

ASSIGNMENT OUTLINE: CURRENCY CONVERTER APPLICATION

Objective: Develop a currency converter application that allows users to convert between multiple currencies using real-time exchange rates fetched from a public API.

1. Project Setup

- Duration: 1 week
- Tools: HTML, CSS, JavaScript, Fetch API or Axios, Public currency exchange API (e.g., ExchangeRate-API, OpenExchangeRates)
- Prerequisites: Basic understanding of HTML, CSS, JavaScript, and API interaction

2. Requirements

- User interface with input fields for the amount, source currency, and target currency.
- Dropdowns or auto-complete suggestions for selecting currencies.
- Real-time exchange rate fetching from a public API.
- Display the converted amount dynamically.
- Error handling for invalid inputs and API failures.
- Responsive design to ensure compatibility with various devices.

3. Features

- Input field for the amount to be converted.

- Dropdowns for selecting the source and target currencies.
- Button to trigger the conversion.
- Display area for the converted amount.
- Fetch exchange rates from a public API.
- Error messages for incorrect inputs or failed API calls.

4. Detailed Steps

1. Initial Setup:

- Create a new directory for the project.
- Set up an HTML file with a basic structure.
- Link a CSS file for styling.
- Link a JavaScript file for functionality.

2. Design the User Interface (UI):

- Create a form with input fields for the amount, source currency, and target currency.
- Use dropdowns for currency selection, populated with a list of common currencies.
- Add a button to trigger the conversion.
- Add a display area for the conversion result.

3. Fetch Exchange Rates:

- Register for an API key from a public currency exchange API provider.

- Write a function in JavaScript to fetch exchange rates using the Fetch API or Axios.

- Handle responses and errors appropriately.

4. Conversion Logic:

- Write a function to perform the currency conversion using the fetched exchange rates.

- Update the UI dynamically to display the converted amount.

5. Error Handling:

- Implement checks for invalid inputs (e.g., non-numeric values, empty fields).

- Display error messages for invalid inputs.

- Handle API errors and display appropriate messages to the user.

6. Styling and Responsiveness:

- Apply CSS to style the form and display area.

- Ensure the layout is responsive and works well on different screen sizes.

7. Testing:

- Test the application with different input values and currency pairs.

- Verify the application handles errors gracefully.

5. Bonus Features (Optional)

- Allow users to swap the source and target currencies with a single click.

- Display historical exchange rates using charts.
- Enable users to select multiple currencies and see all conversion results at once.
- Save the user's preferred currencies and amount using local storage.

6. Submission

- Submit the project folder containing all source files (HTML, CSS, JavaScript).
- Include a README file with instructions on how to run the application and any relevant notes.

7. Evaluation Criteria

- Functionality: Does the application meet all the specified requirements?
- Code Quality: Is the code well-organized, commented, and following best practices?
- UI/UX: Is the user interface intuitive, responsive, and visually appealing?
- Error Handling: Are errors handled gracefully and informative messages displayed to the user?
- Bonus Features: Are any of the bonus features implemented effectively?

This outline provides a clear path for the intern to follow, ensuring they cover all essential aspects of building a functional and user-friendly currency converter application.