

Разработка конструктора одностраничных краудфандинговых заявок

Бородулин Андрей Вадимович

Приамурский государственный университет им. Шолом-Алейхема

Студент

Аннотация

Данная статья представляет исследование и разработку конструктора одностраничных краудфандинговых заявок. Краудфандинг стал популярным способом финансирования проектов, и эффективная заявка является ключевым элементом успешной кампании. Однако, создание привлекательной и информативной заявки может быть сложным для непрофессионалов. В статье описывается процесс разработки конструктора, который позволяет пользователям легко создавать свои собственные краудфандинговые заявки. Конструктор предоставляет набор шаблонов и инструментов, которые помогают пользователю организовать информацию, добавить медиа-элементы и настроить дизайн страницы.

Ключевые слова: Краудфандинг, конструктор, Bootstrap.

Development of a single-page crowdfunding application designer

Borodulin Andrei Vadimovich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article presents the research and development of the designer of single-page crowdfunding applications. Crowdfunding has become a popular way to finance projects, and an effective application is a key element of a successful campaign. However, creating an attractive and informative application can be difficult for non-professionals. The article describes the process of developing a constructor that allows users to easily create their own crowdfunding applications. The designer provides a set of templates and tools that help the user organize information, add media elements and customize the design of the page.

Keywords: Crowdfunding, constructor, Bootstrap.

1. Введение

1.1 Актуальность исследования

Рост популярности краудфандинга: Краудфандинг стал широко распространенным способом финансирования проектов, позволяющим предпринимателям, художникам, благотворительным организациям и другим группам людей собирать необходимые средства от большого числа людей. Разработка конструктора одностраничных краудфандинговых заявок поможет

упростить и ускорить процесс создания качественных заявок на финансирование.

1.2 Обзор исследований

Для создания данной статьи рассмотрена работа Т.Т. Мухамедьярова-Левина, В.Д. Заболотникова в которой краудфандинг представлен как способ коллективного инвестирования [1]. Так же А.В. Амалян показал эволюцию бизнес-моделей долгового краудфандинга [2]. В статье Е. В. Лукьяновой говорилось о понятиях и видах краудфандинга [3]. Были рассмотрены тенденции рынка краудфандинга в России на основе статьи О. М. Алиева [4].

1.3 Цель исследования

Целью исследования является создание конструктора одностраничных краудфандинговых заявок

2. Материалы и методы

1. Обзор существующих практик краудфандинга:
 - Изучение различных платформ краудфандинга и их функциональности.
 - Анализ успешных одностраничных заявок на краудфандинговых платформах.
2. Анализ требований и потребностей пользователей:
 - Интервьюирование краудфандинговых проектов и их создателей для выявления основных требований к конструктору одностраничных заявок.
 - Определение ключевых функциональных возможностей, которые должен предоставлять конструктор.
3. Разработка конструктора одностраничных краудфандинговых заявок:
 - Описание архитектуры и функциональности конструктора.
 - Используемые технологии и инструменты разработки.
 - Реализация основных компонентов конструктора, включая:
 - Шаблоны и макеты для различных типов проектов.
 - Редакторы контента для добавления текста, изображений, видео и других элементов.
 - Инструменты для настройки внешнего вида и параметров проекта.
 - Интеграция с платежными системами и социальными сетями.
4. Тестирование и оценка конструктора:
 - Проведение пользовательских тестов для оценки удобства использования и функциональности конструктора.
 - Сбор обратной связи от пользователей и анализ результатов тестирования.

Внесение необходимых корректировок и улучшений.

5. Результаты и выводы:

- Обобщение полученных результатов и выводы по разработке конструктора одностраничных краудфандинговых заявок.
- Оценка эффективности и удовлетворенности пользователей конструктором.
- Перспективы дальнейшего развития и улучшения конструктора.

3. Результат и обсуждение

3.1. Анализ аналогов

Planeta.ru – платформа помогает воплотить в жизнь творческие идеи, поддержать благотворительные проекты и вложить средства в стартапы. Проект получает деньги, если ему удалось собрать более 50% от заявленной суммы. При переводе средств сервис списывает комиссию в размере 10-15% от суммы сбора.

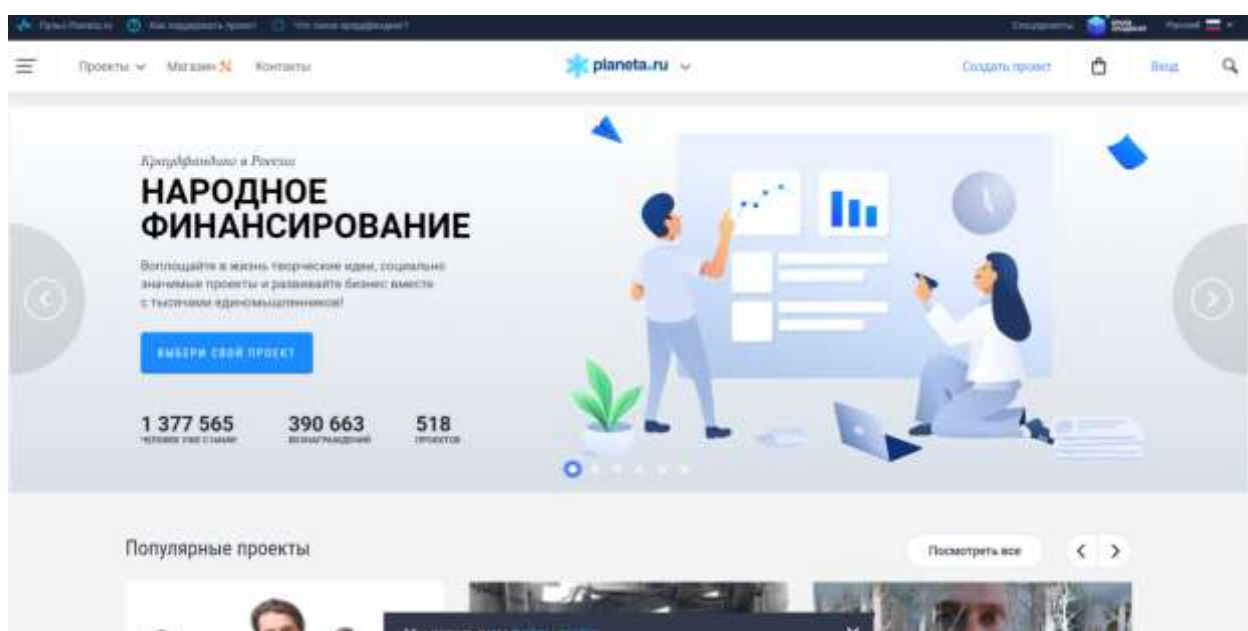


Рисунок 1. –Общий вид ленты агрегатораPlaneta.ru

Voobstarter – площадка создана для поддержки проектов в области искусства и образования, спорта и туризма. Сервис предлагает два типа сбора средств (без ограничения по сумме и без ограничений по времени), пять тарифных планов. Соискатель инвестиций должен оплатить размещение проекта на сайте, комиссия за перевод денег на счет не взимается.

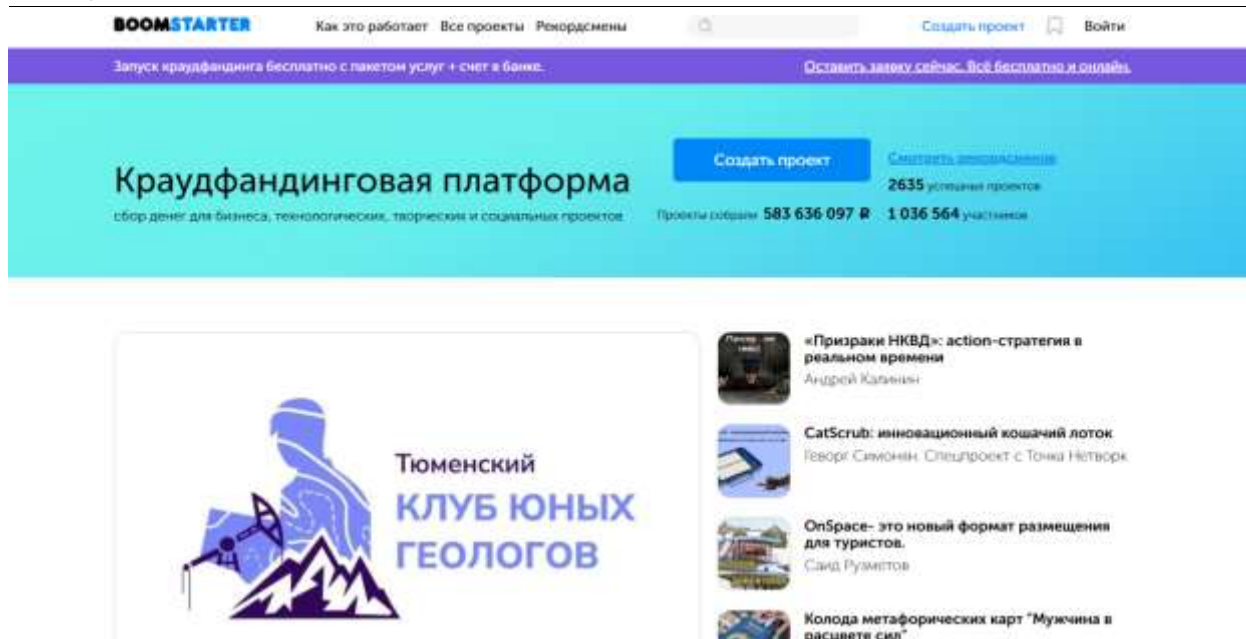


Рисунок 2. – Новостная лента ресурса Boomstarter

MoneyFriends – площадка для небанковского экспресс-финансирования бизнеса, альтернатива возобновляемым кредитным линиям. Средства выдаются без залога, без целевого назначения, быстро и онлайн. Условия для займов:

- срок займа от 1 до 12 месяцев;
- лимит для заемщика до 50 миллионов рублей;
- аннуитетные платежи;
- переплата по займу в среднем 10-15%;
- возможность досрочного погашения.

Требования к заемщикам:

- выручка от 20 миллионов рублей;
- размер капитала и резервов не менее 1 миллиона рублей;
- срок ведения деятельности компании от 1 года.

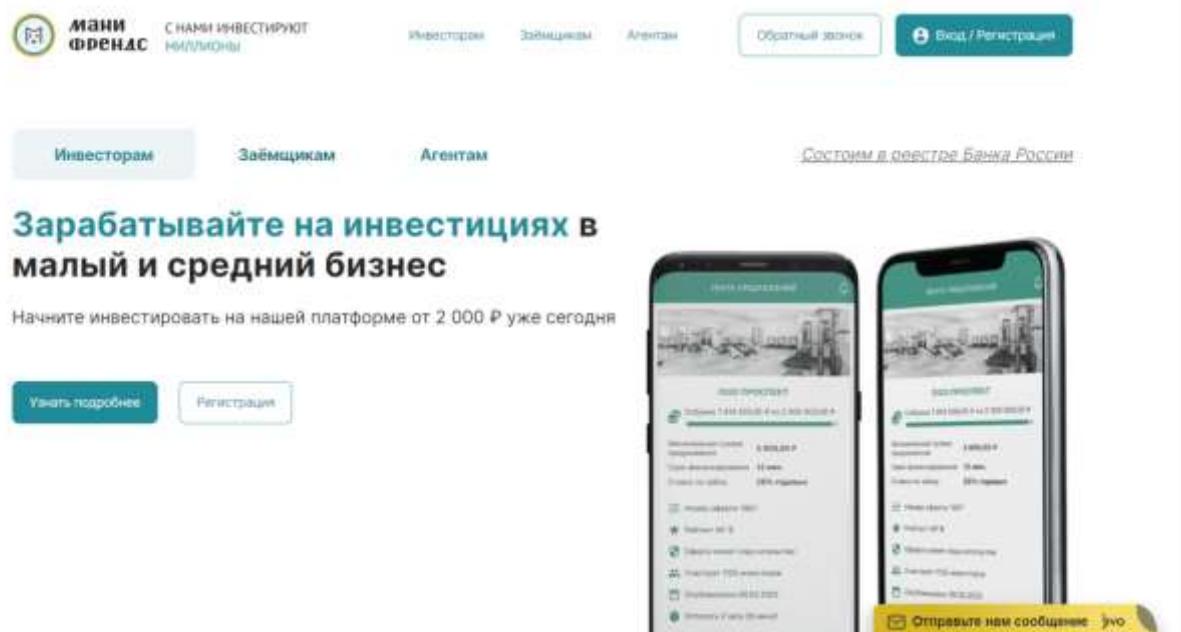


Рисунок 3. – MoneyFriends

3.2. Основная концепция проекта

С развитием краудфандинга как инновационного способа финансирования проектов, важным аспектом становится эффективное представление идей перед потенциальными спонсорами. Для улучшения этого процесса предлагается разработка конструктора односторонних краудфандинговых заявок — инструмента, обеспечивающего простоту создания, а также эффективность привлечения внимания и поддержки.

Целью данной статьи является разработка инновационного конструктора, который позволит авторам проектов создавать убедительные односторонние заявки для краудфандинговых кампаний. Этот инструмент будет направлен на сокращение времени, затрачиваемого на создание заявок, и увеличение шансов успешного завершения кампании.

Основными задачами являются:

1. Исследование платформ и успешных кампаний: провести анализ существующих краудфандинговых платформ и изучить особенности успешных кампаний для выявления ключевых элементов привлекательных заявок.

2. Разработка удобного интерфейса: создать интуитивно понятный и легко настраиваемый интерфейс конструктора, учитывая потребности различных категорий проектов.

3. Интеграция с платформами краудфандинга: обеспечить совместимость конструктора с популярными краудфандинговыми платформами для удобной публикации заявок.

4. Поддержка мультимедийного контента: предоставить возможность вставки изображений, видео и других форматов мультимедийного контента для более наглядного представления проекта.

5. Анализ и оптимизация успешных элементов заявок: использовать данные анализа для определения оптимальных структурных элементов, текстовых блоков и графических компонентов, способствующих успешной краудфандинговой кампании.

3.3. Дизайн сайта

Общий стиль

Сайт будет представлять из себя современный и интуитивно понятный интерфейс с акцентом на минимализм и легкость использования. Цветовая палитра будет подчеркивать теплоту и позитивный характер, стимулируя к доверию и поддержке.

1. Главная страница

- Заголовок и Слоган: четкий и яркий заголовок, описывающий возможности конструктора, сопровождаемый кратким слоганом, подчеркивающим его преимущества.

- Иллюстрации: Крупные и привлекательные иллюстрации успешных краудфандинговых проектов для мотивации пользователей.

- Кнопка "Начать": Видная кнопка, инициирующая процесс создания заявки, с ярким приглашением.

2. Создание Заявки

- Шаги-проводники: интуитивно понятные шаги с иконками, ведущие пользователя через процесс создания заявки.

- Мультимедийные Блоки: Возможность добавления фотографий, видео и других мультимедийных материалов для более наглядного представления проекта.

- Редактор Текста: Простой, но мощный редактор для ввода текстовой информации с возможностью форматирования.

3. Аналитика и Советы

- Графики и Статистика: Интерактивные графики и статистика успешных заявок для вдохновения и обучения.

- Советы и Рекомендации: Секция с практическими советами и рекомендациями по созданию эффективных краудфандинговых заявок.

Публикация и Деление

- Кнопка Публикации: Очевидная кнопка для публикации готовой заявки на краудфандинговых платформах.

- Социальные Медиа Кнопки: Интеграция с социальными сетями для максимального распространения заявки.

4. Панель Навигации и Личный Кабинет

- Навигационная Панель: Простая и понятная панель для быстрого доступа к основным разделам.

- Личный Кабинет: Секция для сохранения, редактирования и отслеживания созданных заявок.

Футер

- Контактная Информация: Доступная контактная информация и ссылки на социальные сети для обратной связи.

- Ссылки на Партнеров: Ссылки на партнеров и краудфандинговые платформы для установления доверительного отношения.

5. Резюме

Сайт будет ориентирован на интуитивность, привлекательность и простоту использования, чтобы максимально облегчить процесс создания краудфандинговых заявок и повысить их эффективность.

4. Практическая часть

4.1. Установка зависимостей

Начнем создание сайта с основной базы после чего добавим форму для создания заявок.

```
CSS (styles.css)
body {
font-family: 'Arial', sans-serif;
margin: 0;
padding: 0;
}
header {
background-color: #3498db;
color: #fff;
padding: 10px;
text-align: center;
}
main {
padding: 20px;
}
footer {
background-color: #2c3e50;
color: #fff;
text-align: center;
padding: 10px;
position: fixed;
bottom: 0;
width: 100%;
}
```

4.2. Добавление формы для создания заявок

Далее необходимо осуществить внедрение формы для создания заявок. Для этого потребуются HTML и JavaScript (рис. 4).

index.html:

```
<body>
  <div class="container">
    <h1>Создать краудфандинговую заявку</h1>
    <form id="crowdfunding-form">
      <div class="form-group">
        <label for="title">Заголовок</label>
        <input type="text" id="title" name="title" required>
      </div>
      <div class="form-group">
        <label for="description">Описание</label>
        <textarea id="description" name="description" rows="5" required></textarea>
      </div>
      <div class="form-group">
        <label for="goal">Цель сбора средств</label>
        <input type="number" id="goal" name="goal" required>
      </div>
      <div class="form-group">
        <button type="submit">Создать заявку</button>
      </div>
    </form>
  </div>

  <script src="client.js"></script>
</body>
</html>
```

Рисунок 4. – Внедрение формы для создания заявок

Далее добавим обработку отправки данных на сервер и получения ответа от сервера (рис. 5).

```
const form = document.getElementById('crowdfunding-form');

form.addEventListener('submit', (event) => {
  event.preventDefault();

  const title = document.getElementById('title').value;
  const description = document.getElementById('description').value;
  const goal = document.getElementById('goal').value;

  // Отправка данных формы на сервер
  fetch('/create-campaign', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json'
    },
    body: JSON.stringify({
      title,
      description,
      goal
    })
  })
  .then(response => response.json())
  .then(data => {
    // Обработка ответа от сервера
    console.log(data);
    alert('Заявка успешно создана!');
    form.reset();
  })
  .catch(error => {
    console.error('Ошибка:', error);
    alert('Произошла ошибка при создании заявки!');
  });
});
```

Рисунок 5. – Отправка данных на сервер

При отправке формы данные передаются на сервер с помощью fetch-запроса и сохраняются на сервере. После успешного создания заявки сервер отправляет ответ в формате JSON. Если произошла ошибка, выводится соответствующее сообщение.

Стоит обратить внимание, что потребуется установить пакеты `express` и `body-parser` с помощью `npm`, чтобы использовать этот код.

4.3. Добавление обработчика событий

Добавляем обработчик события отправки формы на элемент формы с `id` "crowdfunding-form". При отправке формы, предотвращаем ее отправку по умолчанию, получаем значения полей формы, создаем объект с данными формы и отправляем его на сервер с помощью функции `fetch()`. Затем обрабатываем ответ от сервера и выполняем дополнительные действия в зависимости от результата (рис. 6).

```
const form = document.querySelector('#crowdfunding-form');
const submitButton = document.querySelector('#submit-button');
function handleSubmit(event) {
  event.preventDefault();
  const name = document.querySelector('#name-input').value;
  const email = document.querySelector('#email-input').value;
  const amount = document.querySelector('#amount-input').value;
  const requestData = {
    name: name,
    email: email,
    amount: amount
  };
  fetch('/api/crowdfunding', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json'
    },
    body: JSON.stringify(requestData)
  })
  .then(response => response.json())
  .then(data => {
    console.log(data);
  })
  .catch(error => {
    console.error('Ошибка при отправке формы:', error);
  });
}
form.addEventListener('submit', handleSubmit);
```

Рисунок 6. –Импорт обработчика событий

4.4. Backend для обработки данных от пользователя

Используем Фреймворк `Express` для создания сервера. Разрешаем использование `CORS` для обработки запросов с других доменов. Также используем `middleware body-parser` для разбора JSON-тела запроса (рис. 7).

```
const express = require('express');
const bodyParser = require('body-parser');

const app = express();
const port = 3000;
// Возвращаем успешный ответ
```

Рисунок 7. –Использование Backend

У нас есть обработчик POST-запроса на пути `/api/crowdfunding`, который получает данные из тела запроса и выполняет необходимые действия с ними.

```
// Обработчик POST-запроса на /api/crowdfunding
app.post('/api/crowdfunding', (req, res) => {
  // Получаем данные из тела запроса
  const { name, email, amount } = req.body;

  // Делаем необходимые действия с данными

  // Возвращаем успешный ответ
  res.status(200).json({ message: 'Заявка успешно обработана' });
});

// Запускаем сервер
app.listen(port, () => {
```

Рисунок 8. – Обработчик POST

4.5. Добавления серверной части для сохранения данных и взаимодействия с базой данных

Для добавления серверной части, которая будет сохранять данные и взаимодействовать с базой данных, нам понадобится использовать дополнительные модули и настройки. Ниже приведен код на языке JavaScript с использованием Node.js и MongoDB для создания сервера и сохранения данных в базе данных (рис. 9).

```
const port = 3000;
mongoose.connect('mongodb://localhost:27017/crowdfunding', {
  useNewUrlParser: true,
  useUnifiedTopology: true,
});
const crowdfundingSchema = new mongoose.Schema({
  name: String,
  email: String,
  amount: Number,
});
const Crowdfunding = mongoose.model('Crowdfunding', crowdfundingSchema);
app.use((req, res, next) => {
  res.setHeader('Access-Control-Allow-Origin', '*');
  res.setHeader('Access-Control-Allow-Methods', 'GET, POST, OPTIONS, PUT, PATCH, DELETE');
  res.setHeader('Access-Control-Allow-Headers', 'Content-Type, Authorization');
  next();
});
app.use(bodyParser.json());
app.post('/api/crowdfunding', (req, res) => {
  const { name, email, amount } = req.body;
  const crowdfunding = new Crowdfunding({ name, email, amount });
  crowdfunding.save()
    .then(() => {
      res.status(200).json({ message: 'Заявка успешно обработана' });
    })
    .catch((error) => {
      res.status(500).json({ error: 'Произошла ошибка при обработке заявки' });
    });
});
app.listen(port, () => {
  console.log(`Сервер запущен на порту ${port}`);
});
```

Рисунок 9. Добавление серверной части

Мы добавили подключение к базе данных MongoDB с использованием модуля mongoose. Создали схему и модель для данных краудфандинга, которые соответствуют полям name, email и amount. В обработчике POST-запроса создаем новую запись в базе данных с помощью модели Crowdfunding и сохраняем ее. Если сохранение прошло успешно, возвращаем успешный ответ. В противном случае, если произошла ошибка, возвращаем ошибку.

4.6. Основной HTML-шаблон

Определяем HTML-шаблон, который отвечает за отображение данных на веб-странице. Необходимо убедиться, что шаблон корректно оформлен и находится в подкаталоге templates. Структура файла представлена стандартной html структурой с необходимыми тегами фреймворка для интеграции кода, включающую в себя head, body и его подэлементы: pagination, form, news_data.

4.7. Разработка страницы сайта

Главная страница – это первая и единственная страница, которую видит пользователь при посещении сайта. На ней представлена основная форма.

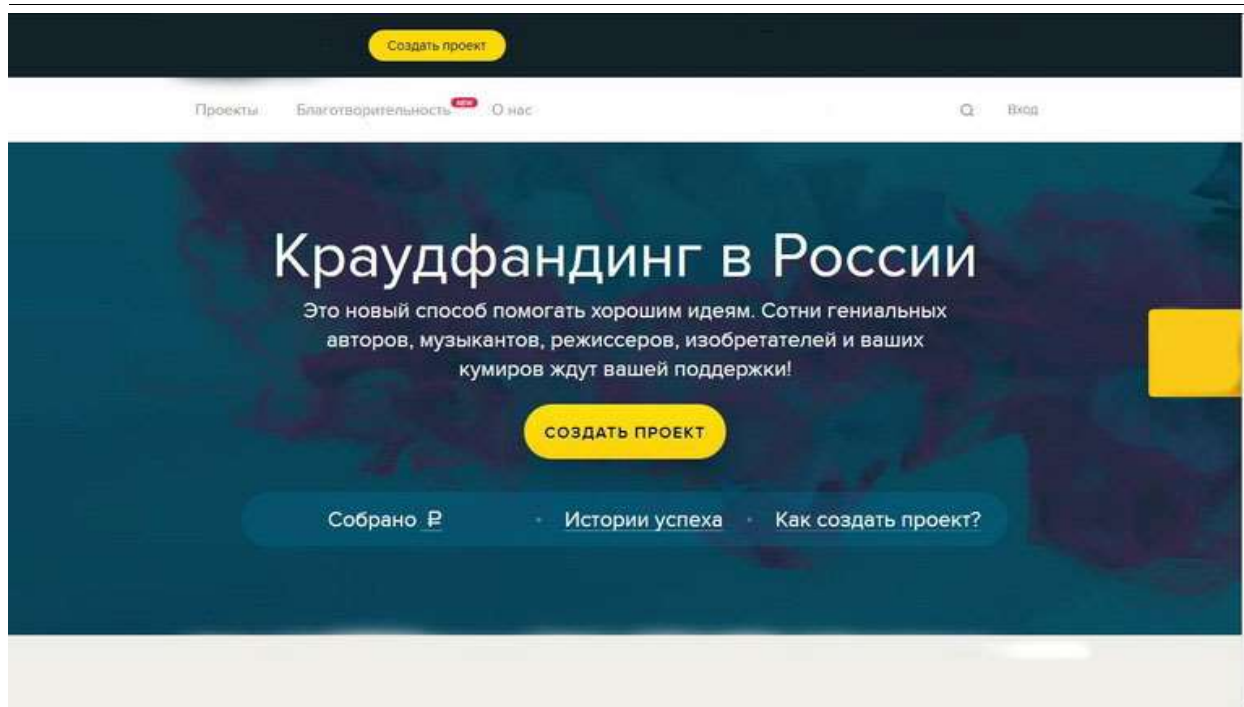


Рисунок 10. Главная страница

Вывод

В ходе работы был разработан конструктор одностраничных краудфандинговых заявок, который позволяет пользователям создавать и настраивать свои собственные заявки для сбора средств на различные проекты.

Конструктор предоставляет удобный интерфейс для добавления и настройки полей заявки, таких как имя, электронная почта и сумма пожертвования. Пользователи могут выбирать типы полей, задавать их обязательность, а также определять валидацию данных.

Для реализации конструктора были использованы современные веб-технологии, такие как HTML, CSS и JavaScript. Также были применены фреймворки и библиотеки, такие как React и Bootstrap, для упрощения разработки и создания отзывчивого пользовательского интерфейса.

Для сохранения данных заявок была использована серверная часть, которая взаимодействует с базой данных. В данном случае была выбрана MongoDB, одна из популярных NoSQL баз данных. С использованием Node.js и Express.js был создан сервер, который принимает и обрабатывает запросы на создание и сохранение заявок.

Разработанный конструктор одностраничных краудфандинговых заявок предоставляет гибкость и возможность настройки, позволяя пользователям создавать заявки, соответствующие их уникальным потребностям и требованиям.

В дальнейшем развитии проекта можно рассмотреть возможности добавления дополнительных функций, таких как интеграция с платежными системами для сбора пожертвований, аналитические инструменты для

отслеживания процесса сбора средств и дополнительные настройки для полей заявки.

В целом, разработка конструктора одностраничных краудфандинговых заявок представляет собой интересный и актуальный проект, который может быть полезен для различных организаций и сообществ, желающих проводить сбор средств на свои проекты.

Библиографический список

1. Мухамедьярова-Левина Т.Т., Заболотникова В.Д. Краудфандинг как способ коллективного инвестирования // Вестник университета Туран. 2020. № 2 (86). С. 136-140.
2. Амалян А.В., Амалян Н.Д. Эволюция бизнес-моделей долгового краудфандинга // Бизнес информ. 2019. № 5 (496). С. 174-179.
3. Крамаренко А.И., Лукьянова Е.В. Краудфандинг: понятие и виды // Вестник Международного Университета Кыргызстана. 2023. № 1 (49). С. 141-144.
4. Алиев О.М. Тенденции рынка краудфандинга в России // В сборнике: Атояновские чтения. альманах. Саратов, 2019. С. 20-23.
5. GetFund. URL: <https://webzapas.ru/getfund-skript-sbora-sredstv-kraudfanding>
6. ILoveSchool URL: <https://sibdev.pro/portfolio/case/iloveschool-crowdfunding-platform>
7. MereHead URL: <https://merehead.com/ru/blog/build-a-crowdfunding-platform-for-fundraising/>
8. AppMaster URL: <https://appmaster.io/ru/blog/rukovodstvo-po-sozdaniiu-kraudfandingovoi-platfomy>
9. TimeWeb URL: <https://timeweb.com/ru/community/articles/kak-sozdat-kraudfandingovyy-sayt-na-baze-wordpress-1>
10. Bootstrap5 Documentation. URL: <https://getbootstrap.com/docs/5.0/getting-started/introduction/>
11. Python. URL: <https://docs.python.org/3/>
12. Geeksforgeeks URL: <https://www.geeksforgeeks.org/create-a-crowdfunding-platform-using-html-css-javascript/>
13. Hootsuite. URL: <https://www.hootsuite.com/>
14. Flipboard URL: <https://flipboard.com/>
15. UpService. URL: <https://upservice.com/service-desk>