VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“Jnana Sangama”, Belagavi – 590 018

Object Oriented Programming with Java (21CSE44)

Assignment

“Expense Tracker System”

Submitted By

Name: Shree Ram A N **USN: 1GA21CS001**

Name: Chandrashekar K R **USN: 1GA21CS042**

Under the Guidance of

Mr. Shyam Sundar Bhushan
Assistant Professor
Dept. of CSE

Department of Artificial Intelligence & Data Science
GLOBAL ACADEMY OF TECHNOLOGY
Rajarajeshwari Nagar, Bengaluru – 560 098
2023 - 2024

CHAPTER 1

PROBLEM DEFINITION

In today's fast-paced world, individuals and businesses need an efficient way to track their expenses. Keeping a record of expenditures helps manage finances, plan budgets, and make informed financial decisions. To address this need, we will develop an Expense Tracker System - a Java-based application that allows users to input, track, view, update, and delete expenses. This system will help users maintain a detailed record of their expenses, categorize them, and analyze their spending habits.

The project aims to create an application with the following key features:

Expense Recording: Users can input expenses, including amount, date, category, and location, with data validation for accuracy.

Expense Viewing: Users can access expenses in various ways, such as by date range, category, location, or view all expenses.

Expense Updating: Users have the option to update existing expense records while maintaining data consistency.

Expense Deletion: The system allows users to delete expenses, with confirmation prompts to prevent accidental deletions.

Data Persistence: Expense data is saved in a structured text file for continuity between program executions.

Total Expense Calculation: The application dynamically calculates and displays the total balance based on recorded expenses.

User-Friendly Interface: A Command-Line Interface (CLI) guides users with clear menu options and prompts.

Error Handling: Robust error handling provides meaningful messages and manages exceptions during file operations.

Code Modularity: The Java code is organized for maintainability and potential future enhancements.

This Expense Tracker project empowers users to efficiently manage their finances, making it suitable for personal use and small businesses.

CHAPTER 2

IMPLEMENTATION

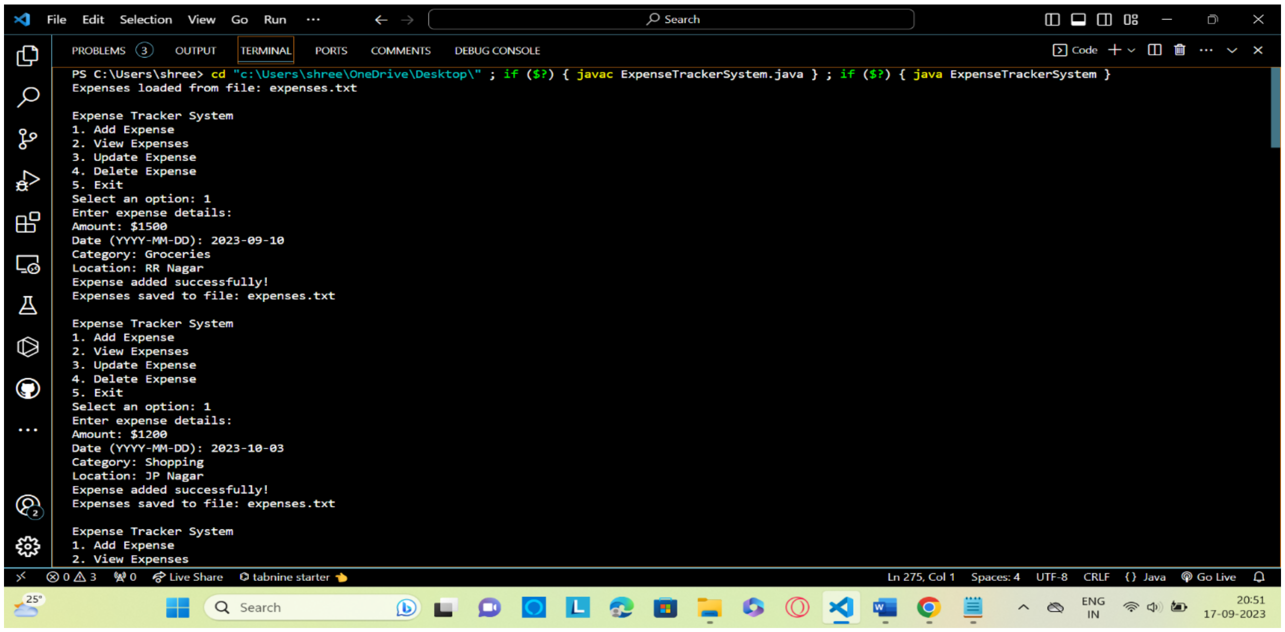
2.1 PROGRAM CODE

```
import java.io.*;
import java.util.ArrayList;
import java.util.Scanner;

class Expense {
    private double amount;
    private String date;
    private String category;
    private String location;

    public Expense(double amount, String date, String category, String location) {
        this.amount = amount;
        this.date = date;
        this.category = category;
        this.location = location;
    }

    public double getAmount() {
        return amount;
    }

    public String getDate() {
        return date;
    }

    public String getCategory() {
        return category;
    }

    public String getLocation() {
        return location;
    }
}

class ExpenseTracker {
    private ArrayList<Expense> expenses = new ArrayList<>();
    private Scanner scanner = new Scanner(System.in);
    private static final String DATA_FILE = "expenses.txt";

    public void addExpense() {
        System.out.println("Enter expense details:");
        System.out.print("Amount: $");
        double amount = scanner.nextDouble();
        scanner.nextLine(); // Consume newline
        System.out.print("Date (YYYY-MM-DD): ");
        String date = scanner.nextLine();
        System.out.print("Category: ");
        String category = scanner.nextLine();
        System.out.print("Location: ");
        String location = scanner.nextLine();

        Expense expense = new Expense(amount, date, category, location);
        expenses.add(expense);
        System.out.println("Expense added successfully!");
        saveExpensesToFile();
    }

    public void viewExpenses() {
        System.out.println("\nView Expenses");
        System.out.println("1. All Expenses");
        System.out.println("2. Expenses by Date");
        System.out.println("3. Expenses by Category");
        System.out.println("4. Expenses by Location");
        System.out.println("5. Back to Main Menu");
        System.out.print("Select an option: ");
        int viewChoice = scanner.nextInt();
        scanner.nextLine(); // Consume newline
        switch (viewChoice) {
            case 1:
                loadExpensesFromFile();
                viewAllExpenses();
                break;
            case 2:
                viewExpensesByDate();
                break;
            case 3:
                viewExpensesByCategory();
                break;
            case 4:
                viewExpensesByLocation();
                break;
            case 5:
                return;
            default:
                System.out.println("Invalid choice. Please select a valid option.");
        }
    }

    private void viewAllExpenses() {
        System.out.println("\nAll Expenses:");
        displayExpenses(expenses);
        displayTotalBalance();
    }

    private void viewExpensesByDate() {
        System.out.print("Enter the date (YYYY-MM-DD): ");
        String targetDate = scanner.nextLine();
        ArrayList<Expense> filteredExpenses = new ArrayList<>();

        for (Expense expense : expenses) {
            if (expense.getDate().equals(targetDate)) {
                filteredExpenses.add(expense);
            }
        }

        System.out.println("\nExpenses on " + targetDate + ":");
        displayExpenses(filteredExpenses);
        displayTotalBalance();
    }

    private void viewExpensesByCategory() {
        System.out.print("Enter the category: ");
        String targetCategory = scanner.nextLine();
        ArrayList<Expense> filteredExpenses = new ArrayList<>();

        for (Expense expense : expenses) {
            if (expense.getCategory().equalsIgnoreCase(targetCategory)) {
                filteredExpenses.add(expense);
            }
        }

        System.out.println("\nExpenses in the category '" + targetCategory + "':");
        displayExpenses(filteredExpenses);
        displayTotalBalance();
    }

    private void viewExpensesByLocation() {
        System.out.print("Enter the location: ");
        String targetLocation = scanner.nextLine();
        ArrayList<Expense> filteredExpenses = new ArrayList<>();

        for (Expense expense : expenses) {
            if (expense.getLocation().equalsIgnoreCase(targetLocation)) {
                filteredExpenses.add(expense);
            }
        }

        System.out.println("\nExpenses in the location '" + targetLocation + "':");
        displayExpenses(filteredExpenses);
        displayTotalBalance();
    }

    private void displayExpenses(ArrayList<Expense> expensesToDisplay) {
        if (expensesToDisplay.isEmpty()) {
            System.out.println("No expenses to display.");
        } else {
            System.out.println("--------------------------------------------------------------------------------
-------------------------------");
            System.out.printf("%-5s %-15s %-15s %-20s %-20s%n", "No.", "Amount ($)",
"Date", "Category", "Location");
            System.out.println("--------------------------------------------------------------------------------
-------------------------------");

            for (int i = 0; i < expensesToDisplay.size(); i++) {
                Expense expense = expensesToDisplay.get(i);
                System.out.printf("%-4d $%-14.2f%-14s%-20s%-20s%n",
                        i + 1, expense.getAmount(), expense.getDate(), expense.getCategory(),
expense.getLocation());
            }

            System.out.println("--------------------------------------------------------------------------------
-------------------------------");
        }
    }

    private void displayTotalBalance() {
        double totalBalance = 0.0;
        for (Expense expense : expenses) {
            totalBalance += expense.getAmount();
        }
        System.out.println("Total Expenses: $" + totalBalance);
    }

    public void updateExpense() {
        viewAllExpenses();
        System.out.print("Enter the number of the expense to update (or 0 to go back to the main
menu): ");
        int expenseNumber = scanner.nextInt();
        scanner.nextLine(); // Consume newline

        if (expenseNumber == 0) {
            return; // Go back to the main menu
        }

        if (expenseNumber > 0 && expenseNumber <= expenses.size()) {
            System.out.println("Enter new expense details:");
            System.out.print("Amount: $");
            double amount = scanner.nextDouble();
            scanner.nextLine(); // Consume newline
            System.out.print("Date (YYYY-MM-DD): ");
            String date = scanner.nextLine();
            System.out.print("Category: ");
            String category = scanner.nextLine();
            System.out.print("Location: ");
            String location = scanner.nextLine();

            Expense updatedExpense = new Expense(amount, date, category, location);
            expenses.set(expenseNumber - 1, updatedExpense);
            System.out.println("Expense updated successfully!");
            saveExpensesToFile();
        } else {
            System.out.println("Invalid expense number.");
        }
    }

    public void deleteExpense() {
        viewAllExpenses();
        System.out.print("Enter the number of the expense to delete (or 0 to go back to the main
menu): ");
        int expenseNumber = scanner.nextInt();
        scanner.nextLine(); // Consume newline
        if (expenseNumber == 0) {
            return; // Go back to the main menu
        }

        if (expenseNumber > 0 && expenseNumber <= expenses.size()) {
            expenses.remove(expenseNumber - 1);
            System.out.println("Expense deleted successfully!");
            saveExpensesToFile();
        } else {
            System.out.println("Invalid expense number.");
        }
    }

   public void saveExpensesToFile() {
    try (FileWriter fileWriter = new FileWriter(DATA_FILE)) {
        fileWriter.write(String.format("%-5s %-15s %-15s %-20s %-20s%n", "No.", "Amount
($)", "Date", "Category", "Location"));

        for (int i = 0; i < expenses.size(); i++) {
            Expense expense = expenses.get(i);
            fileWriter.write(String.format("%-4d $%-14.2f%-14s%-20s%-20s%n",
                    i + 1, expense.getAmount(), expense.getDate(), expense.getCategory(),
expense.getLocation()));
        }
        displayTotalBalance(fileWriter);
        System.out.println("Expenses saved to file: " + DATA_FILE);
    } catch (IOException e) {
        System.err.println("Error saving expenses to file: " + e.getMessage());
    }
}



    private void displayTotalBalance(FileWriter fileWriter) throws IOException {
        double totalBalance = 0.0;
        for (Expense expense : expenses) {
            totalBalance += expense.getAmount();
        }
        fileWriter.write("\nTotal Expenses: $" + totalBalance);
    }

    public void loadExpensesFromFile() {
        try (BufferedReader reader = new BufferedReader(new FileReader(DATA_FILE))) {
            String header = reader.readLine();
            String line;
            while ((line = reader.readLine()) != null) {
                String[] parts = line.split(",");
                if (parts.length == 4) {
                    double amount = Double.parseDouble(parts[0].substring(1)); // Remove the
leading '$' sign
                    String date = parts[1];
                    String category = parts[2];
                    String location = parts[3];
                    expenses.add(new Expense(amount, date, category, location));
                }
            }
            System.out.println("Expenses loaded from file: " + DATA_FILE);
        } catch (IOException e) {
            System.err.println("Error loading expenses from file: " + e.getMessage());
        }
    }

    public void initialize() {
        loadExpensesFromFile();
    }
}

public class ExpenseTrackerSystem {
    public static void main(String[] args) {
        ExpenseTracker expenseTracker = new ExpenseTracker();
        expenseTracker.initialize();
        Scanner scanner = new Scanner(System.in);
        while (true) {
            System.out.println("\nExpense Tracker System");
            System.out.println("1. Add Expense");
            System.out.println("2. View Expenses");
            System.out.println("3. Update Expense");
            System.out.println("4. Delete Expense");
            System.out.println("5. Exit");
            System.out.print("Select an option: ");

            int choice = scanner.nextInt();
            scanner.nextLine(); // Consume newline

            switch (choice) {
                case 1:
                    expenseTracker.addExpense();
                    break;
                case 2:
                    expenseTracker.viewExpenses();
                    break;
                case 3:
                    expenseTracker.updateExpense();
                    break;
                case 4:
                    expenseTracker.deleteExpense();
                    break;
                case 5:
                    System.out.println("Exiting Expense Tracker System. Goodbye!");
                    expenseTracker.saveExpensesToFile();
                    System.exit(0);
                default:
                    System.out.println("Invalid choice. Please select a valid option.");
            }
        }
    }
}
```

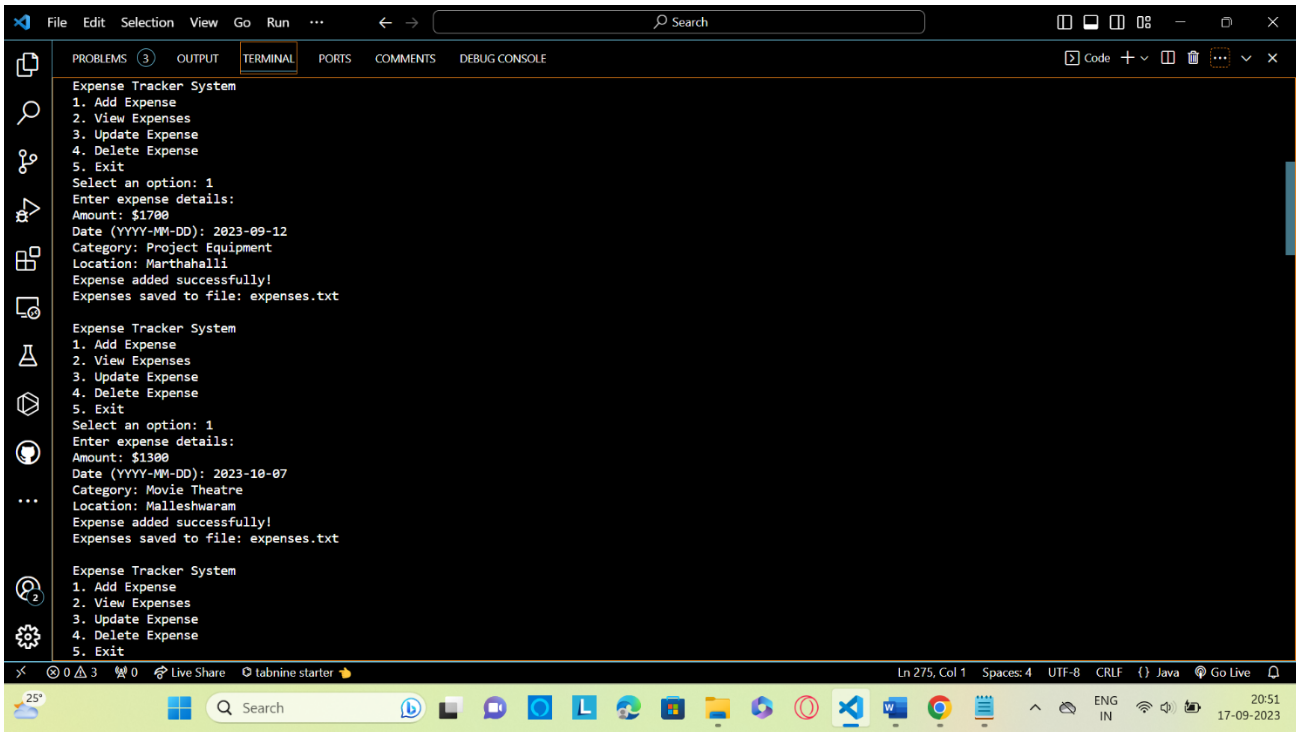
CHAPTER 3

RESULTS

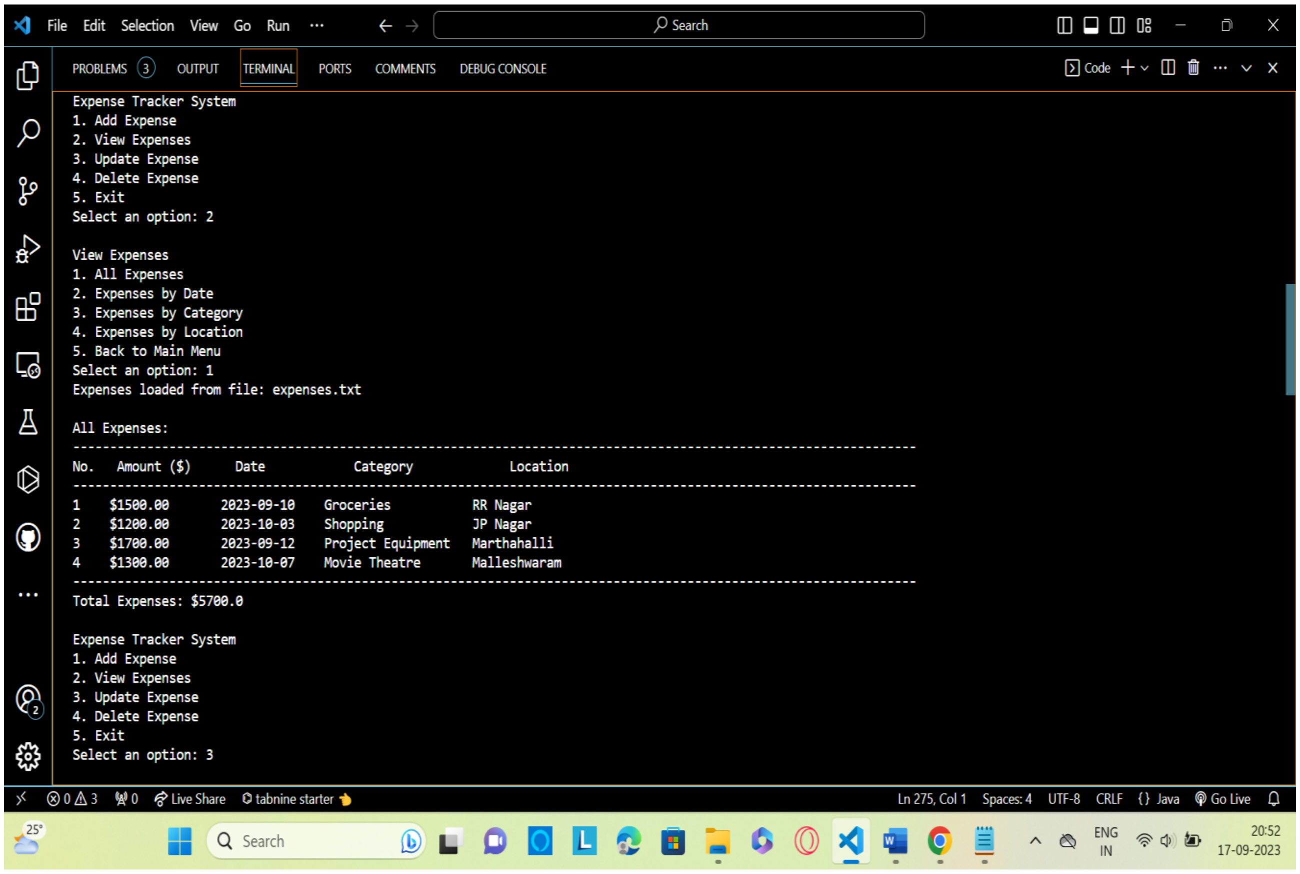
3.1 SNAPSHOTS



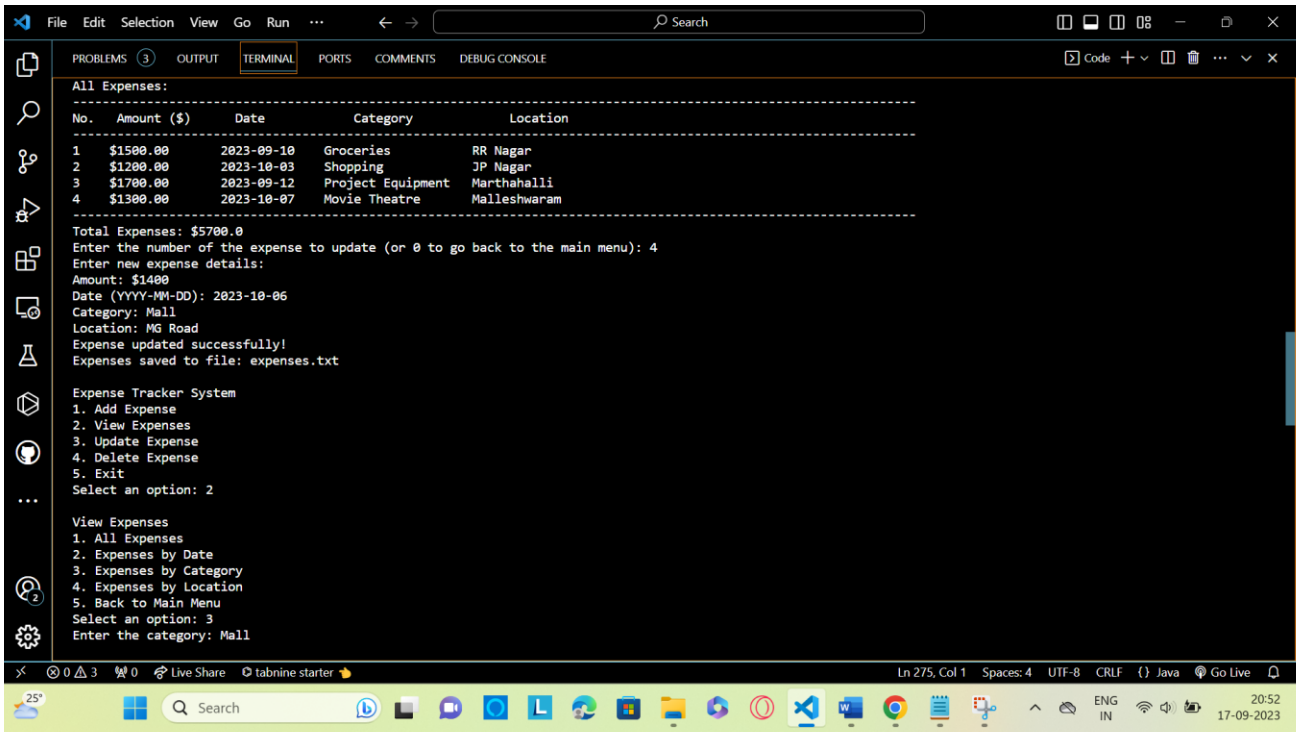
```
PS C:\Users\shree> cd "c:\Users\shree\OneDrive\Desktop\" ; if ($?) { javac ExpenseTrackerSystem.java } ; if ($?) { java ExpenseTrackerSystem }
Expenses loaded from file: expenses.txt

Expense Tracker System
1. Add Expense
2. View Expenses
3. Update Expense
4. Delete Expense
5. Exit
Select an option: 1
Enter expense details:
Amount: $1500
Date (YYYY-MM-DD): 2023-09-10
Category: Groceries
Location: RR Nagar
Expense added successfully!
Expenses saved to file: expenses.txt

Expense Tracker System
1. Add Expense
2. View Expenses
3. Update Expense
4. Delete Expense
5. Exit
Select an option: 1
Enter expense details:
Amount: $1200
Date (YYYY-MM-DD): 2023-10-03
Category: Shopping
Location: JP Nagar
Expense added successfully!
Expenses saved to file: expenses.txt

Expense Tracker System
1. Add Expense
2. View Expenses
```



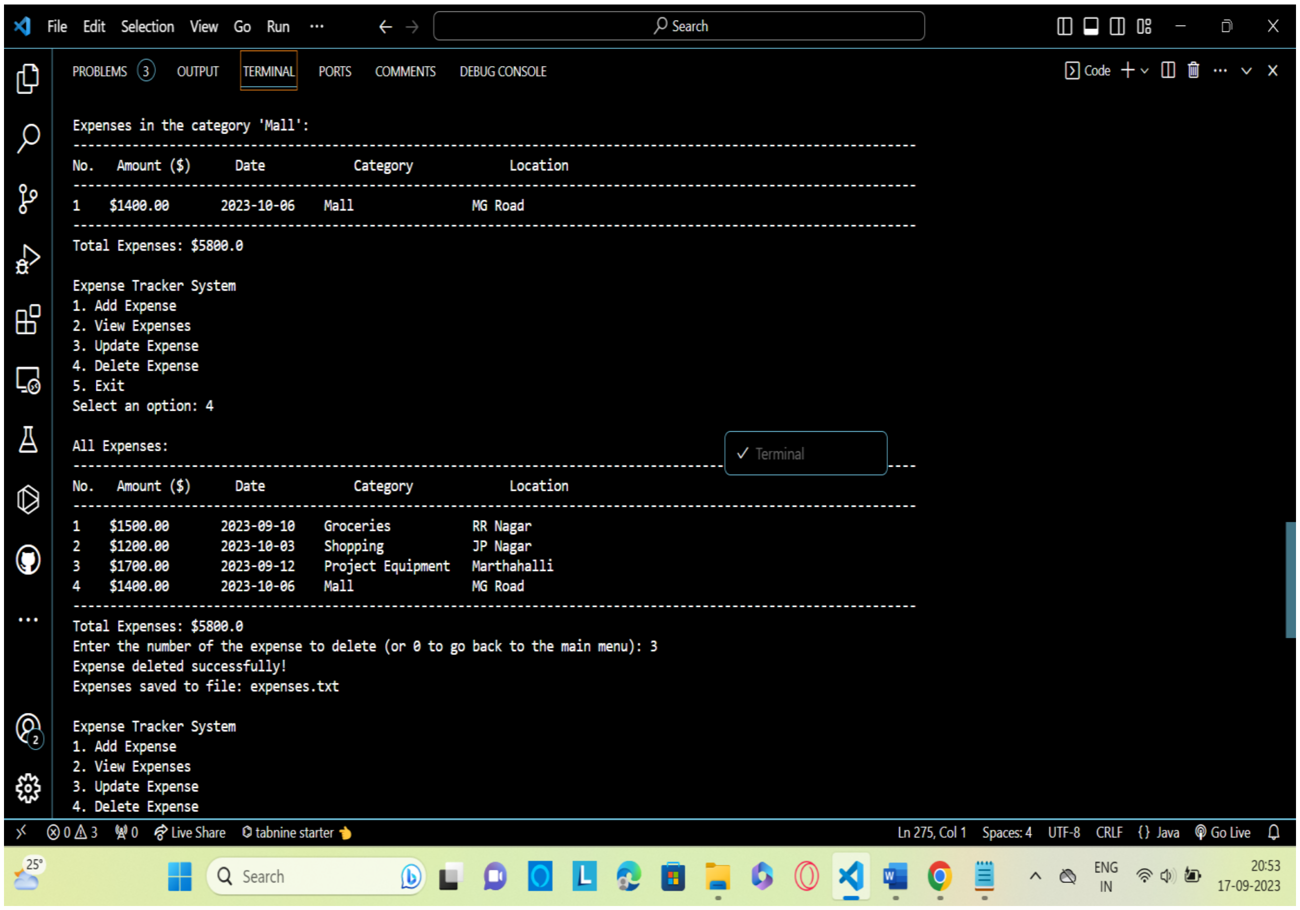
```
Expense Tracker System
1. Add Expense
2. View Expenses
3. Update Expense
4. Delete Expense
5. Exit
Select an option: 1
Enter expense details:
Amount: $1700
Date (YYYY-MM-DD): 2023-09-12
Category: Project Equipment
Location: Marthahalli
Expense added successfully!
Expenses saved to file: expenses.txt

Expense Tracker System
1. Add Expense
2. View Expenses
3. Update Expense
4. Delete Expense
5. Exit
Select an option: 1
Enter expense details:
Amount: $1300
Date (YYYY-MM-DD): 2023-10-07
Category: Movie Theatre
Location: Malleshwaram
Expense added successfully!
Expenses saved to file: expenses.txt

Expense Tracker System
1. Add Expense
2. View Expenses
3. Update Expense
4. Delete Expense
5. Exit
```



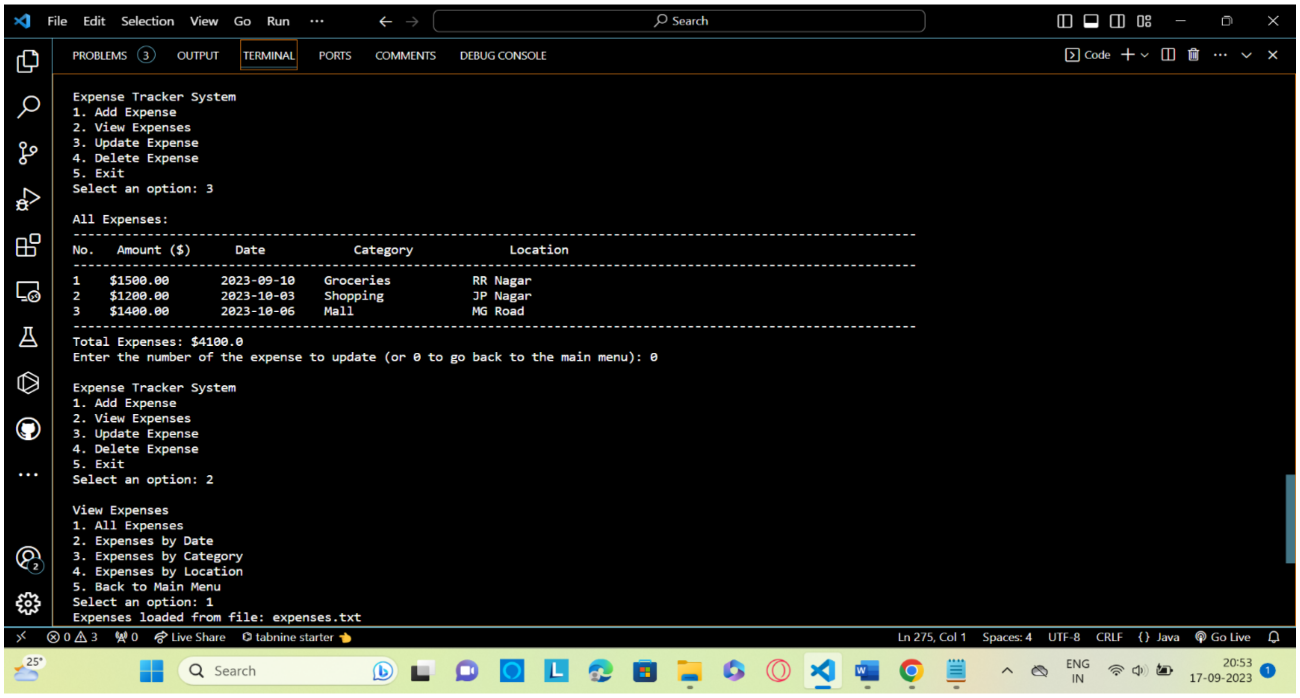
```
Expense Tracker System
1. Add Expense
2. View Expenses
3. Update Expense
4. Delete Expense
5. Exit
Select an option: 2

View Expenses
1. All Expenses
2. Expenses by Date
3. Expenses by Category
4. Expenses by Location
5. Back to Main Menu
Select an option: 1
Expenses loaded from file: expenses.txt

All Expenses:
------------------------------------------------------------------------------------------------------------------
No.   Amount ($)      Date            Category            Location
------------------------------------------------------------------------------------------------------------------
1     $1500.00       2023-09-10     Groceries           RR Nagar
2     $1200.00       2023-10-03     Shopping            JP Nagar
3     $1700.00       2023-09-12     Project Equipment   Marthahalli
4     $1300.00       2023-10-07     Movie Theatre       Malleshwaram
------------------------------------------------------------------------------------------------------------------
Total Expenses: $5700.0

Expense Tracker System
1. Add Expense
2. View Expenses
3. Update Expense
4. Delete Expense
5. Exit
Select an option: 3
```



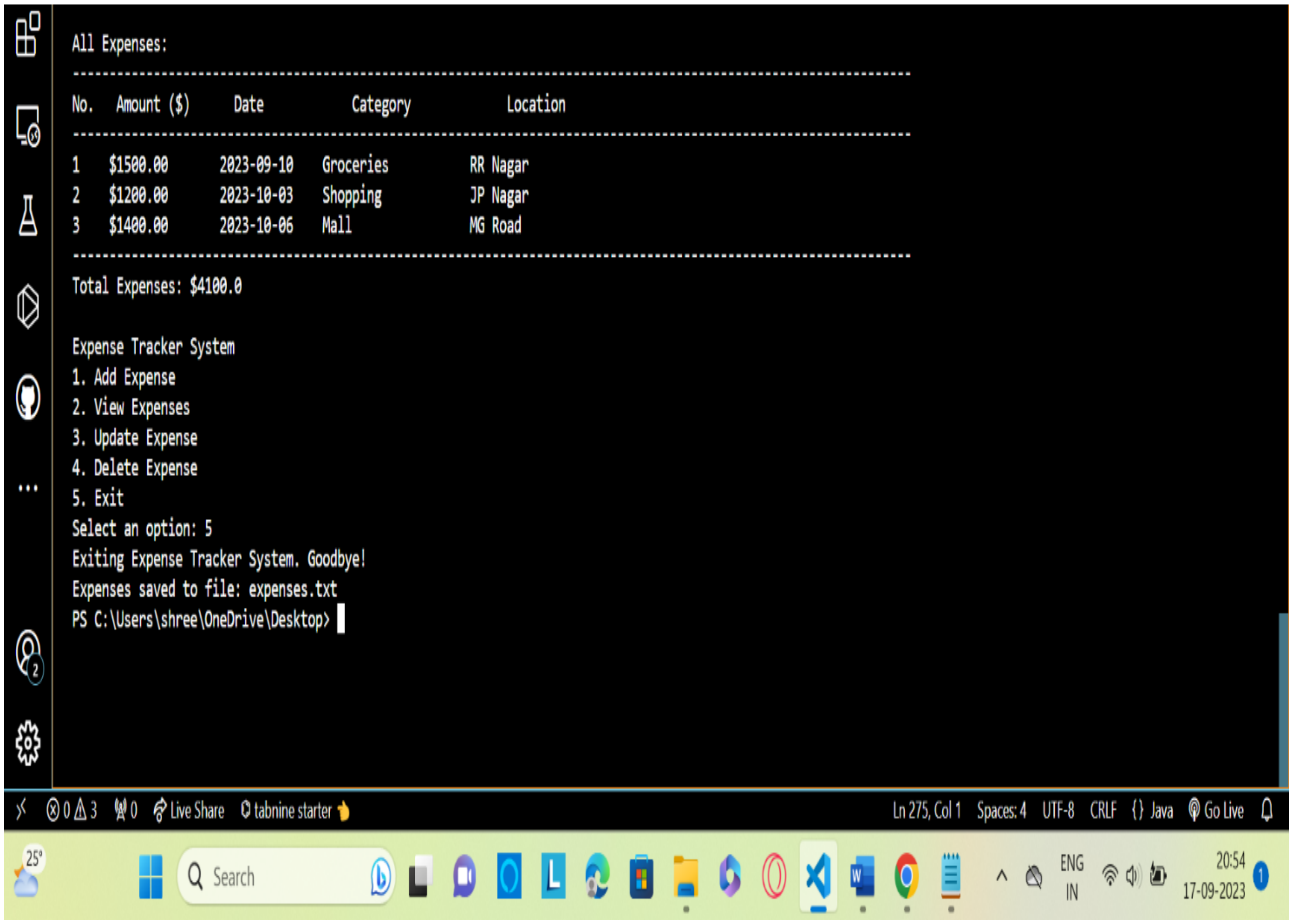
```
All Expenses:
------------------------------------------------------------------------------------------------------------------
No.   Amount ($)      Date            Category            Location
------------------------------------------------------------------------------------------------------------------
1     $1500.00       2023-09-10     Groceries           RR Nagar
2     $1200.00       2023-10-03     Shopping            JP Nagar
3     $1700.00       2023-09-12     Project Equipment   Marthahalli
4     $1300.00       2023-10-07     Movie Theatre       Malleshwaram
------------------------------------------------------------------------------------------------------------------
Total Expenses: $5700.0
Enter the number of the expense to update (or 0 to go back to the main menu): 4
Enter new expense details:
Amount: $1400
Date (YYYY-MM-DD): 2023-10-06
Category: Mall
Location: MG Road
Expense updated successfully!
Expenses saved to file: expenses.txt

Expense Tracker System
1. Add Expense
2. View Expenses
3. Update Expense
4. Delete Expense
5. Exit
Select an option: 2

View Expenses
1. All Expenses
2. Expenses by Date
3. Expenses by Category
4. Expenses by Location
5. Back to Main Menu
Select an option: 3
Enter the category: Mall
```



```
Expenses in the category 'Mall':
------------------------------------------------------------------------------------------------------------------
No.   Amount ($)      Date            Category            Location
------------------------------------------------------------------------------------------------------------------
1    $1400.00       2023-10-06    Mall                MG Road
------------------------------------------------------------------------------------------------------------------
Total Expenses: $5800.0

Expense Tracker System
1. Add Expense
2. View Expenses
3. Update Expense
4. Delete Expense
5. Exit
Select an option: 4

All Expenses:
------------------------------------------------------------------------------------------------------------------
No.   Amount ($)      Date            Category            Location
------------------------------------------------------------------------------------------------------------------
1    $1500.00       2023-09-10    Groceries           RR Nagar
2    $1200.00       2023-10-03    Shopping            JP Nagar
3    $1700.00       2023-09-12    Project Equipment   Marthahalli
4    $1400.00       2023-10-06    Mall                MG Road
------------------------------------------------------------------------------------------------------------------
Total Expenses: $5800.0
Enter the number of the expense to delete (or 0 to go back to the main menu): 3
Expense deleted successfully!
Expenses saved to file: expenses.txt

Expense Tracker System
1. Add Expense
2. View Expenses
3. Update Expense
4. Delete Expense
```

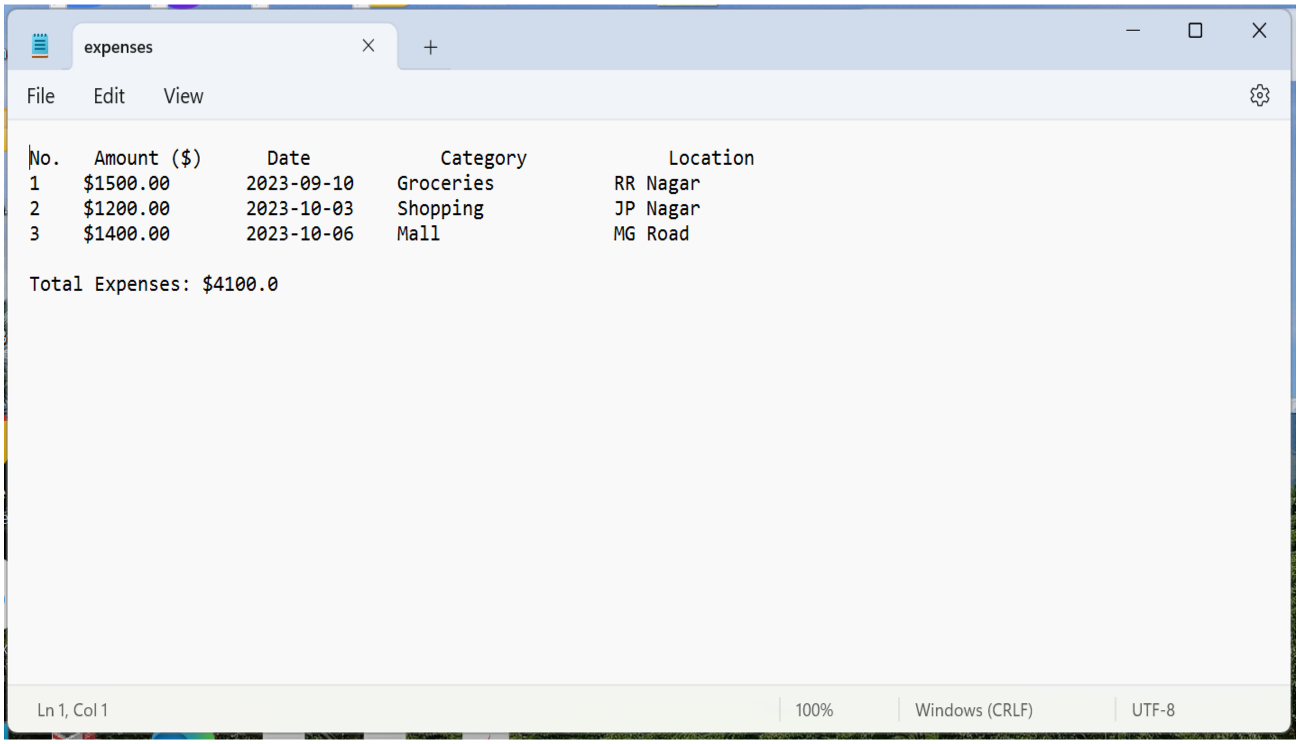


```
Expense Tracker System
1. Add Expense
2. View Expenses
3. Update Expense
4. Delete Expense
5. Exit
Select an option: 3

All Expenses:
------------------------------------------------------------------------------------------------------------------
No.   Amount ($)      Date            Category            Location
------------------------------------------------------------------------------------------------------------------
1    $1500.00       2023-09-10    Groceries           RR Nagar
2    $1200.00       2023-10-03    Shopping            JP Nagar
3    $1400.00       2023-10-06    Mall                MG Road
------------------------------------------------------------------------------------------------------------------
Total Expenses: $4100.0
Enter the number of the expense to update (or 0 to go back to the main menu): 0

Expense Tracker System
1. Add Expense
2. View Expenses
3. Update Expense
4. Delete Expense
5. Exit
Select an option: 2

View Expenses
1. All Expenses
2. Expenses by Date
3. Expenses by Category
4. Expenses by Location
5. Back to Main Menu
Select an option: 1
Expenses loaded from file: expenses.txt
```



```
All Expenses:
------------------------------------------------------------------------------------------------
No.  Amount ($)     Date          Category          Location
------------------------------------------------------------------------------------------------
1   $1500.00      2023-09-10    Groceries         RR Nagar
2   $1200.00      2023-10-03    Shopping          JP Nagar
3   $1400.00      2023-10-06    Mall              MG Road
------------------------------------------------------------------------------------------------
Total Expenses: $4100.0

Expense Tracker System
1. Add Expense
2. View Expenses
3. Update Expense
4. Delete Expense
5. Exit
Select an option: 5
Exiting Expense Tracker System. Goodbye!
Expenses saved to file: expenses.txt
PS C:\Users\shree\OneDrive\Desktop>
```

Ln 275, Col 1   Spaces: 4   UTF-8   CRLF   {} Java   Go Live
Live Share   tabnine starter
Search
ENG IN   20:54   17-09-2023



expenses
File   Edit   View

```
No.   Amount ($)      Date          Category          Location
1    $1500.00       2023-09-10    Groceries         RR Nagar
2    $1200.00       2023-10-03    Shopping          JP Nagar
3    $1400.00       2023-10-06    Mall              MG Road

Total Expenses: $4100.0
```

Ln 1, Col 1   100%   Windows (CRLF)   UTF-8

CONCLUSION

The Expense Tracker mini-project in Java offers a practical solution for tracking expenses. It provides the following features and highlights:

The application allows users to:

Add new expenses, including the amount, date, category, and location.

View expenses in various ways: all expenses, by date, by category, and by location.

Update existing expenses with new details.

Delete expenses that are no longer needed.

Maintain data persistence by storing expenses in a structured text file (expenses.txt).

Key aspects of the project include:

A user-friendly command-line interface (CLI) that guides users through different tasks.

Data validation to ensure the accuracy and integrity of user input.

Error handling to gracefully manage exceptions during file operations and user interactions.

Modularity in code organization, making it easy to understand and maintain.

A structured file format with serial numbers to track and display expenses in the expenses.txt file.

The inclusion of a total balance summary in both the application interface and the saved file.

Overall, this Expense Tracker mini-project serves as a foundational example of expense tracking functionality in Java. It can be further expanded and improved, potentially incorporating additional features like user authentication, graphical user interfaces (GUIs), and data visualization for a more comprehensive expense management system.