1. **Аналитическая часть**
   1. **Характеристика предметной области**
      1. Данный проект посвящен созданию веб-приложения – партнерского сайта для всероссийской экологической акции «Бумбатл» в рамках деятельности «Детского технопарка «Кванториум» в городе Кургане.
      2. Кванториум – Автономная некоммерческая организация «Детский технопарк «Кванториум», основанная 5 февраля 2016 года, реализует инновационную модель дополнительного образования для детей в соответствии с инициативой Агентства стратегических инициатив. Кванториум в Кургане представляет собой успешный пример государственно-частного партнерства, учредителями которого являются Республика Татарстан (ГАУ «Технопарк в сфере высоких технологий «ИТ-парк») и ПАО «КАМАЗ».
      3. Кванториум предлагает широкий спектр образовательных программ, охватывающих различные научно-технические направления, такие как VR/AR-квантум, Автоквантум, Аэроквантум, ИТ-квантум, Геоквантум, Космоквантум, Наноквантум, Нейроквантум, а также программы по промышленной роботехнике и 3D-моделированию. На постоянной основе в Кванториуме обучается 598 школьников 5-11 классов, а ежегодно образовательные программы охватывают более 1700 детей посредством мастер-классов и экскурсий. Кванториум также активно занимается методическим обеспечением и координацией деятельности других образовательных организаций в сфере научно-технического творчества и робототехники.
      4. Акция «Бумбатл» - Всероссийская экологическая акция, направленная на привлечение внимания к проблеме переработки макулатуры и повышению экологической ответственности у детей и взрослых. Партнёрский сайт, разрабатываемый в рамках данного проекта, будет служить платформой для информирования об акции, регистрации участников, публикации результатов и взаимодействия с организаторами. Интеграция с Кванториумом позволяет использовать сайт как инструмент для распространения экологических знаний и привлечение учащихся к активной деятельности по сохранению окружающей среды. В рамках этой интеграции, сайт будет тесно связан с образовательными программами Кванториума, способствуя целостному подходу к формированию экологической культуры у подрастающего поколения.
   2. **Характеристика комплекса задач и обоснование необходимости создания веб-приложения**
      1. Разработка веб-приложения – партнёрского сайта для акции «Бумбатл» в рамках деятельности «Детского технопарка «Кванториум» - обусловлена необходимостью эффективной организации и проведения этой всероссийской экологической инициативы.
      2. Проект направлен на решение следующих задач:
         * Повышение осведомлённости – предоставление исчерпывающей информации об акции «Бумбатл», её целях и правилах участия, а также о важности переработки макулатуры и бережного отношения к окружающей среде. Это будет достигаться за счёт интуитивно понятного интерфейса и доступного представления информации;
         * Мониторинг прогресса – веб-приложение обеспечит возможность отслеживания индивидуальных и командных результатов участников акции, стимулируя их активное участие и соревновательный дух;
         * Привлечение участников – создание удобной платформы для регистрации участников акции, как из числа учащихся образовательной организации, так и из более широкого круга детей и подростков. Система регистрации должна быть простой и быстрой, минимизируя временные затраты;
         * Интеграция с системой сбора макулатуры – ключевой элемент проекта – интеграция с сервисом recyclemap.ru, представляющим интерактивную карту пунктов сбора макулатуры. Это позволит участникам легко находить ближайшие точки сбора и упростит процесс сдачи макулатуры;
         * Образовательный аспект – сайт будет служить дополнительным образовательным ресурсом, интегрированным с программами Кванториума. Он может содержать образовательные материалы по теме экологии и переработки отходов, а также информацию о деятельности Кванториума;
         * Создание сообщества – платформа будет способствовать формированию сообщества участников акции, позволяя им обмениваться опытом, идеями и достижениями;
         * Обоснование выбора веб-приложения – разработка веб-приложения является наиболее эффективным решением для достижения поставленных целей. Веб-доступность позволяет участникам из любого региона принять участие в акции в любое удобное время. Интерактивные элементы сайта, такие как интерактивная карта пунктов сбора макулатуры и система отслеживания результатов, повышают уровень вовлеченности и интереса участников. Интеграция с образовательными программами Кванториума делает сайт не только инструментом для участия в акции, но и важным образовательным ресурсом. Выбор веб-приложения обеспечивает масштабируемость и доступность проекта для большого числа участников.
   3. **Анализ существующих программных средств для создания веб-приложения**
      1. Для разработки веб-приложения для акции «Бумбатл» были рассмотрены следующие программные средства, с учетом требований к производительности, масштабируемости и безопасности.

Таблица 1

| № | Характеристика | Laravel 12 | Django | Express.js | Ruby on Rails |
| --- | --- | --- | --- | --- | --- |
| 1 | Встроенная аутентификация | + | + | - | + |
| 2 | Мощный ORM | + | + | - | + |
| 3 | Встроенная защита (CSRF, XSS) | + | + | - | + |
| 4 | Готовая админ-панель | - | + | - | + |
| 5 | Гибкость маршрутизации | + | - | + | - |
| 6 | Легковесность | - | - | + | - |
| 7 | Производительность | + | - | + | - |
| 8 | Большое сообщество | + | + | + | + |
| 9 | Хорошая документация | + | + | + | + |
| 10 | Простота интеграции с фронтендом | + | + | + | + |
| 11 | CLI для автоматизации задач | + | - | - | + |
| 12 | Подходит для небольших проектов | - | - | + | - |
| 13 | Быстрый старт разработки | + | + | - | + |
| 14 | Зависимость от фреймворка | + | + | - | + |
| 15 | Требует ручной настройки | - | - | + | - |

Таблица 2

| № | Характеристика | Vite | Webpack | Parcel | Rollup |
| --- | --- | --- | --- | --- | --- |
| 1 | Быстрая разработка (HMR) | + | - | + | - |
| 2 | Оптимизированная production-сборка | + | + | - | + |
| 3 | Простота конфигурации | + | - | + | - |
| 4 | Поддержка TypeScript | + | + | + | + |
| 5 | Большая экосистема плагинов | - | + | - | - |
| 6 | Проверенная стабильность | - | + | ± | + |
| 7 | Гибкость настройки | ± | + | - | + |
| 8 | Нулевая/минимальная конфигурация | - | - | + | - |
| 9 | Хорошая оптимизация бандла | + | + | ± | + |
| 10 | Поддержка ES-модулей | + | ± | + | + |
| 11 | Легковесность | + | - | + | + |
| 12 | Совместимость с Laravel | + | + | ± | - |
| 13 | Подходит для сложных проектов | ± | + | - | + |
| 14 | Поддержка старых браузеров | ± | + | + | - |

Таблица 3

| № | Характеристика | Bootstrap | Material UI | Chakra UI | Bulma |
| --- | --- | --- | --- | --- | --- |
| 1 | Утилитарный подход (гибкость) | + | - | + | - |
| 2 | Готовые компоненты | + | + | + | + |
| 3 | Адаптивный дизайн | + | + | + | + |
| 4 | Легкая кастомизация | + | - | + | ± |
| 5 | Большое сообщество | + | + | - | ± |
| 6 | Хорошая документация | + | + | + | + |
| 7 | Маленький размер CSS | - | - | + | + |
| 8 | Material Design | - | + | - | - |
| 9 | Доступность (a11y) | ± | + | + | ± |
| 10 | Простота использования | ± | + | + | + |
| 11 | Консистентный дизайн | ± | + | + | + |
| 12 | Подходит для сложных проектов | + | + | ± | - |
| 13 | Активное развитие | + | + | + | - |

Таблица 4

| № | Характеристика | SQLite 3 | MySQL | PostgreSQL | MongoDB |
| --- | --- | --- | --- | --- | --- |
| 1 | Не требует отдельного сервера | + | - | - | - |
| 2 | Простота в использовании | + | ± | - | ± |
| 3 | Встроенная поддержка в Laravel | + | + | + | + |
| 4 | Легкость в развертывании | + | - | - | ± |
| 5 | Не требует настройки | + | - | - | - |
| 6 | Подходит для небольших проектов | + | + | + | + |
| 7 | Хорошая производительность | - | + | + | + |
| 8 | Большое сообщество | ± | + | + | + |
| 9 | Много хостинг-провайдеров | - | + | ± | + |
| 10 | Мощная функциональность | - | - | + | + |
| 11 | Поддержка сложных запросов | - | ± | + | - |
| 12 | Гибкая схема данных | - | - | - | + |
| 13 | Хорошая масштабируемость | - | ± | + | + |

* + 1. Стек Laravel 12 + Bootstrap CSS + SQLite 3 был выбран благодаря оптимальному сочетанию производительности, скорости разработки и лёгкости поддержки: Laravel 12 обеспечивает быструю разработку с элегантным синтаксисом, встроенной безопасностью и удобной работой с данными через Eloquent ORM; Bootstrap CSS предоставляет готовые адаптивные компоненты с простой кастомизацией; SQLite 3 позволяет работать без сложной настройки сервера, оставаясь идеальным решением для небольших проектов. Этот стек требует минимальных ресурсов, сокращает время выхода на рынок и идеально подходит для MVP, админ-панелей, корпоративных порталов и учебных проектов.
  1. **Функции и параметры программных средств, используемых для решения задач**
     1. В рамках разработки веб-приложения для акции «Бумбатл» используются следующие программные средства и технологии:
     2. Laravel 12.
        + Роль: PHP-фреймворк для разработки серверной части приложения;
        + Функции:

Маршрутизация запросов;

Управление базой данных через ORM Eloquent;

Аутентификация и авторизация пользователей;

Обработка форм и валидация данных;

Шаблонизация через Blade.

* + - * Параметры:

Способы структурирования программ: MVC архитектура;

Встроенные элементы языка: Middleware, Service Providers, Facades;

Средства обмена данными: RESTful API, JSON responses.

* + 1. Vite.
       - Роль: Сборщик фронтенд-ресурсов и инструмент разработки;
       - Функции:

Компиляция и минификация JavaScript и CSS;

Горячая перезагрузка при разработке;

Оптимизация изображений;

Управление зависимостями.

* + - * Параметры:

Способы структурирования программ: Модульная система ES;

Встроенные элементы языка: Import/Export синтаксис;

Средства обмена данными: WebSocket для HMR.

* + 1. Bootstrap 5.
       - Роль: CSS-фреймворк для создания адаптивного интерфейса;
       - Функции:

Сетка для верстки;

Готовые компоненты интерфейса;

Утилиты для типографики и стилизации;

Интерактивные элементы.

* + - * Параметры:

Способы структурирования программ: Компонентный подход;

Встроенные элементы языка: CSS-классы и переменные;

Средства обмена данными: JavaScript события.

* + 1. SQLite 3.
       - Роль: Встраиваемая реляционная база данных;
       - Функции:

Хранение данных пользователей;

Управление сессиями;

Кэширование запросов;

Транзакционная обработка.

* + - * Параметры:

Способы структурирования программ: SQL-запросы;

Встроенные элементы языка: SQLite диалект SQL;

Средства обмена данными: PDO, Eloquent ORM.

Таблица 5

| № | Название элемента | Пример использования | Назначение и применение |
| --- | --- | --- | --- |
| 1 | Маршрут | Route::get('/home', [HomeController::class, 'index']) | Определяет обработчик для конкретного URL-адреса |
| 2 | Миграция | Schema::create('users', function (Blueprint $table) { $table->id(); }) | Создает и изменяет структуру таблиц в базе данных |
| 3 | Модель | class User extends Model { protected $fillable = ['name']; } | Представляет данные из таблицы БД в виде объектов |
| 4 | Контроллер | class UserController extends Controller { public function index() {} } | Содержит логику обработки запросов |
| 5 | Middleware | class CheckAge { public function handle($request, $next) {} } | Фильтрует HTTP-запросы перед обработкой |
| 6 | Blade-шаблон | @section('content')...@endsection | Шаблонизация HTML-страниц с динамическим контентом |
| 7 | Компонент Bootstrap | <div class="alert alert-success">...</div> | Готовые UI-компоненты для быстрой верстки |
| 8 | SQL-запрос | DB::select('SELECT \* FROM users WHERE active = ?', [1]) | Выполнение запросов к базе данных |
| 9 | Vite конфигурация | export default { plugins: [laravel()] } | Настройка сборки фронтенд-ресурсов |
| 10 | API-эндпоинт | Route::apiResource('users', UserController::class) | Создание RESTful API интерфейса |

* 1. **Техническое задание**
     1. Общие требования.
        + Название приложения: "Бумбатл & Кванториум";
        + Доменное имя: bumbattle-kvantorium.ru;
        + Поддерживаемые языки: Русский язык;
        + Адаптивный дизайн:

Десктоп (1920px и выше);

Планшет (768px - 1919px);

Мобильные устройства (320px - 767px).

* + - * Производительность:

Время загрузки страниц не более 3 секунд;

Оптимизация изображений;

Кэширование статического контента.

* + 1. Ролевая модель.

Таблица 6

| № | Роль | Возможности и полномочия | Создается кем |
| --- | --- | --- | --- |
| 1 | Незарегистрированный пользователь | Просмотр публичного контента Просмотр карты пунктов приема Возможность регистрации | Автоматически системой |
| 2 | Зарегистрированный пользователь | Авторизация в системе Управление личным профилем Участие в акциях Загрузка результатов сдачи отходов | Самостоятельная регистрация через форму |
| 3 | Модератор | Проверка загруженных материалов Управление публичным контентом Обработка заявок пользователей Блокировка нарушителей | Назначается администратором |
| 4 | Администратор | Полный доступ к системе Управление всеми пользователями Настройка системных параметров Назначение модераторов Аналитика и отчетность | Главным администратором системы |

* + 1. Функциональные требования к публичной части.
       - Главная страница:

Блок статистики;

Общее количество собранной макулатуры;

Количество участников;

Количество активных команд;

Экологический эффект.

* + - * Информационный блок:

Описание акции;

Правила участия;

Сроки проведения;

Призы и награды.

* + - * Новостная лента:

Последние новости;

Анонсы мероприятий;

Истории успеха.

* + - * Карта пунктов приема:

Интеграция с recyclemap.ru;

Фильтрация по типам пунктов;

Построение маршрута;

Информация о режиме работы.

* + 1. Функциональные требования к личному кабинету клиента.
       - Управление профилем:

Просмотр и редактирование личных данных;

ФИО;

Email;

Телефон;

Фотография профиля.

* + - * Настройки безопасности:

Изменение пароля.

* + - * Управление командой:

Создание команды;

Название команды;

Описание;

Логотип.

* + - * Управление участниками:

Приглашение новых участников;

Удаление участников;

Назначение ролей;

Передача управления командой.

* + - * Командная статистика:

Общий вес собранной макулатуры;

Количество участников;

Рейтинг в общем зачете;

История достижений.

* + - * Работа с результатами:

Загрузка результатов;

Форма отчета;

Загрузка фотографий;

Загрузка документов;

Указание даты сдачи.

* + - * История результатов:

Список всех сдач;

Статус проверки;

Комментарии модератора;

История изменений.

* + - * Статистика:

Сравнение с другими командами;

Прогресс по целям;

Достижения и награды.

* + - * Работа с картой:

Поиск пунктов приема;

Поиск по адресу;

Фильтрация по типу;

Поиск по режиму работы;

Избранные пункты.

* + - * Информация о пунктах:

Подробное описание;

Режим работы;

Контакты;

Отзывы и рейтинг.

* + - * Образовательные материалы:

Доступ к материалам;

Видео-уроки;

Текстовые материалы;

Инфографика;

Тесты и задания.

* + - * Прогресс обучения:

Пройденные материалы;

Достижения.

* + 1. Функциональные требования к административной части
       - Панель управления:

Управление пользователями;

Создание/редактирование/удаление;

Управление ролями;

Блокировка аккаунтов.

* + - * Управление контентом:

Публикация новостей;

Модерация материалов;

Управление страницами.

* + - * Статистика и отчеты:

Общая статистика;

Детализация по регионам;

Экспорт данных.

* + 1. Архитектура сайта

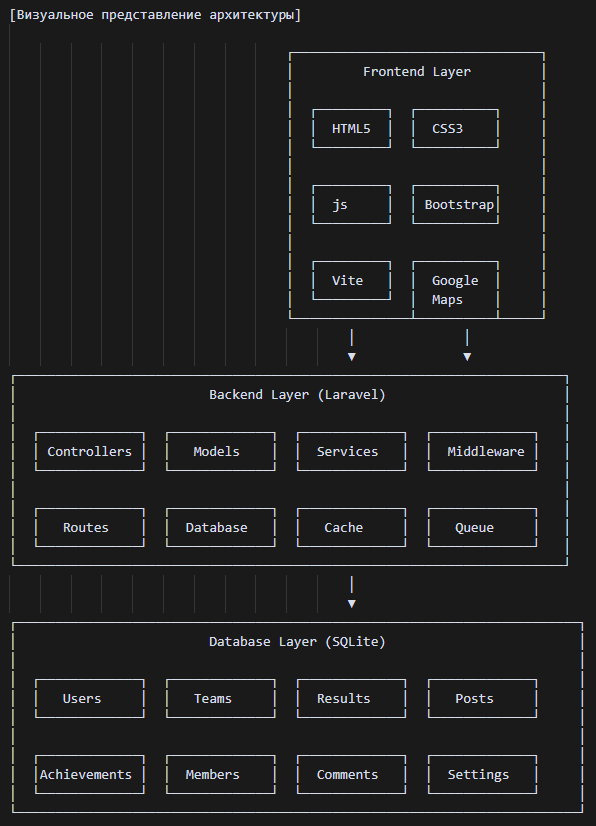


Рисунок 1 – Архитектурное представление

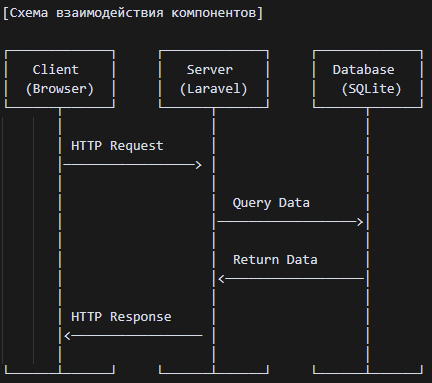


Рисунок 2 – Схема взаимодействия

* + 1. Технические требования
       - Требования к платформе разработки:

Операционная система: Windows 11/10/8.1, macOS 10.15 и выше, Linux (Ubuntu 20.04 LTS и выше);

Средства разработки: PHP 8.2 или выше, Composer 2.0 или выше, Node.js 16.x или выше, npm 8.x или выше, Git 2.x или выше.

* + - * Требования к технологиям:

Серверная часть: Laravel 12.x, SQLite 3, Redis (для кэширования);

PHP Extensions: PDO, SQLite3, GD/ImageMagick, OpenSSL, Mbstring, JSON, XML;

Клиентская часть: HTML5, CSS3, JavaScript (ES6+), Bootstrap 5.3.3, Vite 6.0.11, Vue.js 3.x (для компонентов).

* + - * Требования к техническим средствам:

Серверные требования: Процессор: 2 ядра и выше, оперативная память: минимум 512 МБ, дисковое пространство: минимум 1 ГБ, поддержка SSL/TLS, доступ к интернету;

Клиентские требования: Современный веб-браузер, JavaScript включен, Cookies включены, минимальное разрешение экрана: 320x480, поддержка CSS Grid и Flexbox.

* + - * Требования к адаптивности:

Брейкпоинты: мобильные устройства: 320px - 767px, планшеты: 768px - 1023px, десктопы: 1024px и выше;

Особенности адаптации: гибкая сетка (Grid System), отзывчивые изображения, адаптивная типографика, оптимизированная навигация, Touch-friendly интерфейс.

* + - * Требования к кроссбраузерности:

Поддерживаемые браузеры: Google Chrome (последние 2 версии), Mozilla Firefox (последние 2 версии), Microsoft Edge (последние 2 версии), Safari (последние 2 версии), Opera (последние 2 версии).

* + - * Требования к отображению:

Корректное отображение всех элементов;

Работа всех интерактивных компонентов;

Поддержка современных CSS-свойств;

Совместимость с ES6+ JavaScript;

Корректная работа форм.

* + - * Требования к безопасности:

Общие требования: HTTPS протокол, защита от XSS-атак, защита от CSRF-атак, защита от SQL-инъекций, Rate limiting;

Аутентификация и авторизация: безопасное хранение паролей, двухфакторная аутентификация, защита от брутфорс-атак, безопасные сессии, JWT токены для API.

* + - * Требования к производительности:

Время загрузки: первая загрузка страницы: не более 3 секунд, последующие загрузки: не более 1 секунды, время отклика API: не более 200мс;

Оптимизация: минификация CSS/JS, оптимизация изображений, кэширование статического контента, ленивая загрузка, код-сплиттинг.

* 1. **Организация интерфейса пользователя**
     1. Для создания эффективного и удобного пользовательского интерфейса веб-приложения были использованы современные принципы UI/UX дизайна и методологии проектирования.

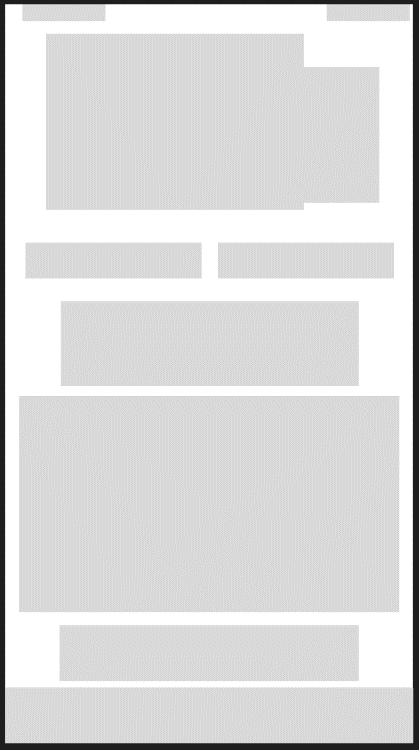


Рисунок 3 – Экран приложения 1920х1080

1. **Проектная часть**
   1. **Разработка дизайна сайта**
      1. Сервисы, использовавшиеся для подбора шрифтов и цветов. Для создания визуального стиля веб-приложения были использованы следующие сервисы:
         * Adobe Color:

Создание цветовых схем;

Проверка контрастности;

Генерация палитр;

Анализ доступности цветов.

* + - * Google Fonts:

Подбор шрифтов;

Анализ сочетаемости;

Оптимизация загрузки;

Проверка поддержки языков;

* + - * Coolors:

Генерация цветовых палитр;

Проверка сочетаемости;

Экспорт в различные форматы;

Создание градиентов.

* + 1. Выбор цветов
       - Основная цветовая палитра:

Основной цвет: #3f865c (Зеленый);

;

Применение: кнопки успешных действий, элементы успешного статуса, основные акценты интерфейса, индикаторы прогресса.

Дополнительный цвет: #351e03 (Коричневый);

;

Применение: информационные кнопки, элементы навигации, второстепенные акценты, статусные индикаторы.

Акцентный цвет: #f5cb45 (Желтый);

;

Применение: кнопки предупреждений, элементы внимания, временные уведомления, статусы ожидания.

Фоновый цвет: #d2e4bc (Светло-зеленый);

;

Применение: основной фон страниц, фон карточек, фон секции, фон модальных окон.

Фоновый цвет: #292121 (Темно-коричневый);

;

Применение: основной текст, заголовки, описания, метаданные.

* + 1. Типографика
       - Основной шрифт: Roboto:

Заголовки: h1: 32px, Regular (400), межстрочный интервал: 1.2, отступ снизу: 24px; h2: 24px, Regular (400), межстрочный интервал: 1.3, отступ снизу: 20px; h3: 20px, Medium (500), межстрочный интервал: 1.4, отступ снизу: 16px.

* + - * Основной текст: размер: 16px, начертание: Light (300), межстрочный интервал: 1.5, цвет: #333333.
      * Вспомогательный текст: размер: 14px, начертание: Regular (400), межстрочный интервал: 1.4, цвет: #666666.
      * Кнопки и интерактивные элементы: размер: 18px, начертание: Medium (500), трансформация: uppercase, межбуквенный интервал: 0.5px.
    1. Обоснование выбора элементов дизайна
       - Обоснование выбора цветовой палитры:

Основной цвет (#3f865c - Зеленый): выбран как символ экологичности и природы, соответствует тематике акции по сбору макулатуры, имеет хорошую контрастность с белым фоном, воспринимается как положительный и активный цвет, соответствует брендингу экологических инициатив.

Дополнительный цвет (#351e03 - Коричневый): создает гармоничное сочетание с основным зеленым, ассоциируется с технологичностью и надежностью, обеспечивает хорошую читаемость текста, используется для второстепенных элементов, не конкурируя с основным цветом, соответствует современным трендам веб-дизайна.

Акцентный цвет (#f5cb45 - Желтый): привлекает внимание к важным элементам, создает контраст с основными цветами, используется для призывов к действию, ассоциируется с энергией и активностью, улучшает визуальную иерархию интерфейса.

Фоновый цвет (#d2e4bc - Светло-зеленый): обеспечивает максимальную читаемость контента, создает ощущение чистоты и порядка, соответствует современным трендам минималистичного дизайна, уменьшает нагрузку на глаза при длительной работе, позволяет другим цветам выступать на переднем плане.

* + 1. Обоснование выбора типографики:

1. Шрифт Roboto:
2. Разработан специально для веб-интерфейсов
3. Имеет отличную читаемость на всех устройствах
4. Поддерживает множество начертаний
5. Оптимизирован для экранов
6. Широко распространен и хорошо поддерживается браузерами
7. Размеры заголовков:
8. h1 (32px):
9. Достаточно крупный для главных заголовков
10. Хорошо выделяется на странице
11. Соответствует иерархии контента
12. Оптимален для SEO
13. Обеспечивает хорошую читаемость
14. h2 (24px):
15. Создает четкую визуальную иерархию
16. Достаточно заметен, но не конкурирует с h1
17. Подходит для подзаголовков разделов
18. Хорошо сочетается с основным текстом
19. Оптимален для мобильных устройств
20. h3 (20px):
21. Идеален для заголовков карточек и блоков
22. Сохраняет читаемость при уменьшении
23. Хорошо сочетается с другими элементами
24. Подходит для навигационных элементов
25. Оптимален для плотного контента
26. Основной текст (16px):
27. Оптимальный размер для чтения на экране
28. Соответствует стандартам веб-типографики
29. Хорошо масштабируется на разных устройствах
30. Обеспечивает комфортное чтение
31. Поддерживает доступность (WCAG)
32. Вспомогательный текст (14px):
33. Достаточно мелкий для второстепенной информации
34. Сохраняет читаемость
35. Экономит место на экране
36. Создает визуальную иерархию
37. Подходит для подписей и метаданных
38. Обоснование выбора сервисов:
39. Adobe Color:
40. Профессиональный инструмент для работы с цветом
41. Позволяет создавать гармоничные цветовые схемы
42. Обеспечивает соответствие стандартам доступности
43. Предоставляет инструменты для анализа контраста
44. Позволяет экспортировать цвета в различные форматы
45. Google Fonts:
46. Бесплатный и надежный сервис
47. Обеспечивает быструю загрузку шрифтов
48. Предоставляет широкий выбор шрифтов
49. Поддерживает оптимизацию для веб
50. Имеет хорошую документацию
51. Coolors:
52. Удобный инструмент для быстрого подбора цветов
53. Позволяет создавать и сохранять палитры
54. Предоставляет инструменты для создания градиентов
55. Поддерживает экспорт в различные форматы
56. Имеет интуитивно понятный интерфейс
57. **Построение логической структуры сайта**
58. Логическая структура сайта представляет собой иерархическую организацию информации и функциональных элементов веб-приложения. Структура разработана с учетом удобства навигации и доступности всех функций для различных групп пользователей.
59. { Вставить карту сайта }
60. **Описание логики сценариев использования программного продукта**
61. В данном разделе описываются основные сценарии использования веб-приложения для акции "Бумбатл", включая последовательность действий пользователей и обработку данных системой.
62. { Вставить Диаграмму последовательностей }
63. **Построение физической структуры веб-приложения**
64. В данном разделе описывается физическая структура веб-приложения, включая организацию файлов, базы данных и компонентов системы.
65. Структура директорий проекта:
66. /
67. ├── app/
68. │ ├── Http/
69. │ │ ├── Controllers/
70. │ │ │ ├── AuthController.php # Контроллер аутентификации
71. │ │ │ ├── TeamController.php # Контроллер управления командами
72. │ │ │ ├── ResultController.php # Контроллер результатов
73. │ │ │ ├── PostController.php # Контроллер новостей
74. │ │ │ └── AdminController.php # Контроллер админ-панели
75. │ │ ├── Middleware/
76. │ │ │ ├── Authenticate.php # Middleware аутентификации
77. │ │ │ ├── VerifyRole.php # Middleware проверки ролей
78. │ │ │ └── ValidateTeam.php # Middleware валидации команды
79. │ │ └── Requests/
80. │ │ ├── TeamRequest.php # Валидация данных команды
81. │ │ └── ResultRequest.php # Валидация данных результатов
82. │ ├── Models/
83. │ │ ├── User.php # Модель пользователя
84. │ │ ├── Team.php # Модель команды
85. │ │ ├── TeamMember.php # Модель участника команды
86. │ │ ├── Result.php # Модель результатов
87. │ │ └── Post.php # Модель новостей
88. │ ├── Services/
89. │ │ ├── AuthService.php # Сервис аутентификации
90. │ │ ├── TeamService.php # Сервис управления командами
91. │ │ ├── ResultService.php # Сервис обработки результатов
92. │ │ └── FileUploadService.php # Сервис загрузки файлов
93. │ └── Providers/
94. │ └── AppServiceProvider.php # Основной провайдер приложения
95. ├── config/
96. │ ├── app.php # Основные настройки приложения
97. │ ├── auth.php # Настройки аутентификации
98. │ ├── database.php # Настройки базы данных
99. │ └── filesystems.php # Настройки файловой системы
100. ├── database/
101. │ ├── migrations/
102. │ │ ├── create\_users\_table.php # Миграция таблицы пользователей
103. │ │ ├── create\_teams\_table.php # Миграция таблицы команд
104. │ │ ├── create\_team\_members\_table.php # Миграция таблицы участников
105. │ │ ├── create\_results\_table.php # Миграция таблицы результатов
106. │ │ └── create\_posts\_table.php # Миграция таблицы новостей
107. │ └── seeders/
108. │ ├── UserSeeder.php # Сидер пользователей
109. │ └── PostSeeder.php # Сидер новостей
110. ├── public/
111. │ ├── assets/
112. │ │ ├── css/
113. │ │ │ └── app.css # Основные стили
114. │ │ └── js/
115. │ │ ├── app.js # Основной JavaScript
116. │ │ ├── auth.js # Скрипты аутентификации
117. │ │ ├── teams.js # Скрипты управления командами
118. │ │ └── map.js # Скрипты работы с картой
119. │ └── storage/
120. │ └── app/
121. │ └── public/ # Публичные файлы
122. ├── resources/
123. │ ├── views/
124. │ │ ├── layouts/
125. │ │ │ └── app.blade.php # Основной шаблон
126. │ │ ├── components/
127. │ │ │ ├── header.blade.php # Компонент шапки
128. │ │ │ ├── footer.blade.php # Компонент подвала
129. │ │ │ └── map.blade.php # Компонент карты
130. │ │ ├── auth/
131. │ │ │ ├── login.blade.php # Страница входа
132. │ │ │ └── register.blade.php # Страница регистрации
133. │ │ ├── teams/
134. │ │ │ ├── index.blade.php # Список команд
135. │ │ │ └── show.blade.php # Страница команды
136. │ │ └── admin/
137. │ │ ├── dashboard.blade.php # Панель управления
138. │ │ └── users.blade.php # Управление пользователями
139. │ ├── js/
140. │ │ └── components/ # Vue.js компоненты
141. │ └── css/
142. │ └── app.scss # Основные стили SCSS
143. ├── routes/
144. │ ├── web.php # Веб-маршруты
145. │ └── api.php # API-маршруты
146. ├── storage/
147. │ ├── logs/ # Логи приложения
148. │ └── framework/ # Фреймворк файлы
149. └── tests/
150. ├── Feature/
151. │ ├── AuthTest.php # Тесты аутентификации
152. │ ├── TeamTest.php # Тесты команд
153. │ └── ResultTest.php # Тесты результатов
154. └── Unit/
155. └── UserTest.php # Модульные тесты
156. **Руководство пользователя**
157. В данном разделе описывается руководство по использованию веб-приложения для акции "Бумбатл", включая инструкции по регистрации, работе с личным кабинетом и участию в акции.
158. { Создать скриншоты страниц и на них всё написать }
159. Регистрация и авторизация
160. Регистрация нового пользователя:
161. На главной странице нажать кнопку "Регистрация"
162. Заполнить форму регистрации:
163. Имя и фамилия
164. Email
165. Пароль (минимум 8 символов)
166. Подтверждение пароля
167. Подтвердить email через ссылку в письме
168. Войти в систему
169. Авторизация:
170. Нажать кнопку "Вход"
171. Ввести email и пароль
172. При необходимости использовать "Забыли пароль?"
173. Работа с личным кабинетом
174. Профиль пользователя:
175. Просмотр личных данных
176. Редактирование информации
177. Настройка уведомлений
178. История активности
179. Управление командой:
180. Создание новой команды:
181. Название команды
182. Описание
183. Выбор капитана
184. Приглашение участников:
185. По email
186. По ссылке
187. Управление участниками:
188. Назначение ролей
189. Исключение участников
190. Передача управления
191. Участие в акции
192. Загрузка результатов:
193. Перейти в раздел "Результаты"
194. Нажать "Загрузить результат"
195. Заполнить форму:
196. Вес собранной макулатуры
197. Дата сдачи
198. Пункт приема
199. Прикрепить подтверждающие документы
200. Отправить на проверку
201. Работа с картой:
202. Поиск пунктов приема:
203. По адресу
204. По типу пункта
205. По режиму работы
206. Просмотр информации:
207. Адрес
208. Режим работы
209. Контакты
210. Построение маршрута:
211. Выбор пункта назначения
212. Просмотр маршрута
213. Сохранение маршрута
214. Работа с контентом
215. Новости и анонсы:
216. Просмотр ленты новостей
217. Фильтрация по категориям
218. Поиск по новостям
219. Подписка на обновления
220. Статистика и рейтинги:
221. Просмотр общей статистики
222. Сравнение результатов
223. История достижений
224. Экспорт данных
225. Поддержка:
226. Форма обратной связи
227. Email поддержки
228. Чат с оператором
229. База знаний
230. Мобильная версия
231. Адаптивный интерфейс:
232. Оптимизация под мобильные устройства
233. Удобная навигация
234. Быстрый доступ к функциям
235. Оптимизированные формы
236. **Тестирование и отладка веб-приложения**
237. В данном разделе описываются методы и инструменты тестирования веб-приложения, а также процесс отладки и исправления ошибок.
238. Ошибки аутентификации:
239. Проблема: Неправильная валидация email при регистрации
240. Описание: Система принимала некорректные email-адреса
241. Решение: Добавлена строгая валидация через регулярное выражение
242. Код исправления:
243. ```php
244. // Было
245. 'email' => 'required|email'
246. // Стало
247. 'email' => [
     1. 'required',
     2. 'email',
     3. 'regex:/^[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/'
248. ]
249. ```
250. Проблема: Утечка сессии при параллельных запросах
251. Описание: Сессия сбрасывалась при одновременных запросах
252. Решение: Реализована блокировка сессии через Redis
253. Код исправления:
254. ```php
255. // Добавлен middleware
256. public function handle($request, Closure $next)
257. {
     1. $lock = Redis::lock('session:' . session()->getId(), 10);
     2. if ($lock->get()) {
     3. return $next($request);
     4. }
     5. return response()->json(['error' => 'Too many requests'], 429);
258. }
259. ```
260. Ошибки работы с базой данных:
261. Проблема: N+1 запросы при загрузке команд
262. Описание: Каждый участник команды загружался отдельным запросом
263. Решение: Добавлена жадная загрузка (eager loading)
264. Код исправления:
265. ```php
266. // Было
267. $teams = Team::all();
268. // Стало
269. $teams = Team::with(['members', 'captain'])->get();
270. ```
271. Проблема: Отсутствие индексов в таблице результатов
272. Описание: Медленный поиск по дате и статусу
273. Решение: Добавлены составные индексы
274. Код исправления:
275. ```php
276. // В миграции
277. Schema::table('results', function (Blueprint $table) {
     1. $table->index(['created\_at', 'status']);
     2. $table->index(['team\_id', 'status']);
278. });
279. ```
280. Ошибки фронтенда:
281. Проблема: Утечка памяти при работе с картой
282. Описание: Маркеры не удалялись при обновлении карты
283. Решение: Добавлена очистка маркеров перед обновлением
284. Код исправления:
285. ```javascript
286. // Было
287. markers.forEach(marker => map.addMarker(marker));
288. // Стало
289. markers.forEach(marker => {
     1. if (existingMarkers[marker.id]) {
     2. existingMarkers[marker.id].remove();
     3. }
     4. existingMarkers[marker.id] = map.addMarker(marker);
290. });
291. ```
292. Проблема: Некорректная валидация форм на фронтенде
293. Описание: Форма отправлялась с невалидными данными
294. Решение: Добавлена синхронная валидация
295. Код исправления:
296. ```javascript
297. // Добавлена функция валидации
298. function validateForm(formData) {
     1. const errors = {};
     2. if (formData.weight <= 0) {
     3. errors.weight = 'Вес должен быть больше 0';
     4. }
     5. if (!formData.date) {
     6. errors.date = 'Укажите дату';
     7. }
     8. return errors;
299. }
300. ```
301. Ошибки безопасности:
302. Проблема: XSS-уязвимость в комментариях
303. Описание: HTML-теги не экранировались при выводе
304. Решение: Добавлена санитизация данных
305. Код исправления:
306. ```php
307. // Было
308. echo $comment->content;
309. // Стало
310. echo htmlspecialchars($comment->content, ENT\_QUOTES, 'UTF-8');
311. ```
312. Проблема: CSRF-защита для API
313. Описание: Отсутствовала защита от CSRF-атак
314. Решение: Добавлен middleware VerifyCsrfToken
315. Код исправления:
316. ```php
317. // В Kernel.php
318. protected $middlewareGroups = [
     1. 'api' => [
     2. \App\Http\Middleware\VerifyCsrfToken::class,
     3. ],
319. ];
320. ```
321. Ошибки производительности:
322. Проблема: Медленная загрузка изображений
323. Описание: Отсутствовала оптимизация изображений
324. Решение: Добавлена обработка через Intervention Image
325. Код исправления:
326. ```php
327. // Добавлена обработка изображений
328. $image = Image::make($request->file('image'))
     1. ->resize(800, null, function ($constraint) {
     2. $constraint->aspectRatio();
     3. $constraint->upsize();
     4. })
     5. ->save();
329. ```
330. Проблема: Отсутствие кэширования API-запросов
331. Описание: Частые запросы к внешнему API
332. Решение: Реализовано кэширование через Redis
333. Код исправления:
334. ```php
335. // Добавлено кэширование
336. $data = Cache::remember('api\_data', 3600, function () {
     1. return Http::get('api.example.com/data')->json();
337. });
338. ```