CURRENCY CONVERTER

A Micro Project Report

Submitted by

SADHANA SHRI R

Reg.no: 99220040717

B.Tech - CSE,

DATA SCIENCE

Kalasalingam Academy of Research and Education

(Deemed to be University)

Anand Nagar, Krishnankoil - 626 126

FEBRUARY 2024

KALASALINGAM
ACADEMY OF RESEARCH AND EDUCATION
(DEEMED TO BE UNIVERSITY)
Under sec. 3 of UGC Act 1956. Accredited by NAAC with "A" Grade

SCHOOL OF COMPUTING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Bonafide record of the work done by **SADHANA SHRI R– 99220040717** in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Specialization of the Computer Science and Engineering, during the Academic Year Even Semester (2023-24).

Dr. C. BalaSubramanian
Project Guide
Assistant Professor
Computer Science And Engineering
Kalasalingam Academy of
Research and Education
Krishnan kovil – 626126

Mr. N. R. Sathis Kumar
Faculty Incharge
Assistant Professor
Computer Science And Engineering
Kalasalingam Academy of
Research and Education
Krishnan kovil - 626126

Dr. P. Anitha
Evaluator
Assistant Professor
Computer Science And Engineering
Kalasalingam Academy of Research and Education
Krishnan Kovil – 626126

Abstract

In the modern interconnected world, the need for currency conversion tools is paramount for individuals and businesses alike. This project aims to develop a user-friendly currency converter using HTML, CSS and JavaScript. The converter will allow users to seamlessly convert between different currencies based on real-time exchange rates. The interface will be designed using HTML and CSS to ensure an intuitive and visually appealing experience for users. Through JavaScript, the converter will fetch the latest exchange rates from a reliable API source, enabling accurate and up-to-date conversions. Additionally, users will have the flexibility to input the desired amount in one currency and instantly see the equivalent amount in another currency, with conversion rates dynamically updated.

Contents

Chapter 1 – Development Environment Setup

1.1 Desktop Environment Setup – Windows

To set up a desktop environment on windows, you'll need a few tools. A basic setup:

a. Text Editor or Integrated Development Environment (IDE): Visual Studio Code (VS Code) - A popular and lightweight code editor with excellent support for HTML, CSS, and JavaScript. You can download it from https://code.visualstudio.com

Extensions Installation in VS Code:

1. Live Server – to host the local server website

2. Live Preview - Hosts a local server in your workspace for you to preview your webpages

b. Web Browser: Google Chrome or Mozilla Firefox: Both browsers have robust developer tools for inspecting and debugging web pages.

1.2 Environment Setup – GitHub

Setting up a development environment with GitHub involves not only the tools you use locally but also the integration of version control and collaboration features provided by GitHub.

a. GitHub Account: If you don't have one already, sign up for a GitHub account at https://github.com/

b. GitHub Repository: Create a new repository on GitHub by clicking the "New" button on the GitHub website.

Choose a name for your repository, add a description if needed, and configure other settings as required.

Optionally, initialize the repository with a README file, a .gitignore file, and a license.

c. Upload: Upload a project file or folder in the created Repository.

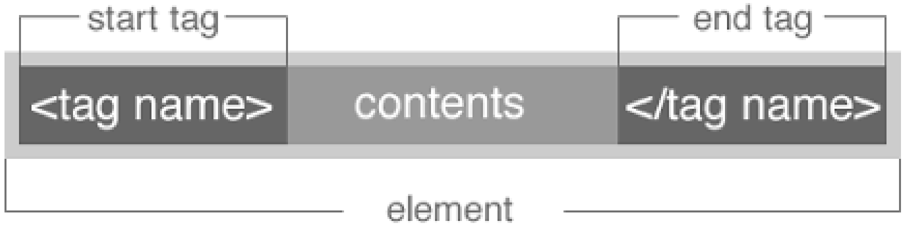
Chapter 2 – HTML

2.1 Introduction to HTML

1. **HTML** stands for **Hyper Text Mark Language.**
2. HTML is the standard markup language for creating Web pages.
3. HTML describes the structure of a Web page.
4. HTML consists of a series of elements.
5. HTML elements tell the browser how to display the content.
6. HTML elements label pieces of content such as "this is a heading", "this is a

paragraph", "this is a link", etc.

Syntax:



2.2 History of HTML

Berners-Lee developed the first version of HTML in 1990, primarily as a simple markup language to create and link documents on the web. The first web page, which also served as a tutorial on how to create web pages, was published by Berners-Lee in 1991.

In 1995, HTML 2.0 was published as the first formal specification by the Internet Engineering Task Force (IETF), standardizing many of the elements and attributes used in early web development.

HTML continued to evolve with the introduction of HTML 3.2 and then HTML 4.0, which introduced more sophisticated layout and styling options.

HTML5, the latest major revision of HTML, was introduced in 2008 with the goal of modernizing the language and addressing the needs of contemporary web development.

The HTML5 specification also incorporated features for offline web applications, geolocation, canvas for drawing graphics, and more.

2.3 Basic of HTML Tags

HTML tags are the building blocks of HTML documents. They are used to define the structure and content of web pages. Here are some basic HTML tags along with their descriptions:

1. **<!DOCTYPE html>:** This declaration specifies the document type and version of HTML being used.

2. **<html>:** This tag defines the root of an HTML document and contains all other HTML elements.

3. **<head>:** This tag contains meta-information about the document, such as its title, links to stylesheets, and scripts. It's not displayed on the web page itself.

4. **<title>:** This tag sets the title of the document, which appears in the browser's title bar or tab.

5. **<body>:** This tag contains the main content of the document, including text, images, links, etc. Everything that you see on a web page is contained within the body tag.

6. **<h1> to <h6>:** These tags define headings of different levels, with <h1> being the highest level and <h6> being the lowest.

7. **<p>:** This tag defines a paragraph of text.

8. **<a>:** This tag defines a hyperlink, allowing you to link to other web pages or resources.

9. **<img>:** This tag embeds an image into the web page.

10. **<div>:** This tag defines a division or section in an HTML document, often used for grouping and styling purposes.

11. **<button>:** This tag in HTML is used to create a clickable button on a web page.

12. **<br>:** This tag inserts a line break within text.

13. **<hr>:** This tag inserts a horizontal line, typically used as a thematic break between sections of content.

14. **<em>:** This tag defines text that should be emphasized, often displayed in italics.

15. **<b>:** This tag defines bold text.

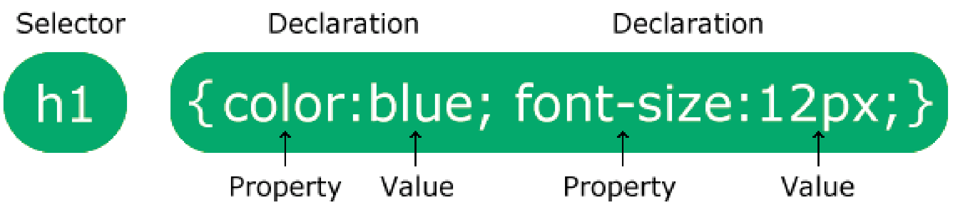
16. **<i>:** This tag defines italicized text.

Chapter 3 CSS

3.1 Introduction of CSS

1. CSS stands for **Cascading Style Sheets**.
2. CSS describes how HTML elements are to be displayed on screen, paper, or in other media.
3. CSS saves a lot of work. It can control the layout of multiple web pages all at once.
4. External stylesheets are stored in CSS files.

Syntax:



3.2 History of CSS

In 1994, Håkon Wium Lie and Bert Bos proposed CSS as part of the proposed HTML standardization process.

CSS Level 1 (CSS1) was officially released in December 1996 as a recommendation by the World Wide Web Consortium (W3C). CSS1 provided basic styling capabilities such as font properties, colors, text alignment, and margins.

CSS Level 2 (CSS2) followed in May 1998, introducing more advanced features such as positioning, floats, and z-index. CSS2 also addressed issues related to accessibility and internationalization.

CSS Level 3 (CSS3) development began in the early 2000s. CSS3 introduced a wide range of new features and enhancements, including advanced selectors, flexible box layout, grid layout, transformations, transitions, animations, and more.

In recent years, there has been a focus on responsive web design, accessibility, and performance optimization, with CSS playing a crucial role in achieving these goals.

3.3 Basic of CSS Elements

1. Selectors: Selectors are patterns used to select the HTML elements you want to style. Here are some common types of selectors:

a) **Element Selector**: Selects elements based on their tag name.
b) **Class Selector:** Selects elements with a specific class attribute.
c) **ID Selector:** Selects a single element with a specific ID attribute.
d) **Universal Selector:** Selects all elements on the page. It's denoted by *.

2. Properties: CSS properties are the attributes you can apply to selected elements to style them.

Here are some common properties:

a) **color:** Sets the color of text.
b) **background-color:** Sets the background color of an element.
c) **font-family:** Sets the font family for text.
d) **font-size**: Sets the font size for text.
e) **font-weight:** Sets the boldness of text.
f) **text-align:** Sets the alignment of text.
g) **margin:** Sets the margin around an element.
h) **padding:** Sets the padding within an element.
i) **border:** Sets the border around an element.
j) **width:** Sets the width of an element.
k) **height:** Sets the height of an element.

3. **Values:** Values are assigned to CSS properties to define how they should be applied. Here are some common values:

a) **Color values:** Keywords (e.g., red, blue), hexadecimal codes (e.g., #ff0000), RGB values (e.g., rgb (255, 0, 0)).
b) **Length values:** Pixels (e.g., 10px), percentages (e.g., 50%), em units (e.g., 1em).
c) **Font values:** Font names (e.g., Arial, Times New Roman), generic font families (e.g., serif, sans-serif).
d) **Text alignment values:** left, center, right.

4. Selectors and Properties Combined: CSS rules consist of selectors and the properties and values you want to apply to the selected elements

5. Box Model: It consists of content, padding, border, and margin. Understanding the box model is crucial for layout design and positioning of elements.

Chapter 4 JavaScript

4.1 Introduction to JavaScript

JavaScript is a versatile programming language primarily used for creating interactive web pages. It is a key component of web development alongside HTML and CSS. Here's an introduction to JavaScript:

Purpose: JavaScript (JS) was originally created to add interactivity to web pages. It enables developers to manipulate the content and behavior of web pages dynamically, allowing for interactive features such as form validation, animations, and responsive interfaces.

Dynamic and Versatile: JavaScript is a dynamic language, which means variables can be assigned different data types and values dynamically during runtime. It is also a versatile language that supports various programming paradigms, including procedural, object-oriented, and functional programming.

Libraries and Frameworks: JavaScript has a vast ecosystem of libraries and frameworks that extend its capabilities and simplify common tasks. JavaScript plays a crucial role in modern web development, enabling developers to create dynamic and engaging web applications that provide rich user experiences.

4.2 Basic of JavaScript

JavaScript is a versatile programming language primarily used for creating interactive and dynamic web content. Below are some basic concepts and features of JavaScript:

1. **Variables:** Variables are used to store data values. In JavaScript, you can declare variables using the var, let, or const keywords.

2. **Data Types:** JavaScript supports various data types, including numbers, strings, booleans, arrays, objects, and more.

3. **Operators:** JavaScript includes arithmetic, assignment, comparison, logical, and other types of operators for performing operations on variables and values.

4. **Functions:** Functions are blocks of reusable code that perform a specific task. They can take parameters as input and return a value.

5. **Conditional Statements:** JavaScript supports conditional statements such as if, else if, and else, allowing you to execute different blocks of code based on specified conditions.

5. Conclusion and Future Work

In conclusion, building a currency converter using HTML, CSS, and JavaScript is a great project that demonstrates practical application of web development skills. By combining HTML for structure, CSS for styling, and JavaScript for functionality, you can create an interactive tool that allows users to convert between different currencies.

For future work on this currency converter project, you could consider the following enhancements:

Improved Design: Enhance the visual design and layout of the currency converter to make it more appealing and user-friendly. Experiment with different color schemes, fonts, and layouts to create a polished interface.

Accessibility: Ensure that the currency converter is accessible to users with disabilities by following best practices for web accessibility. Use semantic HTML, provide alternative text for images, and ensure keyboard navigation is supported.

By continuing to work on this project and implementing these enhancements, you can further develop your skills in web development and create a valuable tool that provides real-world utility to users.

6. References

1. **https://www.coursera.org/learn/html-css-javascript-for-web-developers/home/week/1**
2. **https://www.coursera.org/learn/html-css-javascript-for-web-developers/home/week/2**
3. **https://www.coursera.org/learn/html-css-javascript-for-web-developers/home/week/3**
4. **https://www.coursera.org/learn/html-css-javascript-for-web-developers/home/week/4**
5. **https://www.coursera.org/learn/html-css-javascript-for-web-developers/home/week/5**
6. **https://www.w3schools.com/**

7.https://github.com/SadhanaShriR/SadhanaShriR.github.io/tree/main/Project/CURRENCY%20CONVERTER

7. Appendix

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <link rel="stylesheet" href="style.css">
   <link rel="icon" href="img/calculator-money-icon.png" type="image/x-icon">
   <title>Currency Convertor</title>
</head>
<body>
   <div class="container">
      <h1>Currency Convertor</h1>
      <div class="box">
         <div class="left-box">
            <select name="currency" class="currency">
            </select>
            <input type="number" name="" id="input">
         </div>
         <div class="right-box">
            <select name="currency" class="currency"></select>
            <input type="number" name="" id="result">
         </div>
      </div>
      <div class="btn">
         <button id="btn"><b>Convert</b></button>
      </div>
   </div>
```

```
    <script src="script.js"></script>
</body>
</html>
```

CSS

```
html,
body{
            font-family: 'Poppins', sans-serif;
background: rgb(238,174,202);
background: radial-gradient(circle, rgba(238,174,202,1) 0%, rgba(148,187,233,1) 100%);
             margin: 0;
            padding: 0;
              box-sizing: border-box;
}
  body{
              height: 100vh;
            display: flex;
            align-items: center;
             justify-content: center;
  }
  .container{
            background-color:#1f2029;
            padding: 10px 24px;
             border-radius: 20px;
            width: 490px;
}
  h1{
             color:#328079;
             text-align: center;
             margin-bottom: 0.5em;
```

```
          font-family: sans-serif;
}
.container .box{
          width: 100%;
          display: flex;
}
.box div{
          width: 100%;
}
select{
          width: 95%;
           height: 40px;
          font-size: 1.2em;
          cursor: pointer;
          background-color:rgba(95, 180, 186, 0.395);
          outline: none;
          color: rgb(0, 0, 0);
           margin: 0.2em 0;
  padding: 0 1em;
  border-radius: 10px;
  border: none;
}
input{
           width: 80%;
           height: 40px;
           font-size: 1em;
           margin: 0.2em 0;
          border-radius: 10px;
           border: none;
           background: #b7b7b7;
          outline: none;
          padding: 0 1em;
 }
 .btn{
           display:flex;
          justify-content: center;
 }
 button{
          width: 50%;
          height: 40px;
          background-color:#6b4269;
          color: #000000;
           border-radius: 10px;
          border: none;
          cursor: pointer;
            font-size: 1em;
           margin: 0.5em 0;
 }
```

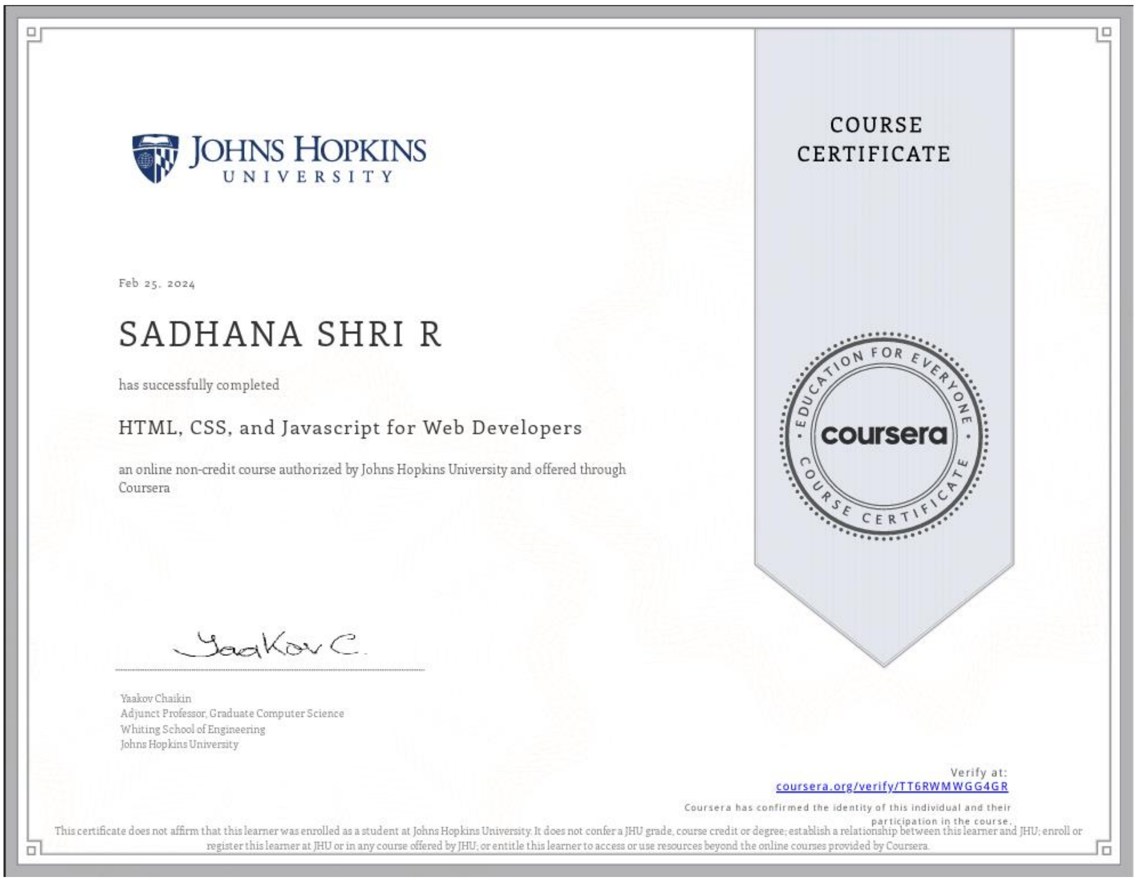
JavaScript

```
let select = document.querySelectorAll('.currency')
let btn = document.getElementById('btn')
let input = document.getElementById('input')
fetch('https://api.frankfurter.app/currencies')
.then(res=>res.json())
.then(res=>displayDropDown(res))
function displayDropDown(res){
  //console.log(Object.entries(res)[2][0])
  let curr = Object.entries(res)
  for(let i=0;i<curr.length;i++){
   let opt = `<option value="${curr[i][0]}">${curr[i][0]}</option>`
   select[0].innerHTML += opt
   select[1].innerHTML += opt
  }
}
btn.addEventListener('click',()=>{
  let curr1 = select[0].value
  let curr2 = select[1].value
  let inputVal = input.value
  if(curr1===curr2)
    alert("Choose different currencies")

 else
    convert(curr1,curr2,inputVal)
});

function convert(curr1,curr2,inputVal){
  const host = 'api.frankfurter.app';
  fetch(`https://${host}/latest?amount=${inputVal}&from=${curr1}&to=${curr2}`)
  .then(resp => resp.json())
  .then((data) => {
    document.getElementById('result').value = Object.values(data.rates)[0]
  });
}
```

8. Certification



COURSE CERTIFICATE

JOHNS HOPKINS UNIVERSITY

Feb 25, 2024

SADHANA SHRI R

has successfully completed

HTML, CSS, and Javascript for Web Developers

an online non-credit course authorized by Johns Hopkins University and offered through Coursera

Yaakov C.

Yaakov Chaikin
Adjunct Professor, Graduate Computer Science
Whiting School of Engineering
Johns Hopkins University

EDUCATION FOR EVERYONE · coursera · COURSE CERTIFICATE

Verify at:
coursera.org/verify/TT6RWMWGG4GR

Coursera has confirmed the identity of this individual and their participation in the course.

This certificate does not affirm that this learner was enrolled as a student at Johns Hopkins University. It does not confer a JHU grade, course credit or degree; establish a relationship between this learner and JHU; enroll or register this learner at JHU or in any course offered by JHU; or entitle this learner to access or use resources beyond the online courses provided by Coursera.