CURRENCY CONVERTER APPLICATION

A PROJECT REPORT

Submitted by

SUBIKSA G R (2303811724322110)

in partial fulfilment of requirements for the award of the course

CGB1201 – JAVA PROGRAMMING

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

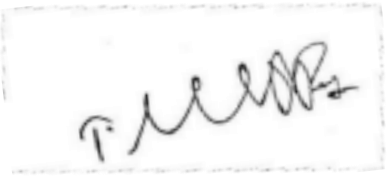
SAMAYAPURAM – 621 112 DECEMBER, 2024

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report on **"CURRENCY CONVERTER APPLICATION"** is the bonafide work of **SUBIKSA G R (2303811724322110)** who carried out the project work during the academic year 2024 - 2025 under my supervision.



Signature

Dr. T. AVUDAIAPPAN M.E., Ph.D.,

HEAD OF THE DEPARTMENT,

Department of Artificial Intelligence,
K. Ramakrishnan College of Technology,
Samayapuram, Trichy-621 112.

Signature

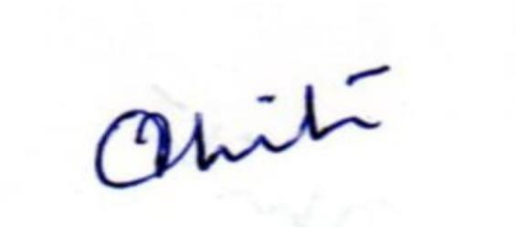
Mrs. S. GEETHA M.E.,

SUPERVISOR,

Department of Artificial Intelligence,
K. Ramakrishnan College of technology,
Samayapuram, Trichy-621 112.

Submitted for the viva-voce examination held on 3.12.24.

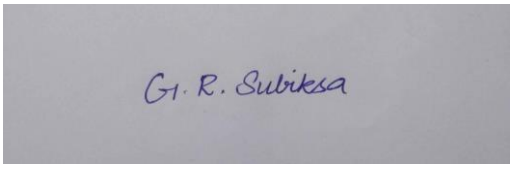
INTERNAL EXAMINER



EXTERNAL EXAMINER

DECLARATION

I declare that the project report on "**CURRENCY CONVERTER APPLICATION**" is the result of original work done by me and best of my knowledge, similar work has not been submitted to "**ANNA UNIVERSITY CHENNAI**" for the requirement of Degree of **BACHELOR OF TECHNOLOGY**. This project report is submitted on the partial fulfilment of the requirement of the award of the **CGB1201 – JAVA PROGRAMMING.**



Signature

SUBIKSA G R

Place: Samayapuram

Date: 3/12/2024

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and indebtedness to our institution, **"K. Ramakrishnan College of Technology (Autonomous)",** for providing us with the opportunity to do this project.

I extend our sincere acknowledgment and appreciation to the esteemed and honourable Chairman, **Dr. K. RAMAKRISHNAN, B.E.,** for having provided the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director, **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding our project and offering an adequate duration to complete it.

I would like to thank **Dr. N. VASUDEVAN, M.TECH., Ph.D.,** Principal, who gave the opportunity to frame the project to full satisfaction.

I thank **Dr.T. AVUDAIAPPAN, M.E., Ph.D.**, Head of the Department of **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**, for providing her encouragement in pursuing this project.

I wish to convey our profound and heartfelt gratitude to our esteemed project guide **Mrs. S. GEETHA M.E**., Department of **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**, for her incalculable suggestions, creativity, assistance and patience, which motivated us to carry out this project.

I render our sincere thanks to the Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

VISION OF THE INSTITUTION

To serve the society by offering top-notch technical education on par with global standards.

MISSION OF THE INSTITUTION

- Be a centre of excellence for technical education in emerging technologies by exceeding the needs of industry and society.
- Be an institute with world class research facilities.
- Be an institute nurturing talent and enhancing competency of students to transform them as all- round personalities respecting moral and ethical values.

VISION AND MISSION OF THE DEPARTMENT

To excel in education, innovation and research in Artificial Intelligence and Data Science to fulfill industrial demands and societal expectations.

Mission 1: To educate future engineers with solid fundamentals, continually improving teaching methods using modern tools.

Mission 2: To collaborate with industry and offer top-notch facilities in a conductive learning environment.

Mission 3: To foster skilled engineers and ethical innovation in AI and Data Science for global recognition and impactful research.

Mission 4: To tackle the societal challenge of producing capable professionals by instilling employability skills and human values.

PROGRAM EDUCATIONAL OBJECTIVES (PEOS)

PEO 1: Compete on a global scale for a professional career in Artificial Intelligence and Data Science.

PEO 2: Provide industry-specific solutions for the society with effective communication and ethics.

PEO 3: Hone their professional skills through research and lifelong learning initiatives.

PROGRAM OUTCOMES

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

- **PSO 1:** Capable of working on data-related methodologies and providing industryfocussed solutions.
- **PSO2:** Capable of analysing and providing a solution to a given real-world problem by designing an effective program.

ABSTRACT

This project focuses on the design and development of a real-time currency converter application, enabling users to seamlessly convert amounts between different currencies. The application aims to provide accurate and reliable currency conversion by fetching the latest exchange rates from trusted sources. It is designed to address the needs of diverse users, including travelers, business professionals, and individuals managing international transactions, offering a quick, efficient, and userfriendly solution. Key features include a simple interface, real-time updates, and the ability to handle multiple currencies. By integrating modern development tools and APIs, the application ensures high accuracy, responsiveness, and accessibility. This project highlights the significance of leveraging technology to simplify financial transactions and enhance user convenience. . This currency converter application is designed to deliver seamless currency conversion services, addressing the needs of modern users. The application sources the latest exchange rates from trusted providers, offering real-time accuracy and reliability.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

In today's globalized world, currency conversion plays a crucial role in facilitating international trade, travel, and financial transactions. A currency converter application is a digital tool designed to simplify the process of converting one currency to another, providing users with real-time exchange rate information and accurate conversion results. This application is particularly beneficial for travelers, businesses, and individuals managing cross-border transactions, offering a reliable and efficient solution to currency-related challenges.

1.2 OBJECTIVE

The primary objectives of the currency converter application are as follows:

1. To develop a reliable tool for converting amounts between different currencies in real time.
2. To integrate APIs for fetching the latest exchange rates from trusted and reliable sources.
3. To provide accurate and precise currency conversion results for users.
4. To design a user-friendly interface that ensures ease of use for a diverse range of users.
5. To offer a convenient solution for travelers, businesses, and individuals handling international financial transactions.

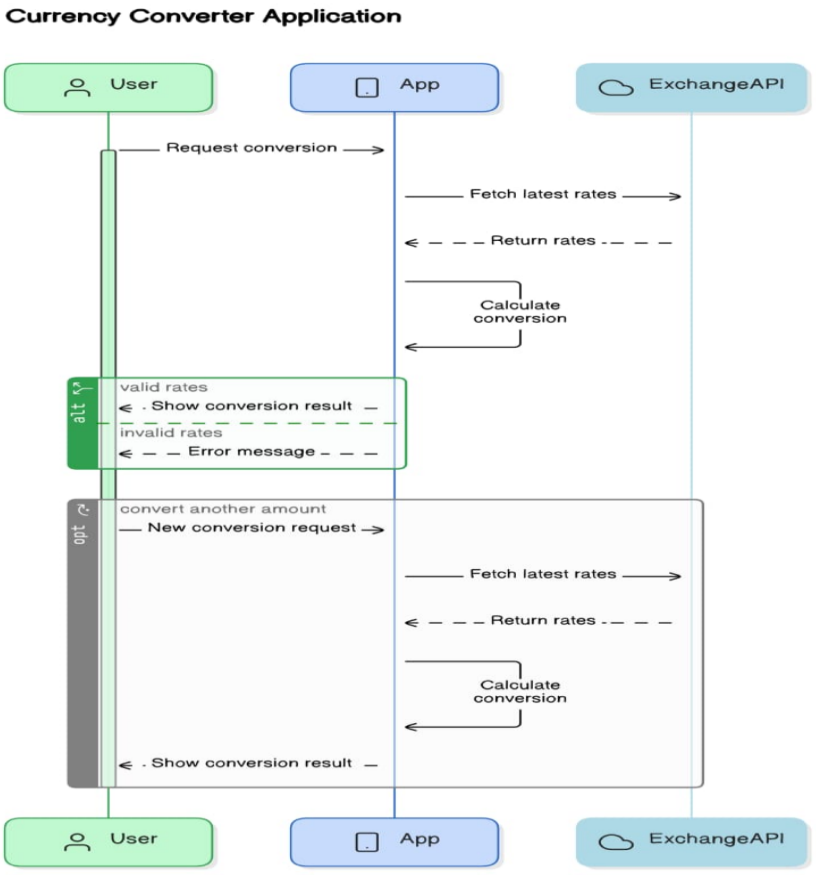
This project aims to address the challenges of currency conversion with an effective and modern solution, enhancing user experience and streamlining financial process .

CHAPTER 2
PROJECT METHODOLOGY

2.1 PROPOSED WORK

The proposed work for the Currency Converter Application involves the design and development of a reliable, efficient, and user-friendly tool that provides real-time currency conversions. The application aims to cater to individuals needing accurate exchange rate information for travel, business, or everyday transactions.

2.2 BLOCK DIAGRAM



CHAPTER 3
JAVA PROGRAMMING CONCEPTS

3.1 IMPLEMENTATION OF JAVA CONCEPTS

1. Classes and Objects

Java classes and objects help organize the application's functionality. The CurrencyConverter class, which handles conversion logic, and the ExchangeRateFetcher, responsible for fetching real-time exchange rates via API calls. The Currency class represents details like currency codes

2. Exception Handling

Exception handling ensures the application runs smoothly even when errors occur. Exception handling plays a critical role in ensuring robust performance. while custom exceptions handle scenarios such as invalid currency codes or negative input amounts.

3. API Integration

To get real-time exchange rates. API integration is implemented using Java's HttpURLConnection or third-party libraries like OkHttp, enabling the application to fetch the latest exchange rates from trusted sources.

4. User Interface

Java provides libraries like Swing**.** The user interface is designed using Java Swing to provide a seamless experience. It includes input fields for the amount, dropdowns for selecting currencies, and a button to perform conversions.

CHAPTER 4

MODULE DESCRIPTION

4.1 USER INTERACTION

This module manages all interactions between the user and the application, serving as the front-end interface. It captures user inputs, provides feedback, and ensures the application is user-friendly**.**

Key Functions:

- startApplication(): Sets up the application environment and introduces the user to the converter.
- getInputAmount(): Captures the amount to be converted, ensuring user input is obtained correctly.
- selectCurrencies(): Displays a list of available currencies for the user to select source and target currencies.

4.2 CONVERSION PROCESSING

This is the core module responsible for performing the actual currency conversion. It validates inputs, retrieves exchange rates, and calculates the conversion.

Key Functions:

- fetchExchangeRates(): Retrieves up-to-date exchange rates from an API or database, ensuring accurate conversions.
- calculateConversion(): Applies the formula:

  text{Converted Amount} = \text{Amount} \times \text{Exchange Rate}

4.3 DATA MANAGEMENT

This module deals with storing, retrieving, and updating exchange rates. It allows the application to function even when offline by using cached data.

Key Functions:

- updateExchangeRates(): Fetches and updates exchange rates periodically.
- storeRatesLocally(Map<String, Double> rates): Saves exchange rates to local storage for offline access.

4.4 ERROR HANDLING

This module ensures the application handles unexpected situations gracefully, such as invalid inputs or connectivity issues.

Key Functions:

- handleInvalidInputError(): Provides feedback if the user enters an invalid amount or selects invalid currencies.
- handleAPIFetchError(): Manages issues encountered while fetching exchange rates (e.g., network errors).

4.5 SCHEDULER

This module automates the process of updating exchange rates, ensuring that the application always has the latest data without requiring manual intervention.

Key Functions:

- scheduleRateUpdates(): Schedules periodic updates (e.g., every hour or day) for exchange rates.
- triggerRateUpdateNow(): Allows the user or administrator to manually fetch the latest rate

CHAPTER 5 CONCLUSION

The Currency Converter Application effectively demonstrates the use of Java programming concepts in a practical scenario. By combining object-oriented programming, API integration, exception handling, and GUI development, the application provides a user-friendly solution for real-time currency conversion. The project addresses challenges such as accurate exchange rate retrieval, user input validation, and error management, ensuring a robust and reliable tool for users. The integration of Java Swing enhances the user experience by offering an intuitive interface, while HTTP networking libraries enable seamless access to real-time exchange rate data from APIs. This project not only meets its objective of simplifying currency conversions for travel or everyday transactions but also serves as a foundation for future enhancements such as offline functionality, support for multiple languages, and advanced financial features.

REFERENCES:

1. Books:

Herbert Schildt, Java: The Complete Reference, McGraw Hill.

Cay S. Horstmann, Core Java Volume I - Fundamentals, Pearson Education.

2. Websites:

Official Java Documentation: https://docs.oracle.com/

Java Swing Tutorials: https://www.javatpoint.com/java-swing

API Integration Guide: https://www.baeldung.com/java-httpclient

APPENDICES

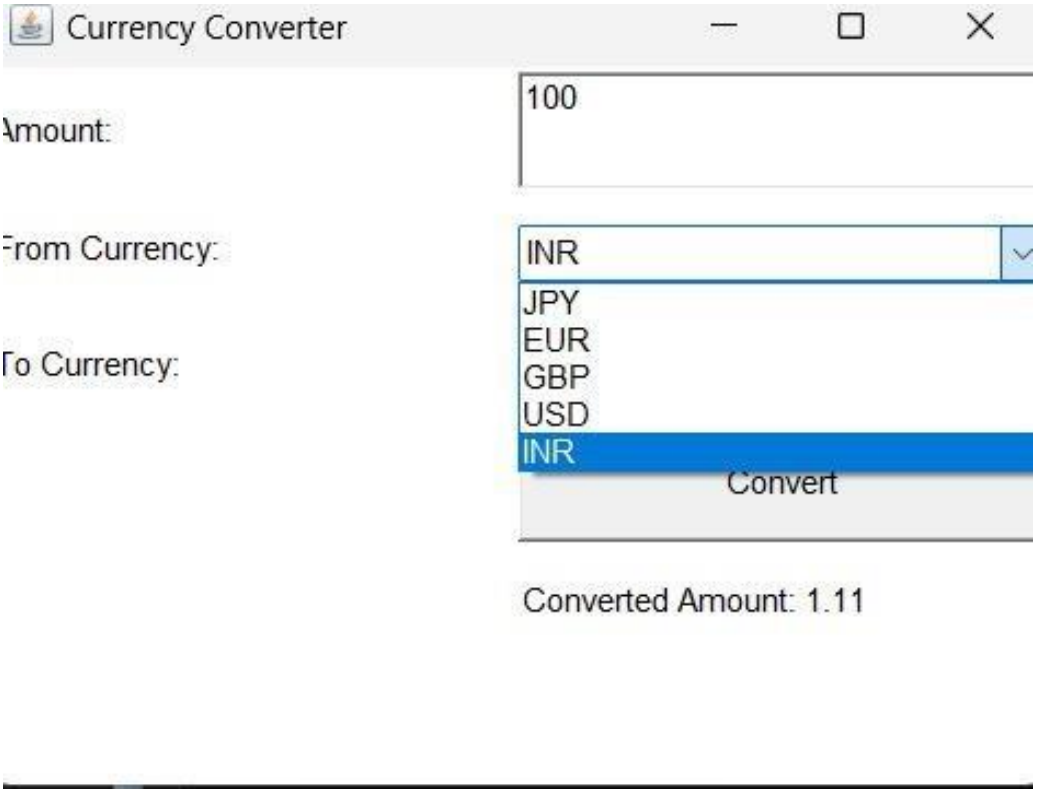
APPENDIX A – SOURCE CODE

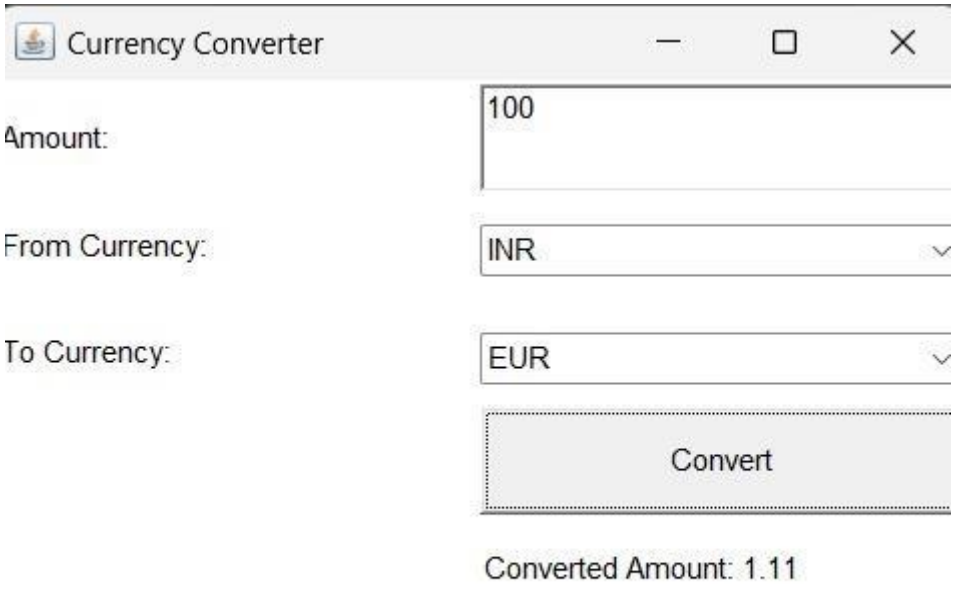
```
import java.awt.*; import
java.awt.event.*; import
java.util.HashMap;


public class CurrencyConverter extends Frame {


    private final HashMap<String, Double> exchangeRates;


    private TextField amountField;
private Choice sourceCurrency;     private
Choice targetCurrency;     private Label
resultLabel;
     public CurrencyConverter() {
exchangeRates = new HashMap<>();
initializeExchangeRates();

setupUI();


        addWindowListener(new WindowAdapter() {
public void windowClosing(WindowEvent we) {
                System.exit(0);
            }
        });
    }      private void
initializeExchangeRates() {
exchangeRates.put("USD", 1.0);
exchangeRates.put("EUR", 0.92);
exchangeRates.put("INR", 82.67);
exchangeRates.put("GBP", 0.78);
exchangeRates.put("JPY", 130.5);
    }


    private     void     setupUI()     {
setTitle("Currency         Converter");
setSize(400, 300);
        setLayout(new GridLayout(6, 2));


        Label amountLabel = new Label("Amount:");
amountField = new TextField();


        Label sourceLabel = new Label("From Currency:");
sourceCurrency = new Choice();
        for (String currency : exchangeRates.keySet()) {
sourceCurrency.add(currency);
        }
```

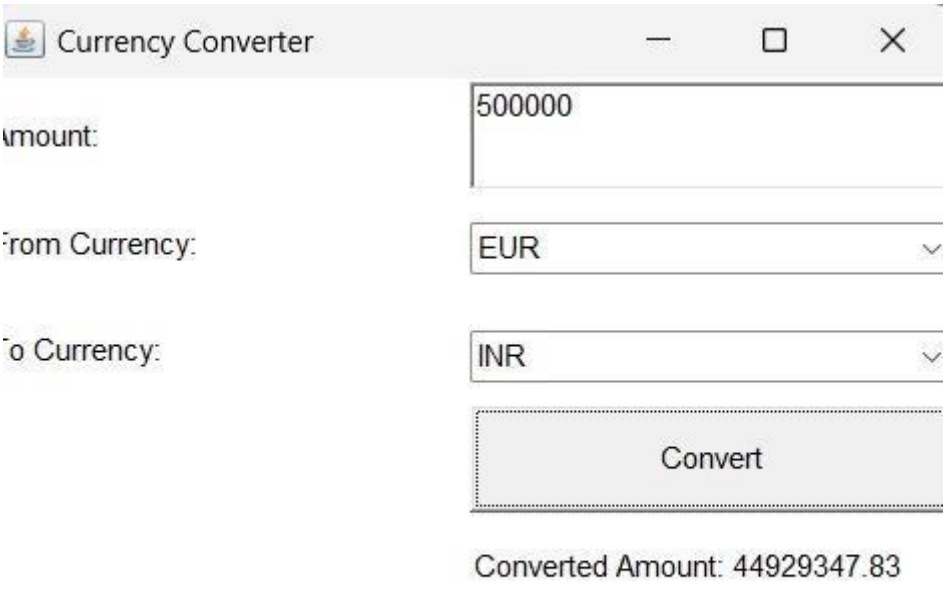
```
        Label targetLabel = new Label("To Currency:");
targetCurrency = new Choice();
        for (String currency : exchangeRates.keySet()) {
targetCurrency.add(currency);
        }

        Button convertButton = new Button("Convert");
resultLabel = new Label("Converted Amount: ");
         convertButton.addActionListener(new ActionListener()
{             public void actionPerformed(ActionEvent e) {
convertCurrency();
            }
        });

        add(amountLabel);
add(amountField);
add(sourceLabel);
add(sourceCurrency);
add(targetLabel);
add(targetCurrency);
add(new Label());
add(convertButton);         add(new
Label());         add(resultLabel);
    }

    private void convertCurrency() {
try {
            double amount = Double.parseDouble(amountField.getText());
            String from = sourceCurrency.getSelectedItem();
            String to = targetCurrency.getSelectedItem();

            if (from.equals(to)) {
                showErrorDialog("Source and target currencies must be
different.");
                return;
            }
             double convertedAmount = calculateConversion(amount, from, to);
resultLabel.setText("Converted    Amount:    "    +    String.format("%.2f",
convertedAmount));
        } catch (NumberFormatException ex) {
            showErrorDialog("Please enter a valid numeric amount.");
        }
    }

    private double calculateConversion(double amount, String from, String
to) {
        double fromRate = exchangeRates.get(from);
double toRate = exchangeRates.get(to);         return
amount * (toRate / fromRate);
    }
        private void showErrorDialog(String message) {
        Dialog errorDialog = new Dialog(this, "Error", true);
errorDialog.setSize(300, 150);         errorDialog.setLayout(new
FlowLayout());
```

```
        Label errorMessage = new Label(message);
        Button okButton = new Button("OK");

        okButton.addActionListener(e -> errorDialog.dispose());

errorDialog.add(errorMessage);
errorDialog.add(okButton);
errorDialog.setVisible(true);
    }

    public void updateExchangeRates(String currency, double rate) {
exchangeRates.put(currency, rate);
    }

    public static void main(String[] args) {
        CurrencyConverter converter = new CurrencyConverter();
converter.setVisible(true);
    } }
```

APPENDIX B - SCREENSHOTS




Currency Converter
Amount:
100
From Currency:
INR
To Currency:
EUR
Convert
Converted Amount: 1.11


Currency Converter
500000
EUR
INR
Convert
Converted Amount: 44929347.83