

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»**

Кафедра математической теории упругости и биомеханики

**Разработка веб-сервиса для организации благотворительных
мероприятий**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 442 группы

направление 09.03.03 – Прикладная информатика

механико-математического факультета

Тарантина Михаила Олеговича

Научный руководитель
доцент, к.ю.н.

Р.В. Амелин

Зав. кафедрой
зав. кафедрой, д.ф.-м.н., профессор

Л.Ю. Коссович

Саратов 2025

Введение. Современный этап развития цифровых технологий характеризуется активным внедрением специализированных веб-решений в сферу благотворительности. Несмотря на рост количества социальных инициатив, анализ существующих платформ выявляет значительный разрыв между потребностями организаторов благотворительных мероприятий и функциональными возможностями доступных инструментов. Большинство решений ориентированы на коммерческие события и не учитывают специфику благотворительной деятельности, такую как работа с волонтерами, обработка пожертвований и обеспечение прозрачной отчетности.

Актуальность исследования обусловлена необходимостью создания специализированного веб-сервиса, способного комплексно автоматизировать процессы планирования, проведения и мониторинга благотворительных мероприятий. Новизна работы заключается в разработке интегрированного решения, объединяющего управление событиями, сбор средств и координацию волонтеров с учетом требований прозрачности и удобства взаимодействия для всех участников.

Целью настоящей работы является разработка веб-сервиса для организации благотворительных мероприятий, который позволит упростить процесс создания мероприятий, привлечения участников и сбора средств.

Для достижения заявленной цели были поставлены следующие **задачи**:

- провести анализ существующих решений в области организации благотворительных мероприятий;
- определить функциональные и нефункциональные требования к веб-сервису;
- спроектировать структуру веб-приложения и определить технологический стек для реализации пользовательского интерфейса и серверной части;
- реализовать основные функции платформы, включая создание мероприятий, сбор средств и удобный поиск ближайших или интересующих мероприятий;

- провести тестирование разработанного решения.

Структура и объём работы. Бакалаврская работа включает в себя введение, три раздела основной части, заключение, приложения и содержит 56 страниц. Список использованных источников включает 20 наименований.

Раздел 1. Аналитическая часть.

Раздел 2. Проектная часть.

Раздел 3. Реализация приложения.

Основное содержание работы. Работа посвящена разработке веб-сервиса для организации благотворительных мероприятий, который позволит упростить процесс создания мероприятий, привлечения участников и сбора средств.

Во введении представлены основные сведения о теме бакалаврской работы, обоснована её актуальность, сформулированы цель и задачи исследования.

Первый раздел содержит анализ предметной области.

В результате проведенного анализа предметной области были выявлены системные проблемы организации благотворительных мероприятий, обусловленные комплексным характером данной деятельности, сочетающей элементы event-менеджмента, фандрайзинга и волонтерского управления. Научное исследование показало, что основными препятствиями являются технологическая разобщенность существующих решений и несоответствие их функциональных возможностей специфическим требованиям благотворительного сектора. Согласно данным проведенного в 2024 году исследования, около 70% организаторов сталкиваются с отсутствием комплексных платформ, способных обеспечить полный цикл управления благотворительными мероприятиями.

Критический анализ современных цифровых решений позволил выделить три основные категории платформ, каждая из которых обладает существенными ограничениями применительно к благотворительной деятельности. Универсальные системы управления мероприятиями, такие как Eventbrite и TimePad, демонстрируют недостаточную адаптацию к особенностям

некоммерческого сектора. Специализированные фандрайзинговые платформы, включая GoFundMe и Planeta.ru, фокусируются исключительно на сборе средств, игнорируя другие аспекты организации мероприятий. Комплексные CRM-решения для некоммерческих организаций, например Bloomerang, отличаются избыточной сложностью интерфейсов и отсутствием специализированных инструментов для управления событиями.

Технологический анализ выявил, что разрабатываемый веб-сервис должен удовлетворять комплексным требованиям, которые можно разделить на функциональные и нефункциональные. К числу ключевых функциональных требований относится обеспечение полного цикла управления мероприятиями, включая создание событий, обработку пожертвований, координацию волонтеров и генерацию отчетности. Особое внимание уделяется интеграции с платежными системами и обеспечению прозрачности финансовых потоков. Нефункциональные требования включают обеспечение высокой производительности при пиковых нагрузках, соответствие стандартам информационной безопасности и создание доступного пользовательского интерфейса.

Предлагаемое решение призвано устранить существующие пробелы в нише цифровых инструментов для благотворительности.

Во втором разделе описан процесс проектирования веб-приложения, включающий:

- выбор технологий;
- проектирование архитектуры;
- прототипирование интерфейса.

Веб-сервис для организации благотворительных мероприятий разработан с применением современных технологий, обеспечивающих высокую производительность, масштабируемость и удобство разработки.

Для клиентской части веб-сервиса были подобраны современные технологии, обеспечивающие высокую производительность, отзывчивость интерфейса и удобство сопровождения кодовой базы. В качестве базового

фреймворка был выбран Next.js, представляющий собой расширенную реализацию библиотеки React с поддержкой серверного рендеринга (SSR) и статической генерации страниц (SSG). Данный выбор обусловлен необходимостью обеспечения быстрой первоначальной загрузки интерфейса и соблюдения требований поисковой оптимизации, что особенно важно для платформы, ориентированной на широкий круг пользователей.

Использование TypeScript в качестве основного языка разработки позволит существенно повысить надежность кодовой базы за счет статической типизации. Статический анализ типов обеспечивает раннее выявление потенциальных ошибок на этапе компиляции, улучшает читаемость кода и облегчает процесс рефакторинга, что критически важно для долгосрочной поддержки проекта. Интеграция TypeScript с экосистемой React/Next.js обеспечивает плавный процесс разработки благодаря расширенным возможностям современных IDE.

Архитектура управления состоянием приложения будет построена на комбинации локального состояния компонентов и глобального хранилища, реализованного с помощью библиотеки Zustand. Данное решение обеспечивает оптимальный баланс между производительностью и удобством разработки, позволяя эффективно управлять общими данными, доступными из различных частей приложения. Минималистичный API Zustand снижает порог вхождения для новых разработчиков и уменьшает количество шаблонного кода.

Для организации взаимодействия с серверным API выбрана библиотека TanStack Query, предоставляющая комплексные инструменты для управления асинхронными операциями. Данное решение автоматизирует такие аспекты работы с данными, как кэширование, фоновое обновление, обработка ошибок и управление состоянием загрузки. TanStack Query позволит существенно сократить объем шаблонного кода и сосредоточиться на реализации бизнес-логики приложения.

В качестве базового фреймворка для разработки серверной компоненты рассматривается NestJS - прогрессивная платформа для создания масштабируемых серверных приложений на Node.js. Выбор данного решения

обусловлен его архитектурными преимуществами, включающими модульную организацию кода, поддержку принципов объектно-ориентированного программирования и паттерна Dependency Injection. Важным фактором является встроенная поддержка TypeScript, обеспечивающая статическую типизацию и повышающая надежность кодовой базы. NestJS предоставляет комплексный набор инструментов для реализации типовых backend-задач, таких как работа с базами данных, аутентификация, валидация входных данных и обработка ошибок.

Для хранения и управления данными системы планируется использование реляционной СУБД PostgreSQL. Данный выбор объясняется требованиями к надежному хранению структурированных данных о мероприятиях, пользователях и финансовых операциях. PostgreSQL обеспечивает поддержку транзакций ACID, выполнение сложных аналитических запросов и имеет развитые механизмы обеспечения целостности данных. Особое значение имеет способность СУБД эффективно работать с большими объемами информации, что соответствует потенциальным потребностям масштабируемой платформы.

Система построена по трехуровневой архитектуре, включающей уровень представления (Next.js), уровень бизнес-логики (NestJS) и уровень данных (PostgreSQL). Frontend будет реализован с применением Feature-Sliced Design для модульности, backend следует принципам чистой архитектуры.

Логическая модель данных включает ключевые сущности: Пользователи, Мероприятия, Пожертвования, Отчеты, Платежные системы и Аналитика. Между сущностями установлены связи, обеспечивающие целостность данных. Физическая реализация в PostgreSQL включает 6 нормализованных таблиц с соответствующими связями через внешние ключи.

База данных спроектирована с соблюдением принципов нормализации, что минимизирует избыточность и обеспечивает согласованность данных. Реализованы механизмы транзакций, триггеров и ограничений для поддержания целостности. Выбор PostgreSQL обусловлен требованиями к надежности, производительности и поддержке сложных запросов.

Пользовательский интерфейс разрабатываемого веб-сервиса спроектирован с учетом принципов user-centered design, что обеспечивает его соответствие потребностям целевых пользовательских групп. Визуальная составляющая интерфейса базируется на концепции эмоционального дизайна, где продуманное сочетание нейтральной цветовой палитры и акцентных элементов способствует созданию доверительной атмосферы и стимулирует социальную активность участников благотворительных мероприятий.

Архитектура интерфейса включает комплекс взаимосвязанных компонентов, обеспечивающих полноценное взаимодействие пользователей с системой. Карточки мероприятий с индикаторами прогресса сбора средств позволяют быстро оценить актуальность и востребованность благотворительных инициатив. Формы ввода данных оснащены механизмами контекстной валидации, что минимизирует вероятность ошибок при заполнении. Навигационные элементы, включая адаптивные меню и систему "хлебных крошек", обеспечивают интуитивно понятное перемещение между разделами сервиса.

Функциональные страницы интерфейса организованы в соответствии с ключевыми пользовательскими сценариями. Главная страница интегрирует три основных информационных блока: перечень актуальных мероприятий, интерактивную карту их расположения и ленту новостей благотворительной сферы. Детализированное представление каждого мероприятия содержит исчерпывающее описание, инструментарий для осуществления пожертвований и наглядную визуализацию процесса сбора средств. Специализированные интерфейсы для организаторов предусматривают поэтапный ввод данных о новых мероприятиях с автоматической проверкой корректности заполнения полей.

Адаптивность интерфейса достигается за счет применения различных стратегий компоновки элементов в зависимости от типа устройства. В десктопной версии используется сеточная организация пространства, позволяющая оптимально расположить многочисленные информационные

блоки. Мобильная версия интерфейса построена на принципах вертикального скроллинга с упрощенными элементами управления, что обеспечивает комфортное взаимодействие на устройствах с ограниченным экранным пространством.

Особое внимание при проектировании интерфейса уделено соответствию международным стандартам доступности WCAG 2.1 уровня AA. Это гарантирует возможность использования сервиса людьми с различными формами инвалидности. Минимизация когнитивной нагрузки достигается за счет продуманной информационной архитектуры, последовательного визуального языка и сокращения количества действий, необходимых для выполнения целевых операций.

Третий раздел содержит описание реализации веб-сервиса.

Серверный компонент системы реализован с использованием облачной платформы Supabase, представляющей современное решение для разработки backend-приложений. Выбор данной технологии обусловлен комплексом преимуществ, включая предоставление полнофункциональной базы данных PostgreSQL с расширенными возможностями управления, а также встроенным REST API интерфейсом, что существенно ускоряет процессы разработки и интеграции.

При проектировании структуры базы данных особое внимание уделено вопросам нормализации отношений между сущностями. Основная таблица Events спроектирована с учетом всех требований предметной области и содержит полный набор полей, обеспечивающих исчерпывающее описание благотворительных мероприятий. Помимо базовых идентификаторов и описательных полей, таблица включает временные параметры, географические координаты, контактные данные организатора, а также ссылку на визуальное представление мероприятия.

Для обеспечения целостности данных реализована многоуровневая система валидации, включающая проверку форматов электронной почты и телефонных номеров.

Клиентская часть веб-приложения разработана с применением современного технологического стека, включающего библиотеку React и инструмент сборки Vite. Выбор React обусловлен его компонентной архитектурой, позволяющей создавать масштабируемые и поддерживаемые пользовательские интерфейсы. Инструмент сборки Vite выбран благодаря высокой производительности и полной поддержке современных стандартов ECMAScript.

Архитектура проекта построена по модульному принципу, обеспечивающему четкое разделение функциональных модулей. Каждый компонент системы выделен в автономный модуль с четко определенными интерфейсами взаимодействия. Ключевые компоненты интерфейса включают элементы отображения карточки мероприятия, комплексную форму создания мероприятия с поддержкой валидации, а также компонент интеграции с картографическим сервисом Yandex Maps.

Управление состоянием приложения осуществляется посредством библиотеки TanStack Query, предоставляющей эффективные механизмы кэширования данных и управления асинхронными запросами. Данный подход позволил существенно сократить количество обращений к серверу и значительно улучшить пользовательский опыт.

Процесс интеграции фронтенд и бэкенд компонентов системы реализован через REST API Supabase с использованием официального JavaScript клиента. Для обеспечения типобезопасности на всех уровнях приложения разработаны специализированные TypeScript-интерфейсы, полностью соответствующие структуре данных Supabase.

Особое внимание при интеграции уделено тщательной обработке ошибок через многоуровневую систему обработки исключений, реализации индикации процессов загрузки данных, а также оптимизации запросов с применением передовых методов кэширования.

Для обеспечения актуальности данных в режиме реального времени задействован механизм подписок Supabase, позволяющий мгновенно отражать

изменения в пользовательском интерфейсе. Взаимодействие с Yandex Maps API организовано через специализированный сервис-обертку, обеспечивающий необходимый уровень абстракции от конкретной реализации картографического сервиса.

Процедура тестирования системы организована на нескольких уровнях, что позволило гарантировать высокие показатели качества конечного продукта. Модульное тестирование UI-компонентов выполнено с использованием связки Vitest и React Testing Library, что обеспечило стопроцентное покрытие тестами ключевых элементов интерфейса.

Особый акцент при тестировании сделан на всесторонней проверке бизнес-логики форм, включая комплексные тесты валидации, анализ всех возможных состояний приложения и сценариев взаимодействия, а также тестирование граничных случаев и нестандартных ситуаций.

Ручное тестирование включало исчерпывающую проверку пользовательских сценариев на различных устройствах и в разных браузерах. Проведено детальное тестирование производительности с тщательным анализом временных характеристик загрузки и отклика интерфейса. Для API endpoints выполнена полномасштабная проверка обработки граничных случаев и ошибок.

Результаты проведенных испытаний подтвердили полное соответствие системы всем заявленным требованиям по надежности, производительности и удобству эксплуатации. Достигнутые показатели эффективности позволяют рекомендовать данное решение для промышленного внедрения.

Заключение. Проведенное исследование подтвердило возможность создания специализированного веб-сервиса для организации благотворительных мероприятий. Разработанный прототип, реализованный на базе React, TypeScript и Supabase, демонстрирует работоспособность ключевых функций, включая интеграцию с платежной системой ЮKassa.

Теоретическая значимость работы заключается в систематизации требований к подобным платформам и разработке подходов к проектированию пользовательских интерфейсов для социально ориентированных сервисов.

Практическая реализация подтвердила эффективность выбранного технологического стека на начальном этапе разработки.

Перспективные направления развития включают расширение функциональности (систему авторизации, механизмы управления мероприятиями), оптимизацию производительности и проведение юзабилити-тестирования. Полученные результаты демонстрируют потенциал цифровых технологий для поддержки благотворительных инициатив и создают основу для дальнейших исследований в области социально ориентированных IT-решений.