# Batua

## Personal Income and Expense Tracker


**SRM University AP**

**Department of Computer Science**


**Course:** Mobile Application Development using Java

**Course Code:** CSE - 307


**Submitted By:**

| | |
|---|---|
| A R Gopikrishna | **AP22110010118** |
| Aaditya Singh | **AP22110010210** |
| Simant Mishra | **AP22110010211** |
| Parth Kiran Hanchate | **AP22110010671** |
| Shreya Bipin Roy | **AP22110010690** |

**Under the Supervision of**

**Dr. Aurobindo Behera**


**April 19, 2025**

# Contents

**1 Introduction** **2**

**2 Project Overview** **2**

**3 Code Structure** **3**

**4 Technologies Utilized** **4**

**5 Challenges Encountered** **4**

**6 Application Screenshots** **4**

**7 Conclusion** **6**

**8 Future Enhancements** **6**

# 1. Introduction

Financial management plays a crucial role in ensuring personal financial stability. In today's fast-paced society, manually tracking income and expenses can be difficult and error-prone.

To simplify this process, we have designed a desktop-based solution named **Personal Income and Expense Tracker** using Java Swing and MySQL. This tool enables users to effectively record, manage, and analyze their financial activities. The development of this project is a part of our coursework in Java Programming, aimed at enhancing our hands-on knowledge of GUI-based applications integrated with a relational database system.

## Acknowledgements

We wish to extend our heartfelt thanks to **Dr. Aurobindo Behera**, our esteemed professor, for his unwavering support and expert guidance throughout the project. His valuable feedback and constant encouragement played a significant role in the successful completion of this project.

We also express our gratitude to our classmates and faculty members for their continuous support and for contributing to the effective execution of this project.

# 2. Project Overview

Below is a brief summary of the tasks completed during the course of this project:

- **GUI Development:** We created a user-friendly graphical interface using `JFrame`, `JPanel`, `JLabel`, `JButton`, and `JTable`. A custom title bar was incorporated along with specialized buttons for minimizing and closing the window, giving the application a modern aesthetic.
- **Database Connectivity:** A MySQL database was integrated to store and manage user transaction data. We developed a `DatabaseConnection` class to handle database connections and a `TransactionDAO` class to manage data operations, such as retrieving and updating records.
- **Transaction Management:** We designed a `Transaction` model class to encapsulate transaction details such as ID, type, description, and amount. We also created methods to add and delete transactions from the system.
- **Displaying Transactions:** We utilized `JTable` to display transaction data. A custom cell renderer was used to apply different color schemes to income and expense transactions, enhancing the visual distinction between them.
- **Total Calculation Logic:** We developed the `TransactionValuesCalculation` class to compute dynamic totals for income, expenses, and balance, based on the user's transaction history.
- **User Interface Customization:** The user interface was enhanced with gradient effects, rounded corners, and custom scrollbars. Additionally, the title bar was made draggable to improve the user's interaction with the window.

## 3. Code Structure

This section provides an overview of the main classes in the project:

### 1. DatabaseConnection.java

This class manages the connection to the MySQL database through JDBC. It defines the connection parameters such as DB URL, username, and password, and returns a connection object upon successful connection.

### 2. Transaction.java

A simple POJO (Plain Old Java Object) that holds the details of each transaction. The fields include ID, type (Income/Expense), description, and amount. Getters and setters are provided for each field.

### 3. TransactionDAO.java

This class implements the Data Access Object (DAO) pattern to manage database operations. It defines a method `getAllTransaction()` that retrieves all records from the database and converts them into `Transaction` objects for display in the GUI.

### 4. TransactionValuesCalculation.java

This utility class provides methods to calculate:

- `getTotalIncomes()` – Sums all income transactions
- `getTotalExpenses()` – Sums all expense transactions
- `getTotalBalance()` – Computes the net balance by subtracting total expenses from total income

These calculations are based on the transaction type and are updated dynamically.

### 5. GUI Implementation

The GUI features:

- A draggable title bar
- Buttons to add and remove transactions
- A table that lists all transactions
- Dialog boxes for adding new transactions

The table utilizes a custom cell renderer to differentiate income and expenses visually by applying distinct colors.

## 4. Technologies Utilized

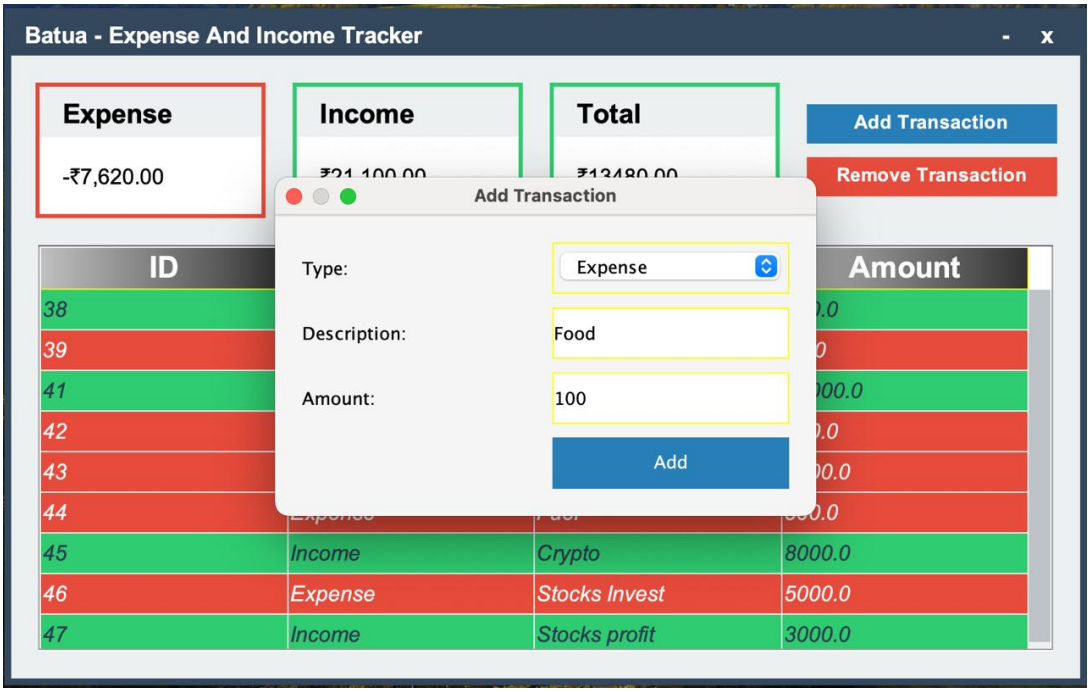
- **Java Swing** – for developing the graphical user interface
- **MySQL** – for data storage and retrieval
- **JDBC** – for establishing the connection between Java and MySQL
- **NetBeans IDE** – for application development and testing

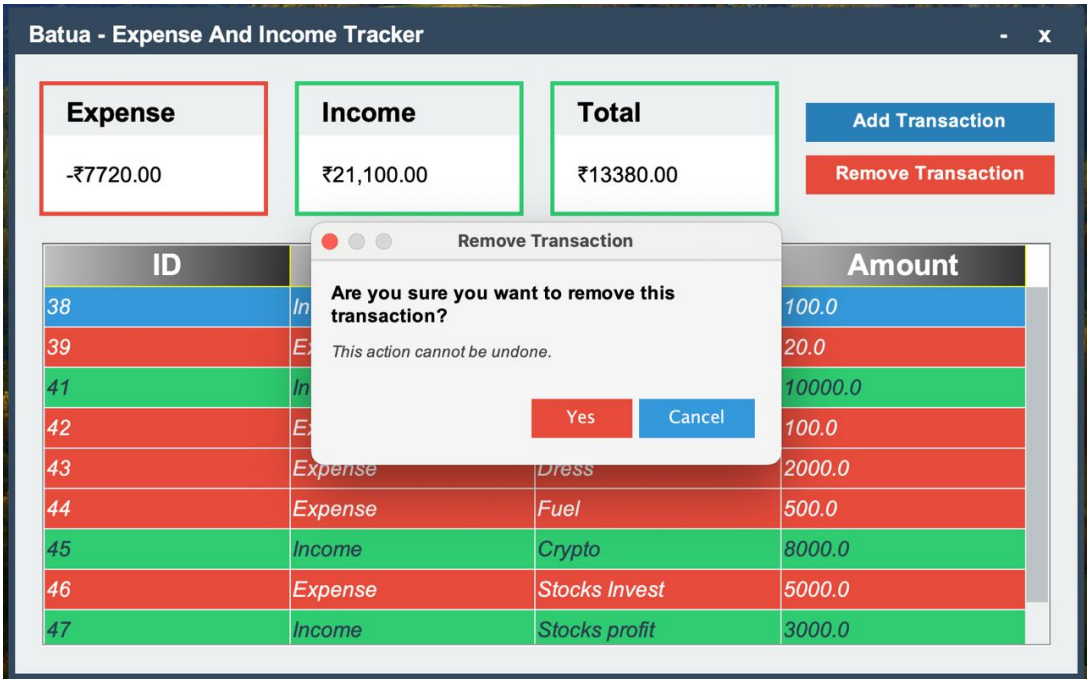
## 5. Challenges Encountered

- Synchronizing real-time updates between the GUI and database
- Designing an intuitive and responsive user interface using Swing
- Developing a scrollable and color-coded table display
- Managing SQL exceptions and ensuring data integrity

## 6. Application Screenshots

The following are some screenshots from the application interface:


Batua - Expense And Income Tracker
Expense
-₹7,620.00
Income
Total
Add Transaction
Remove Transaction
Add Transaction
Type:
Expense
Description:
Food
Amount:
100
Add
ID
Amount
38
39
41
42
43
44
45 Income Crypto 8000.0
46 Expense Stocks Invest 5000.0
47 Income Stocks profit 3000.0


Batua - Expense And Income Tracker
Expense
-₹7720.00
Income
₹21,100.00
Total
₹13380.00
Add Transaction
Remove Transaction
Remove Transaction
Are you sure you want to remove this transaction?
This action cannot be undone.
Yes
Cancel
ID
Amount
38 100.0
39 20.0
41 10000.0
42 100.0
43 2000.0
44 Expense Fuel 500.0
45 Income Crypto 8000.0
46 Expense Stocks Invest 5000.0
47 Income Stocks profit 3000.0

## 7. Conclusion

This project has provided valuable hands-on experience in creating a desktop application with Java Swing and integrating it with MySQL for data storage. Through this project, we explored important design patterns such as MVC and DAO, as well as event-driven programming in Java.

The application successfully allows users to track and manage their income and expenses, visualize their financial data, and calculate net balances efficiently.

## 8. Future Enhancements

The following features can be added in future updates:

- Graphical representations of data, such as pie and bar charts, using JFreeChart
- Ability to export reports in PDF or CSV formats
- Integration of monthly budgeting and automatic notifications
- Migration of the application to a web-based platform using JavaFX or Spring Boot

## References


- Oracle Java Swing Tutorials
- MySQL Official Documentation
- GeeksforGeeks Java Swing Resources