

(CASHFLIP)

A PROJECT REPORT

Submitted by

GAURAV KUMAR(22BCA10329)

in partial fulfillment for the award of the degree of

BACHELOR OF COMPUTER APPLICATIONS

IN

UNIVERSITY INSTITUTE OF COMPUTING



CHANDIGARH UNIVERSITY

April, 2025



BONAFIDE CERTIFICATE

Certified that this project report “**(CASH FLIP)**” is the bonafide work of
“**GAURAV KUMAR**” who carried out the project work under my/our supervision.

SIGNATURE

Dr. Kavita Gupta
Head of Department
UIC - BCA

SIGNATURE

Miss.Isha Dhingra
Project Supervisor
UIC - BCA

Submitted for the project viva-voce examination held on

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success. On the submission of our thesis report on “**(CASHFLIP)**”, we would like to extend our gratitude and sincere thanks to our supervisor **Miss.Isha Dhingra**, Department of Computer Applications for his constant motivation and support. We truly appreciate and value her esteemed guidance and encouragement from the beginning to the end of this thesis. I am indebted to him for having helped us shape the problem and providing insights towards the solution. I want to thank my teacher **Miss.Isha Dhingra** for providing a solid background for our studies and research thereafter. I am grateful to my supervisor for the guidance, inspiration and constructive suggestions that were helpful in the preparation of this project and its successful completion.

He has been a great source of inspiration to me and I thank him from the bottom of my heart. Above all, I would like to thank all my friends whose direct and indirect support helped me in the last month to go ahead with my Thesis. The thesis would have been impossible without their perpetual moral support.

(GAURAV KUMAR – 22BCA10329)

TABLE OF CONTENT

Abstract	05
CHAPTER 1. INTRODUCTION & DEMO AND CODE	06
CHAPTER 2. PROJECT OBJECTIVES	07
CHAPTER 3. TECHNOLOGIES IMPLEMENTED	08
CHAPTER 4. PROJECT FEATURES	10
CHAPTER 5. DEVELOPMENT PROCESS	12
CHAPTER 6. CONCLUSION	14

ABSTRACT

This project, titled "Currency Converter GUI", is a simple yet effective desktop application developed using the Java Swing framework. The primary objective of this project is to create an intuitive and user-friendly interface that allows users to convert amounts between various currencies such as USD, EUR, INR, and JPY. The conversion is done using predefined exchange rates that are hardcoded into the application.

The application is designed with a graphical user interface (GUI), which provides an easy way for users to interact with the program. The interface consists of dropdown menus that allow users to select the source and target currencies, input fields for entering the amount they wish to convert, and a convert button to initiate the conversion. Once the user enters the data and clicks the button, the program uses the corresponding exchange rate from the predefined data to calculate and display the converted amount in real-time.

The backend of the application utilizes several core Java concepts such as arrays and 2D arrays to store the conversion rates. The event handling mechanism is employed to detect and respond to user actions, such as clicking the Convert button. Additionally, exception handling is used to manage input errors, ensuring the program remains stable even when users input invalid data, such as non-numeric values for the amount.

This project also uses the Swing framework, a powerful set of GUI components, for designing and structuring the user interface. The use of Swing allows for the creation of windows, buttons, text fields, and other elements necessary for an interactive program. The choice of Swing was particularly beneficial because it provides a straightforward way to create cross-platform applications that can run on any operating system with a Java Virtual Machine (JVM).

Beyond basic functionality, this project serves as a foundation for more advanced applications. For instance, in the future, I plan to integrate live exchange rates using an API to retrieve real-time data from the web. This will ensure that users always have up-to-date exchange rates, making the converter even more practical. Additionally, the system could be enhanced by adding a feature to save a history of conversions in a database, allowing users to keep track of their past transactions. Such improvements would turn this tool from a basic converter into a more comprehensive financial application.

Overall, the Currency Converter GUI project is a beginner-friendly Java project that demonstrates my practical knowledge of GUI programming, event handling, and real-world utility in financial conversions. It has allowed me to apply theoretical programming concepts to build a useful and functional tool. Through this project, I have gained hands-on experience in working with Java's GUI library and backend logic, and it has paved the way for me to develop more complex applications in the future.

INTRODUCTION & DEMO AND CODE

Project Overview

The Currency Converter GUI is a user-friendly application that provides an intuitive graphical interface to perform currency conversions. Users can select source and target currencies from dropdown menus, enter the amount to be converted, and press a button to calculate and display the converted amount. The application is designed to handle multiple currencies, including USD, EUR, INR, and JPY, with exchange rates predefined within the system.

Objectives

- The primary goals of the Currency Converter GUI project are:
- Develop a simple currency conversion tool that allows users to easily convert between multiple currencies.
- Integrate a user-friendly graphical interface that simplifies the conversion process for non-technical users.
- Apply core Java concepts such as arrays, event handling, and exception handling in the context of a practical application.
- Ensure real-time conversion calculations based on predefined exchange rates.
- Provide a foundation for future enhancements, including the integration of live exchange rates via APIs and the ability to save transaction history.
- Enhance programming skills in Java, specifically in GUI development and event-driven programming.

Project Requirements

To successfully implement the Currency Converter GUI, the following requirements need to be met:

Software Requirements:

- Java Development Kit (JDK) version 8 or higher: Required for compiling and running Java code.
- Integrated Development Environment (IDE): Eclipse, IntelliJ IDEA, or any IDE that supports Java development and Swing-based applications.
- Java Swing Library: A part of the JDK that is used to create the graphical user interface.

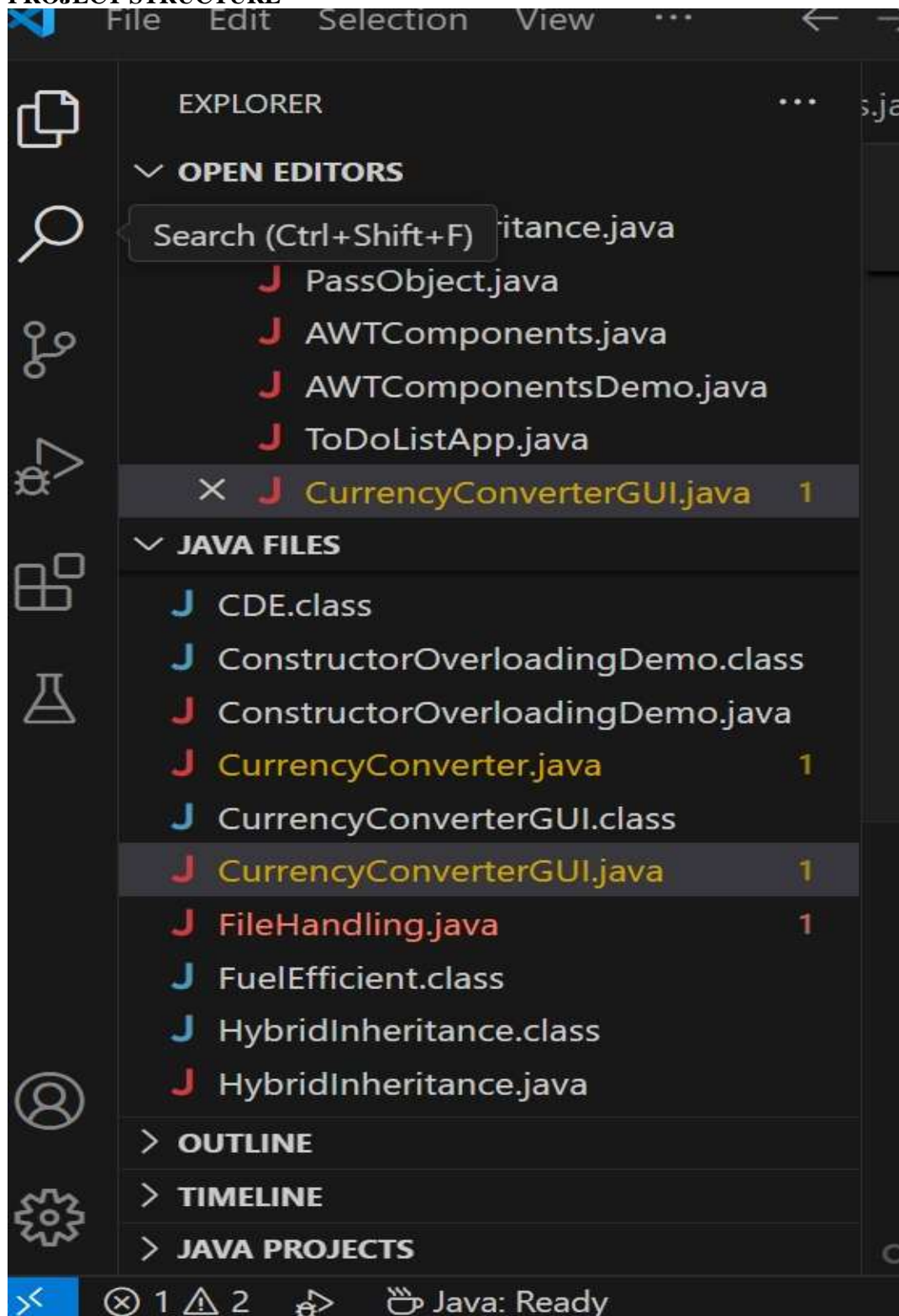
Hardware Requirements:

- A system with at least 4 GB of RAM to ensure smooth execution of the application.
- A keyboard and mouse for user interaction with the GUI.
- A display screen for viewing the application interface.

Other Requirements:

Basic understanding of Java programming: Knowledge of Java programming concepts such as arrays, loops, and event handling.

PROJECT STRUCTURE



CurrencyConverterGUI.java

```
J CurrencyConverterGUI.java > ...
1  import javax.swing.*;
2  import java.awt.*;
3  import java.awt.event.*;
4
5  public class CurrencyConverterGUI extends JFrame {
6      private JComboBox<String> fromCurrency, toCurrency;
7      private JTextField amountField, resultField;
8      private JButton convertButton;
9
10     private final String[] currencies = {"USD", "EUR", "INR", "JPY"};
11     private final double[][] rates = {
12         // USD    EUR    INR    JPY
13         {1.0,    0.85,   83.12,  151.23}, // USD
14         {1.18,   1.0,    97.56,  177.84}, // EUR
15         {0.012,  0.010,  1.0,    1.82},  // INR
16         {0.0066, 0.0056, 0.55,   1.0}    // JPY
17     };
18
19     public CurrencyConverterGUI() {
```

```
20         setTitle(title:"Currency Converter");
21         setSize(width:400, height:200);
22         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
23         setLayout(new GridLayout(rows:5, cols:2, hgap:10, vgap:10));
24
25         // Components
26         add(new JLabel(text:"From Currency:"));
27         fromCurrency = new JComboBox<>(currencies);
28         add(fromCurrency);
29
30         add(new JLabel(text:"To Currency:"));
31         toCurrency = new JComboBox<>(currencies);
32         add(toCurrency);
33
34         add(new JLabel(text:"Amount:"));
35         amountField = new JTextField();
36         add(amountField);
```



```

37
38     add(new JLabel(text:"Converted Amount:"));
39     resultField = new JTextField();
40     resultField.setEditable(b:false);
41     add(resultField);
42
43     convertButton = new JButton(text:"Convert");
44     add(convertButton);
45
46     // Empty cell
47     add(new JLabel());
48
49     // Convert button action
50     convertButton.addActionListener(e -> convertCurrency());
51
52     setVisible(b:true);
53 }

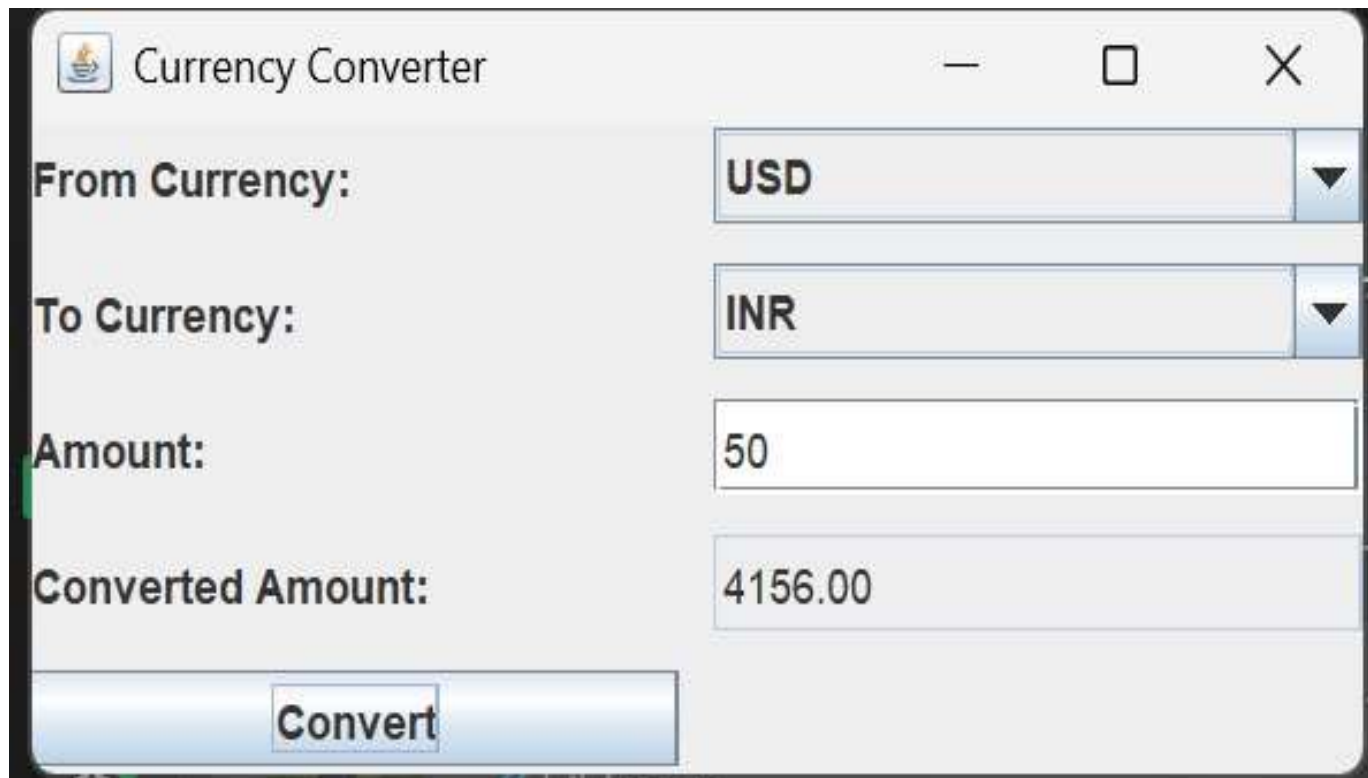
```

```

55 private void convertCurrency() {
56     try {
57         int from = fromCurrency.getSelectedIndex();
58         int to = toCurrency.getSelectedIndex();
59         double amount = Double.parseDouble(amountField.getText());
60
61         double result = amount * rates[from][to];
62         resultField.setText(String.format(format:"%.2f", result));
63     } catch (NumberFormatException ex) {
64         JOptionPane.showMessageDialog(this, message:"Please enter a valid numl
65     }
66 }
67
68 Run | Debug
69 public static void main(String[] args) {
70     SwingUtilities.invokeLater(CurrencyConverterGUI::new);
71 }

```

RESULT



The image shows a screenshot of a 'Currency Converter' application window. The window has a title bar with the text 'Currency Converter' and standard minimize, maximize, and close buttons. The main interface consists of four labeled input fields on the left and their corresponding values on the right. The 'From Currency' field is set to 'USD', the 'To Currency' field is set to 'INR', the 'Amount' field contains the number '50', and the 'Converted Amount' field displays '4156.00'. At the bottom left, there is a prominent blue 'Convert' button.

Label	Value
From Currency:	USD
To Currency:	INR
Amount:	50
Converted Amount:	4156.00

Convert

PROJECT OBJECTIVES

- The main objectives of the Currency Converter GUI project are as follows:
- To develop a desktop application that converts currency values from one currency to another with ease.
- To design a clean, interactive, and user-friendly GUI using Java Swing.
- To implement a working model that uses fixed exchange rates with accurate mathematical conversions.
- To practice and demonstrate the use of core Java programming concepts such as:
- Arrays and 2D arrays
- Event handling and GUI interaction
- Exception handling (e.g., handling wrong inputs)
- To provide a tool that can be further extended with features like:
- Real-time exchange rates using APIs
- Conversion history or logs
- More currency options and settings
- To enhance understanding of how backend logic interacts with frontend user actions in a GUI environment.

TECHNOLOGIES IMPLEMENTED

- The following technologies and tools were used in the development of this project:
- Programming Language: Java
- Used to write the core logic and backend calculations.
- Java Swing:
- A part of Java's standard library used for GUI development.
- Components used: JFrame, JLabel, JTextField, JComboBox, JButton, etc.
- Integrated Development Environment (IDE):
- IntelliJ IDEA / Eclipse / NetBeans (any one based on what you used)
- Java Development Kit (JDK):
- Minimum version 8 recommended.
- Operating System:
- Windows / Linux / macOS (whichever OS you used during development)
- Optional (if included in your future version):
- JSON or API handling (for live exchange rates)

PROJECT FEATURES

- The Currency Converter GUI includes the following features:
- Graphical User Interface:
- Built using Java Swing with a clean and structured layout.
- Currency Selection:
- Dropdowns (ComboBoxes) for selecting "From" and "To" currencies.
- Real-Time Conversion:
- Upon entering the amount and clicking "Convert", the result is calculated instantly.
- Input Validation:
- Prevents invalid inputs (e.g., letters or empty fields).

DEVELOPMENT PROCESS

The development process was carried out in the following steps:

- *Requirement Gathering:*
Identified the need for a simple currency converter with a GUI.
- *Design Phase:*
Planned the layout of the GUI using paper sketches and toolkits like Swing.
- *Coding Phase:*
Wrote the Java code to implement logic and connect it to GUI components.
Created currency rate matrix (2D array) for conversion logic.
- *Testing Phase:*
Manually tested different scenarios including valid/invalid inputs.
Checked GUI alignment, calculations, and error handling.
- *Debugging:*
Resolved issues like incorrect rates, input parsing errors, and GUI glitches.
- *Final Review:*
Cleaned up code, added comments, and finalized the layout for submission.

CONCLUSION

The Currency Converter GUI project helped me understand how to integrate backend logic with frontend design using Java. I was able to build a fully working desktop application that converts currencies with a user-friendly interface. The project also allowed me to apply key programming concepts such as array manipulation, event handling, and exception management.

While the current version uses fixed exchange rates, this project has great potential for future development. With API integration, live currency rates can be fetched, and with database support, user history can be saved and managed. Overall, this mini project gave me valuable practical experience and boosted my confidence in Java GUI programming.