#### ****РАЗДЕЛ 2. ТЕХНИЧЕСКОЕ ЗАДАНИЕ****

##### ****2.1 Основание для разработки****

Разработка веб-портала для ГБОУ "Школа № 41 г.о. Горловка" обусловлена необходимостью создания современного, удобного и доступного инструмента для размещения информации о деятельности школы, её новостях, мероприятиях и других активностях. Существующие решения не отвечают современным требованиям к адаптивности, быстродействию и удобству использования на различных устройствах. Новый портал должен стать централизованным источником информации для учащихся, родителей, учителей и других заинтересованных лиц, обеспечивая простоту навигации и оперативное обновление данных.

##### ****2.2 Назначение разработки****

Основное назначение разработки — создание веб-портала, который будет служить платформой для публикации актуальной информации о школе, её новостях, мероприятиях и других активностях. Портал должен быть доступен на различных устройствах (ПК, планшеты, смартфоны), обладать высокой скоростью загрузки и простотой в использовании. Кроме того, сайт должен быть легко администрируемым, что позволит сотрудникам школы оперативно обновлять контент.

##### ****2.3 Требования к web-ресурсу****

1. **Функциональные требования:**
   * Возможность публикации новостей и мероприятий.
   * Адаптивный дизайн для корректного отображения на различных устройствах.
   * Минималистичный и интуитивно понятный интерфейс.
   * Высокая скорость загрузки страниц.
2. **Технические требования:**
   * Использование современных технологий: React, Tailwind CSS, TypeScript.
   * Хранение данных в базе данных jsonbin.
   * Размещение сайта на GitHub Pages.
   * Поддержка кросс-браузерности (работа в современных браузерах: Chrome, Firefox, Safari, Edge).
3. **Требования к безопасности:**
   * Обеспечение защиты данных пользователей.
   * Регулярное обновление зависимостей и библиотек для предотвращения уязвимостей.

##### ****2.4 Требования к программной документации****

Программная документация должна включать:

* Описание архитектуры проекта.
* Руководство по установке и настройке.
* Инструкцию по администрированию сайта.
* Описание API для взаимодействия с базой данных.
* Руководство по добавлению нового контента.
* Документацию по используемым технологиям (React, Tailwind CSS, TypeScript).

##### ****2.5 Порядок контроля и приёмки****

Контроль качества разработки включает:

* Тестирование функциональности на различных устройствах и браузерах.
* Проверка скорости загрузки страниц.
* Тестирование безопасности и защиты данных.
* Проверка корректности отображения контента.
* Приёмка проекта заказчиком с подписанием акта выполненных работ.

#### ****РАЗДЕЛ 3. АНАЛИЗ ЦЕЛЕСООБРАЗНОСТИ ИСПОЛЬЗОВАНИЯ СРЕДСТВ РАЗРАБОТКИ****

##### ****3.1 Обоснование выбора языка программирования****

Для разработки портала был выбран стек технологий, включающий React, Tailwind CSS и TypeScript.

* **React** был выбран благодаря своей гибкости, производительности и возможности создания компонентного интерфейса, что позволяет легко масштабировать проект и поддерживать его в будущем.
* **Tailwind CSS** обеспечивает минималистичный и адаптивный дизайн, что соответствует требованиям к удобству и быстродействию сайта. Кроме того, Tailwind CSS позволяет быстро разрабатывать интерфейсы без необходимости написания большого количества кастомного CSS-кода.
* **TypeScript** был выбран для повышения надёжности кода за счёт статической типизации, что особенно важно при разработке крупных проектов с возможностью дальнейшего расширения.

##### ****3.2 Система управления базами данных****

В качестве базы данных используется сервис **jsonbin**, который предоставляет простой и удобный способ хранения и управления данными в формате JSON. Это решение было выбрано благодаря своей простоте и минимальным требованиям к настройке, что позволяет быстро интегрировать базу данных в проект. Кроме того, jsonbin обеспечивает достаточную производительность для задач портала, таких как хранение новостей, мероприятий и другой информации.

##### ****3.3 Система управления содержимым****

Администрирование сайта осуществляется путём прямого доступа к программному коду и базе данных. Это позволяет сотрудникам школы оперативно обновлять контент без необходимости использования сложных CMS. Для добавления новых новостей или мероприятий достаточно внести изменения в базу данных jsonbin, что делает процесс администрирования простым и понятным даже для пользователей без глубоких технических знаний.

#### ****РАЗДЕЛ 4. ОПИСАНИЕ ПРОГРАММЫ****

##### ****4.1 Общие сведения****

Веб-портал для ГБОУ "Школа № 41 г.о. Горловка" представляет собой одностраничное приложение (SPA), разработанное с использованием React, Tailwind CSS и TypeScript. Сайт размещён на GitHub Pages и использует базу данных jsonbin для хранения информации. Основная цель портала — предоставление актуальной информации о школе, её новостях и мероприятиях.

Каждый компонент сайта разработан с использованием React, что обеспечивает высокую производительность и удобство в использовании. Проект придерживается концепции «компонентного подхода». Каждая кнопка, каждый блок текста может беспрепятственно использоваться в любом месте сайта, и везде он будет смотреться уместно. Логика взаимодействия с базой данных реализована через API jsonbin, что позволяет оперативно получать данные.

##### ****4.2 Описание логической структуры****

Логическая структура портала включает следующие основные компоненты:

1. **Главная страница** — содержит краткую информацию о школе, последние новости и анонсы мероприятий.
2. **Раздел новостей** — список всех новостей с возможностью фильтрации по дате или категориям.
3. **Раздел мероприятий** — информация о предстоящих и прошедших мероприятиях.
4. Раздел «Сведения о образовательной организации» - структурированная информация о всём, что связанно с руководством и деятельностью школы. Включает в себя:
5. Основные сведения
6. Структура и органы управления образовательной организацией
7. Документы
8. Образование
9. Педагогический состав

### ****РАЗДЕЛ 5. РУКОВОДСТВО ОПЕРАТОРА****

#### ****5.1 Назначение программы****

Веб-портал для ГБОУ "Школа № 41 г.о. Горловка" предназначен для предоставления актуальной информации о школе, её новостях, мероприятиях и других активностях. Программа обеспечивает удобный доступ к информации для учащихся, родителей, учителей и других заинтересованных лиц. Администрирование сайта осуществляется путём прямого доступа к программному коду и базе данных, что позволяет оперативно обновлять контент без необходимости использования сложных систем управления содержимым (CMS).

#### ****5.2 Условия выполнения программы****

Для развёртки проекта на компьютере требуется:

1. **Технические требования:**
   * Наличие стабильного интернет-соединения.
   * Поддержка современных браузеров (Google Chrome, Mozilla Firefox, Safari, Microsoft Edge).
   * Доступ к GitHub для управления репозиторием с программным кодом.
   * Доступ к сервису jsonbin для управления базой данных.
2. **Программное обеспечение:**
   * Установленный Node.js (версия 16 или выше) для локальной разработки и тестирования.
   * Git для управления версиями и развёртывания на GitHub Pages.
   * Текстовый редактор (например, Visual Studio Code) для редактирования кода.
3. **Аппаратные требования:**
   * Компьютер или ноутбук с операционной системой Windows, macOS или Linux.
   * Минимальные требования к процессору и оперативной памяти для комфортной работы с кодом и базой данных.

Однако для доступа к сайту в качестве обычного пользователя сгодится любой более-менее современный девайс с доступом к глобальной сети Интернет

#### ****5.3 Установка сайта на хостинг****

Сайт размещён на **GitHub Pages**, что обеспечивает его доступность для широкой аудитории. Процесс установки и развёртывания сайта включает следующие шаги:

1. **Клонирование репозитория:**
   * Перейдите в репозиторий проекта на GitHub.
   * Склонируйте репозиторий на локальный компьютер с помощью команды:

bash

Copy

git clone https://github.com/username/repository-name.git

1. **Установка зависимостей:**
   * Перейдите в папку проекта и установите необходимые зависимости с помощью команды:

bash

Copy

npm install

1. **Сборка проекта:**
   * Для сборки проекта выполните команду:

bash

Copy

npm run build

1. **Развёртывание на GitHub Pages:**
   * Убедитесь, что в проекте настроен файл gh-pages для автоматического развёртывания.
   * Выполните команду для публикации сайта:

bash

Copy

npm run deploy

1. **Проверка доступности:**
   * После завершения развёртывания сайт будет доступен по адресу, указанному в настройках GitHub Pages.

#### ****5.4 Администрирование сайта****

Администрирование сайта осуществляется путём прямого доступа к программному коду и базе данных. Это позволяет оперативно обновлять контент и вносить изменения в структуру сайта. Процесс администрирования включает следующие этапы:

1. **Обновление контента через базу данных jsonbin:**
   * Перейдите в сервис jsonbin и авторизуйтесь.
   * Найдите базу данных, используемую для сайта.
   * Внесите необходимые изменения в данные (например, добавьте новую новость или мероприятие).
   * Сохраните изменения, которые автоматически отобразятся на сайте.
2. **Изменение программного кода:**
   * Для внесения изменений в структуру сайта или его функциональность необходимо отредактировать исходный код.
   * Откройте проект в текстовом редакторе (например, Visual Studio Code).
   * Внесите необходимые изменения в файлы (например, добавьте новый компонент или измените стили).
   * Протестируйте изменения локально с помощью команды:

bash

Copy

npm start

* + После проверки внесите изменения в репозиторий GitHub:

git add .

git commit -m "Описание изменений"

git push origin main

* + Разверните обновлённый сайт на GitHub Pages с помощью команды:

bash

Copy

npm run deploy

1. **Управление доступом:**
   * Для обеспечения безопасности рекомендуется ограничить доступ к репозиторию GitHub и базе данных jsonbin только для авторизованных сотрудников.
   * Используется двухфакторная аутентификация для защиты аккаунтов.