - командная защита с презентацией
6. Написание индивидуального отчета

## **2.2 Задачи, поставленные передо мной**

Я принимаю участие в проекте, как тестировщик. Передо мною стоят следующие задачи: Изучить технологию Cypress и написать автоматические тесты для проекта.

## **2.3 Ход работы**

Чтобы начать проект, мне нужно изучить принципы работы Cypress — инструмента, который помогает в автоматизированном тестировании [1]. Я прочитала некоторые документы, такие как: документацию на официальном сайте Cypress [2]. Статья на Хабре [3], в которой описан простой способ автоматизации с помощью Cypress. Он включает в себя пошаговые инструкции по установке и написанию автоматических тестов с использованием JavaScript, а также практические примеры тестирования веб-приложений. Мне также нужно было научиться создавать новый проект Cypress, настраивать среду и использовать функции поддержки WebStorm, такие как завершение кода и тестирование кода [4].

Дальше я пошла изучать проект. В ходе обсуждений я определила некоторые основные функции приложения, которые необходимо протестировать, такие как: регистрация и вход в систему, создание собраний (конференц-залов), передача видео-аудиоданных.

Поскольку рабочий процесс между позициями в команде происходит параллельно, мне сначала нужно создать сайт-симулятор для проведения тестирования.

Создаю симулированная веб-страница [3] используя файл Mock.html. Затем написала функциональные тесты:

1. Bход в систему:
2. Case 1. Отображение регистрационной формы: Проверяет наличие полей ввода (#username, #email, #password) и кнопки отправки (#submit-button).
3. Case 2. Успешная регистрация: Заполнит поля и нажмите кнопку отправить. Проверить, отображается ли сообщение об успехе.
4. Case 3. Обязательная проверка поля: Нажмит кнопку «Отправить», не заполняя никаких полей. Проверяет, не отмечены ли поля как недействительные.
5. Case 4. Проверка неверного формата электронной почты: Введит неверный адрес электронной почты и нажмите кнопку «Отправить». Проверяет, не помечено ли поле электронной почты как недействительное.

На рисунке 1 показан смоделированный веб-сайт, проверка функции входа выполняется в 4 случаях.

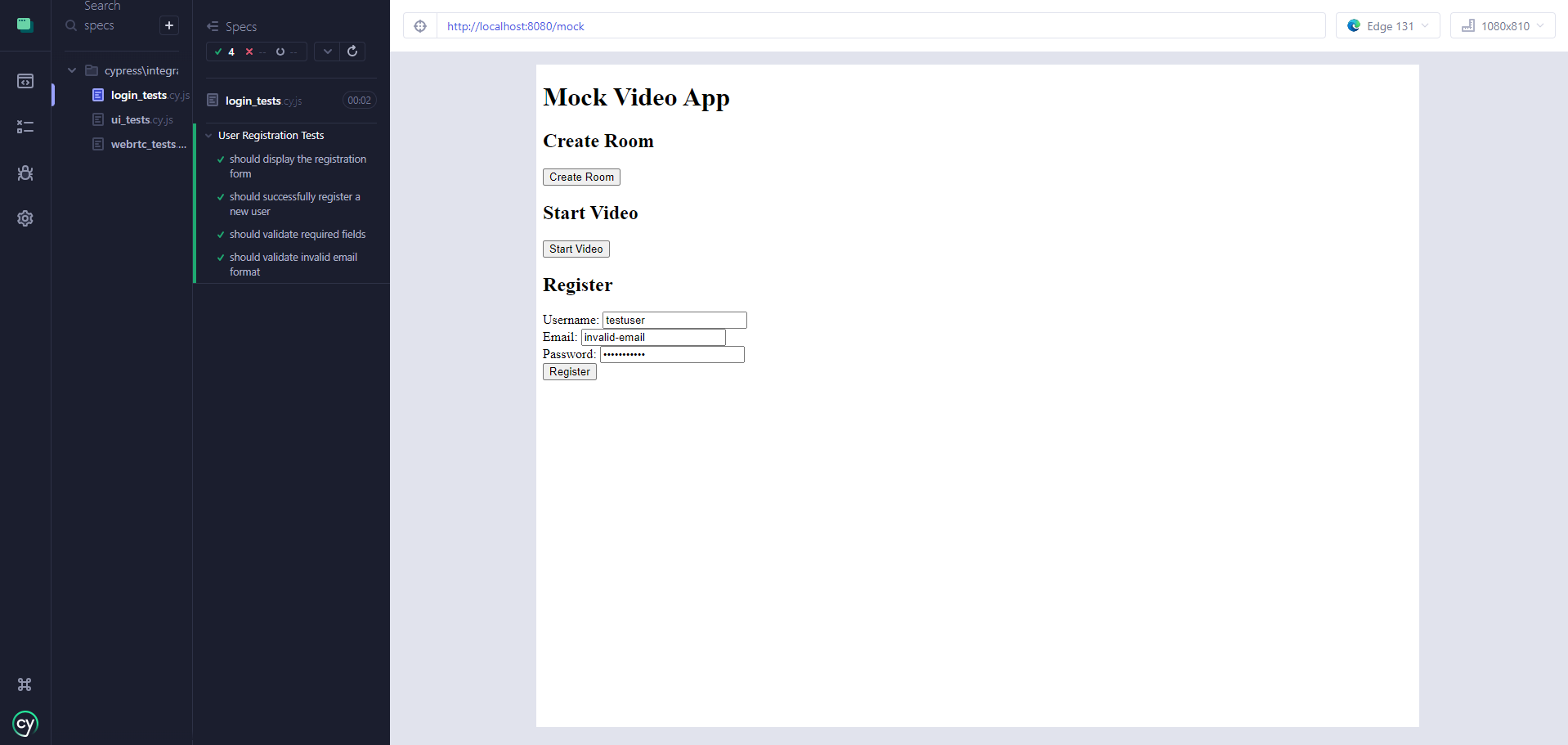


Рисунок 1 – тест входа в систему

b. Создать встречу: Прежде чем нажать на нее, проверяет, видна ли кнопка «Create Room». Убедит, что идентификатор комнаты был успешно создан.

На рисунке 2 показан смоделированный веб-сайт, на котором выполняется проверка функции создания собрания (переговорной комнаты).

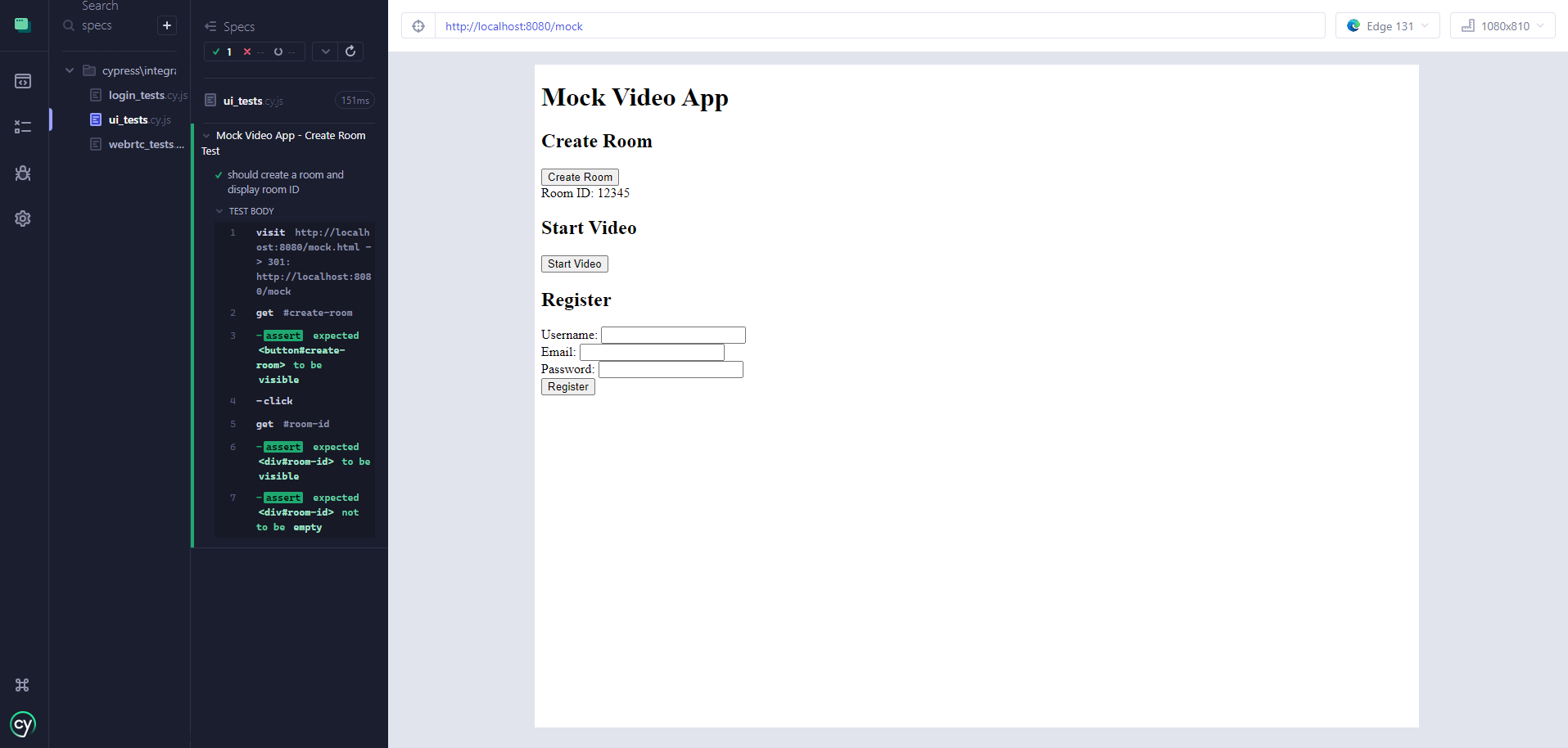


Рисунок 2 – тест функции создания встречи

c. Запустить видео: Найдет кнопку «Начать видео» и проверяет, видна ли она. Проверяет, воспроизводится ли видео после получения старта.

На фото 3 показан тест функции передачи аудио и видеоданных на моделируемом сайте.

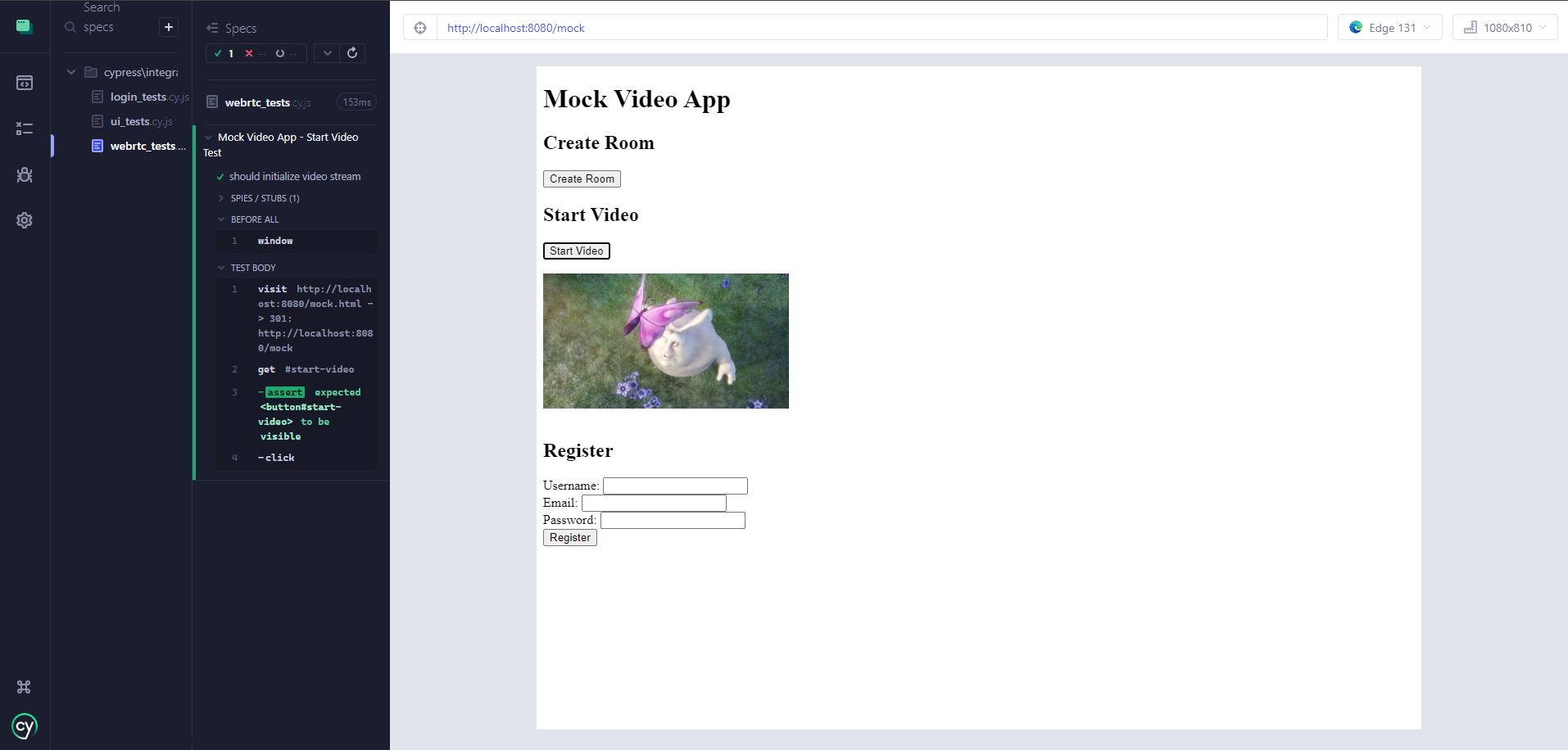


Рисунок 3 – тест функции запуска видео

Затем я внесла улучшения и расширения для более надежного тестирования. Cypress — это мощный инструмент автоматизации тестирования современных веб-приложений. Он помогает эффективно проводить тестирование.

# **3 Анализ выполненной работы**

В этом разделе я проанализирую свой опыт работы над проектом "Разработка веб-приложения для обмена видео-потоками (ВКС)", выделяя как достижения, так и трудности, с которыми я столкнулся в процессе реализации.

## **3.1 Успехи в проекте**

В процессе работы над проектом "Разработка модуля обратной связи для обучающей платформы" я успешно освоил Cypress, что позволило разработать автоматизированные тесты для проверки функционала модуля. В частности, были протестированы такие функции, как отправка обратной связи, валидация пользовательских данных и корректное отображение модуля на различных устройствах. Благодаря этим тестам удалось существенно снизить количество ошибок на этапах ручного тестирования и внедрения продукта.

## **3.2 Проблемы и трудности**

Одной из основных трудностей стало недостаточное знание JavaScript и ограниченный опыт работы с Cypress. Это замедляло процесс написания тестов и приводило к необходимости доработок. Особенные сложности вызывали асинхронные операции и оптимизация стабильности тестов, что потребовало значительного времени на изучение документации и методов работы инструмента.

## **3.3 Планомерная работа**

Для преодоления трудностей я активно изучала документацию, проходил курсы по JavaScript и Cypress, а также делила сложные задачи на этапы, чтобы постепенно улучшать навыки. Важную роль сыграло взаимодействие с коллегами, которые помогли решать технические вопросы и давали полезные советы по оптимизации тестов.

## **3.4 Извлеченные уроки**

Работа над проектом показала важность планирования тестирования и выбора приоритетных сценариев, что позволило эффективно распределять ресурсы. Я поняла, что обучение через практику и постоянное совершенствование навыков играют ключевую роль в освоении новых технологий. Также я осознала, насколько важно писать лаконичные и точные тесты, чтобы обеспечить их стабильность и удобство в поддержке. Этот опыт стал важным шагом в моем профессиональном развитии как тестировщика.

# **4 Взаимодействие с командой**

В процессе работы над проектом мы использовали социальную сеть Telegram в качестве основного инструмента для общения. Это решение было обусловлено рядом преимуществ, таких как удобный и знакомый интерфейс, возможность организации видеозвонков и доступность приложения для всех членов команды. Коммуникация осуществлялась следующим образом: примерно раз в несколько дней все участники проекта собирались на заранее запланированное видеозвонок, где обсуждали организационные вопросы. Каждый член команды делился достигнутыми результатами с момента последнего общения, описывал возникшие трудности и, при необходимости, запрашивал советы у коллег. Также происходила согласование результатов работы и корректировка действий в соответствии с поставленными целями.

Я являлась единственным иностранным участником группы, что усложнило взаимодействие с остальными членами команды. Лидер группы прилагал все усилия для того, чтобы объяснить мне задачи и помочь наладить контакт с другими участниками. В целом атмосфера в команде была дружелюбной: все доверяли друг другу и старались поддерживать как морально, так и практическими действиями. Изначально я испытывала беспокойство по поводу возможных задержек в работе из-за языкового барьера, однако работа в команде приносила мне удовлетворение.

# **5 Взаимодействие с руководителем проекта**

Теперь следует отметить взаимодействие с руководителем проекта, опытным фронтенд-разработчиком Николаем Константиновым. Николай был организатором всех процессов разработки и моральной опорой для команды. Он всегда был доступен для помощи и выступал в роли модератора наших видеоконференций, оценивая работу каждого участника и учитывая мнения команды при определении направлений доработки. Его стиль руководства не отличался жесткостью; он полагался на наше чувство ответственности за результаты работы. Я уверена, что мы не подвели ни его, ни себя.

Особое внимание следует уделить просветительской деятельности Николая. Он записывал информативные видеоролики на темы, актуальные для каждого участника разработки. Например, он подготовил для меня видео, в котором подробно объяснял функции, которые развивает наше приложение, и предлагал сделать тест более полным и оптимальным. Эти акции оказались очень полезны при разработке нашего приложения.

Николай также занимался организацией нашей подготовки к выступлению. Он попросил каждого участника подготовить примерную речь, после чего мы несколько раз репетировали с учетом его замечаний по интонации и содержанию.

Подводя итог описанию взаимодействия с командой и руководителем проекта, могу с уверенностью утверждать, что без внимательного руководства Николая нашей команде вряд ли удалось бы создать продуктивный и практико-ориентированный продукт. Я могу дать ему исключительно положительную оценку как руководителю.

# **6 Оценка работы руководителя**

Работа нашего руководителя – Константинов Николай Юрьевич, оказалась исключительно полезной и результативной на протяжении всего проекта. Его высокий профессионализм и обширный опыт в сфере разработки программного обеспечения внесли значительный вклад в обучение и развитие всей команды.

Прежде всего, Николай Константинов всегда был готов дать консультации и ответить на любые наши вопросы. Его оперативность и стремление помочь создали атмосферу взаимного доверия и поддержки, что укрепило нашу уверенность в процессе работы. Помимо предоставления необходимых материалов и ресурсов, он делился собственными знаниями, что позволило нам лучше разбираться в сложных аспектах проекта.

Кроме того, его умение давать конструктивную обратную связь оказалось крайне ценным. Он указывал на ошибки нашей группы и предлагал конкретные способы их устранения, что способствовало профессиональному развитию каждого участника команды. Благодаря его советам мы смогли повысить свои навыки программирования и более эффективно справляться с возникающими задачами.

Также стоит отметить, что Николай Константинов регулярно контролировал наш прогресс и выполнение задач, что помогало нам сохранять фокус и не упускать важных деталей. Его внимание к мелочам и стремление к качественному результату вдохновляли нас достигать высоких целей.

В заключение, мы убеждены, что вклад Николай Константинов в успех нашего проекта был бесценным. Его поддержка, профессионализм и опыт стали ключевыми факторами нашего обучения и развития как команды разработчиков. Мы искренне благодарны ему за его усилия и уверены, что его наставничество окажет положительное влияние на карьеру каждого из нас в будущем.

# **ЗАКЛЮЧЕНИЕ**

Цели нашего проекта были успешно достигнуты, что является значительным достижением для всей команды. Мы разработали инновационное приложение для обмена видео, которое отвечает современным требованиям пользователей. В процессе реализации проекта мы не только выполнили поставленные задачи, но и обеспечили пользователей необходимыми функциональными возможностями.

В качестве результата проекта получилось создать приложение, посредством которого пользователи могут организовывать видеоконференции для общения друг с другом. Конечно же, данная программа ещё нуждается в крупных доработках, но, давая оценку проделанной работе, я с уверенностью могу заявить, что поставленная перед командой цель была выполнена. Разработчики и прочие члены команды смогли реализовать тот минимум, который необходим для совершения пользователем звонка.

Мой личный вклад в достижение целей проекта заключался в активном участии в тестировании сервиса. В рамках этой ответственности я занимался автоматизацией тестирования ключевых функций приложения, таких как регистрация и вход в систему, создание встреч, а также передача аудио и видеоданных. Эти действия значительно повысили надежность приложения и оптимизировали нашу систему предоставления услуг. Автоматизация тестирования позволила сократить время на выявление и устранение ошибок, что в свою очередь способствовало более быстрому выходу продукта на рынок.

Отдельно нужно сказать про те моменты, которые команда не успела добавить в разрабатываемое приложение. Как я писал выше, поставленная перед командой цель выполнена, однако у нас получилось не всё, что мы хотели реализовать. Одной из особенностей нашего приложения должна была быть возможность менять фон у собеседника во время звонка, как, например, это сделано в Zoom. Причина недоработки - нехватка времени: были объявлены слишком короткие сроки, чтобы команда успела реализовать это дополнение. Однако я уверен, что в будущем мы обязательно добавим это!

Несмотря на некоторые трудности, с которыми мы столкнулись в процессе работы над проектом, результаты демонстрируют его успешность. Мы уверены, что достигнутые результаты станут основой для дальнейшего развития нашего приложения и позволяют нам продолжать улучшать пользовательский опыт.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

[1] Официальный сайт Сайпресса – [Testing Frameworks for Javascript | Write, Run, Debug | Cypress](https://www.cypress.io/)

[2] Ресурсы по использованию Cypress – [Cypress Documentation](https://docs.cypress.io/app/get-started/why-cypress)

[3] [Хабр](https://habr.com/ru/articles/792894/) – [Cypress для Junior QA. Легкий способ войти в автоматизацию / Хабр](https://habr.com/ru/articles/792894/)

[4] JetBrains – [Web UI Test Automation | WebStorm Documentation](https://www.jetbrains.com/help/webstorm/ui-test-automation.html#debug-tests)

[5] Имитированный веб-сайт для проведения автоматического тестирования – <http://localhost:8080>

# 

# **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 01 ноября 2024 г.

Конец 20 декабря 2024 г.

**4 Руководитель проекта**

Константинов Николай Юрьевич

**5 Требования к проекту**

**5.1 Технические требования**

* 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)

**5.2 Требования к стеку**

* Фронтенд: React, Vite, TypeScript, WebRTC
* Бэкенд: Node.js, Express, PostgreSQL, WebRTC
* Дизайн: Figma
* Тестирование: Cypress, Webstorm

**6 Термины и сокращения**

* WebRTC (Web Real-Time Communication) — это технология, которая позволяет веб-браузерам и приложениям обмениваться видео, аудио и данными в реальном времени без использования серверов-посредников.

**7 Содержание работы**

Таблица 1 - Этапы проекта и сроки их выполнения

| Этапы проекта | Сроки выполнения этапов | Ответственный за этап | Вид представления результатов этапа |
| --- | --- | --- | --- |
| Разработка технического задания | 10 ноября | Николай Константинов | Файл Google doc |
| Разработка фронтенда | 10 декабря | Егор Янин  Владимир Глазыкин | Разработанные интерфейсы |
| Разработка бэкенда | 10 декабря | Владимир Глазыкин  Андрей Останин | Серверная часть прототипа |
| Создание дизайна | 10 декабря | Александр Кучин | Разработанные компоненты на Figma |
| Тестирование | 10 декабря | Буй Тхук Хуен | Наборы тестов на Cypress |
| Подготовка к защите проекта | 17 декабря | Bce | Файл Google slide |
| Защита проекта | 20 декабря | Bce | Сдача отчета, предоставление доклада с презентацией и демонстрация работы прототипа платформы. |

**7.1 Задачи фронтенда – Егор Янин, Владимир Глазыкин**

* Реализовать логику обмена видео-потоками во фронтальном приложении: Создание хуков для работы с WebRTC.
* Реализовать серверную логику фронтального приложения: Создание логики приложения для обработки форм и отправки данных на сервер.
* Реализовать UI фронтального приложения: Сверстать интерфейсы с использованием React и css-фреймворком.

**7.2 Задачи бэкенда – Владимир Глазыкин, Андрей Останин**

* Реализовать сигнальный сервер: Создание эндпоинтов для передачи данных с помощью WebRTC
* Инициализировать серверное приложение: Воспользоваться документацией и развернуть приложение: ﻿[﻿https://vite.dev/guide/](https://vite.dev/guide/)
* Реализовать сервисные эндпоинты: Создать эндпоинты для обработки запросов с фронта для регистрации изменения данных пользователя.
* Реализовать базу данных: Создание базы данных для записи логов и данных пользователей

**7.3 Задачи дизайна – Александр Кучин**

* Создать дизайн интерфейса приложения: Создание борда в Figma с дизайном формы регистрации и трансляции.
* Создать кейсы для ручного тестирования: Создание тестовой модели с описанием кейсов для проверки работоспособности приложения.

**7.4 Задачи тестирования – Буй Хуен**

* Написать автоматических тестов: Изучить технологию cypress и написать автоматические тесты для проекта  
  ﻿[﻿https://www.cypress.io/](https://www.cypress.io/)