



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий (ИТ)
Кафедра инструментального и прикладного программного обеспечения (ИиППО)

КУРСОВАЯ РАБОТА

по дисциплине: Разработка клиентских частей интернет-ресурсов

по профилю: Разработка программных продуктов и проектирование информационных систем

направления профессиональной подготовки: 09.03.04 «Программная инженерия»

Тема: Клиентская часть интернет-ресурса «Реки мира»

Студент: Орлов Данил Дмитриевич

Группа: ИКБО-02-21

Работа представлена к защите _____ (дата) _____ /Орлов Д. Д./
(подпись и ф.и.о. студента)

Руководитель: Дешко Игорь Петрович, старший преподаватель

Работа допущена к защите _____ (дата) _____ /Дешко И. П./
(подпись и ф.и.о. рук-ля)

Оценка по итогам защиты: _____

_____/ _____ /
_____/ _____ /

(подписи, дата, ф.и.о., должность, звание, уч. степень двух преподавателей, принявших защиту)



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт информационных технологий (ИТ)

Кафедра инструментального и прикладного программного обеспечения (ИиППО)

ЗАДАНИЕ

на выполнение курсовой работы

по дисциплине: Разработка клиентских частей интернет-ресурсов

по профилю: Разработка программных продуктов и проектирование информационных систем

направления профессиональной подготовки: Программная инженерия (09.03.04)

Студент: Орлов Данил Дмитриевич

Группа: ИКБО-02-21

Срок представления к защите: 12.12.2022

Руководитель: Дешко Игорь Петрович, старший преподаватель

Тема: Клиентская часть интернет-ресурса «Реки мира»

Исходные данные: используемые технологии: HTML5, CSS3, JavaScript, редактор кода Visual Studio Code, наличие: интерактивного поведения веб-страниц, межстраничной навигации, внешнего вида страниц, соответствующего современным стандартам веб-разработки; технологий адаптивной верстки для полноценного отображения контента на различных браузерах и видах устройств. Нормативный документ: инструкция по организации и проведению курсового проектирования СМКО МИРЭА 7.5.1/04.И.05-18.

Перечень вопросов, подлежащих разработке, и обязательного графического материала: 1. Провести анализ предметной области разрабатываемой клиентской части интернет-ресурса. 2. Обосновать выбор технологий разработки клиентской части интернет-ресурса. 3. Создать пять и более веб-страниц интернет-ресурса. 4. Организовать межстраничную навигацию.

5. Реализовать слой клиентской логики веб-страниц с применением технологии JavaScript.

6. Провести оптимизацию веб-страниц и размещаемого контента для браузеров и различных видов устройств. 7. Создать презентацию по выполненной курсовой работе.

Руководителем произведён инструктаж по технике безопасности, противопожарной технике и правилам внутреннего распорядка.

Зав. кафедрой ИиППО: _____ /Р. Г. Болбаков/, «_____» _____ 2022 г.

Задание на КР выдал: _____ /И.П. Дешко/, «_____» _____ 2022 г.

Задание на КР получил: _____ /Д.Д. Орлов/, «_____» _____ 2022 г.

РЕФЕРАТ

Отчет 49 с., 55 рис., 1 табл., 8 источн.

ИНТЕРНЕТ РЕСУРС, САЙТ, РЕКИ, КЛИЕНТСКАЯ ЧАСТЬ, СТРАНИЦА

Объект исследования – визуальное представление сайта про Реки мира в сети интернет.

Предмет исследования – клиентская часть интернет-ресурса «Реки мира».

Цель работы – разработка клиентской части интернет-ресурса «Реки мира».

В главе «Общие сведения» описывается наименование интернет-ресурса, функциональное назначение и прикладное программное обеспечение.

В главе «Описание логической структуры» содержится анализ предметной области, выбор технологий разработки интернет-ресурса, подробное описание создания веб-страниц, межстраничной навигации и клиентской логики при помощи технологий HTML5, CSS3 и JavaScript.

В главе «Оптимизация клиентской части интернет-ресурса» содержится описание способов оптимизации продукта для разных устройств и браузеров.

В заключении описываются выполненные задачи, в результате разработки интернет-ресурса на тему «Клиентская часть интернет-ресурса «Реки мира».

В ходе выполнения курсовой работы была проанализированная предметная область и разработан продукт клиентской части интернет-ресурса на тему «Реки мира».

СОДЕРЖАНИЕ

РЕФЕРАТ	3
СОДЕРЖАНИЕ	4
ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ	6
ВВЕДЕНИЕ	7
1 ОБЩИЕ СВЕДЕНИЯ	9
1.1 Наименование клиентской части интернет-ресурса	9
1.2 Функциональное назначение как совокупность свойств клиентской части интернет-ресурса	9
1.3 Прикладное программное обеспечение, необходимое для разработки и функционирования интернет-ресурса	9
2 ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ	10
2.1 Анализ предметной области разрабатываемой клиентской части интернет-ресурса	10
2.2 Выбор технологий разработки клиентской части интернет-ресурса	11
2.3 Создание веб-страниц клиентской части интернет-ресурса с использованием технологий HTML5, CSS3 и JavaScript	12
2.4 Создание межстраничной навигации	17
2.5 Реализация слоя клиентской логики веб-страниц с применением технологии JavaScript	Ошибка! Закладка не определена.
2.5.1 Используемые библиотеки	18
2.5.2 Загрузка списка рек и последующие действия	20
2.5.3 Интерактивный глобус на основной странице	22
2.5.4 Сайдбар: список субъектов территории	24
2.5.5 Сайдбар: поиск рек по названию	26
2.5.6 Панель навигации: выбор темы страницы	27
2.5.7 Панель навигации: выбор языка страницы	29
2.5.8 Панель навигации: выбор типа территории	31
2.5.9 Панель навигации: информация о проекте	31
2.5.10 Функции для мобильной версии	32

2.5.11 Реки по территории: вертикальный слайдер	33
2.5.12 Реки по территории: самые длинные реки	35
2.5.13 Реки по территории: список рек	36
3 ОПТИМИЗАЦИЯ ИНТЕРНЕТ-РЕСУРСА	40
3.1 Оптимизация клиентской части интернет-ресурса для основных браузеров	40
3.2 Оптимизация клиентской части интернет-ресурса для стационарных и мобильных устройств	43
ЗАКЛЮЧЕНИЕ	48
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	49

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

В настоящем отчете применяют следующие сокращения и обозначения:

CSS	– Cascading Style Sheets (формальный язык описания внешнего вида документа, написанного с использованием языка разметки)
HTML	– HyperText Markup Language (стандартизированный язык разметки документов для просмотра веб-страниц в браузере)
JS	– JavaScript (язык программирования)

ВВЕДЕНИЕ

Сегодня существует множество интернет-ресурсов практически любой тематики, каждому из которых найдено применение. Человек с информационным устройством, имеющим доступ к сети интернет способен получить любую необходимую ему информацию, .

Актуальность темы обусловлена тем, что, вопреки ожиданиям, в сети интернет не оказалось ни одного ресурса, который был бы направлен на хранение и отражение всей доступной информации о всех реках мира.

Объектом курсовой работы является визуальное представление сайта про Реки мира в сети интернет.

Предмет исследования – клиентская часть интернет-ресурса «Реки мира».

Целью данной курсовой работы является разработка, тестирование и отладка интернет-ресурса на тему «Реки мира» с применением технологий HTML5, CSS3, JavaScript, с организацией межстраничной навигации, с оптимизацией веб-страницы и размещаемого контента для браузеров и различных видов устройств.

В связи с поставленной целью в данной работе решаются следующие задачи:

- Анализ предметной области разрабатываемого интернет-ресурса;
- Изучение синтаксиса технологий HTML5 [\[1\]](#), CSS3 [\[2, 3\]](#) и JavaScript[\[4, 5\]](#);
- Создание интернет-ресурса по заданной теме;
- Организация межстраничной навигации;
- Оптимизация веб-страниц для различных браузеров и устройств.

В ходе выполнения работы были использованы методы наблюдения, сравнения, анализа, классификации, обобщения, описания и моделирования.

Информационной базой исследования выступили, знания, полученные на основе выполнения практических работ в течение курса «Разработка

клиентских частей интернет-ресурсов», а также дополнительные источники информации.

Работа состоит из введения, перечня сокращений, содержания, трех основных разделов, заключения и списка использованных источников.

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Наименование клиентской части интернет-ресурса

В качестве наименования продукта клиентской части интернет-ресурса было выбрано название, соответствующее выбранной теме – «Riverwood».

1.2 Функциональное назначение как совокупность свойств клиентской части интернет-ресурса

Данный интернет-ресурс направлен на пользователей, интересующихся географией, в частности реками.

Основная задумка сайта заключена в создании доступного источника информации о всех реках мира, протяженность которых более ста километров.

На сайте имеется простой и понятный поиск рек по названию, возможность менять тему страницы и язык. Кроме того, имеется возможность выбора типа территории для навигации - по стране или по материку.

Сайт разработан для всех основных устройств просмотра интернет-ресурсов, включая стационарные устройства с разрешением экрана от 800x600 до 1920x1080 и мобильные устройства с разрешением экрана от 320x640 и выше, а также для все основных браузеров.

1.3 Прикладное программное обеспечение, необходимое для разработки и функционирования интернет-ресурса

В ходе тестирования различных средств разработки, был выявлен наиболее подходящий и удобный редактор кода Visual Studio Code.

В качестве основного браузера был выбран «Microsoft Edge», в котором имеется поддержка консоли разработчика. Для проверки кросс-браузерности созданного продукта были выбраны следующие браузеры:

- Mozilla Firefox;
- Google Chrome;
- Yandex Browser.

2 ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ

2.1 Анализ предметной области разрабатываемой клиентской части интернет-ресурса

Был проведен анализ предметной области разрабатываемого интернет-ресурса на тему «Реки мира». Это позволило составить информационную базу исследования.

Чтобы понять, какой контент будет содержать веб-сайт, с использованием метода наблюдения и метода сравнения, были посещены ресурсы [6] (Рисунок 2.1) и [7], рисунок 2.2.

Категория:Реки по алфавиту

Материал из Википедии — свободной энциклопедии

В эту категорию **автоматически** помещаются статьи, содержащие шаблон {{Река}}.

Непатрулированные правки в категории (первые 10 штук)

- Яхсу (история) (+16) (проверить) (2 часа)

Оглавление:

А Б В Г Д Е Ж З И Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ъ Ы Э Ю Я

Ада Без Вве Гек Дел Его Жев Зан Идо Йес Кас Лас Мар Нат Ожи Пер Рац Сев Тва Уим Фед Хар Цви Чел Шва Щеп Ыйв Эйф Юли Яко

Алт Бит Вин Гно Дин Ели Жиг Зее Имл Йон Кож Леп Мер **Нет** Опо Плу Рич Ско Тих Уни Фин Хим Цер Чес Шин Щит Ылг Эли Юри Ямб

Апп Боя Вок Гра Дор Еро Жуа Зин Иос Йос Кра Лов Мож Нов Осл Пре Рот Сле Три Урю Фоф Хоп Цин Чка Шта Щуч Ысп Эпо Юрь Яро

... в начало

Страницы в категории «Реки по алфавиту»

Показано 200 страниц из 34 349, находящихся в данной категории. Список ниже может не отражать последних изменений.

(Предыдущая страница) (Следующая страница)

А

- А (приток Мёне)
- А (приток Нете)
- А (река, впадает в Северное море)
- Аапста (река)
- Аба (приток Томи)
- Аба-эль-Кул
- Абава
- Абадим
- Абазинка (приток Кубани)
- Абазинка (приток Пшехи)
- Абазулгол
- Абай (приток Коксы)
- Абакан (река)
- Абакашис

- Агер
- Агдабанчай
- Аге
- Агеда (река)
- Агеевка (река)
- Агерзе (река)
- Агиапук (река)
- Агир
- Агитка
- Агиян
- Аглинручей
- Аглуона
- Агма
- Агнево
- Агно (река)
- Агой (река)

Рисунок 2.1 – Сайт [6]

10

Список рек мира по длине

Список самых длинных рек-речных систем на Земле, длина которых превышает 1000 км. Реки, упомянутые в речной системе не упоминаются отдельно (например: Иртыш и Обь). В списке приведены длины речных систем, то есть рек с учётом всех притоков. Определение длины реки считается очень сложной задачей, учитывая большое количество притоков. в связи с эти длины многих рек являются приближёнными.

Самыми длинными реками в мире считаются Амазонка и Нил. Причем длительное время учёные спорили о том, какая из двух крупнейших речных систем длиннее – Нил или Амазонка. Раньше считалось, что Нил, но данные экспедиций 2008 года установили местоположение истоков реки Укаяли, что поставило на первое место Амазонку.

N	Река	Длина (км)	Водосборный бассейн (км²)	Бассейн	Страны в
1	Амазонка	7062	6915000	Атлантический океан	Бразилия,
2	Нил	6700	3349000	Средиземное море	Бурунди, Е
3	Миссисипи – Миссури – Джефферсон	6275 (6420)	2980000	Мексиканский залив	США (98,5
4	Янцзы	5800 (6300)	1800000	Восточно-Китайское море	КНР
5	Хуанхэ	5464	745000	Бохайский залив	КНР
6	Обь – Иртыш	5410	2990000	Обская губа	Россия, Ка
7	Енисей – Ангара – Селенга – Идэр	5238	2580000	Карское море	Россия, Мс

Рисунок 2.2 – Сайт [7]

В ходе анализа было выявлено, что помимо основной страницы необходимо выделить страницу для просмотра рек по выбранному субъекту региона (континенту или стране), страницу на случай, если в базе данных нет рек у выбранного субъекта региона, страницу для просмотра информации об отдельной реке и страницу с контактами автора и информацией об интернет-ресурсе.

2.2 Выбор технологий разработки клиентской части интернет-ресурса

Интернет-ресурс создан с применением технологий HTML5, CSS3 и JS. С их помощью можно реализовать веб-сайты с привлекающим внимание пользователей дизайном.

Для разработки интернет-ресурса был определен редактор Visual Studio Code[8], который позволяет писать код на различных языках программирования, а наличие подсветки синтаксиса ускоряет разработку и

делает ее более удобной. Так же он обладает интуитивно понятным и простым интерфейсом и имеет множество других преимуществ: быстрая навигация, поиск и замена, командная палитра, одновременное редактирование, высокая степень настраиваемости. Это обосновывает выбор данного редактора.

Для просмотра страниц интернет-ресурса используется браузер Microsoft Edge. Он позволяет исследовать элементы ресурса и работать с контентом на различных устройствах.

2.3 Создание веб-страниц клиентской части интернет-ресурса с использованием технологий HTML5, CSS3 и JavaScript

В продукте курсовой работы было создано 5 веб-страниц. Header сайта – верхняя часть страницы, в которой расположена навигационная панель и логотип сайта, присутствует на всех страницах интернет-ресурса. Она имеет две вариации: на основной странице присутствуют возможность просмотра информации о проекте и возможность выбора типа территории. (Рисунок 2.3), на остальных страницах данные опции недоступны, но появилась кнопка, возвращающая пользователя на первоначальную страницу, рисунок 2.4.



Рисунок 2.3 – Навигационная панель основной страницы



Рисунок 2.4 – Навигационная панель остальных страниц

Это именно та страница, на которую пользователь попадает при первом посещении сайта. Здесь присутствует интерактивный глобус, позволяющий перейти на страницу с реками по территории. Слева расположен сайдбар, где находится список субъектов территории. Так же в сайдбаре реализован поиск рек по названию и возможность перехода на страницу с рекой или на страницу с реками по региону на случай, если пользоваться глобусом не представляется возможным или удобным, рисунок 2.5.

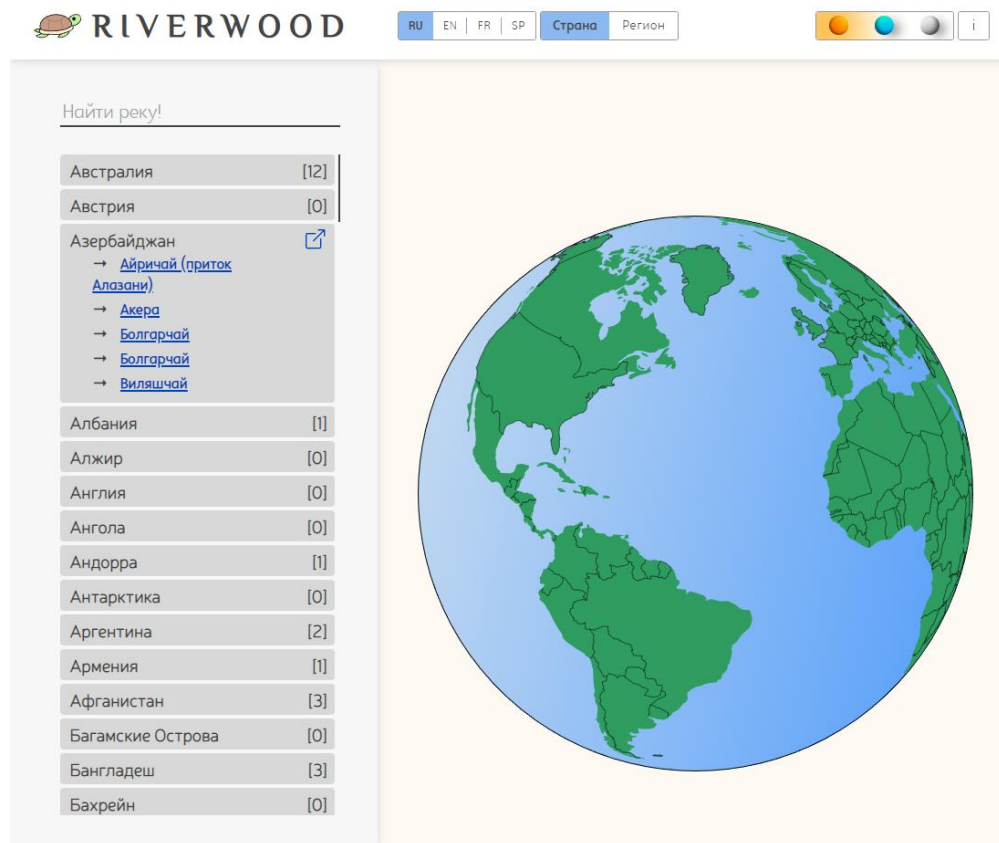


Рисунок 2.5 – Основная страница сайта

Нажав на соответствующую кнопку в header, пользователь попадает на страницу с информацией о проекте, рисунок 2.6.

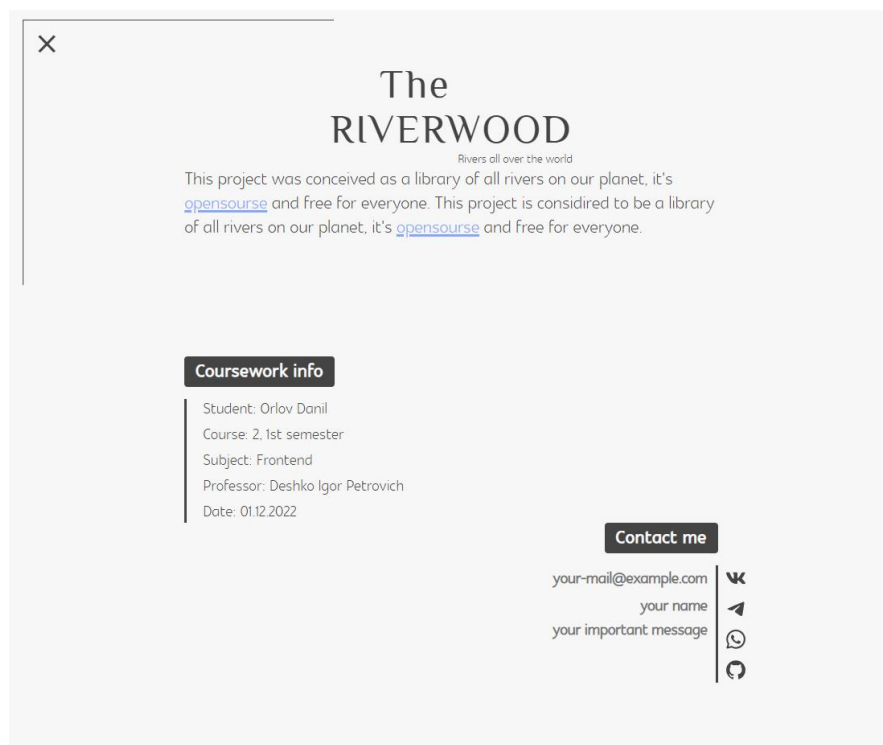


Рисунок 2.6 – Страница с информацией о проекте

Страница информации о реках по территории представлена в виде вертикального слайдера с пятью блоками контента, часть которых представлена ниже, рисунки 2.7 – 2.9.

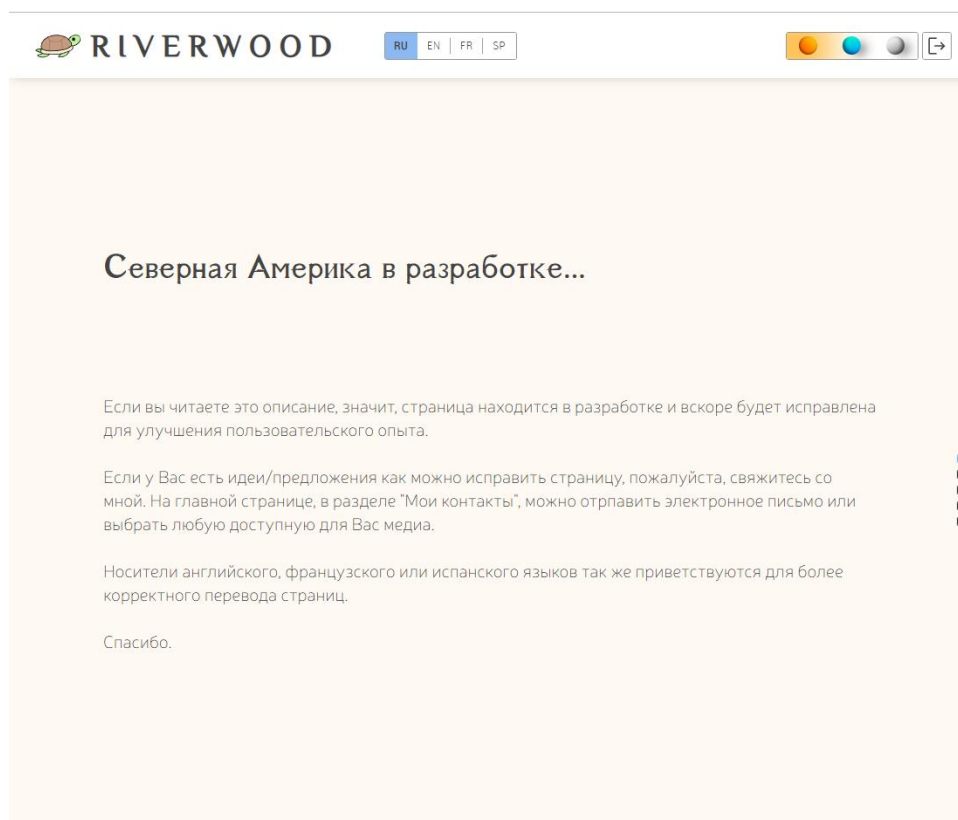


Рисунок 2.7 – Страница информации о реках по территории (Блок №1)

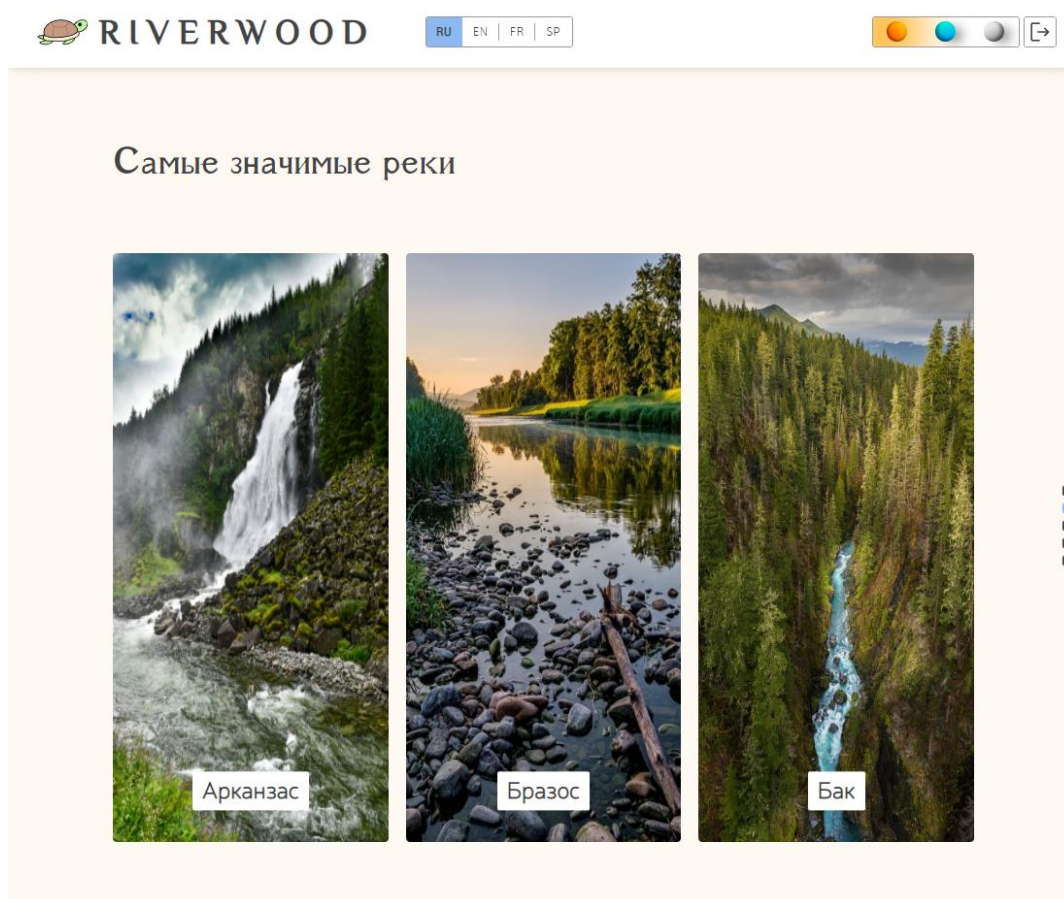


Рисунок 2.8 – Страница информации о реках по территории (Блок №2)

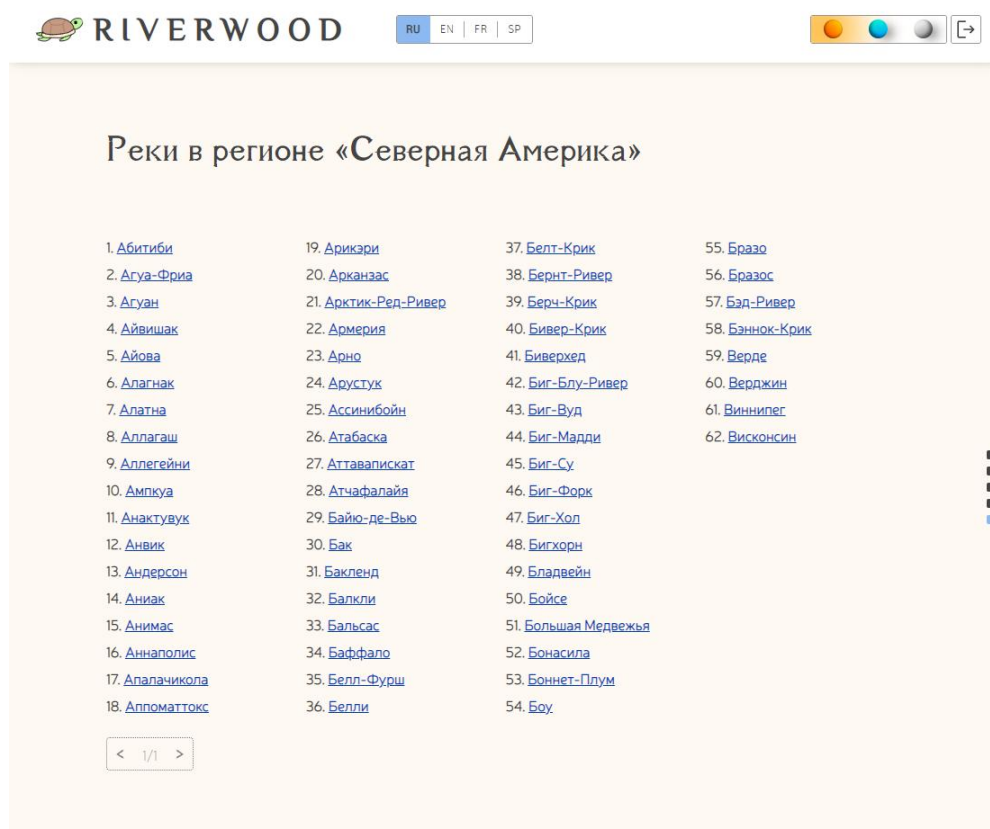


Рисунок 2.9 – Страница информации о реках по региону (Блок №5)

На странице информации о реке (рисунок 2.10) располагается ее название, список свойств и краткое описание. Так же есть возможность перейти на соответствующую страницу википедии.

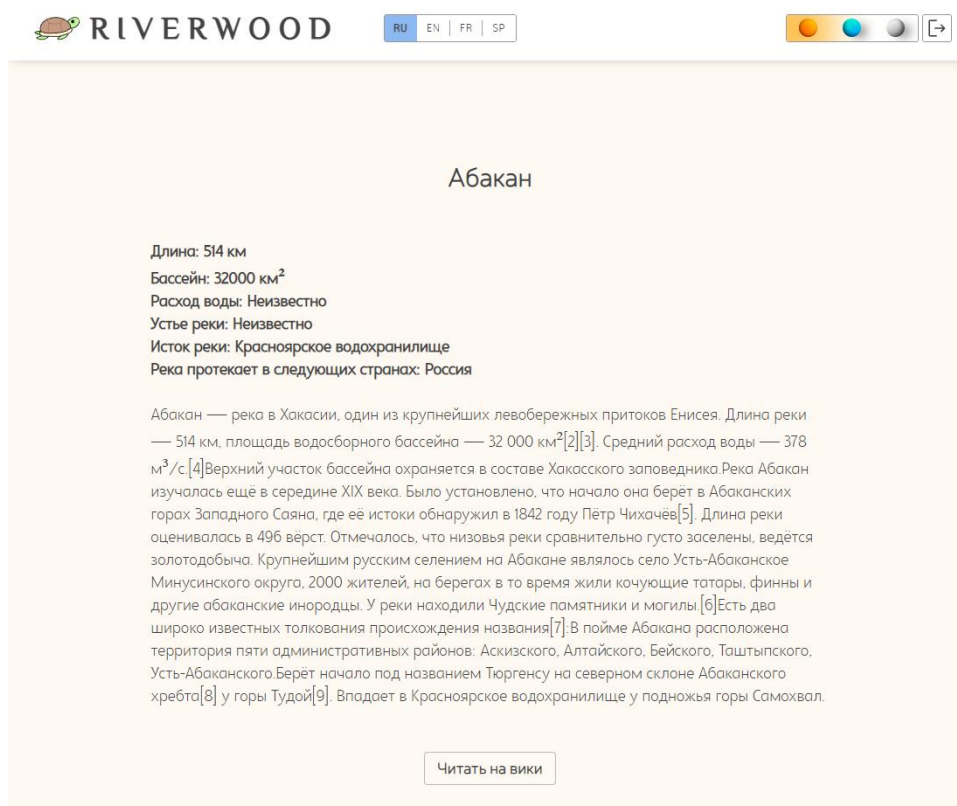


Рисунок 2.10 – Страница информации о реке

На страницу NotFound (Рисунок 2.11) пользователь попадает при попытке открыть страницу с информацией о реках по территории, у которой в базе данных отсутствуют реки, например, страницу с Антарктикой, рисунок 2.12.

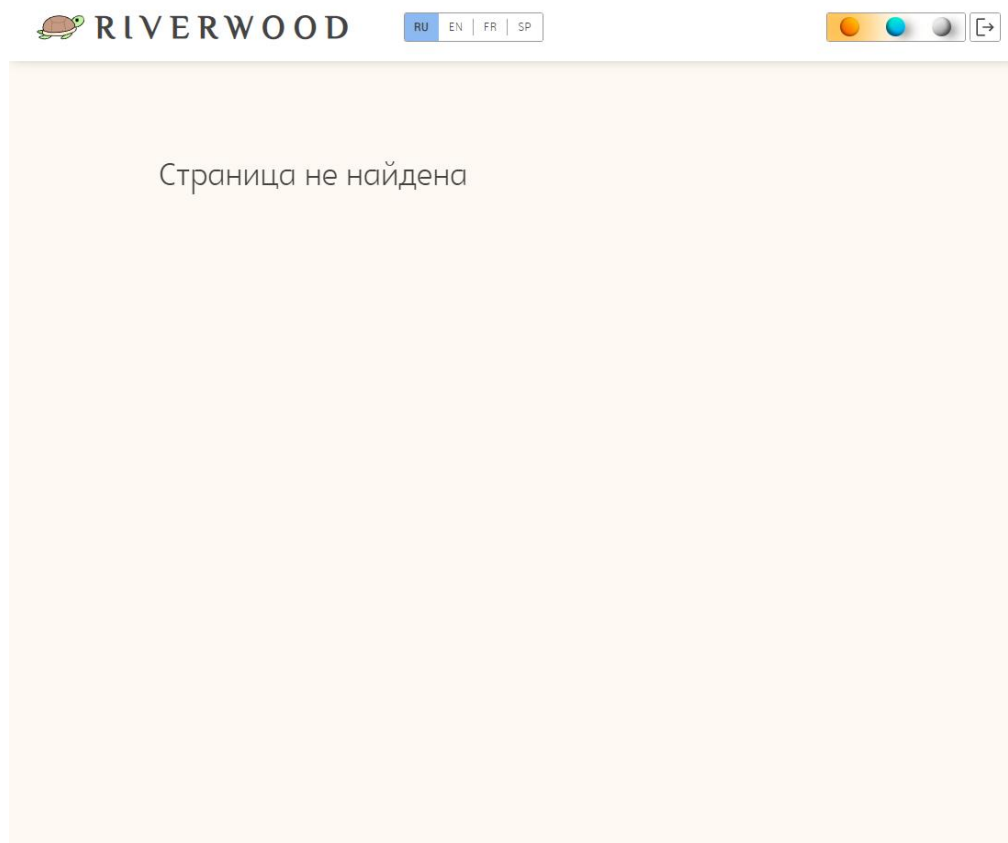


Рисунок 2.11 – Страница NotFound

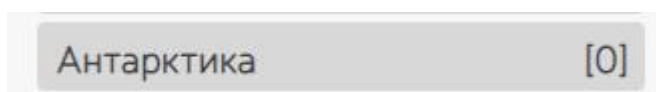


Рисунок 2.12 – Отсутствие рек у Антарктики

2.4 Создание межстраничной навигации

Приоритетами при проектировании структуры межстраничной навигации выступали в первую очередь эффективность и удобство. С любой дочерней страницы всегда можно вернуться на основную, а с основной можно перейти на любую дочернюю.

Структура межстраничной навигации представлена в виде графа на рисунке 2.13.



Рисунок 2.13 – Структура межстраничной навигации

2.5 Реализация слоя клиентской логики вебстраниц с применением технологии JavaScript

При написании функционала сайта активно использовалась технология JavaScript. Ее применение позволило разнообразить пользовательский опыт взаимодействия с сайтом, добавить большую интерактивность.

2.5.1 Используемые библиотеки

Для корректной работы интернет ресурса и, в то же время, для простоты разработки были использованы три библиотеки: d3.js, smtp.js и безымянная «библиотека» собственного сочинения, которая является упрощенной версией известной библиотеки jQuery. Первые две библиотеки будут раскрыты разделах, где описывается код с их применением. Листинг класса безымянной «библиотеки» представлен на рисунке 2.14, где показана часть функций и рисунке 2.15 с примером использования.

```

class JSFeatures {
  e; depthElem;

  > constructor(elem = null) { ...
  }
  > get(index = 0) { ...
  }
  > size() { ...
  }
  > find(elem, depth = false, fromConstructor = false) { ...
  }
  > not(elem) { ...
  }

  > empty() { ...
  }
  > css(obj) { ...
  }
  > animate(obj, transition, f = null) { ...
  }
  > fadeIn(duration, display, f) { ...
  }
  > fadeOut(duration, f) { ...
  }
  > scroll(y = null, duration = 0) { ...
  }

  // getters & setters
  > text(text = null) { ...
  }
  > value(text = null) { ...
  }
  > ihtml(text = null) { ...
  }
  > ohtml(text = null) { ...
  }

```

Рисунок 2.14 – Листинг класса собственной «библиотеки».

```

// найти элемент по ID, проиграть анимацию с длительностью 500мс,
// затем добавить элементу класс
$js(`#elemID`)
  .animate({'opacity': '0.5'}, 500)
  .addClass('new-class-name')

const $js = (str = null) => {
  return new JSFeatures(str);
};

```

Рисунок 2.15 – Листинг использования собственной «библиотеки»

2.5.2 Загрузка списка рек и последующие действия

Для корректного отображения контента на сайте все скрипты должны быть загружены после полной отрисовки страницы и после загрузки сокращенного объекта рек (Листинг кода приведен на рисунке 2.16), которого хватает для построения визуальной части главной страницы. После отрисовки страницы, сокращенный список рек меняется на более подробную версию (Листинг кода приведен на рисунке 2.17), которая нужна для корректного отображения других страниц. Разбиение списка рек на полную и сокращенную версии сделано в целях оптимизации и ускорения отрисовки основной страницы. Листинг функции, запускающей весь функционал загрузки списков рек и отрисовки страницы в нужном порядке приведен на рисунке 2.18.

```
RIVERS = [{
  "ru": "Абава",
  "en": "Абава",
  "fr": "Абава",
  "sp": "Абава",
  "location": "Латвия"
},
{
  "ru": "Абакан ",
  "en": "Абакан ",
  "fr": "Абакан ",
  "sp": "Абакан ",
  "location": "Россия"
},
{
  "ru": "Абакашис",
  "en": "Абакашис",
  "fr": "Абакашис",
  "sp": "Абакашис",
  "location": "Бразилия"
},
{
```

Рисунок 2.16– Листинг фрагмента сокращенного списка рек

```

RIVERS = [{
  "ru": "Абава",
  "en": "Абава",
  "fr": "Абава",
  "sp": "Абава",
  "ruLink": "%D0%90%D0%B1%D0%B0%D0%B2%D0%B0",
  "enLink": "%D0%90%D0%B1%D0%B0%D0%B2%D0%B0",
  "frLink": "%D0%90%D0%B1%D0%B0%D0%B2%D0%B0",
  "spLink": "%D0%90%D0%B1%D0%B0%D0%B2%D0%B0",
  "length": "129",
  "pool": "2042",
  "consumption": "",
  "location": "Латвия",
  "ruHead": "",
  "enHead": "",
  "frHead": "",
  "spHead": "",
  "ruEstuary": "Вента",
  "enEstuary": "Вента",
  "frEstuary": "Вента",
  "spEstuary": "Вента",
  "image": "2/25/Abava_River%283%29.jpg/250px-Abava_River%283%29.jpg",
  "ruInfo": "Абава – река в Латвии, правый приток реки Вента. Река берёт нача",
  "enInfo": "Абава – река в Латвии, правый приток реки Вента. Река берёт нача",
  "frInfo": "Абава – река в Латвии, правый приток реки Вента. Река берёт нача",
  "spInfo": "Абава – река в Латвии, правый приток реки Вента. Река берёт нача",
},
],

```

Рисунок 2.17– Листинг фрагмента полного списка рек

```

document.addEventListener("DOMContentLoaded", () => {
  let RiversLight = LoadScript(`src/RiversLight.js`, false);

  RiversLight.onload = () => {
    for (let i = 0; i < AreaObj.COUNTRIES.length; i++) { ...
    }
    for (let i = 0; i < AreaObj.REGIONS.length; i++) { ...
    }
    log(RIVERS.length)

    Phone();
    EarthReady();

    AsideObj.Recreate();
    About();
    Search();

    let RiversFull = LoadScript(`src/Rivers.js`, true);
    RiversFull.onload = () => { ...
    }
  }
});

```

Рисунок 2.18 – Листинг загрузочной функции

2.5.3 Интерактивный глобус на основной странице

Для реализации глобуса была использована ранее упомянутая библиотека d3.js, которая реализуется во вспомогательном классе D3 (Листинг кода приведен на рисунке 2.19). Так же для обработчиков событий были разработаны отдельные классы (Листинг кода приведен на рисунках 2.20 – 2.22). Листинг функции, запускающей работу вышеописанных классов представлен на рисунке 2.23.

```
class D3 {  
  static QueueData() { ...  
  }  
  static LoadData = (error, world) => { ...  
  }  
  static RenderGlobe() { ...  
  }  
  
  static setScale = () => { ...  
  }  
  static setAngles() { ...  
  }  
  static setTimer() { ...  
  }  
  static timerTick = (elapsed) => { ...  
  }  
  static setRotation = (state) => { ...  
  }  
  
  static AssignBackground() { ...  
  }  
  static RemoveStars() { ...  
  }  
}
```

Рисунок 2.19 – Листинг класса D3

```
class D3_CLICK {  
  static setClick() {  
    canvas.on('click', AreaPage(getObj(currentPolygon) || currentRegion))  
  }  
}
```

Рисунок 2.20 – Листинг класса, связанного с событием click

```

class D3_HOVER {
  static setHover() {
    canvas.on('mousemove', D3_HOVER.CountryHover)
  }
  static CountryHover = () => { ...
  }

  static setCountry = () => { ...
  }
  static setRegion = () => { ...
  }
  static getPolygon = (event) => { ...
  }
  static setName = () => { ...
  }
}

```

Рисунок 2.21 – Листинг класса, связанного с событием hover

```

class D3_DRAG {
  static setDrag() {
    canvas.call(
      d3.drag()
        .on('start', D3_DRAG.Start)
        .on('drag', D3_DRAG.Drag)
        .on('end', D3_DRAG.End)
    )
  }

  static Start() { ...
  }
  static Drag() { ...
  }
  static End() { ...
  }
}

```

Рисунок 2.22 – Листинг класса, связанного с событием drag


```

//
const starsAmount =      150
const scaleFactor =      1
const rotationDelay =    5000 // * 1000
const rotationDirection = -1
const degPerSec =        7
const angles =           { x: 50, y: -20, z: 0 }
const colorLand =        '#309d60' // '#F19BFE'
const colorActive =      '#00000099' // '#F6C1BC'
const styleBorders =     { 'color': '#000', 'thickness': 0.5 };
const styleGlobeBorder = { 'color': '#000', 'thickness': 2 };
// all we need to work with
const EarthReady = () => {
  D3.QueueData();

  D3_DRAG.setDrag();
  D3_HOVER.setHover();
  if (!isPhone()) D3_CLICK.setClick();

  $canvas.onEvent('mouseleave', () => {
    setName('')

    currentPolygon = null;
    currentRegion = null;
    D3.RenderGlobe();

    D3.setRotation(true);
  });

  window.addEventListener('resize', () => {
    D3.setScale();
    D3.RenderGlobe();
  });
}

```

Рисунок 2.23 – Листинг функции запуска

2.5.4 Сайдбар: список субъектов территории

В соответствии с загруженным объектом рек в сайдбар добавляется список элементов в соответствии с субъектами текущего типа территории, (Листинг элемента представлен на рисунке 2.24). При раскрытии элемента списка к нему добавляется и сразу же отображается подсписок рек (Листинг кода приведен на рисунке 2.25). Для оптимизации и уменьшения количества элементов DOM при закрытии элемента списка подсписок рек удаляется. Листинг класса `AsideObj`, отвечающего за список субъектов территории представлен на рисунке 2.26.


```

class AsideObj {
    static $list;

    static CreatelistElem(text, length) {
        return `...
    };
}

static CreateRiversList(riversArr) { ...
}

static ClickEvent = (e) => { ...
}

static Recreate() {
    const $areaList = $js(`#areaList`);
    let newHTML = '';
    AreaObj.currentList.forEach(area => {
        newHTML += AsideObj.CreatelistElem(area.name, area.rivers.length);
    });
    $areaList.ihtml(newHTML);

    AsideObj.$list = $js(`.list-title`);
    AsideObj.$list.onClick(AsideObj.ClickEvent)
}
}

```

Рисунок 2.26 – Листинг класса AsideObj

2.5.5 Сайдбар: поиск рек по названию

Принцип работы поиска таков: изначально в соответствии с загруженным объектом рек создается список рек (Листинг кода приведен на рисунке 2.27). При наборе символов показываются этот список и навигация по поисковой системе, заменяя список субъектов территории. Затем вызывается функция, которая, проходясь по списку, ищет соответствия и, если таковое находится, искомое значение в элементе списка подсвечивается, после чего список пролистывается до первого найденного совпадения. Если из поля ввода были удалены все символы, список всех рек и навигация по поисковой системе исчезают, а список субъектов текущего типа территории. Ниже, на рисунке 2.28 представлен листинг функции поиска.

```

let riversHTML = '';
RIVERS.forEach(river => {
  riversHTML += `
    <li>
      <a href="${river.link}" target="_blank" rel="noopener noreferrer">
        ${river.name}
      </a>
    </li>
  `;
});
$riverList.ihtml(riversHTML);

```

Рисунок 2.27 – Листинг создания списка всех рек

```

$input.onEvent(`input`, () => {
  let val = $input.value();
  if (!val) return HideSearch();

  if (newSearch) { ...
  }

  $counter.text(`0/0`);
  if (removeOthers) $riverList.find('li').css({'display': 'list-item'});

  let newHTML = '';
  $aList.every((e, i) => {
    let text = e.innerText;
    let indexOf = text.toUpperCase().indexOf(val.toUpperCase());

    newHTML += `<li ${removeOthers ? 'style="display:none">' : '>'}`;
    if (indexOf > -1) { ...
    }
    else
      newHTML += `<a href="${RIVERS[i].link}" target="_blank" rel="no
    newHTML += '</li>'

  });
  $riverList.ihtml(newHTML)

  if (!newSearch) { ...
  }
  newSearch = false;
});

```

Рисунок 2.28 – Листинг поиска рек по названию

2.5.6 Панель навигации: выбор темы страницы

За все манипуляции с темой страницы отвечает класс `ThemesObj`. Смена темы происходит за счет изменения переменных CSS, значения для

которых прописаны в объекте темы (Листинг кода приведен на рисунке 2.29). Так же для лучшего пользовательского опыта сделана анимация при смене темы. Листинг класса представлен на рисунке 2.30.

```
static Light = {
  '--based-color': '#444',
  '--revert-color': '#fff',
  '--accent-color': '#8cb9f1',
  '--light-color': '#fff',
  '--dark-color': '#444',
  '--disabled-color': '#bbb',

  '--main-background': '#fef9f2',
  '--main-background-size': 'initial',
  '--globe-background': 'linear-gradient(104deg, #a0c7ef8f, #0070ffad)',
  '--area-title-background': '#000',
  '--about-background': '#f7f7f7',
  '--edge-shadow': '0px 1px 14px -7px #777',

  '--h-background': '#fff',
  '--controls-border': '1px solid #999',
  '--controls-disabledcolor': '#585858',
  '--controls-hoverBackground': '#e4e4e5',
  '--settings-background': '#d7ddf3c2',
  '--settings-filter': 'drop-shadow(6px 8px 19px #bbb)',
  '--settingsBtn-background': '#e3dfdf',
  '--settingsBtn-border': '1px solid #cfcfcf',

  '--sd-background': '#f7f7f7',
  '--placeholder-color': '#a0a0a0',
  '--list-background': '#5050502e',
  '--list-headerColor': '#000',
  '--list-linkColor': '#0142b9',
  '--list-fontWeight': '900'
}
```

Рисунок 2.29 – Листинг объекта светлой темы

```

class ThemesObj {
    static current;
    static isDynamic;
    static $themeAnim = $js(`#themeAnimation`);

    static Start() { ...
    }
    static getNextTheme = (id) => { ...
    }

    static setTheme(id = null) { ...
    }
    static Animate($btn) { ...
    }

    static Light = { ...
    }
    static Dark = { ...
    }
    static DynamicLight = { ...
    }
    static DynamicDark = { ...
    }
    static Default = ThemesObj.Light;
}
ThemesObj.Start()

```

Рисунок 2.30 – Листинг класса ThemesObj

2.5.7 Панель навигации: выбор языка страницы

За все манипуляции, связанными с языком страницы отвечает класс LanguagesObj. При выборе языка всем необходимым элементам присваивается соответствующий текст. Стоит отметить, что различия данного класса на различных страницах несущественны и состоят лишь в измененном объекте с текстом (Листинг кода представлен на рисунке 2.31) и методе присваивания элементам на странице соответствующего текста. Листинг класса LanguagesObj представлен на рисунке 2.32.

```

static TEXT = {
  'ru': [
    'Страна',
    'Регион',
    ['К Земле', 'См. в списке'],
    'Найти реку!'
  ],
  'en': [
    'Country',
    'Region',
    ['To the Earth', 'See in list'],
    'Find river!'
  ],
  'fr': [
    'Pays',
    'Region',
    ['Vers La Terre', 'Voir la liste'],
    'Trouvez la riviere!'
  ],
  'sp': [
    'Pais',
    'Region',
    ['A La Tierra', 'Ver en la lista'],
    'Encontrar el rio!'
  ]
}

```

Рисунок 2.31 – Листинг объекта с текстом основной страницы

```

class LanguagesObj {
  static current;
  static CONTENT;
  static LANGTYPES = { ...
  };
  static TEXT = { ...
  }

  static Start() { ...
  }

  static setLang(id = null) { ...
  }
  static TranslatePage(GLOBE_VIEW) {
    $js(`#countryBtn`).text(LanguagesObj.currentText[0]);
    $js(`#regionBtn`).text(LanguagesObj.currentText[1]);
    $js(`#scrollBtn`).text(LanguagesObj.currentText[2][GLOBE_VIEW ? 1 : 0]);
    $js(`#searchInput`).attr('placeholder', LanguagesObj.currentText[3])
  }
  static TranslateObj(obj) { ...
  }
}
LanguagesObj.Start()

```

Рисунок 2.32 – Листинг класса LanguagesObj

2.5.8 Панель навигации: выбор типа территории

За все манипуляции, связанными с текущим типом территории отвечает класс `AreaObj`. С помощью него определяется, что будет выделяться на глобусе: страны или материки, и что будет показано в сайдбаре соответственно. `AreaObj` хранит в себе два объекта для типов территории, (Листинг фрагмента приведен на рисунке 2.33). Листинг класса `AreaObj` представлен на рисунке 2.34.

```
static COUNTRIES = [
  { 'id': '12', 'en': 'Algeria', 'fr': 'Algerie', 'sp': 'Angelia', 'ru': 'Алжир' },
  { 'id': '24', 'en': 'Angola', 'fr': 'Angola', 'sp': 'Angola', 'ru': 'Ангола' },
  { 'id': '72', 'en': 'Botswana', 'fr': 'Botswana', 'sp': 'Botswana', 'ru': 'Ботсвана' },
```

Рисунок 2.33 – Листинг фрагмента объекта типа территории

```
class AreaObj {
  static current;
  static currentList;

  static Start() { ...
  }

  static isRegion() { ...
  }

  static setArea(id = null) { ...
  }

  static setNames() { ...
  }

  static AREATYPES = {
    'country': 'country',
    'region': 'region'
  };

  static COUNTRIES = [ ...
  ]

  static REGIONS = [ ...
  ]
}
AreaObj.Start()
```

Рисунок 2.34 – Листинг класса `AreaObj`

2.5.9 Панель навигации: информация о проекте

Для отправки электронных сообщений на странице с информацией об интернет-ресурсе была использована библиотека `smtp.js`, листинг кода с ее использованием представлен на рисунке 2.35.

```

$sendEmail.onClick((e, event) => {
  if (e.hasClass('btn-active')) {
    Email.send({
      SecureToken: "0e080ffd-8308-425e-8cf9-093fb6419f45",
      To : 'kaarst@ya.ru',
      From : "kaarst@ya.ru", // fixme
      Subject: 'Riverwood: ' + $emailName.value(),
      Body : $emailLetter.value() + '\n\n From ' + $emailAddress.value()
    })
    e.text('Sent!')
    $emailAddress.value('');
    $emailName.value('');
    $emailLetter.value('');
    $errorState.value('');
    e.css({
      'color': 'var(--based-color)',
      'background': 'var(--accent-color)'
    })
    setTimeout(() => {
      setBtn(false);
      e.css({
        'color': '#',
        'background': '#'
      })
    }, 3000);
  }
  else {
    showError('Why did you... click me...')
  }
})

```

Рисунок 2.35 – Листинг функции с применением библиотеки smtp.js

2.5.10 Функции для мобильной версии

В мобильной версии панель навигации изначально скрыта, а для ее отображение в «шапку» добавлена кнопка. Для нее и для отображения на мобильной версии первоначально глобуса, а не списка, была написана соответствующая функция. Ниже представлен листинг программы, рисунок 2.36.


```

const Phone = () => {
  const $mainChilds = $js();
  $mainChilds.e = [$js(`main aside`).get(), $js(`main article`).get()]
  const $settings = $js(`.header-settings`)
  const scrollSpeed = 500;
  const settingsSpeed = 350;
  const mediaWidth = 768;
  let GLOBE_ACTIVE = true;
  let SETTINGS_ACTIVE = false;

  const ShowSettings = () => { ...
  }
  const HideSettings = () => { ...
  }

  const ShowGlobe = () => { ...
  }
  const ShowList = () => { ...
  }

  (isPhone() || window.innerWidth < mediaWidth) && ShowGlobe() && LanguagesObj.TranslatePage
  window.onresize = (e) => {
    (!isPhone() && window.innerWidth < mediaWidth && ShowGlobe() && HideSettings()) ||
    (window.innerWidth >= mediaWidth && ShowList() && ShowSettings())
  }

  $js(`#settingsBtn`).onEvent('mousedown', () => {
    SETTINGS_ACTIVE ?
      HideSettings() :
      ShowSettings();
  })
  $js(`#scrollBtn`).onEvent('mousedown', () => {
    GLOBE_ACTIVE ?
      ShowList() :
      ShowGlobe()
    setTimeout(() => {
      HideSettings();
      LanguagesObj.TranslatePage(GLOBE_ACTIVE);
    }, scrollSpeed * (6/10));
  })

  $js(`main`).onClick((e) => { ...
  });
}

```

Рисунок 2.36 – Функция для мобильной версии сайта

2.5.11 Реки по территории: вертикальный слайдер

Как уже говорилось ранее, на странице с информацией о реках по территории есть вертикальный слайдер с пятью блоками. Он зациклен и может прокручиваться бесконечно. Ниже представлен листинг программы, рисунок 2.37 – 2.38.

```

const InfiniteScroll = () => {
  let scrollAllowed = true;
  let prevDelta = 0;
  const $main = $js(`main`);
  const $views = $js(`main article`);
  const $viewsList = $views.toJSF();
  const $navs = $js(`#scrollNav span`);
  const $navsList = $navs.toJSF();
  let currentIndex = 0;
  let timer;
  // phone touch events
  let touchEvent = null;
  let direction = null;

  // hotfix for transitionless first scroll
  for(let i = currentIndex; i < $viewsList.size(); i++) {
    $viewsList.get(i).css({
      'transform': `translateY(0%)`
    });
  }

  function scrollNext(index = null) {
    index = index != null ? index : (currentIndex + 1) % $navs.size()
    if (currentIndex == $navs.size() - 1 && index == 0) return scrollPrev(0)

    $navsList.get(currentIndex).removeClass(`active`);
    $navsList.get(index).addClass(`active`)

    for(let i = currentIndex; i < index; i++) {
      $viewsList.get(i).css({
        'transform': `translateY(-${(i + 1) * 100}%)`
      });
    }
    currentIndex = index;
    $viewsList.get(currentIndex).css({
      'transform': `translateY(${currentIndex * -1 * 100}%)`
    });
  }
  > function scrollPrev(index = null) {...
  }

```

Рисунок 2.37 – Листинг слайдера (Часть 1)

```

> function scrollPrev(index = null) {...
}
$main.onEvent('mousewheel', (el, e) => {
  if (!scrollAllowed && Math.abs(e.deltaY) > prevDelta)
    prevDelta = Math.abs(e.deltaY);
  if (Math.abs(e.deltaY) == 0) prevDelta = 0;
  if (scrollAllowed && Math.abs(e.deltaY) >= prevDelta) {
    prevDelta = Math.abs(e.deltaY);
    scrollAllowed = false;
    e.deltaY >= 0 ? scrollNext() : scrollPrev()
    clearTimeout(timer)
    timer = setTimeout(() => {
      scrollAllowed = true;
    }, 1000);
  }
});

$main.onEvent("touchmove", (e, event) => {
  touchEvent = touchEvent || event;
  direction = (event.touches[0].pageY - touchEvent.touches[0].pageY) > 0;
});
$main.onEvent("touchend", () => {
  if (touchEvent) {
    direction ? scrollPrev() : scrollNext();
    touchEvent = null;
    direction = null;
  }
});

$navs.onClick((el, e, i) => {
  if (el.hasClass('active')) return;
  if (i > currentIndex) scrollNext(i);
  else scrollPrev(i);
});
}

```

Рисунок 2.38 – Листинг слайдера (Часть 2)

2.5.12 Реки по территории: самые длинные реки

В отражении трех самых длинных рек нет ничего примечательного, за исключением эффекта исчезающего текста в описании реки (Листинг кода приведен на рисунке 2.39), который достигается с помощью выставления уменьшаемой прозрачности для каждого слова. В телефонной версии сайте текст скрыт в целях адаптивности. Листинг кода представлен на рисунке 2.40.

```
function HidingText(topRivers) {
  let rect = $js(`#significant section p`).rect();
  let step = (Math.log(rect.width) * Math.log(rect.width) * rect.width / rect.height) * 1.5;
  for(let i = 0; i < topRivers.length; i++) {
    let opacity = 1;
    let html = '';
    let words = topRivers[i].info.split(' ');
    for(let j = 0; j < step; j++) {
      html += `<span style="opacity:${opacity}">${words[j]} </span>`;
      opacity -= 1/step;
    }
    $js(`#significant section p`).find(i).innerHTML(html);
  }
}
```

Рисунок 2.39 –Листинг функции исчезающего текста

```
const Significant = () => {
  function TopByLength(areaItem) { ... }
  function HidingText(topRivers) { ... }

  let topRivers = TopByLength(areaItem);
  $js(`#significant section h3`).find(0).text(topRivers[0].name.replace(/\([\D\d^\\]+\)/g, ''));
  $js(`#significant section h3`).find(1).text(topRivers[1].name.replace(/\([\D\d^\\]+\)/g, ''));
  $js(`#significant section h3`).find(2).text(topRivers[2].name.replace(/\([\D\d^\\]+\)/g, ''));

  if (!isPhone() && !window.innerWidth < 768)
    HidingText(topRivers)
  window.onresize = () => {
    HidingText(topRivers)
  }

  $js(`#significant section button`).onClick((e) => {
    let index = e.parent('div').parent('div').index();
    window.location.href = topRivers[index].link;
  })
}
```

Рисунок 2.40 – Листинг функции, отображающей самые длинные реки

2.5.13 Реки по территории: список рек

Список рек представляет собой горизонтальный слайдер. В целях адаптивности при изменении размера экрана выбирается определенное количество колонок списка (Листинг кода представлен на рисунке 2.41), и в соответствии с размером ячейки определяется количество блоков в списке, затем список заполняется реками (Листинг кода представлен на рисунке 2.42). Листинг функции слайдера представлен на рисунке 2.43-2.44.

```
function getCols() {
  if (window.innerWidth < 700) return 2;
  else if (window.innerWidth < 900) return 3;
  else return 4;
}
```

Рисунок 2.41 – Определение количества колонок в списке

```

const ListLayout = () => {
  function CreateItem(river, index) { ...
  }
  function getCols() { ...
  }
  const $list = $js(`#listWrap`);
  let rect = $list.rect();
  let cellRect = $js(`#nextBtn`).rect();
  cellRect = { ...
  }
  let col = getCols()
  let row = Math.floor(rect.height / (cellRect.height))
  let newHTML = '';
  let current = 0;
  let mod = 0;
  let riversAmount = areaItem.rivers.length;
  while (current < riversAmount) {
    let pageHTML = '<div>';
    for (let i = 0; i < col && current < riversAmount; i++) {
      for (let j = 0; j < row && current < riversAmount; j++) {
        current++;
        pageHTML += CreateItem(areaItem.rivers[i * (row) + j + mod], i * (row)
      )
      }
      mod = current;
      newHTML += pageHTML + '</div>';
    }
    $list.ihtml(newHTML);
    $list.find(`a`).onClick((e, event, index) => {
      RiverPage(areaItem.rivers[index])
    })
  }
  ListSwipes();
}

```

Рисунок 2.42 – Заполнение списка рек


```

const ListSwipes = () => {
  const $views = $js(`#listWrap div`);
  const $viewsList = $views.toJSF();
  const $listPage = $js(`#listPage`);
  const transition = 'all 700ms ease-in-out';
  let currentIndex = 0;

  if ($views.size() > 1) { ...
  }

  $listPage.value(`1/${$views.size()}`);

  function scrollNext(index = null) {
    index = index != null ? index : (currentIndex + 1) % $views.size()
    if (currentIndex == $views.size() - 1 && index == 0) { ...
    }

    for(let i = 0; i < currentIndex; i++) {
      $viewsList.get(i).animate({
        'transform': `translateX(${(i + 1) * -1 * 100}%)`
      }, transition);
    }
    $viewsList.get(currentIndex).animate({
      'transform': `translateX(${(currentIndex + 1) * -1 * 100}%)`
    }, transition);
    currentIndex = index;
    $viewsList.get(currentIndex).animate({
      'transform': `translateX(${currentIndex * -1 * 100}%)`
    }, transition);
    for(let i = currentIndex + 1; i < $views.size(); i++) {
      $viewsList.get(i).css({
        'transform': `translateX(${(i - 1) * -1 * 100}%)`
      });
    }
  }

  function scrollPrev(index = null) { ...
  }

  function setText() { ...
  }
}

```

Рисунок 2.43 – Листинг кода горизонтального слайдера (часть 1)

```

    }
    function setText() {
        $listPage.value(`${currentIndex + 1}/${$views.size()}`);
    }

    $js(`#prevBtn`).onClick(e => {
        if ($views.size() > 1) {
            scrollPrev();
            setText();
        }
    })
    $js(`#nextBtn`).onClick(e => {
        if ($views.size() > 1) {
            scrollNext();
            setText()
        }
    })

    $listPage.onClick((e) => {
        e.value('');
    })
    $listPage.onEvent('keydown', (e, event) => {
        log(event)
        if (event.keyCode == 13) {
            let num = parseInt(e.value()) - 1
            if (num > $views.size() || num < 0 || num == currentIndex)
                return setText();
            num > currentIndex ?
                scrollNext(num) :
                scrollPrev(num);
            setText()
            e.get().blur()
        }
    })
}

```

Рисунок 2.44 – Листинг кода горизонтального слайдера (часть 2)

3 ОПТИМИЗАЦИЯ ИНТЕРНЕТ-РЕСУРСА

3.1 Оптимизация клиентской части интернет-ресурса для основных браузеров

Сайт оптимизирован под все основные браузеры: Mozilla Firefox, Microsoft Edge, Google Chrome и Яндекс Браузер. Оптимизированы стили CSS и соответствующие теги HTML. Результаты проверки приведены на рисунках 3.1-3.4.

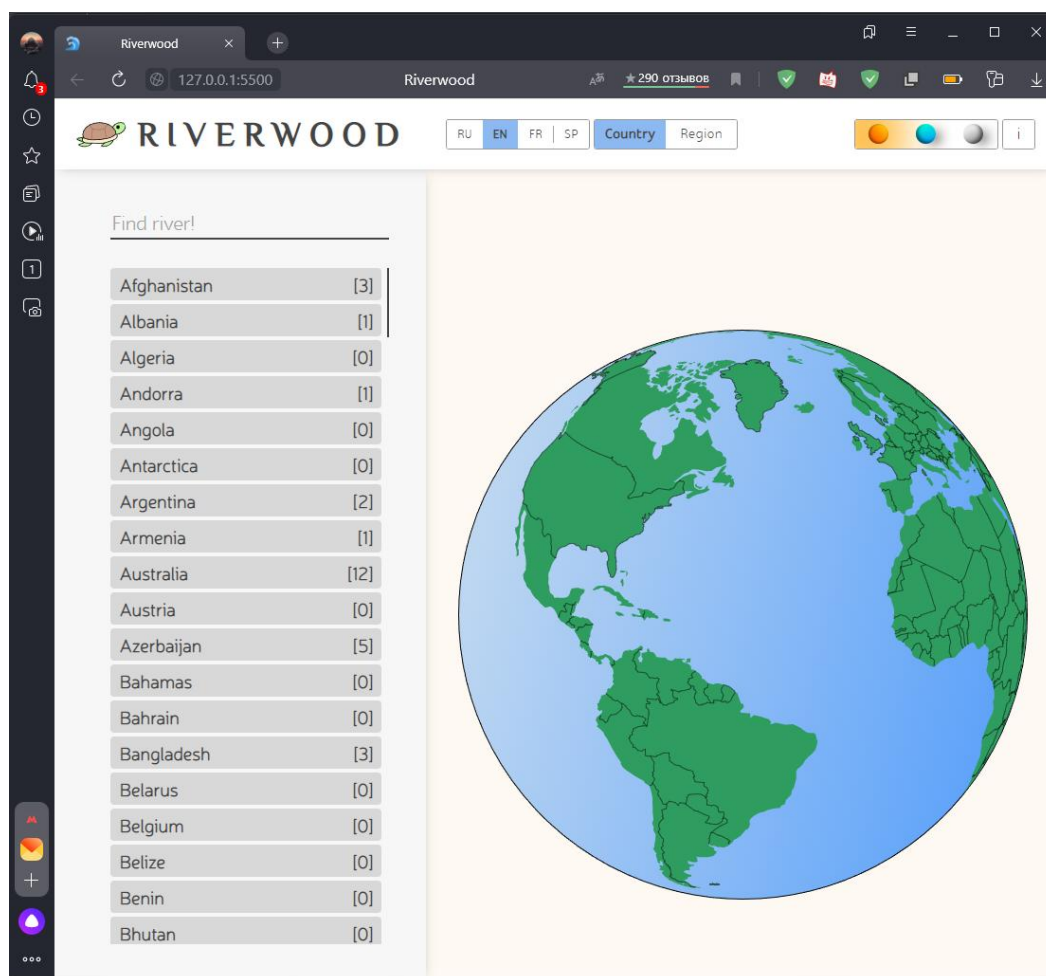


Рисунок 3.1 – Страница сайта в браузере Яндекс

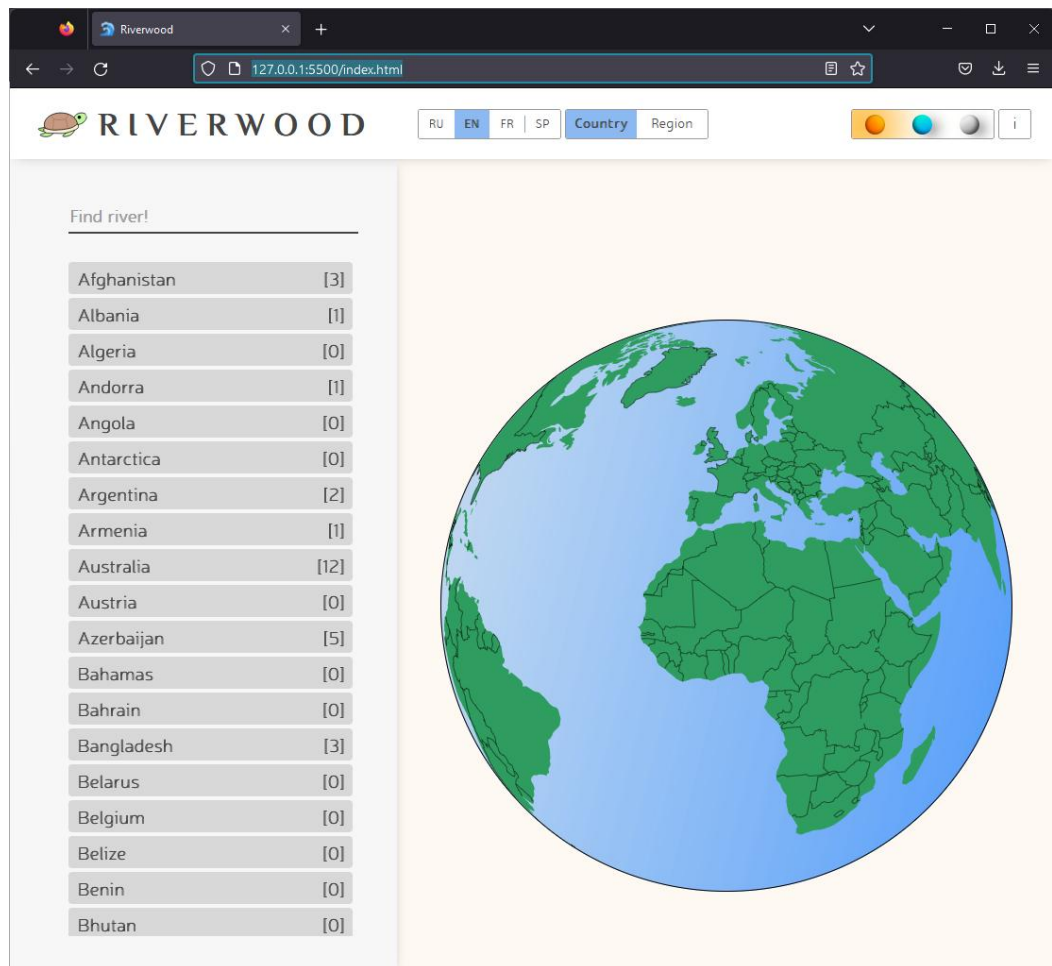


Рисунок 3.2 – Страница сайта в браузере Mozilla Firefox

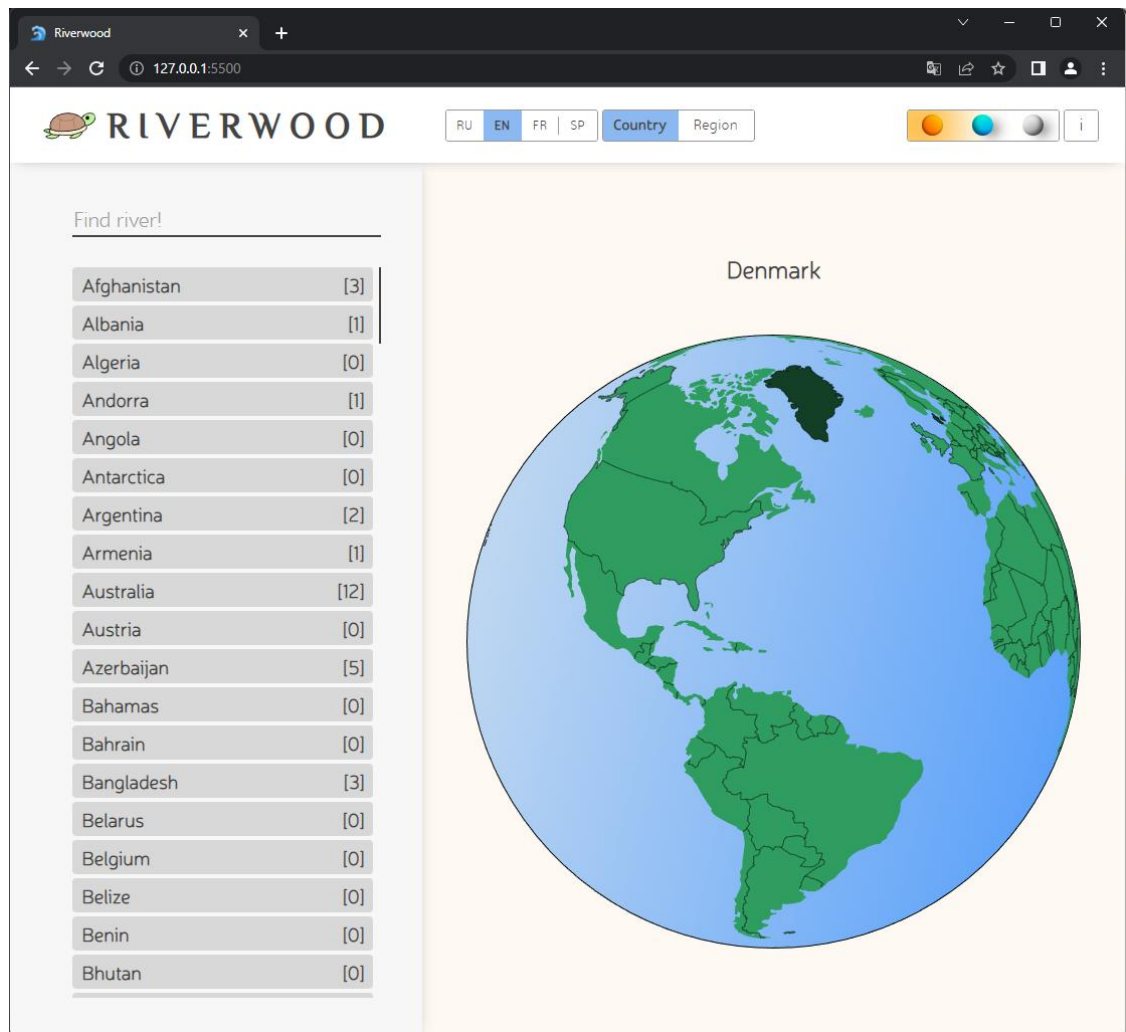


Рисунок 3.3 – Страница сайта в браузере Google Chrome

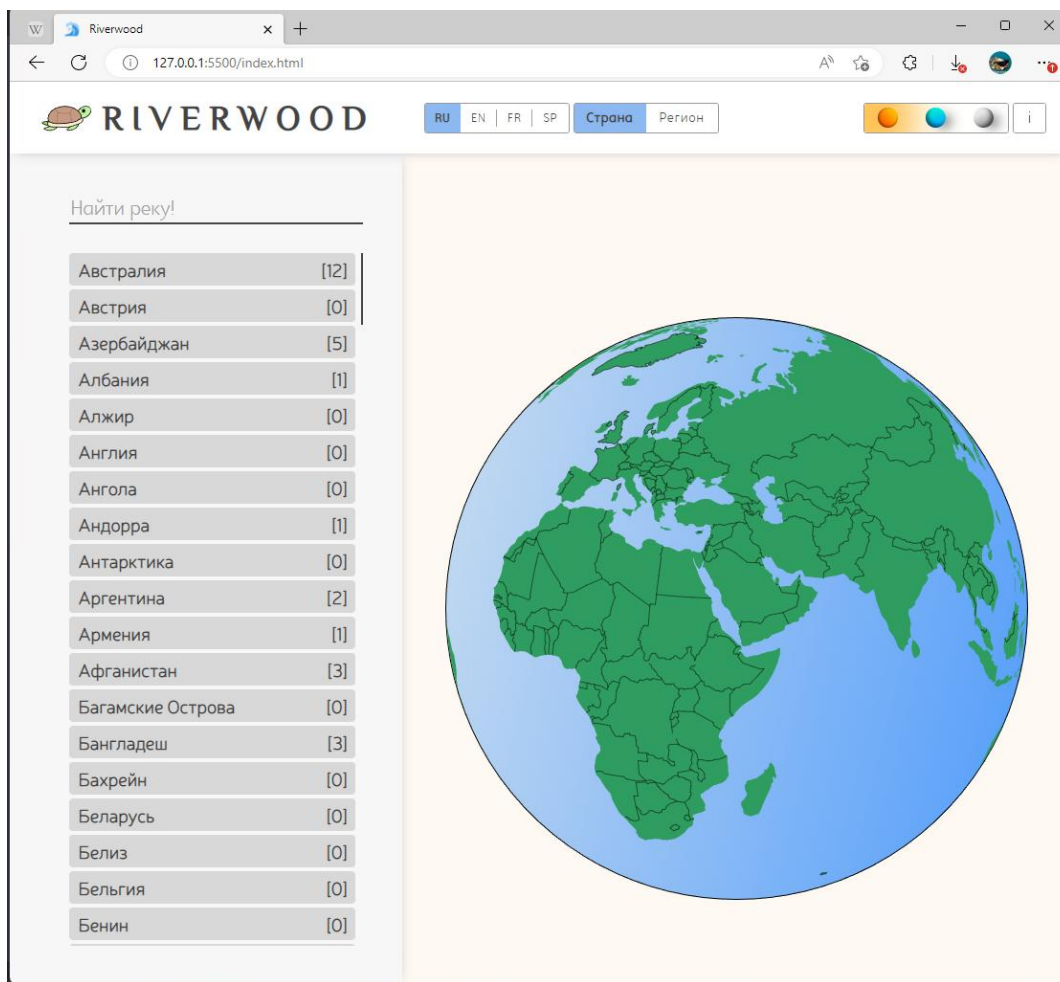


Рисунок 3.4 – Страница сайта в браузере Microsoft Edge

3.2 Оптимизация клиентской части интернет-ресурса для стационарных и мобильных устройств

Было реализовано четыре состояния для различной ширины экрана: первое (больше 920 пикселей,) второе (от 767 до 920 пикселей), третье (от 600 до 767 пикселей), четвертое (меньше 600 пикселей). Каждое состояние для экрана компьютера, планшета, телефона, телефона с особо узким экраном соответственно.

При уменьшении ширины экрана навигационная панель последовательно меняется три раза: первый – размеры кнопок и шрифт слегка уменьшаются, второй – картинка в логотипе уменьшается, элементы управления скрываются и вместо них появляется бургер-меню, третий – шрифт в логотипе уменьшается, изменения представлены на рисунке 3.5 – 3.8.



Рисунок 3.5 – Исходное состояние навигационной панели



Рисунок 3.6 – Второе состояние навигационной панели

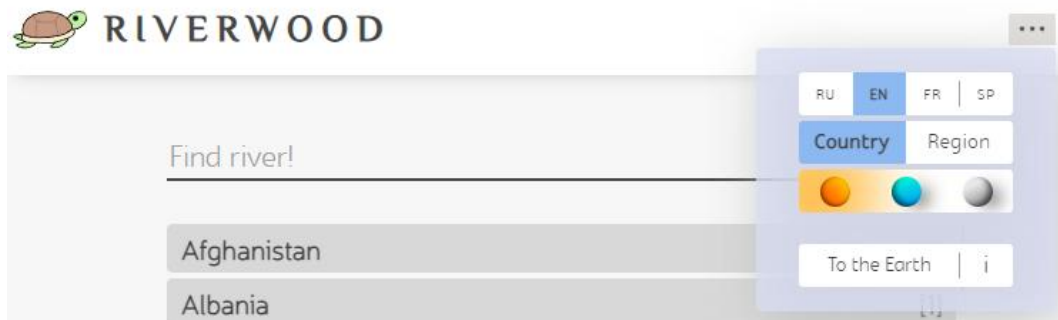


Рисунок 3.7 –Третье состояние навигационной панели



Рисунок 3.8 – Четвертое состояние навигационной панели

Основная страница имеет все два состояния (Рисунок 3.9 – 3.11). При переходе в состояние отображения на планшете и в дальнейшем список и глобус растягиваются на всю ширину, переключение между ними осуществляется через бургер-меню.

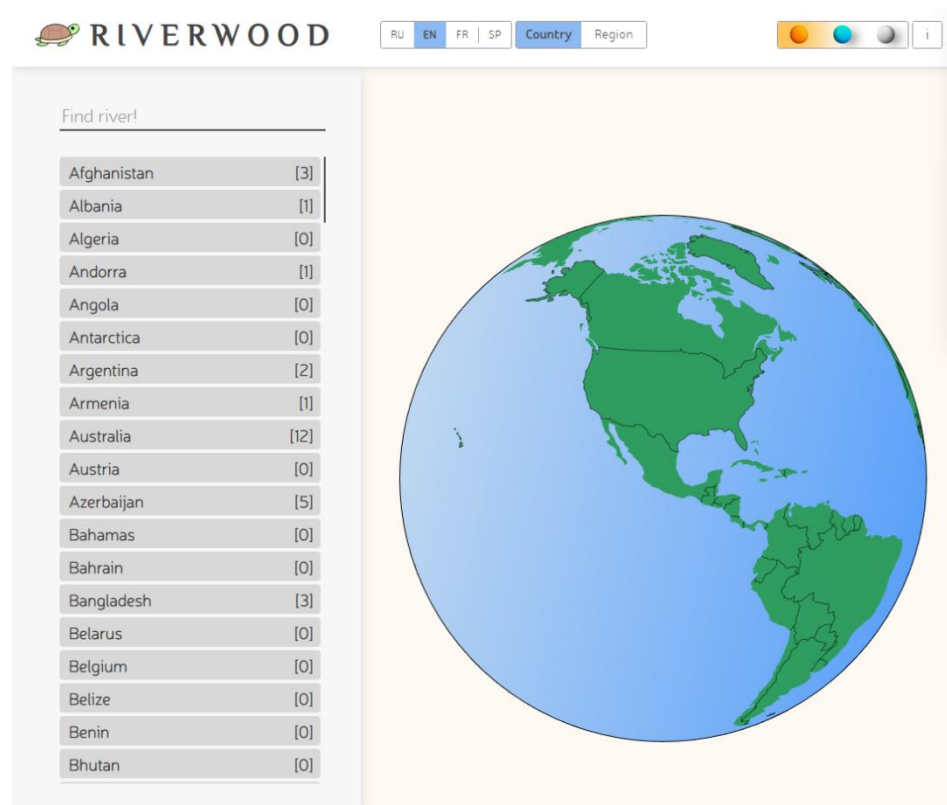


Рисунок 3.9 – Исходное состояние основной страницы



Find river!

Afghanistan	[3]
Albania	[1]
Algeria	[0]
Andorra	[1]
Angola	[0]
Antarctica	[0]
Argentina	[2]
Armenia	[1]
Australia	[12]
Austria	[0]
Azerbaijan	[5]
Bahamas	[0]
Bahrain	[0]
Bangladesh	[3]
Belarus	[0]
Belgium	[0]
Belize	[0]

Рисунок 3.10 – Второе состояние основной страницы (Вид списка)

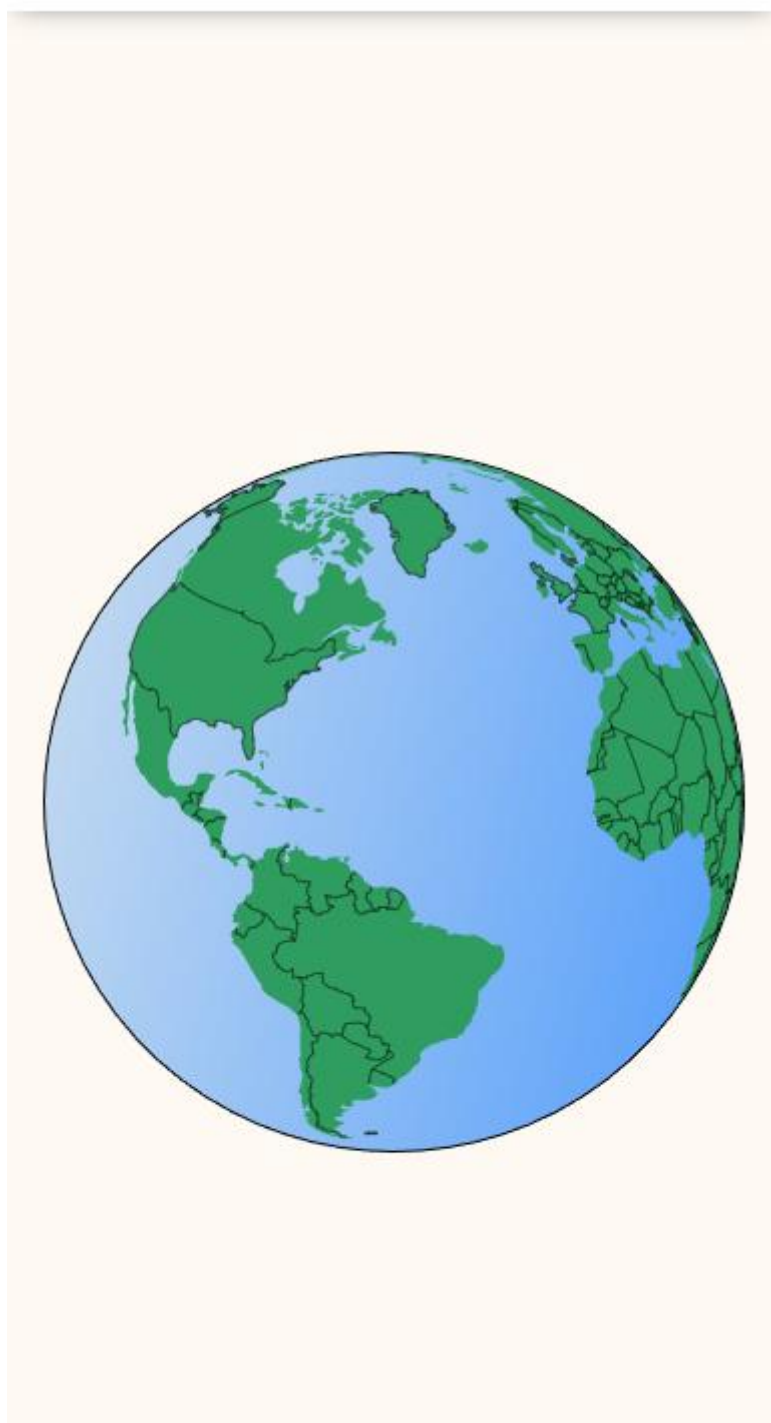


Рисунок 3.11 – Второе состояние основной страницы (Вид глобуса)

ЗАКЛЮЧЕНИЕ

В результате выполнения курсовой работы были выполнены все поставленные задачи, а именно: проведен анализ предметной области, были задействованы и использованы такие технологии, как HTML5, CSS3 и JavaScript.

Был создан интернет-ресурс по теме «Реки мира» для пользователей, интересующихся географией. Для продукта была проведена межстраничная навигация и проведена оптимизация веб-страниц для браузеров и различных видов устройств.

В ходе тестирования созданного продукта на разных устройствах и в разных браузерах, было выявлено, что продукт работает корректно.

Исходный код интернет-ресурса доступен по ссылке – https://github.com/khsdf404/Coursework_3-Frontend-.

Доменный доступ к интернет-ресурсу доступен по ссылке – <http://riverwood.h1n.ru/>.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. HtmlBook – Электронный самоучитель по HTML [Электронный ресурс]. – URL: <http://htmlbook.ru/> (дата обращения 28.10.2022).
2. CSS Tutorial – База знаний по CSS [Электронный ресурс]. – URL: <https://www.w3schools.com/css/> (дата обращения 28.10.2022).
3. Справочник по CSS [Электронный ресурс]. – URL: <https://developer.mozilla.org/ru/docs/Web/CSS/Reference> (дата обращения 28.10.2022).
4. Современный учебник JavaScript [Электронный ресурс]. – URL: <https://learn.javascript.ru/> (дата обращения 29.11.2022).
5. Справочник Javascript [Электронный ресурс]. – URL: <https://javascript.ru> (дата обращения 29.11.2022).
6. Категория: Реки мира по алфавиту – Википедия [Электронный ресурс]. – URL: https://ru.wikipedia.org/wiki/Категория:Реки_по_алфавиту (дата обращения 28.10.2022).
7. Крупнейшие реки и озера мира | Ифнокаталог [Электронный ресурс]. – URL: <https://infoselection.ru/infokatalog/obuchenie-i-znaniya/geografiya/item/798-krupnejshie-reki-i-ozera-mira> (дата обращения 28.10.2022).
8. 10 лучших IDE и редакторов кода для веб-разработчиков [Электронный ресурс]. – URL: <https://www.reg.ru/blog/10-luchshih-ide-i-redaktorov-koda-dlya-veb-razrabotchikov/> (дата обращения 28.10.2022).