“Currency Converter Web Application”

Minor Project Report

Submitted by

Md Adil Raza (2022-310-103)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY
(COMPUTER SCIENCE & ENGINEERING)

Under the supervision of

Dr. Sherin Zafar

Department of Computer Science & Engineering
School of Engineering Sciences & Technology

JAMIA HAMDARD

(Deemed to be University)

New Delhi-110062

2025

.

DECLARATION

I, **Md Adil Raza** a student of **BTech CSE, Enrolment No: (2022-310-103)** hereby declare that the Project/Dissertation entitled **“Currency Converter Web Application”** Hamdard, New Delhi in partial fulfillment of the requirement for the award of the degree of **B.Tech CSE,** is my original work and has not been submitted anywhere else for the award of any Degree, Diploma, Associateship, Fellowship or other similar title or recognition.

Md Adil Raza (2022-310-103)

Date:

Place: Jamia Hamdard

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisor, **Dr. Sherin Zafar**, for his expert guidance, valuable insights, and continuous support throughout the course of this project. His encouragement and mentorship were instrumental in helping me shape this project into a successful learning experience.

I am also thankful to all the faculty members and staff of the Department of Computer Science & Engineering, **Jamia Hamdard**, for creating a supportive academic environment and for providing the resources needed to carry out this work.

Lastly, I extend heartfelt thanks to my friends and family, whose motivation, encouragement, and unwavering support kept me focused and inspired during the development of this project.

Md Adil Raza

TABLE OF CONTENT

TITLE

Currency Converter Web Application

1. OBJECTIVE

The primary objective of this project is to design and develop a **Currency Converter Web App** that enables users to convert amounts between different currencies using real-time exchange rates. The application aims to provide a fast, reliable, and user-friendly platform for currency conversion without the need for manual calculations or third-party tools.

Specific objectives include:

- Implementing a responsive front-end using HTML, CSS, and JavaScript.
- Integrating a public API to fetch live currency exchange rates.
- Allowing users to select source and target currencies from an intuitive interface.
- Providing instant and accurate conversion results.
- Ensuring cross-browser and multi-device compatibility.

This project focuses on enhancing the understanding of client-side web technologies and demonstrates the ability to work with real-world APIs in a practical application.

2. ABSTRACT

In today's interconnected global economy, real-time currency conversion has become a crucial tool for travelers, businesses, and online shoppers. This project presents a **Currency Converter Web Application** that enables users to convert amounts between various international currencies using live exchange rates. Developed using **HTML, CSS, and JavaScript**, the application features a responsive and user-friendly interface, ensuring compatibility across different devices and screen sizes.

The system utilizes a public API to fetch up-to-date currency rates and instantly performs conversions based on user input. Additional features, such as country flags and error handling for API failures, enhance the user experience. The absence of a backend database or server makes the application lightweight and fast, with all processing handled on the client side.

This project demonstrates practical knowledge of front-end web development, asynchronous data handling via APIs, and responsive UI design. It serves as a useful and accessible tool for anyone needing quick and accurate currency conversions in real time.

3. INTRODUCTION

With people traveling more, shopping online from different countries, and businesses working globally, converting currency has become an everyday need. Instead of checking exchange rates manually or using complicated tools, a simple, easy-to-use currency converter can make a big difference. That's exactly what this project is about — creating a **Currency Converter Web Application** that helps users quickly convert amounts between different currencies in real time.

The app is built using basic but powerful web technologies: **HTML, CSS, and JavaScript**. It connects to a public API to fetch the latest exchange rates and does the conversion instantly when the user enters an amount and selects the currencies. It's fast, lightweight, and works right in the browser — no installation or backend server needed.

The goal was to build something useful while also getting hands-on experience with real-world web development skills, like working with APIs, designing a clean user interface, and handling user input dynamically. This project is not just a coding exercise — it's a practical tool that could actually help people in their day-to-day activities.

In this report, you'll find everything from how the app works to the choices made during development and the challenges faced along the way.

4. PROBLEM STATEMENT

In today’s global environment, individuals and businesses frequently deal with foreign currencies. However, accessing accurate and up-to-date exchange rates, and performing currency conversions quickly, can be challenging without the right tools. Most online converters are either cluttered, require internet-heavy platforms, or are tied to advertisements and sign-ups.

There is a need for a **simple, fast, and user-friendly web application** that allows users to convert currencies in real time using live exchange rates. This project addresses that need by creating a lightweight, browser-based solution that performs conversions instantly without relying on a backend server or database.

5. Scope of the Project

This tool is designed mainly for students and beginners The **Currency Converter Web App** is designed to provide real-time currency conversion in a simple and efficient way. The project focuses on front-end web development using **HTML, CSS, and JavaScript**, and integrates with a public API to fetch up-to-date exchange rates.

Key features within the project scope:

- Convert currency amounts instantly using real-time data.
- Select source and target currencies from a list.
- Display national flags for better user experience.
- Ensure the application is responsive and works on different devices and browsers.

Out of scope:

- User authentication or login features.
- Historical exchange rate tracking.
- Backend storage or database support.

6. Software Requirements Specification (SRS)

6.1 Introduction

6.1.1 Purpose

The purpose of this document is to specify the software requirements for the Currency Converter Web Application. It outlines the intended features, behavior, constraints, and external interfaces of the system, serving as a guide for design, development, and testing.

6.1.2 Scope

This project is a browser-based currency converter that allows users to convert amounts between different international currencies using real-time exchange rates fetched from a public API. The application is built using HTML, CSS, and JavaScript, and runs entirely on the client side, with no need for a backend server or database.

6.1.3 Definitions, Acronyms, and Abbreviations

- API: Application Programming Interface
- HTML: HyperText Markup Language
- CSS: Cascading Style Sheets
- JS: JavaScript
- UI: User Interface
- IEEE: Institute of Electrical and Electronics Engineers

6.1.4 References

- FawazAhmedCurrencyAPI: https://github.com/fawazahmed0/currency-api
- MDN Web Docs – JavaScript: https://developer.mozilla.org
- W3Schools: https://www.w3schools.com
- Font Awesome Icons: https://fontawesome.com
- FlagCDN: https://flagcdn.com

6.1.5 Overview

This document is divided into sections that describe the system's overall functionality, specific requirements, constraints, and interfaces. It is intended for developers, testers, and academic evaluators.

6.2 Overall Description

6.2.1 Product Perspective

The Currency Converter Web App is a self-contained, standalone client-side application. It does not rely on any server-side logic or database. It uses an external API to fetch real-time exchange rate data.

6.2.2 Product Functions

- Take user input for the amount to be converted.
- Allow selection of source and target currencies.
- Fetch the latest exchange rates from the API.

- Display the converted amount.
- Show flags for selected countries for better user experience.

6.2.3 User Characteristics

The target users are individuals needing quick and accurate currency conversions. No technical expertise is required. Users should have basic internet and browser usage knowledge.

6.2.4 Constraints

- Requires internet access to fetch exchange rates.
- Dependent on third-party API availability.
- No support for historical data or offline use.
- Runs only on modern browsers with JavaScript enabled.

6.2.5 Assumptions and Dependencies

- The API will provide correct and up-to-date exchange rates.
- The user will access the application on a supported device and browser.
- The frontend will handle all logic without backend support.

6.3 Specific Requirements

6.3.1 Functional Requirements

- **FR1:** The system shall allow users to input a numeric amount.
- **FR2:** The system shall provide dropdowns to select the source and target currencies.

- **FR3:** The system shall fetch exchange rates from the external API on user action or page load.
- **FR4:** The system shall calculate and display the converted amount.
- **FR5:** The system shall update currency flag images based on selection.

6.3.2 Non-Functional Requirements

- **NFR1:** The system shall respond to conversion requests within 2–3 seconds.
- **NFR2:** The application UI shall be mobile-friendly and responsive.
- **NFR3:** The system shall display appropriate messages in case of API failure.
- **NFR4:** The application shall load within 5 seconds on a standard internet connection.

6.3.3 Hardware Requirements

Minimum Requirements:

- **Processor**: Dual-core 1.0 GHz or higher
- **RAM**: 2 GB
- **Storage**: No installation required (only browser access needed)
- **Display**: 1024×768 resolution or higher
- **Internet**: Required for real-time API communication

Recommended Requirements:

- **Processor**: Quad-core 2.0 GHz or higher
- **RAM**: 4 GB or more
- **Display**: 1366×768 resolution or higher
- **Browser**: Latest version of Chrome, Firefox, Edge, or Safari

- **Internet Speed**: At least 1 Mbps for smooth performance

6.4 External Interface Requirements

6.4.1 User Interface

- The user interacts with the application through a web browser.
- Features include:
- Input field for amount
- Dropdown menus to select source and target currencies
- Flag icons for selected countries
- Button to fetch exchange rate
- Output display showing the converted amount
- The interface is responsive and works on desktops, tablets, and mobile devices.

6.4.2 Software Interface

Currency API:

- **URL:** https://cdn.jsdelivr.net/npm/@fawazahmed0/currency-api@latest/v1/currencies/
- **Purpose:** Fetches the latest currency exchange rates in JSON format.
- **Method:** HTTP GET

Flag API:

- **URL format:** https://flagsapi.com/{country_code}/flat/64.png

- **Purpose:** Displays national flags based on selected currencies.

Font Awesome CDN:

- Provides icons used in the UI (e.g., exchange arrows).

6.4.3 Hardware Interface

- The app runs on standard computing devices including:
- PCs, laptops, tablets, and smartphones

Requires:

- Internet connection
- A browser that supports JavaScript and CSS

6.4.4 Communication Interfaces

- All interactions (form inputs, button clicks, dropdown selections) are handled through JavaScript event listeners.
- There is no need for sockets, message queues, or backend communication.

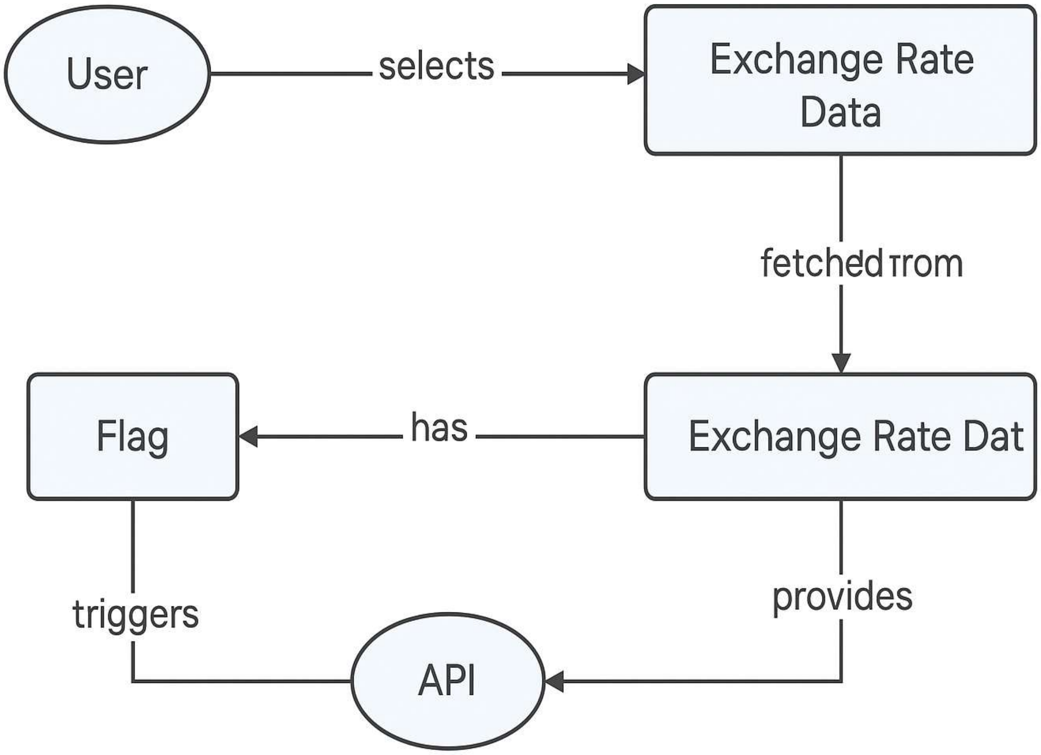
7. Sample Test Cases

Test Case ID	Description	Input	Expected Output	Actual Output	Status
TC01	Convert 100 USD to INR	Amount: 100 From: USD To: INR	Show converted amount (e.g., 100 USD = 8300 INR)	Correct result shown	Pass
TC02	Empty amount field	Amount: (empty) From: USD To: INR	Default to 1 and show result	Defaulted to 1, result shown	Pass
TC03	Same currency conversion	Amount: 50 From: USD To: USD	50 USD = 50 USD	Displayed correctly	Pass
TC04	No internet / API failure	Any amount and currency (disconnect internet)	Show error: 'Failed to fetch exchange rate'	Error message displayed	Pass

8. Data Flow Diagram

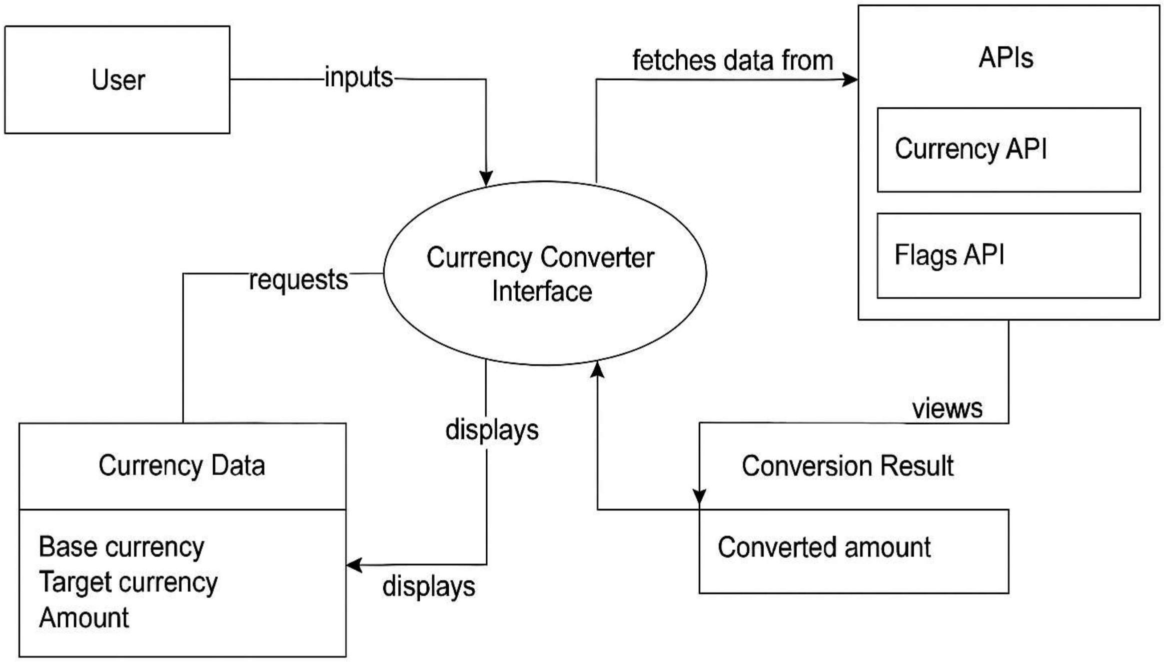
8.1 Entity Relationship Diagram

The following ER diagram shows the relationship between the key components in the currency converter project:

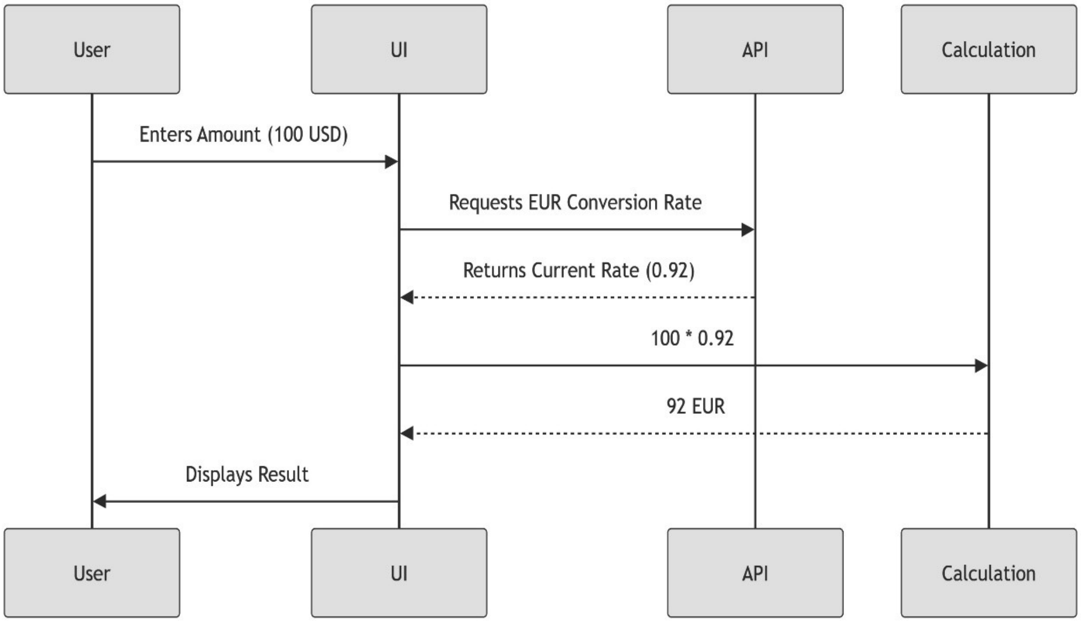


8.2 Data Flow Diagram

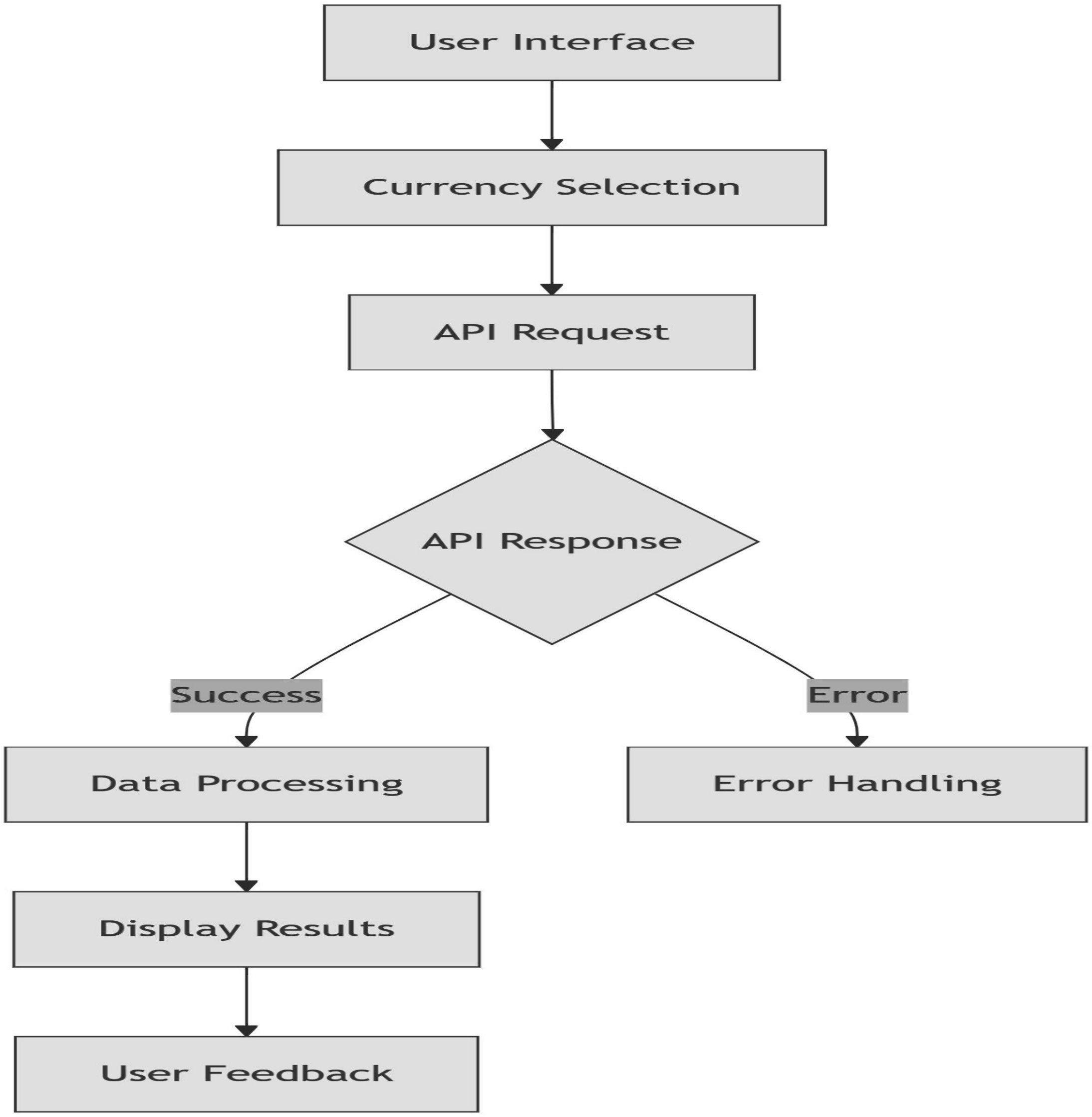
The following Data Flow Diagram (DFD) illustrates the flow of data through the currency converter system, from user input to fetching API results and displaying the final converted value.



8.3 Flow Diagram



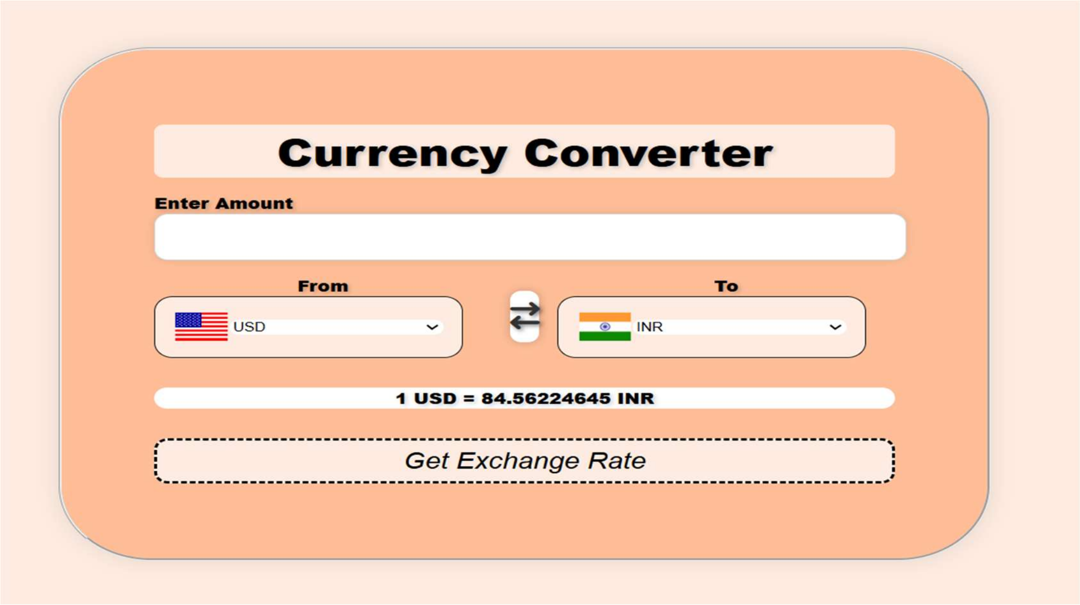
8.4 System Architecture



8. Snapshots of Output

8.1 Home Page Interface

Figure 1: Home page of the Currency Converter Web App



Currency Converter

Enter Amount

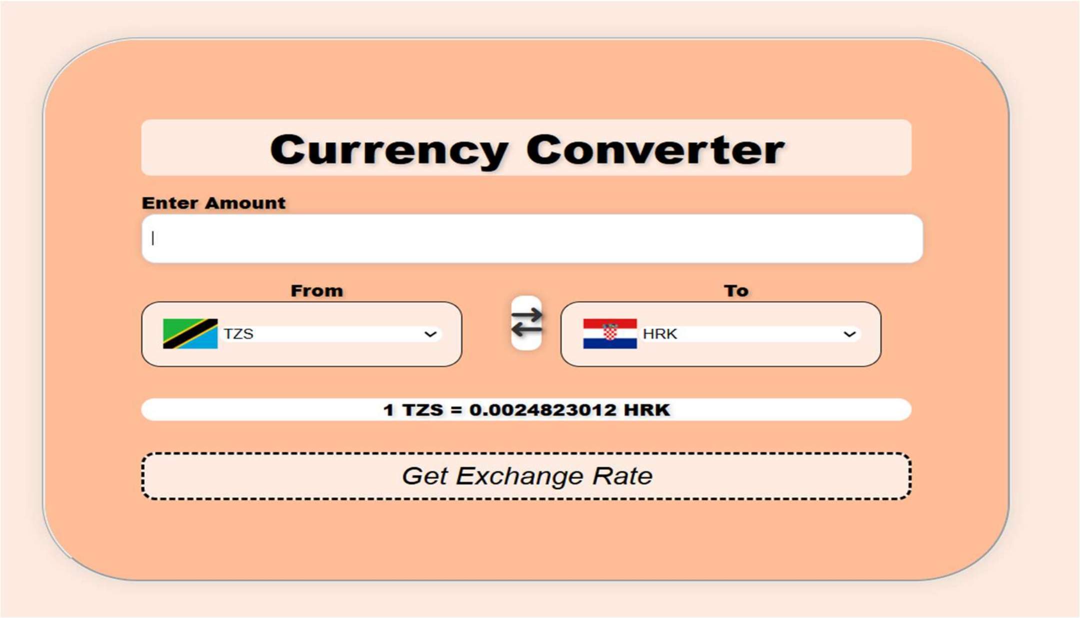
From: USD

To: INR

1 USD = 84.56224645 INR

8.2 Currency Selection and Flag Display

Figure 2: Currency selection with corresponding flags



Currency Converter

Enter Amount

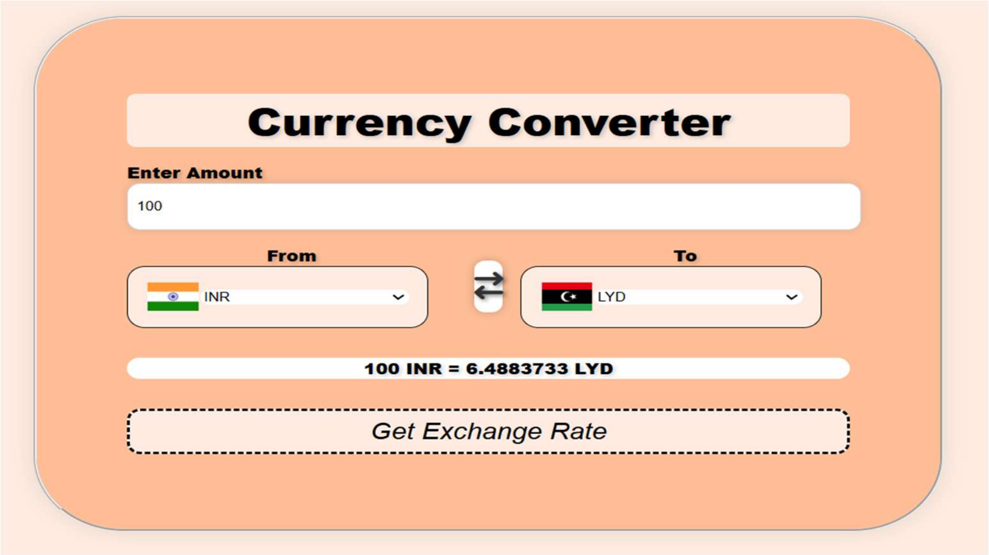
From: TZS

To: HRK

1 TZS = 0.0024823012 HRK

8.3 Conversion Result Display

Figure 3: Output displaying the converted currency value



Currency Converter

Enter Amount: 100

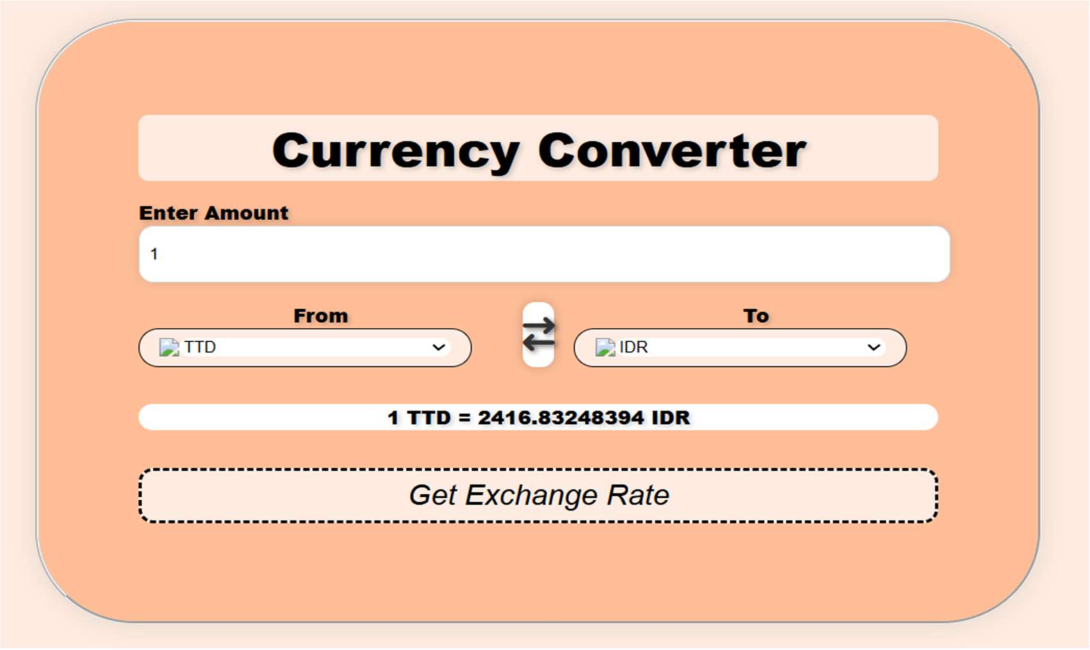
From: INR

To: LYD

100 INR = 6.4883733 LYD

8.4 API Error Handling

Figure 4: Error message on API failure



Currency Converter

Enter Amount: 1

From: TTD

To: IDR

1 TTD = 2416.83248394 IDR

9. CONCLUSION

The Currency Converter Web App project successfully demonstrates how real-time exchange rates can be integrated into a user-friendly interface to facilitate seamless currency conversions. By leveraging a third-party API and implementing responsive design principles, the application provides users with accurate, fast, and intuitive conversion results.

Throughout the development process, key software engineering principles such as modularity, usability, and error handling were emphasized to ensure reliability and maintainability. The project not only enhances understanding of web technologies like HTML, CSS, JavaScript, and API integration, but also highlights the practical importance of financial applications in today's globally connected economy.

In conclusion, this project lays a solid foundation for future enhancements such as multi-language support, historical exchange rate tracking, and user login features, thus opening avenues for further academic and real-world application.

10. LIMITATIONS

- **API Dependency**

  The app relies on a third-party exchange rate API. If the API service is down or rate-limited, the application cannot fetch or display live conversion rates.

- **Internet Requirement**

  The application requires an active internet connection to function, as exchange rates are fetched in real time. It cannot operate in offline mode.

- **No Historical Data**

  The app currently does not support viewing historical exchange rates, which could be useful for financial analysis or comparisons over time.

- **No Authentication or User Profiles**

  There is no login system or personalized settings for users. All conversions are done anonymously without the option to save preferences or history.

- **Limited Currency Set**

  Only a predefined set of major currencies are supported. It may not include all global currencies or exotic pairs.

- **Basic Error Handling**

  While the app shows basic error messages (e.g., for empty input or API failures), it lacks advanced validation and fallback mechanisms.

11. FUTURE ENHANCEMENTS

- **Offline Support**

  Implement caching mechanisms to allow currency conversions even without an internet connection using previously fetched exchange rates.

- **Historical Exchange Rates**

  Add a feature to view and compare historical exchange rates for selected currencies over specific time periods.

- **User Authentication and Profiles**

  Introduce login functionality to let users save their conversion history, preferred currencies, and personalized settings.

- **Mobile App Integration**

  Develop a mobile version of the web app using frameworks like React Native or Flutter for better accessibility on smartphones.

- **Dark Mode and Accessibility Improvements**

  Add themes like dark mode and improve accessibility for visually impaired users by complying with WCAG standards.

- **Multi-language Support**

  Translate the app interface into multiple languages to reach a more global user base.

11. BIBLIOGRAPHY

- W3Schools. *HTML, CSS, and JavaScript Tutorials*.

  Available at: https://www.w3schools.com

- Mozilla Developer Network (MDN). *Web Technology Documentation*.

  Available at: https://developer.mozilla.org

- ExchangeRate-API. *Currency Exchange Rate API Documentation*.

  Available at: https://www.exchangerate-api.com

- Stack Overflow. *Developer Q&A and Solutions*.

  Available at: https://stackoverflow.com

- GeeksforGeeks. *Programming Tutorials and Code Snippets*.

  Available at: https://www.geeksforgeeks.org

- GitHub. *Open-source Project Repositories and Resources*.

  Available at: https://github.com

- IEEE. *Software Requirements Specification Guidelines*.

  Available at: https://www.ieee.org