МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение высшего образования

«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Экономики и управления на предприятии природопользования»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

(бакалаврская работа)
по направлению подготовки 09.03.03 Прикладная информатика
(квалификация – бакалавр)

На тему «Интерактивная карта Краснодарского края для самостоятельных путешествий»

Исполнитель Мельничук Кирилл Валерьевич

Руководитель к.т.н. Яготинцева Наталья Владимировна

«К защите допускаю»

Руководитель кафедры _____

кандидат экономических наук

Продолятченко Павел Алексеевич

«21» 01 2022 г.

Филиал Российского государственного гидрометеорологического университета в г. Туапсе
НОРМОКОНТРОЛЬ ПРОЙДЕН
«14» 01 2022 г.
Мартынова Т.В.
ПОДПИСЬ РАСШИФРОВКА ПОДПИСИ

Туапсе
2022

Оглавление

Введение ... 3
1 Интерактивные туристские карты ... 6
1.1 Общее представление о туристской карте, требования к ним и классификация ... 6
1.2 Интерактивная карта, её виды и требования к ним ... 12
2 Программные средства и технологии создания интерактивных карт ... 24
2.1 Обзор сервисов для создания интерактивных карт ... 24
2.2 Анализ технологий для разработки интерактивной карты ... 28
3 Проектирование и разработка интерактивной карты Краснодарского края (г. Сочи) для самостоятельных путешествий ... 37
3.1 Проектирование интерактивной карты Краснодарского края (г. Сочи) для самостоятельных путешествий ... 37
3.2 Описание процесса разработки интерактивной карты Краснодарского края (г. Сочи) для самостоятельных путешествий ... 43
Заключение ... 57
Список использованной литературы ... 59
Приложение ... 62

Введение

В настоящее время Интернет затронул практически все сферы человеческой жизни. Интернет может удовлетворить большинство потребностей современного человека: совершить покупку, послушать музыку, ознакомиться с работами художников или фотографов, заказать услуги, найти необходимую информацию и т.д. Процесс глобализации не оставил в стороне и такую сферу как туризм.

В настоящее время бумажные карты и атласы заменили электронные интерактивные, карты. Они размещены в глобальной сети и позволяют пользователю находить объекты в кратчайшее время. Так же в них, в отличие от обычных, реализован интерактивный диалог с пользователем, они реагируют на действия пользователя. Не все географические объекты представлены на них подробно. На некоторых из них представлены схемы движения транспорта, на других социально-значимые объекты, на следующих достопримечательности данного региона. Все они имеют свой функционал и как правило предоставляют очень полезную для пользователей информацию. Интерактивные карты стали неотъемлемой частью веб-сайтов. Веб-сайт-совокупность общедоступных взаимосвязанных веб-страниц, которые используют одно доменное имя. Веб-сайты могут создаваться и поддерживаться отдельным лицом, группой, бизнесом или организацией для различных целей. Вместе все общедоступные веб-сайты составляют всемирную паутину.

В данной работе суть разработки интерактивной карты, заключается в предоставлении информации об достопримечательностях Краснодарского края для самостоятельных путешествий на примере города Сочи.

В Краснодарском крае находится множество курортов с хорошо развитой инфраструктурой. Здесь есть санатории, отели, детские лагеря, рестораны и кафе, а также разнообразные достопримечательности. Курорты края ежегодно

принимают около 17 млн туристов. Уже несколько лет неизменным лидером является город Сочи. По данным предоставленным турстатом [3] Большой Сочи по годовому количеству туристов и объемам бронирований путевок у турагентств занимает первое место. Инфраструктура города после Зимних Олимпийских игр 2014 города вышла на новый уровень и ничуть не уступает зарубежным курортам. Все больше и больше жителей нашей страны планируют свой отдых в этом чудесном городе. Не смотря на высокую популярность Большого Сочи, в глобальной сети интернет недостаточно актуальной и полезной информации в виде интерактивных карт. Практически отсутствуют карты с точным местоположением достопримечательностей, что определило актуальность вопроса исследования.

Объект исследования – достопримечательности Краснодарского края для самостоятельного посещения туристами.

Предметом исследования являются интерактивные карты на базе открытых продуктов.

Цель дипломной работы - разработать интерактивную карту Краснодарского края для самостоятельных путешествий.

Для этого необходимо решить следующие задачи:

- изучение теоретических материалов по классификации интерактивных карт;
- анализ предметной области и предъявляемых требований к интерактивным картам;
- анализ и выбор средств и технологий для разработки;
- освоение технологий разработки в среде Visual Studio Code;
- анализ объектов посещения туристов на территории курорта и выделение наиболее значимых;
- проектирование дизайна интерактивной карты;
- тестирование готового программного решения с выявлением и устранением ошибок;
- провести расчет эффективности разрабатываемого проекта.

Дипломная работа состоит из введения, трех глав, заключения и списка использованной литературы.

Первая глава раскрывает основные понятия и виды интерактивных карт, их классификации; определяет требования для создания интерактивных карт.

Вторая глава содержит изучение технологии создания интерактивных карт и обоснование выбора средств, и технологий для разработки интерактивных карт

В третьей главе представлен процесс проектирования интерактивной карты, расположенной на web-странице.

Источники информации для решения задач включают в себя данные преддипломной практики. В приложении представлены программный код и изображения, являющиеся основанием для разработки.

Практическая часть представляет собой интерактивную карту Краснодарского края для самостоятельных путешествий.

Практическая ценность работы заключается в применении разработанного программного решения в сфере туризма и закреплении практических навыков, полученных в процессе обучения.

1 Интерактивные туристские карты

1.1 Общее представление о туристской карте, требования к ним и классификация

Люди из древне предпринимали попытки определения и фиксирования своего местоположения в наскальной живописи, древних картах, папирусах с помощью визуальных элементов. На картах отображаются не только природные, но и социально-культурные объекты, созданные руками человека, устанавливается их уровень взаимосвязанности [5, с. 32-36].

В развитии туристско-рекреационной картографии выделяют два этапа.

Первый этап связан с появлением карт, с тематикой близкой к туристской. На них отражались дороги и населенные пункты, а также некоторые элементы современных карт.

Второй этап (середина XIX века). В это время путешествия перестают быть эксклюзивными, появляется потребность в создании туристских карт. Современными предшественниками туристских карт это планы городов, перспективные панорамы и первые карты путеводители. С 30-х гг. XX века, карты, созданные специально для туристов, получили название «туристские карты». Именно в это время и наблюдается всплеск развития туристского картографирования.

Туристская карта – это географическая карта, на которой представлены объекты, наиболее интересные для туризма, как правило сопровождаемые текстовым описанием, а также различными фотографиями и рисунками.

Туристская карта представляет собой массовый продукт, имеющий прикладное значение, и который можно использовать для планирования путешествий.

В связи с эти, данные карты должны соответствовать требованиям:

- отражать проверенную и актуальную информацию;
- иметь высокую степень детализации и высокий уровень

репрезентативности;

- содержать достоверную систему условных обозначений: иконок, значков, пиктограмм и т.д.;

- иметь высокую степень практичности и удобства использования.

На сегодняшний день туристские карты разделяются на два типа: «для туризма» – это карты, предназначенные для туристов и для путешествий это карты «о туризме», в том числе к ним относятся научные рекреационные карты, созданные для исследования.

Карты «для туризма»

Основная цель создания карт «для туризма» - это знакомство туриста с интересующей его местностью. На таких картах представлены культурно-исторические объекты, данные о флоре и фауне региона. Один из главных критериев таких карт является полное и достоверное отражение объектов туристской инфраструктуры (рисунок 1). Большое распространение нашли специализированные карты. Они разрабатываются специально для любителей конкретного вида отдыха или для увеличения популярности конкретного маршрута.

Рисунок 1 – Карта «30 уникальных мест России» [2]

Чаще всего туристские карты отражают спектр взаимосвязанных объектов и явлений, т. е. несут аналитический и комплексный характер. В большинстве своем эти карты – инвентаризационные, т.е. на них представлена

точная объективная информация об объектах, реже – оценочная.

Другая категория карт «о туризме» важна при научном изучении туристской отрасли и отражает наиболее точную информацию для исследователей и организаторов туризма. В отличие от первой категории карт, данные карты более сложные и более разнообразны по содержанию. Основная задача составления этих карт - это обобщение результатов исследований в области географии досуговой деятельности (рисунок 2). В связи с эти на картах представлены отдельные характеристики, позволяющие представить общую картину состояния различных отраслей туристской индустрии [17, С. 11].

Для их создания проводится большая подготовительная работа, такая как: сбор данных, анализ большого количества источников, оценка репрезентативности полученных данных и возможности использования их в работе. Основные материалы для создания это: статистические данные, рейтинги и т.д.

Субъективными являются методы сбора информации и ее интерпретации, в связи с эти тематические карты представлены у разных авторов могут по-разному. Неверная интерпретация собранных автором данных может привести к необратимым последствиям при их практическом использовании. Отмеченные выше факты указывают на необходимость тщательного поиска и обоснования методов, а также критериев использования объектов, явлений и характеристик.

Рисунок 2 – Карта оценки природных условий для развитий пешеходного туризма [4]

Карты о туризме можно классифицировать по степени обобщения данных и по практическому назначению. Они делятся на аналитические, комплексные, инвентаризационные (карты центров культурно-познавательного туризма), синтетические (туристско-рекреационное районирование), оценочные (оценка климатических условий для конкретного вида отдыха) и прогнозные (отражают потенциал территории и перспективы ее развития).

Систематизация туристских карт

Применение туристских карт рассчитано на практическое массовое повсеместное использование. Они отличаются тематикой и специфическое отображение пространства. Основная цель туристской карты доступность к пониманию и удобство для использования туристам. В первую очередь она должна помочь не заблудиться человеку в незнакомом для него месте, найти интересующий его объект и определить путь, который приведет его к нему.

На данных картах нет необходимости отображать какую-либо информацию об экономической и производственной составляющей данной территории, так как эта информация будет считаться избыточной.

При узкой специализации этого картографического продукта, при их составлении должна присутствовать стандартная для большинства карт основа. Основные составляющие карт: подробное описание неровностей местности, отображение водных объектов, ареалы распространения различных видов растительности, указание урбанистической и сельской застройки, границы административных территорий. Также можно выделить карты с указанными маршрутами различной сложности, предназначенные для любителей пеших прогулок.

Главным изображением туристских карт являются достопримечательности. Это могут быть как исторические и культурные памятники, национальные парки и заповедники, объекты, относящиеся к духовной сфере обогащения человека, творческой или даже спортивной. Эти

объекты представлены в путеводителях, сопровождающих туриста в поездках, а в повседневной жизни эти они как правило являются обыденной и привычной частью жизни местного населения.

Одним из ярких примеров является стадион «Фишт», после проведения Зимних Олимпийских игр 2014, он переквалифицирован в спортивный комплекс, предназначенный для общего доступа, на его территории проводятся спортивные тренировки, и он часто становится площадкой для проведения соревнований. Также можно выделить объекты, которые первоначально были построенные для привлечения туристов и нацелены обеспечить их пребывания в комфортных условиях. Примером этого является целая страна – Объединенные Арабские Эмираты. Часть инфраструктуры была создана только для привлечения большого количества иностранных туристов. Но эти блага повысили уровень и комфорт жизни местного населения.

Спецификой туристских карт является интересная и красочная компоновка. При их составлении могут использоваться различные современные дизайнерские приемы, а также применятся 3D-модели определенных зданий и сооружений (рисунок 3). Существуют возможности создания карт с применением художественных изображений или фотографий отдельных объектов. В современных интернет-картах наблюдаются панорамные снимки объектов. Такие изображения являются очень подробными и помогают туристу быстро сориентироваться в пространстве.

Рисунок 3 – Карта достопримечательностей Санкт-Петербурга [15]

Туристские карты могут отличаться по разным параметрам: масштабу, площади отображенной территории, наполнению и назначению. Исходя из стандартной системы разделения географических карт по масштабу, туристские карты делятся на: крупномасштабные (1:200000 и более); среднемасштабные (от 1:200000 и до 1:1000000); мелкомасштабные (менее 1:1000000).

По площади охватываемой территории карты делятся на: мировые, материковые, общегосударственные, региональные и локальные.

По тематике: карты с изображением туристских объектов, элементов туристской инфраструктуры, туристских маршрутов.

По назначению карты имеют справочный, обзорный, экскурсионный, познавательный характер.

Наиболее полная информацию о классификации туристских карт [19] представлена на схеме (рисунок 4).

Рисунок 4 – Разновидности туристских карт

Наиболее популярна общая классификация. Обзорные карты, охватывают отдельные территориально рекреационные комплексы, большие туристские районы, а также множество элементов административно-территориального устройства страны. В их основе лежат общегеографические изображения земной поверхности, на которых представлены элементы рельефа, гидрографии, шкалы высот, транспортных путей, населенных пунктов и т.п. Они в свою очередь дополняются различными туристскими объектами, такими как исторические или архитектурные памятники, учреждения размещения и питания, и иными элементами проявления туристской инфраструктуры.

На маршрутных картосхемах представлены территории, пролегающие вдоль туристских маршрутов это могут быть пешеходные, автомобильные, велосипедные, водные, подземные и т.п. Маршрутные картосхемы включают в себя подробную информацию и анализ самого маршрута, это и протяженность и конфигурация, перепад высот и глубин, а они содержат множество информации о расположении на маршруте туристских достопримечательностей, учреждений обслуживания, так же в нее входят необходимые туристу кемпинги, гостиницы, заведения питания, АЗС, СТО и т.п.

С появлением Web-технологии компьютер начинают использовать все большее количество людей. Быстрые темпы развитие информационных технологий затронули так же и область туристских карт. Туристские карты, представленные в электронном виде в большинстве своем, являются интерактивными.

1.2 Интерактивная карта, её виды и требования к ним

Термин «интерактивность» берет начало от английского слова interaction, и означает в переводе «взаимодействие». Одним из участников взаимодействия в процессах, связанных с проявлением информационных технологий выступает человек (пользователь).

Одним из важных показателей, характеризующих быстроту и удобство с помощью которого пользователь может добиться своей цели является степень интерактивности.

Уровни интерактивности, с точки зрения степени взаимодействия:

– линейное взаимодействие, или отсутствие интерактивности, в тех случаях, когда посылаемое сообщение не связано с предыдущими сообщениями;

– реактивное взаимодействие, посылаемое сообщение связано с предыдущим сообщением;

– множественное или диалоговое взаимодействие, сообщение связано с множеством предыдущих сообщений и с отношениями между ними.

Элементы интерактивности - это все программные модули (функции сайта), при помощи которых человек (пользователь) взаимодействует с сайтом или другим человеком, посредством этих инструментов.

На данный момент элементы интерактивности включают также в электронные карты. Крупные геопорталы предоставляют полную информацию о различных территориях по всему миру. По сути такая информация является обзорной. При необходимости более подробной информации пользователь продолжает поиск на каких-то других информационных порталах. При стремительно развивающихся информационных коммуникациях и сети Интернет, исчезает необходимость постоянного наличия копий карт, потому что пользователь при условии наличия доступа к сети всегда может обратиться к различными сетевыми ресурсами. На электронных интерактивных картах существует возможность рассмотрения мелких участков территории, используя углубленный подход к информационной составляющей описываемых объектов. Данный подход позволяет пользователю получить все нужные ему данные, не применяя дальнейших поисков.

Интерактивная карта - это электронная карта, которая осуществляет работу в режиме двухстороннего взаимодействия человека с электронным носителем. Кроме информации, которую пользователь получает при чтении

карты, на ней представлена скрытая информацией, получить которую можно при выполнении определенных действий.

На интерактивных картах существует возможность поиска любых объектов за несколько секунд. Она позволяет выводить информацию по мере надобности пользователя. Отличием от простых статических карт является то что, у каждого условного знака на карте по мимо его обычной информационной составляющей, есть скрытая, которая выводится по мере надобности. Данный подход дает возможность не перегружать карту условными знаками, позволяет представить ее более понятной и легко читаемой.

Интерактивные карты обладают наглядность и многофункциональностью. Они позволяют не только осуществить просмотром местоположения. Выделяют интерактивные flash-карты, классические карты и карты с визуализацией, являющиеся наиболее интересными и набирающими популярность [14].

Развитие интерактивных карт относится ко второй половине 90-х годов. Интерактивные карты имеют так же возможность демонстрировать процессы и явления в динамике.

Разработанная в картографии классификация для статических карт актуальна и для интерактивных карт. Карты можно разделить по масштабу (планы, крупномасштабные, среднемасштабные и мелкомасштабные), по охвату территории (карты Солнечной системы, планеты, материков, стран, областей, населенных пунктов и т. д.), по содержанию выделяют общегеографические (топографические, обзорные, обзорно-топографические), тематические (политические, социальные, экономические и многие другие), специальные (навигационные, кадастровые, технические, проектные и др.) [2, с. 12 - 18]. Все чаще и чаще интерактивные карты заменяют собой статические карты. Ниже приведена классификация интерактивных карт по сфере применения.

- Интерактивные образовательные карты применятся в образовательном процессе, например, при изучении географии, истории и других дисциплин

образовательных учреждениях или при получении самообразования. Например, интерактивная историческая карта (рисунок 5), с ее помощью можно просмотреть, сколько было заселено мировой территории, какие государства существовали и как менялись, начиная с 3000 года до нашей эры до современности. Для получения этих сведений пользователь лишь должен ввести год из этого промежутка и получить результат.

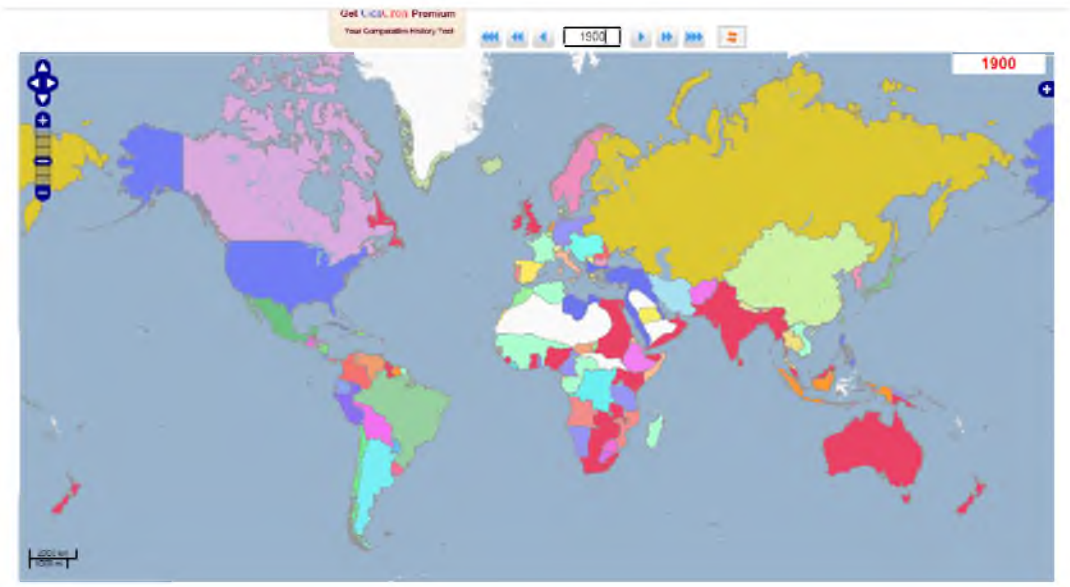


Рисунок 5 - Интерактивная мировая историческая карта с 3000 до н. э. [13]

- Интерактивные навигационные карты получили широкое применение. Они оснащаются все большими возможностями. Их используют для нахождения нужного объекта, построения маршрута, поиска ближайших необходимых объектов, прогулки по улицам или помещениям, отображения справочной информации и многого другого. Примеры таких карт являются карты транспорта, карты расположения АЗС, магазинов, туристических достопримечательностей и памятников архитектуры, карты погоды, навигационные карты в торговых центрах, офисах, музеях и т.д.

- Интерактивные развлекательные карты не обучают пользователя и не помогают ему с навигацией, они его занимают и веселят. Эти карты применяются, например, для привлечения большего числа посетителей на сайт или рекламы какого-то ресурса. Примером такой развлекательной интерактивной карты является карта Space Station Earth (Космическая станция

«Земля»), созданная Элеонорой Лутц и отсылающая к научно-фантастическим мирам. Созданные в космической стилистике города вместо морей и океанов окружены «галактическим океаном», а названия всех крупных городов заменены на космические колонии [30].

При классификации интерактивных карт актуально применение уже устоявшихся в науке оснований (масштаб, охват, содержание и др.), а также нужно учитывать специфику и возможности интерактивности. Интерактивные карты можно разделить по степени интерактивности на карты с: низким, средним и высоким уровнем интерактивности.

Карта низкого уровня интерактивности дает возможность пользователю сориентироваться в информации и выбирать контент, используя внутренние ссылки, масштабирование, клики для отображения деталей, при этом сохраняя содержимое без изменения. Данный способ взаимодействия пользователя с интерактивной картой является наиболее простым.

При среднем уровне интерактивности пользователю доступны инструменты управления (временная шкала, меню), которые позволяют незначительно изменять и сравнивать информацию.

При высоком уровне интерактивности пользователь может изучать инфографику и взаимодействовать с информацией путем ввода, фильтрации или поиска данных.

Примеры сайтов с различной степенью интерактивности представлены в приложении 1.

Интерактивная карта представляет собой программно-аппаратную информационную систему, работающую в режиме двухстороннего взаимодействия пользователя и компьютера при максимальном использовании визуальной составляющей. Это дает возможность рассмотреть территорию с нужной степенью получения данных об описываемом объекте или явлении. То есть каждый пользователь применяет нужную ему глубину поиска, изменяя объем поисковой выборки по мере необходимости [20, с. 16].

Мультимедиа – это интерактивная технология, обеспечивающая работу

со статическими изображениями, анимированной компьютерной графикой, видеоизображением, текстом и звуком.

С внедрением новых технологий увеличились возможности и инструментарий проектирования, создания и интеграции мультимедийных элементов, используемых при разработке карт. С применением мультимедийных технологий и средств поменялась содержательная часть карт, произошли модификации в используемых системах условных знаков, увеличилась информативность, появились новые особенности восприятия картографического изображения [9, с. 95 - 98].

Интерактивные карты с мультимедийной информацией дают возможность менять не только масштаб и положение видимой части карты, но и период времени, который охватывает отображаемую информацию.

При создании карты учитывается, то что карта не должна быть перегруженной ненужной информацией, и при этом содержать полную информацию об объектах и явлениях. При создании интерактивных карт, с помощью мультимедийных технологий таких противоречий не возникает. При гиперссылочном принципе просмотре карты пользователь имеет возможность двигаться вглубь от общего к частному по определенному поиску и/или выбору информации и получать дополнительную информацию из других Интернет-ресурсов.

Интерактивные карты отображаются на экранах электронных устройств, качество и размер которых отличаются. При разработке интерактивной карты следует учитывать особенности способа ее использования (как в целом, так и отдельных функций) на определенной программно-аппаратной платформе.

Туристская интерактивная карта с мультимедийной информацией дает возможность пользователю осуществлять поисковые операции об информации о территории, которая сопровождается графическими, текстовыми составляющими, аудио- и видеоинформацией, позволяет управлять легендой, выключать и подключать слои, например, показывать все памятники природы, скрывая другие объекты, масштабировать и передвигать изображение.

Графическая информация в мультимедийном картографическом произведении может быть представлена в следующих видах: растровые изображения, векторные изображения, трехмерные модели, анимации.

Эффекты анимации, применяемые в мультимедийных интерактивных картографических произведениях, нами были разделены на относящиеся к элементам интерфейса, карте в целом и отдельным объектам карты или их совокупностям (рисунок 6).

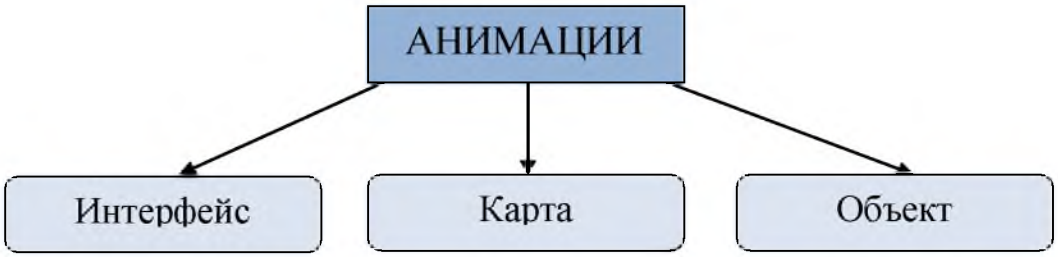


Рисунок 6 - Схема категорий картографических анимаций

Использование эффектов анимации в интерфейсе мультимедийных картографических произведений наиболее уместно в следующих случаях:

- при пояснении идеи или концепции элементов интерфейса;
- при изменении состояния интерфейса;
- при организации обратной связи с пользователем.

Анимации, применяемые в интерфейсе интерактивных приложений, можно разделить в соответствии с тремя группами визуальных компонентов интерфейса (рисунок 7).

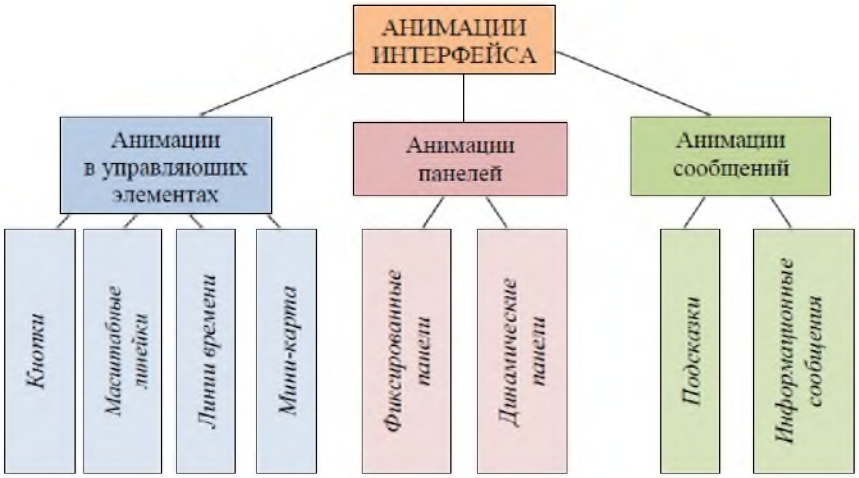


Рисунок 7 - Категории анимаций в интерфейсе

Анимации в интерфейсе картографических приложений могут быть применены для следующих типовых элементов: кнопки, масштабные линейки, мини-карта, информационные сообщения.

Эффекты анимации для карты

Анимации, применяемые к карте, как правило, связаны с изменением ее отображения как самим пользователем, так и автоматически, в результате каких-либо внутренних факторов, например, изменений в семантических параметрах объектов карты, и позволяют лучше сориентироваться в произошедших изменениях и местоположении отображаемого в данный момент участка местности, а также улучшить визуальное восприятие.

Анимации, применяемые для карты, можно разделить на следующие категории в соответствии с изменяемым в процессе анимирования параметром (рисунок 8).

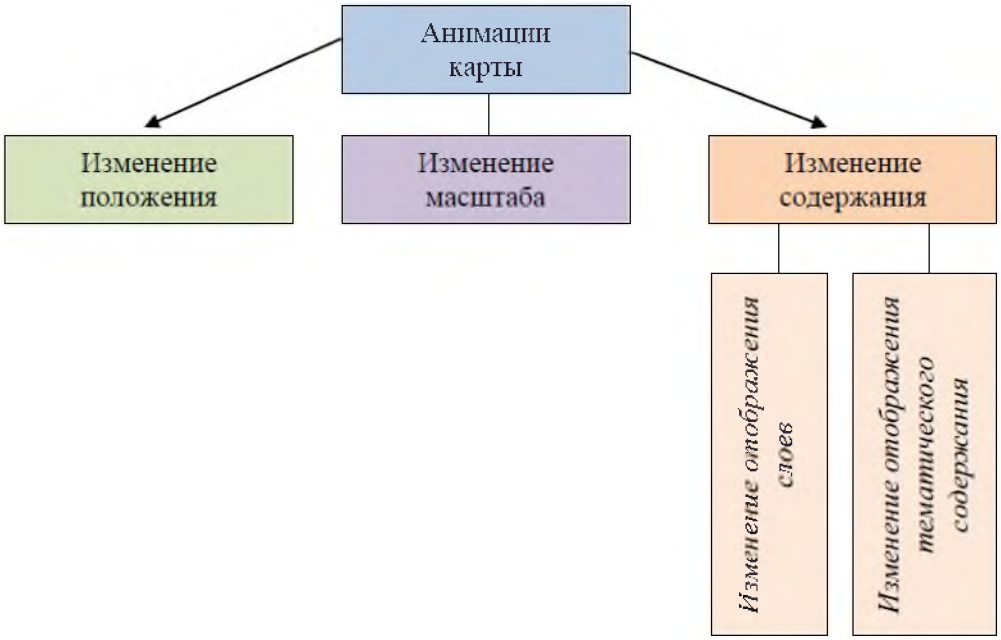
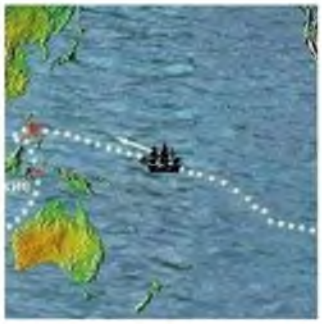


Рисунок 8 - Категории эффектов анимации карты в целом

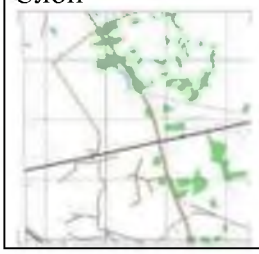
Примерами анимаций с картой могут служить плавное изменение масштаба, перемещение окна просмотра.

Систематизация видов анимационных эффектов, применяемыми к карте представлены в таблице 1.

Таблица 1 - Виды эффектов анимаций для карты

<table>
<tr><th>Название</th><th>Назначение</th><th>Параметры</th></tr>
<tr><td>Масштабирование
  </td><td>Реализация эффекта плавного увеличения/уменьшения масштабаотображения.</td><td>Устанавливаемые при создании:
Продолжительность анимации
Изменяемые пользователем:
Начальный масштаб
Конечный масштаб</td></tr>
<tr><td colspan="3">При функционировании данного эффекта используются некоторые принципы генерализации картографических изображений. Это реализуется с помощью установки определенных параметров в слоях карты:
– отображение – определяет, в каких масштабных границах объекты отображаются на карте;
– фильтрация узлов объекта – определяет степень детальности отображения геометрии линейных и площадных объектов;
– изменение цвета – в мелкомасштабных картах возможно использовать более контрастные наборы цветов для лучшей читаемости картографической информации;
– изменение текстуры – изменение текстуры объектов при различных масштабах (сплошная заливка цветом – схематичное отображение – фотография);
– изменение детализации – управление отображением отдельных элементов площадных объектов (границы видимости в зависимости от масштаба или разные варианты отображения для групп масштабов)</td></tr>
<tr><td>Перемещение
  </td><td>Реализация эффекта плавного линейного перемещения окна просмотра</td><td>Устанавливаемые при создании:
Продолжительность анимации
Изменяемые пользователем:
Исходное положение
Требуемое положение</td></tr>
<tr><td colspan="3">В данной анимации реализовано изменение скорости воспроизведения в зависимости от расстояния на карте между точками начала и окончания перемещения. Также существует возможность включить плавное изменение скорости воспроизведения анимации при начале и окончании перемещения для лучшей наглядности</td></tr>
<tr><td>Перемещение по траектории
  </td><td>Реализация эффекта плавного перемещения окна просмотра по сложной траектории (существующий объект, ломаная линия, сглаживающийсплайн)</td><td>Устанавливаемые при создании:
Исходное положение
Требуемое положение
Продолжительность анимации
Траектория
Изменяемые пользователем:
Исходное положение
Требуемое положение
Продолжительность анимации
Траектория</td></tr>
</table>

Продолжение Таблицы 1

В данной анимации реализовано изменение скорости воспроизведения в зависимости от расстояния на карте между ключевыми точками перемещения. Также существует возможность включить плавное изменение скорости воспроизведения анимации при начале, окончании перемещения и в ключевых точках для лучшей наглядности		
Тематические карты 	Реализация эффектов анимации для изменения внешнего вида объектов (цвет, заливка) и построения дополнительных элементов тематических карт (диаграммы, таблицы)	Устанавливаемые при создании: Тип анимации Продолжительность анимации
В данной анимации реализовано последовательное отображение элементов тематической карты в зависимости от показателей (по градациям увеличения параметра). Также реализовано подсвечивание однотипных элементов при наведении		
Слои 	Реализация эффектов анимации для изменения отображения отдельных слоев или их групп	Устанавливаемые при создании: Тип анимации Продолжительность анимации

Параметры анимационных эффектов карты приведены в сводной таблице, представленной в приложении 2. Они устанавливаются при разработке интерактивной карты и могут быть изменены при модификациях информации.

Эффекты анимации для объектов карты

Анимации объектов карты, реализуемые в разрабатываемом объекте, подразделяются на следующие категории – изменение положения, изменение масштаба, изменение внешнего вида, изменение прозрачности (рисунок 9).

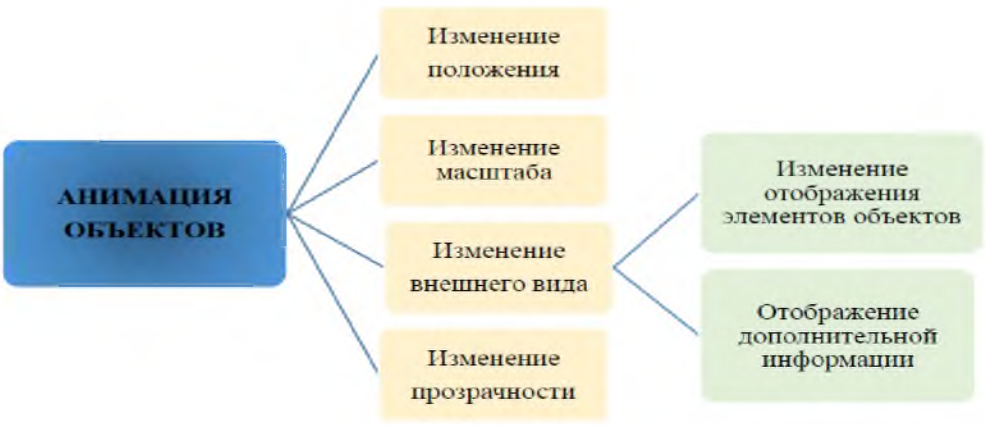
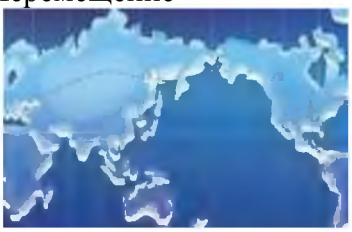


Рисунок 9 - Категории эффектов анимации для объектов карты

В таблице 2 представлены виды анимационных эффектов картографического приложения, применяемые к отдельным объектам карты.

Таблица 2 - Виды эффектов анимации для объектов карты

Название	Назначение	Параметры
Масштабирование 	Реализация эффекта плавного увеличения/уменьшения масштаба отображения объекта карты	Устанавливаемые при создании: Исходный масштаб Требуемый масштаб Продолжительность анимации
Перемещение 	Реализация эффекта плавного линейного перемещения объекта карты по простой (линия, ломаная линия) или сложной (сглаживающий сплайн, существующий объект) траектории с произвольным изменением скорости перемещения	Устанавливаемые при создании: Исходное положение Требуемый масштаб Продолжительность анимации Траектория
Изменение элементов 	Реализации эффекта анимации, в процессе выполнения которой у объекта или группы объектов отображаются или исчезают (генерализация) отдельные элементы	Устанавливаемые при создании: Исходное состояние Требуемое состояние Продолжительность анимации Описание промежуточных состояний
Отображение дополнительной информации 	Реализация эффектов анимации для отображения произвольной дополнительной информации об объекте	Устанавливаемые при создании: Тип дополнительной информации Тип анимации Продолжительность анимации Изменяемые ользователем: Исходные параметры
Изменение прозрачности 	Реализация эффектов анимации для изменения отображения объекта	Устанавливаемые при создании: Тип анимации Продолжительность анимации

Изменяемые параметры описанных выше эффектов анимации объектов карты приведены в таблице - Сводная таблица параметров

эффектов анимации для объектов карты, представленной в приложении 2.

Туристская интерактивная карта с мультимедийной информацией должна соответствовать следующим требованиям:

1) наглядное представление картографической и справочной информации;

2) интуитивно понятные способы ввода новой информации;

3) удобные и многофункциональные инструменты поиска и фильтрации информации;

4) возможность интеграции пространственных и справочных данных из иных программных продуктов и сервисов;

5) наличие инструментов онлайн-редактирования картографических данных;

6) отображение картографической информации и возможности поиска в приложениях на всех распространенных мобильных платформах.

2 Программные средства и технологии создания интерактивных карт

2.1 Обзор сервисов для создания интерактивных карт

Интерактивная карта является картографическим изображением, дополненным мультимедийными элементами, привязанными к определенным объектам карты, и реализованное с помощью программных и технических средств в системе с различными приемами и методами традиционной картографии и мультимедиа. На такой карте картографическое изображение и мультимедийные элементы интегрированы в единой информационной среде.

Интерактивная карта работает в двустороннем режиме, то есть человек может задействовать какие-то ее функции и передать какие-то данные. На нах представлена не только открытая информация, которую пользователи видят с первых секунд пользования, но и закрытая, которая проявляется только после наведения курсора на какой-то объект или клика по нему [21, с. 85 – 91].

Все современные электронные карты можно разделить на следующие подгруппы:

1. Неинтерактивные, зависящие от программного обеспечения.
2. Интерактивные, зависящие от программного обеспечения.
3. Интеррактивные, не зависящие от ПО.

Неитерактивные, программно-зависящими картами называются являются те, которые создаются с помощью следующего программного обеспечения, например, Macromedia Freehand, Adobe Illustrator, CorelDRAW и другие. Чтобы использовать такую карту, необходимо иметь компьютер с установленной операционной системой и софт, в котором создавалась карта (или же программную среду, которая поддерживает формат созданной карты).

Интерактивные програмно-зависимые карты появляются на свет с помощью такого софта, как Нева, Microstation, Panorama, MapInfo. Такая карта в готовом виде выглядит как один или несколько файлов. Для того, чтобы открыть такую карту, нужна либо иметь ту программу, в которой она создавалась, либо программу, поддерживающую формат карты.

Интерактивные, програмно-независимые карты создаются в программах MapGps, Curious World Maps или Gisware и так далее. В двух предыдущих случаях нужно иметь специфическое программное обеспечение, а в этом случае для просмотра карты нужен только компьютер с операционной системой и как браузером, т. к. интерактивная карта расположена на Web-странице. Функция визуализация в данном случае внедряется в карту еще на момент ее создания.

Для создания программно-независимых карты возможно использовать онлайн-сервисы и языки программирования.

Онлайн-сервисы для создания интерактивных карт

Самые популярные сервисы - Google Maps (рисунок 10) и Яндекс.Карты (рисунок 11). Они довольно простые, их функционала будет достаточно для краткого описания маршрута, иллюстрации местоположения или расстояния, а также демонстрации дорожной ситуации.

С их помощью можно собрать собственную карту, изменив дизайн, добавив слои, метки и изображения. Интерактивные карты Гугла и Яндекса можно использовать на своём сайте или в приложении.

У Google есть сервис Wikimapia. Это карты которые совемещены с Википедией, каждый объект карты кликабелен, имеет фото и снабжен описанием.

Рисунок 10 – Окно сервиса Google Maps

Аналогичный ресурс есть у Яндекса - Народная карта. Этот сервис дает пользователям возможность самим наносить и описывать объекты.

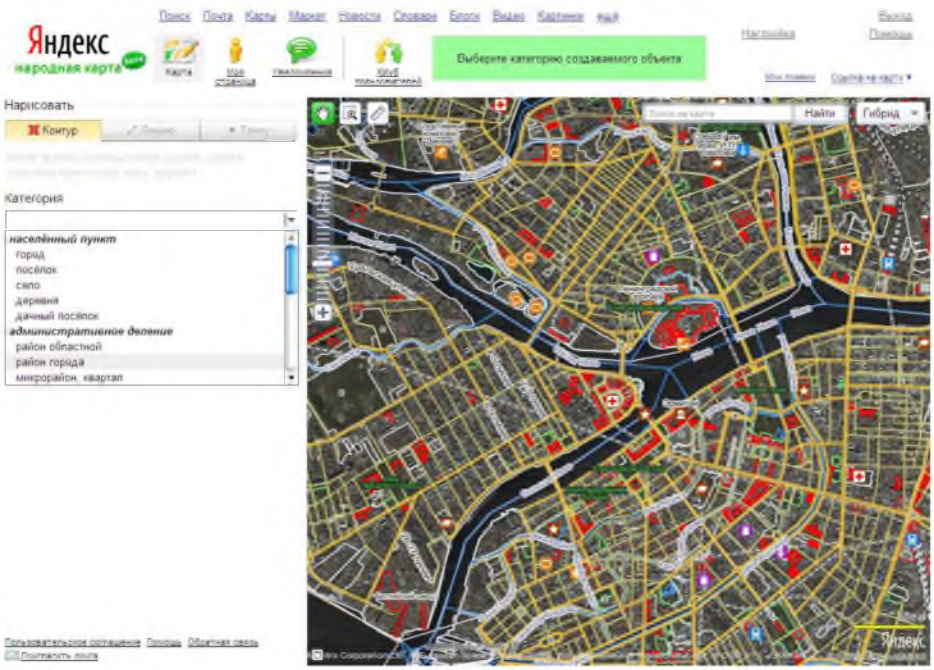


Рисунок 11 – Интерфейс Народной карты

Интерактивную модель мира позволяет создавать приложение - Google Earth (рисунок 12). В нем можно представлять рельеф и реальные размеры объектов. Многие крупные города воссозданы с помощью 3D-зданий. У «Планеты Земля» масса графических слоёв для изучения, а также карты Марса, Луны и всего звёздного неба.

Рисунок 12 – Окно приложения Google Earth

Данное приложение дает возможность отрисовать объекты, которых ещё нет. Хотя Google Earth работает как отдельное приложение, которое нельзя просто взять и вставить в свой проект, его можно использовать, к примеру, для съёмки 3D-туров. Google Планета Земля Pro предлагает бизнес-инструменты.

При создании интерактивных карт с помощью данных сервисов предполагает платное использование библиотек. А так же не получится ограничится частью карты, например территорией определенного города или района.

Другие сервисы для создания интерактивных карт:

- OpenStreetMap: предусмотрена возможность использования нескольких слоёв на выбор, среди которых, например, карта для велосипедистов, существует возможность отдельной выгрузки городов и континентов;

- 2ГИС: специализируется на точных данных об организациях, которые можно найти на карте, полезной функцией является возомжность встроить карту, которая покажет в указанном регионе заведения, отсортированные по категориям;

- ZeeMaps: содержит инструменты для добавления больших объёмов информации и её анализа на карте;

- ArcGIS: содержит огромное количество инструментов для визуализации данных, рассчитаны в первую очередь на профессионалов, поэтому имеют только платную подписку, так же имеет огромные возможности для аналитики данных и разработки новых интерактивных карт;

- CartoDB тоже имеет богатый набор инструментов для визуализации данных, карты получаются динамичными и информативными;

- Scribble Maps предлагает возможность выгружать карты в самых разных форматах.

- StoryMap JS позволяет создавать из карт целые истории, предусмотрена возможность составления маршрута и добавления текста, фото или видео к каждой точке на карте

- StoryMap JS дает возможность использовать свои карты или

фотографии, загрузив их в формате изображения и расставив на них точки с описаниями, сервис может автоматически собрать карту из 20 последних фотографий в Instagram;

- Tripline кроме Инстаграма, интегрируется с Foursquare, Flickr, Twitter, Facebook и другими сервисами;

- GeaCron позволяет создавать исторические карты, может пригодиться для проекта по истории;

- Windyty красивый погодный глобус с неплохим набором инструментов.

Большинство рассмотренных сервисов являются платными или условно-бесплатными с ограниченным пробным периодом. Большая часть интерактивных карт является Web-страницами. Поэтому для их разработки в зависимости от цели разработки применяются технологии программирования.

2.2 Анализ технологий для разработки интерактивной карты

Интегрированная среда разработки (ИСР) - это система программных средств, используемая программистам для разработки программного обеспечения. В английском языке такая среда называется Integrated development environment (сокр. IDE).

ИСР обычно включает в себя текстовый редактор, компилятор, интерпретатор, средства автоматизации разработки и сборки программного обеспечения и отладчик. Иногда также содержит средства для интеграции с системами управления версиями и разнообразные инструменты для упрощения конструирования графического интерфейса пользователя.

Программ, предназначенных для написания и редактирования исходного кода веб-приложений, великое множество. Современные и удобные редакторы способны закрывать забытые кавычки, расставлять отступы, скобки и даже дописывать за программистом команды. Мощные IDE сами обновляют содержимое файлов на удаленных серверах и хранят историю изменений проекта. Подобрать идеальный продукт - задача почти невыполнимая. Сберечь

время поможет подборка самых популярных решений.

Для удобства все инструменты разбиты на три группы:

- редакторы кода - относительно простые программы, основная функция которых состоит непосредственно в создании и изменении файлов с программным кодом;
- многофункциональные интегрированные среды (IDE) - мощные инструменты, объединяющие десятки прикладных функций;
- облачные IDE - функциональность десктопных сред для веб-разработки в виде интернет-сервиса.

Adobe Brackets

Относительно молодой кодовый редактор Brackets сделал основной упор на визуализацию и упрощение работы с CSS-препроцессорами. Проектирование веб-страницы прямо в браузере становится очень быстрым и легким делом. Brackets - отличное решение для программистов и дизайнеров, имеющих дело с пользовательскими интерфейсами и фронтендом в целом.

Основные функции:

- визуализация HTML-кода и стилей, возможность просмотра изменений в режиме реального времени и мгновенного редактирования;
- работа с PSD-файлами, возможность импорта изображений без использования Adobe Photoshop;
- удобная компиляция CSS-препроцессоров.

Sublime Text 3

Sublime Text - это удобный и быстрый редактор, работать с которым одно удовольствие. Он кроссплатформенный, нетребовательный к ресурсам компьютера и легко расширяемый. Бесплатная базовая версия предоставляет всю необходимую функциональность:

- подсветку синтаксиса распространенных языков;
- автодополнение;
- множественное выделение (очень удобно!);
- фолдинг (сворачивание блоков кода);

- удобные комбинации горячих клавиш;
- возможность разделить рабочую область на несколько окон;
- широкие возможности кастомизации.

Visual Studio Code

Многофункциональный, но компактный кодовый редактор VSC изначально был предназначен для обработки JavaScript и его расширений, но плагины позволяют добавить другие популярные языки, например, PHP и C#. Программа отлично поддерживает платформу Node.JS [25].

Технология IntelliSense обеспечивает умное автодополнение кода: редактор может дописывать названия уже объявленных функций, а подсказки содержат ссылки на нужные главы документации.

Он поддерживает ряд языков программирования, подсветку синтаксиса, IntelliSense, рефакторинг, отладку, навигацию по коду, поддержку Git и другие возможности. Многие возможности Visual Studio Code недоступны через графический интерфейс, зачастую они используются через палитру команд или JSON-файлы (например, пользовательские настройки). Палитра команд представляет собой подобие командной строки, которая вызывается сочетанием клавиш.

Visual Studio также позволяет заменять кодовую страницу при сохранении документа, символы перевода строки и язык программирования текущего документа [1].

С 2018 года появилось расширение Python для Visual Studio Code с открытым исходным кодом. Оно предоставляет разработчикам широкие возможности для редактирования, отладки и тестирования кода.

На март 2019 года посредством встроенного в продукт пользовательского интерфейса можно загрузить и установить несколько тысяч расширений только в категории «programming languages» (языки программирования).

Исходя из всех вышеперечисленных возможностей, для разработки интерактивной карты была выбрана среда Visual Studio Code.

По технологии создания Web-страницы делятся на:

1. созданные вручную;

2. созданные с помощью онлайн-конструкторов;

3. созданные на базе готовых движков.

Простые HTML сайты

Web-страницу можно создать вручную с помощью HTML и CSS. Такие Web-страницы не используют базы данных и не создают нагрузку на сервер, а создать их может любой человек с минимальными знаниями вёрстки. При желании такую Web-страницу можно дополнить скриптами JavaScript, тем самым расширив функционал.

HTML (HyperText Markup Language - «язык гипертекстовой разметки») - стандартизированный язык разметки документов во Всемирной паутине. Большинство веб-страниц содержат описание разметки на языке HTML (или XHTML). Язык HTML интерпретируется браузерами; полученный в результате интерпретации форматированный текст отображается на экране монитора компьютера или мобильного устройства.

CSS (Cascading Style Sheets - каскадные таблицы стилей) - формальный язык описания внешнего вида документа, написанного с использованием языка разметки. Преимущественно используется как средство описания, оформления внешнего вида веб-страниц, написанных с помощью языков разметки HTML и XHTML, но может также применяться к любым XML-документам, например, к SVG или XUL [26].

JavaScript - мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией стандарта ECMAScript (стандарт ECMA-262). JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам [27].

JavaScript - это интерпретируемый язык программирования, разработанный для взаимодействия с веб-страницами. JavaScript запускается на

стороне клиента Интернета и используется для программирования того, как веб-страницы будут вести себя при наступлении определенных событий. JavaScript позволяет разработчику веб-сайта управлять тем, как ведет себя веб-страница.

В браузере JavaScript может делать всё, что относится к манипуляции с HTML-документом, взаимодействию с посетителем и, с некоторыми ограничениями, с сервером:

- Проверять правильность заполнения пользовательских HTML-форм.
- Взаимодействовать с веб-камерой, микрофоном и другими устройствами.
- Менять стили HTML-элементов, прятать, показывать элементы и т.п.
- Отображать всплывающие и диалоговые окна.
- Реагировать на действия посетителя, обрабатывать клики мыши, перемещения курсора и т.п.
- Посылать запросы на сервер и загружать данные без перезагрузки страницы.

PHP

Главное преимущество PHP - код языка не конфликтует с HTML верстkой и может использоваться одновременно для разметки внешнего вида страницы с помощью HTML-тегов и функционала страницы php-частью. Он легок в освоении практически на всех этапах изучения. Отличается развитой поддержкой данных, подходит под аппаратные платформы и известные ОС. Этот язык программирования предназначен специально для работы на стороне сервера. Библиотека языка подходит для задач, выполняемых многократно во время разработки сайта.

CMS

Системы управления (CMS) представляют собой основу для формирования будущего сайта, содержат шаблоны для создания дизайна и разработки функционала под любые задачи. Сборка страниц осуществляется на основе блоков (header, footer, sidebar и др.), представленных в виде фрагментов

кода с готовой структурой. Контент и настройки сайта хранятся в базе данных MySQL.

Конструкторы

Онлайн-конструкторы (Wix, Tilda, Ucoz и пр.) - специальные сервисы, с помощью которых можно самостоятельно создавать веб-ресурсы, не имея глубоких познаний в веб-разработке. Преимущества использования конструктора - не нужно искать хостинг и специалистов для обслуживания программной части. С помощью конструктора можно создавать сайты с красивым дизайном и удобным меню, добавлять опцию приема платежей и форму заказа.

Главный недостаток конструкторов - любое расширение функционала предоставляется либо на платной основе, либо трудно осуществима. Также невозможна также полная адаптация дизайна под свои потребности. Такие сайты проблематично или невозможно переносить на CMS, нельзя изменять код, ограничены настройки для SEO-продвижения. Если конструктор перестает работать, владелец теряет доступ к своему сайту. Подобные сервисы рекомендуется использовать начинающим пользователям для приобретения опыта и представителям бизнеса для быстрого тестирования ниши. Во всех остальных случаях для построения проекта рекомендуется выбрать более надежный метод.

Движки предоставляются как на платной (Bitrix и др.), так и бесплатной основе (WordPress, Joomla, Drupal, OpenCart и пр.), различаются между собой функционалом, гибкостью настроек, возможностями интеграции с различными сервисами (например, с 1С) и прочими параметрами. Их функционал легко дополняется при помощи плагинов и скриптов. Многие веб-студии создают собственные движки, заточенные под определенные нужды.

Web-приложение - приложение, в котором клиентом выступает браузер, а сервером - веб-сервер. Браузер может являться реализацией так называемых тонких клиентов. Браузер способен отображать web-страницы и, как правило, входит в состав операционной системы, а функции его обновления и

сопровождения лежат на поставщике операционной системы. Логика приложения сосредотачивается на сервере, а функция браузера заключается в основном в отображении информации, загруженной по сети с сервера, и передаче обратно данных пользователя. Одним из преимуществ такого подхода является тот факт, что клиенты не зависят от конкретной операционной системы пользователя, и веб-приложения, таким образом, являются межплатформенными сервисами. По причине этой универсальности и относительной простоты разработки веб-приложения стали широко популярными в конце 1990-х -- начале 2000-х годов.

Существенным преимуществом построения Web приложений для поддержки стандартных функций браузера заключается в том, что функции должны выполняться, независимо от операционной системы данного клиента. Вместо того, чтобы писать различные версии для Microsoft Windows, Linux, Unix и других операционных систем, приложение создается один раз и разворачивается на любой платформе. Однако различная реализация HTML, CSS, DOM и других спецификаций в браузерах может вызвать проблемы при разработке веб-приложений и последующей поддержки. Кроме того, возможность пользователя настраивать многие параметры браузера (например, размер шрифта, цвета, отключение поддержки сценариев) может препятствовать корректной работе приложения.

Web-приложение получает запрос от клиента и выполняет вычисления, после этого формирует web-страницу и отправляет её клиенту по сети с использованием протокола HTTP. Само Web-приложение может выступать в качестве клиента других служб, например, базы данных или другого веб-приложения, расположенного на другом сервере. Ярким примером веб-приложения является система управления содержимым статей Википедии: множество её участников могут принимать участие в создании сетевой энциклопедии, используя для этого браузеры своих операционных систем (будь то Microsoft Windows, GNU/Linux или любая другая операционная система) и

не загружая дополнительных исполняемых модулей для работы с базой данных статей.

В настоящее время набирает популярность новый подход к разработке веб-приложений, single page. При использовании такого подхода страницы веб-приложения не перезагружаются целиком, а лишь догружают необходимые данные с сервера, что делает их более интерактивными и производительными.

Данное разрабатываемое Web-приложение интерактивной карты планируется создавать по типу single page application – как одностраничный продукт, что должно оптимизировать работу с кодом в браузере и повысить реактивность приложения.

В разработке используются классические технологии для построения и проектирования Web-страниц, а именно язык гипертекстовой разметки HTML, стилизация блоков посредством CSS. В качестве языка программирования был использован клиентский язык JavaScript, особенность данного выбора заключается в том, что обработка кода данного языка программирования происходит на стороне клиента, что позволяет сэкономить на ресурсах серверной части проекта. В качестве интерпретатора используется любой доступный браузер на устройстве клиента, что делает разрабатываемый продукт кроссплатформенным и повышает его потенциальную аудиторию пользователей.

Основными стеками используемых технологий разработку по мимо классических инструментов для построения Web-приложений, используются JavaScript фреймворк Vue.js, платформы Node.js и Firebase.

Vue.js - это прогрессивный JavaScript фреймворк с открытым исходынм кодам, который применяется для создания пользовательских интерфейсов. В отличие от фреймворков-монолитов, Vue создан пригодным для постепенного внедрения. Его ядро в первую очередь решает задачи уровня представления, что упрощает интеграцию с другими библиотеками и существующими проектами. С другой стороны, Vue полностью подходит и для создания сложных одностраничных Web-приложений (SPA, Single-Page Applications),

если использовать его совместно с современными инструментами и дополнительными библиотеками.

Node.js — программная платформа, основанная на движке V8 (транслирующем JavaScript в машинный код), превращающая JavaScript из узкоспециализированного языка в язык общего назначения. Node.js добавляет возможность JavaScript взаимодействовать с устройствами ввода-вывода через свой API, написанный на C++, подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из JavaScript-кода. Node.js применяется преимущественно на сервере, выполняя роль веб-сервера, но есть возможность разрабатывать на Node.js и десктопные оконные приложения (при помощи NW.js, AppJS или Electron для Linux, Windows и macOS) и даже программировать микроконтроллеры (например, tessel, low.js и espruino). В основе Node.js лежит событийно-ориентированное и асинхронное (или реактивное) программирование с неблокирующим вводом/выводом.

Firebase - это облачная база данных, которая позволяет пользователям хранить и получать сохраненную информацию, а также имеет удобные средства и методы взаимодействия с ней. Данная платформа хранит текстовые данные в JSON формате и предоставляет удобные методы для чтения, обновления и извлечения данных. Также, Firebase может помочь с регистрацией и авторизацией пользователей, хранением сессий (авторизованные пользователи), медиафайлов к которым с легкостью предоставляет доступ благодаря Cloud Storage. Таким образом данный инструмент отвечает за хранение данных Web-приложения и позволяет с минимальной ресурсозатраностью реализовать серверную часть разрабатываемого продукта.

На данном этапе были определены и описаны все используемые в разработке Web-приложения интерактивной карты инструменты.

Среда для реализации выбрана Visual Studio Code.

3 Проектирование и разработка интерактивной карты Краснодарского края (г. Сочи) для самостоятельных путешествий

3.1 Проектирование интерактивной карты Краснодарского края (г. Сочи) для самостоятельных путешествий

Структура разрабатываемого Web-приложения интерактивной карты представляет из себя каталог файлов Web-разработки, с использованием следующих форматов файлов:

- HTML;
- CSS;
- JS;
- Vue;
- Json.

Структуру разрабатываемого продукта условно можно разделить на 2 части, клиентскую и серверную.

Клиентской часть приложения является интерфейс, реализованный средствами Web-технологий, при помощи которого планируется взаимодействие пользователя с интерактивной картой. Структура каталога клиентской части представлена на рисунке 13.

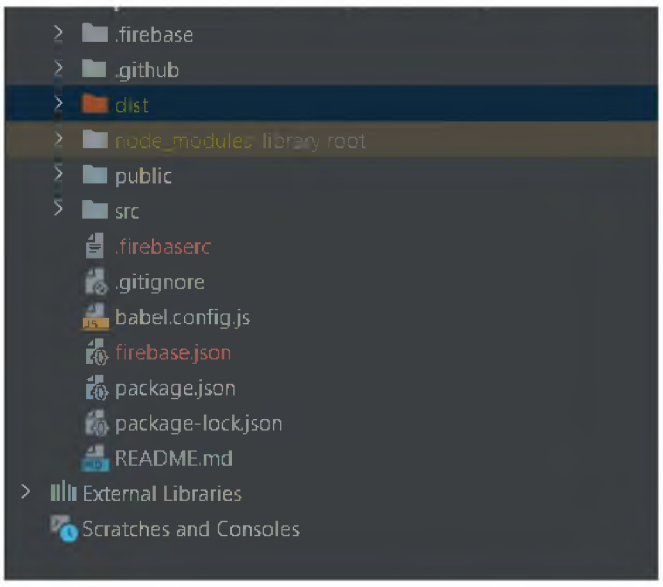


Рисунок 13 – Структура каталога клиентской части приложения

На рисунке выше представлена классическая структура приложения, спроектированного с применением фреймворка Vue.js. Основные директории данного проекта — это директории «src» и «node_modules». В директории «src» представлен весь программный код приложения, в том числе логическая работа клиентской части и сверстанные компоненты интерфейса, структура данной директории представлена на рисунке 14.

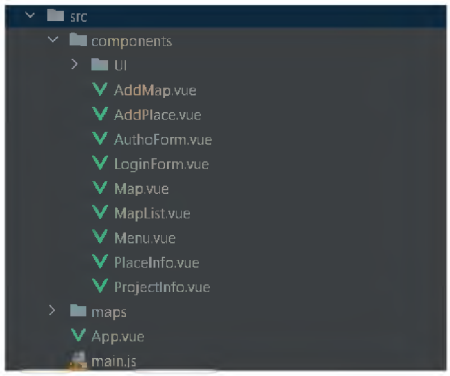


Рисунок 14 – Структура директории «src» клиентской части приложения

Однако современное Web-приложение это не только написанный с нуля код, но и определенный набор вспомогательных модулей, все подключаемые модули представлены в директории «node_modules». За подключение вышеупомянутых модулей отвечает важный файлы – package.json, данный файл содержит в себе конфигурационные настройки подключаемых модулей, название и версию приложения, а также набор терминальных команд. Блок подключенных модулей конфигурационного файла package.json представлен на рисунке 15.

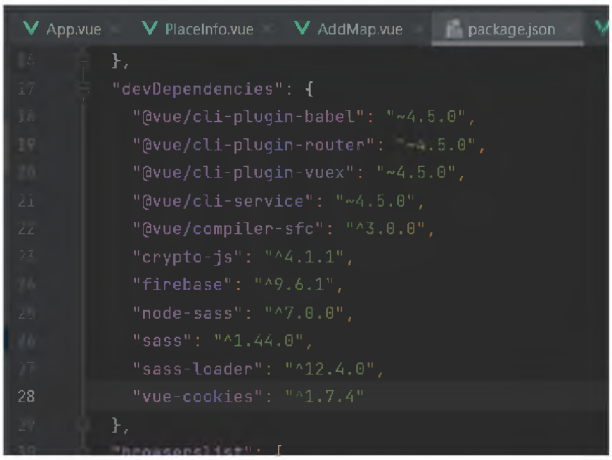


Рисунок 15 – Блок кода подключенных модулей Web-приложения

Использование такого подхода разработки продукта позволяет компактно упаковывать готовый продукт и заниматься его распространением, так как в данном файле представлены все необходимые зависимости модулей и их версий, то для пользователя который захочет установить себе данное Web-приложение нет необходимости скачивать вместе с разработанным кодом все модули и настраивать их, а достаточно только скачать основную часть проекта и через команду «npm run install» подтянуть и установить все описанные в конфигурационном файле модули.

Разобравшись с файловой структурой проекта, можно перейти к структуре интерфейса Web-приложения. Согласно спроектированному интерфейсу, основную часть на экране планируется заполнить самой картой, а меню будет представлено ввиде одной небольшой кнопки, при нажатии на которую появляются пункты меню, при помощи которых осуществляется обращение к разовым функциям, таким как:

- Авторизация;
- Выбор и загрузка карты;
- Окно информации о проекте;
- Управление масштабом карты.

Основное меню для интерактивной карты представлено на рисунке 16

Рисунок 16 – Основное меню интерактивной карты

Остальные компоненты интерфейса представлены в виде модальных окон, которые появляются только в момент взаимодействия с функционалом. Пример такого модального окна представлен на рисунке 17.

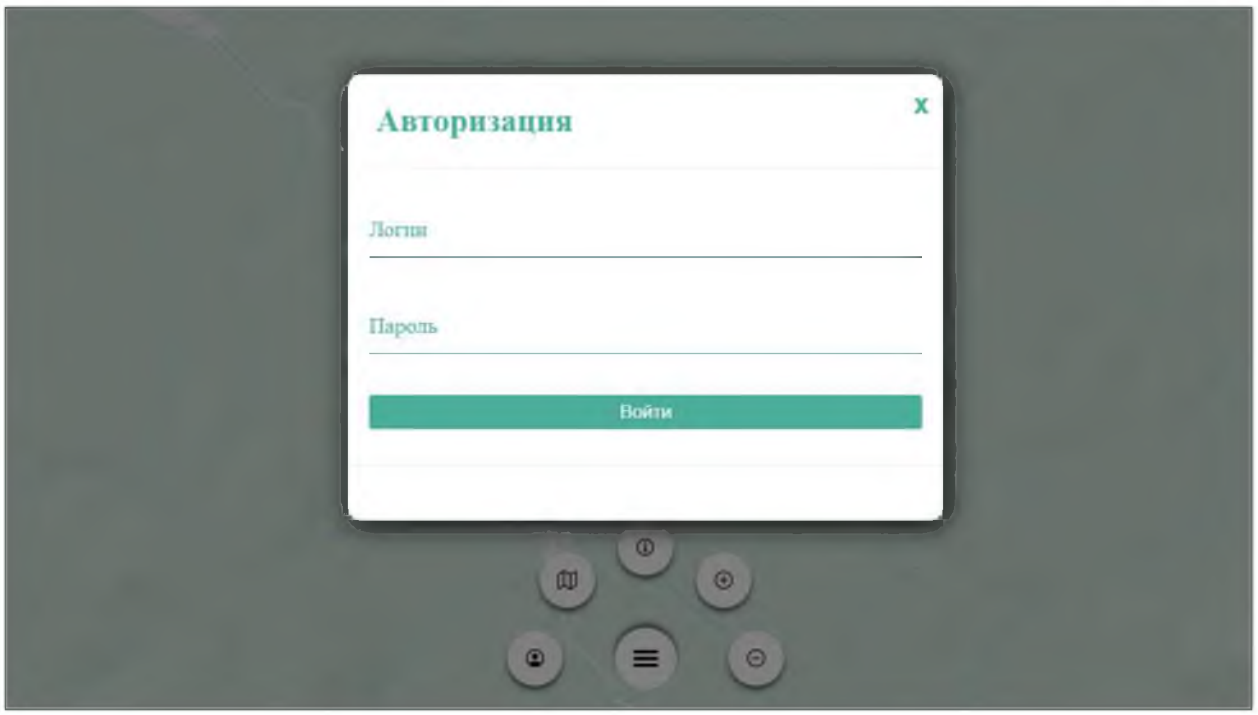


Рисунок 17 – Модальное окно авторизации

Весь интерфейс клиентской части предварительно был сверстан с использованием языка гипертекстовой разметки HTML и таблицы каскадных стилей CSS.

На данном этапе описание структуры клиентской части Web-приложения завершено и можно перейти к описанию серверной части.

За серверную часть проекта интерактивной карты отвечает платформа Firebase. Данная платформа предлагает несколько функций, которые решают внутренние задачи, позволяя разработчику сосредоточиться на функциях интерфейса приложения. Разработчики Firebase имеют доступ к широкому спектру сервисов и продуктов, полезных для проектов разработки приложений. Например, платформа предлагает опции базы данных реального времени и Firestore для базы данных.

Другие функции включают в себя прямую интеграцию облачного хранилища файлов с Web-приложениями и многие другие функции, которые

улучшают разработку приложений в бессерверной среде.

Инструменты и ресурсы, предоставляемые Firebase, обеспечивают выполнение всех возможных заданий в цикле разработки приложений.

В текущей разработке в функционал серверной части были задействованы сервисы авторизации пользователей, сервис базы данных и облачного хранилища для файлов.

Структура базы данных состоит из трех сущностей, а именно:

- Пользователи;
- Карты;
- Метки.

Описанная структура представлена в развернутом виде на рисунке 18.

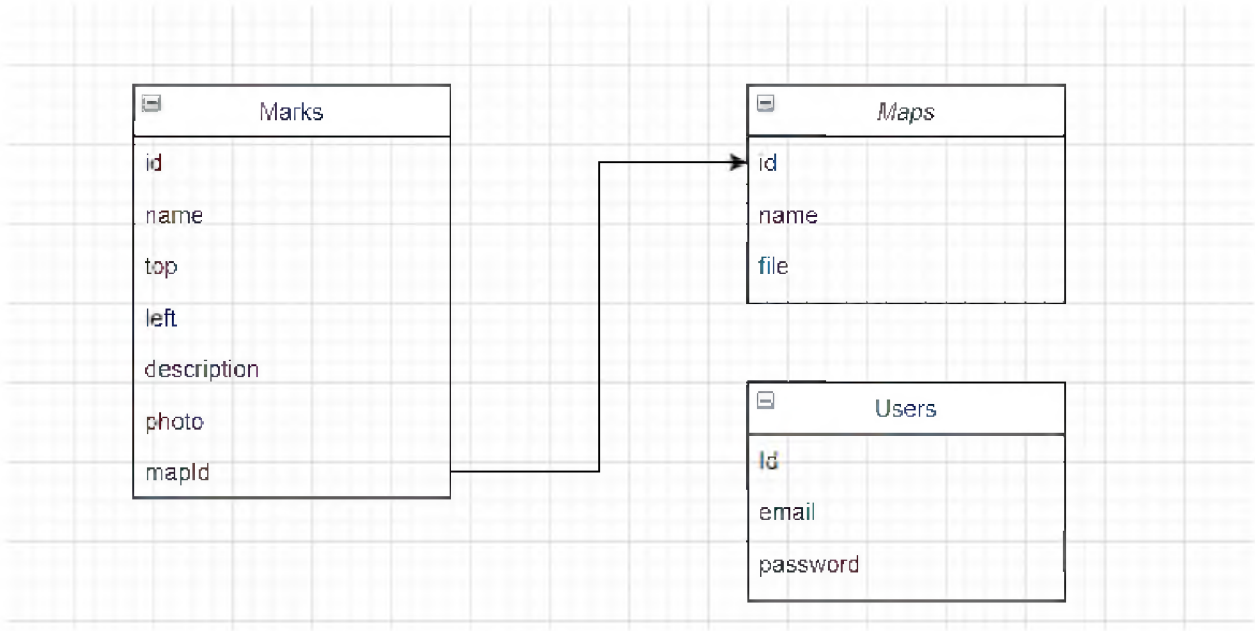


Рисунок 18 – Структура базы данных Web-приложения

Структура базы данных была спроектирована при помощи сервиса «www.draw.io», данный сервис позволяет осуществлять проектирование баз данных разной сложности и принципа построения.

Более подробное описании сущностей представлено в табличном виде ниже.

Коллекция сущности «Метки» состоит из 7 полей, ниже в таблице 1 представлены необходимые поля и их описание.

Таблица 3 - Коллекция сущности «Метки»

№	Поле	Описание
1	id	Поле содержит уникальный идентификатор для метки, расположенной на карте
2	name	Поле содержит наименование метки
3	top	Поле содержит позицию относительно верхней части карты
4	left	Поле содержит позицию относительно левой части карты
5	description	Поле содержит описание метки
6	photo	Поле содержит изображение метки
7	mapId	Поле содержит идентификатор карты, за которым закреплена метка.

Коллекция сущности «Карты» состоит из 3 полей, ниже в таблице 2 представлены необходимые поля и их описание.

Таблица 4 - Коллекция сущности «Карты»

№	Поле	Описание
1	id	Поле содержит уникальный идентификатор карты
2	name	Поле содержит наименование карты
3	file	Поле содержит путь к файлу карты

Коллекция сущности «Пользователи» состоит из 3 полей, ниже в таблице 3 представлены необходимые поля и их описание.

Таблица 5 - Коллекция сущности «Пользователи»

№	Поле	Описание
1	id	Поле содержит уникальный идентификатор пользователя
2	email	Поле содержит уникальный email пользователя, необходимый для авторизации
3	password	Поле содержит зашифрованный пароль пользователя, необходимый для авторизации.

Из описанной структуры можно с легкостью выделить две основные сущности проекта, это карты и метки на них, данная схема обусловлена тем, что каждая карта должна иметь собственные метки. Идея разделения карты Краснодарского края на отдельные участки используется в целях оптимизации проекта и облегчения работы пользователя с приложением. Третья сущность, содержащая параметры авторизации пользователей является необходимым атрибутом, без которого не получится реализовать контроль за данными самой карты. Исходя из спроектированной логики Web-приложения не

авторизованный пользователь может только ознакомиться с метками на карте и их описанием, а уже авторизованный пользователь может добавлять и удалять метки, а также загружать дополнительные участки карт.

3.2 Описание процесса разработки интерактивной карты Краснодарского края (г. Сочи) для самостоятельных путешествий

Процесс разработки проекта начинается с скачивания и установки платформы Node.js и инициализации пустого Web-приложения фреймворка Vue.js.

После установки платформы Node.js достаточно создать директорию, в которой будет вестись разработка и открыть её в терминале или IDE для разработки. В качестве IDE при разработке использовался PhpStorm – это интеллектуальный редактор для PHP, HTML и JavaScript с возможностями анализа кода на лету, предотвращения ошибок в коде и автоматизированными средствами рефакторинга для языков программирования PHP и JavaScript.

Открыв каталог для будущего проекта необходимо выполнить установку фреймворка, для этого необходимо в терминале ввести команду, которая представлена на рисунке 19.

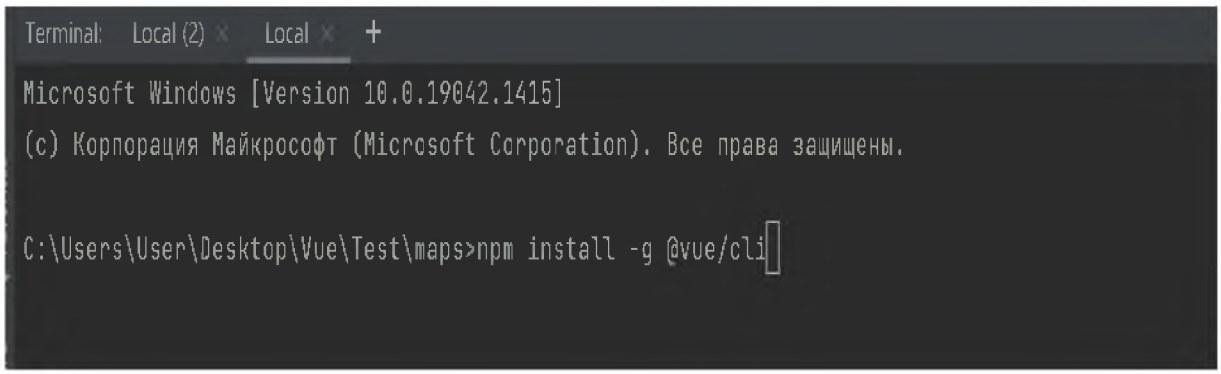


Рисунок 19 - Команда платформы Node.js для установки vue-cli

После установки фреймворка Vue.js можно приступить к инициализации приложения. Для того, чтобы инициализировать пустой проект необходимо ввести в терминал команду, представленную на рисунке 20.

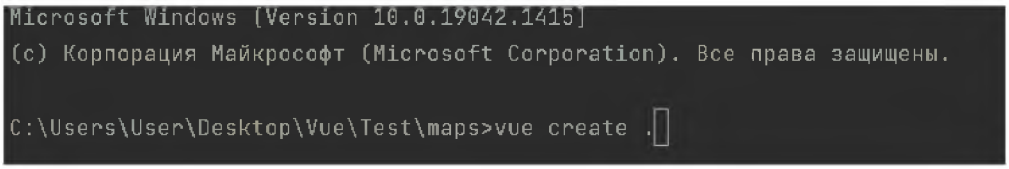


Рисунок 20 - Команда Vue.js для инициализации проекта

Необходимо обратить внимание на точку, после части «create», данный символ говорит фреймворку, что проект должен быть инициализирован в текущей директории, если бы вместо этого было указано какое-либо другое слово, то фреймворк создал бы дополнительную папку с таким же названием.

Главным файлом Web-приложения является файл «main.js», он является центральным файлом в текущем проекте, через него подключаются все дополнительные модули и в дальнейшем монтируется само приложение. Для начала необходимо сконфигурировать сам файл, а именно подключить и настроить все модули. Листинг сконфигурированного файла «main.js» представлен на рисунке 21.

```
import { createApp } from 'vue'
import App from './App'
import { initializeApp } from "firebase/app"
import { getAnalytics } from "firebase/analytics"
const firebaseConfig = {
    apiKey: "AIzaSyD6krnRBOF-E_NQcBY_cREXGF41nh8FzW8",
    authDomain: "mapper-51947.firebaseapp.com",
    projectId: "mapper-51947",
    storageBucket: "mapper-51947.appspot.com",
    messagingSenderId: "544616540915",
    appId: "1:544616540915:web:1d3694473f272fddf8a586",
    measurementId: "G-6YZBN31N3Z"
};
const fireApp = initializeApp(firebaseConfig);
export {
    firebaseConfig
}
const app = createApp(App)
app.mount( rootContainer: '#app')
```

Рисунок 21 - Листинг кода файла «main.js»

Подключив и сконфигурировав все необходимые модули будущего продукта, можно перейти к этапам декомпозиции приложения на компоненты.

Центральным компонентом представлен файл «App.vue», данный файл содержит весь необходимый код логики приложения, и является связующей

частью между другими компонентами. Классическая структура компонента Vue.js состоит из трех основных блоков:

- Template;
- Script;
- Style.

Не сложно догадаться из названий, что первый блок отвечает за HTML компоненты страницы, второй за логическую часть, а третий за стилистическую часть. Таким образом это позволяет провести декомпозицию ранее спроектированного интерфейса на отдельно состоящие компоненты и подключить их в центральный файл «App.vue».

Первый и один из самых важных — это компонент карты, он представляет из себя родительский блок, растянутый на весь экран и дочерний блок, размер которого может значительно превышать родительский, что позволяет использовать в приложении карты разных размеров. Код структуры компонента карты представлен на рисунке 22.

```
<template>
    <div  class="map">
        <div class="map-img" v-if="currentMap"
        :style="{
          height: (currentMap.height/200)*zoomPercent + 'px',
          width: (currentMap.width/200)*zoomPercent + 'px',
          backgroundImage: 'url(' + currentMap.file + ')'
        }"
        @dblclick="createMark"
        >
        </div>
  </div>
</template>
```

Рисунок 22 - Код структуры компонента карты

В данном коде уже заданы необходимые переменные, посредством которых компонент будет получать данные о файле карты, его параметрах высоты и ширины.

Исходя из логики спроектированного интерфейса, на карте должны располагаться метки с достопримечательностями, а это говорит о том, что

необходимо реализовать компонент метки и зарегистрировать его внутри компонента карты. Таким образом компоненты имеют структуру вложенности и могут состоять из нескольких компонентов. Компонент метки спроектирован также из двух блоков, это родительский блок самой метки и блок с всплывающим названием при наведении. Данный компонент на вход принимает 3 параметра, а именно значение координат точки относительно верхней и левой части карты, а также наименование. Код компонента метки представлен на рисунке 23.

Рисунок 23 – Код компонента метки

Следующий компонент, это основное меню Web-приложения. Он состоит уже из большего количества блоков, а именно родительский блок, отвечающий за позиционирование меню относительно страницы, второй блок за кнопку инициализации, а третий содержит в себе список пунктов меню. Данный компонент не принимает на вход какие-либо параметры, его код представлен на рисунке 24.

Рисунок 24 – Код компонента основного меню

Следующий компонент в проекте является типовым, это модальное окно с полями и кнопками для взаимодействия, данный компонент будет описан на примере формы авторизации пользователя.

Компонент формы авторизации состоит из одного родительского блока и трех дочерних. Родительский отвечает за параметры позиционирования и размеры формы на экране, а дочерние элементы представлены в виде полей и кнопок для взаимодействия пользователя, код структуры компонента представлен на рисунке 25.

```
<template>
  <form class="login-form" @submit.prevent>
    <div class="user-box">
      <input v-model="userInput.login" type="text" name="" required="">
      <label>Логин</label>
    </div>
    <div class="user-box">
      <input v-model="userInput.password" type="password" name="" required="">
      <label>Пароль</label>
    </div>
    <button @click="login" class="btn-green">Войти</button>
  </form>
</template>
```

Рисунок 25 - Код компонента формы авторизации

После реализации основных компонентов Web-приложения можно завершить этап декомпозиции проекта на составные части и перейти к описанию логики работы и взаимодействия между компонентами. Всего за этап декомпозиции было разработано 11 отдельных компонентов.

Для взаимодействия компонентов между собой используется описанная в блоке «script» логика на языке программирования JavaScript, известно, что компоненты могут на вход принимать какие-либо параметры для дальнейшего их обработки и использования. Для получения входных параметров необходимо описать так называемые свойства «props» для компонента, пример описания данного свойства представлен на рисунке 26.

```
<script>
export default {
  props: {
    left:{
      type:Number,
      required: true
    },
    top:{
      type:Number,
      required: true
    },
    name:{
      type:String,
      required: true
    },
  },
  name: "Mark"
}
</script>
```

Рисунок 26 – Описание входных данных для компонента метки

Параметр «props» содержит внутри себя массив с описанием входных параметров, в данном массиве указывается наименование параметров, их тип и являются ли они обязательными для работы компонента или нет.

Разобравшись с способом установки входных параметров для компонентов, необходимо настроить эту передачу данных в сам компонент. Для этого нужно в родительском компоненте, где будет инициализирован компонент с входными параметрами, задать значение модели для описанных в компоненте свойств. Пример кода для присвоения значений дочернему компоненту представлен на рисунке 27.

```
<Mark v-for="mark in marks"
      :left="(mark.left/mark.zoomPercent)*zoomPercent"
      :top="(mark.top/mark.zoomPercent)*zoomPercent"
      :name="mark.name"
      @click="openPlaceInfo(mark.id)"
/>
```

Рисунок 27 – Присвоение входных значений для компонента метки

Для присвоения значений достаточно указать наименование входного параметра и перед ним поставить двоеточие, а далее передать саму модель данных через равно. Для передачи данных можно использовать как уже обработанные данные, так и проводить их обработку непосредственно в момент передачи.

Таким образом были реализованы способы передачи данных из родительского компонента в дочерний. Однако так как компоненты должны быть переиспользуемые и не содержать в себе логику, выходящую за рамки самого компонента, то встает необходимость передавать данные уже не из родительского компонента в дочерний, а наоборот из дочернего в родительский и выше по структуре. Для реализации подобного функционала необходимо использовать пользовательские события, в результате срабатывания которых вызывается метод «$emit», данный метод позволит передать определенное событие компонента и вместе с ним данные для родительского элемента из дочернего. Пример реализации описанного функционала представлен на рисунке 28.

Рисунок 28 – Код передачи данных из дочернего в родительский компонент

Исходя из представленного выше примера, можно заметить, что первым параметром метод «$emit» принимает наименование события, которое будет отлавливать родительский компонент, а вторым параметром набор данных.

Завершив реализацию взаимодействия компонентов между собой, остается только запрограммировать логику работы самого Web-приложения и его общения с серверной частью представленную в виде Firebase.

Как упоминалось ранее, центральным компонентом является файл – «App.vue» в нем и описаны основные методы по взаимодействию с данными.

Первая и необходимая часть логики для Web-приложения интерактивной карты является реализация функционала приближения и отдаления от карты, для этого было создано два метода – zoomPlus и zoomMinuse, которые не отличаются особой логической структурой, а всего лишь редактируют значение

для параметра zoomPercent, данный параметр участвует в расчётах параметров ширины и высоты для файла карты. Код методов редактирования масштаба и расчёта параметров ширины и высоты карты представлен на рисунке 29.

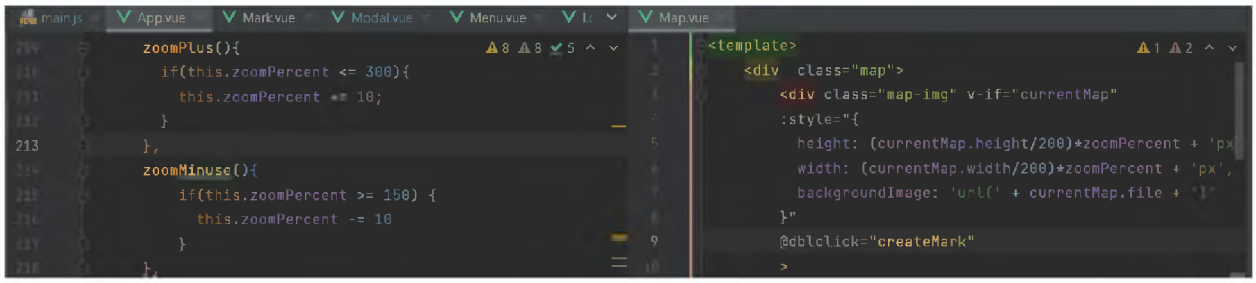


Рисунок 29 – Код методов редактирования масштаба и расчёта параметров ширины и высоты карты

На рисунке выше слева на права представлены методы увеличения и уменьшения масштаба, а также код, отвечающий за расчёт размера карты.

После реализации функционала для управления масштабом карты необходимо приступить к реализации логики, при помощи будет происходить авторизация пользователя. Для этого используется библиотека firebase. Установив пользовательское событие на нажатие, кнопки «Войти» из формы авторизации, данное срабатывание перенаправлено на вызов метода «login ()», который на вход из дочернего компонента принимает значения, которые ввел пользователь для авторизации. Получив данные значения, выполняется метод «signInWithEmailAndPassword» библиотеки Firebase, который отправляет запрос на API серверной части для проверки факта авторизации, если введенные данные верны, то метод возвращает значение «true». Однако данного события недостаточно для реализации факта авторизации в самом Web-приложении, необходимо при успешной авторизации зарегистрировать за пользователем уникальный идентификатор, который будет подтверждать, что он авторизован и будет существовать какое-то время. Для регистрации уникального значения, подтверждающего факт авторизации используются Cookie браузера, уникальная метка создается в ресурсы самого браузера и хранится на стороне клиента. В качестве значения для метки используется

случайно сгенерированная комбинация чисел и символов. Код реализации метода авторизации представлен на рисунке 30.

Рисунок 30 – Код метода авторизации пользователя

Так же по аналогии был реализован метод, позволяющий пользователю деавторизоваться в случае необходимости, данный метод просто удаляет созданный ранее уникальный идентификатор.

После реализации авторизации пользователя, можно приступить к созданию логики добавления новых меток на карте. Так как добавление новых меток связано одновременно и с взаимодействием с базой данных и с работай с облачным хранилищем файлов, то дополнительно к файлу «App.vue» был подключен модуль, отвечающий за взаимодействие с облачным хранилищем Firebase.

Логика добавления новой метки берет свое начало из компонента карты, где при двойном клике по точке на карте происходит вызов события с передачей параметров координат метки в родительский элемент, где они регистрируются в модели для объекта метки. Следующий этап в создании метки это обращение к компоненту, который содержит форму с параметрами для добавления метки и полем для загрузки фотографии. Основная функция компонента формы добавления новой метки — это загрузка изображения на сервер и передача введенных данных в родительский компонент. Так как загрузка изображения происходит до факта регистрации данных по новой метке, то в модель метки передается не наименование файла, а уже сгенерированный URL адрес сразу на само изображение, это позволяет в дальнейшем избежать дополнительных запросов на серверную часть

приложения. Код реализации загрузки изображения в облачное хранилище Firebase представлен на рисунке 31.

```
addNewMark() {
  //Загрузка изображения в firebase
  const storage = getStorage()
  const mountainsRef = ref(storage, this.placeModel.photo.name);
  const mountainImagesRef = ref(storage, url: 'general/' + this.placeModel.photo.name);
  if (mountainsRef.name === mountainImagesRef.name) {
    uploadBytes(mountainImagesRef, this.placeModel.photo).then((snapshot) => {
      console.log(snapshot)
      console.log('Изображение места загружено успешно на сервер');
      getDownloadURL(mountainImagesRef)
          .then((url) => {
            this.placeModel.photo = url
            this.$emit('addMark', this.placeModel)
          })
          .catch((error) => {
            console.log(error)
          })
    });
  }
}
}
```

Рисунок 31 – Код загрузки изображение в Firebase

Сформировав полноценную модель данных для нового объекта метки и получив уведомление об успешно загруженном изображении на сервер, срабатывает метод для добавление собранной информации в базу данных. Код регистрации новой метки в базе данных представлен на рисунке 32.

```
async addNewMark(placeModel){
  if(!this.checkAuth()){
    console.log('Недостаточно прав для создания метки')
    return false;
  }
  this.newMark.name = placeModel.name
  this.newMark.description = placeModel.description
  this.newMark.photo = placeModel.photo
    const db = getFirestore()
    try {
      const docRef = await addDoc(collection(db, path: "Marks"), this.newMark);
      console.log("Запись в базу выполнена успешно ID: ", docRef.id);
    } catch (e) {
      console.error("Ошибка записи в базу данных: ", e);
    }
  this.modalClose();
  this.clearMarkModel();
  await this.fetchMarks()
},
```

Рисунок 32 – Код регистрации новой метки в базе данных

Описанный выше метод регистрации новой метки является асинхронным

(async), а это значит внутри находящийся метод await не даст сработать коду, который представлен после пока не получит успешный ответ и не внесет данные на сервер.

Реализовав способ добавления метки, так же был реализован по аналогии метод добавления новых карт.

Описав и реализовав все необходимые методы по добавлению и удалению данных, остается только реализовать методы, отвечающие за актуализацию и сбор данных в момент запуска Web-приложения пользователем. Данные методы необходимы так как клиентская и серверная часть приложения расположены в разных местах и не имеют прямого доступа к друг другу, а общаются только посредством запросов к API.

Для получения меток по выбранной карте реализован метод «fetchMarks», данный метод проверяет какая карта загружена в приложении на данный момент и делает выборку меток по значению идентификатора выбранной карты. Код реализации метода «fetchMarks» представлен на рисунке 33

Рисунок 33 – Код реализации метода «fetchMarks»

Метод для получения списка доступных карт реализован по аналогии с вышеописанным методом. Однако стоит отметить, что на момент запуска проекта впервые у пользователя не будет открыта какая либо карта и ему необходимо будет выбрать её самостоятельно через основное меню приложения.

Расчет экономической эффективности разрабатываемого проекта

Экономическая эффективность - это соотношение полезного результата и затрат факторов производственного процесса. Для количественного определения экономической эффективности используется показатель эффективности. Показатель эффективности - относительный показатель эффективности операции, проекта или процесса. Определяется как частное от деления потенциального эффекта к её ресурсоемкости.

Потенциальный экономический эффект измеряется максимальной экономией затрат совокупного общественного труда, которая может быть достигнута на основе применения определенной технологии. Реальный экономический эффект измеряется той экономией затрат труда и средств, которая может быть получена при применении альтернативных технологий.

Разрабатываемый проект выполняется в отдельных программных компонентов. Каждый компонент распространяется бесплатно со свободной лицензией. Чтобы рассчитать потенциальный экономический эффект, необходимо сравнить стоимость разработки проекта при применении аналогичных проприетарных компонентов.

СУБД

В проекте используется Firebase - это облачная база данных. На рынке имеется несколько альтернативных серверов пространственных данных, они приведены в таблице 6.

Таблица 6 - Альтернативные проприетарные СУБД

Производитель	Продукт	Стоимость, тыс. руб.
Oracle	Oracle Spatial 12c	750
Microsoft	SQL Server	180
ESRI	ArcSDE	300
Интервал стоимости		180-750

Настольная ГИС.

Для наполнения проекта данными и их редактирования используется

функционал приложения. В качестве альтернативных программ могут использоваться некоторые проприетарные

Таблица 7 - Альтернативные проприетарные настольные ГИС

Производитель	Продукт	Стоимость, тыс. руб.
Pitney Bowes	MapInfo Professional	75,5
Autodesk	AutoCAD Map 3D	142,7
Autodesk	AutoCAD Civil 3D	161,8
ESRI	ArcView 10.0	126,0
Интервал стоимости		75,5-161,8

Картографический веб-сервер.

Кроме используемого в проекте GeoServer, существуют проприетарные серверы. Основные из них приведены в таблице 8.

Таблица 8 - Альтернативные проприетарные картографические веб-серверы

Производитель	Продукт	Стоимость, тыс. руб.
Pitney Bowes	MapXtreme	1375,0
ESRI	ArcGIS Server 9.2	1955,6
Интервал стоимости		1375,0-1955,6

Интерактивная карта.

Авторизированный пользователь имеет возможность добавления объектов на карты и их редактирования существуют также и проприетарные аналоги. Они приведены в таблице 9.

Таблица 9 - Альтернативные проприетарные интерактивные карты

Производитель	Продукт	Стоимость, тыс. руб.
WebUnion Media	iMapBuilder	3,0
MosMap	MosMap Interactive	2,5
Интервал стоимости		2,5-3,0

Расчет экономической эффективности приведен в таблице 9.

Таблица 10 - Расчет экономической эффективности

Потенциальный эффект		Ресурсоёмкость	
Позиция	Стоимость, тыс. руб.	Позиция	Стоимость, тыс. руб.
СУБД	180 - 750	СУБД	-
Настольная ГИС	75,5 - 161,8	Настольная ГИС	-

Продолжение таблицы 10

Картографический веб-сервер	1375,0 - 1955,6	Картографический веб-сервер	-
Интерактивная карта	2,5 - 3,0	Интерактивная карта	-
Оплата труда	20,0	Оплата труда	50,0
Итого	1663,0 - 2890,4	Итого	50,0
Экономическая эффективность	$E = \frac{[1663 - 2890]}{50,0} = 33,26 - 57,8$		

Экономическая эффективность оценена как от 33,26 до 57,8 рублей на рубль вложений потенциально. Этот показатель очень высок, это говорит о рентабельности использования открытого программного обеспечения.

Заключение

В данное время актуальность интерактивных карт с каждым днем возрастает, так как они наглядно показывают аналитическую, статистическую и иную информацию.

В данной выпускной квалификационной дипломной работе было рассмотрено создание интерактивной карты Краснодарского края (г. Сочи) для самостоятельных путешествий. Были решены поставленные в выпускной квалификационной дипломной работе задачи:

- проведен анализ предметной области и предъявляемых требований к интерактивным картам,
- были изучены теоретические материалы по классификации интерактивных карт,
- проведен анализ и выбор средств и технологий для разработки,
- освоены технологий разработки в среде Visual Studio Code,
- проанализированы объекты посещения туристов на территории курорта и выделены наиболее значимые для отображения на интерактивной карте,
- произведено проектирование дизайна интерактивной карты.
- разработано программное решение;
- протестировано готовое программное решение с целью выявлениея и устранения ошибок.

В ходе данной работы была разработана интерактивная карта в виде Web-страницы для самостоятельных путешествий, которая имеет действительно оригинальный и простой интерфейс, который будет не сложно освоить даже неопытному пользователю. Данный сайт ориентирован на людей, предпочитающих самостоятельно посещать достопримечательности.

К перспективам развития можно отнести:

- размещение карты на сайтах с информацией о местах проживания туристов (отелях, гостиницах),
- создание онлайн чата с консультантом,

- для удобной работы с ними, так же возможна доработка интерфейса сайта с целью дальнейшего повышения его информативности, привлекательности и удобства

- перенос сайта на платный полнофункциональный надежный хостинг с поддержкой серверов POP3/SMTP, данные сервера необходимы для регистрации посетителей сайта.

По итогу, разработанный информационный ресурс может быть использован для представления информации достопримечательностях. С его помощью пользователи смогут получать необходимую информацию.

Список использованной литературы

1. Visual Studio Code: мощное руководство пользователя (dev-gang.ru) [Электронный ресурс] (дата обращения: 12.09.2021).

2. Буравчикова, Д. «30 уникальных мест в России для отдыха, туризма и экскурсий» / Д. Буравчикова // AIF.ru: [сайт] – URL: https://aif.ru/travel/30_unikalnyh_mest_v_rossii_dlya_otdyha_turizma_i_ekskursiy (дата обращения: 07.12.2021).

3. Внутренний туризм, въездной туризм, статистика // Самые посещаемые города и регионы страны: [сайт] – URL http://turstat.com/toprussianresortssummer2021 (дата обращения: 08.12.2021)

4. География //Оценка природных условий для туризма: карты: [сайт] – URL https://geographyofrussia.com/ocenka-prirodnyx-uslovij-dlya-turizma/ (дата посещения: 29.11.2021)

5. Гладкий, А.В. Современные картографические модели и особенности их использования в туризме / А.В. Гладкий, А.А. Скляров // Псковский региональный журнал – 2017. – № 2 (29). с. 32 - 36.

6. Дакетт Джон_HTML и CSS_Все что нужно знать.2013.pdf - Google Диск, 478 с. [Электронный ресурс] (дата обращения: 14.09.2021).

7. Дакетт Джон_JavaScript и jQuery. Интерактивная разработка_2017.pdf - Google Диск, 641 с. [Электронный ресурс] (дата обращения: 15.09.2021).

8. Дунаев Вадим_Самоучитель JavaScript_2-е изд._2005.pdf - Google Диск, 402 с. [Электронный ресурс] (дата обращения: 16.09.2021).

9. Зятькова Л. К., Комиссарова Е. В., Колесников А. А. Современные web-технологии для создания интерактивных мультимедийных картографических произведений // Изв. вузов. Геодезия и аэрофотосъемка. – 2012. – № 2-1. – с. 95 - 98.

10. Интерактивная инфографика «Какое место я занимаю в мировом населении? Как долго я буду жить?». – URL: https://population.io/ (дата

обращения: 06.10.2021).

11. Интерактивная карта Urban Layers.– URL: http://io.morphocode.com/urban-layers/ (дата обращения: 06.11.2021).

12. Интерактивная карта изменения границ России на протяжении веков. – URL: https://histography.ru/#welcome (дата обращения: 07.10.2021).

13. Интерактивная мировая историческая карта с 3000 до н. э. – URL: http://geacron.com/home-en/ (дата обращения: 04.10.2021).

14. Интерактивные карты [Электронный ресурс]. Режим доступа: http://www.scanex.ru/service/razrabotka-i-vnedrenie-veb-gis/interaktivnye-karty. Дата доступа: 09.11.2021

15. Интересное на карте: [сайт] – Туристические карты достопримечательностей Санкт-Петербурга. – URL http://mapinmap.ru/archives/7561 (дата обращения: 29.11.2021)

16. Кан М._Основы программирования на JavaScript.2016.pdf - Google Диск, 168 с [Электронный ресурс] (дата обращения: 17.05.2021).

17. Картоведение: Учебник для вузов / А.М. Берлянт, А.В. Востокова, В.И. Кравцова и др.; Под ред. А.М. Берлянта. – М.: КДУ, 4-е издание, дополненное, 2014. – 448 с.

18. Клименко Р.А._Веб-мастеринг. Изучаем HTML5, CSS3, JavaScript, PHP, CMS, AJAX, SEO (на 100%)_2013.pdf - Google Диск, 512 с [Электронный ресурс] (дата обращения: 18.05.2021).

19. Куприна Л.Е. Туристская картография: учеб. пособие. 2-е издание / Л.Е. Куприна – Москва: Флинта, 2016. – 278с.

20. Лисицкий, Д. В. Мультимедийная картография: учеб. пособие / Д. В. Лисицкий, Е. В. Комиссарова, А. А. Колесников. – Новосибирск : СГУГиТ, 2016. – 108 с.

21. Лисицкий, Д. В. Методические основы веб-картографии / Д. В. Лисицкий, П. М. Кикин. Изв. Вузов. Геодезия и аэрофотосъёмка. 2014. с. 85–91

22. Полуэктова, Н. Р. Разработка веб-приложений : учебное пособие для среднего профессионального образования / Н. Р. Полуэктова. - Москва :

Издательство Юрайт, 2021. - 204 с. - (Профессиональное образование). - ISBN 978-5-534-14744-5. - Текст : электронный // ЭБС Юрайт [сайт]. - URL: https://urait.ru/bcode/479863 (дата обращения: 14.09.2021).

23. Полуэктова, Н. Р. Разработка веб-приложений: учебное пособие для вузов / Н. Р. Полуэктова. - Москва : Издательство Юрайт, 2021. - 204 с. - (Высшее образование). - ISBN 978-5-534-13715-6. - Текст : электронный // ЭБС Юрайт [сайт]. - URL: https://urait.ru/bcode/466449 (дата обращения: 15.09.2021).

24. Прохорова, Е.А. Социально-экономические карты: учеб. пособие / Е.А. Прохорова – Москва: КДУ, Добросвет, 2018. – URL https://bookonlime.ru/product/socialno-ekonomicheskie-karty (обращения: 20.09.2021)

25. Редактор кода Visual Studio Code. Самый подробный гайд по настройке и установке плагинов для начинающих / Хабр (habr.com) [Электронный ресурс] (дата обращения: 19.10.2021).

26. Роббинс Дженнифер_HTML5, CSS3 и JavaScript_Исчерпывающее руководство.2014.pdf - Google Диск, 516 с [Электронный ресурс] (дата обращения: 19.10.2021).

27. Справочник по JavaScript [Электронный ресурс]/ Справочник по JavaScript - Электрон. Дан - Режим доступа: https://javascript.ru/manual (дата обращения 19.10.2021)

28. Сысолетин, Е. Г. Разработка интернет-приложений [Электронный ресурс]: учебное пособие для среднего профессионального образования / Е. Г. Сысолетин, С. Д. Ростунцев. - Москва: Издательство Юрайт, 2020. - 90 с. - Режим доступа: http://biblio-online.ru/bcode/456393 (дата обращения: 19.09.2021)

29. Тузовский, А. Ф. Проектирование и разработка web-приложений [Электронный ресурс]: учебное пособие для среднего профессионального образования / А. Ф. Тузовский. - Москва: Издательство Юрайт, 2020. - 218 с. - Режим доступа: https://urait.ru/bcode/456394 (дата обращения: 25.09.2021)

30. Элеонара Лутц Интерактивная карта Space Station Earth – URL: https://a.tiles.mapbox.com/v3/eleanor.ipncow29/page.html?secure=1#17/38.91295/-

77.03229 (дата обращения: 03.10.2021).

Приложение 1

Низкий уровень интерактивности

Примером является интерактивная карта изменения границ России на протяжении веков, пользователю нужно всего лишь крутить колесико мышки для того, чтобы перед ним постепенно появилось все содержимое карты (рисунок 34).

Рисунок 34 - Интерактивная карта изменения границ России на протяжении веков [12]

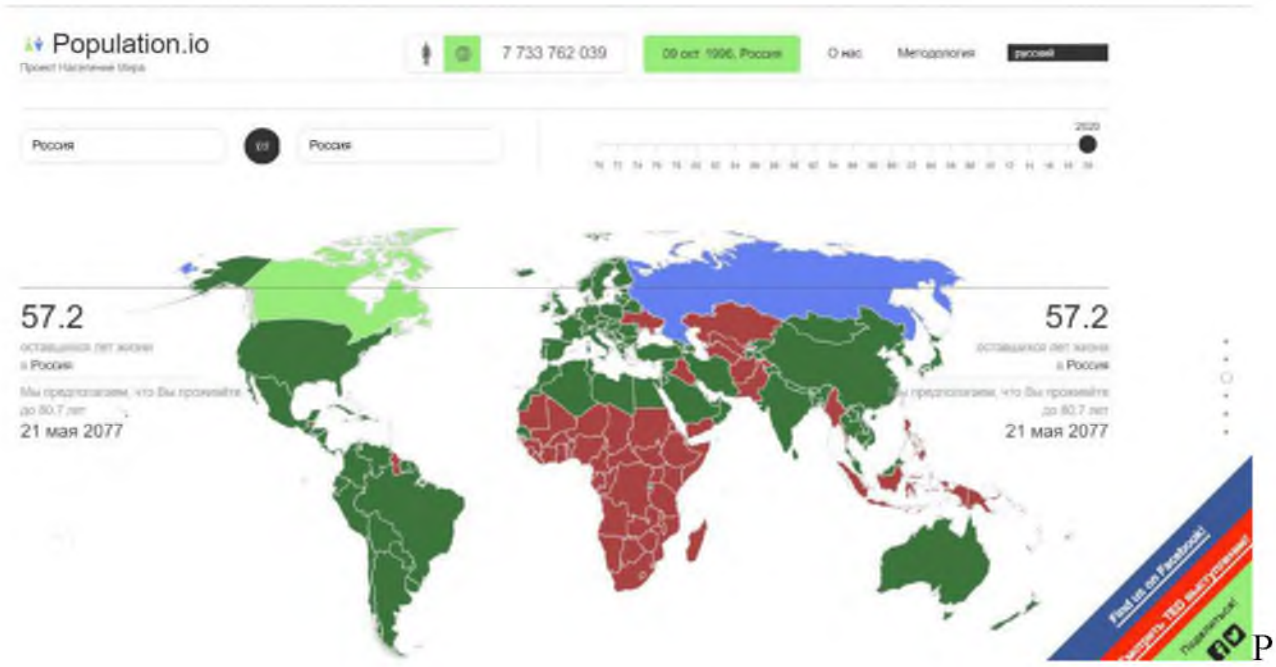
Средний уровень интерактивности

Например, Urban Layers –интерактивная карта, исследующая структуру городских слоев Манхэттена. С помощью карты можно посмотреть, как архитектурно развивался Манхэттен, начиная с 1765 года, до настоящего времени, и из скольких слоев архитектуры состоит каждый этап. Для пользователя доступен таймлайн, масштабирование, перетаскивание, а сами архитектурные уровни выделяются разными цветами по периодам, что позволяет их сравнивать и следить за изменениями (рисунок 35).

Рисунок 35 - Интерактивная карта Urban Layers [11]

Высокий уровень интерактивности

Примером может служить инфографика под названием «Какое место я занимаю в мировом населении? Как долго я буду жить?», в которой пользователю необходимо ввести свою дату рождения и пол, чтобы инфографика определила предположительную продолжительность жизни. При этом все сведения появляются на карте мира (рисунок 36).



исунок 36 - Интерактивная инфографика «Какое место я занимаю в мировом населении? Как долго я буду жить?» [10]

Приложение 2

Таблица 6 - Сводная таблица параметров эффектов анимации для карты

	Масштабирование	Перемещение	Перемещениепо траектории	Масштабиро-вание и пере- мещение	Слои	Тематическиеслои
Исходный масштаб	+			+		
Требуемый масштаб	+			+		
Исходное положение		+	+	+		
Требуемое положение		+	+	+		
Тип траектории			+	+		
Координаты траектории			+	+		
Продолжительность анимации	+	+	+	+	+	+
Тип анимационного перехода	+	+	+	+	+	+

Таблица 7 - Сводная таблица параметров эффектов анимации для объектов карты

	Масштабирование	Перемещение	Перемещение по траектории	Масштабированиеи перемещение	Изменение элементов	Отображение дополнительной информации	Изменение прозрачности
Исходный масштаб	+			+			
Требуемый масштаб	+			+			
Исходное положение		+	+	+			
Требуемое положение		+	+	+			
Тип траектории			+	+			
Координаты траектории			+	+			
Продолжительность анимации	+	+	+	+	+	+	+
Тип анимационного перехода	+	+	+	+	+	+	+
Количество состояний	+		+	+	+	+	+
Описание состояний	+		+	+	+	+	+