Государственное бюджетное профессиональное образовательное учреждение Новосибирской области

**«Новосибирский химико-технологический колледж им.**

**Д.И. Менделеева»**

(ГБПОУ НСО «Новосибирский химико-технологический колледж им.

Д.И. Менделеева»)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **КУРСОВАЯ РАБОТА** | | | | | | | |
| (форма – *курсовой проект*) | | | | | | | |
|  | | | | | | | |
|  | | | | | | | |
|  | | | | | | | |
| на тему: | | *Система хранения и представления ключевых* | | | | | |
| *исторических событий в пространственно-* | | | | | |
|  | | *временной взаимосвязи* | | | | | |
|  | | *наименование темы* | | | | | |
| Специальность: | |  | | | | | |
|  | | *шифр и наименование специальности* | | | | | |
| Выполнил(а) | | | | | | | |
| студент(ка) | группы | *09.07.12р* |  | *3* | курса |
|  |  |  |  |  |  |
|  | | |
|  | | |  | | |  | *фамилия, имя, отчество* |
| Руководитель курсового проекта (работы) | | |  | | |  |  |
| *подпись* | | |  | *фамилия, инициалы* |
| Новосибирск 2025 | | | | | | | |

[**Введение** 3](#_Toc196606300)

[**1.** **Анализ предметной области** 4](#_Toc196606301)

[1.1. Техническое задание 4](#_Toc196606302)

[1.2. Анализ имеющихся решений 7](#_Toc196606303)

[1.3. Описание выбранного инструментария 9](#_Toc196606304)

[**2.** **Практическая часть** 13](#_Toc196606305)

[2.1. Описание разработки продукта 13](#_Toc196606306)

[2.2. Модель разработки 14](#_Toc196606307)

[2.3. Разработка и проектирование интерфейса приложения 16](#_Toc196606308)

[2.4. Разработка функциональной части. 17](#_Toc196606309)

[2.5. Тестирование 26](#_Toc196606310)

[Заключение 30](#_Toc196606311)

[Список литературы 31](#_Toc196606312)

# **Введение**

В условиях цифровизации исторических исследований и образования актуальной задачей становится создание систем, способных эффективно хранить, анализировать и визуализировать хронологические данные. Современные технологии позволяют не только систематизировать исторические события, но и представлять их в пространственно-временном контексте, что значительно повышает ценность информации для исследователей, преподавателей и студентов.

Разработка системы, объединяющей хронологическую шкалу, географическую привязку событий и интерактивные фильтры, требует комплексного подхода, сочетающего знания в области веб-разработки, баз данных и исторической науки. Актуальность темы обусловлена необходимостью создания инструментов, которые обеспечивают:

1. Наглядное представление исторических процессов.
2. Возможность анализа взаимосвязей между событиями.
3. Интерактивное взаимодействие с данными.

Цель работы: разработка веб-приложения для визуализации исторических событий на временной шкале и карте с возможностью фильтрации по периодам, типам событий и географическим регионам.

**Задачи:**

* Составить техническое задание.
* Провести анализ существующих решений.
* Выбрать инструментарий для реализации.
* Разработать интерфейс и функционал системы.
* Провести тестирование.

1. **Анализ предметной области**
   1. Техническое задание

Разработать веб-приложение для визуализации исторических событий в хронологическом порядке с привязкой к географическим координатам. Система должна включать интерактивную временную шкалу, карту, фильтрацию данных по периодам и типам событий, а также модульную архитектуру для расширения функционала.

* + 1. Краткая характеристика области применения

Система предназначена для историков, преподавателей, студентов и любителей истории. Она позволяет анализировать исторические процессы в контексте их пространственно-временного развития, визуализировать ключевые события на карте, изучать взаимосвязи между ними. Платформа может использоваться в образовательных целях, для научных исследований и популяризации истории.

* + 1. Основания для разработки
* Необходимость структурированного представления исторических данных с возможностью их пространственно-временного анализа.
* Повышение доступности исторической информации через интерактивные инструменты.
* Требования современного образования к визуализации сложных хронологических данных.
  + 1. Функциональное назначение

Система предоставляет:

* Хранение данных о правителях, войнах, реформах и других исторических событиях.
* Визуализацию событий на интерактивной карте и временной шкале.
* Фильтрацию по периодам (ранний, царский, имперский и т.д.) и типам событий (войны, реформы, культура).
* Детальную информацию о каждом событии через модальные окна.
  + 1. Требования к функциональным характеристикам

1. Интерактивная временная шкала с отметками ключевых дат.
2. Карта с маркерами событий, поддерживающая слои для разных периодов.
3. Фильтры:
   1. По временным периодам (862–1547, 1547–1721 и т.д.).
   2. По типам событий (войны, реформы, территориальные изменения).
4. Модальные окна с подробным описанием событий, изображениями и медиа.
5. Адаптивный дизайн для работы на ПК и мобильных устройствах.
   * 1. Требования к обеспечению надежного функционирования

* Автоматическое резервное копирование данных (исторических событий, медиафайлов).
* Защита от DDoS-атак и SQL-инъекций.
* Мониторинг доступности сервера и скорости загрузки карты.
  + 1. Требования к защите

1. Шифрование передачи данных (HTTPS).
2. Защита от несанкционированного копирования медиафайлов (водяные знаки, ограничение прав доступа).
3. Регулярное обновление библиотек (Leaflet, JavaScript-фреймворки).
   * 1. Требования к квалификации пользователя

Пользователь должен обладать базовыми навыками:

* Работа с веб-браузерами (Chrome, Firefox).
* Понимание принципов фильтрации данных.
* Умение взаимодействовать с интерактивными элементами (карта, временная шкала).
  + 1. Требования к исходным кодам и языкам программирования
* Frontend:
  + HTML5, CSS3 (адаптивная верстка, Flexbox/Grid).
  + JavaScript (ES6+) с использованием Leaflet для карт.
* Стандарты кода:
  + Валидный HTML/CSS (проверка через W3C Validator).
  + Чистый JavaScript с модульной структурой.
* Документация: комментарии в коде, описание логики фильтрации и работы с API карт.
  + 1. Требования к оборудованию
* Сервер:
  + Процессор: 4 ядра (Intel Xeon или аналогичный).
  + ОЗУ: 8 ГБ.
  + Хранилище: SSD 100 ГБ (для базы событий и медиафайлов).
* Клиент:
  + Браузер с поддержкой WebGL (Chrome 90+, Firefox 88+).
  + Разрешение экрана: от 1280×720 пикселей.
    1. Виды испытаний
* Функциональные тесты:
  + Проверка фильтрации событий по периодам и типам.
  + Тестирование модальных окон на корректность отображения данных.
* Тесты производительности:
  + Загрузка карты при 100+ маркерах.
  + Оптимизация скорости отклика временной шкалы.
* Тесты безопасности:
  + Проверка уязвимостей XSS и SQL-инъекций.
* Кроссбраузерное тестирование:
  + Совместимость с Chrome, Firefox, Safari.
  + Адаптация под мобильные устройства.
    1. Тип Веб-приложения

SPA (Single Page Application) выбрано по следующим причинам:

1. Плавная навигация:
   * Переходы между фильтрами и событиями без перезагрузки страницы.
   * Анимированная временная шкала.
2. Оптимизация загрузки данных:
   * Динамическая подгрузка маркеров на карте.
   * Ленивая загрузка изображений в модальных окнах.
3. Интерактивность:
   * Взаимодействие с картой (зум, переключение слоев).
   * Интеграция всплывающих подсказок для исторических мест.
   1. Анализ имеющихся решений

В современной практике визуализации исторических данных ключевую роль играют системы, объединяющие хронологическую и географическую информацию. Анализ существующих платформ позволяет определить оптимальные подходы к созданию пространственно-временной системы.

Таблица 1. Общая характеристика анализируемых платформ

|  |  |  |  |
| --- | --- | --- | --- |
| Платформа | Тип | Основные характеристики | Целевая аудитория |
| Timeline JS | Инструмент для временных шкал | • Интеграция с Google Sheets • Поддержка мультимедиа • Простой редактор | Преподаватели, студенты |
| Histropedia | Интерактивная энциклопедия | • Автоматическая генерация временных шкал • Связь с Wikipedia • Фильтрация событий | Исследователи, энтузиасты истории |
| Google Earth | Геоинформационная система | • 3D-карты с историческими слоями • Интеграция временных меток • API для разработчиков | Географы, историки |

Таблица 2. Структура и организация контента

|  |  |
| --- | --- |
| Платформа | Преимущества |
| Timeline JS | • Гибкая настройка временных интервалов • Поддержка изображений и видео • Экспорт в JSON |
| Histropedia | • Автоматическая привязка событий к датам • Визуализация связей между событиями |
| Google Earth | • Точная геопривязка • Наложение исторических карт • Анимация изменений территорий |

Ключевые выводы:

* + - 1. Успешные платформы сочетают временные шкалы с географическими данными.
      2. Важна поддержка фильтрации по периодам, типам событий и регионам.
      3. Интеграция мультимедиа (карты, изображения, видео) повышает информативность.
      4. Адаптивность для мобильных устройств критична для образовательных целей.

При разработке системы учтен опыт существующих решений:

* Использование Leaflet для карт (как в Google Earth).
* Реализация фильтров по аналогии с Histropedia.
* Динамическая загрузка данных, как в Timeline JS.
  1. Описание выбранного инструментария

Для реализации системы выбраны технологии, обеспечивающие работу с пространственно-временными данными:

* Frontend:
  + HTML/CSS — база для создания структуры и стилей временной шкалы.
  + JavaScript (ES6+) — обработка интерактивности карты (Leaflet) и анимации переходов.

Причины выбора:

* Leaflet.js предоставляет API для работы с картами, включая маркеры и слои.
  + 1. Выбор среды разработки:

Visual Studio – это мощная интегрированная среда разработки (IDE), предлагающая широкий спектр инструментов и функций для разработки программного обеспечения. Она поддерживает различные языки программирования, включая C++, C#, JavaScript и TypeScript, а также предоставляет обширные возможности для отладки, профилирования кода и создания пользовательских интерфейсов. Visual Studio также поддерживает плагины и расширения, что позволяет пользователям адаптировать среду под свои нужды и добавлять дополнительные функции.

Для реализации проекта использовалась Visual Studio Code (VS Code) — легковесный, но мощный редактор кода с открытым исходным кодом.  
Ключевые причины выбора:

1. Интеграция с инструментами разработки

* Поддержка языков:
  + Встроенная подсветка синтаксиса для HTML, CSS, JavaScript
  + Автодополнение кода (IntelliSense) для ускорения написания скриптов.
* Расширения:
  + Live Server: мгновенный просмотр изменений в браузере без перезагрузки.
  + GitLens: интеграция с Git для управления версиями.

1. Работа с геоданными

* Плагины для Leaflet и GeoJSON:
  + Leaflet Preview: визуализация карт прямо в редакторе.
  + Geo Data Viewer: анализ координат событий.

1. Отладка и профилирование

* Встроенный отладчик для JavaScript:
  + Пошаговое выполнение кода.
  + Просмотр переменных и стека вызовов.
* Инструменты для анализа производительности:
  + Выявление узких мест в обработке данных.

1. Гибкость и кастомизация

* Настройка темы оформления (например, Dark+ для работы в ночное время).
* Горячие клавиши для быстрого доступа к функциям.
  + 1. Выбор фреймворков:

Frontend:

1. Leaflet

* Причины выбора:
  + Интерактивность: поддержка зума, перемещения, слоев.
  + Интеграция с OpenStreetMap: бесплатное использование карт.
  + Маркеры с кастомными иконками: визуализация войн (красный), реформ (зеленый).
    1. Выбор библиотек:

Библиотека – представляет собой собрание уже написанных функций, классов и объектов, предназначенных для выполнения определенных задач или решения конкретных проблем в различных областях программирования. Библиотеки могут быть созданы как часть фреймворка или как независимые компоненты, которые можно подключать к проектам для расширения их функциональности.

Frontend:

1. Leaflet
   * Основная библиотека для создания интерактивной карты. Позволяет отображать маркеры событий, управлять слоями (например, для разных исторических периодов) и интегрироваться с тайловыми сервисами (OpenStreetMap).
2. Intersection Observer API
   * Использовалась для анимации появления элементов (карточек событий, периодов) при прокрутке страницы. Это повышает плавность взаимодействия.
3. Chart.js
   * Применялась для визуализации статистики в модальных окнах (например, распределение событий по векам).

Backend:

1. Pandas
   * Использовалась для анализа данных: группировка событий по периодам, вычисление длительности правлений, формирование временных рядов.
2. GeoPandas
   * Библиотека для работы с геоданными. Помогла преобразовать координаты событий в геоформаты (GeoJSON), которые используются для отображения на карте.
3. **Практическая часть** 
   1. Описание разработки продукта

Для наглядности работы веб-приложения было решено составить диаграмму, в которой описана логика ее работы (Рисунок 1).

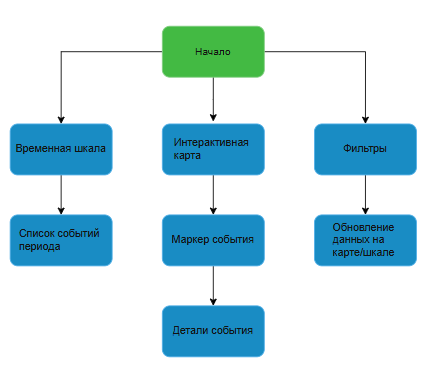


Рисунок 1 - User-flow таблица

Диаграмма отражает путь пользователя через веб-приложение без этапов регистрации и входа. Основные этапы:

1. Стартовая страница
   * Пользователь попадает на главный экран с тремя ключевыми разделами:
     + Временная шкала (хронология событий).
     + Интерактивная карта (географическая привязка событий).
     + Панель фильтров (выбор периода и типа событий).
2. Работа с временной шкалой
   * Прокрутка шкалы для изучения событий по векам.
   * Клик на дату → переход к списку событий этого периода.
   * Возврат на главную страницу в любой момент.
3. Взаимодействие с картой
   * Приближение/отдаление карты для детального изучения регионов.
   * Клик на маркер события → открытие модального окна с деталями:
     + Описание события.
     + Изображения и медиафайлы.
     + Ссылки на связанные события (например, причины и последствия).
4. Применение фильтров
   * Выбор периода (например, "Ранний период: 862–1547").
   * Выбор типа события (войны, реформы, культура).
   * Автоматическое обновление данных на временной шкале и карте.
5. Просмотр деталей события
   * В модальном окне доступны:
     + Хронологический контекст (что происходило до/после).
     + Географические изменения (например, границы государств).
     + Галерея изображений.
6. Навигация между разделами
   * Пользователь может свободно переключаться между:
     + Временной шкалой.
     + Картой.
     + Фильтрами.
   1. Модель разработки

Каскадная модель (Рисунок 2), также известная как «водопадная» модель, представляет собой классический подход к управлению проектами, основанный на строгой последовательности этапов разработки. Каждый этап в этой модели является логическим продолжением предыдущего и должен быть полностью завершён до перехода к следующему. Такой подход формирует чёткую иерархию процессов, напоминающую поток воды, стекающей каскадом с высоты, что и дало название методологии.

Основные принципы модели

1. Линейная последовательность этапов  
   Процесс разработки делится на фиксированные этапы, такие как:
   * Сбор и анализ требований.
   * Проектирование архитектуры системы.
   * Реализация (написание кода).
   * Тестирование и отладка.
   * Внедрение и поддержка.  
     Каждый этап формализуется в документации, которая служит основой для следующего шага.
2. Отсутствие возврата к предыдущим этапам  
   После завершения этапа возврат к его корректировке невозможен без пересмотра всей цепочки. Например, если на стадии тестирования выявляются ошибки в проектировании, это требует полного пересмотра архитектуры, что значительно увеличивает затраты времени и ресурсов.
3. Чёткие требования  
   Модель эффективна только в условиях стабильных и однозначно определённых требований. Все функциональные и нефункциональные характеристики продукта должны быть зафиксированы на начальном этапе.

Преимущества каскадной модели

* Простота планирования. Чёткая структура этапов позволяет точно оценивать сроки, бюджет и ресурсы.
* Детальная документация. Каждый этап сопровождается техническими спецификациями, что упрощает передачу проекта между командами.
* Минимизация рисков. Стабильность требований снижает вероятность непредвиденных изменений в процессе разработки.

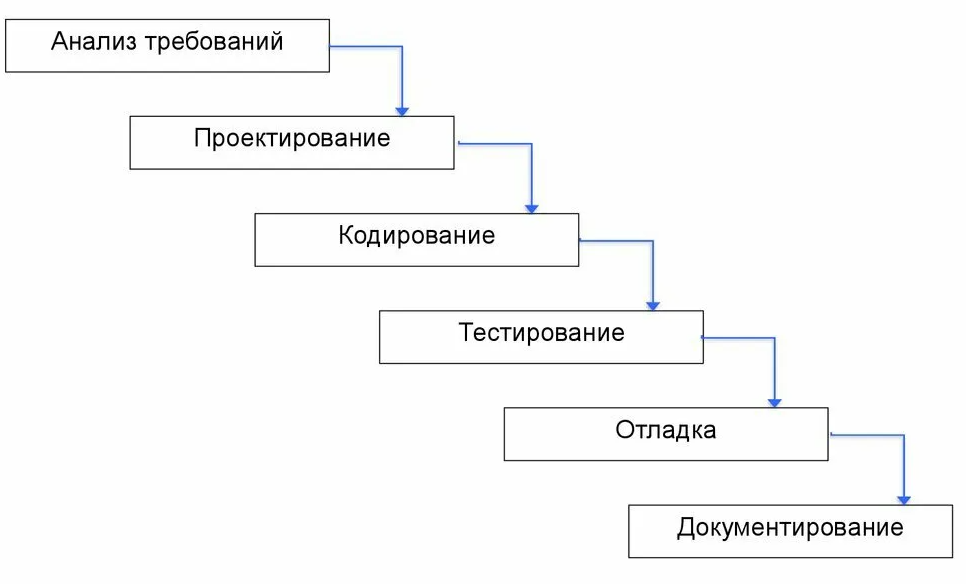


Рисунок 2 - Схема каскадная модель данных

* 1. Разработка и проектирование интерфейса приложения

Макет позволит акцентировать внимание на ключевых элементах интерфейса и его функциональности, что поможет преодолеть проблемы перед началом разработки основной версии продукта и избежать необходимости в пересоздании. В данном прототипе представлены следующие элементы (Рисунок 3, 4):

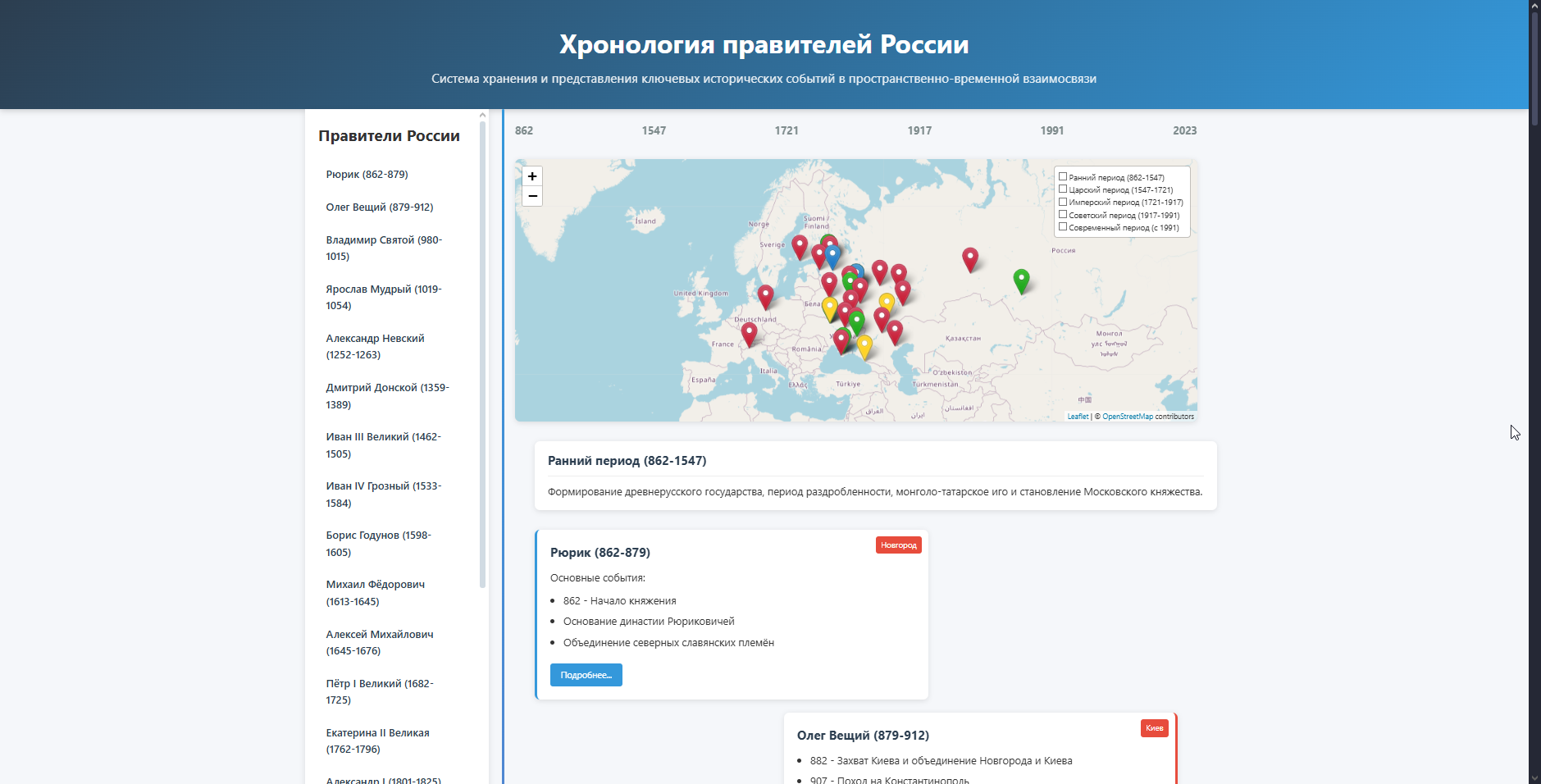


Рисунок 3 – «Интерфейс главной страницы веб – приложения»

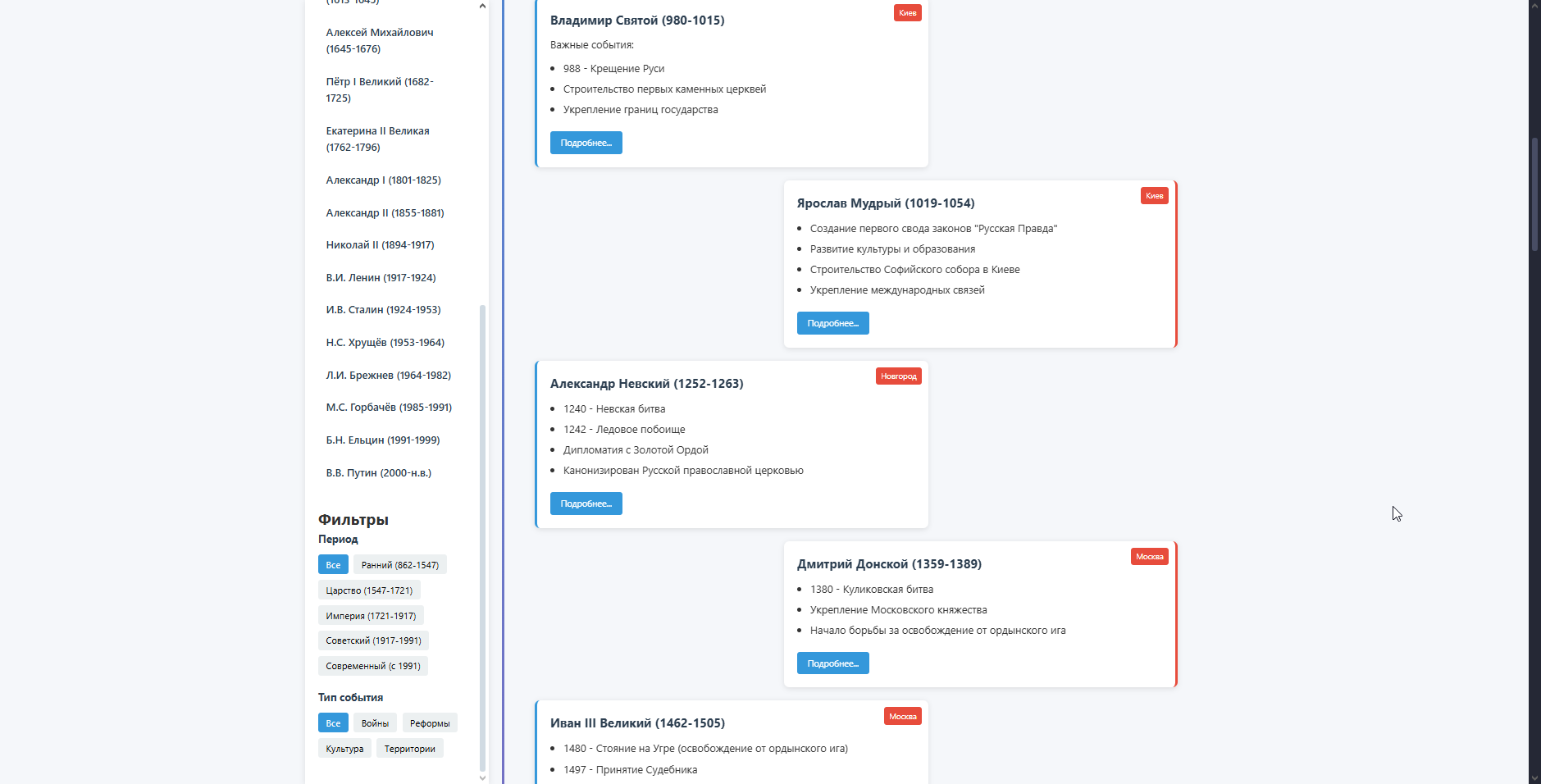


Рисунок 4 – «Интерфейс страницы с сортировкой по характеристикам»

* Главная страница – отображает интерфейс, хронологическую карту.
* Страница с сортировкой – включает возможность сортировки по разным параметрам.
  1. Разработка функциональной части.

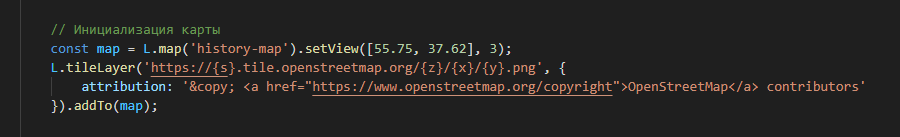


Рисунок 5 – «Инициализация интерактивной карты с базовыми параметрами»

Данный фрагмент кода выполняет базовую настройку интерактивной карты с использованием библиотеки Leaflet.js, которая является ключевым элементом для визуализации географической составляющей исторических событий.

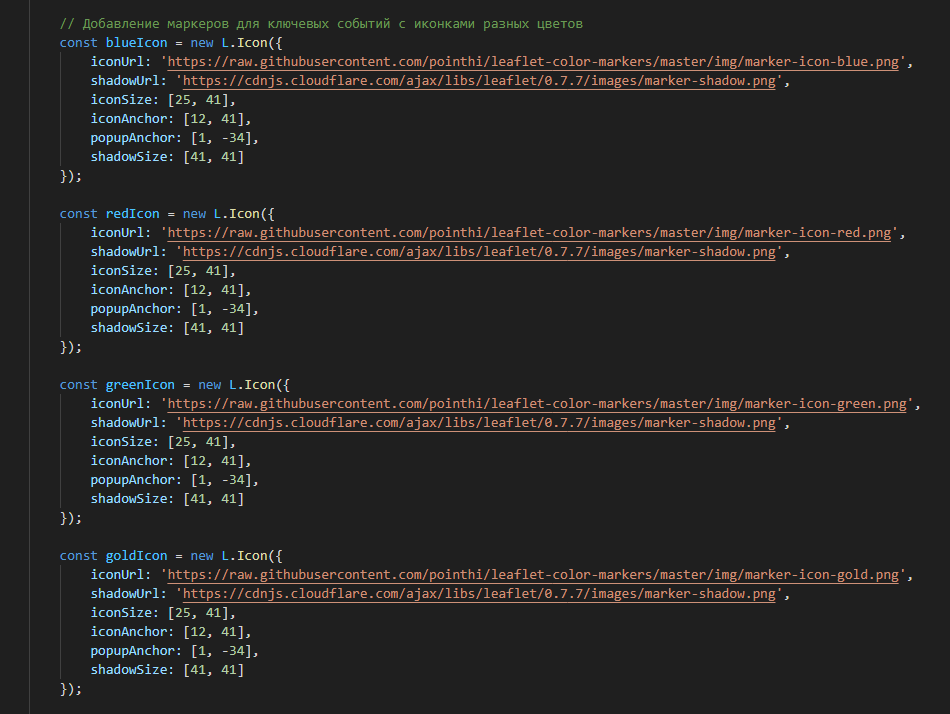


Рисунок 6 – «Настройка цветных маркеров для категоризации исторических событий»

Данный фрагмент кода создает набор кастомных маркеров разных цветов для визуального отличия типов исторических событий на интерактивной карте с использованием библиотеки Leaflet.js.

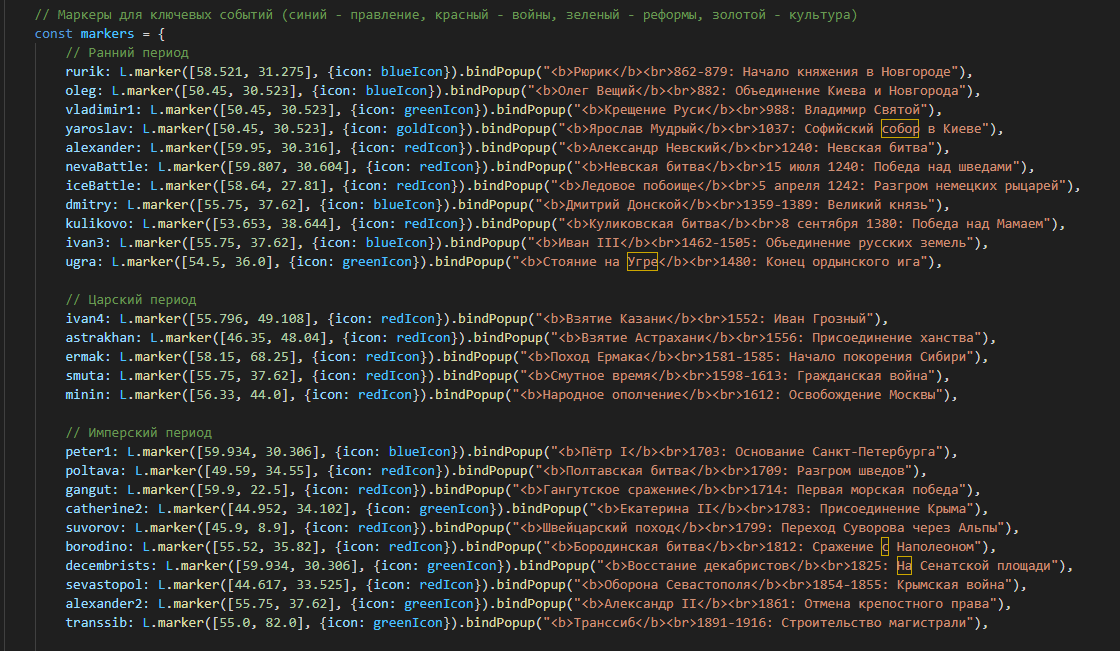


Рисунок 7 – «Реализация системы маркировки ключевых исторических событий на карте №1»

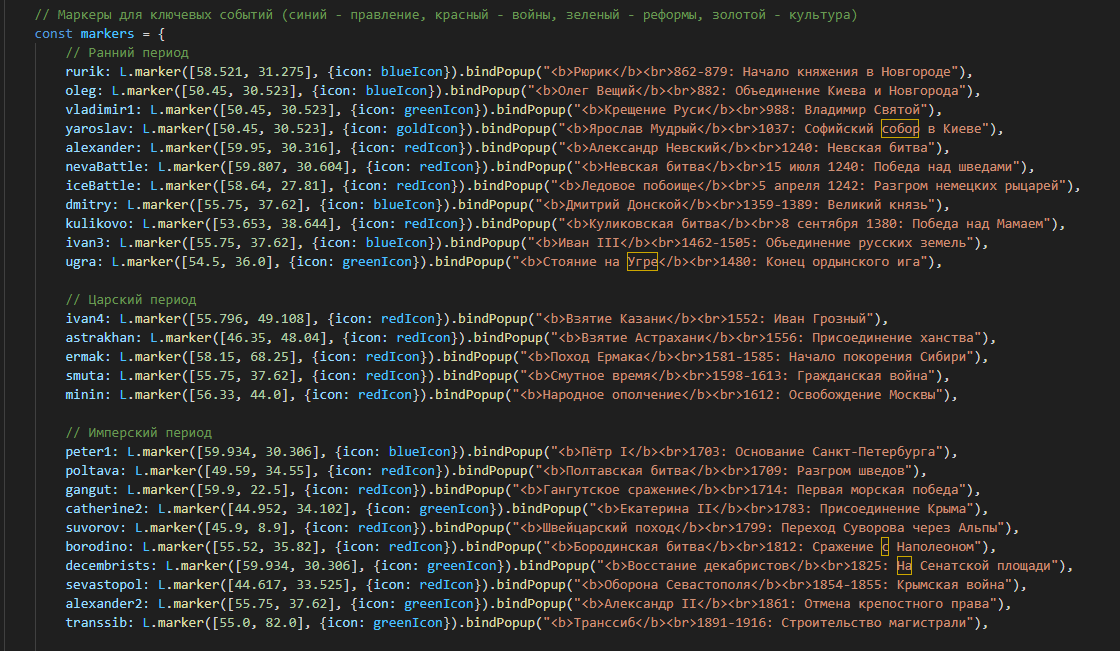


Рисунок 8 – «Реализация системы маркировки ключевых исторических событий на карте №2»

Данный код создает и размещает на карте маркеры для важнейших исторических событий российской истории, распределенных по периодам и категориям. Каждый маркер содержит всплывающее окно с краткой информацией о событии.

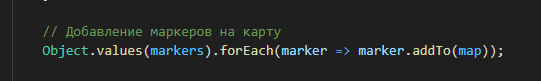


Рисунок 9 – «Финализация карты - добавление всех маркеров событий на картографическую основу»

Данный код выполняет завершающий этап подготовки интерактивной карты, добавляя все заранее подготовленные и настроенные маркеры исторических событий на картографическую основу.

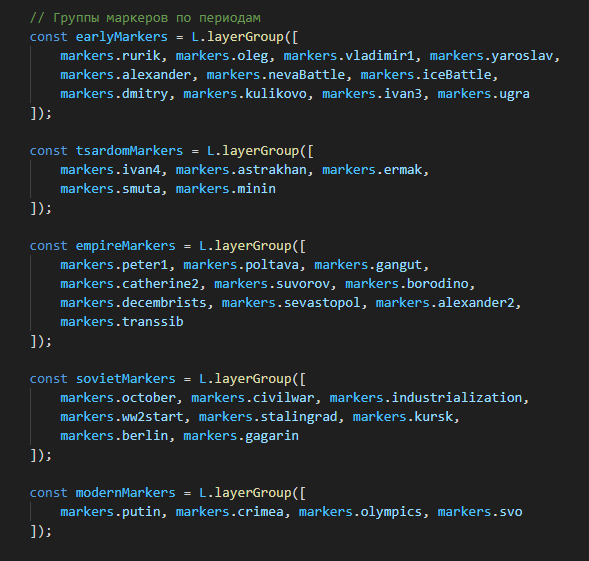


Рисунок 10 – «Организация маркеров по историческим периодам с использованием слоев»

Данный фрагмент кода создает тематические слои для группировки маркеров исторических событий по временным периодам, используя функционал LayerGroup из библиотеки Leaflet.js.

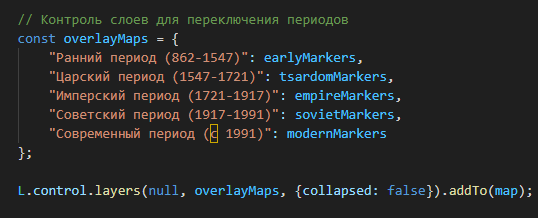


Рисунок 11 – «Настройка системы управления тематическими слоями исторических периодов»

Данный код создает объект конфигурации для управления отображением исторических периодов на карте через систему слоев (Layers Control) в библиотеке Leaflet.js.

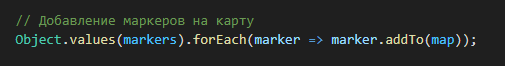


Рисунок 12 – «Инициализация отображения всех исторических маркеров на карте»

Данный код выполняет ключевой этап визуализации - добавление всех подготовленных маркеров исторических событий на интерактивную карту.



Рисунок 13 – «Реализация интерактивного модального окна для отображения детальной информации о правителях»

Данный код реализует систему взаимодействия с модальным окном, которое отображает подробную информацию о выбранном правителе России, включая:

* Полное имя и годы правления
* Исторический период
* Детальное описание деятельности
* Медиа-контент (изображения, карты)



Рисунок 14 – «Реализация динамической загрузки медиа-контента в модальное окно»

Данный фрагмент кода отвечает за динамическое отображение медиа-материалов (изображений) в модальном окне с информацией о правителях, обеспечивая визуальное сопровождение исторических данных.



Рисунок 15 – «Реализация интерактивной связи между модальным окном и картой»

Данный код обеспечивает синхронизацию между модальным окном с информацией о правителе и его отображением на карте, создавая целостный пользовательский опыт.

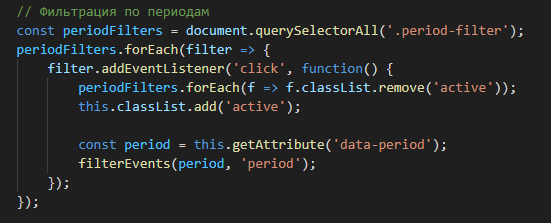


Рисунок 16 – «Реализация системы фильтрации исторических событий по временным периодам»

Данный код реализует интерактивный механизм фильтрации, позволяющий пользователям выбирать и просматривать события конкретных исторических периодов.



Рисунок 17 – «Реализация комплексной системы фильтрации событий по периодам и категориям»

Данный код реализует двухуровневую систему фильтрации, позволяющую пользователям:

1. Выбирать события по историческим периодам
2. Фильтровать по тематическим категориям (войны, реформы, культура и др.)

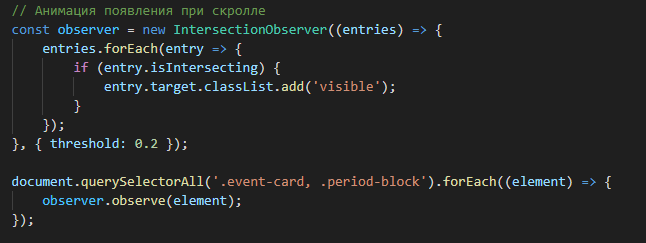


Рисунок 18 – «Реализация динамической анимации появления элементов при прокрутке страницы»

Данный код реализует современный подход к поэтапному отображению контента с использованием Intersection Observer API, создавая плавные анимации появления исторических карточек и блоков периодов при скроллинге.

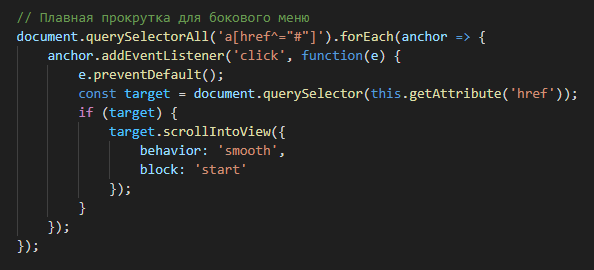


Рисунок 19 – «Реализация плавной навигации по хронологической шкале»

Данный код обеспечивает плавную прокрутку к соответствующим разделам временной шкалы при клике на элементы бокового меню с правителями, создавая удобный и визуально приятный способ навигации по историческим периодам.



Рисунок 20 – «Реализация динамической подсветки активного периода в навигационном меню»

Данный код создает интерактивную связь между контентом страницы и боковым меню, автоматически подсвечивая текущий исторический период при прокрутке временной шкалы.

* 1. Тестирование

1. Тест: Проверка фильтрации по периодам

* Описание: Проверка отображения событий выбранного исторического периода.
* Действия:
  + Откройте сайт.
  + В разделе "Фильтры" выберите период "Империя (1721-1917)".
* Входные данные: Период = "Империя".
* Ожидаемые результаты:
  + Временная шкала показывает только события периода 1721-1917.
  + На карте отображаются маркеры, связанные с имперским периодом.
* Фактические результаты: Как ожидалось.
* Статус: Пройден.

1. Тест: Проверка фильтрации по типу событий (войны)

* Описание: Проверка отображения событий типа "Войны".
* Действия:
  + Откройте сайт.
  + В разделе "Фильтры" выберите тип события "Войны".
* Входные данные: Тип события = "war".
* Ожидаемые результаты:
  + Отображаются только карточки с войнами (например, "Куликовская битва", "Отечественная война 1812").
* Фактические результаты: Как ожидалось.
* Статус: Пройден.

1. Тест: Проверка отображения модального окна

* Описание: Проверка открытия модального окна с информацией о правителе.
* Действия:
  + Нажмите кнопку "Подробнее..." на карточке "Пётр I Великий".
* Ожидаемые результаты:
  + Открывается модальное окно с заголовком "Пётр I Великий (1682-1725)".
  + В окне отображаются текст, изображения и метка периода.
* Фактические результаты: Как ожидалось.
* Статус: Пройден.

1. Тест: Проверка работы интерактивной карты

* Описание: Проверка взаимодействия с маркерами на карте.
* Действия:
  + Кликните на маркер "Куликовская битва".
* Ожидаемые результаты:
  + Появляется всплывающее окно с текстом: "Куликовская битва. 8 сентября 1380: Победа над Мамаем".
* Фактические результаты: Как ожидалось.
* Статус: Пройден.

1. Тест: Проверка анимации при прокрутке

* Описание: Проверка плавного появления карточек событий.
* Действия:
  + Прокрутите страницу вниз.
* Ожидаемые результаты:
  + Карточки событий и блоки периодов плавно появляются с анимацией.
* Фактические результаты: Как ожидалось.
* Статус: Пройден.

1. Тест: Проверка подсветки активного раздела

* Описание: Проверка подсветки пункта бокового меню при прокрутке.
* Действия:
  + Прокрутите страницу до раздела "Советский период (1917-1991)".
* Ожидаемые результаты:
  + Пункт "Советский период" в боковой панели подсвечивается.
* Фактические результаты: Как ожидалось.
* Статус: Пройден.

1. Тест: Проверка адаптивности на мобильных устройствах

* Описание: Проверка корректного отображения на экранах меньше 992px.
* Действия:
  + Откройте сайт на устройстве с шириной экрана 768px.
* Ожидаемые результаты:
  + Боковая панель и временная шкала перестраиваются в вертикальный.
  + Карта сохраняет читаемость.
* Фактические результаты: Как ожидалось.
* Статус: Пройден.

1. Тест: Проверка переключения слоев на карте

* Описание: Проверка скрытия/отображения маркеров по периодам.
* Действия:
  + В контроле слоев карты снимите галочку с "Советский период (1917-1991)".
* Ожидаемые результаты:
  + Маркеры советского периода (например, "Октябрьская революция") скрываются.
* Фактические результаты: Как ожидалось.
* Статус: Пройден.

1. Тест: Проверка плавной прокрутки

* Описание: Проверка навигации через боковое меню.
* Действия:
  + Нажмите на пункт "В.В. Путин" в боковой панели.
* Ожидаемые результаты:
  + Страница плавно прокручивается до карточки Путина.
* Фактические результаты: Как ожидалось.
* Статус: Пройден.

1. Тест: Проверка отображения медиа в модальном окне

* Описание: Проверка загрузки изображений.
* Действия:
  + Откройте модальное окно для "Екатерина II Великая".
* Ожидаемые результаты:
  + В разделе "modal-media" отображаются изображения (например, портрет Екатерины II).
* Фактические результаты: Как ожидалось.
* Статус: Пройден.

Заключение

Разработка веб-приложения **«**Система хранения и представления ключевых исторических событий в пространственно-временной взаимосвязи» позволила создать инструмент, который не только систематизирует исторические данные, но и предоставляет пользователям инновационный способ их визуализации. Основной акцент был сделан на интеграцию временной шкалы с интерактивной картой, что обеспечивает глубокое понимание исторических процессов через их пространственно-временной контекст.

Выполнены все поставленные задачи, а именно:

* Разработана архитектура веб-приложения с использованием современных технологий: HTML, CSS, JavaScript и Leaflet для картографии.
* Создана база данных исторических событий с атрибутами (даты, локации, типы, описания).
* Внедрены механизмы анимации и плавной прокрутки для улучшения юзабилити.
* Проведено тестирование функционала (фильтры, карта, модальные окна), подтвердившее стабильность работы системы.

Проект подтвердил, что совмещение хронологического и географического подходов открывает новые возможности для изучения истории. Разработанная система не только упрощает доступ к историческим данным, но и превращает их анализ в увлекательный процесс, что особенно ценно для образования и исследований. Выполненные задачи демонстрируют эффективность применения веб-технологий для решения сложных задач визуализации, а также потенциал для дальнейшего развития в области цифровой гуманитаристики.

Данная работа стала важным этапом в освоении методов фронтенд-разработки, работы с картографическими библиотеками и проектирования пользовательских интерфейсов, заложив фундамент для создания более сложных аналитических систем в будущем.

Список литературы

1. Leaflet.js Documentation (2023). [Электронный ресурс]. – URL: <https://leafletjs.com/>
2. Murray, S. *«*Interactive Data Visualization for the Web*»* (2017). – O’Reilly Media. – 450 с.
3. D3.js in Action (2021). – Manning Publications. – 512 с.
4. OpenStreetMap Wiki (2023). [Электронный ресурс]. – URL: [https://wiki.openstreetmap.org](https://wiki.openstreetmap.org/)<https://www.awwwards.com/>
5. Carto Blog (2023). «Spatio-Temporal Data Visualization». [Электронный ресурс]. – URL: <https://carto.com/blog/>
6. Грин, Д. «Исторические данные в цифровую эпоху: методы анализа и визуализации» (2020). – Москва: Изд-во МГУ. – 320 с.
7. Underwood, T. «Distant Horizons: Digital Evidence and Literary Change» (2019). – University of Chicago Press.