

Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования «Уральский федеральный университет имени первого Президента России Б. Н. Ельцина» (УрФУ)

Институт радиоэлектроники и информационных технологий – РТФ

ОТЧЕТ

о проектной работе

по теме: «EventMaster – Умное управление вашими мероприятиями 2025: автоматизация учёта и регистрации участников мероприятий в студенческом кампусе НВК»

по дисциплине: Проектный практикум 1

Команда: Nexa

Тимлид, Бэкенд-разработчик:

Насибулин Данила Евгеньевич РИ-140943

Аналитик, Фронтенд-разработчик:

Артемьева Дарья Владиславовна РИ-140944

Бэкенд-разработчик, Дизайнер:

Ядрышников Данил Алексеевич РИ-140943

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc199611643)

[1. Целевая аудитория 5](#_Toc199611644)

[2. Определение проблемы 7](#_Toc199611645)

[3. Подходы к решению проблемы 10](#_Toc199611646)

[4. Анализ аналогов 14](#_Toc199611647)

[5. Календарный план проекта 17](#_Toc199611648)

[6. Сценарии использования 20](#_Toc199611649)

[7. Требования к продукту и к MVP 23](#_Toc199611650)

[8. Стек для разработки 27](#_Toc199611651)

[9. Прототипирование 30](#_Toc199611652)

[10. Проектирование и разработка системы 32](#_Toc199611653)

[ЗАКЛЮЧЕНИЕ 35](#_Toc199611654)

[СПИСОК ЛИТЕРАТУРЫ 39](#_Toc199611655)

[ПРИЛОЖЕНИЕ А – ПРОТОТИПЫ И ВИЗУАЛЬНЫЕ ПРЕДСТАВЛЕНИЯ 40](#_Toc199611656)

[ПРИЛОЖЕНИЕ Б – ГРАФИКИ И СРАВНИТЕЛЬНЫЕ СХЕМЫ 41](#_Toc199611657)

[ПРИЛОЖЕНИЕ В – РЕАЛИЗОВАННЫЙ ИНТЕРФЕЙС MVP ВЕБ-ПРИЛОЖЕНИЯ «МОЙКАМПУС» 42](#_Toc199611658)

ВВЕДЕНИЕ

Проведение массовых мероприятий в студенческой среде всегда сопровождается множеством организационных сложностей, особенно в части регистрации участников и учета их посещаемости. На территории Новокольцовского кампуса УрФУ (далее — НВК) такие события проводятся регулярно: это и развлекательные мероприятия, и образовательные встречи, и торжественные события, связанные с жизнью студенческого сообщества. Несмотря на устойчивую традицию их проведения, сам процесс регистрации участников до сих пор остается малотехнологичным и крайне трудоемким. В большинстве случаев организаторы используют Google Forms и Excel-таблицы, вручную формируя списки и отмечая присутствующих. При этом возникают задержки на входе, очереди, человеческие ошибки и необходимость тратить дополнительное время на редактирование и отправку отчетов. Мы на собственном опыте убедились в масштабах этой проблемы, когда стояли в очереди на мероприятие более 40 минут, наблюдая, как участники замерзают на улице, а организаторы судорожно листают списки.

На сегодняшний день существует множество платформ для проведения мероприятий, таких как Timepad, Eventbrite, Qtickets и другие. Однако ни одна из них не ориентирована на студенческую инфраструктуру, в частности — на реалии жизни в кампусе НВК. Эти платформы либо избыточны по функциональности, либо ориентированы на коммерческое применение и не учитывают специфики вузовской среды: отсутствует интеграция с системами внутренней отчетности, не поддерживается адаптивная и простая модель участия, и, что особенно важно — они не учитывают нехватку ресурсов и времени у студенческих организаторов. Существующее приложение «Молодежь NMSTU», хоть и решает схожие задачи, работает исключительно в рамках одного вуза и не может быть использовано в УрФУ.

Таким образом, несмотря на наличие аналогов, ниша простых, локализованных решений для внутрикампусных мероприятий остается слабо проработанной. Это делает задачу создания специализированного веб-сервиса как никогда актуальной. Сервис, который бы учитывал особенности локальных мероприятий, не требовал от участников и организаторов лишних действий, и позволял бы за считанные минуты организовать регистрацию, учёт и экспорт отчетов — является необходимым и востребованным решением.

Целью нашего проекта является разработка удобного веб-приложения для автоматизации процессов регистрации и учета посещаемости мероприятий, проводимых в Новокольцовском кампусе УрФУ. Сервис должен устранить ручной труд, сократить время на организационные действия, исключить очереди на входе и предоставить организаторам готовые отчеты в удобном формате.

Для достижения поставленной цели необходимо решить следующие задачи:

* Изучить и описать текущий процесс регистрации и учета участников мероприятий в кампусе НВК;
* Выявить основные проблемы и трудозатраты организаторов и участников при проведении мероприятий;
* Проанализировать существующие решения и определить, какие из функций могут быть адаптированы для нужд НВК;
* Разработать веб-приложение, включающее в себя создание мероприятий, регистрацию участников и сканирование QR-кодов;
* Составить систему хранения и экспорта данных, позволяющую формировать готовые отчеты по мероприятиям.

Таким образом, данная работа направлена на создание продукта, который будет не просто цифровым инструментом, но и решением реальной, повседневной проблемы студентов и организаторов, значительно упрощающим жизнь студенческому сообществу кампуса.

1. Целевая аудитория

Проект «МойКампус» разрабатывался для решения конкретной и давно назревшей проблемы в рамках университетской среды, поэтому определение целевой аудитории не представляет трудности и не требует гипотетического моделирования — мы хорошо понимаем, кто наши пользователи, чего они хотят и почему их потребности до сих пор не были удовлетворены.

**Что (What?)** — наш продукт представляет собой веб-приложение, предназначенное для автоматизации процесса организации мероприятий: от создания события и регистрации участников до учёта посещаемости и формирования отчетов. Сервис предоставляет две основные функции: быструю регистрацию с генерацией персонального QR-кода и инструмент для сканирования этих кодов, что позволяет мгновенно фиксировать посещение и автоматически собирать данные. Продукт не требует установки, работает на любом устройстве с камерой и не предполагает регистрации или создания личных кабинетов.

**Кто (Who?)** — целевая аудитория сервиса делится на две основные группы. Первая — это организаторы мероприятий: студенты, входящие в Союз студенческого кампуса НВК, кураторы направлений, а также администрация кампуса. Обычно это активные студенты от 18 до 22 лет, которые несут ответственность за проведение событий и нуждаются в технологических инструментах для упрощения своей работы. Вторая группа — участники мероприятий, то есть обычные студенты, проживающие в кампусе. Это молодые люди того же возраста, которым важно, чтобы регистрация была быстрой, понятной и не отнимала лишнего времени, особенно при участии в спонтанных или вечерних мероприятиях.

**Почему (Why?)** — мотивация у обеих групп разная, но объединяет их одно: усталость от ручной работы и желание сэкономить время. Организаторы стремятся избавиться от рутинных задач, таких как ручное составление списков, отметка явки и подготовка отчетов. В текущей системе они тратят до полутора часов только на учёт участников, а любой сбой или ошибка приводит к конфликтам и проблемам с отчётностью. Участники, в свою очередь, не хотят стоять в очередях и заполнять одни и те же анкеты по несколько раз. Они хотят зарегистрироваться за пару секунд — хоть за неделю до мероприятия, хоть на месте перед входом. Наш сервис решает обе проблемы, предлагая организаторам простой инструмент управления, а участникам — мгновенный способ зарегистрироваться.

**Когда (When?)** — необходимость в использовании сервиса возникает всякий раз, когда в кампусе проводится мероприятие, требующее регистрации участников и отчетности о посещении. Это может быть что угодно: праздник, образовательная встреча, собрание, тренинг. Особенно актуален наш инструмент в холодное время года, когда участники не могут долго ждать на улице, и при массовых мероприятиях (до 150 человек), где ручной учёт просто не справляется с потоком.

**Где (Where?)** — принятие решения о «покупке» (в данном случае — о применении сервиса) происходит внутри студенческих объединений, когда планируется событие и назначается ответственный за его проведение. Именно в этот момент организатор нуждается в инструменте, который упростит ему подготовку. Само взаимодействие с сервисом происходит через браузер: организаторы используют панель создания мероприятий и интерфейс сканирования, участники — форму регистрации. Таким образом, точками контакта становятся как внутренняя кухня подготовки мероприятия, так и его фактическое проведение на месте.

1. Определение проблемы

Проблема, которую решает наш проект, не теоретическая — она возникла из реального опыта и регулярно повторяется в жизни студентов и организаторов мероприятий Новокольцовского кампуса УрФУ. Мы не проводили масштабные маркетинговые исследования или массовые опросы, потому что в рамках студенческого проекта важно быстро получить живую обратную связь от непосредственных пользователей. Вместо формализованных анкет мы пошли другим путем: пообщались напрямую с организаторами мероприятий, наблюдали за процессом регистрации, собирали жалобы, смотрели, как всё происходит на практике. Этот метод оказался куда более эффективным, потому что позволил не только услышать о проблемах, но и почувствовать их — увидеть усталость организаторов, ощутить дискомфорт участников, понять, насколько утомительна и неэффективна текущая система учёта.

Основная боль, с которой сталкиваются студенты-организаторы, заключается в огромных временных затратах на подготовку и проведение мероприятий. Сегодня весь процесс построен вручную: сначала создаётся Google-форма, затем скачиваются ответы, формируется таблица, в неё вручную вносятся правки, а во время самого мероприятия участники называют свои данные, а организаторы ищут их в списках и вручную отмечают посещение. Такая система требует участия нескольких человек, часто приводит к ошибкам, а после завершения мероприятия снова требует ручной обработки таблиц, чтобы подготовить отчёт. Этот процесс наглядно показан в **Приложении Б, рисунок 3**, где отображён действующий порядок регистрации и отметки. Для сравнения, схема с использованием нашей системы приведена на **рисунке 4**, демонстрируя значительное упрощение всех этапов. В среднем на одно массовое мероприятие организаторы тратят до 85 минут только на процесс регистрации и учёта. И это не считая подготовки формы, общения с участниками и рассылки ссылок.

С другой стороны, участники — обычные студенты — сталкиваются с очередями, задержками на входе, необходимостью заполнять одни и те же поля каждый раз заново. Это особенно раздражает в холодное время года или при спонтанной регистрации, когда хочется просто подойти к мероприятию и пройти внутрь без лишних действий. Для многих участие в событии начинается с долгого ожидания и неудобства — а значит, создаётся отрицательный опыт, который демотивирует студентов посещать мероприятия в будущем.

Мы также изучили конкурентов: такие решения как Timepad, Qtickets, Eventbrite и даже специализированные университетские платформы вроде «Молодежь NMSTU». Несмотря на их функциональность, все они ориентированы либо на коммерческие мероприятия, либо жёстко привязаны к конкретным вузам, как в случае с МГТУ. Их интерфейсы перегружены, в них много лишнего, а в ряде случаев даже требуется авторизация или мобильное приложение. Это делает их непригодными для оперативной, неформальной студенческой среды кампуса НВК.

Если взглянуть на проблему с точки зрения уровней «боли», которые описаны в маркетинговом подходе, можно чётко увидеть все три:

* На **техническом уровне** — неудобство ручного учета, медленная работа с таблицами, постоянные сбои и ошибки.
* На уровне **влияния на ресурсы** — затраты времени и сил, потеря мотивации как у участников, так и у организаторов, снижение качества мероприятий.
* На уровне **личной мотивации** — организаторы хотят, чтобы их труд был эффективным и заметным, а участники — чтобы мероприятие было удобным и приятным с первых секунд.

Таким образом, выявленная проблема не только объективна, но и глубоко ощущается всеми сторонами: и теми, кто организует мероприятия, и теми, кто их посещает. Это делает её актуальной, понятной и требующей незамедлительного решения — именно таким решением и является наш проект.

1. Подходы к решению проблемы

Работая над проектом, мы постарались подойти к проблеме не как к узкой технической задаче, а как к системной и повторяющейся ситуации, в которой есть разные стороны, интересы и ограничения. Основным этапом в формировании решения стало понимание текущего процесса: как именно организуются мероприятия, как создаются формы регистрации, как заполняются таблицы и как происходит отметка участников. Мы опирались не на теоретические источники, а на реальную обратную связь от студентов и организаторов в кампусе, на личный опыт и живые наблюдения. Это позволило не просто обозначить проблему, а почувствовать, насколько она масштабна и системна.

На ранних этапах работы мы провели обсуждения внутри команды в формате мозгового штурма, где каждый участник предлагал идеи того, как можно улучшить текущий процесс. Мы рассматривали разные сценарии, в том числе и те, которые в итоге были отвергнуты, но всё равно помогли нам лучше понять ограничения и риски. Например, изначально обсуждалась идея создать Telegram-бота, который бы обрабатывал регистрацию и генерировал коды. Такой формат выглядел современно, но на практике оказался неудобным: не все студенты активно пользуются Telegram, а работа с ботами требует лишнего времени и может вызывать ошибки. Также возникали сложности с безопасностью и поддержкой интерфейса на всех типах устройств. Поэтому идея была отложена как резервная.

Другим возможным решением было использование мобильного приложения. Мы оценили плюсы: постоянный доступ, возможность хранить историю посещений, интеграция с камерой. Но здесь проявился критический недостаток — установка. Большинство участников мероприятий сталкиваются с нашим продуктом один раз, максимум — несколько раз в семестр. Заставлять каждого скачивать отдельное приложение ради одноразовой регистрации оказалось бы избыточным и просто отпугнуло бы студентов. Мы также учли, что не все телефоны имеют свободную память или позволяют легко устанавливать сторонние приложения, особенно в условиях нестабильного интернет-соединения.

Мы рассматривали и более «классические» подходы: доработать Google Forms с автогенерацией кодов, интегрировать их с Excel или даже создать шаблоны макросов, чтобы ускорить ручной процесс. Однако все подобные решения оставались в рамках старой системы, которая требует слишком много ручного труда и не избавляет от главной проблемы — длительного времени на регистрацию и отметку участников.

Изучив существующие платформы для регистрации мероприятий, мы также поняли, что большинство из них перегружены функциональностью, не адаптированы под кампусную среду и требуют авторизации или оплаты. Это тоже стало аргументом в пользу собственного, минималистичного и специализированного решения.

Чтобы систематизировать рассмотренные идеи, мы провели сравнительный анализ всех подходов, выделив их сильные и слабые стороны. Результаты этого анализа приведены в таблице ниже:

Таблица 1 – Сравнение вариантов решений

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Вариант** | **Описание** | **Преимущества** | **Недостатки** | **Причина отклонения** |
| **Telegram-бот** | Регистрация и генерация QR-кодов через мессенджер | Быстрое взаимодействие, нет браузера | Не все пользуются Telegram, сложная поддержка интерфейса, возможные ошибки | Не универсален, ограниченный охват, сложность внедрения |
| **Мобильное приложение** | Установка приложения для участников и организаторов | Постоянный доступ, можно добавить функции | Требует установки, не все готовы скачивать, избыточно для разового использования | Участники не станут устанавливать ради одного события |
| **Доработка Google Forms** | Улучшение текущей схемы с Google Формами, макросы для Excel | Используется уже сейчас, не требует нового обучения | Всё равно остаётся ручной труд, сложно масштабировать | Не устраняет корневую проблему: медленный, ручной процесс |
| **Сторонние платформы (Timepad, Qtickets и др.)** | Использование существующих сервисов для мероприятий | Готовые решения, поддержка | Перегрузка функциями, не адаптированы под студенческую среду | Слишком сложны и громоздки для нужд НВК |
| **Веб-приложение без авторизации (наш выбор)** | Специализированный сайт для НВК: создание мероприятий, регистрация, QR, отчёты | Простота, доступ с любого устройства, мгновенная отметка, автоматизация | Требует первоначальной разработки | Оптимален по всем критериям: быстро, просто, удобно, без лишних действий |

На основе анализа всех альтернатив и их сопоставления с потребностями студентов и организаторов, мы пришли к оптимальному формату — **веб-приложению без авторизации**, доступному через ссылку, с возможностью:

* создать мероприятие за пару минут,
* опубликовать форму регистрации для участников,
* автоматически генерировать QR-коды,
* сканировать эти коды через камеру прямо на сайте,
* и, наконец, экспортировать готовую таблицу с отметками в Excel.

Такой подход оказался самым универсальным, удобным и реалистичным в условиях, в которых мы и наши пользователи работают каждый день. Он не требует установки, не ограничен устройствами, исключает ручной труд, снижает нагрузку на организаторов и сохраняет гибкость.

Возможность простого масштабирования (например, добавление ленты мероприятий, анкет для обратной связи, интеграции с календарём) также стала дополнительным преимуществом выбранной архитектуры. Она позволяет системе не просто решить проблему здесь и сейчас, но и развиваться вместе с потребностями студентов и организаторов.

1. Анализ аналогов

Анализ конкурентов был для нас важной частью на этапе раннего проектирования. Он помог понять, какие решения уже существуют на рынке, какие из них действительно работают с нашей целевой аудиторией, а какие — лишь формально похожи, но по сути решают другие задачи. Мы подошли к анализу с практической целью — не просто найти конкурентов, а выделить сильные стороны чужих решений, чтобы учесть их в разработке собственного продукта, и увидеть слабые, чтобы их избежать.

Первым делом мы определили цели анализа: понять, чем отличается наш подход, насколько он уникален, какие элементы можно переиспользовать и какие ошибки не стоит повторять. Далее мы составили список конкурентов, разделив их на **прямых** и **косвенных**. Прямыми конкурентами стали платформы, которые позволяют организовывать мероприятия, собирать участников и учитывать их присутствие — такие как **Timepad**, **Qtickets**, а также **Google Forms** в сочетании с Excel — именно это решение сейчас чаще всего используется организаторами в кампусе. Косвенными аналогами стали более специализированные решения для университетов, например, приложение **«Молодежь NMSTU»**, которое эффективно используется в МГТУ, но недоступно и нефункционально в рамках нашего университета.

В качестве критериев анализа мы выбрали:

* наличие QR-учета участников;
* адаптацию под образовательную среду;
* удобство интерфейса для студентов;
* наличие отчетности не только по зарегистрированным, но и по реально присутствующим;
* необходимость регистрации или установки приложения.

На основе этих критериев мы провели сравнительный анализ. Результаты представлены в таблице:

Таблица 2 – Анализ конкурентов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Критерии сравнения** | **Google Forms + Excel** | **Timepad** | **Qtickets** | **«Молодеж NMSTU»** |
| **QR-отметка** | нет | да | да | да |
| **Специализация под ВУЗ** | нет | нет | нет | да |
| **Удобство интерфейса** | нет | \*Перегружен | \*Перегружен | да |
| **Отчетность по пришедшим** | да  (вручную) | нет | нет | да |
| **Требует регистрацию/приложение** | нет | да | да | да |
| **Подходит для мероприятий в НВК** | \*\*Частично | нет | нет | нет  (недоступно) |

Примечание: «\*» — интерфейс перегружен функциями, неудобен для студентов; «\*\*» — частично применимо, но с большими ограничениями.

Анализ показал, что ни одно из существующих решений не подходит для задач, стоящих перед организаторами в Новокольцовском кампусе УрФУ. Все сторонние сервисы либо рассчитаны на коммерческие массовые мероприятия, либо работают только в рамках отдельных университетов, либо требуют слишком много ручной работы.

**Наше решение закрывает именно тот сегмент, который остался незанятым** — простая, доступная и не требующая установки система, предназначенная специально для внутренних студенческих мероприятий. Она не перегружена, не требует ни авторизации, ни установки, не привязана к конкретной структуре вуза, работает через браузер и обеспечивает быструю регистрацию, отметку и отчетность. Мы также представили визуально, как выглядит процесс учёта участников при использовании нашей системы: он становится линейным, быстрым и требует минимум действий со стороны организатора (см. **рисунок 4** в приложении А). Такой продукт на текущий момент отсутствует среди прямых и косвенных аналогов.

1. Календарный план проекта

Название проекта: МойКампус

Руководитель проекта: Насибулин Данила Евгеньевич

Таблица 3 – Календарный план реализации проекта

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№** | **Название этапа / задачи** | **Ответственный** | **Длительность** | **Дата начала** | **Временные рамки проекта** | | | |
| 1 нед | 2 нед | 3 нед | 4 нед |
| ***Анализ*** | | | | | | | | |
| *1.1* | *Определение проблемы* | Артемьева Дарья | 4 дня | 12.02 | X |  |  |  |
| *1.2* | *Выявление целевой аудитории* | Артемьева Дарья | 3 дня | 14.02 | X |  |  |  |
| *1.3* | *Анализ аналогов и конкурентов* | Артемьева Дарья | 5 дней | 16.02 | X |  |  |  |
| *1.4* | *Формирование концепта проекта* | Насибулин Данила | 5 дней | 16.02 | X |  |  |  |
| *1.5* | *Презентация аналитики* | Артемьева Дарья | 3 дня | 12.03 |  | X |  |  |
| *1.6* | *Создание ТЗ (черновик)* | Насибулин Данила | 5 дней | 13.03 |  | X |  |  |
| *1.7* | *Определение технологии и стека* | Насибулин Данила | 3 дня | 16.03 |  | X |  |  |
| *1.8* | *Итоговая доработка ТЗ* | Насибулин Данила | 2 дня | 22.03 |  |  | X |  |
| ***Проектирование*** | | | | | | | | |
| *2.1* | *Создание макетов интерфейсов в Figma* | Камилла Фишер | 6 дней | 22.03 |  |  | X |  |
| *2.2* | *Мастер-класс по Figma, корректировка макетов* | Камилла Фишер | 3 дня | 03.04 |  |  |  | X |
| *2.3* | *Подготовка архитектуры продукта* | Ядрышников Данил | 5 дней | 28.03 |  |  | X |  |
| *2.4* | *Создание карточки проекта* | Насибулин Данила | 1 день | 07.04 |  |  |  | X |
| ***Разработка*** | | | | | | | | |
| *3.1* | *Разработка backend (Flask + SQLite)* | Ядрышников Данил | 10 дней | 02.04 |  |  | X | X |
| *3.2* | *Разработка frontend (HTML, Bootstrap, JS)* | Артемьева Дарья | 10 дней | 02.04 |  |  | X | X |
| *3.3* | *Связка фронтенда и бэкенда* | Насибулин Данила | 5 дней | 13.04 |  |  |  | X |
| ***Внедрение и завершение*** | | | | | | | | |
| *4.1* | *Финализация MVP и тестирование* | Насибулин Данила | 5 дней | 18.04 |  |  |  | X |
| *4.2* | *Подготовка к неофициальной предзащите* | Насибулин Данила, Арьемьева Дарья | 2 дня | 24.04 |  |  |  | X |
| *4.3* | *Подготовка к официальной предзащите* | Насибулин Данила, Арьемьева Дарья | 10 дней | 06.05 |  |  |  | X |
| *4.4* | *Подготовка презентации к предзащите* | Насибулин Данила, Арьемьева Дарья | 10 дней | 06.05 |  |  |  | X |
| *4.5* | *Подготовка финального отчета* | Насибулин Данила | 15 дней | 20.05 |  |  |  | X |
| *4.6* | *Подготовка финальной презентации* | Насибулин Данила, Арьемьева Дарья | 5 дней | 01.06 |  |  |  | X |
|  | *Защита проекта* | Насибулин Данила, Арьемьева Дарья, Ядрышников Данил | — | 19.06 – 21.06 |  |  |  |  |

1. Сценарии использования

Система «МойКампус» предполагает два типа пользователей (акторов), которые взаимодействуют с системой в разных ролях: **организатор мероприятия** и **участник мероприятия**. Каждый из них преследует свою цель и взаимодействует с системой по своему сценарию. Ниже описаны основные сценарии использования, которые покрывают ключевые задачи, ради которых создавался сервис.

**Сценарий 1: Организатор создаёт мероприятие**

**Актор:** Организатор

**Цель:** Создать мероприятие, которое будет видно участникам, и получить инструмент для учёта присутствия.

Организатор заходит на сайт «МойКампус», нажимает кнопку «Создать мероприятие» и заполняет форму: название события, дату, время, метсро проведения, краткое описание. После отправки формы система генерирует уникальную ссылку на страницу регистрации и предоставляет организатору интерфейс для последующего сканирования QR-кодов. Также отображается кнопка для экспорта отчёта по участникам в формате Excel.

Интерфейс создания мероприятия показан в **Приложении В, рисунок 6**.

**Сценарий 2: Участник регистрируется на мероприятие**

**Актор:** Участник

**Цель:** Зарегистрироваться на мероприятие и получить подтверждение участия.

Участник переходит по ссылке, которую ему передал организатор (например, в мессенджере, в группе общежития или на плакате с QR). Он попадает на простую форму регистрации, где вводит своё имя, группу, ссыку на ВК. После отправки формы система генерирует персональный QR-код и отображает его на экране с возможностью скачать. QR-код можно сохранить в галерею или показать со смартфона на входе в день мероприятия.

Форма регистрации участника и получение QR-кода показаны в **Приложении В, рисунок 8 и рисунок 9 соответственно**.

**Сценарий 3: Организатор проверяет участников на входе**

**Актор:** Организатор

**Цель:** Отметить присутствие участников на мероприятии.

В день проведения мероприятия организатор открывает встроенный в сайт сканер QR-кодов. Участники подходят и показывают свои коды на экране телефона. Организатор наводит камеру (можно использовать смартфон или ноутбук с камерой), система сканирует код и автоматически помечает участника как «пришедшего». При успешном сканировании система отображает подтверждение регистрации. Участник может проходить внутрь без задержек. Система не требует ручной отметки — всё происходит в реальном времени.

Интерфейс сканера QR-кодов показан в **Приложении В, рисунок 10**.

**Сценарий 4: Организатор выгружает отчёт**

**Актор:** Организатор

**Цель:** Получить итоговый список зарегистрированных и реально пришедших участников.

После завершения мероприятия организатор нажимает кнопку «Экспорт в Excel». Система формирует таблицу в формате Excel, в которой указаны ФИО участников, их группы, статус (пришёл/не пришёл), дата и время регистрации. Этот файл можно отправить в РВД (Рейтинг внеучебной деятельности) для начисления баллов внеучебной деятельности.

Интерфейс мероприятия (где и «Экспорт в Excel») показан в **Приложении В, рисунок 7**.

**Сценарий 5: Участник выбирает мероприятие из ленты**

**Актор:** Участник

**Цель:** Найти интересное мероприятие и зарегистрироваться.

Участник открывает сайт «МойКампус» и попадает на главную страницу, где отображаются все активные мероприятия. Он может пролистать список, выбрать нужное и перейти на форму регистрации. Такой подход особенно удобен для студентов, которые не получили прямую ссылку, но хотят участвовать в событиях.

Интерфейс ленты мероприятий показан в **Приложении В, рисунок 11**.

**Сценарий 6: Организатор управляет своими мероприятиями**

**Актор:** Организатор

**Цель:** Просматривать и редактировать все созданные им мероприятия.

Организатор открывает специальный раздел, в котором видит список всех созданных им мероприятий. Здесь доступны действия: редактирование, удаление, публикация/снятие с публикации. Это даёт организатору полный контроль над событиями и позволяет быстро менять статус мероприятий в зависимости от ситуации.

Интерфейс управления мероприятиями со стороны организатора представлен в **Приложении В, рисунок 12**.

Эти шесть базовых сценария охватывают весь минимально необходимый функционал (MVP), ради которого создавался проект. Каждый из них ориентирован на конкретную цель пользователя и демонстрирует, как система помогает её достичь. При этом архитектура построена таким образом, что дальнейшее масштабирование возможно без изменений основных сценариев. Например, можно добавить аутентификацию, интерфейс для просмотра статистики, интеграцию с календарём или уведомления — при сохранении уже отработанных взаимодействий.

1. Требования к продукту и к MVP

Система «МойКампус» разрабатывалась как продукт, решающий конкретную задачу — упростить и ускорить процесс регистрации и учёта участников мероприятий в студенческом кампусе. Исходя из проблем, которые мы выявили на раннем этапе, а также на основе общения с будущими пользователями, мы сформулировали набор требований к продукту, который можно условно разделить на **требования клиентов (бизнес-требования)**, **пользовательские**, **функциональные**, **нефункциональные** и **производные**. Эти требования определяли как конечную структуру системы, так и минимально необходимый функционал для MVP — минимально жизнеспособной версии продукта, которую мы могли реализовать и протестировать в рамках учебного проекта.

**1. Бизнес-требования (требования клиентов)**

Основным заказчиком продукта выступает студенческое сообщество Новокольцовского кампуса (НВК), в лице студенческого совета и организаторов мероприятий. Их главная потребность — сократить время на ручной учёт участников, автоматизировать процесс отметки присутствующих и упростить составление отчётов.

Бизнес-требования:

* устранить необходимость ручной работы с таблицами и формами регистрации;
* исключить очереди на входе на мероприятие;
* автоматизировать формирование отчётности для РВД;
* упростить взаимодействие участников с системой до пары кликов.

**2. Пользовательские требования**

Пользовательские требования — это действия, которые должны быть доступны конечным пользователям системы: организаторам и участникам. Они определяют, что именно пользователь должен иметь возможность сделать.

Пользовательские требования:

* организатор должен иметь возможность создать мероприятие через форму на сайте;
* участник должен иметь возможность зарегистрироваться, не создавая аккаунт;
* участник должен получить персональный QR-код после регистрации;
* организатор должен иметь возможность сканировать QR-код участника и отмечать его как присутствующего;
* организатор должен иметь возможность выгрузить итоговый отчёт по мероприятию в формате таблицы.

**3. Функциональные требования**

Функциональные требования описывают поведение системы, которое обеспечивает реализацию пользовательских сценариев. Они являются прямыми следствиями пользовательских требований.

Функциональные требования:

* система должна предоставлять организатору интерфейс для создания мероприятия;
* система должна принимать и сохранять данные участников, отправленные через форму регистрации;
* система должна генерировать QR-коды для каждого зарегистрированного участника;
* система должна сканировать QR-коды через камеру устройства и обновлять статус участника;
* система должна экспортировать данные о мероприятии и посещении в Excel-таблицу;
* система должна отображать активные мероприятия на главной странице сайта;
* организатор должен иметь возможность видеть, редактировать и публиковать/снимать с публикации свои мероприятия.

**4. Нефункциональные требования**

Нефункциональные требования определяют технические ограничения и свойства системы, не связанные напрямую с её поведением, но влияющие на её эксплуатацию и удобство использования.

Нефункциональные требования:

* веб-приложение должно работать в браузере без необходимости установки;
* интерфейс должен быть адаптивным и корректно отображаться на смартфонах;
* сканирование QR-кодов должно происходить без задержек, с минимальным временем реакции;
* система должна быть устойчива к случайным ошибкам ввода или потере связи;
* все данные должны сохраняться в лёгкой реляционной базе (SQLite), подходящей для MVP.

**5. Производные требования**

Эти требования появились как следствия бизнес-ограничений и технических условий проекта. Например, из необходимости обеспечить работу сервиса без регистрации пользователей возникла необходимость создавать уникальные мероприятия по короткому коду, а из отсутствия серверной инфраструктуры — использование SQLite и простого Flask-бекенда.

Производные требования:

* регистрация участников должна работать без авторизации и личных кабинетов;
* база данных должна быть легковесной, простой в сопровождении;
* отчёты должны генерироваться в широко распространённом формате — Excel;
* приложение должно быть кроссплатформенным, без привязки к ОС.

Всё перечисленное выше легло в основу MVP-версии сервиса, которую мы реализовали в рамках проекта. Она включает в себя:

* форму создания мероприятия,
* форму регистрации участников,
* генерацию QR-кодов,
* сканирование QR-кодов с отметкой о посещении,
* экспорт Excel-файла с отчётом.

Этого набора функционала оказалось достаточно, чтобы покрыть основные пользовательские сценарии, протестировать продукт в условиях реального использования и убедиться в его работоспособности. В дальнейшем система может быть расширена за счёт добавления ленты событий, аналитики, оповещений и других элементов, но именно эти требования стали каркасом базовой версии — той, которая уже решает реальную проблему в кампусе.

1. Стек для разработки

При выборе технологий для проекта «МойКампус» мы исходили не из желания применить «модные» или «продвинутые» решения, а из реальных ограничений и целей: запустить рабочее MVP за короткий срок, сделать его понятным, поддерживаемым, и, главное, доступным для всей команды. Мы понимали, что наша задача — не выстраивать архитектуру на годы вперёд, а реализовать минимально жизнеспособный продукт, который будет решать проблему уже сейчас. Поэтому выбор стека определялся балансом между простотой, скоростью и достаточной гибкостью.

Первоначально мы рассматривали популярные связки вроде React + Node.js + PostgreSQL, но быстро поняли, что без уверенного опыта и времени на изучение таких решений мы можем увязнуть в технических деталях. В результате было принято разумное решение отказаться от сложных фреймворков и использовать хорошо документированные, быстрые в освоении и удобные в разработке технологии.

**Frontend**

Для клиентской части мы выбрали **HTML** + **CSS** + **JavaScript** с подключением библиотеки **Bootstrap**. Этот стек позволил нам:

* быстро сверстать адаптивные страницы, которые корректно работают на любых устройствах;
* использовать готовые компоненты интерфейса (модальные окна, формы, кнопки) без необходимости писать их с нуля;
* подключить сторонние библиотеки для генерации QR-кодов (например, qrcode.js) и обработки пользовательского взаимодействия.

Такой подход сделал фронтенд лёгким, понятным и доступным для изменения любому участнику команды без сложной сборки и зависимостей.

**Backend**

Серверную часть мы реализовали на языке **Python** с использованием фреймворка **Flask**. Flask — это минималистичный веб-фреймворк, который идеально подошёл под задачи MVP:

* он не требует большого количества конфигураций;
* легко подключается к базе данных и обрабатывает запросы от фронтенда;
* имеет огромное количество документации и примеров;
* позволяет за короткое время построить стабильный API для работы с данными пользователей и мероприятий.

Также Python дал возможность использовать библиотеки для генерации отчетов (например, openpyxl для Excel) и внедрить первичную логику обработки данных и ролей (например, ограничение прав доступа для организаторов).

**База данных**

В качестве хранилища данных мы выбрали **SQLite**. Это встроенная легковесная реляционная база данных, которая:

* не требует установки сервера и дополнительных зависимостей;
* отлично работает в рамках небольших проектов и MVP;
* проста в настройке и сопровождении;
* легко переносится между средами разработки.

Такое решение позволило нам сразу сосредоточиться на логике приложения, а не на инфраструктуре.

Таким образом, весь стек был подобран не ради технологий, а ради результата. Мы поставили перед собой цель — **реально работающий продукт за 2–3 месяца** — и достигли её за счёт использования понятных инструментов, с которыми можно быстро развернуть MVP и при необходимости доработать его. Если в будущем проект продолжит развиваться, мы сможем безболезненно перейти на более масштабируемые решения (например, PostgreSQL, полноценный SPA-фреймворк или REST/GraphQL API), но для текущего этапа наш выбор оказался оптимальным.

1. Прототипирование

Создание прототипа для нашего проекта было важнейшим этапом в разработке. Оно позволило нам не только визуализировать будущий продукт, но и проверить, насколько наши представления об интерфейсе совпадают с ожиданиями пользователей. Прототип стал средством коммуникации между членами команды, заказчиками и потенциальными пользователями — мы могли показать, как будет выглядеть система, не дожидаясь завершения разработки. Такой подход помог быстро собрать обратную связь и внести корректировки ещё до начала реализации функциональности.

Процесс создания прототипа включал в себя четыре основных этапа.

Во-первых, мы начали с определения начальных требований. Для этого была собрана информация о том, что именно должно присутствовать на сайте: форма регистрации для участников, окно создания мероприятия для организаторов, интерфейс для сканирования QR-кодов и страница с выгрузкой отчёта. Мы ориентировались на пользовательские сценарии и базовую логику MVP.

Во-вторых, на основе этих требований был разработан первый визуальный прототип. Сначала тимлид составил черновой макет интерфейса в самом простом редакторе — Paint. Несмотря на примитивность инструмента, этот начальный вариант чётко отображал структуру страниц, порядок переходов, расположение кнопок и ключевых полей. Такой подход позволил быстро донести идею до дизайнера и всей команды без лишней траты времени. После этого на основе нарисованного макета в Paint дизайнер начал работу в **Figma** — наиболее простом, гибком и доступном инструменте для создания прототипов с возможностью совместного редактирования. Первая версия прототипа в Figma включала базовые страницы без лишней графики — главное было сфокусироваться на логике взаимодействия и удобстве для конечного пользователя. Мы использовали типовые элементы управления и хорошо знакомые шаблоны юзабилити, чтобы интерфейс выглядел интуитивно понятным.

Визуальные материалы, включая черновой набросок в Paint, итоговый макет в Figma, а также схемы текущего и предлагаемого процессов регистрации, приведены в **Приложении А**.

На третьем этапе мы представили этот прототип нашему куратору и потенциальным пользователям — студентам, проживающим в НВК. Мы получили обратную связь по ключевым моментам: оформление форм, структура взаимодействий, порядок действий для организатора. Именно на этом этапе стало понятно, что регистрация через личные кабинеты — избыточное решение, и было принято решение отказаться от неё. Также мы пересмотрели количество полей, убрали лишние шаги, сделали интерфейс максимально «плоским» — без лишней навигации. Эта обратная связь была критически важной: она помогла избежать ошибок, которые стоили бы времени и усилий на этапе разработки.

После сбора замечаний мы перешли к четвёртому этапу — переработке прототипа. Макеты были полностью обновлены: структура форм упростилась, элементы стали визуально легче, а все ключевые сценарии взаимодействия — более прямолинейными. На этом этапе прототип стал не просто наброском, а визуальным отражением всей будущей системы. Его можно было использовать как основу для верстки, а не только как демонстрационный материал. Такой подход обеспечил нам связность между дизайном и реализацией, что значительно ускорило разработку фронтенда.

На момент реализации финального прототипа он стал максимально близким к итоговому продукту по логике, визуальному оформлению и структуре страниц. Он хорошо воспринимается как пользователями, так и преподавателями, так как выглядит завершённо, не вызывает вопросов и демонстрирует полный пользовательский путь — от регистрации до сканирования и отчётности.

1. Проектирование и разработка системы

На этапе проектирования и разработки системы «МойКампус» мы определили основные функциональные блоки продукта и сформировали структуру, в которой каждый программный модуль отвечает за отдельную часть логики. Такой подход позволил упростить разработку, разделить задачи между участниками команды и ускорить процесс интеграции. Архитектура системы строится по принципу модульности: каждая часть приложения выполняет чётко определённую функцию и взаимодействует с другими через ограниченные интерфейсы.

**Общая структура системы**

Приложение условно делится на три уровня:

1. **Клиентская часть (Frontend)** — отвечает за отображение страниц, обработку действий пользователя и взаимодействие с сервером;
2. **Серверная часть (Backend)** — отвечает за обработку запросов, хранение данных, генерацию QR-кодов и логическую обработку действий;
3. **База данных (SQLite)** — хранит сведения о мероприятиях, участниках и статусах их регистрации.

**Описание основных модулей**

**1. Модуль создания мероприятия (организатор)**

Позволяет организатору задать параметры мероприятия (название, описание, дата, место проведения), сохранить его и получить ссылку на форму регистрации. Система сохраняет мероприятие в базу данных и создаёт интерфейс для его управления.

**2. Модуль регистрации участника (публичная форма)**

Участник по ссылке заполняет форму (ФИО, группа, ссылка на ВК). После отправки данных система сохраняет их в базу и генерирует уникальный QR-код, который отображается участнику. Код можно сохранить или показать с экрана.

**3. Модуль сканирования QR-кодов (организатор)**

Позволяет организатору сканировать коды с помощью камеры (на телефоне или ноутбуке). QR-код содержит ID участника и код мероприятия. Система проверяет данные и помечает участника как "присутствующий".

**4. Модуль экспорта отчёта (организатор)**

Формирует таблицу в формате Excel, где указаны ФИО участников, статус посещения, дата, время. Используется библиотека openpyxl.

**5. Внутренний API (backend)**

Обрабатывает HTTP-запросы от frontend, работает с базой данных и реализует логику: добавление записей, проверку кодов, формирование отчётов.

**6. Модуль генерации QR-кодов**

На основе ID участника и кода мероприятия создаёт изображение QR-кода (через JavaScript-библиотеку на фронте).

**Алгоритм работы приложения**

1. Организатор создаёт мероприятие → данные сохраняются в БД, генерируется уникальная ссылка.
2. Участник переходит по ссылке и регистрируется → данные сохраняются, генерируется QR-код.
3. На мероприятии организатор сканирует QR-коды → статус участника обновляется в базе.
4. После мероприятия организатор нажимает кнопку "Экспорт в Excel" → Excel-файл формируется автоматически.
5. Все данные доступны только организатору по уникальному коду мероприятия.

**Схема взаимодействия модулей (поток данных)**

Система «МойКампус» построена на клиент-серверной архитектуре и включает три основных компонента: клиентскую часть, серверную часть и базу данных.

* **Клиентская часть** (frontend) — это интерфейс для организаторов и участников, через который выполняются основные действия: создание мероприятий, регистрация, сканирование QR-кодов и экспорт отчётов. Все действия пользователя передаются на сервер в виде HTTP-запросов.
* **Серверная часть** (backend) — принимает запросы от клиента, обрабатывает их и взаимодействует с базой данных. Здесь реализована логика обработки регистраций, генерации QR-кодов, проверки посещения и формирования отчётности.
* **База данных** (SQLite) — хранит сведения о мероприятиях, участниках и статусах. Все обращения к ней проходят через сервер.

Передача данных осуществляется строго по цепочке: клиент → сервер → база данных, и обратно. Такая структура обеспечивает чёткое разделение обязанностей и упрощает развитие системы в будущем.

Такое разделение позволило нам вести параллельную разработку модулей: один участник работал над серверной частью, другой — над клиентской, третий — занимался дизайном интерфейсов и форм. Мы тестировали каждую часть отдельно, что упростило поиск и исправление ошибок, а в конце связали модули в единое приложение.

Примеры ключевых экранов и интерфейсов реализованного MVP-продукта приведены в **Приложении В**: форма регистрации, генерация и сканирование QR-кодов, интерфейс организатора и отображение ленты мероприятий.

ЗАКЛЮЧЕНИЕ

В ходе проектной деятельности нами была разработана информационная система «МойКампус» — веб-сервис для регистрации и учёта участников мероприятий в Новокольцовском кампусе УрФУ. Проект стал результатом последовательной аналитики, проектирования, командной разработки и тестирования, направленных на решение реальной проблемы, с которой сталкиваются студенты-организаторы и участники мероприятий.

На начальном этапе были выявлены основные недостатки существующего подхода, основанного на Google-формах и ручной отметке посещений: чрезмерная трудоёмкость, высокая вероятность ошибок, долгие очереди на входе и сложности с отчётностью. Мы провели анализ целевой аудитории, подробно рассмотрели поведение как организаторов, так и участников, и определили их мотивации и «боли». Были изучены аналоги, среди которых ни один не соответствовал полностью специфике и потребностям студенческого кампуса.

Команда предложила и проанализировала несколько альтернативных решений, таких как Telegram-бот, мобильное приложение и доработанные таблицы. Однако в процессе мозгового штурма и сравнения критериев эффективности было принято решение остановиться на разработке собственного веб-приложения — лёгкого, быстрого и понятного в использовании. В нём учтены особенности повседневной студенческой жизни: отсутствие необходимости в установке, мгновенная регистрация, простой сканер QR-кодов и автоматическая генерация отчёта.

На этапе проектирования была определена архитектура системы: приложение разделено на три основных компонента — frontend, backend и базу данных. Каждый модуль реализует отдельную часть логики: создание мероприятия, регистрация участников, сканирование кодов и формирование отчёта. Эта структура обеспечивает чёткость в работе и простоту сопровождения. Сценарии использования были подробно проработаны и реализованы в MVP-версии системы.

Особое внимание было уделено созданию прототипа: начиная с первых черновиков, нарисованных в Paint, и заканчивая полнофункциональными макетами в Figma. Полученная обратная связь от кураторов и студентов позволила уточнить структуру, упростить интерфейс и избежать ненужных элементов, которые могли бы усложнить взаимодействие.

Разработка велась на простом и доступном стеке: HTML, CSS, JavaScript (с Bootstrap) на клиентской стороне, Flask (Python) на серверной и SQLite в качестве базы данных. Такой выбор был обусловлен желанием обеспечить быструю разработку, лёгкость тестирования и полное понимание кода каждым участником команды. Это позволило запустить MVP в ограниченные сроки и сосредоточиться на ключевой логике, а не на инфраструктурных задачах.

Положительными сторонами разработанного решения стали:

* высокая скорость регистрации и отметки участников;
* отсутствие необходимости в установке или авторизации;
* простота интерфейса, адаптированного под мобильные устройства;
* автоматическое формирование отчётности;
* гибкость архитектуры и возможность расширения.

К потенциальным ограничениям можно отнести:

* отсутствие личных кабинетов и истории мероприятий (на данный момент);
* базовая система прав доступа (реализована через код мероприятия);
* отсутствие масштабируемости на нагрузку более сотни участников (ограничено ресурсами MVP).

На текущем этапе проект завершён в рамках минимального жизнеспособного продукта (MVP), и в ближайшее время планируется провести **полевое тестирование** на реальных мероприятиях в кампусе. Это позволит собрать ещё больше пользовательской обратной связи и определить направления дальнейшего развития.

В перспективе проект может быть дополнен следующими функциями:

* лента предстоящих мероприятий и возможность выбрать событие по интересам;
* история посещённых мероприятий для участников;
* более гибкая система ролей для организаторов;
* визуальная аналитика по участникам и активности;
* интеграция с другими платформами университета.

Подводя итог, можно сказать, что проект «МойКампус» стал успешной реализацией идеи, выросшей из наблюдения за реальными проблемами. Мы не просто разработали сайт, а создали инструмент, который уже готов приносить пользу. Командная работа, гибкий подход к технологиям и постоянное взаимодействие с потенциальными пользователями позволили нам добиться главного — реализовать работающий продукт, решающий конкретную задачу в жизни кампуса.

MVP-версия приложения была полностью реализована, включая все ключевые интерфейсы. Их визуальное представление включено в **Приложение В**.

Отдельно стоит отметить количественный эффект внедрения системы. При старом подходе организаторы тратили в среднем **35 минут на подготовку** и ещё до **85 минут на регистрацию и отметку участников** — особенно при потоке в 150+ человек. Благодаря автоматизации всех ключевых процессов: создания мероприятия, регистрации, отметки и формирования отчёта — общее время работы сократилось **в 10 раз**. Теперь организатору требуется всего **3 минуты на подготовку**, а вся отметка занимает около **10 минут** при тех же масштабах. Участники, которые раньше ждали своей очереди по **30–40 секунд**, теперь проходят сканирование QR-кода за **3 секунды**. Это означает не только значительную экономию времени и ресурсов, но и устранение очередей, снижение ошибок и повышение удовлетворённости всех сторон мероприятия — как организаторов, так и участников.

СПИСОК ЛИТЕРАТУРЫ

1. Flask Documentation. – 2024. – URL: <https://flask.palletsprojects.com/en/2.3.x/> (дата обращения: 04.04.2025).
2. SQLite Documentation / SQLite.org. – 2024. – URL: <https://www.sqlite.org/docs.html> (дата обращения: 04.04.2025).
3. Bootstrap Documentation / Bootstrap v5. – 2024. – URL: <https://getbootstrap.com/docs/5.3/getting-started/introduction/> (дата обращения: 05.04.2025).
4. QRCode.js – библиотека генерации QR-кодов. – URL: <https://davidshimjs.github.io/qrcodejs/> (дата обращения: 06.04.2025).
5. MDN Web Docs – Основы HTML, CSS и JavaScript. – URL: <https://developer.mozilla.org/ru/> (дата обращения: 05.04.2025).
6. VC.RU. Анализ конкурентов на практике / VC.RU. – 2022. – URL: <https://vc.ru/marketing/181297-analiz-konkurentov-na-praktike-10-shagov-shablon> (дата обращения: 16.02.2025).
7. VC.RU. Как составить портрет клиента (целевой аудитории) / VC.RU. – 2020. – URL: <https://vc.ru/marketing/156147-kak-sostavit-portret-klienta-celevoy-auditorii-instrukciya-s-primerami> (дата обращения: 14.02.2025).
8. Proglib. Прокрастинация: почему разработчики к ней склонны и что с этим делать. – 2024. – URL: <https://proglib.io/p/prokrastinaciya-pochemu-razrabotchiki-k-ney-sklonny-i-chto-s-etim-delat-2024-12-20> (дата обращения: 21.03.2025).
9. SimilarWeb. Аналитика сайтов конкурентов / SimilarWeb.com. – URL: <https://www.similarweb.com> (дата обращения: 16.02.2025).
10. Figma Help Center – Руководство по созданию макетов. – URL: <https://help.figma.com/hc/ru> (дата обращения: 25.03.2025).

ПРИЛОЖЕНИЕ А – ПРОТОТИПЫ И ВИЗУАЛЬНЫЕ ПРЕДСТАВЛЕНИЯ

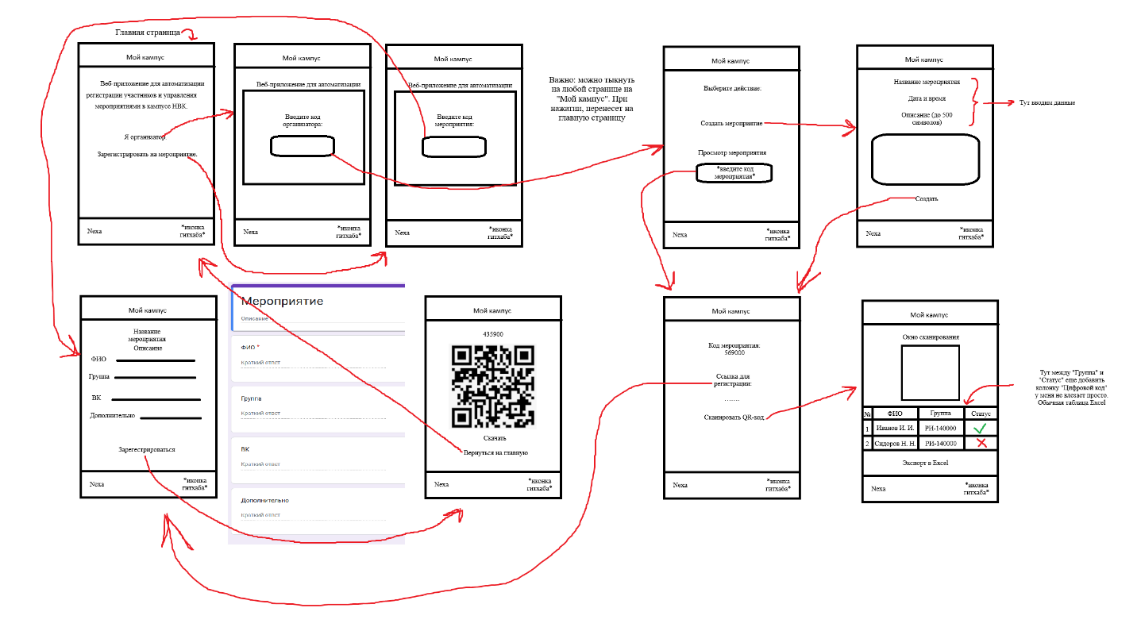


Рисунок 1 — Черновой прототип интерфейса, нарисованный в Paint 

Рисунок 2 — Итоговый макет в Figma

Ссылка на макет: <https://www.figma.com/design/gF5QKhsz85SQxRohIuwvKT/%D0%9C%D0%BE%D0%B9%D0%9A%D0%B0%D0%BC%D0%BF%D1%83%D1%81?node-id=0-1&t=lIlhy1HJ1tmd0Ib7-1>

ПРИЛОЖЕНИЕ Б – ГРАФИКИ И СРАВНИТЕЛЬНЫЕ СХЕМЫ

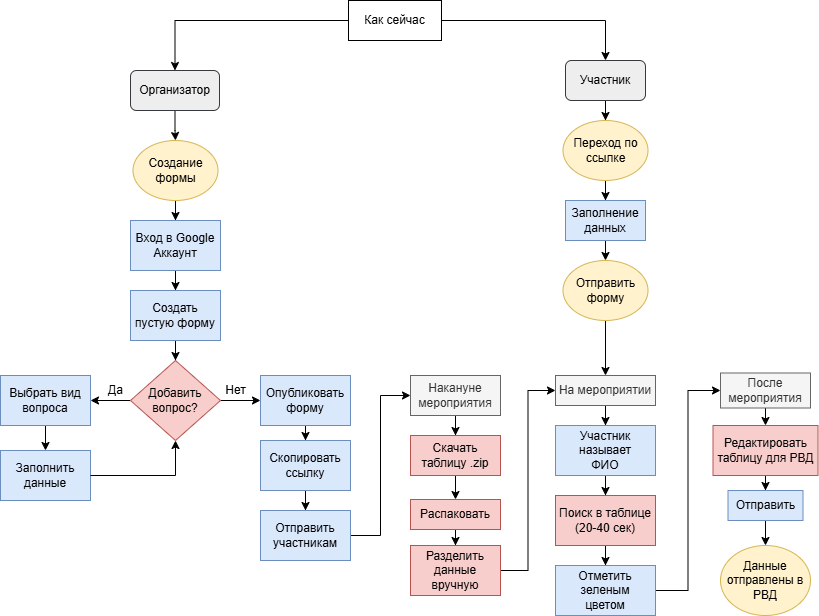


Рисунок 3 — Сценарий прохождения регистрации и учёта участников до внедрения системы

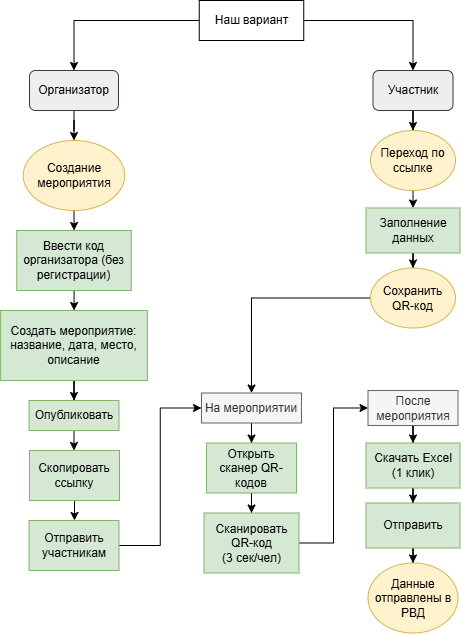


Рисунок 4 — Сценарий прохождения регистрации и учёта участников с системой «МойКампус»

ПРИЛОЖЕНИЕ В – РЕАЛИЗОВАННЫЙ ИНТЕРФЕЙС MVP ВЕБ-ПРИЛОЖЕНИЯ «МОЙКАМПУС»

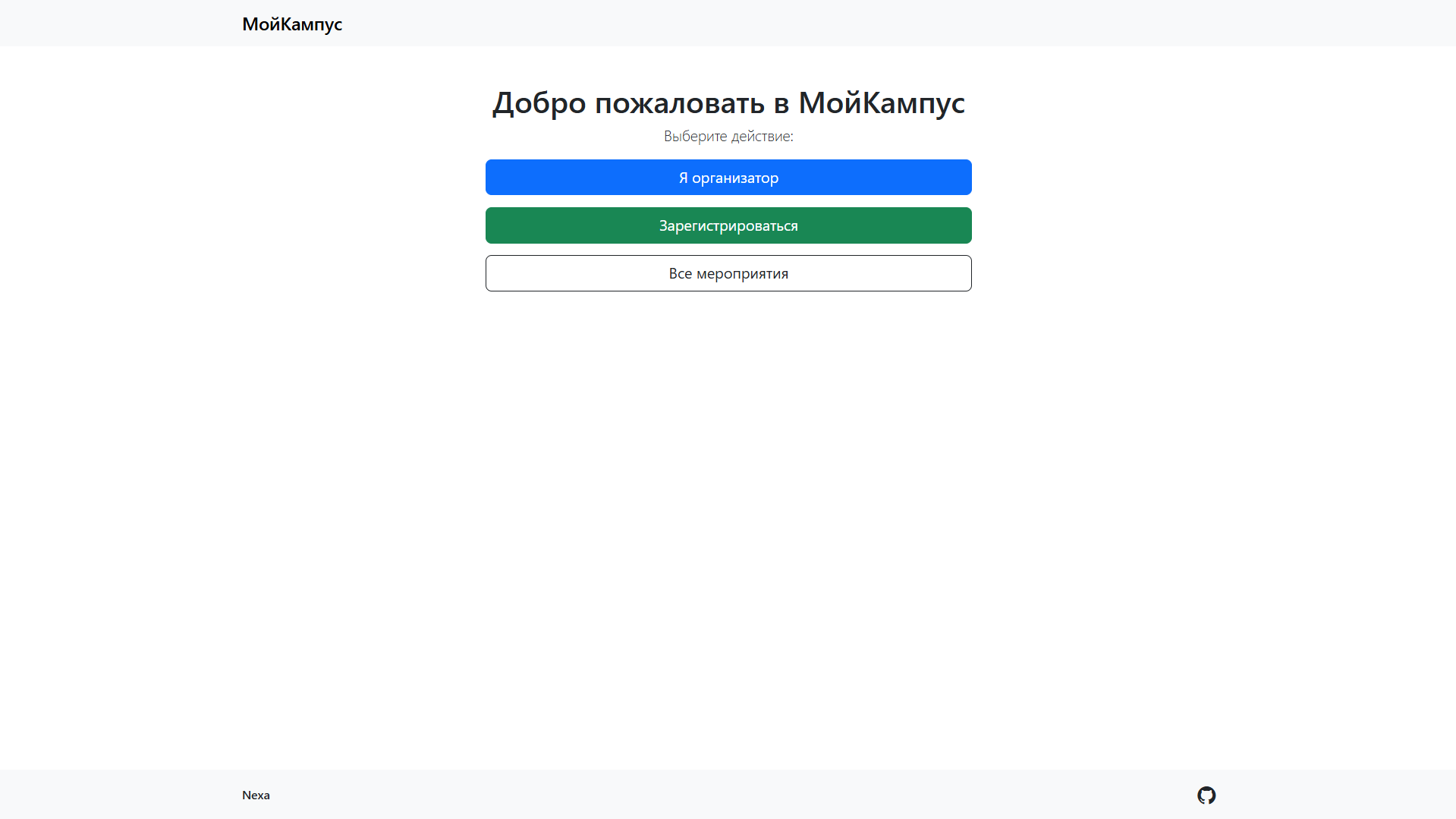


Рисунок 5 — Главная страница сайта

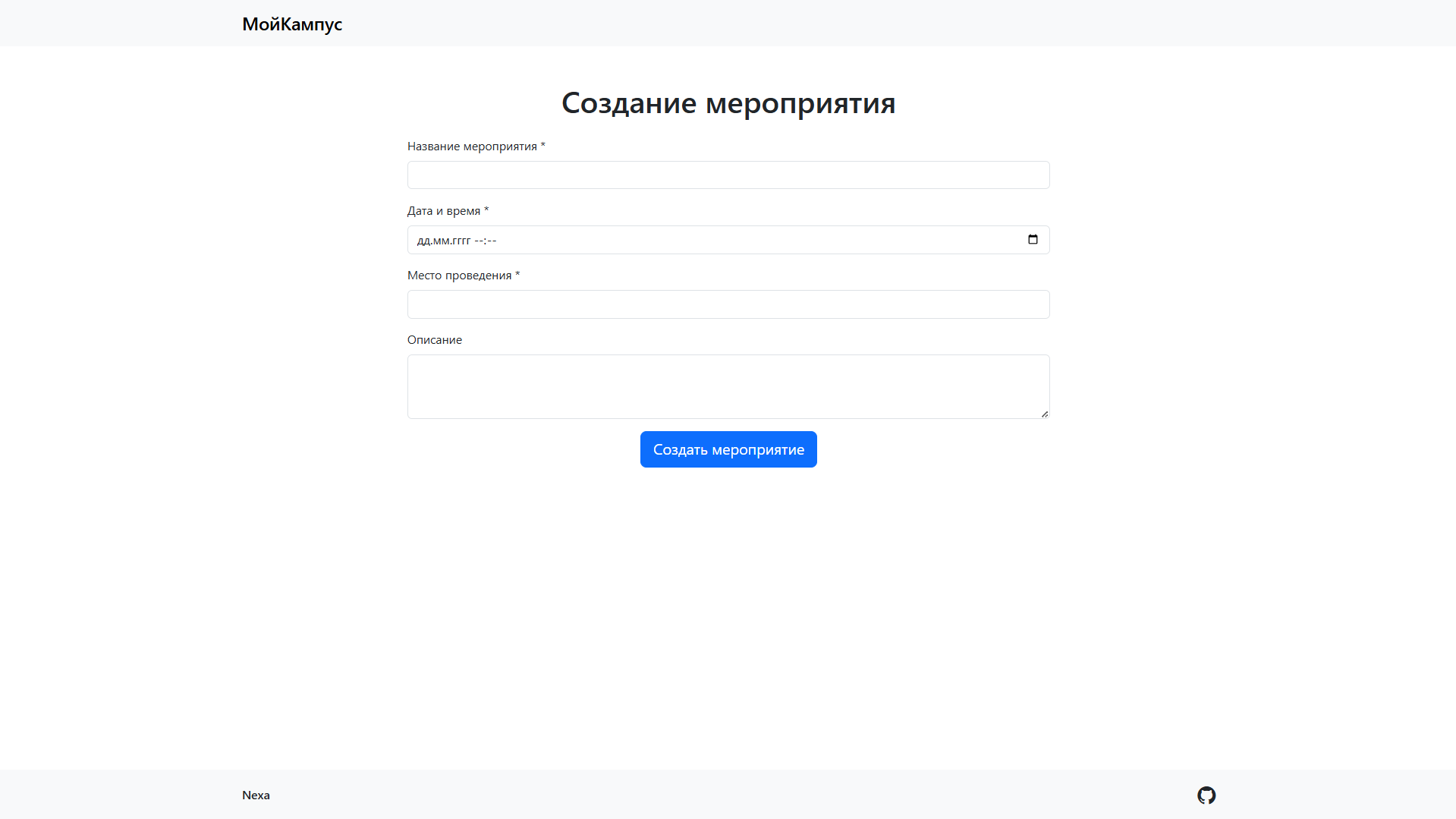


Рисунок 6 — Интерфейс создания мероприятия

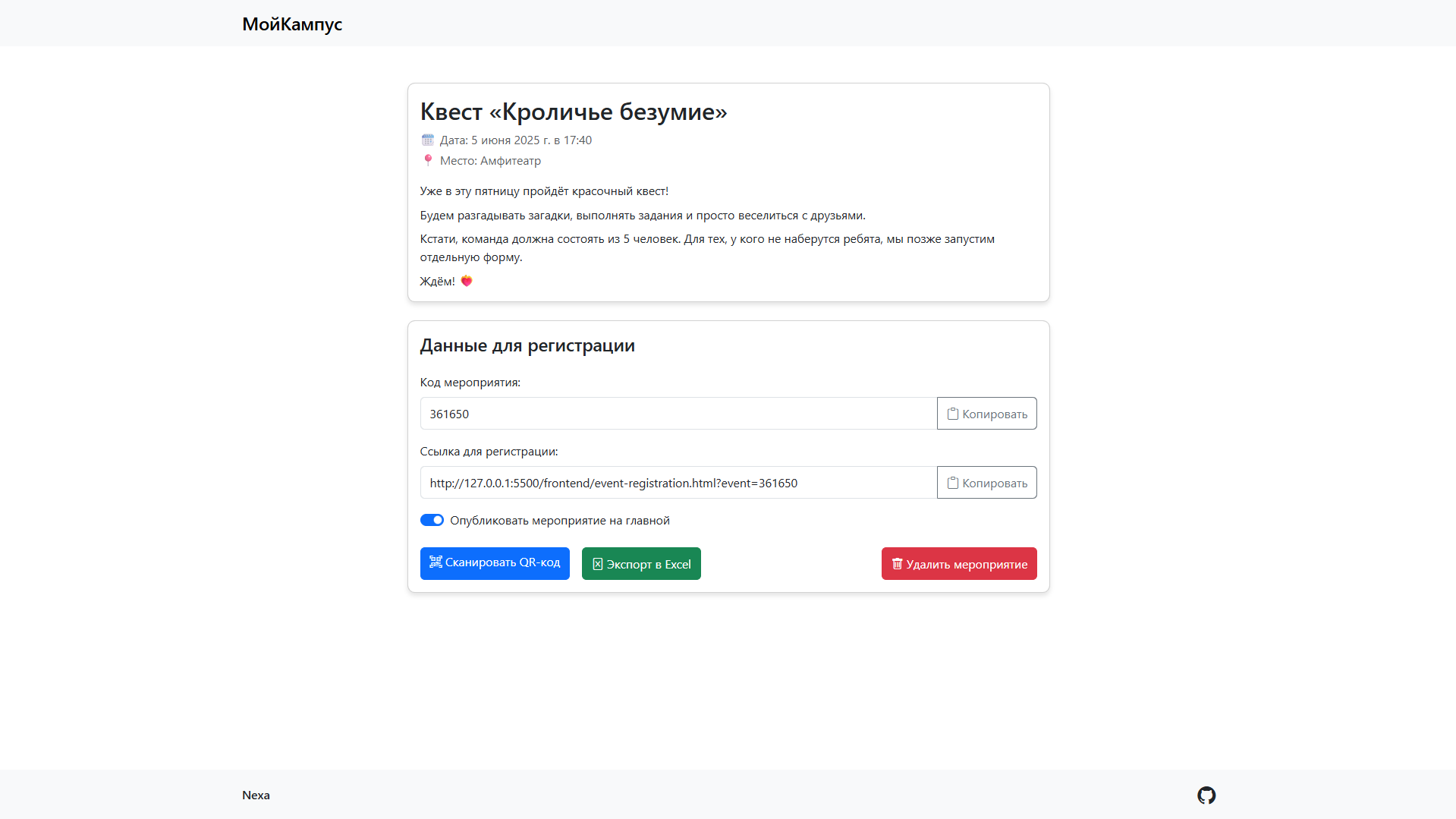


Рисунок 7 — Интерфейс мероприятия

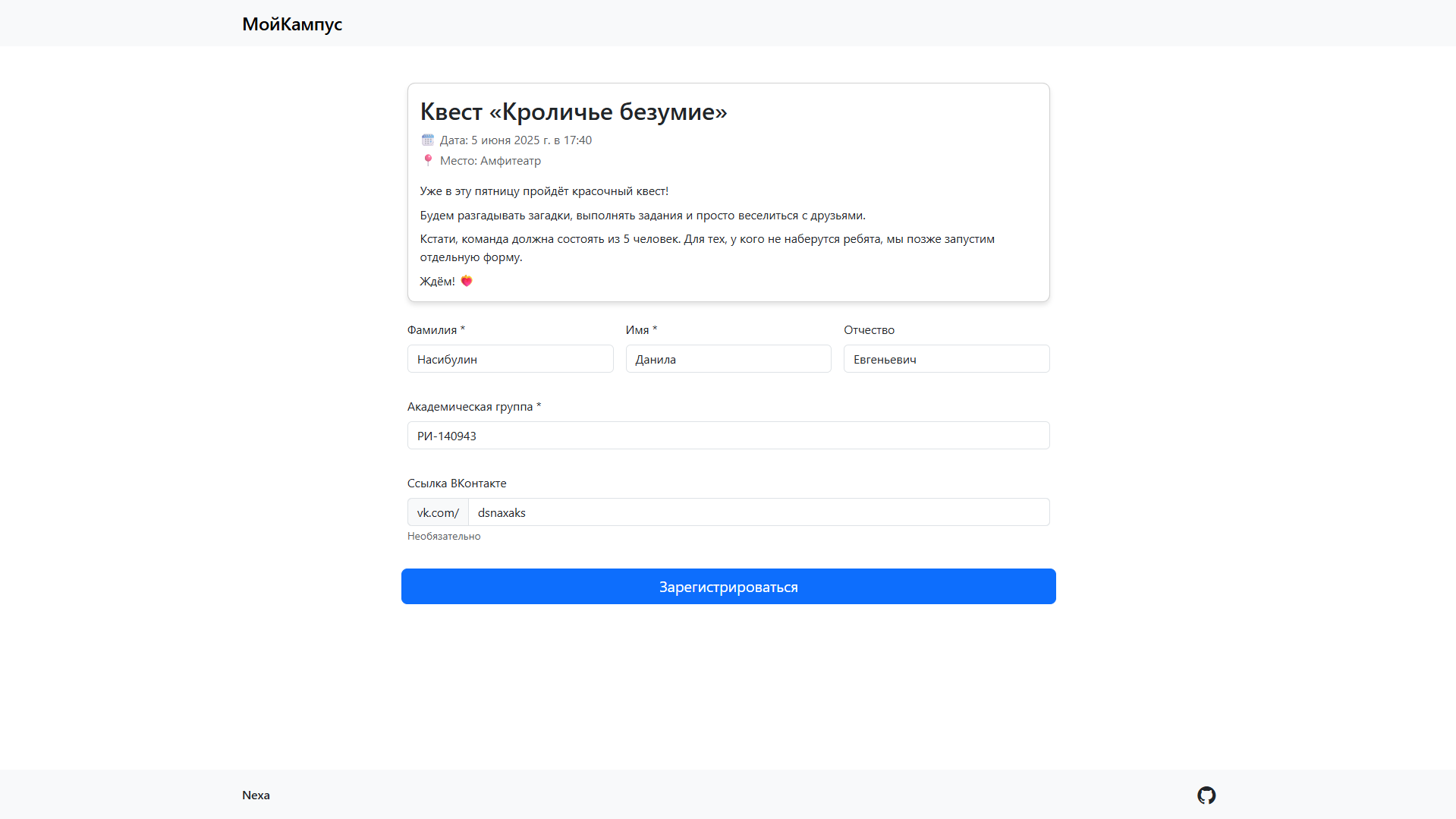


Рисунок 8 — Форма регистрации участника

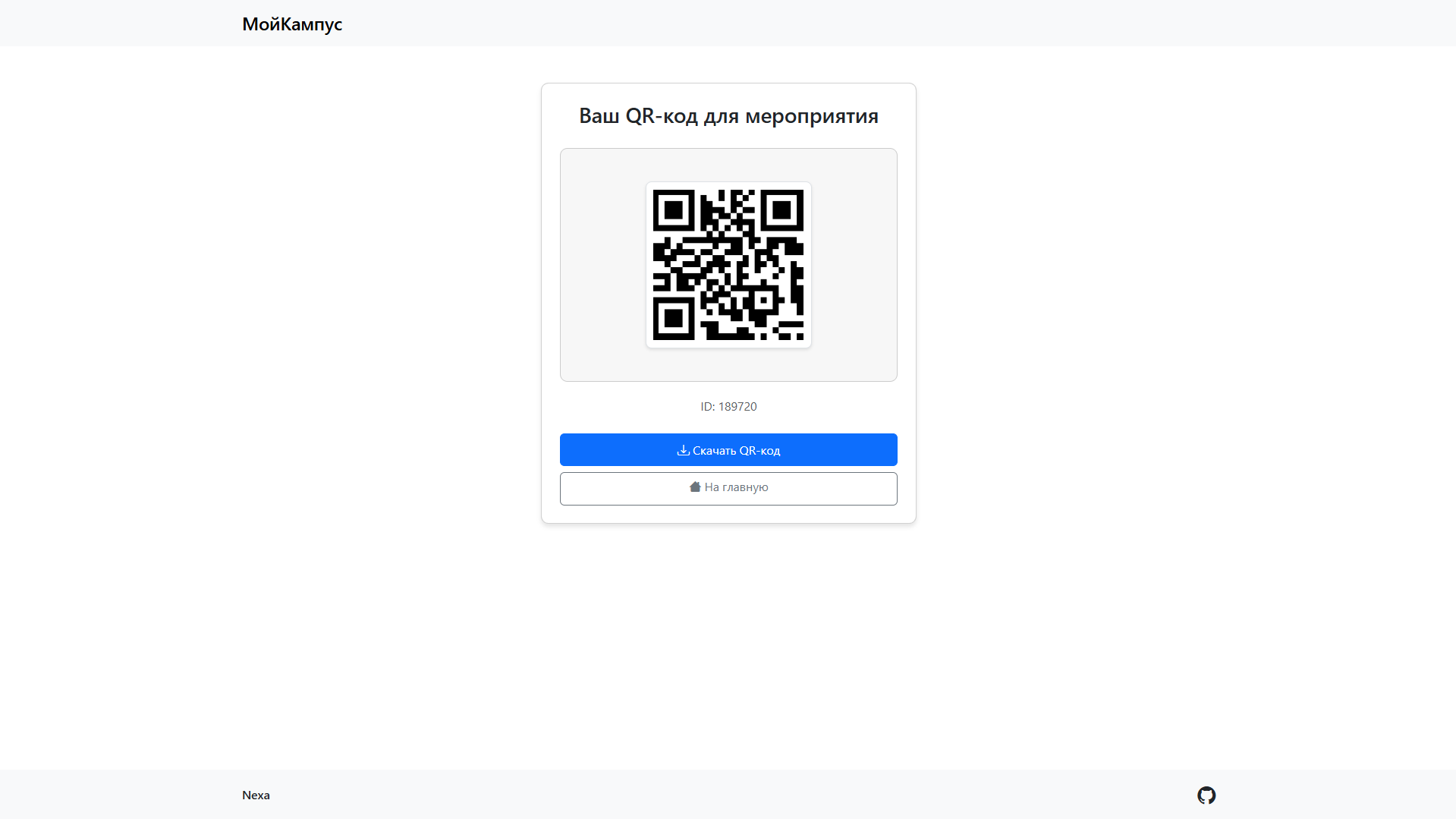


Рисунок 9 — Получение QR-кода участником

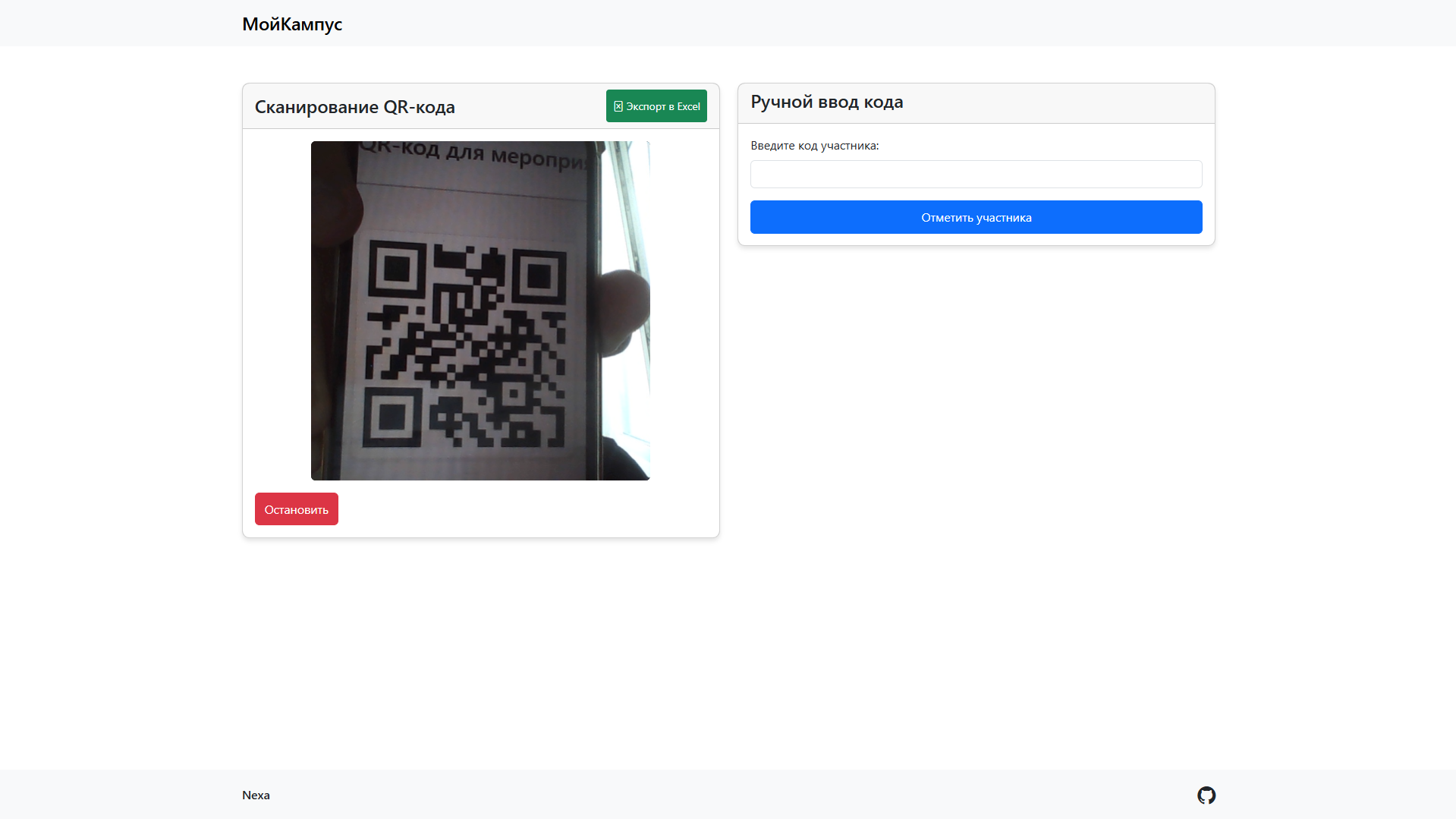


Рисунок 10 — Интерфейс сканера QR-кодов

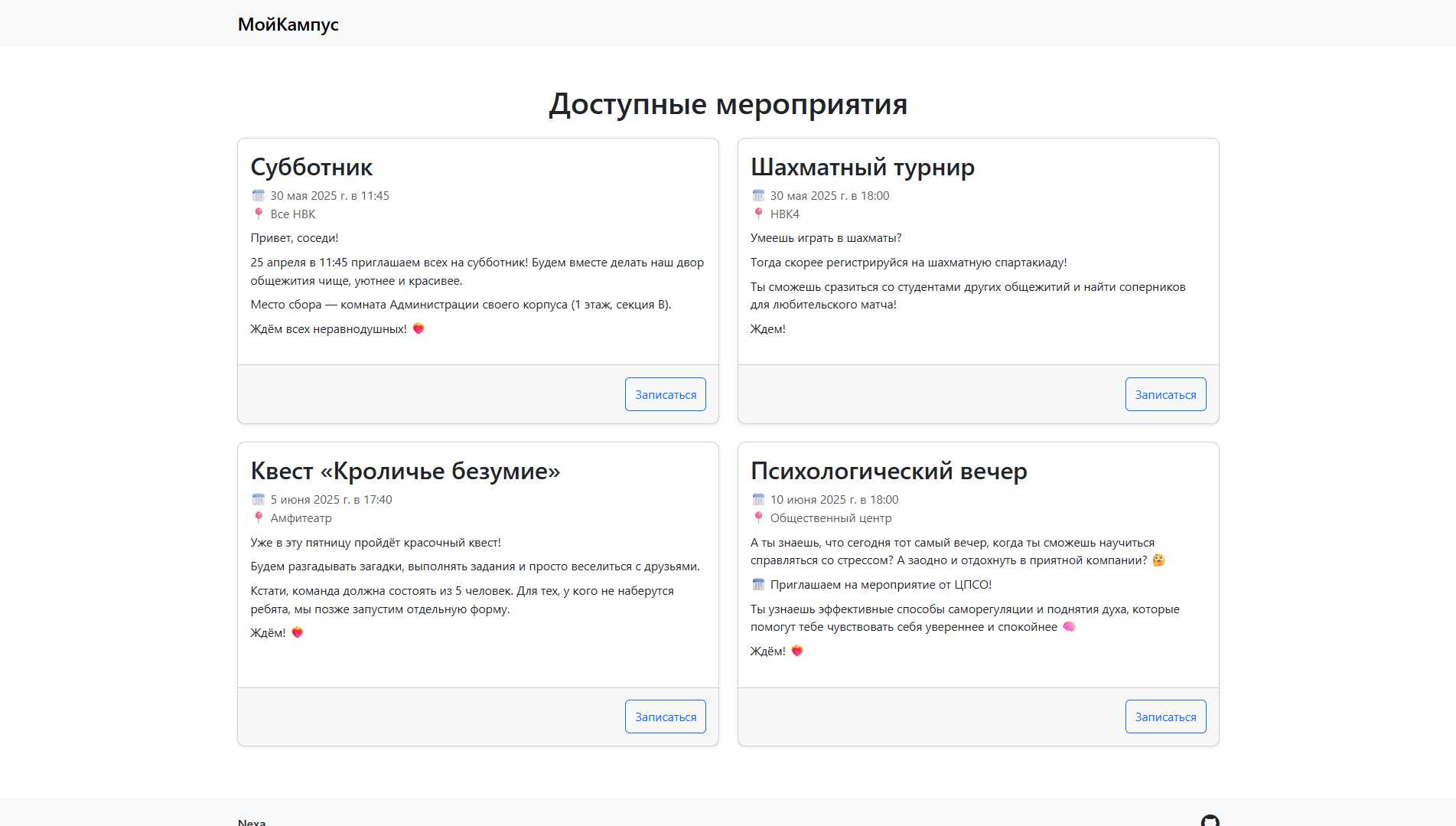


Рисунок 11 — Доступные мероприятия со стороны участника

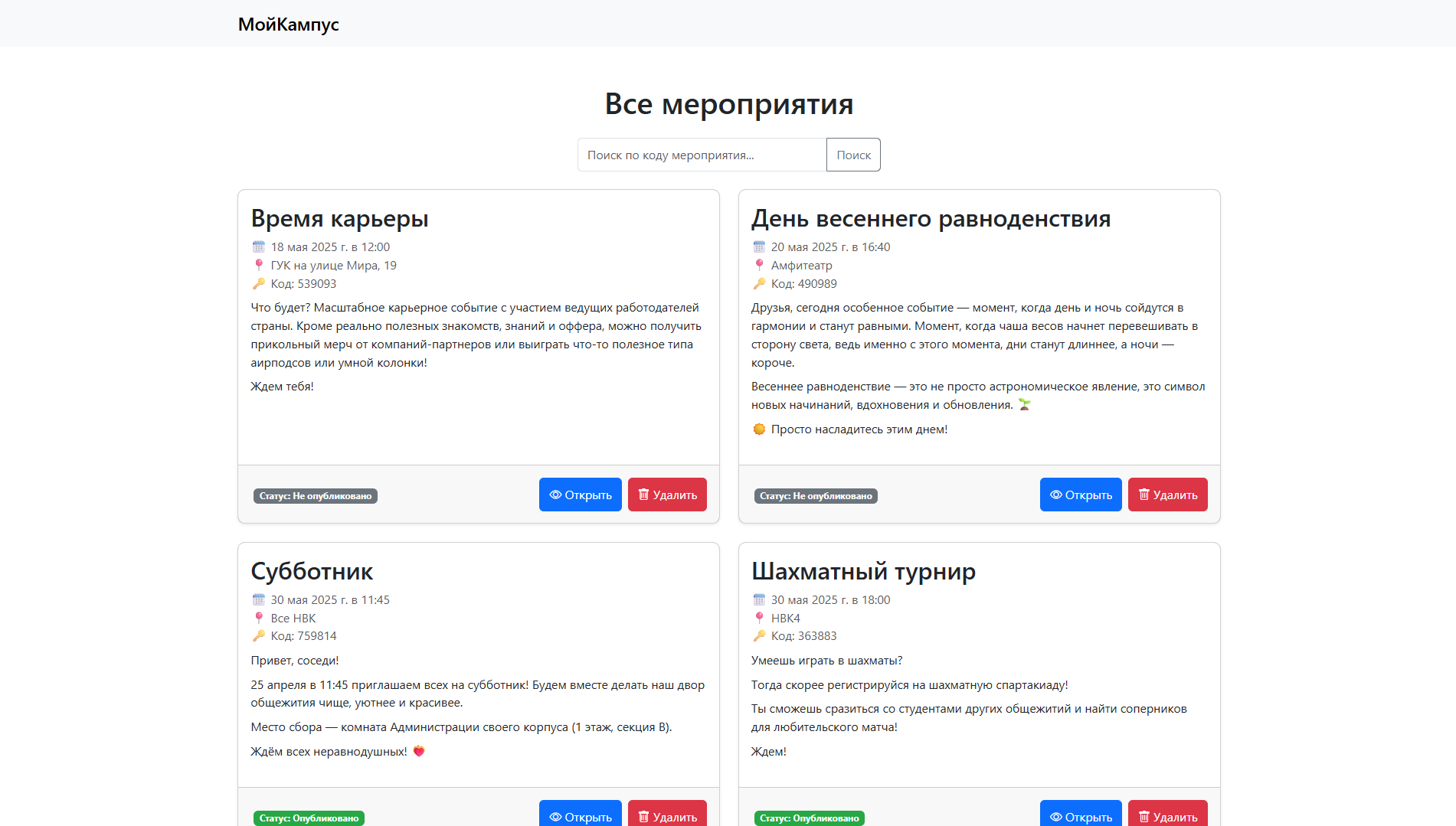


Рисунок 12 — Все мероприятия, созданные организаторами