CURRENCY CONVERTER PROJECT

A Web-based Currency Converter Using Real-time Exchange Rates

•**By-**

• Utkarsh Jaiswal (utkarshj19@gmail.com)

INTRODUCTION

- A currency converter is an essential tool to convert one currency to another.
- This project provides real-time conversion based on exchange rates fetched from an API.
- The converter allows users to easily convert popular currencies such as USD, INR, GBP, JPY, and more.

OBJECTIVES

- Build a currency converter that fetches real-time exchange rates from an external API.
- Provide users with an easy-to-use interface to convert between multiple currencies.
- Display the converted amount with real-time updates.

FEATURES

- **Live Currency Conversion**: Fetches up-to-date rates from an API.
- **Multiple Currencies Supported**: Supports major currencies like USD, INR, JPY, GBP, and more.
- **User Input for Conversion**: Users input an amount and select the source and target currency
.
- **Responsive Design**: Works across devices (desktops, tablets, and mobile phones).

TECHNOLOGY USED

- **HTML**: Used for the structure and layout of the web page.
- **CSS**: Used for styling the page and making it visually appealing.
- **JavaScript**: Handles API calls, currency conversion logic, and user Interactions.
- **API**: https://v6.exchangerate-api.com/v6/c97e21e7b04e14199d39cce2/latest/USD for fetching real-time currency exchange rates.

ARCHITECTURE

- **User Interface (UI)**: The user enters an amount and selects currencies.
- **API Call**: The JavaScript fetches exchange rates from the external API.
- **Conversion Logic**: JavaScript converts the amount using the fetched exchange rate.
- **Result Display**: The result is displayed to the user, showing the converted currency.

HOW IT WORKS

- **User Input**: The user enters the amount they want to convert.
- **Currency Selection**: The user selects the source and target currencies.
- **API Request**: The JavaScript sends a request to the API to fetch the latest exchange rates.
- **Calculation**: The amount is multiplied by the exchange rate to get the converted value.
- **Display Result**: The result is displayed on the web page.

USER INTERFACE

- Simple and clean layout with input fields for amount and dropdowns for currency sel
- Button to trigger conversion.
- Result section to display the converted value.
- Responsive design to ensure usability on different devices.

CODE SNIPPETS

HTML

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
    <title>Currency Converter</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <div class="container">
        <h2>Currency Converter</h2>
        <div class="converter">
            <input type="number" id="amount" placeholder="Enter
            <select id="fromCurrency"></select>
            <span>to</span>
            <select id="toCurrency"></select>
            <button onclick="convertCurrency()">Convert</button
        </div>
        <p id="result"></p>
    </div>

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

CSS

```css
body {
    font-family: Arial, sans-serif;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    background-color: #f4f4f4;
}

.container {
    text-align: center;
    background: white;
    padding: 20px;
    border-radius: 10px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

.converter {
    margin-top: 20px;
}

input, select, button {
    padding: 10px;
    margin: 5px;
    font-size: 16px;
}

button {
    background: #28a745;
    color: white;
    border: none;
    cursor: pointer;
}

button:hover {
    background: #218838;
```

javascript

```javascript
const apiKey = "c97e21e7b04e14199d39cce2";
const apiUrl = `https://v6.exchangerate-api.com/v6/${apiKey}/latest/USD`;

const majorCurrencies = [
    "USD", "INR", "JPY", "KRW", "RUB", "GBP", "EUR", "AUD", "CAD", "CNY", "CHF"
];

document.addEventListener("DOMContentLoaded", async () => {
    try {
        const response = await fetch(apiUrl);
        const data = await response.json();

        const fromCurrency = document.getElementById("fromCurrency");
        const toCurrency = document.getElementById("toCurrency");

        majorCurrencies.forEach(currency => {
            const option1 = document.createElement("option");
            option1.value = currency;
            option1.textContent = currency;
            fromCurrency.appendChild(option1);

            const option2 = document.createElement("option");
            option2.value = currency;
            option2.textContent = currency;
            toCurrency.appendChild(option2);
        });
        fromCurrency.value = "USD";
        toCurrency.value = "INR";  // Default to INR
    } catch (error) {
        console.error("Error fetching exchange rates:", error);
    }
});

async function convertCurrency() {
    const amount = document.getElementById("amount").value;
    const fromCurrency = document.getElementById("fromCurrency").value;
    const toCurrency = document.getElementById("toCurrency").value;

    if (amount === "" || isNaN(amount)) {
        alert("Please enter a valid amount");
        return;
    }

    try {
        const response = await fetch(`https://v6.exchangerate-api.com/v6/${
        const data = await response.json();
        const rate = data.conversion_rates[toCurrency];
        const convertedAmount = (amount * rate).toFixed(2);

        document.getElementById("result").textContent = `${amount} ${fromCu
    } catch (error) {
        console.error("Error converting currency:", error);
    }
}
```

FUTURE IMPROVEMENTS

- **Support for More Currencies**: Include additional currencies and their rates.
- **Historical Data**: Allow users to view historical exchange rates
- **Conversion History**: Save user conversion history for future reference.
- **Improved UI**: Enhance the user interface with more features like themes and currency symbols.

CONCLUSION

•This currency converter project efficiently provides real-time currency exchange rate conversion.

•It is built using HTML, CSS, and JavaScript, fetching live rates from an API.

•The project is designed to be simple, effective, and user-friendly, and can be extended with additional features in the future.