**АЗЕРБАЙДЖАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ НЕФТИ И ПРОМЫШЛЕННОСТИ**

**Факультет: Информационные технологии и управление**

**Кафедра: Компьютерная инженерия**

**Предмет: «Веб-системы и технологии»**

Курсовая работа

**Тема** «Разработка онлайн-системы калькулятора обменных курсов»

**Группа:** 680.22

**Курc:** 3

**Специальность: 050616** Информационные технологии

**Студент**: Гадирова Эльвина

**Руководитель:** **препод.Халилов М. Э.**

**Зав.кафедрой: доц.Рагимова Н.А.**

**Баку – 2025**

**АЗЕРБАЙДЖАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ НЕФТИ И ПРОМЫШЛЕННОСТИ**

**АЗЕРБАЙДЖАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ НЕФТИ И ПРОМЫШЛЕННОСТИ**

**ЗАДАНИЕ ПО КУРСОВОЙ РАБОТЕ**

**ФАКУЛЬТЕТ**: Информационные технологии и управление

**КАФЕДРА**: Компьютерная инженерия

**Группа** 680.22 **курс** III

**Специальность** **050616** Информационные технологии

**Студент** Гадирова Эльвина

**Зачетная книжка**

**Руководитель курсовой работы**  **препод. Халилов М. Э.**

**Срок выдачи**

**Дата сдачи**

**Тема курсовой работы** «Разработка онлайн-системы калькулятора

обменных курсов »

**Отзыв руководителя курсовой**:

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

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

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

**Подпись студента**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Подпись рук.курсовой**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Подпись зав.кафедрой**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Дата защиты курсовой**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**Оценка**\_\_\_\_\_\_\_\_\_\_

**Пред.комиссии**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

**Члены комиссии**: 1.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

2.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

3.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

**Содержание**

**[ВВЕДЕНИЕ 4](#_Toc14271)**

**[1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 6](#_Toc25466)**

**[1.1.Понятие Web-сайта 6](#_Toc27350)**

**[1.1.1. Классификация Web-сайта 7](#_Toc17866)**

**[1.1.2. Web-программирование. 9](#_Toc28606)**

**[1.2. Основы языка HTML 9](#_Toc22435)**

**[1.3.Технология каскадных таблиц стилей CSS 14](#_Toc27683)**

**[1.4. Основы JavaScript 16](#_Toc4554)**

**[1.4.1.Основные элементы языка JavaScript 17](#_Toc24658)**

**[2. ПРАКТИЧЕСКАЯ ЧАСТЬ 18](#_Toc15212)**

**[2.1 Описание предметной области 18](#_Toc1659)**

**[Архитектура онлайн-системы 18](#_Toc16106)**

**[2.2 Установка и настройка компонентов проекта 20](#_Toc31927)**

**[3. РЕЗУЛЬТАТ 39](#_Toc12847)**

**[4. СПИСОК ЛИТЕРАТУРЫ 40](#_Toc6300)**

**[5. ПРИЛОЖЕНИЯ 41](#_Toc5766)**

# **ВВЕДЕНИЕ**

В условиях глобализации и активного развития международных экономических отношений обменные курсы валют играют ключевую роль в финансовой деятельности компаний и частных лиц. Колебания валютных курсов оказывают значительное влияние на стоимость товаров, услуг, а также на инвестиционные решения. В связи с этим возникает необходимость в инструментах, позволяющих оперативно и точно рассчитывать обменные курсы.​

Современные технологии предоставляют возможность создания онлайн-систем, способных автоматически обновлять данные о курсах валют и производить необходимые расчеты. Разработка такого калькулятора обменных курсов требует интеграции с надежными источниками финансовой информации и обеспечения удобного пользовательского интерфейса.​

Цель данной курсовой работы заключается в разработке онлайн-системы калькулятора обменных курсов, которая обеспечит пользователям быстрый и точный доступ к актуальной информации о валютных курсах и позволит выполнять необходимые расчеты в реальном времени.

Разработка онлайн-системы калькулятора обменных курсов будет включать в себя следующие этапы:

1. **Анализ требований и постановка задачи:** На этом этапе определяется целевая аудитория системы, формулируются основные функциональные требования: поддержка актуальных обменных курсов, возможность конвертации различных валют, удобный и интуитивно понятный интерфейс.

2. **Проектирование архитектуры системы:** Создается архитектура приложения, включая определение клиентской и серверной частей, баз данных, методов интеграции с внешними API для получения курсов валют.

3. Разработка: Это предполагает создание веб-сайта с использованием HTML, CSS, JavaScript и других веб-технологий.

4. **Тестирование системы:** Проведение функционального, нагрузочного и пользовательского тестирования для выявления и устранения ошибок.

WEB-сайт системы калькулятора обменных курсов– Это система, ориентированная в первую очередь на пользователя, которому требуется быстро и удобно произвести конвертацию валют. Она должна предоставлять сервисы актуального отображения курсов валют, легкость в выборе валютной пары и простоту выполнения расчета.  
 Таким образом, целью моей курсовой работы является создание веб-сайта калькулятора обменных валют, обеспечивающего пользователю удобство, надежность и интуитивно понятный интерфейс для выполнения валютных операций.

# **ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

## **1.1.Понятие Web-сайта**

Web-сайт – это информация, представленная в определенном виде, которая располагается на Web-сервере и имеет свое имя (адрес). Для просмотра Web-сайтов на компьютере пользователя применяются специальные программы, которые называются браузерами.

Наиболее распространенными браузерами в настоящее время являются Microsoft Edge и Google Chrome. В зависимости от того, какое имя (адрес) сайта мы зададим в строке «Адрес», браузер будет загружать в свое окно соответствующую информацию.

Web-сайт состоит из связанных между собой Web-страниц. Web-страница представляет собой текстовый файл с расширением \*.htm, который содержит текстовую информацию и специальные команды – HTML-коды, определяющие, в каком виде эта информация будет отображаться в окне браузера. Вся графическая, аудио- и видеоинформация непосредственно в Web-страницу не входит и представляет собой отдельные файлы с расширениями \*.gif, \*.jpg (графика), \*.mid, \*.mp3 (звук), \*.avi (видео). В HTML-коде страницы содержатся только указания на такие файлы (рис. 1.1).

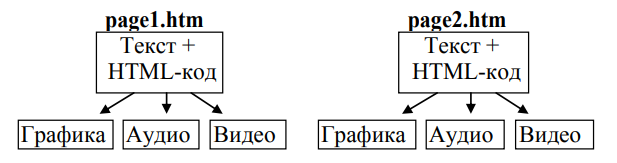


Рис.1.1. Структура Web-сайта

Каждая страница Web-сайта также имеет свой интернет адрес, который состоит из адреса сайта и имени файла, соответствующего данной странице. Таким образом, Web-сайт – это информационный ресурс, состоящий из связанных между собой гипертекстовых документов (Web-страниц), размещенный на Web-сервере и имеющий индивидуальный адрес.

### Классификация Web-сайта

В настоящее время во всемирной паутине размещено несколько миллионов Web-сайтов и их число постоянно растет.

1. **По доступности сервисов:**

* Открытые – все сервисы полностью доступны для любых посетителей. 
* Полуоткрытые – для доступа необходимо зарегистрироваться (обычно бесплатно). 
* Закрытые – полностью закрытые служебные сайты организаций, личные сайты частных лиц. Такие сайты доступны для узкого круга людей. Доступ новым людям даѐтся через инвайты (приглашения).

1. **По природе содержимого:**

* Статические – всѐ содержимое заранее подготавливается. Пользователю выдаются файлы в том виде, в котором они хранятся на сервере. 
* Динамические – содержимое генерируется специальными скриптами (программами) на основе других данных из любого источника.

1. **По физическому расположению:**

* Внешние сайты сети Интернет. 
* Локальные сайты – доступны только в пределах локальной сети. Это могут как корпоративные сайты организаций, так как и сайты частных лиц в локальной сети провайдера.

1. **По отношению к посетителю:**

* Вовлекающий сайт. 
* Безразличный к посетителю.

*Компоненты веб-страницы:*

− контент (текст, изображения, видео, аудио и т.п.,);

− логическая структура (HTML);

− оформление (CSS);

− поведение (языки программирования).

Этапы разработки веб-сайтов представлены в таблице 1.1

Таблица 1.1. Этапы разработки веб-сайтов

|  |  |  |
| --- | --- | --- |
| **№** | **Этап** | **Результат** |
| 1 | Формирование требований | Техничское задание |
| 2 | Разработка дизайна | Макет сайта (набор графических файлов) |
| 3 | Верстка сайта | Шаблон сайта (HTML, CSS, JS) |
| 4 | Разработка логики приложения (программирование) | Готовое приложение |
| 5 | Тестирование | Исправление ошибок, оптимизация |
| 6 | Публикация сайта в сети Интернет | Размещение сайта на хостинге, настройка хостинга |
| 7 | Продвижение сайта, SEO(Search Engine Optimization) | Увеличение видимости сайта в поисковых системах |

### Web-программирование.

Web-программирование (Web-разработка) - это бурно развивающийся раздел программирования, ориентированный на разработку динамических Интернет-приложений.

Языки Web-программирования делятся на две группы: клиентские и серверные.

Клиентские языки обрабатываются на стороне пользователя (в основном в браузере). Соответственно, обработка скрипта зависит от браузера пользователя, и пользователь имеет полномочия настроить свой браузер так, чтобы тот вообще игнорировал скрипты. При этом если браузер старый, он может не поддерживать тот или иной язык или версию языка, на которую опирался разработчик. С современными браузерами таких проблем возникать не должно, к тому же языки программирования не так уж часто кардинально обновляются (раз в несколько лет) . Код клиентского скрипта может посмотреть каждый, выбрав в меню своего браузера «Исходный код страницы».

Самым распространенным из клиентских языков признан JavaScript, разработчиками которого является компания Netscape совместно с компанией Sun Microsystems.

Серверные языки программирования открывают перед программистом большие просторы для деятельности. Когда пользователь делает запрос на какую-либо страницу (переходит на нее по ссылке, или вводит адрес в адресной строке своего браузера), то вызванная страница сначала обрабатывается на сервере (то есть выполняются все программы, связанные со страницей) и только потом возвращается к посетителю по сети в виде файла.

## **1.2. Основы языка HTML**

Web-страницы – это текстовые файлы, содержащие собственно текст содержимого страницы и команды форматирования, называемые тегами (tag) . Язык разметки гипертекста HTML (HyperText Markup Language) является базовой технологией разработки web-страниц, которые также называют HTML-документами.

Код HTML интерпретируется браузером (средством просмотра webстраниц), который выводит отформатированное содержимое webстраницу на экран.

**Основные элементы HTML**

Основной элемент языка HTML – тег. Тег – указатель разметки, который задается следующим образом:

*[Анатомия HTML элемента](https://developer.mozilla.org/ru/docs/Learn_web_development/Getting_started/Your_first_website/Creating_the_content" \l "%D0%B0%D0%BD%D0%B0%D1%82%D0%BE%D0%BC%D0%B8%D1%8F_html_%D1%8D%D0%BB%D0%B5%D0%BC%D0%B5%D0%BD%D1%82%D0%B0)*

Главными частями нашего элемента являются:

1. Открывающий тег (Opening tag): Состоит из имени элемента , заключённого в открывающие и закрывающие угловые скобки. Открывающий тег указывает, где элемент начинается или начинает действовать.
2. Закрывающий тег (Closing tag): Это то же самое, что и открывающий тег, за исключением того, что он включает в себя косую черту перед именем элемента. Закрывающий элемент указывает, где элемент заканчивается. Отсутствие закрывающего тега является одной из наиболее распространённых ошибок начинающих и может приводить к странным результатам.
3. Контент (Content): Это контент элемента.
4. Элемент(Element): Открывающий тег, закрывающий тег и контент вместе составляют элемент.

Атрибуты содержат дополнительную информацию об элементе, которую вы не хотите показывать в фактическом контенте.

Атрибут всегда должен иметь:

1. Пробел между ним и именем элемента (или предыдущим атрибутом, если элемент уже имеет один или несколько атрибутов).
2. Имя атрибута, за которым следует знак равенства.
3. Значение атрибута, заключённое с двух сторон в кавычки.

*Пустые элементы*

Некоторые элементы не имеют контента, и называются пустыми элементами. Возьмём элемент [<img>](https://developer.mozilla.org/ru/docs/Web/HTML/Reference/Elements/img)

<img src="images/firefox-icon.png" alt="Моё тестовое изображение" />

Он содержит два атрибута, но не имеет закрывающего тега </img>, и никакого внутреннего контента. Это потому, что элемент изображения не оборачивает контент для влияния на него.

Его целью является вставка изображения в HTML страницу в нужном месте.

*[Анатомия HTML документа](https://developer.mozilla.org/ru/docs/Learn_web_development/Getting_started/Your_first_website/Creating_the_content" \l "%D0%B0%D0%BD%D0%B0%D1%82%D0%BE%D0%BC%D0%B8%D1%8F_html_%D0%B4%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0)*

Теперь мы посмотрим, как отдельные элементы объединяются в целую HTML страницу. Пример простого документа представлен на рис.1.2

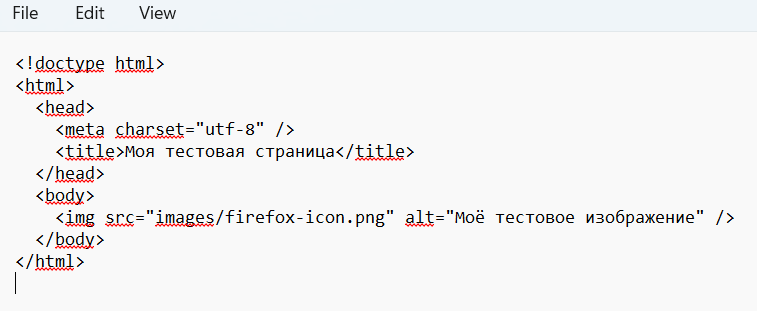


Рис. 1.2. Простой документ, показывающий анатомию HTML документа

Здесь мы имеем:

<!DOCTYPE html> — доктайп.

<html></html> — элемент [<html>](https://developer.mozilla.org/ru/docs/Web/HTML/Reference/Elements/html). Этот элемент оборачивает весь контент на всей странице, и иногда известен как корневой элемент.

<head></head> — элемент [<head>](https://developer.mozilla.org/ru/docs/Web/HTML/Reference/Elements/head). Этот элемент выступает в качестве контейнера для всего, что вы пожелаете включить на HTML страницу, но не являющегося контентом, который вы показываете пользователям вашей страницы.

<body></body> — элемент [<body>](https://developer.mozilla.org/ru/docs/Web/HTML/Reference/Elements/body). В нем содержится весь контент, который вы хотите показывать пользователям, когда они посещают вашу страницу, будь то текст, изображения, видео, игры, проигрываемые аудиодорожки или что-то ещё.

<meta charset="utf-8"> — этот элемент устанавливает UTF-8 кодировку вашего документа, которая включает в себя большинство символов из всех известных человечеству языков. По сути, теперь документ может обрабатывать любой текстовый контент, который вы в него вложите. Нет причин не устанавливать её, так как это может помочь избежать некоторых проблем в дальнейшем.

<title></title> — элемент [<title>](https://developer.mozilla.org/ru/docs/Web/HTML/Reference/Elements/title). Этот элемент устанавливает заголовок для вашей страницы, который является названием, появляющимся на вкладке браузера загружаемой страницы, и используется для описания страницы, когда вы добавляете её в закладки/избранное.

*[Изображения](https://developer.mozilla.org/ru/docs/Learn_web_development/Getting_started/Your_first_website/Creating_the_content" \l "%D0%B8%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D0%B8%D1%8F)*

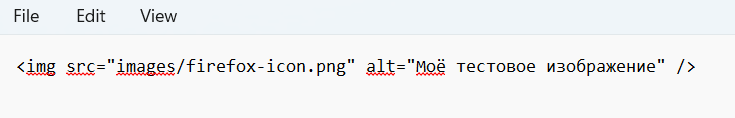


Рис. 1.3. Тег<img> и его атрибуы

Код встраивает изображение на нашу страницу в нужном месте.

Это делается с помощью атрибута

* src (source, источник) - который содержит путь к нашему файлу изображения.

И атрибут

* alt(alternative, альтернатива) - В атрибуте, указывается поясняющий текст для пользователей, которые не могут увидеть изображение, возможно, по следующим причинам:
* У них присутствуют нарушения зрения.
* Что-то пошло не так, в результате чего изображение не отобразилось.

**[Разметка текста](https://developer.mozilla.org/ru/docs/Learn_web_development/Getting_started/Your_first_website/Creating_the_content" \l "%D1%80%D0%B0%D0%B7%D0%BC%D0%B5%D1%82%D0%BA%D0%B0_%D1%82%D0%B5%D0%BA%D1%81%D1%82%D0%B0)**

*[Заголовки](https://developer.mozilla.org/ru/docs/Learn_web_development/Getting_started/Your_first_website/Creating_the_content" \l "%D0%B7%D0%B0%D0%B3%D0%BE%D0%BB%D0%BE%D0%B2%D0%BA%D0%B8)*

Элементы заголовка позволяют вам указывать определённые части контента в качестве заголовков или подзаголовков. HTML включает шесть уровней заголовков <h1>–<h6>, хотя обычно используются не более 3-4 :

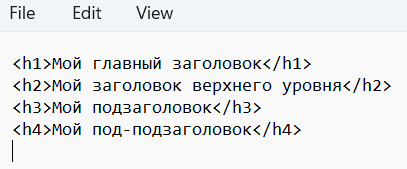


Рис.1.4. Заголовки и подзаголовки

*Абзацы*

Элемент [<p>](https://developer.mozilla.org/ru/docs/Web/HTML/Reference/Elements/p) предназначен для абзацев текста; вы будете использовать их регулярно при разметке текстового контента

*[Списки](https://developer.mozilla.org/ru/docs/Learn_web_development/Getting_started/Your_first_website/Creating_the_content" \l "%D1%81%D0%BF%D0%B8%D1%81%D0%BA%D0%B8)*

Большая часть веб-контента является списками и HTML имеет специальные элементы для них. Разметка списка всегда состоит по меньшей мере из двух элементов. Наиболее распространёнными типами списков являются нумерованные и ненумерованные списки:

* Ненумерованные списки - это списки, где порядок пунктов не имеет значения, как в списке покупок. Они оборачиваются в элемент [<ul>](https://developer.mozilla.org/ru/docs/Web/HTML/Reference/Elements/ul).
* Нумерованные списки - это списки, где порядок пунктов имеет значение, как в рецепте. Они оборачиваются в элемент [<ol>](https://developer.mozilla.org/ru/docs/Web/HTML/Reference/Elements/ol).

Каждый пункт внутри списков располагается внутри элемента [<li>](https://developer.mozilla.org/ru/docs/Web/HTML/Reference/Elements/li) (list item, элемент списка).

*[Ссылки](https://developer.mozilla.org/ru/docs/Learn_web_development/Getting_started/Your_first_website/Creating_the_content" \l "%D1%81%D1%81%D1%8B%D0%BB%D0%BA%D0%B8)*

 Чтобы добавить ссылку, нам нужно использовать простой элемент

[<a>](https://developer.mozilla.org/ru/docs/Web/HTML/Reference/Elements/a) — a это сокращение от "anchor" ("якорь").

<a>Интересный журнал</a>

Задавая элементу [<a>](https://developer.mozilla.org/ru/docs/Web/HTML/Reference/Elements/a) атрибут href, нужно заполнить значение этого атрибута веб-адрессом, на который нужно указать ссылку:

<a href="https://vartumashvili.livejournal.com"

>Интересный журнал</a>

## **1.3.Технология каскадных таблиц стилей CSS**

Таблицы каскадных стилей CSS (Cascade Style Sheets) – это простая технология определения и присоединения стилей оформления к документам HTML.

Стиль оформления – это все то, что определяет внешний вид документа: размер, цвет и вид шрифта текста, цвет фона текста, наличие границ, подчеркивания, выравнивание текста и т. д. Стиль определяется набором правил отображения тегов, задаваемых таблицей стилей.

Таблица стилей – это шаблон, который управляет форматированием тегов HTML в web-документе. Таблица стилей состоит из набора правил описания стиля.

Любое правило каскадных таблиц стилей состоит из двух частей: селектора и определения. Селектор – это любой элемент или группа элементов web-страницы, для которых определяется форматирование. Определение описывает конкретный вид форматирования и состоит из двух частей: свойства и значения, разделенных знаком двоеточия (рис.3.1 ).

Например, цвет текста абзаца может быть задан с помощью стиля как:

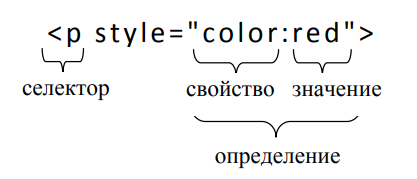


Рис.1.5. Правило таблицы стилей

В ряде случаев использование CSS позволяет создать более компактный и эффективный код HTML-документа.

В одном правиле можно задать несколько определений, разделенных знаком точки с запятой (;).

Например, задание красного цвета, жирного написания и выравнивания «по ширине» текста абзаца с помощью стиля выглядит как:

<p style="color:red; font-weight:bold; text-align:justify"> Текст </p>

Некоторые свойства стиля приведены в табл. 1.2

.

Таблица 1.2 Свойства стилей

|  |  |  |
| --- | --- | --- |
| **Свойство** | **Описание свойства** | **Значения** |
| font-family | Тип шрифта | Имя шрифта, список имен |
| font-size | Размер шрифта | small, large, ex-large и т.п. |
| font-weight | Толщина шрифтпа | normal, bold и т.п. |
| font-style | Наклон шрифтпа | normal, italic и т.п. |
| text-decoration | Подчеркивание | underlne, overline, line-through, none |
| color | Цвет текста | цвет |
| background-color | Цвет фона | цвет |
| background-image | Фоновый рисунок | url('URL рисунка') |
| background-position | Позиционирование фоновой картинки | left, right, center, top, bottom |
| text-align | Выравнивание текста | left, right, center, justify |
| padding | Отступы(внутренние) | %, pt, px, mm, cm, in, em, и т.п. |
| margin | Границы блока(внешние отступы) | %, pt, px, mm, cm, in, em, и т.п. |

Встраивание стиля в теги документа производится с помощью параметра style и позволяет изменить форматирование конкретных экземпляров тегов страницы.

Связывание с таблицей стилей – таблица стилей сохраняется в виде внешнего файла, а в HTML-документ помещается ссылка на нее. Ссылка на файл с таблицей осуществляется тегом link, который указывается в разделе заголовка head. Связывание позволяет использовать одну таблицу CSS для форматирования элементов сразу нескольких web-страниц

## **1.4. Основы JavaScript**

**JavaScript** — это интерпретируемый язык программирования высокого уровня, предназначенный для создания интерактивных элементов на веб-страницах. Он является одним из ключевых компонентов технологии **frontend-разработки**, наряду с HTML и CSS.

JavaScript выполняется непосредственно в браузере пользователя, позволяя:

* динамически изменять содержимое веб-страницы без её перезагрузки;
* обрабатывать действия пользователя (нажатия кнопок, ввод данных и т.д.);
* создавать интерактивные интерфейсы, формы, галереи, анимации и многое другое.

С помощью JavaScript можно управлять **DOM** (Document Object Model) — структурой HTML-документа, представляющей его в виде дерева объектов, доступных для манипуляций.

### 1.4.1.Основные элементы языка JavaScript

*Переменные и типы данных*

JavaScript поддерживает динамическую типизацию. Для объявления переменных используются ключевые слова var, let, и const. Основные типы данных:

* Числа (number)
* Строки (string)
* Булевы значения (boolean)
* Объекты (object)
* Массивы (array)
* Функции (function)
* Специальные значения: null и undefined

*Операторы и выражения*

Как и другие языки, JavaScript поддерживает арифметические, логические, сравнительные и строковые операторы.

*Условные конструкции*

Классические операторы if, else, switch позволяют управлять логикой программы в зависимости от условий.

*Циклы*

Поддерживаются конструкции for, while, do...while для итеративных операций.

*Функции*

Функции в JavaScript — ключевой элемент. Они могут быть именованными, анонимными или стрелочными (=>), и используются для повторного использования кода.

# **ПРАКТИЧЕСКАЯ ЧАСТЬ**

## **2.1 Описание предметной области**

**Предметной областью данной курсовой работы является онлайн-информационная система для расчёта валютных операций**, предоставляющая пользователям удобный инструмент для пересчёта денежных сумм по актуальным обменным курсам.

Онлайн-калькулятор курсов валют представляет собой сервис, позволяющий в реальном времени получать данные о текущих курсах валют (манат, рубль доллар, евро, юань и др.) и производить вычисления между ними. Это особенно актуально для пользователей, совершающих международные финансовые операции, интернет-покупки, планирующих поездки за границу и т.д.

### Архитектура онлайн-системы

Архитектура онлайн-системы — это систематизация информации, логики работы и взаимодействия элементов, позволяющая пользователю эффективно пользоваться функционалом сервиса.

Грамотно спроектированная архитектура сайта обеспечивает:

* удобную навигацию по разделам;
* быстрый доступ к основным функциям (ввод суммы, выбор валют, получение результатов);
* минимизацию времени на поиск нужных данных;
* высокую доступность и понятность даже для неподготовленного пользователя.

Ключевым элементом системы является интерфейс — через него происходит всё взаимодействие между пользователем и сервисом. От его продуманности зависит простота восприятия информации и точность работы с калькулятором.

Размещение полей ввода, кнопок, выпадающих списков с выбором валют, отображение результатов — всё это логично выстроено, чтобы пользователь быстро ориентировался и получал нужную информацию.

Для реализации системы были применены следующие технологии:

* **HTML** — базовый язык разметки, используемый для создания структуры веб-страницы и отображения контента.
* **CSS** — язык каскадных таблиц стилей, применяемый для оформления внешнего вида и адаптации сайта под различные устройства.
* **JavaScript** — язык программирования, реализующий интерактивность: обработку событий, пересчёт сумм, обращение к API обменных курсов.

## **2.2 Установка и настройка компонентов проекта**

Для разработки используется Visual Studio Code. В нём открывается подготовленная рабочая папка, после чего формируется структура проекта путём создания требуемых файлов для дальнейшей работы.

Создаем папку “kursovaya” и добавим все файлы проекта. Сайт будет состоять из 7 страниц. Создадим для главной страницы файл index.html. И для этой главной страницы создаем “style.css”и “script.js” файлы.

Для остальных дополнительных 6 страниц, стили для этих страниц уже будут в их файле , соответственно “ azerbaijan.hmtl”, “china.html”, “europe.html”, “japan.html”, “russia.html”, “usa.html”.   
Компоненты проекта готовы и можно начинать разработку (рис. 2.1)

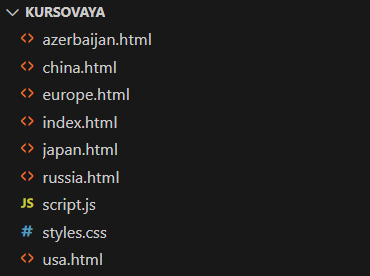


Рис. 2.1 Конечный вид проекта

В созданный ранее “index.html” и пишем стандартный HTML5 код (рис. 2.2).

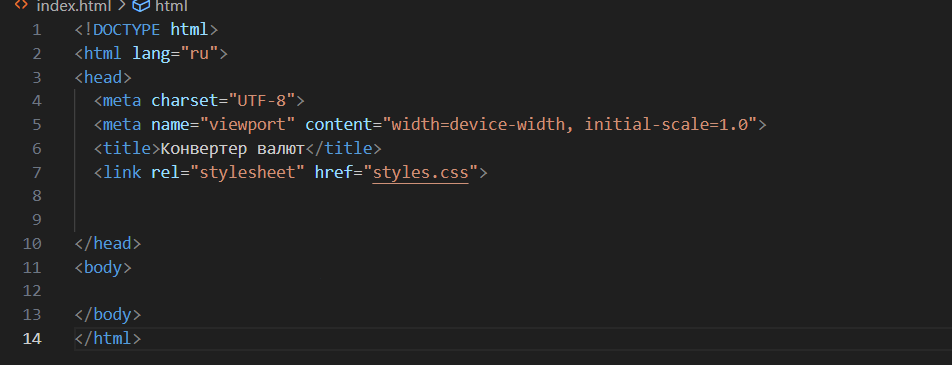


Рис. 2.2 Изначальный вид HTML5 страницы

* <!DOCTYPE html> — объявление HTML5, гарантирует современную вёрстку.
* <html lang="ru"> — корневой элемент, атрибут lang="ru" помогает браузеру и поисковикам определить язык страницы.
* <meta charset="UTF-8"> — кодировка документа, поддерживает кириллицу.
* <meta name="viewport" …> — задаёт адаптивность на мобильных устройствах (масштабирование, ширина экрана).
* <title> — заголовок вкладки браузера.
* <link rel="stylesheet" href="styles.css"> — подключение внешнего CSS-файла для стилей.

<title> - используется для задания заголовка страницы, который отображается на вкладке браузера. Он должен находиться внутри тега <head> .

Чтобы применить стили из внешнего файла на странице, используйте тег <link> (рис.2.3).

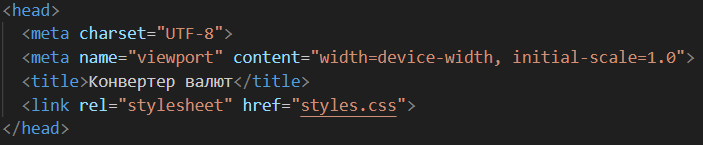


Рис. 2.3.Теги <title> и <link>

Теперь в <body> будем прописывать контент.

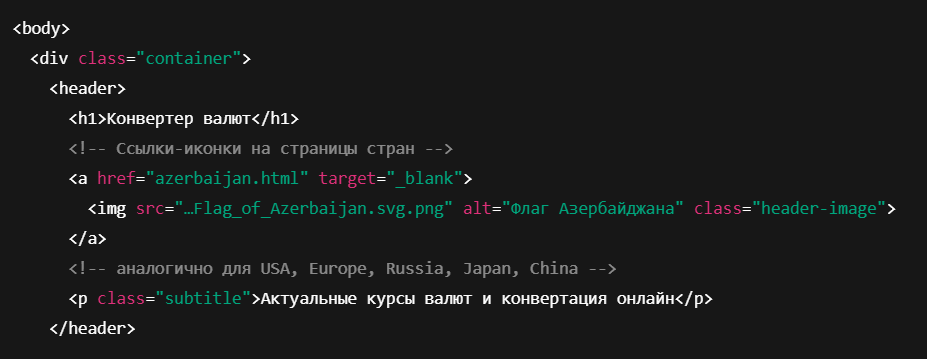


Рис. 2.4

* .container — общий «контейнер» для центрирования и ограничения ширины контента.
* <header> — семантический блок шапки:
* <h1> — главный заголовок.
* Последующие <a><img></a> — навигация по флагам, ссылки открываются в новой вкладке (target="\_blank").
* <p class="subtitle"> — подзаголовок с кратким описанием.

Для добавления изображения на фон мы указываем путь к изображению.

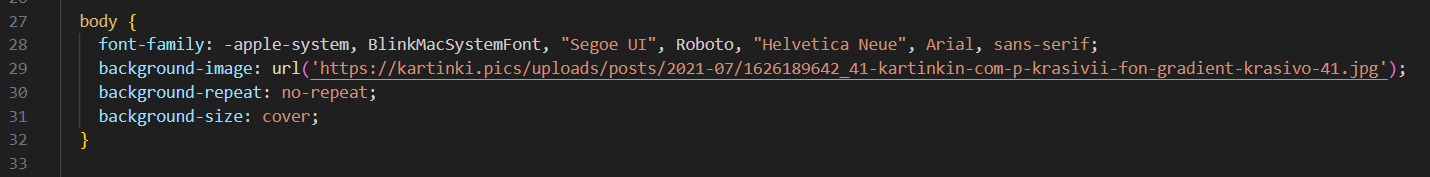


Рис.2.5

Применяемые стили к общему контейнеру, семантическому блоку шапки и флагам в блоке шапки.

Рис.2.6

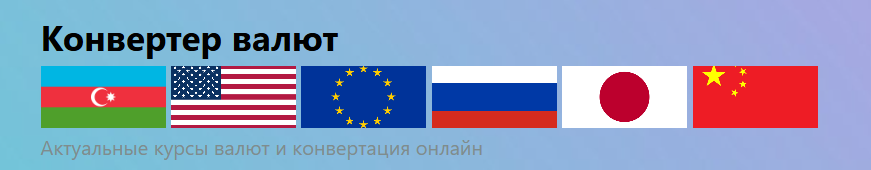


Рис.2.7 Итоговый вид сайта

*Основная часть <main>.*

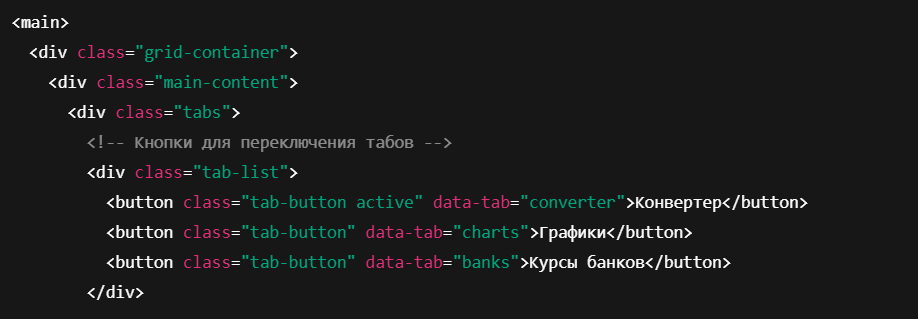


Рис.2.8

* <main> — семантика основной части страницы.
* grid-container — CSS-грид, делит экран на основную область и сайдбар.
* .tabs + .tab-list + button[data-tab] — кнопки переключения вкладок

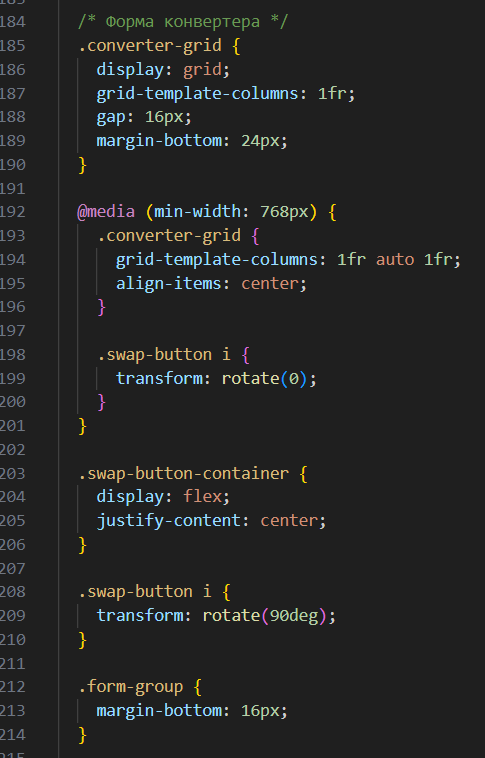
Рис.2.9

Рис. 2.10

* .converter-grid — CSS-грид для трёх колонок: ввод, swap(переключатель), вывод.
* #amount, #fromCurrency, #toCurrency, #result — поля для JS-логики конвертации.
* swapButton — меняет местами выбранные валюты.



Рис.2.11

* .exchange-rate-info — блок с текущим курсом, процентным изменением и временем последнего обновления.
* Классы positive/negative меняют цвет и иконку.
* #rateValue — туда подставляется обновлённое значение курса.

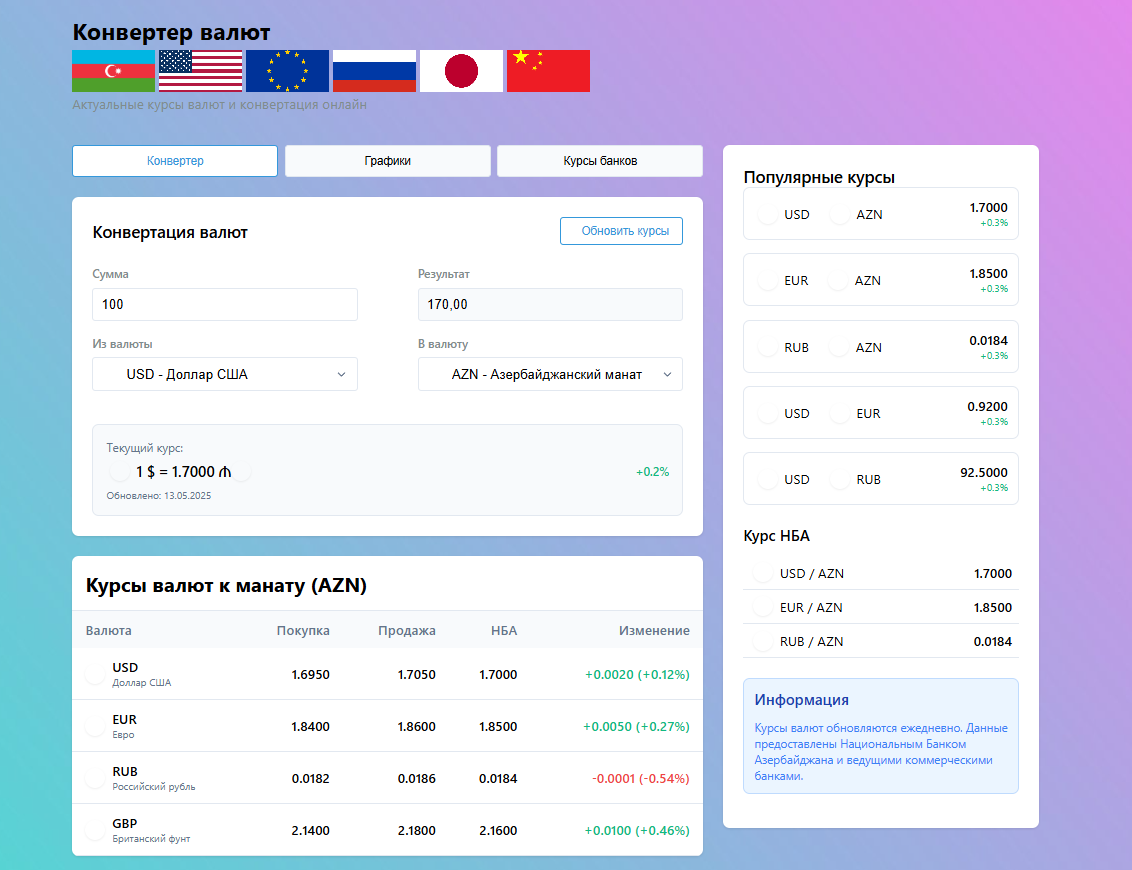


Рис. 2.12

*Таблица курсов*

<div class="rates-table-container">

<h2>Курсы валют к манату (AZN)</h2>

<table>

<thead>

<tr> <th>Валюта</th><th>Покупка</th><th>Продажа</th><th>НБА</th><th>Изменение</th>

</tr>

</thead>

<tbody>

<tr>

<td>

<div class="currency-info">

<div class="currency-flag flag-us"></div>

<div>

<div class="currency-code">USD</div>

<div class="currency-name">Доллар США</div>

</div>

</div>

</td>

<td>1.6950</td><td>1.7050</td><td>1.7000</td>

<td>

<div class="change-info positive">

<i class="fas fa-chart-line"></i>

<span>+0.0020 (+0.12%)</span>

</div>

</td>

</tr>

<!-- Аналогичные строки для EUR, RUB, GBP -->

</tbody>

</table>

</div>

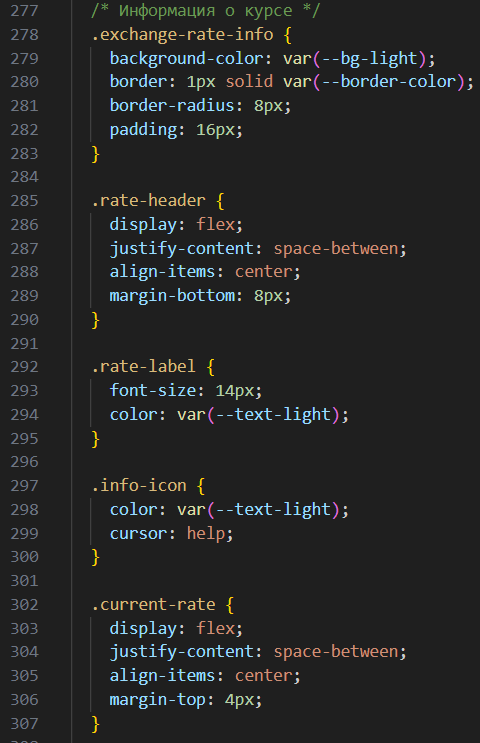
* Семантическая таблица:
* <thead> — заголовки столбцов.
* <tbody> — данные по каждой валюте.
* Классы и вложенные блоки для вывода флага, кода, названия и динамики.



Рис. 2.14

То , как выглядит в итоге контейнер «rates-table-container» на самом сайте.

Так же тут указаны, 5 столбцов ниже будет указан исходный код.

<div class="rates-table">

                  <table>

                    <thead>

                      <tr>

                        <th>Валюта</th>

                        <th>Покупка</th>

                        <th>Продажа</th>

                        <th>НБА</th>

                        <th>Изменение</th>

                      </tr>

                    </thead>

                    <tbody>

                      <tr>

                        <td>

                          <div class="currency-info">

                            <div class="currency-flag flag-us"></div>

                            <div>

                              <div class="currency-code">USD</div>

                              <div class="currency-name">Доллар США</div>

                            </div>

                          </div>

                        </td>

                        <td>1.6950</td>

                        <td>1.7050</td>

                        <td>1.7000</td>

                        <td>

                          <div class="change-info positive">

                            <i class="fas fa-chart-line"></i>

                            <span>+0.0020 (+0.12%)</span>

                          </div>

                        </td>

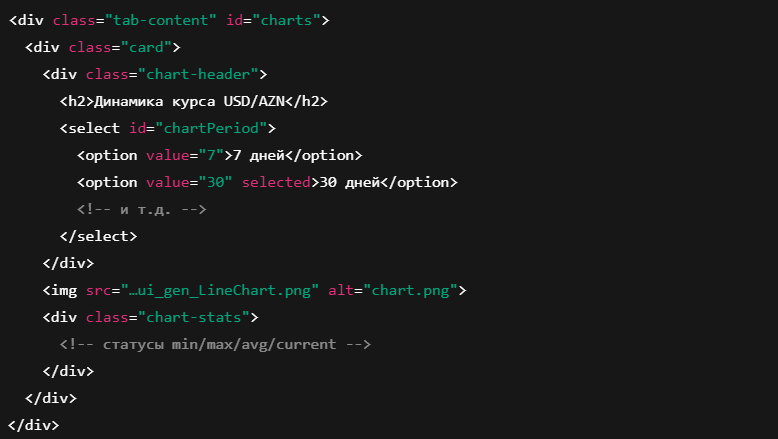
                      </tr>

                      <tr>

                        <td>

Рис.2.15

*Вкладка «Графики»*

Рис.2.16

* #chartPeriod — выбор периода, нужен обработчик для перерисовки графика.
* Сам график статический <img> в виде фотографии .

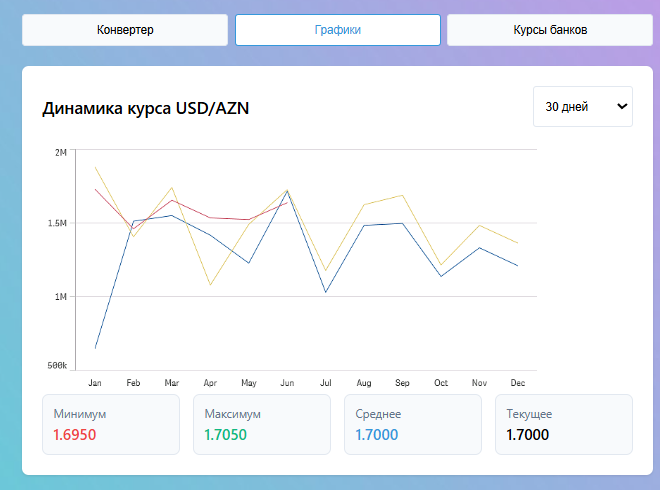


Рис.2.17

*Вкладка «Курсы банков»*

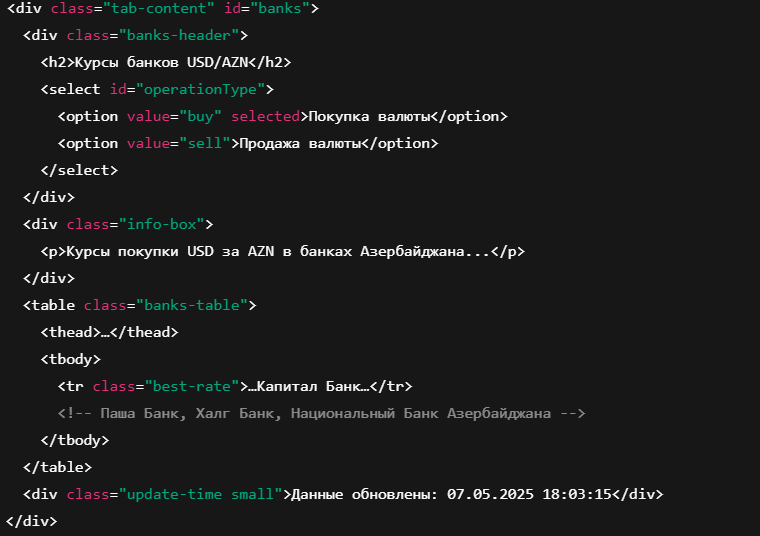


Рис.2.18

* #operationType переключает между покупкой и продажей (JS-логика фильтрации/перестройки таблицы).
* В строке с классом best-rate отмечается лучшая цена.

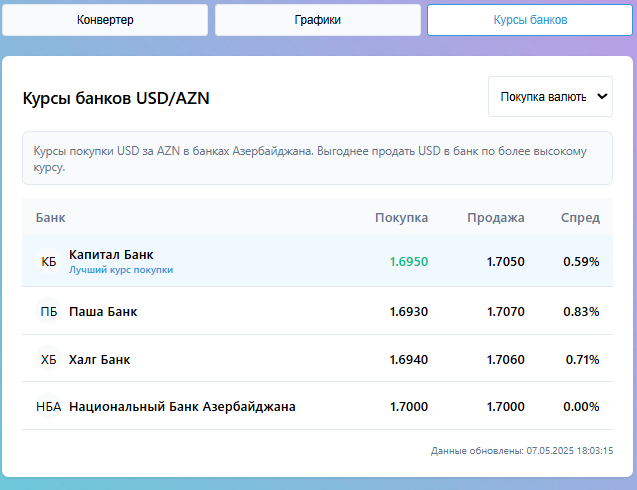
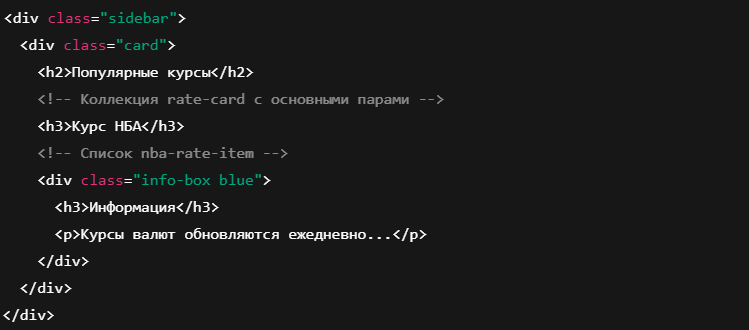


Рис.2.19

*Сайдбар(боковая панель сайта)*

Рис.2.20

Повторяются похожие карточки с наиболее востребованными парами валют.

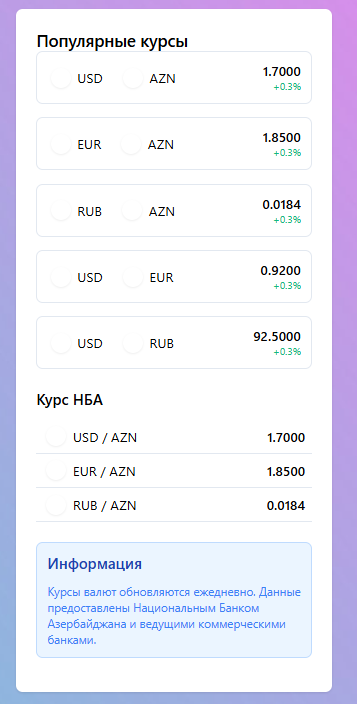
Блок «Информация» внизу.

Рис.2.21

*Подвал и подключение скрипта*

Рис.2.22

· Семантический <footer> с копирайтом.

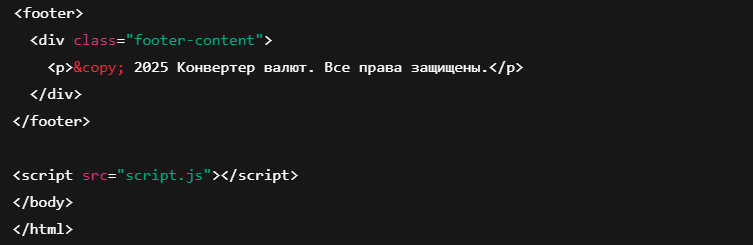
· Подключение script.js для:

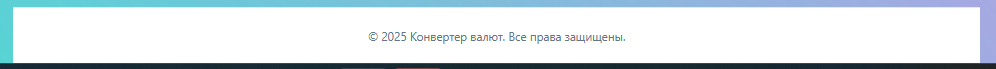
переключения табов,

конвертации чисел,

перерисовки графика,

обработки кнопки swap и селектов.

В конце файла “index.html” имеется ссылка на другой файл с javascript “script.js”.

Рис.2.23

Сейчас рассмотрим остальные 6 идентичных страниц.

Подробный разбор кода страницы «Валюта Азербайджана (AZN)» и его ключевых блоков. Обратите внимание, что такая же структура и стили (с соответствующей заменой флагов, заголовков и контента) применяются и к другим страницам вашего сайта, посвящённым валютам других стран. Поэтому рассмотрим только одну из них.

*Обёртка и общий контейнер*

**<div class="currency-info-container">**

**… весь контент страницы …**

**</div>**

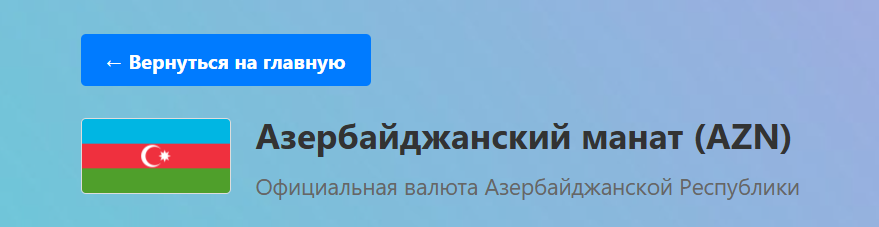
«.currency-info-container»

* Ограничивает максимальную ширину (max-width:1200px), центрирует блок на странице (margin:0 auto) и даёт внутренние отступы (padding:20px).
* Используется на каждой «странице страны» одинаково, меняются лишь вложенные данные.

*Навигационная кнопка «Назад»*

**<a href="index.html" class="back-link">← Вернуться на главную</a>**

* **Класс** .back-link задаёт стили кнопки:
  + Синий фон, белый текст, скруглённые углы и эффект при наведении.
* В конце страницы дублируется такая же ссылка.

Рис.2.24

**<div class="currency-header">**

**<img src="…Flag\_of\_Azerbaijan.svg" alt="Флаг Азербайджана" class="currency-flag">**

**<div class="currency-title">**

**<h1>Азербайджанский манат (AZN)</h1>**

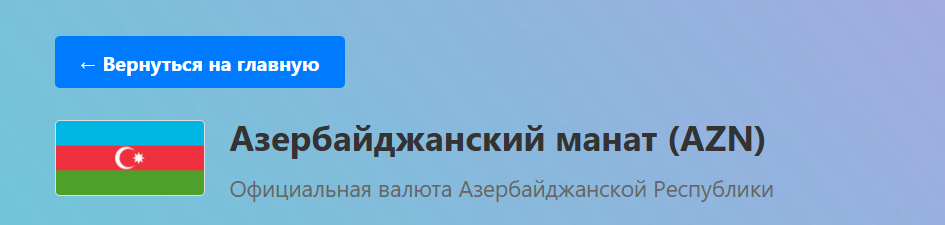
**<p>Официальная валюта Азербайджанской Республики</p>**

**</div>**

**</div>**

* .currency-header — горизонтальный флекс-контейнер, выравнивает флаг и заголовок по центру.
* .currency-flag — фиксированная ширина 120 px, обводка и скруглённые углы.
* .currency-title содержит основной заголовок <h1> и подпись <p>.

Для других стран просто замените URL флага, текст в <h1> и описание в <p>.

Рис.2.25

*Секции с информацией*

**<div class="currency-sections">**

**<div class="currency-section">…</div>**

**<div class="currency-section">…</div>**

**<div class="currency-section">…</div>**

**<div class="currency-section">…</div>**

**</div>**

* .currency-sections — CSS-грид с двумя колонками (grid-template-columns:1fr 1fr) и отступами между ними.

Каждый блок .currency-section оформлен так:

**.currency-section {**

**background: #fff;**

**border-radius: 8px;**

**box-shadow: 0 2px 10px rgba(0,0,0,0.1);**

**padding: 20px;**

**margin-bottom: 30px;**

**}**

**.currency-section h2 {**

**color: #333;**

**border-bottom: 2px solid #f0f0f0;**

**padding-bottom: 10px;**

**margin-top: 0;**

**}**

**.currency-section p,**

**.currency-section ul,**

**.currency-section ol {**

**color: #444;**

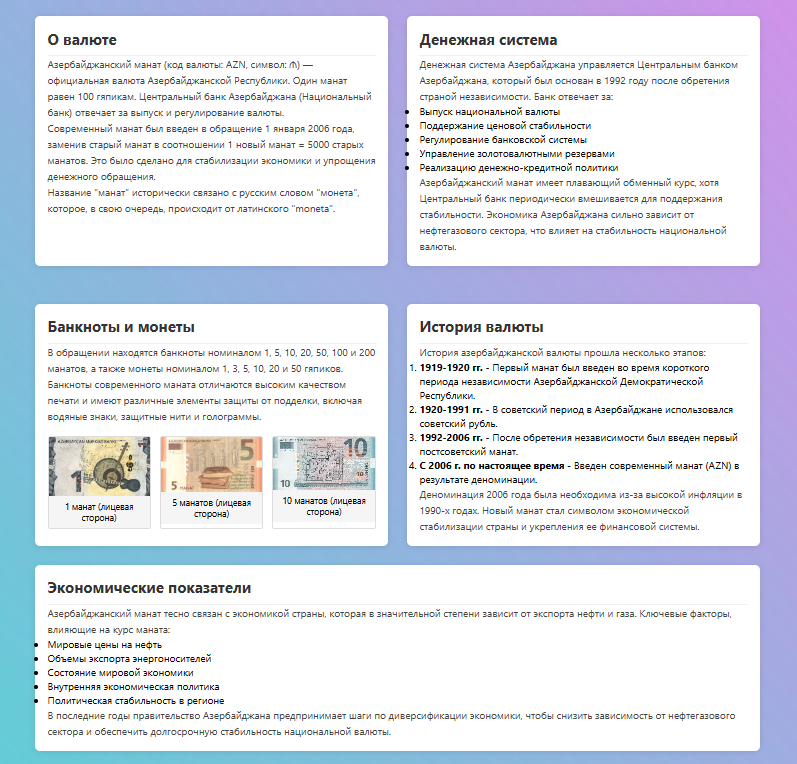
**line-height: 1.6;**

**}**

Основные секции:

1. «О валюте»  
   Краткая справка об AZN, дате введения в обращение, историческом корне слова «манат».
2. «Денежная система»  
   Роль Центрального банка, функции, курс (плавающий), зависимость от нефтегазового сектора.
3. «Банкноты и монеты»  
   Перечень номиналов, защита от подделок и галерея банкнот.
4. «История валюты»  
   Хронология: Демократическая Республика, советское время, первый манат, деноминация 2006.
5. «Экономические показатели»  
   Факторы влияния на курс (цены на нефть, экспорт, политика, стабильность).

Для других стран просто меняйте заголовки, текст и изображения банкнот/монет на соответствующие.



*Галерея банкнот*

**<div class="banknotes">**

**<div class="banknote">**

**<img src="…1\_manat.jpg" alt="1 манат">**

**<div class="banknote-caption">1 манат (лицевая сторона)</div>**

**</div>**

**<!-- ещё блоки .banknote -->**

**</div>**

**3. РЕЗУЛЬТАТ**

В процессе выполнения курсовой работы были изучены и успешно применены **основы веб-разработки**, включая создание структуры сайта, оформление стилей и адаптация интерфейса под разные устройства. В частности:

**1.Разработана структура веб-страницы с использованием языка HTML**, обеспечивающая логичную организацию контента и удобную навигацию.

**2. Для стилизации использован CSS**, что позволило оформить сайт в соответствии с современными визуальными стандартами, придать элементам индивидуальность и обеспечить читаемость контента.

**3. Реализована адаптивная вёрстка с помощью медиазапросов**, благодаря чему сайт корректно отображается как на компьютерах, так и на мобильных устройствах, обеспечивая комфортное взаимодействие для пользователей на любых платформах.

В результате проделанной работы был создан **современный, функциональный и эстетичный веб-сайт**, отвечающий требованиям проекта.

# **4. СПИСОК ЛИТЕРАТУРЫ**

1. ЯКУБОВИЧ Д.А. ЕРОПОВА Е.С. ЕРОПОВ И.А.Основы WEB-разработки
2. Магомедова Динара Сахратулаевна Бекбулатова Зайнаб Абдулмуслимовна.Учебное пособие по дисциплине «Web-программирование»
3. Е.В. Киргизова, А.В. Рубцов - WEB-ТЕХНОЛОГИИ: от теории к практике
4. Учебник по JavaScript от Ильи Кантора – Learn JavaScript
5. И. Н. ВАСИЛЬЕВА, Д. Ю. ФЕДОРОВ WEB-ТЕХНОЛОГИИ
6. Т.В. Зудилова, М.Л. Буркова Web-программирование HTML
7. <https://skillbox.ru/media/code/chto_takoe_html/>
8. <https://sky.pro/wiki/html/osnovy-html-struktura-html-dokumenta/>
9. <https://developer.mozilla.org/ru/docs/Learn_web_development/Getting_started/Your_first_website/Creating_the_content>
10. <https://learn.javascript.ru>
11. <https://studfile.net/preview/21325586/>
12. <https://books.ifmo.ru/file/pdf/878.pdf>

# **5. ПРИЛОЖЕНИЯ**

Репозиторий:

<https://github.com/QadirovaElvina/13QadirovaElvina>

Страница сайта на хостинге GitHub Pages:

<https://qadirovaelvina.github.io/13QadirovaElvina/>