МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ   
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

Кафедра Информационных систем и технологий

Специальность 6-05-0612-01 «Программная инженерия»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА КУРСОВОГО ПРОЕКТА**

по дисциплине «Компьютерные языки разметки»

Тема «Веб-сайтБанка»

**Исполнитель**

студент(ка) 1 курса 8 группы \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д. А. Мартыненко

подпись, дата

**Руководитель**

старший преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е. В. Барковский

должность, ученая степень, ученое звание подпись, дата

Допущен(а) к защитe \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

дата, подпись

Курсовой проект защищен с оценкой

Руководитель \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_ Е. В. Барковский

подпись дата инициалы и фамилия

Содержание

[Введение 3](#_Toc197294181)

[1 Постановка задачи 4](#_Toc197294182)

[1.1 Обзор аналогичных решений 4](#_Toc197294183)

[1.1.1 Аналог «Альфа Банк» 4](#_Toc197294184)

[1.1.2 Аналог «Тинькофф Банк» 5](#_Toc197294185)

[1.1.3 Аналог «Беларусбанк» 6](#_Toc197294186)

[1.2 Техническое задание 8](#_Toc197294187)

[1.3 Выбор средств реализации программного продукта 8](#_Toc197294188)

[1.4 Вывод 9](#_Toc197294189)

[2. Проектирование веб-сайта 10](#_Toc197294190)

[2.1 Выбор способа верстки 10](#_Toc197294191)

[2.2 Выбор стилевого оформления 10](#_Toc197294192)

[2.3 Выбор шрифтового оформления 11](#_Toc197294193)

[2.4 Разработка логотипа 11](#_Toc197294194)

[2.5 Разработка пользовательских элементов 12](#_Toc197294195)

[2.6 Разработка спецэффектов 13](#_Toc197294196)

[2.7 Выводы 16](#_Toc197294197)

[3. Реализация структуры веб-сайта 17](#_Toc197294198)

[3.1 Структура HTML-документа 17](#_Toc197294199)

[Листинг 3.1 – Структура HTML главной страницы 17](#_Toc197294200)

[3.2 Добавление таблиц стилей Scss и CSS 18](#_Toc197294201)

[3.3 Использование стандартов XML (SVG) 18](#_Toc197294202)

[3.4 Использование JavaScript 20](#_Toc197294203)

[3.5 Выводы 22](#_Toc197294204)

[4 Тестирование веб-сайта 23](#_Toc197294205)

[4.1 Адаптивный веб-сайта 23](#_Toc197294206)

[4.2 Кроссбраузерность веб-сайта 25](#_Toc197294207)

[4.3 Руководство пользователя 26](#_Toc197294208)

[4.4 Тестирование кода 27](#_Toc197294209)

[4.5 Выводы 27](#_Toc197294210)

[Заключение 28](#_Toc197294211)

[Список использованных источников 29](#_Toc197294212)

Введение

Сайт – обязательная часть продвижения и развития компании, визитка и официальное лицо, который сочетает в себе современный и модный подход к ведению бизнеса. Также это место, где можно найти подробную информацию о компании, услугах, выполнении услуг и контактах. Информацию на нём можно легко и быстро публиковать и изменять, что является преимуществом перед различными видами традиционной рекламы. Это значит, что сайт является ещё и хорошим рекламным инструментом, инструментом привлечения новых клиентов. С помощью каталогов и поисковиков Интернет-пользователь может ознакомится с предлагаемыми услугами и стать потенциальным клиентом. Что и подчёркивает актуальность курсового проекта.

Цели курсового проекта:

* разработка структуры web-сайта для мелкой бытовой техники с использованием современных средств проектирования и разработки;
* применение навыков, полученных в результате прохождения курса «Компьютерные языки разметки».

Задачи курсового проекта:

* проанализировать существующие языки разметки, инструменты и библиотеки для создания веб-сайта.
* разработать макет и прототип сайта.
* разработать структуру веб-сайта.
* наполнить сайт информацией по теме.
* протестировать веб-сайт.

1 Постановка задачи

1.1 Обзор аналогичных решений

Перед тем как приступать к разработке своей системы рекомендуется изучить уже готовые решения и посмотреть, что по мнению других разработчиков является наиболее необходимым для пользователей, чтобы приложение было полезным и удобным в использовании.

1.1.1 Аналог «Альфа Банк»

На рисунке 1.1 представлен внешний вид главного каталога сайта.

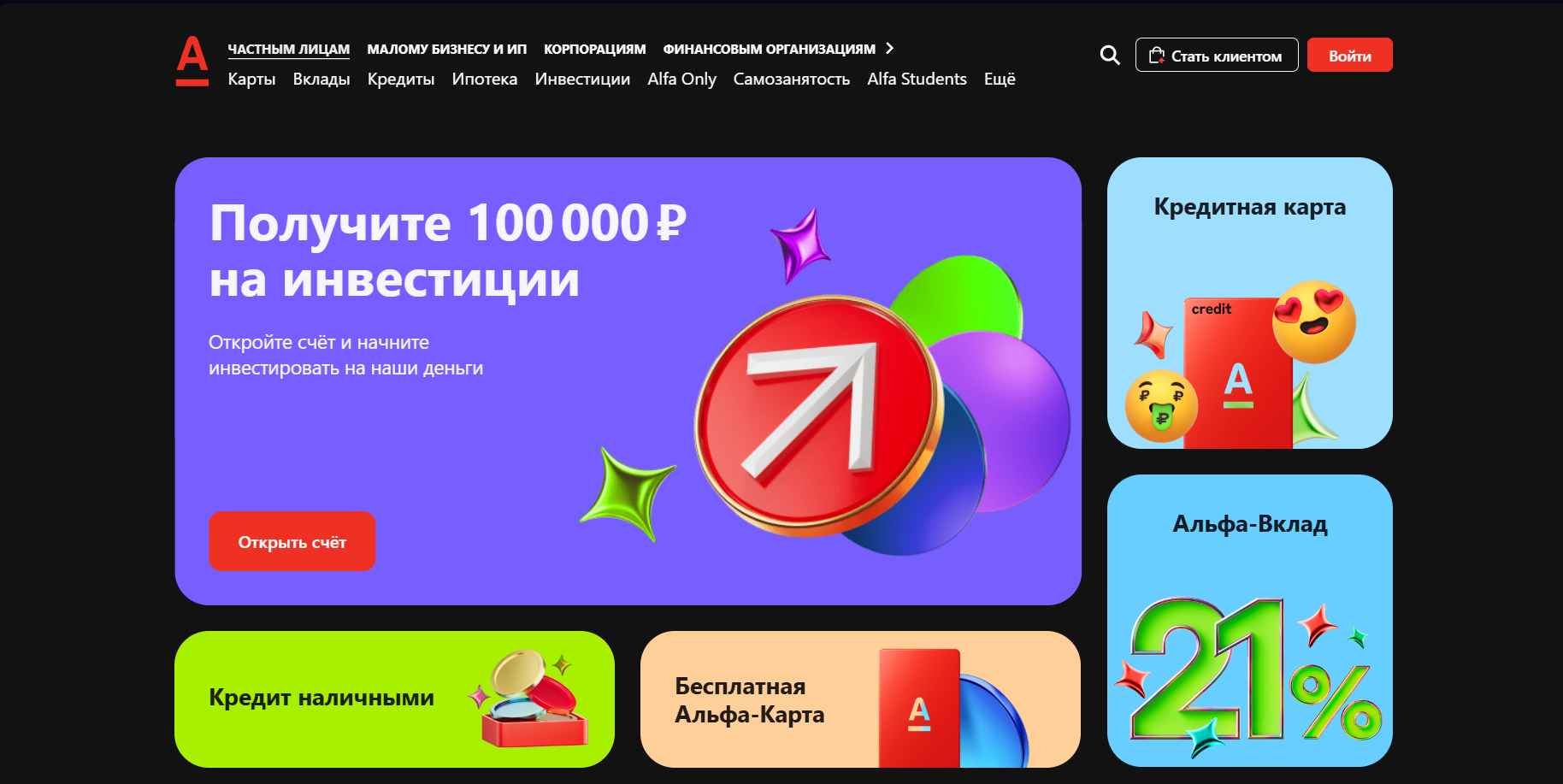


Рисунок 1.1 – Окно главного каталога сайта «Альфа Банк»

Сайт выполнен в стиле максимализма, что позволяет обратить внимание в первую очередь на глобальность проекта. При входе на страницу пользователь видит все, что необходимо стандартному пользователю: возможность оформить карту, кредит и инвестиции. Помимо этого в шапке сайта располагается навигационное меню с возможностью зарегистрировать пользователя в сеть сайта. Универсальность сайта заключается в обхватывании большого количества различных пользователей: от частных лиц до корпораций.

Далее необходимо рассмотреть подробнее один из разделов, чтобы определить наиболее важные данные, которые необходимо предоставить пользователю далее. Данная страница представлена на рисунке 1.2.

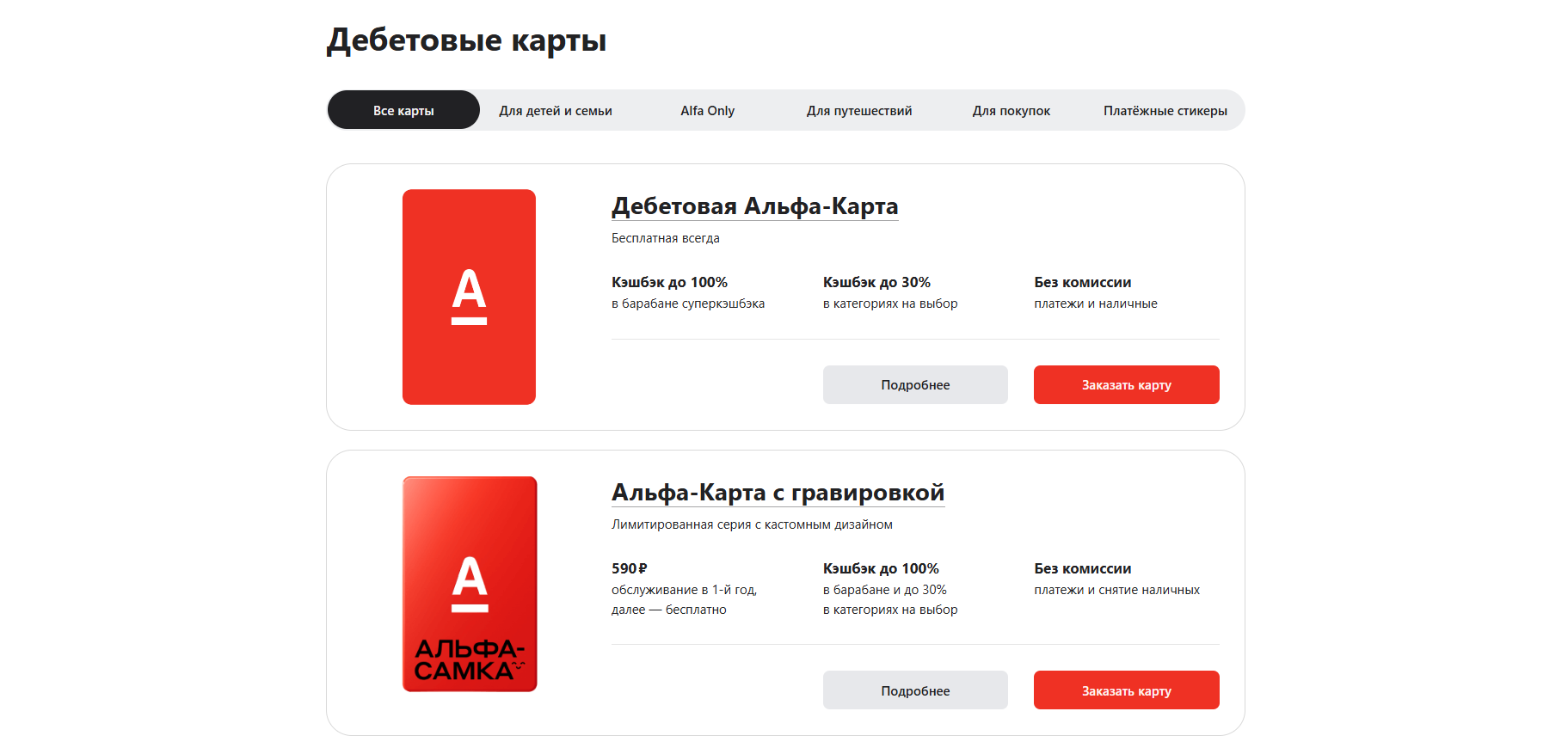


Рисунок 1.2 – Окно выбора карт «Альфа Банк»

Каталог представляет собой вертикальный список различных банковских карт, сверху которого располагается разделение на пользователей, которым необходима конкретная банковская карта в той или иной сфере. По умолчанию значение стоит на «Все карты», так как большинство пользователей не требуют конкретной задачи от банковской карты. Аналогично поэтому первые несколько вариантов являются самые выгодные и самые стандартные, подходящие для обычных пользователей. Минусов у данного дизайна я не обнаружил.

1.1.2 Аналог «Тинькофф Банк»

На рисунке 1.3 представлен внешний вид главного сайта.



Рисунок 1.3 – Внешний вид главного сайта «Тинькофф Банка»

Рассматриваемый вариант имеет схожесть с предыдущим, выполняя свой главный сайт в стиле максимализма, чтобы каждый пользователь при входе на сайт обратил внимание на нужные как для компании, так и для пользователя вещи: реклама Premium серии для выгодного пользования картой и реклама переноса номера для получения выгодного тарифа. В шапке сайта, как и в предыдущем варианте, располагается меню по всем необходимым для обычным пользователям запросам, таким как: оформление карт, кредитов, вкладов, инвестиций, страхования – и отдельный пункт для бизнес пользователей.

Далее рассмотрим один из разделов, коим будет самый популярный по посещаемости раздел – каталог оформления банковских карт. На рисунке 1.4 представлена данная страница.

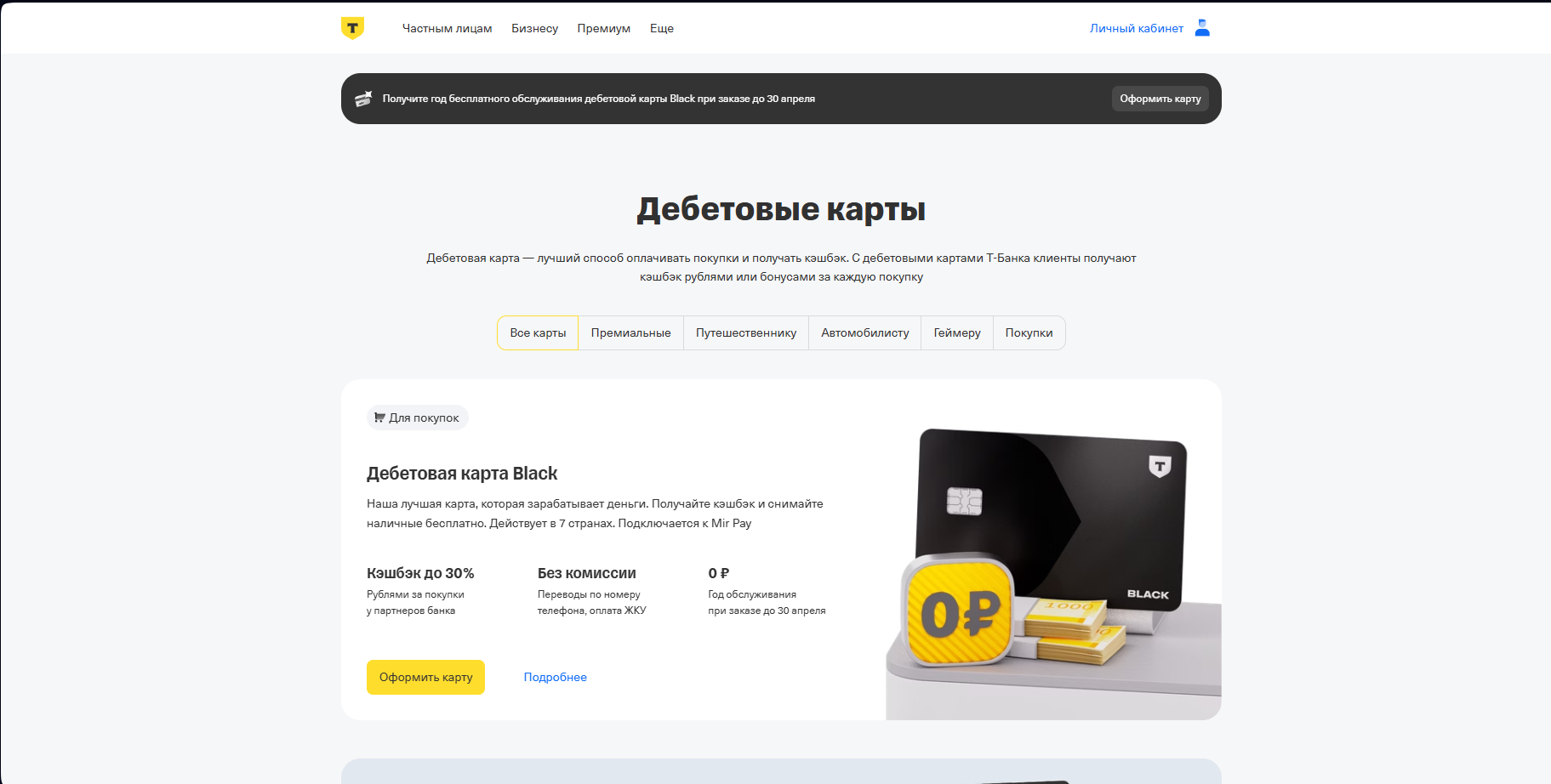


Рисунок 1.4 – Окно каталога оформления карт «Тинькофф Банка»

Стоит заметить, что каталог приближенно похож к тому варианту, что мы рассматривали в предыдущем пункте (рисунок 1.2). Здесь также имеются подразделы, где уже автоматически выбран пункт “Все карты”. В данном подразделе в первую очередь идет карта, которая рекламируется банком в первую очередь из-за выгодности сделки между пользователем и компанией. Ниже по убыванию популярности в использовании идут карты. Минусов у данного дизайна я также не обнаружил.

1.1.3 Аналог «Беларусбанк»

Главная страница сайта представлена на рисунке 1.5

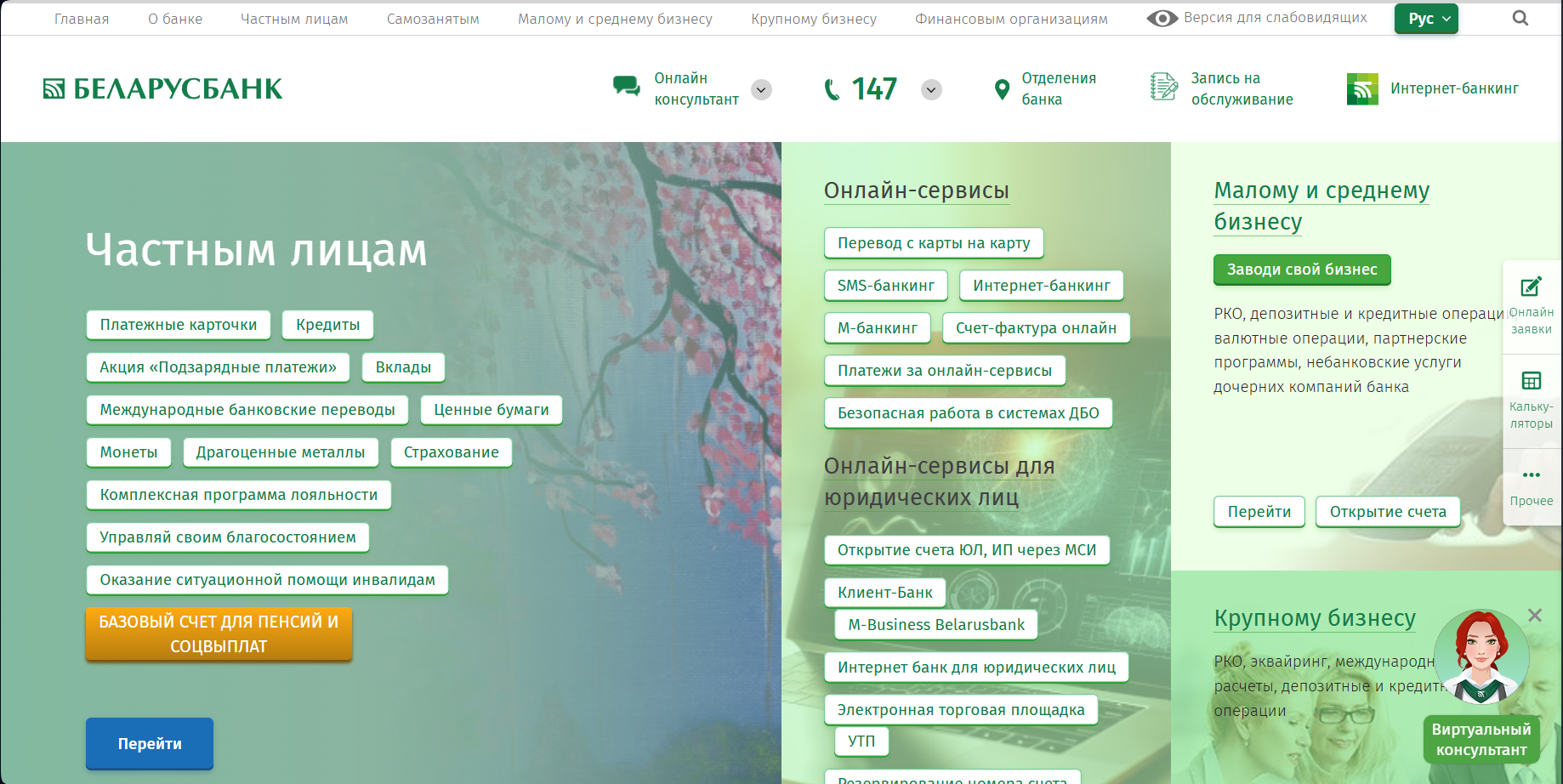


Рисунок 1.5 – Окно главного сайта «Беларусбанка»

Проблемой такого дизайна является огромное количество информации, “брошенной” в пользователя, в которой довольно сложно сориентироваться быстро и четко. Несмотря на это, шапка также имеет полезные навигационные ссылки. Разделы распложены в прямоугольниках, однако подразделы распложены неупорядоченно, что заставляет пользователя искать необходимый ему пункт.

Далее рассмотрим один из разделов, коим будет являться выбор платежных карт. Сайт такого раздела представлен на рисунке 1.6.



Рисунок 1.6 – Окно выбора карт «Беларусбанка»

Заметим, что в отличие от предыдущих вариантов данный сайт имеет боковую панель, занимающую значительную часть экрана, что изменяет положение основного контента, смещая его правее центра. Минусы у данного сайта очевидны, однако найти необходимую страницу не составит большого труда.

1.2 Техническое задание

Поставлена задача реализовать полноценный многостраничный веб-сайт на тему «Веб-сайтБанка». На сайт должна быть представлена информацию о банке, такую как новости банка, текущий курс валют, а также каталог оформления банковских карт.

Для привлечения потенциальных клиентов главная задача сделать удобный интерактивный сайт с подробной информацией о продукте и актуальной информацией.

Основными задачами сайта являются:

1. Предоставление навигационного меню;

2. Предоставление информации о текущем экономическом положении;

3. Предоставление информацию о банковских карт;

4. Предоставление возможности регистрации и/или подать заявку на желание работать в банке.

На главной странице меню будет располагаться справа сверху. В меню будут предоставляться ссылки на остальные страницы банка.

Веб-страница должна быть кроссбраузерной и адаптивной для мобильной версии, планшетного устройства.

1.3 Выбор средств реализации программного продукта

Курсовая работа выполняется в редакторе кода Visual Studio Code.

При создании веб-страниц были использованы языки: HTML, CSS, JS, XML.

HTML (HyperText Markup Language) — стандартный язык разметки гипертекстовых страниц в Интернете. Страницы успешно интерпретируются браузерами, которые отображают их на экранах различных электронных устройств в удобном для человека виде.

CSS – это фактически язык стилей, который определяет отображение HTML-документов. CSS работает со шрифтами, с цветами символов и фона, с полями, со строками, с высотой и с шириной элементов отображения, с фоновыми изображениями, с позиционированием элементов и со многим другим.

Если HTML необходим для структурирования содержания страницы, то CSS необходим для того, чтобы форматировать это структурированное содержание.

JavaScript это язык, который позволяет применять сложные вещи на web странице — каждый раз, когда на ней происходит что-то большее, чем просто её статичное отображение —без JavaScript не обошлось. Во все основные браузеры встроен интерпретатор JavaScript, именно поэтому они могут выполнять скрипты на странице. JavaScript можно использовать не только в браузере, это полноценный язык, программы на котором можно запускать и на сервере.

XML (eXtensible Markup Language) — это язык разметки, который используется для описания и передачи данных в удобочитаемом формате. XML позволяет определять пользовательские теги, что делает его более гибким, чем предыдущие языки разметки, такие как HTML. XML широко используется для обмена данными между приложениями и системами, включая веб-сервисы, базы данных и многие другие.

1.4 Вывод

В данном разделе были рассмотрены плюсы и минусы аналогичных решений на выбранную тему сайта для предотвращения ошибок в реализации веб-сайта. Были определены основные технические задания по созданию страницы. Также выбраны средства реализации программного продукта.

2. Проектирование веб-сайта

2.1 Выбор способа верстки

Требования проекта включают в себя адаптивность и кроссбраузерность. По этой причине была выбрана grid и flex-box вёрстки.

Grid-модель позволяет создавать структуры, необходимые для обеспечения отзывчивости сайтов на различных устройствах. Это означает, что сайт будет одинаково хорошо смотреться на компьютере, телефоне и планшете.

Flex – это новая технология, которая уже имеет достаточно широкую поддержку браузеров. Flexbox предоставляет инструменты для быстрого создания сложных, гибких макетов, и функции, которые были сложны в традиционных методах CSS.

2.2 Выбор стилевого оформления

Выбор стилевого оформления является важным этапом разработки проекта по нескольким причинам:

Создание уникального имиджа. Стилевое оформление проекта может помочь создать уникальный и запоминающийся имидж. Оно может помочь подчеркнуть уникальность проекта, выделить его на фоне конкурентов и привлечь больше внимания потенциальных пользователей.

Улучшение пользовательского опыта. Стилевое оформление проекта может влиять на пользовательский опыт. Четко продуманное и привлекательное оформление может сделать проект более удобным и привлекательным для пользователей, что может повысить их удовлетворенность использованием проекта.

Усиление бренда. Стилевое оформление может помочь усилить бренд проекта. Оно может помочь создать определенный образ и ассоциации, которые будут связаны с проектом. Это может помочь увеличить узнаваемость и лояльность пользователей.

Улучшение узнаваемости. Хорошо продуманное и привлекательное стилевое оформление может помочь увеличить узнаваемость проекта. Люди запоминают лучше визуальную информацию, поэтому стилевое оформление может помочь проекту стать более узнаваемым и запоминающимся.

Для акцентирования внимания клиента на блоках при оформлении веб-сайта был выбран максимализм.

Максимализм как стиль оформления банка имеет свои преимущества и может быть эффективным выбором по следующим причинам:

Фокус на основных блоках. Максимализм основан на привлечении внимания своими габаритами, что позволяет как пользователю найти нужную информацию довольно быстро, так и компании продвигать свои идеи более эффективно.

Стиль и элегантность. Максималистичный дизайн может создать ощущение простора и открытости, что может быть важным для привлечения пользователей и установления доверия к бренду.

В целом, стилевое оформление является важным аспектом разработки проекта, который может оказать значительное влияние на его успех и эффективность.

2.3 Выбор шрифтового оформления

Было выбрано разношрифтовое оформление текста на веб-странице, это имеет ряд плюсов:

Улучшение читаемости: разношрифтовое оформление текста может помочь повысить читабельность страницы, так как различные шрифты могут использоваться для различных элементов страницы, таких как заголовки, подзаголовки и основной текст. Это может помочь сделать текст более легким для чтения и более привлекательным для посетителей сайта.

Создание уникального дизайна: использование разных шрифтов может помочь создать уникальный дизайн страницы и помочь выделить ее среди других сайтов. Разнообразие шрифтов может помочь улучшить общий визуальный облик страницы и сделать ее более привлекательной для посетителей.

В целом, использование разношрифтового оформления текста может помочь улучшить читаемость, выделить важные элементы, создать уникальный дизайн и подчеркнуть тему и настроение страницы.

2.4 Разработка логотипа

Логотип – это фирменный знак, который люди ассоциируют с конкретным брендом. Логотип – это не абстрактный набор символов или просто красивое изображение.

Логотип сайта является важным элементом брендинга и визуальной идентичности. Вот несколько причин, почему логотип сайта является важным:

Узнаваемость бренда. Логотип является визуальным символом бренда и помогает создать узнаваемость среди потенциальных клиентов. Когда пользователи видят логотип, они могут легко связать его с брендом и лучше запомнить его.

Отражение имиджа компании. Логотип может отражать имидж компании и ее ценности. Например, если компания заботится об окружающей среде, ее логотип может содержать зеленый цвет и символ природы.

Отличие от конкурентов. Логотип может помочь бренду отличиться от конкурентов и выделиться на фоне других компаний.

Привлечение внимания. Логотип может быть ярким и привлекательным, что поможет привлечь внимание потенциальных клиентов.

Усиление брендовой лояльности. Логотип может помочь усилить брендовую лояльность клиентов. Когда клиенты видят логотип, связанный с их любимым брендом, они могут чувствовать более сильную привязанность к бренду.

Логотип выполнен под концепцию сайта в стиле svg. Минимум объектов показывает остроту намерений сайта, а также такой логотип легко запоминается. На рисунке 2.1 показан сам логотип.



Рисунок 2.1 – Логотип сайта «MoneyGo»

Логотип является частью фирменного стиля, который позволяет сформировать имидж или образ компании ([сайта](https://beseller.by/uslugi/sozdaniye-saytov.html) или [интернет-магазина](https://beseller.by/)), связать товары, услуги, рекламу именно с вашим бизнесом, выделить вас среди конкурентов.

2.5 Разработка пользовательских элементов

В проекте будут представлены элементы пользовательского интерфейса, с которыми пользователь сможет взаимодействовать: блоки с информацией о картах, навигационное меню, футер с основной информацией.

Навигация – ключевой критерий удобства сайта, своеобразная «карта местности». Представляет собой набор специальных приемов, методов и элементов, дающих возможность посетителям перемещаться между различными страницами ресурса. Навигационное меню представлено на рисунке 2.2.

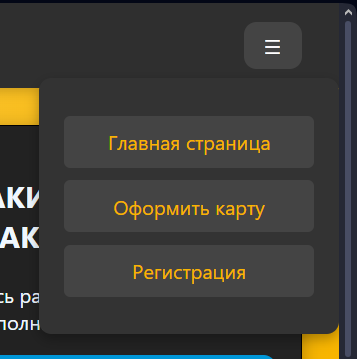


Рисунок 2.2 – Навигационное выпадающее меню сайта

Блоки, содержащие информацию о карточках, должны быть понятными и четко описывать данный продукт для быстрого понимания пользователем. Каталог сайта представлен на рисунке 2.3.

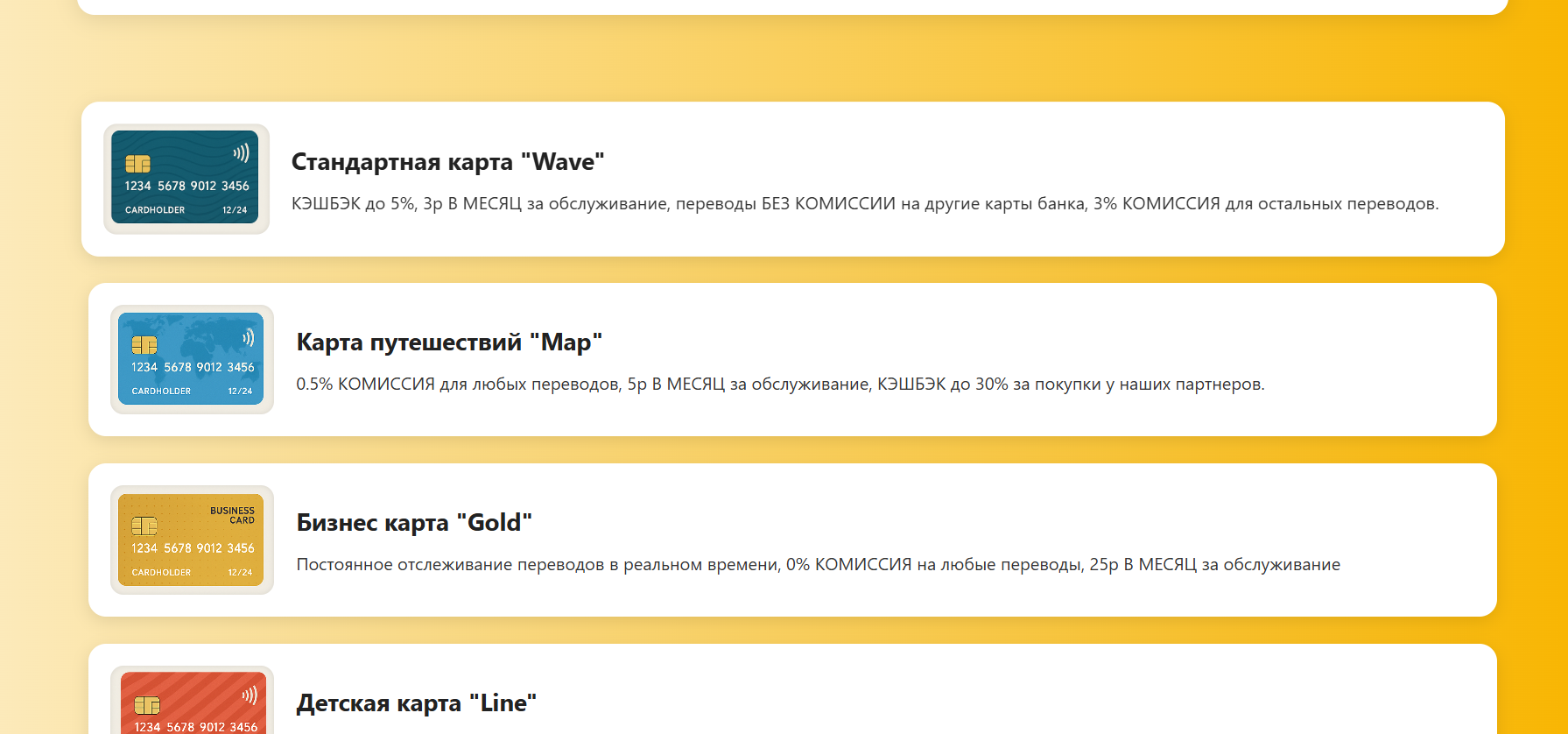


Рисунок 2.3 – Каталог оформления карт.

В данном каталоге представлен набор банковских карт, которые может заказать пользователь. Примечательно то, что в блоках представлена исключительно нужная пользователю информация: изображение карты, название карты и ее описание. При наведении на карту производится небольшая анимация увеличения блока для большей интерактивности пользователя с сайтом.

Подвал сайта должен содержать необходимую информацию о создателе, включая контактные данные, то не должен привлекать внимание пользователя заместо тела сайта.

2.6 Разработка спецэффектов

На сайте присутствуют динамические эффекты, которые придадут сайту интерактивность и сделают его более привлекательным для пользователя. Например: небольшое увеличение блока с карточкой представлен на рисунке 2.4.

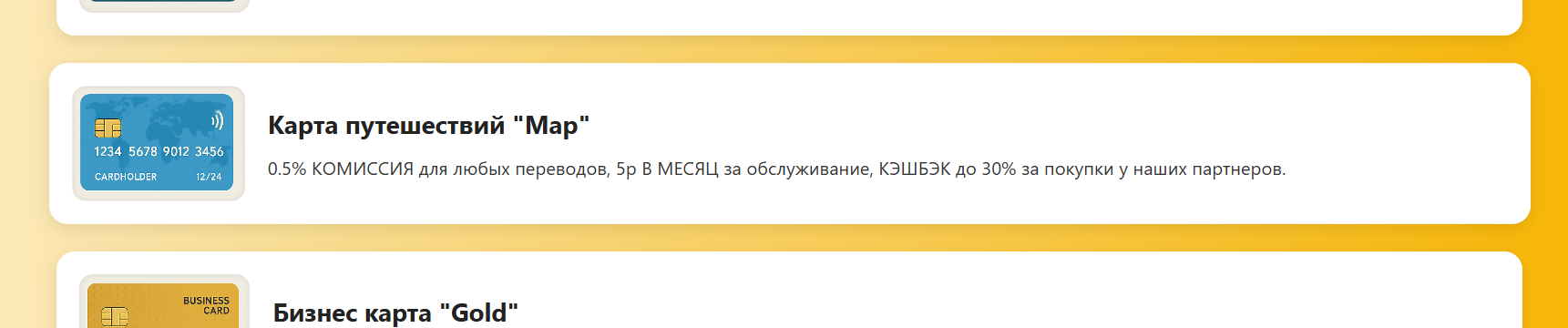


Рисунок 2.4 – Пример спецэффекта с карточками

На главном сайте присутствует калькулятор валют, который также как и выпадающее меню показывает результат в зависимости от текущего состояния калькулятора: нет ввода – нет результата(рисунок 2.5а); есть ввод – есть результат (рисунок 2.5б).

В том же калькуляторе наблюдается выбор направления конвертации (из BYN в валюту или из валюты в BYN) и выбор валюты (USD, EUR, RUB), представленные на рисунках 2.5в и 2.5г соответственно.

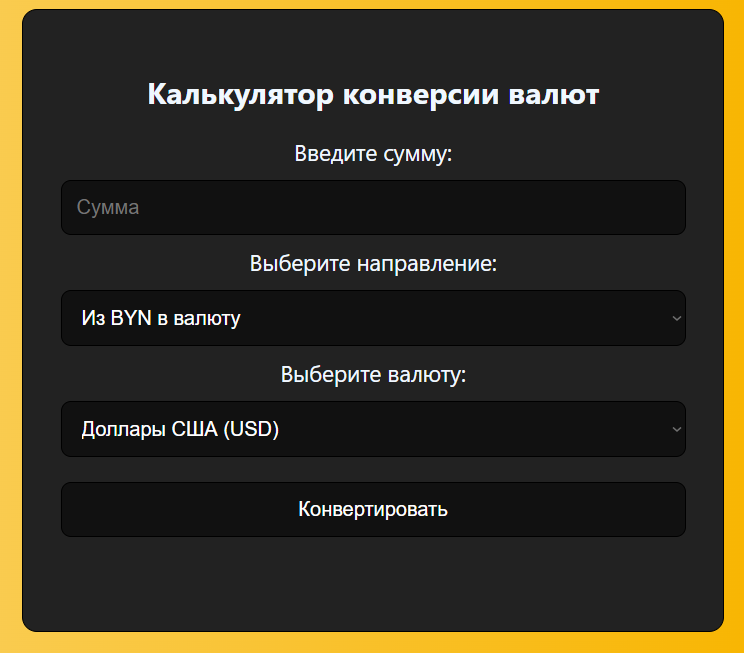


Рисунок 2.5а – Калькулятор без результата

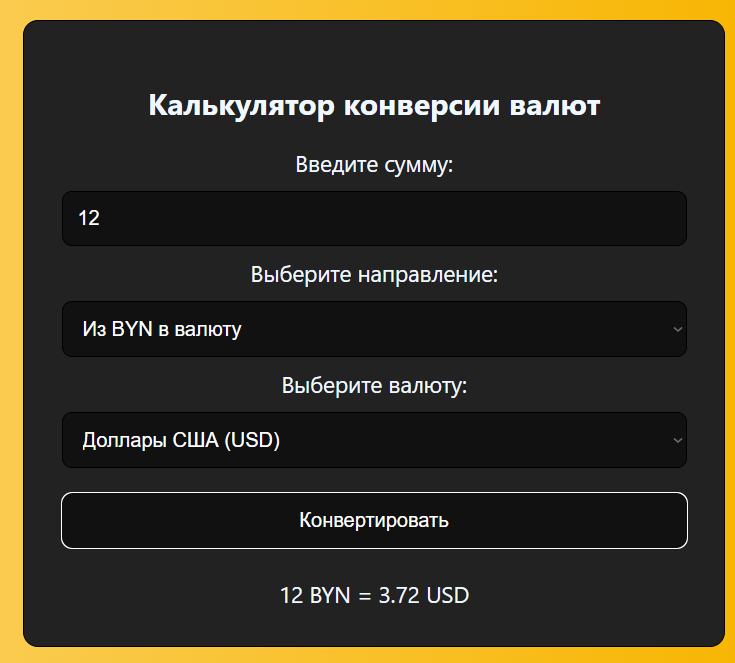


Рисунок 2.5б – Калькулятор с результатом

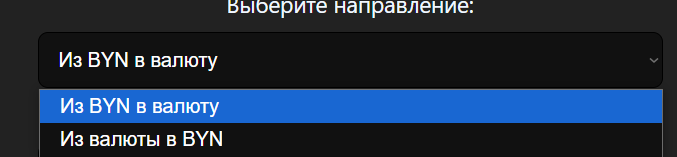


Рисунок 2.5в – Выбор направления на калькуляторе

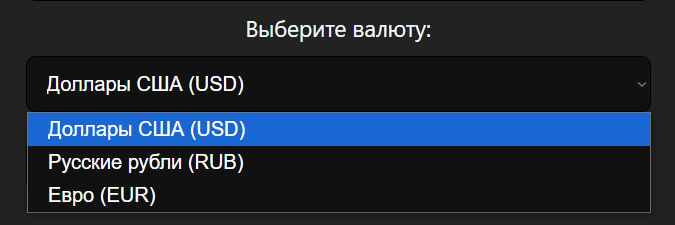


Рисунок 2.5г – Выбор валюты на калькуляторе

2.7 Выводы

В данном разделе был создан прототип веб-сайта, определены задачи по созданию содержания на веб-сайте. Выбрано единое цветовое и шрифтовое оформления сайта для удобства просмотра пользователем. Разработан дизайн и расположение на сайте пользовательских элементов, спецэффектов и анимации. Логотип выполнен под концепцию сайта.

3. Реализация структуры веб-сайта

3.1 Структура HTML-документа

Структура веб-сайта представляет собой организацию связанных между собой документов, которая позволяет пользователям легко перемещаться по страницам. При формировании структуры необходимо учитывать потребности посетителей, их запросы и интересы.

Хорошая структура HTML-документа имеет большое значение для эффективной работы сайта. Вот несколько причин, почему это так важно:

Улучшение доступности: хорошая структура HTML-документа может сделать сайт более доступным для пользователей с ограниченными возможностями. Например, правильное использование заголовков и других элементов разметки может помочь пользователям, использующим программы чтения с экрана, легче понимать контент.

Хорошая структура HTML-документа может помочь поисковым системам лучше понимать контент страницы. Это может привести к улучшению рейтинга сайта и его поисковой выдаче.

Улучшение скорости загрузки: хорошая структура HTML-документа может ускорить загрузку страницы. Например, правильное использование семантических тегов может помочь браузеру быстрее отрисовать страницу.

Улучшение совместимости: хорошая структура HTML-документа может помочь сделать сайт более совместимым с различными браузерами и устройствами. Это может улучшить пользовательский опыт и увеличить количество посетителей на сайте.

Тег <head> предназначен для хранения служебных элементов, которые не отображаются на странице, за исключением заголовка <title>.

Основное содержимое сайта размещается внутри тега <body>, который включает в себя семантические теги, такие как header, section и footer, которые помогают программам понимать тип информации, содержащейся на странице.

Структура HTML главной страницы представлена в листинге 3.1.

|  |
| --- |
| < ! DOCTYPE html>  <html>  <head>…</head>  <body>  <header class=”head”>…</header>  <section class=”links-section”>…</section>  <section class=”text-cards”>…</section>  <section class=”calculator”>…</section>  <section class=”media-block”>…</section>  <footer class=”footer”>…</footer>  </body>  </html> |

Листинг 3.1 – Структура HTML главной страницы

С помощью тега <header> была создана шапка сайта для всех страниц и создано навигационное меню. “Подвал” страницы был, идентичный на всех страницах сайта, был реализован с помощью тега <footer>. Теги <section> были созданы для размещения на страницы и разметки основной информации.

3.2 Добавление таблиц стилей Scss и CSS

В ходе разработки проекта были применены различные подходы к подключению таблиц стилей, включая внешнее и строковое подключение.

|  |
| --- |
| <head>  <title>Главная страница</head>  <link rel=”stylesheet” href=”../CSS\_SCSS/Main.css”>  </head> |

Листинг 3.2 – Пример внешнего подключения

Для определения стилей для повторяющихся элементов и основной разметки страниц было использовано внешнее подключение, при котором стили размещаются в отдельном файле и могут быть использованы на любой странице сайта. Для небольших изменений в стиле текста на страницах сайта было использовано строковое подключение. Все таблицы стилей были организованы с использованием селекторов по классу, что позволило улучшить структуру кода и обеспечить более эффективную поддержку сайта в будущем. Такой подход позволяет быстро и легко вносить изменения в стили сайта, облегчает сопровождение проекта и ускоряет его загрузку.

|  |
| --- |
| <div class="img-placeholder" style="background-image:url(../Images/typing\_edited.png);></div> |

Листинг 3.3 – Пример внутреннего подключения

3.3 Использование стандартов XML (SVG)

Для создания некоторых картинок было принято использование SVG-формата. Этот формат позволяет картинкам сохранять свое качество, несмотря на размер экрана. Такой формат применялся для создания логотипа (листинг 3.4а) и названия (листинг 3.4б) компании.

SVG-формат удобен для использования веб-дизайнерами и разработчиками по нескольким причинам:

Масштабируемость: SVG-файлы могут быть масштабированы без потери качества изображения.

Малый размер: SVG-файлы обычно имеют меньший размер, чем другие форматы изображений, такие как JPG или PNG, что уменьшает время загрузки страницы и повышает производительность сайта.

|  |
| --- |
| <svg width="160" height="120" viewBox="0 0 160 120" xmlns="http://www.w3.org/2000/svg">    <path d="M60 60             L30 50             L45 48             L25 40             L42 38             L22 30             L50 35             Z"          fill="#ecf0f1" stroke="#bdc3c7" stroke-width="2"/>    <path d="M100 60             L130 50             L115 48             L135 40             L118 38             L138 30             L110 35             Z"          fill="#ecf0f1" stroke="#bdc3c7" stroke-width="2"/>    <circle cx="80" cy="60" r="20" fill="#f1c40f" stroke="#f39c12" stroke-width="3"/>    <circle cx="72" cy="52" r="5" fill="#fff" opacity="0.4"/>  </svg> |

Листинг 3.4а – XML код логотипа компании

|  |
| --- |
| <svg width="300" height="100" viewBox="0 0 300 100" xmlns="http://www.w3.org/2000/svg">  <defs>  <filter id="shadow" x="-50%" y="-50%" width="200%" height="200%">  <feDropShadow dx="2" dy="2" stdDeviation="2" flood-color="gray" flood-opacity="0.5"/>  </filter>  </defs>  <text x="50%" y="50%" dominant-baseline="middle" text-anchor="middle"  font-family="Arial, sans-serif" font-size="36" fill="#FFD700" font-weight="bold"  filter="url(#shadow)" letter-spacing="2px">  MoneyGo  </text>  </svg> |

Листинг 3.4б – XML код названия компании

Данные для связи удобно хранятся отдельно в XML-документе.

XML-формат удобен для хранения данных по нескольким причинам:

Независимость от программного обеспечения: XML-формат не зависит от программного обеспечения или операционной системы, что делает его переносимым между различными платформами.

Возможность расширения: XML-формат позволяет создавать собственные теги и атрибуты, что дает возможность расширять его для конкретных нужд проекта.

3.4 Использование JavaScript

JavaScript (JS) является одним из самых распространенных языков программирования для веб-разработки. Этот язык позволяет добавлять динамичность и интерактивность на веб-страницы, что делает пользовательский опыт более увлекательным и позволяет создавать более сложные веб-приложения.

В данном сайте использовался калькулятор для конвертации валют из одной в другую, код для которого был написан на JS, код которого изображен на листинге 3.5.

|  |
| --- |
| const rates = {      usd: 0.31,      rub: 27.34,      eur: 0.27    };    function convertCurrency() {      const amount = parseFloat(document.getElementById('amount').value);      const currency = document.getElementById('currency').value;      const customRate = parseFloat(document.getElementById('custom-rate').value);      let result;        if (isNaN(amount) || amount <= 0) {        document.getElementById('result').innerHTML = "Пожалуйста, введите корректную сумму.";        return;      }        if (currency === 'custom' && isNaN(customRate)) {        document.getElementById('result').innerHTML = "Пожалуйста, введите корректный коэффициент.";        return;      }      if (document.getElementById('direction').value === "to") {        switch (currency) {          case 'usd':            result = amount \* rates.usd;            document.getElementById('result').innerHTML = `${amount} BYN = ${result.toFixed(2)} USD`;            break;          case 'rub':            result = amount \* rates.rub;            document.getElementById('result').innerHTML = `${amount} BYN = ${result.toFixed(2)} RUB`;            break;          case 'eur':            result = amount \* rates.eur;            document.getElementById('result').innerHTML = `${amount} BYN = ${result.toFixed(2)} EUR`;            break;          case 'custom':            result = amount \* customRate;            document.getElementById('result').innerHTML = `${amount} BYN = ${result.toFixed(2)} (по вашему коэффициенту)`;            break;        }      } else {        switch (currency) {          case 'usd':            result = amount / rates.usd;            document.getElementById('result').innerHTML = `${amount} USD = ${result.toFixed(2)} BYN`;            break;          case 'rub':            result = amount / rates.rub;            document.getElementById('result').innerHTML = `${amount} RUB = ${result.toFixed(2)} BYN`;            break;          case 'eur':            result = amount / rates.eur;            document.getElementById('result').innerHTML = `${amount} EUR = ${result.toFixed(2)} BYN`;            break;          case 'custom':            result = amount / customRate;            document.getElementById('result').innerHTML = `${amount} (по вашему коэффициенту) = ${result.toFixed(2)} BYN`;            break;        }      }    }    document.getElementById('direction').addEventListener('change', function () {      if (this.value === 'to') {        document.getElementById('custom-rate-container').style.display = 'none';      } else {        document.getElementById('custom-rate-container').style.display = 'block';      }    }); |

Листинг 3.5 – Пример использования JS

JavaScript может использоваться для создания различных интерактивных элементов на странице, таких как выпадающие меню, слайдеры, анимации, формы обратной связи и многое другое. С помощью JS можно изменять содержимое, стиль и расположение элементов на странице, а также обрабатывать действия пользователя, такие как клики и ввод текста.

3.5 Выводы

На данном этапе была реализована структура на HTML, а также созданы таблицы стилей Scss/CSS. Было продемонстрировано использование стандартов XML (SVG), а также языка программирования JS. Было определено стилевое оформление веб-сайта, созданы анимации, а также веб-сайт был подготовлен к следующему этапу – к тестированию.

4 Тестирование веб-сайта

4.1 Адаптивный веб-сайта

Для достижения адаптивности в проекте применялись методы верстки grid и flex, а также использовались медиа-запросы. Ресурс должен одинаково хорошо демонстрироваться на экране компьютера (рис. 4.1), планшете(рис. 4.2), смартфоне(рис. 4.3).



Рисунок 4.1 – Вид страницы для компьютера



Рисунок 4.2 – Вид страницы для планшетных устройств

Адаптивность является очень важным аспектом веб-разработки, так как позволяет сайту корректно отображаться на различных устройствах с разными размерами экранов. С увеличением числа устройств с различными размерами экранов, таких как смартфоны, планшеты, ноутбуки и настольные компьютеры, становится критически важным, чтобы сайт был доступен и удобочитаем на всех устройствах.

Адаптивность позволяет автоматически изменять расположение и размер элементов страницы в зависимости от размера экрана устройства, на котором происходит просмотр сайта. Таким образом, пользователь получает оптимальное визуальное восприятие сайта на любом устройстве и не испытывает неудобств при просмотре страницы.

Кроме того, адаптивность сайта имеет важное значение для улучшения SEO-оптимизации, так как поисковые системы высоко оценивают мобильную доступность сайтов и ранжируют их в соответствии с этим фактором. Адаптивный сайт также может увеличить время пребывания пользователей на сайте и увеличить вероятность выполнения целевых действий (например, покупки товара или заполнения формы заказа).



Рисунок 4.3 – Вид страницы для мобильных устройств

Для этого создается адаптивный дизайн и используется адаптивная верстка. В итоге сайт отлично позиционируется на всех основных типах устройств, что гарантирует полноценный охват аудитории.

4.2 Кроссбраузерность веб-сайта

Кроссбраузерность – важный критерий корректной работы веб-ресурса. Это инструмент поддержания посетительской лояльности, повышения конверсии и поднятия рейтинга в поисковых системах.

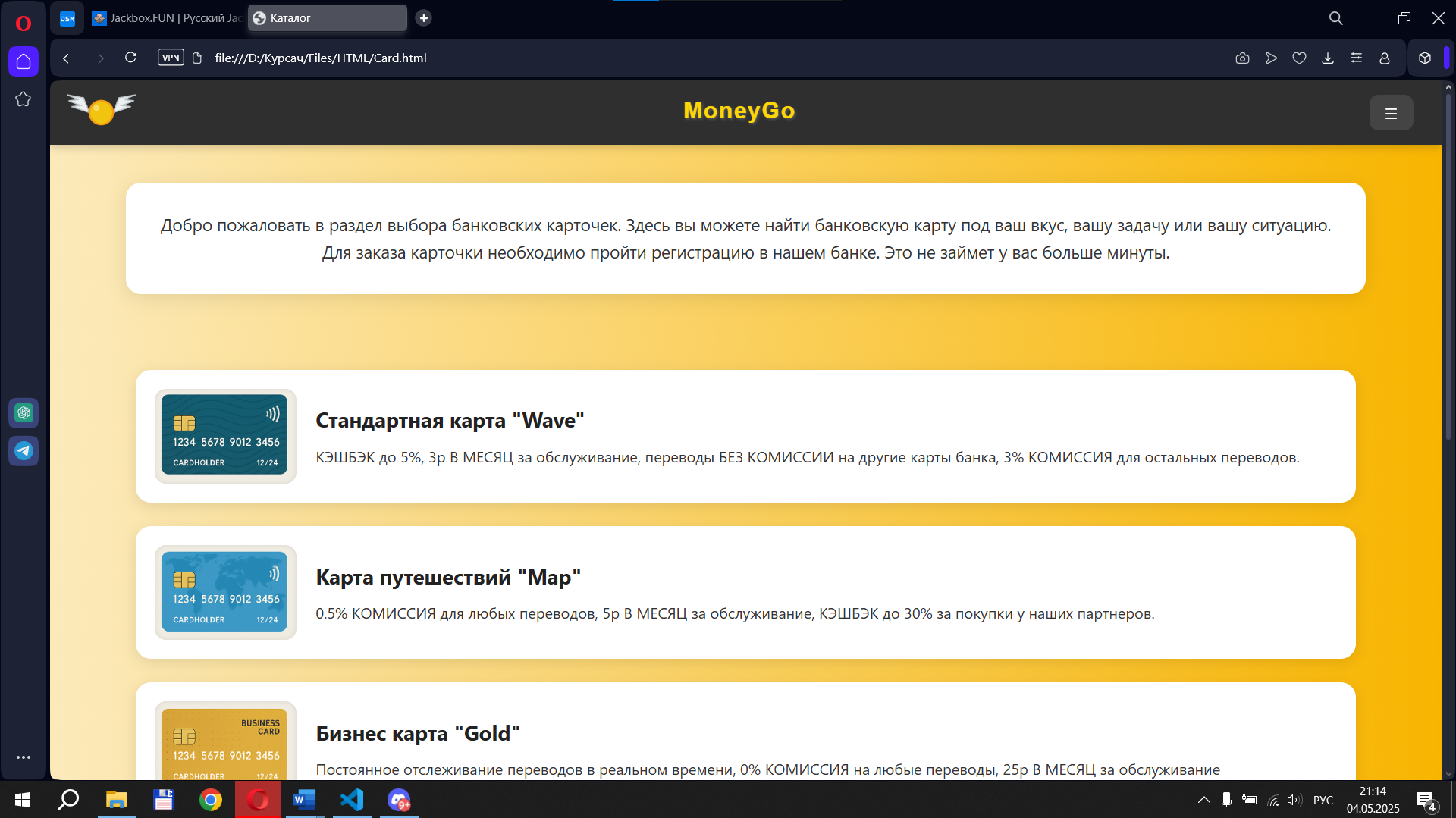


Рисунок 4.4 - Вид страницы в браузере Opera

После написания основной структуры страницы на HTML и внешнего стилевого оформления появился вопрос о тестировании. Веб-сайт был открыт при помощи различных браузеров. Существует множество браузеров, каждый из которых имеет свои особенности и способы интерпретации кода. Кроссбраузерность гарантирует, что ваш сайт будет работать одинаково хорошо во всех этих браузерах. Пользователи могут использовать разные браузеры в зависимости от своих предпочтений, операционной системы или устройства. Кроссбраузерность обеспечивает лучший пользовательский опыт для всех, независимо от выбора браузера. Более того, кроссбраузерное тестирование позволяет разработчикам уверенно выпускать обновления, зная, что функциональность сайта останется стабильной. Это также способствует улучшению общего качества кода, поскольку требует строгого соблюдения стандартов веб-разработки. В связи с этим, тестирование на кроссбраузерность становится неотъемлемой частью процесса разработки, помогая обнаружить и исправить возможные проблемы совместимости. Таким образом, кроссбраузерность является ключевым фактором при разработке веб-сайтов, чтобы обеспечить широкую доступность и функциональность для всех пользователей.

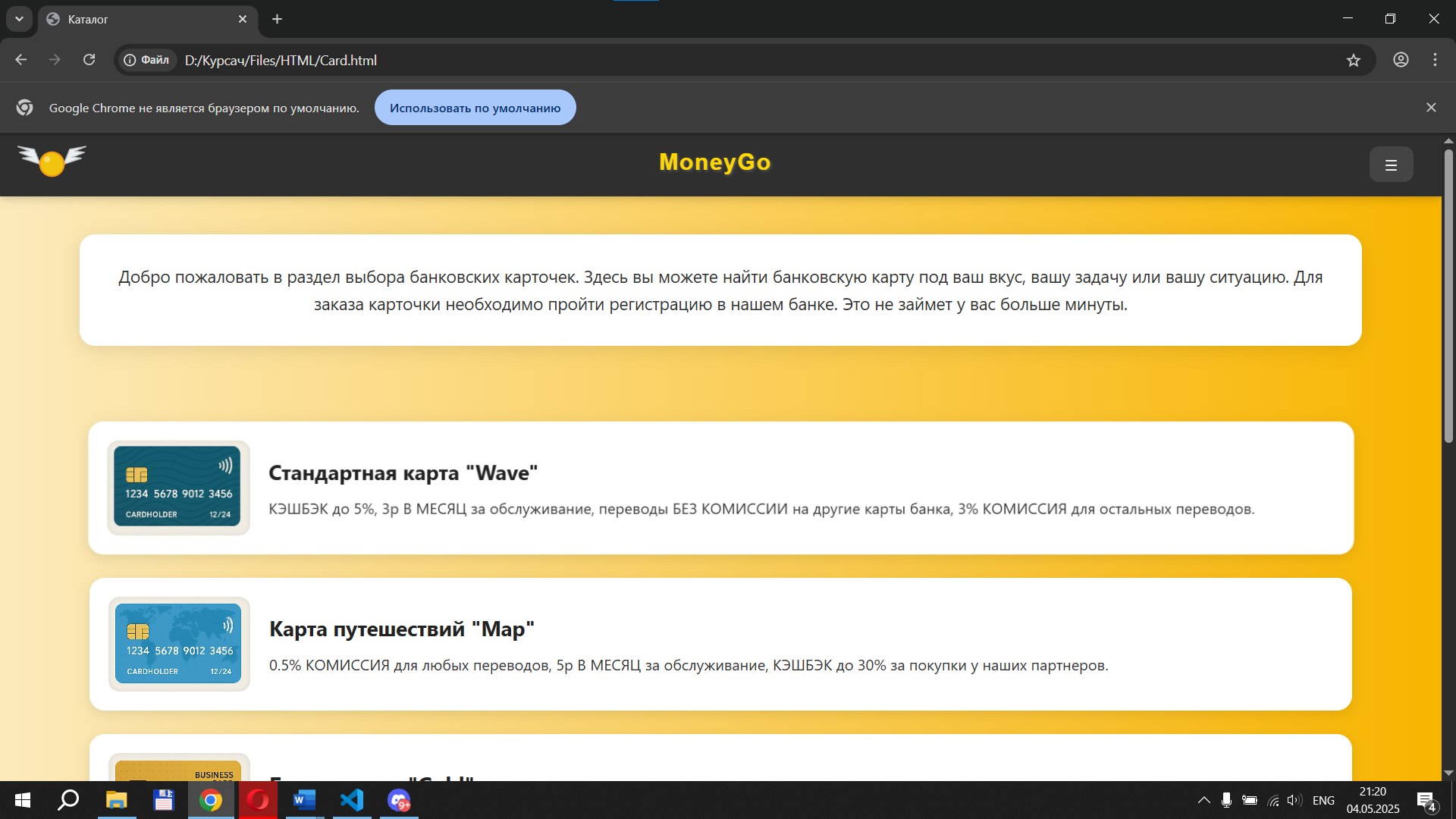


Рисунок 4.5 – Вид страницы в браузере Chrome

Сайт, который выглядит и работает хорошо во всех браузерах, создает профессиональное впечатление и демонстрирует заботу о пользователях.

4.3 Руководство пользователя

Интерфейс сайта интуитивно понятен пользователю. Основное взаимодействие с сайтом в поиске информации удобнее всего осуществлять с помощью грамотного расставления блоков с информацией, необходимой для пользователя в первую очередь. Например, на главном сайте (рис 4.6) располагается ссылки на внешние страницы, необходимые от компании стандартному пользователю.

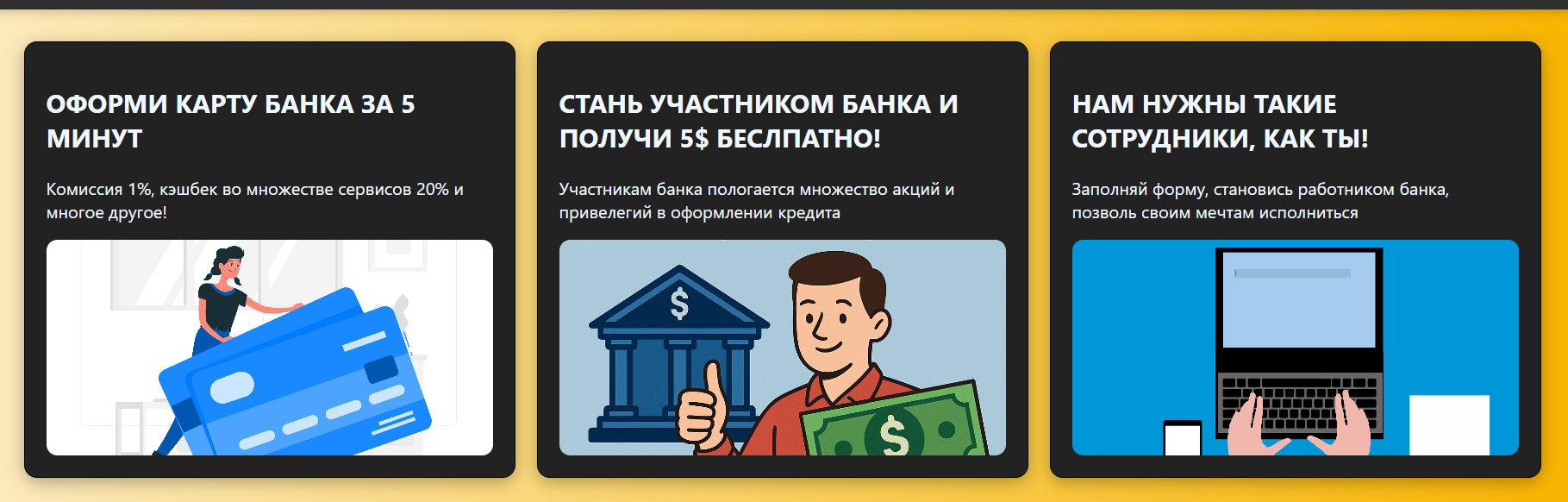


Рисунок 4.6 – Ссылки на внешние страницы на главной странице

Таким образом, был разработан сайт с целью обеспечить максимально простое и интуитивно понятное использование.

4.4 Тестирование кода

Тестирование кода веб-страниц является критически важной частью процесса веб-разработки. Это процесс проверки программного кода, чтобы убедиться, что он работает так, как должен работать, и что он соответствует определенным стандартам и требованиям.

Тестирование позволяет убедиться, что код работает правильно и соответствует заданным требованиям. Это помогает обеспечить высокое качество и надежность сайта. Тестирование помогает обнаружить уязвимости и потенциальные угрозы безопасности, что позволяет разработчикам принимать меры для их устранения до того, как они могут быть использованы злоумышленниками. Ошибка в коде, которая обнаруживается на ранней стадии, может быть исправлена быстрее и с меньшими затратами, чем если бы она была обнаружена позже в процессе разработки или после выпуска сайта. Правильно работающий сайт, который не вызывает ошибок и проблем, улучшает опыт пользователя, что может привести к большей лояльности и повторному использованию сайта.

Весь код был протестирован с помощью онлайн-сервиса The W3C Markup Validation Service. Результат представлен на рисунке ниже.



Рисунок 4.7 – Результат тестирования кода с помощью The W3C Markup Validation Service.

В целом, тестирование кода веб-страниц является неотъемлемой частью процесса веб-разработки и помогает гарантировать высокое качество, безопасность и удобство использования сайта.

4.5 Выводы

В данном разделе было рассмотрено, как сайт ведет в себя на разных устройствах и в разных браузерах. С помощью тестирования были выявлена некоторые недочеты для дальнейшего усовершенствования продукта.

Заключение

В ходе проделанной работы был создан полноценный сайт магазина для онлайн-заказа мелкой бытовой техники.

При разработке данного веб-сайта был использован широкий ряд языков разметки и таблиц стиля, который позволил увеличить функциональность и многократно повысил качество конечного продукта. Была использована адаптивная верстка для корректного отображения на всех устройствах.

В ходе выполнения данной задачи были рассмотрены плюсы и минусы аналогичных решений. В процессе осмотра выяснилось, каких тенденции в разработке дизайна веб-сайта стоит придерживаться.

При формировании технического задания были продумано содержание основных страниц веб-сайта.

Были выбраны инструменты для разработки веб-сайта.

При проектировании были разработаны прототипы, а далее созданы макеты дизайна страниц веб-сайта, при опоре на который был реализован сам веб-сайт. Для разработки прототипов и макетов станиц было использована программа Figma, ранее не изученная в ходе учебного процесса.

Были разработаны пользовательские элементы, спецэффекты и логотип, для идентификации веб-сайта, среди миллионов других интернет ресурсов.

Был создан XML-документ для последующей вставки на веб-страницу HTML.

В ходе тестирования была продемонстрирована валидность, кроссбраузерность и адаптивность веб-сайта.

Таким образом была полностью достигнута поставленная цель по разработке веб-сайта и соблюдены все требования.

Список использованных источников

1. Аналог «Альфа Банк» [Электронный ресурс]. — Режим доступа: https://www.alfabank.by
2. Аналог «Тинькофф Банк» [Электронный ресурс]. — Режим доступа: https://www.tbank.ru
3. Аналог «Беларусбанк» [Электронный ресурс]. — Режим доступа: https:// https://belarusbank.by
4. Документация по Sass/SCSS [Электронный ресурс]. — Режим доступа: https://sass-scss.ru/guide/
5. Документация по JavaScript/HTML [Электронный ресурс]. — Режим доступа: https://developer.mozilla.org
6. Введение в кросс-браузерное тестирование [Электронный ресурс]. — Режим доступа:

https://developer.mozilla.org/ru/docs/Learn/Tools\_and\_testing/Cross\_browser\_testing/Introduction

1. Репозиторий проекта на GitHub [Электронный ресурс]. — Режим доступа: https://github.com/danila175