Project Overview

The **Currency Converter** project is a simple yet fully functional web application built using **HTML**, **CSS**, and **JavaScript**. It allows users to convert one currency to another using real-time exchange rate data fetched from an external API.

This project demonstrates core web development concepts including:

- Structuring content with HTML5.
- Styling with modern CSS3.
- Interacting with live data using Fetch API in JavaScript.
- Dynamically updating the Document Object Model (DOM).

The project's primary goal is to help users easily convert amounts between different world currencies through a clean, user-friendly interface.

Technologies Used

Technology	Purpose
HTML5	To design the structure and layout of the web page.
CSS3	To style the application and make it visually appealing.
JavaScript (ES6)	To add interactivity, fetch data from the API, and handle dynamic updates.
Frankfurter API	To fetch real-time currency exchange rates.
Font Awesome	To include icons for better UI representation.
FlagsAPI	To display country flags corresponding to the selected currencies.

Project Structure

File Name	Description
index.html	Contains the main structure of the web app: input fields, dropdown menus, and buttons.
style.css	Defines the styling of HTML elements, making the design clean, responsive, and visually consistent.
codes.js	Contains the list of currencies and corresponding country codes used to dynamically populate the dropdowns.
app.js	Handles API calls, fetches exchange rates, updates the UI dynamically, and manages event listeners.

Working of the Project

Step 1: Initial Setup (HTML)

The index.html file contains:

- A form that allows users to enter the amount.
- Two dropdown menus labeled **"From"** and **"To"** for selecting currencies.
- A **Get Exchange Rate** button which triggers the conversion.
- A section showing the converted amount.

The logic and content are later linked to style.css for design and app.js/codes.js for functionality.

Step 2: Styling (CSS)

The design emphasizes simplicity:

- The entire interface is centered horizontally and vertically using flexbox.
- Containers have rounded corners and clean borders to achieve a modern UI look.
- The color palette uses a combination of **cream (#f4e4ba)** and **white**, making it pleasant to the eyes.
- Buttons have interactive colors and cursor effects for a smooth user experience.

Step 3: Currency List (codes.js)

This file defines a JavaScript object countryList that maps each **currency code** (like USD, INR, EUR) to its **country code** (like US, IN, FR). This mapping is used to:

- Fill the dropdown options dynamically.
- Update the flag icons beside the currency selections.

Example:

javascript

```
const countryList = {
  USD: "US",
  INR: "IN",
  EUR: "FR",
  // and many more...
};
```

Step 4: Functionality (app.js)

The core logic lies here.

1. **Populating the Dropdowns:**
   Each currency in countryList is added as an option in both the "From" and "To" dropdowns. By default:
   - From = USD
   - To = INR

2. **Flag Updates:**
   When a user selects a currency, the corresponding flag is fetched from **FlagsAPI** and updated beside the dropdown using:

javascript

```
img.src = `https://flagsapi.com/${countryCode}/flat/64.png`;
```

3. **Fetching Exchange Rates:**
   The app makes an asynchronous request to the **Frankfurter API** using:

javascript

```
const BASE_URL = "https://api.frankfurter.app/latest?from=";

const URL = `${BASE_URL}${fromCurr.value}&to=${toCurr.value}`;
```

This API returns a JSON object containing real-time exchange rates between currencies.

4. **Calculating the Conversion:**
   After fetching data, the conversion is done by:

javascript

```
let finalAmount = amtVal * rate;
```

and displayed as:

text

```
1 USD = 80 INR
```

5. **Event Handling:**
   - The **button click** triggers updateExchangeRate().
   - On **page load**, the default conversion (USD → INR) is displayed automatically.

API Used

The project integrates the free Frankfurter Exchange Rate API. This API provides up-to-date foreign exchange rates for several global currencies.

Sample Response:

json

```
{
  "amount": 1,
  "base": "USD",
  "date": "2025-12-26",
  "rates": {
    "INR": 83.15
  }
}
```

Features

- Converts between any two world currencies in real time.
- Displays country flags dynamically based on the selected currency.
- Responsive design suitable for desktop and mobile browsers.
- Default conversion (USD → INR) on load.
- Input validation to ensure valid amount entries.

Future Enhancements

Some potential improvements could include:

- Adding **historical exchange rate charts** using APIs like Chart.js.
- Allowing **multiple currency conversions** at once.
- Implementing **dark mode** for better user experience.
- Adding **offline currency rate caching** using local storage.
- Making the UI more interactive using animations or transitions.

Conclusion

The **Currency Converter** mini project is a perfect demonstration of integrating frontend design with live data using APIs. It showcases how basic web technologies can work together to create a dynamic, responsive, and practical application.

This project enhances both **programming logic** and **web development skills**, making it an ideal addition to a portfolio for computer science students or aspiring web developers.