

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Уральский федеральный университет  
имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий – РТФ  
Школа бакалавриата

## ОТЧЕТ

По проекту  
«Разработка портала "Лига речи"»  
по дисциплине «Проектный практикум»

Заказчик: Пинженин П.С

Куратор: Новиков М.Ю.

к.п.н., доцент

Студенты команды Fish.Net

Вакалюк С.С.

Волгин М.Р.

Гамова О.А.

Муллаярова А.Р.

Чубаренко С.А.

---

---

---

---

---

---

---

---

Екатеринбург, 2025

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1 Основная часть .....	5
1.1 Аналитика .....	5
1.1.1 Работа с требованиями и анализ аналогов .....	5
1.1.2 Работа с пользовательскими сценариями.....	6
1.2 Дизайн .....	10
1.3 Frontend-разработка .....	14
1.3.1 Реализация основного функционала .....	14
1.3.2 Тестирование и отладка.....	18
1.4 Backend-разработка.....	19
1.4.1 Архитектура продукта .....	19
1.4.2 Тестирование и разбор ошибок .....	21
1.5 Организация работы .....	22
ЗАКЛЮЧЕНИЕ .....	25
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	27
ПРИЛОЖЕНИЕ А (обязательное) Диаграмма бизнес-процессов .....	28

## ВВЕДЕНИЕ

В условиях цифровизации образования и бизнес-коммуникаций растёт спрос на качественные платформы для развития ораторского мастерства. Проект "Лига речи" создаётся как специализированный портал, адаптированный под потребности целевой аудитории с учётом региональных особенностей.

Разработка платформы обусловлена необходимостью предоставить пользователям современный инструмент для совершенствования навыков публичных выступлений. Проект учитывает требования к локализации контента, технической надёжности и соответствию законодательным нормам.

Цель работы: разработать веб-портал "Лига речи"

Для достижения этой цели были поставлены следующие задачи:

- 1) ознакомиться с техническим заданием, предоставленным заказчиком и проанализировать его;
- 2) проанализировать доступные аналоги для выявления их основных недостатков;
- 3) разработать дизайн-макеты согласно требованиям заказчика;
- 4) спроектировать архитектуру сайта;
- 5) сверстать сайт по дизайн-макетам;
- 6) разместить продукт на хостинге.

Портал предназначен для:

- тренеров и преподавателей ораторского искусства;
- участников курсов и тренингов;
- организаторов мероприятий в сфере публичных выступлений;
- бизнес-сообществ, заинтересованных в развитии коммуникационных навыков у своих сотрудников.

В результате работы планируется представить заказчику работающий прототип сайта, соответствующий, по меньшей мере, большинству утвержденных требований. Кроме этого, разработанный продукт послужит

основой для дальнейшей разработки версий портала «Лига речи» для других городов.

## 1 Основная часть

### 1.1 Аналитика

#### 1.1.1 Работа с требованиями и анализ аналогов

Перед началом работы над проектом от заказчика было получено техническое задание, представленное в неудобном для работы виде, а также содержащее требования, которые команды сочли невыполнимыми в текущих временных рамках. В связи с этим было проведено интервью с заказчиком для актуализации и утверждения требований. После этого предоставленное заказчиком техническое задание было изменено, формализовано и повторно утверждено.

Был проведен анализ аналогов, в числе которых сообщество «Лига Речи» Вконтакте [1], Telegram-канал «Лига Речи ЕКБ» [2], сайт лигаречи.рф [3]. Сравнение аналогов по ключевым признакам представлено в таблице 1.

Таблица 1 – Сравнение аналогов

Характеристика для сравнения	Сообщество Вконтакте	Канал в Телеграм	Сайт
Календарь мероприятий	Анонс каждого мероприятия в отдельном посте	Анонс каждого мероприятия в отдельном посте + напоминания в реальном времени	Мероприятия указаны в хронологическом порядке
Донаты	Для просмотра информации нужно перейти на сайт	Для просмотра информации нужно перейти на сайт	Указаны банковские реквизиты, страница оплаты отсутствует
Аккаунты	Все действия осуществляются с личного аккаунта ВК	Все действия осуществляются с личного аккаунта ТГ	Возможность создания аккаунта не предусмотрена
Взаимодействие с организаторами	Аккаунты организаторов, номер телефона, лс сообщества	Чат канала, ссылки на аккаунты организаторов	Номер телефона, аккаунты ВК организаторов

Продолжение таблицы 1

Характеристика для сравнения	Сообщество Вконтакте	Канал в Телеграм	Сайт
Медиаархив	Альбомы с фото с каждого мероприятия в соответствующем разделе	Ссылка на ВК	Ссылка на ВК
Управление контентом	Через аккаунт со статусом администратора	Через аккаунт со статусом администратора	Через интерфейс редактирования страницы

В результате сравнительного анализа стали явно видно недостатки существующих альтернатив и понятно желание заказчика создать более удобный продукт.

### 1.1.2 Работа с пользовательскими сценариями

Далее были составлены пользовательские сценарии, описывающие взаимодействие пользователей с разрабатываемым сервисом. Составленные сценарии были категоризированы по ролям пользователей с разными уровнями доступа:

1) Для гостя (неавторизованного пользователя):

- Как гость, я хочу просматривать главную страницу, чтобы ознакомиться с деятельностью Лиги Речи;
- Как гость, я хочу видеть список ближайших мероприятий, чтобы планировать свое участие;
- Как гость, я хочу фильтровать мероприятия по типу (проекты/сообщества), чтобы найти подходящие мне события;
- Как гость, я хочу просматривать галерею фото с прошедших мероприятий, чтобы понять атмосферу событий;
- Как мобильный пользователь, я хочу иметь удобный интерфейс на маленьком экране, чтобы пользоваться порталом в дороге;

2) Для активиста:

- Как активист, я хочу зарегистрироваться в системе, чтобы получить доступ ко всем возможностям;
- Как активист, я хочу заполнить свой профиль, чтобы другие участники могли меня найти;
- Как активист, я хочу записываться на мероприятия, чтобы участвовать в событиях;
- Как активист, я хочу просматривать историю своих посещений, чтобы отслеживать активность;

3) Для тренера:

- Как тренер, я хочу создавать мероприятия, чтобы привлекать участников;
- Как тренер, я хочу управлять списком участников, чтобы координировать мероприятие;
- Как тренер, я хочу редактировать информацию о своем сообществе, чтобы поддерживать актуальность данных;

4) Для администратора:

- Как администратор, я хочу управлять проектами, чтобы поддерживать актуальность информации;
- Как администратор, я хочу настраивать систему донатов, чтобы мотивировать участников;
- Как администратор, я хочу управлять пользователями, чтобы поддерживать порядок в системе.

В соответствии со сценариями использования были составлены критерии приемки. Аналогично сценариям, они разделены по ролям пользователей системы:

1) Для гостя (неавторизованного пользователя):

- Когда пользователь открывает сайт, он видит заголовок, краткое описание деятельности Лиги Речи и основные разделы сайта;

- Когда пользователь заходит в раздел "Мероприятия", он видит список событий, отсортированных по дате;
- Когда пользователь применяет фильтр "Проекты" или "Сообщества", в списке остаются только мероприятия выбранного типа;
- Когда пользователь заходит в раздел "Галерея", он видит фотографии с прошедших событий, сгруппированные по мероприятиям;
- Когда пользователь открывает сайт на смартфоне, все элементы интерфейса (меню, кнопки, текст) адаптируются под размер экрана и остаются удобными для использования;

## 2) Для активиста:

- Когда пользователь заполняет форму регистрации (email/телефон, пароль) и подтверждает данные, он получает доступ в личный кабинет;
- Когда пользователь вводит данные в разделе "Мой профиль" (город, соцсети, навыки), информация сохраняется и становится видимой для других участников;
- Когда пользователь нажимает кнопку "Зарегистрироваться" на странице мероприятия, его статус меняется на "Участник", а событие добавляется в его расписание;
- Когда пользователь открывает раздел "Мои мероприятия", он видит список прошлых и текущих событий, в которых участвовал, с датами и статусами;

## 3) Для тренера:

- Когда тренер заполняет форму создания мероприятия (название, дата, описание, тип) и подтверждает, событие появляется в общем календаре и становится доступным для регистрации участников;
- Когда тренер открывает страницу своего мероприятия, он видит список зарегистрированных участников и может изменять их статус;
- Когда тренер вносит изменения в описание сообщества (логотип, текст, контакты), обновленные данные сразу отображаются в соответствующем разделе сайта;



4) Для администратора:

- Когда администратор создает, редактирует или архивирует проект, изменения моментально применяются на сайте, включая связанные мероприятия и списки участников;
- Когда администратор устанавливает новые пороги для регалий, эти условия автоматически применяются к профилям пользователей после их следующих пожертвований;
- Когда администратор изменяет роль пользователя, эти действия сразу влияют на доступ пользователя к функциям системы.

Согласно пользовательским сценариям была составлена диаграмма прецедентов (рисунок 1). Она представляет собой графическое изображение вариантов использования (use case) системы, акторов (actors) и их взаимодействия в виде эллипсов и прямоугольников. Варианты использования описывают функциональность системы с точки зрения ее пользователей, а акторы представляют внешние сущности, которые используют систему.

Суть диаграммы вариантов использования заключается в том, что она помогает лучше понять требования к системе и определить ее функциональные возможности. Она используется для определения сценариев использования системы и для выявления потенциальных проблем взаимодействия между пользователями и системой. Диаграмма вариантов использования является основным инструментом для описания поведения системы на ранней стадии ее проектирования.

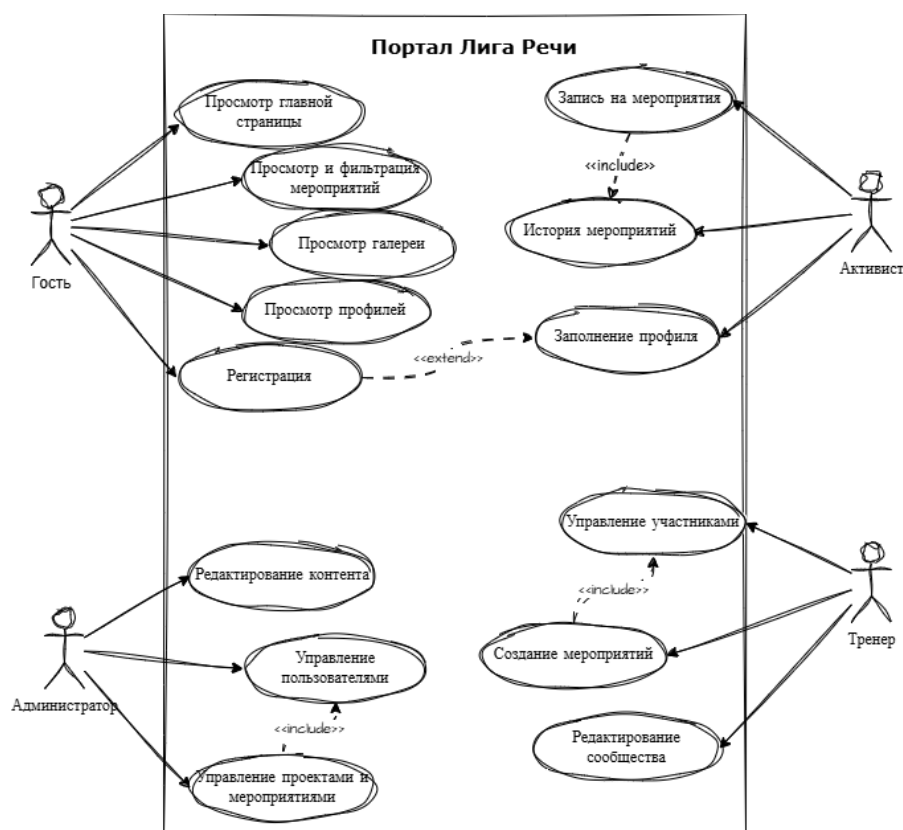


Рисунок 1 – Диаграмма прецедентов

Также была составлена диаграмма бизнес-процессов (приложение А). С помощью таких диаграмм работа идет динамичнее, так как разработчикам не нужно тратить время на то, чтобы вникать в систему и разбираться в процессах.

Команда разработки и заказчик лучше понимают друга, BPMN-нотация исключает возможность «двойного прочтения», а значит и недопониманий тоже. BPMN наглядно показывает слабые места, где потенциальные клиенты могут уйти. А значит, исправить или вовсе предотвратить “утечку” будет намного проще.

## 1.2 Дизайн

В рамках разработки веб-сайта для организации «Лига речи» по части UI/UX-дизайна был создан комплексный дизайн-проект, включающий витрину сайта, интерфейсы входа и регистрации, а также набор страниц

личного кабинета пользователя и разделов, связанных с мероприятиями и проектами.

Основные экраны:

1) Витрина сайта: стартовая страница (рисунок 2) содержит краткое описание деятельности «Лиги речи», ссылки для связи, информацию о резидентских сообществах и партнерах, команду организации, а также календарь ближайших мероприятий. Интерфейс выполнен в едином визуальном стиле, с акцентом на доступность информации и интуитивную навигацию.

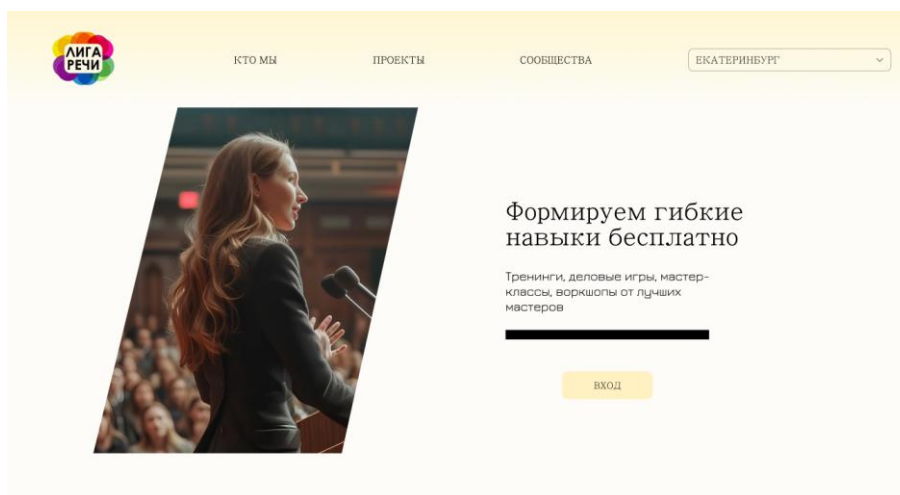


Рисунок 2 – Частичная демонстрация витрины сайта

2) Страницы аутентификации: реализованы страницы входа (рисунок 3) и регистрации с учетом пользовательского опыта и современных требований к безопасности и удобству.

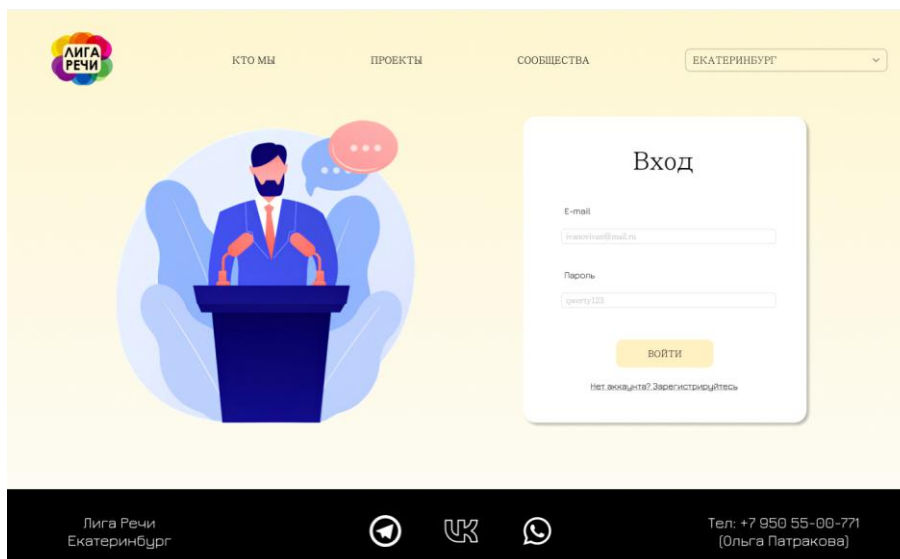


Рисунок 3 – Страница входа

- 3) Личный кабинет пользователя: спроектированы следующие разделы:
- Донаты – пользователь может отслеживать свои пожертвования;
  - Мой уровень – роль участника (лидер, тренер, куратор, гость);
  - Обо мне (рисунок 4) – персональная информация;
  - Портфолио – проекты, в которых участвовал пользователь;
  - Мои проекты – управление проектами и сообществами (для продвинутых аккаунтов);
  - Участники – управление участниками платформы (для продвинутых аккаунтов);
  - Мероприятия – управление мероприятиями (для продвинутых аккаунтов).

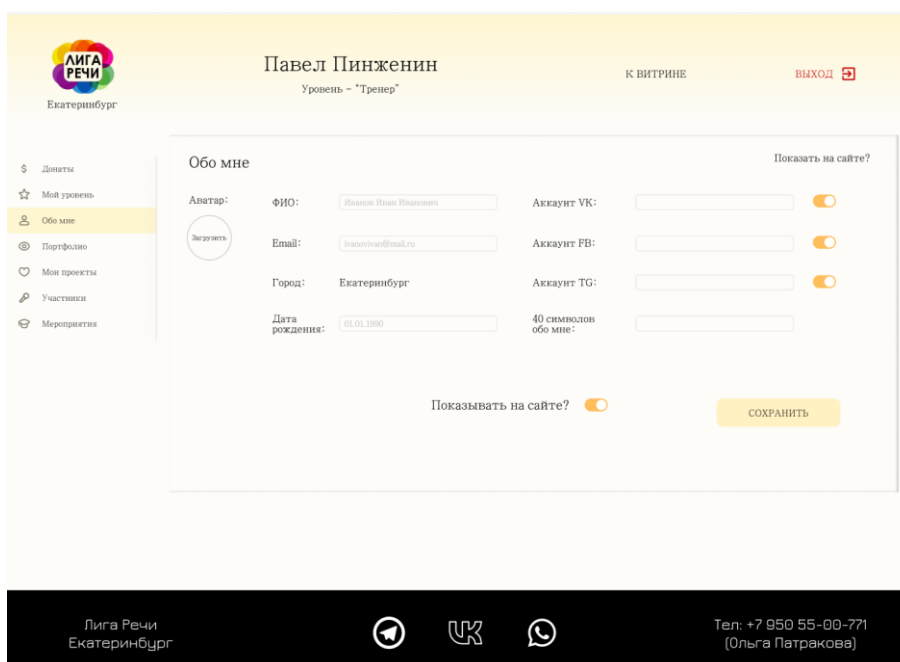


Рисунок 4 – Раздел «Обо мне» в аккаунте пользователя

4) Страницы проектов (рисунок 5) и карточки мероприятий: спроектированы отдельные страницы для отображения деталей проектов и мероприятий, включая расписание, описание, участников.

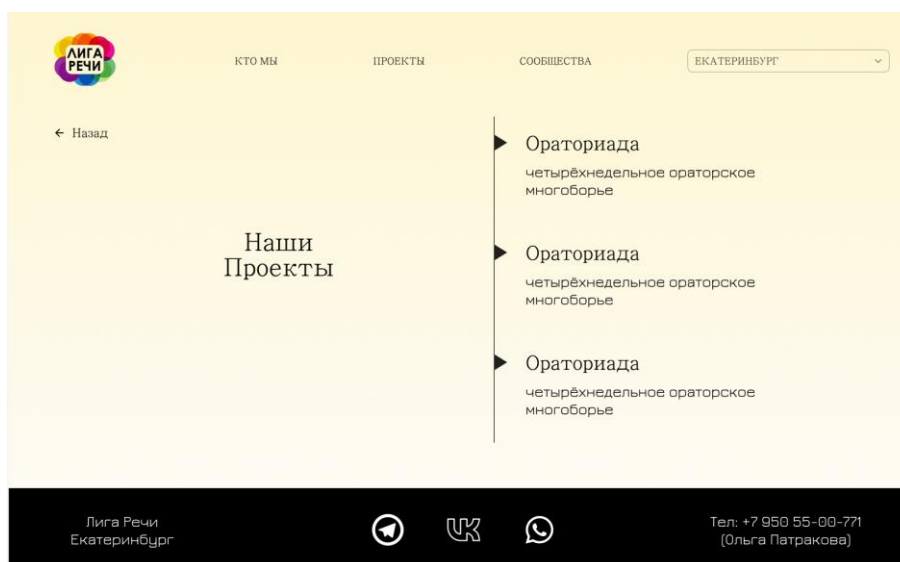


Рисунок 5 – Страница проектов

Использованные инструменты и подходы:

- Для создания некоторых компонентов и ускорения разработки интерфейсов использовалась библиотека Gravity UI, что позволило обеспечить визуальную согласованность.

- Были проработаны состояния элементов интерфейса (наведение, клик, активные и неактивные состояния).

- Настроен кликабельный прототип, демонстрирующий основные пользовательские сценарии. Это позволило протестировать навигацию и логику взаимодействия до этапа разработки.

Готовый прототип охватывает все ключевые пользовательские сценарии и может быть использован как полноценная основа для фронтенд-разработки.

## 1.3 Frontend-разработка

### 1.3.1 Реализация основного функционала

В рамках реализации проекта «Лига Речи» была разработана клиентская часть веб-приложения с использованием современных фронтенд-технологий. Основной задачей являлось создание интуитивно понятного, отзывчивого и функционального интерфейса, обеспечивающего пользователям удобное взаимодействие с сервисом.

Для разработки был выбран следующий стек технологий:

- **React** – основной фреймворк для построения пользовательского интерфейса [4].

- **TypeScript** – обеспечивает статическую типизацию, повышает надежность кода и упрощает его поддержку и рефакторинг [5].

- **Axios** – позволяет выполнять HTTP-запросы к серверной части приложения [6].

- **React Router** – позволяет реализовать навигацию в формате Single Page Application (SPA) без перезагрузки страницы [7].

– **Gravity UI** – UI-библиотека, использованная для создания согласованного дизайна и ускорения разработки компонентов [8].

Данный стек обеспечил высокую производительность, масштабируемость и удобство дальнейшего развития проекта.

В ходе разработки были созданы следующие ключевые разделы приложения:

Главная страница («Витрина») – информационный раздел с основными возможностями сервиса. (рисунки 6, 7, 8)

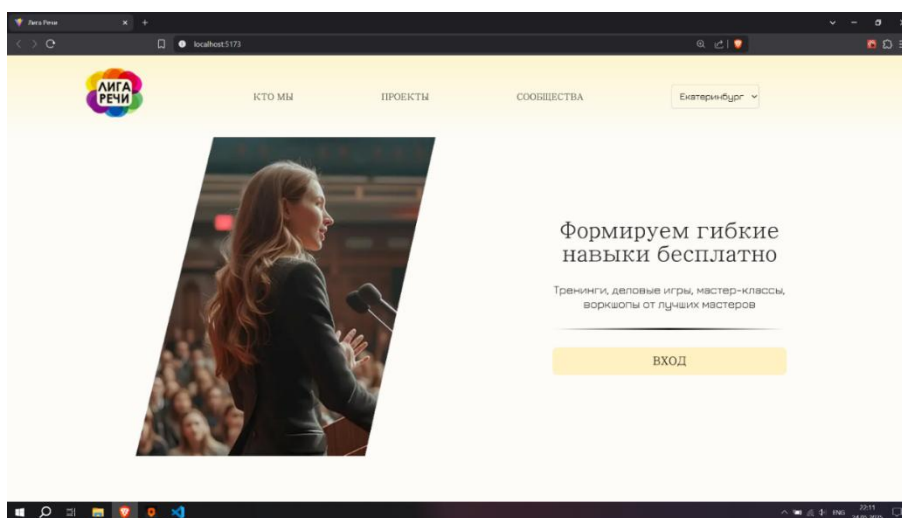


Рисунок 6 – главная страница портала, секция «Него»

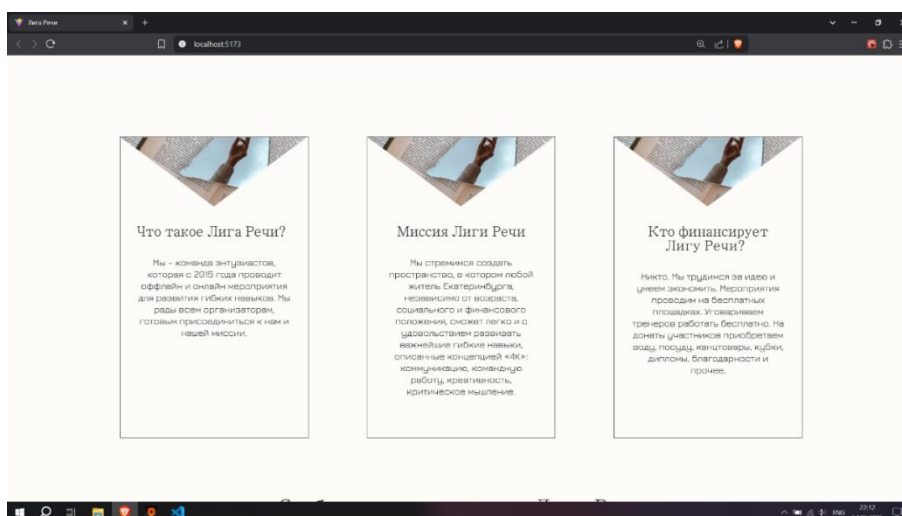


Рисунок 7 – главная страница портала, секция «О нас»

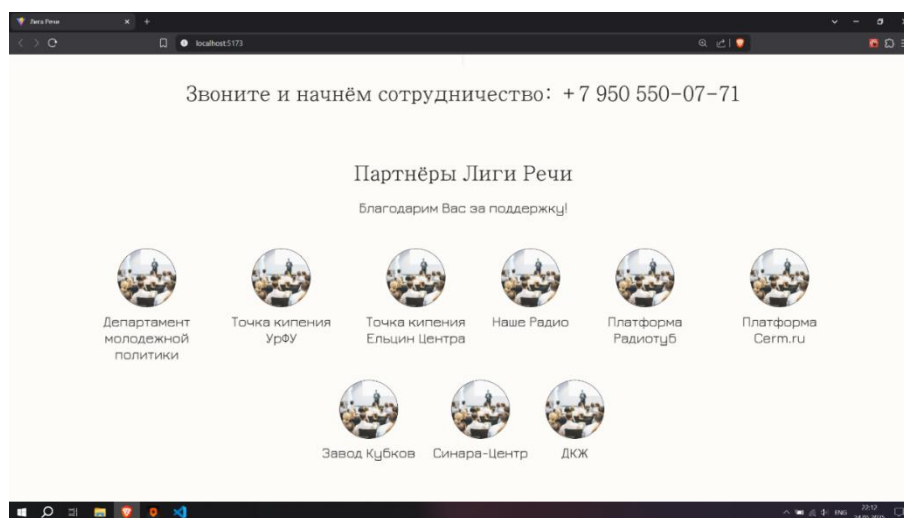


Рисунок 8 – главная страница портала, секция «Партнёры Лиги Речи»

Система аутентификации – страницы регистрации и входа в личный кабинет с валидацией данных (рисунок 9, 10).

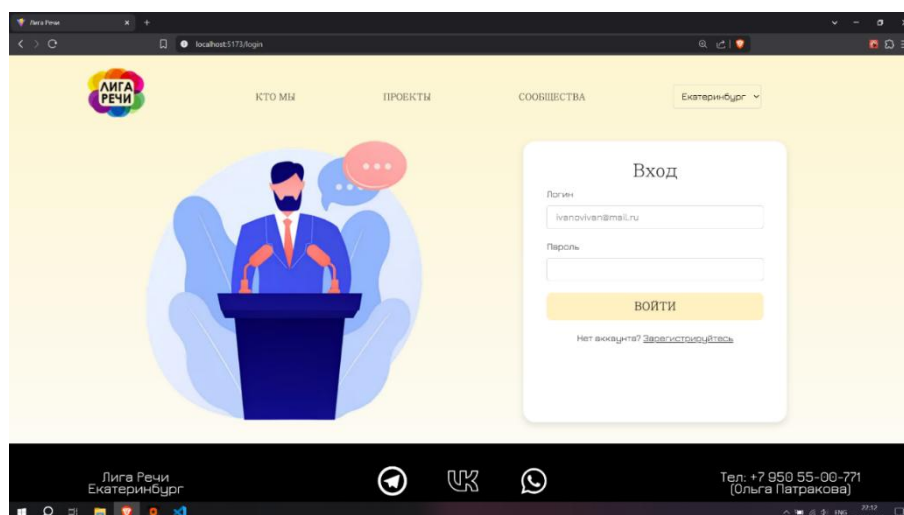


Рисунок 9 – аутентификация, страница входа



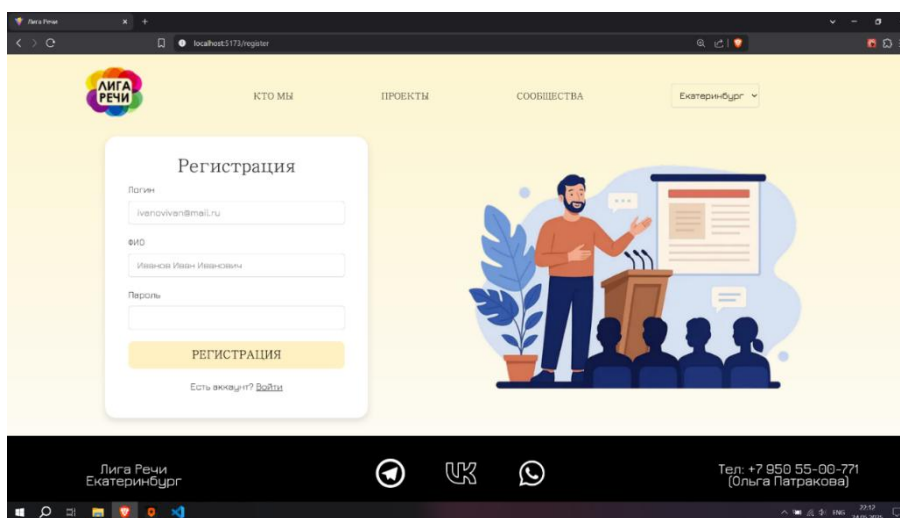


Рисунок 10 – аутентификация, страница регистрации

Личный кабинет пользователя включает следующие функциональные модули:

- Донаты – отслеживание своего уровня в зависимости от пожертвованной суммы;
- Мой уровень – отображение статуса пользователя;
- Обо мне – просмотр и редактирование персональных данных (рисунок 11);
- Портфолио – отслеживание мероприятий, в которых участвовал пользователь;
- Мои проекты – список текущих проектов;
- Участники – просмотр других пользователей системы;
- Мероприятия – календарь событий и управление участием;

Каждый раздел обеспечивает пользователей возможностью просмотра, редактирования и взаимодействия с данными в зависимости от их роли (например, сотрудники организации имеют расширенные права доступа).

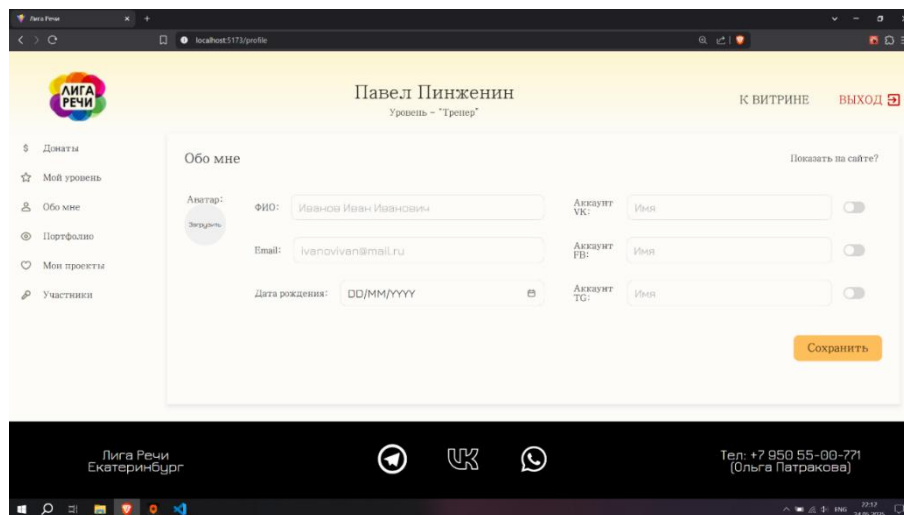


Рисунок 11 – изображение личного кабинета, раздел «Обо мне»

### 1.3.2 Тестирование и отладка

На протяжении разработки проводилось тестирование фронтенд-части, включающее:

- Проверку корректности API-запросов (обработка ошибок, валидация данных);
- Адаптивность интерфейса на различных устройствах и разрешениях экрана;
- Визуальную согласованность элементов интерфейса;

Основные выявленные и устранённые проблемы:

- Ошибки взаимодействия с API (некорректная обработка ответов сервера);
- Визуальные артефакты при отображении на мобильных устройствах;
- Оптимизация перерисовки компонентов для повышения производительности;

Разработанный интерфейс полностью соответствует поставленным требованиям, обеспечивая удобство использования и стабильную работу. Применение TypeScript и модульной структуры проекта позволяет легко масштабировать функционал в будущем.

## 1.4 Backend-разработка

### 1.4.1 Архитектура продукта

Система разделена на три основные части:

- Витрина сайта (публичная часть): интерфейс для пользователей с блоками: хэдер, баннер, контакты, календарь, о нас, клубы-резиденты, донаты, будем партнерами, партнеры, команда, футер.
- Административная панель сайта: панель управления для администратора с возможностью редактирования контента, управления каталогами, пользователями и настройками.
- Аккаунты участников: личные кабинеты с ролевой моделью (3 уровня: Гость, Активист, Тренер).

Архитектура программного продукта состоит из нескольких взаимосвязанных компонентов (рисунок 12).

С витрины сайта отправляются API-запросы к Strapi для получения данных о контенте блоков. Личный кабинет взаимодействует с Strapi через аутентифицированные запросы с использованием JWT-токенов.

Бэкенд Strapi является ядром системы. Он принимает запросы от витрины, личного кабинета и административной панели. Предоставляет встроенные CRUD-операции для моделей, а также позволяет изменять API-эндпоинты для реализации собственной логики.

Админ-панель позволяет управлять контентом через предоставляемый Strapi пользовательский интерфейс. Она взаимодействует с бэкендом через API-запросы.

В PostgreSQL хранятся данные. Выбрана эта база данных, так как она реляционная и позволяет хранить сложные связи между сущностями.

Процесс аутентификации происходит прямо на сервере. Генерируется JWT токен, позволяющий однозначно идентифицировать пользователя. Также есть возможность назначать политики доступа и управлять разрешениями.

С помощью upload provider обеспечивается хранения медиафайлов. На стороннем сервере хранятся медиафайлы, загружаемые через личный кабинет либо административную панель. С помощью API, предоставляемого Strapi, обеспечивается взаимодействие с провайдером.

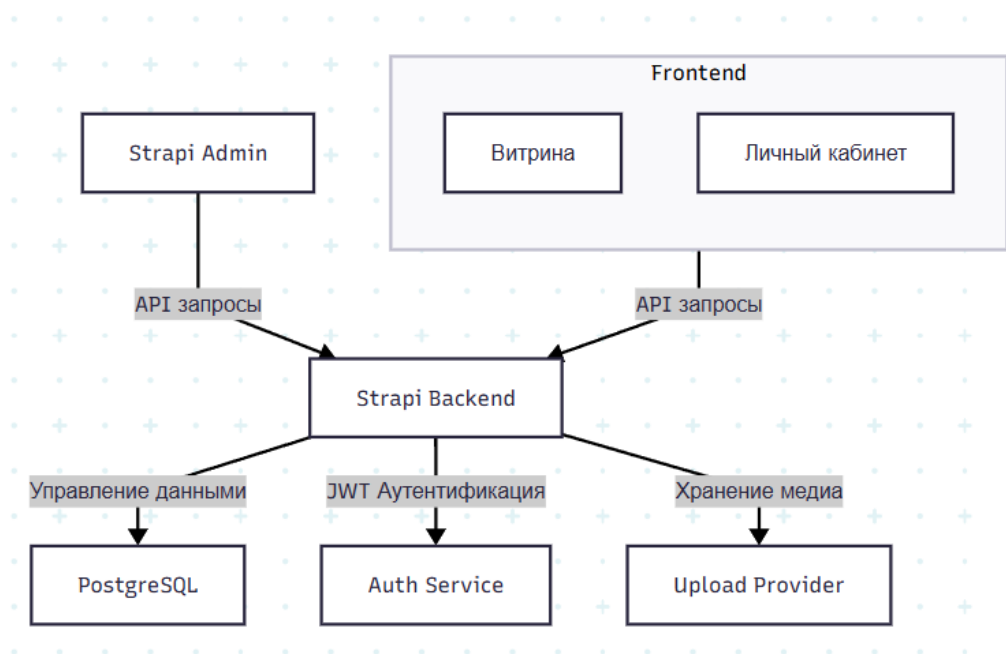


Рисунок 12 – Компоненты архитектуры сайта

Выбор архитектурного решения обусловлен следующим:

- Strapi предоставляет готовую админ-панель и реализацию CRUD-операций, что ускоряет создание функционала для управления контентом и позволяет сконцентрировать внимание на разработке более сложной бизнес-логики в условиях, когда требуется разработать много компонентов системы. Возможность кастомизации API через контроллеры, сервисы и middleware позволяет адаптировать систему под специфичные требования. JWT-авторизация и ролевая модель (RBAC) позволяют предоставить безопасное решение. В Strapi есть интеграция с PostgreSQL, провайдерами загрузки файлов. При этом клиентскую часть приложения (витрина и личный кабинет) можно разрабатывать с использованием отдельного фреймворка

- PostgreSQL позволяет реализовывать сложные запросы и сложные связи между сущностями
- Загрузка медиа на отдельный сервер снижает нагрузку на основной бэкенд-сервер и базу данных. Также это позволяет хранить большие объемы данных.

#### **1.4.2 Тестирование и разбор ошибок**

Разработка бэкенда происходила следующим образом:

- 1) создавались модели данных и реализовывались связи между ними.
- 2) эти модели использовались в реализации бизнес-логики внутри сервисов и контроллеров;
- 3) настраивались права доступа к определенным эндпоинтам;
- 4) проводилась настройка пользовательского интерфейса административной панели: например, отображение полей сущностей, сортировка сущностей в списке.

В процессе разработки бэкенда проводилось тестирование. При тестировании эндпоинта для получения контента витрины было обнаружено, что ответ на запрос приходится ожидать довольно долго – примерно 1 секунду, поскольку для реализации логики приходилось делать несколько запросов к базе данных. Получение контента витрины было оптимизировано на бэкенде с помощью одновременных асинхронных запросов к базе данных [9].

Также при тестировании эндпоинтов обнаружилось, что relation поля (поля, определяющие связи между сущностями) и их содержание по умолчанию не отображается в API-ответах. Поэтому пришлось использовать параметр populate при запросах к базе данных.

При тестировании была обнаружена проблема получения доступа к контенту витрины, если не был произведен вход в систему. Поэтому был настроен доступ с помощью встроенной роли Public, применяемой ко всем

входящим запросам, которые не содержат заголовка Authorization с действительным JWT-токеном.

При тестировании добавления пользователей внутри административной панели обнаружилось, что невозможно однозначно идентифицировать пользователя при назначении лидера сообщества или проекта. Для решения проблемы было добавлено поле fio, позволяющее сочетать в себе поля для имени, фамилии и отчества. При создании пользователя или редактировании его имени/фамилии/отчества автоматически изменяется и содержимое поля fio.

Также при тестировании обнаружено, что при создании пользователя и добавлении ему роли в интерфейсе административной панели отображается предупреждение о том, что есть проблемы с политикой доступа. Сущность пользователя в итоге создается с нужной ролью. Однако такое предупреждение может вызывать вопросы у администратора панели и вводить в заблуждение. В интерфейсе административной панели также иногда появляется ошибка запроса к API, поэтому приходится перезагружать страницу в таком случае, что тоже может вызывать неудобство у конечного пользователя.

## **1.5 Организация работы**

В рамках проектной деятельности обязанности тимлида включали в себя:

- организацию командной работы и контроль выполнения задач;
- планирование этапов разработки и определение приоритетов задач;
- проведение регулярных командных созвонов для синхронизации участников;
- распределение задач между членами команды с учётом их специализаций и текущей загрузки;

- взаимодействие с заказчиком: уточнение требований, сбор обратной связи, внесение корректировок в процесс разработки;
- ведение общего списка задач и контроль сроков их выполнения;
- общее соблюдение сроков проекта;
- контроль за выполнением ключевых этапов.

Для организации работы команды использовалась методология Scrum с 1-2-недельными спринтами. По окончании спринтов проходили онлайн-встречи с командой. На встречах обсуждались итоги прошедшего спринта, проблемы, которые возникли, новые требования, если были. Также на встречах обсуждался следующий спринт: распределялись рабочие задачи и устанавливались сроки их выполнения. Помимо онлайн-созвонов для коммуникации использовался чат в Telegram, где можно было оперативно решить появившиеся вопросы.

На начальном этапе мы провели анализ требований и приоритизацию задач по методике MoSCoW (Must, Should, Could, Won't). Это позволило нам сфокусироваться на наиболее критичных функциях и сформировать предварительное видение продукта.

Был выбран подход Agile и использован Scrum-подход к планированию. Работа велась в спринтах продолжительностью 1–2 недели. В начале каждого спринта проводилось планирование, где определялись приоритетные задачи и назначались ответственные. В конце спринта — ревью с демонстрацией результатов и ретроспектива.

Аналитик занимался сбором и формализацией требований от заказчика. На основе этой информации дизайнер разрабатывал интерфейсы и подготавливал макеты. Бэкенд-разработчик занимался проектированием базы данных, серверной частью и разработкой эндпоинтов. После утверждения дизайна, фронтенд-разработчик приступил к реализации части.

Разработчики связывались друг с другом и дизайнером, чтобы не было разрыва между пониманием технической реализации.

Таблица 2 – Бэклог продукта по методу MoSCoW

ID	Задача	Приоритет	Статус
1	Анализ Аналогов	Should	Выполнено
2	Составление ТЗ	Must	Выполнено
3	Проектирование базы данных и сущностей	Must	Выполнено
4	Создание дизайна	Must	Выполнено
5	Регистрация и статус аккаунта	Must	Выполнено
6	Админская панель	Must	Выполнено
7	Страница "Кто мы»	Must	Выполнено
8	Раздел "Проекты»	Must	Выполнено
9	Раздел "Сообщества»	Must	Выполнено
10	Раздел "Витрина»	Must	Выполнено
12	Раздел «Клубы-резиденты»	Must	Выполнено
13	Команда проекта	Must	Выполнено
14	Личный кабинет	Must	Выполнено
15	Донаты (без оплаты, только форма)	Could	Выполнено
16	Календарь мероприятий (общий)	Must	Выполнено
17	Раздел «Партнеры»	Must	Выполнено
18	Портфолио пользователя	Must	Выполнено



## ЗАКЛЮЧЕНИЕ

Разработанный портал "Лига Речи" соответствует большинству ключевых требований, заявленных заказчиком. Были успешно реализованы:

- Основной функционал для пользователей: регистрация, личный кабинет, запись на мероприятия, просмотр галереи и проектов;
- Административные возможности: управление контентом, пользователями и мероприятиями через панель администратора;
- Адаптивный дизайн: обеспечена корректная работа на различных устройствах;

Однако некоторые второстепенные функции (например, полноценная интеграция платежей для донатов) были упрощены или отложены из-за ограниченных сроков. Тем не менее, продукт полностью готов к использованию и служит надежной основой для дальнейшего развития.

Тестирование выявило ряд проблем, которые были успешно устранены:

- Оптимизация API: уменьшено время ответа сервера за счет асинхронных запросов;
- Исправление ошибок доступа: настроены права для публичных ролей и авторизованных пользователей;
- Улучшение интерфейса: устранены визуальные артефакты на мобильных устройствах;

Критические дефекты, влияющие на работоспособность, отсутствуют. Однако остаются незначительные недочеты в административной панели (например, ложные предупреждения при создании пользователей), которые требуют доработки.

Для дальнейшего развития продукта рекомендуется:

- 1) расширение функционала:
  - реализация полноценной системы донатов с интеграцией платежных систем;

- добавление чата для взаимодействия участников и организаторов мероприятий;

2) оптимизация производительности:

- внедрение кэширования для часто запрашиваемых данных;

3) улучшение пользовательского опыта:

- добавление push-уведомлений о новых мероприятиях

- разработка мобильного приложения для большего удобства участников.

Продукт демонстрирует высокую готовность к эксплуатации и обладает значительным потенциалом для масштабирования. Команда рекомендует сосредоточиться на доработке второстепенных функций и активном внедрении портала среди целевой аудитории.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Лига Речи : сайт – URL: <https://vk.com/oratoriada>
2. Лига Речи ЕКБ – Telegram : сайт – URL: <https://t.me/oratoriada>
3. Лига Речи – Главная страница проекта: сайт – URL: <https://лига-речи.рф/>
4. React : сайт – URL: <https://react.dev/>
5. TypeScript: JavaScript With Syntax For Types : сайт – URL: <https://www.typescriptlang.org/>
6. Axios : сайт – URL: <https://axios-http.com/>
7. React Router Official Documentation : сайт – URL: <https://reactrouter.com/>
8. Gravity UI : сайт – URL: <https://gravity-ui.com/>
9. Strapi 5 Documentation : сайт – URL: <https://docs.strapi.io/cms/>

# ПРИЛОЖЕНИЕ А

## (обязательное)

### Диаграмма бизнес-процессов

