INTERNSHIP REPORT

EXPENSE TRACKER SYSTEM

A report submitted in partial fulfillment of the requirements for the Award of Degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

SATYAMSETTI SIVABABU

21B91A05S0

Under Supervision of

MR RAFIKH

Henotic Technology Pvt Ltd, Hyderabad

(Duration: 5th July 2023 to 5th September 2023)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

S.R.K.R. ENGINEERING COLLEGE

(Autonomous)

SRKR MARG, CHINNA AMIRAM, BHIMAVARAM-534204, A.P

(Recognized by A.I.C.T.E New Delhi) (Accredited by NBA & NAAC)

(Affiliated to JNTU, KAKINADA)

SAGI RAMA KRISHNAM RAJU ENGINEERING COLLEGE

(Autonomous)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Summer Internship Report titled "EXPENSE TRACKER SYSTEM" is the bonafide work done by Mr/Mrs. SATYAMSETTI SIVABABU (21B91A05S0) at the end of second year second semester at Henotic Technology Pvt Ltd, Hyderabad from 5th July 2023 to 5th September 2023 in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering.

Department Internship coordinator **Dean -T & P Cell** **Head of the Department**

ACKNOWLEDGEMENT

I would like to extend my heartfelt gratitude to **Mr.Rafikh** working in **Henotic IT Solutions Pvt Ltd in Hyderabad**, for affording me the valuable opportunity to embark on an enriching internship journey within the organization. His belief in me and the chance he provided will forever be appreciated

I am also deeply indebted to the incredible team at **Henotic IT Solutions Pvt Ltd Hyderabad**, who worked alongside me during this internship. Their unwavering patience, openness, and collaborative spirit not only made the workplace enjoyable but also fostered an environment conducive to growth and learning. Their support was instrumental in making this experience truly remarkable.

I wish to express my profound gratitude to **Principal Prof. M. Jagapathi Raju sir** for the facilities and resources provided, which were instrumental in the successful completion of my internship. I would like to acknowledge and thank my Head of the Department **Prof. DR. V Chandra Sekhar sir**, for his invaluable constructive criticism and guidance throughout my internship, which played a pivotal role in enhancing my skills and knowledge.

I am also indebted to **Sri. T. Srinivasa Rao sir**, the internship coordinator of the Department of CSE, for their unwavering support and valuable advice that not only helped me secure this internship but also guided me in successfully completing it.

I am extremely grateful to my department staff members and friends who helped me in successful completion of this internship.

S. SIVABABU

21B91A05S0

TABLE OF CONTENTS

S.NO	CONTENTS	PAGE NO
1	ABSTRACT	05
2	INTERNSHIP OBJECTIVES	06
3	PROBLEM STATEMENT AND SOLUTION	07
4	MODULES	08
5	SOFTWARE REQUIREMENTS	09
6	TECHNOLOGIES	10
7	CODING	11-20
8	OUTPUT SCREENSHOTS	21-24
9	CONCLUSION	25

ABSTRACT

PROJECT TITLE: EXPENSE TRACKER SYSTEM

The project aims to develop a Java application that allows users to track their daily income and expenses, and manage their personal finance. The application uses data structures and algorithms (DSA) to store, retrieve, and to manipulate the data efficiently. The application also uses SQLite as the database to store the data persistently. The application provides various operations such as adding, viewing, editing, and deleting income and expense records, setting budget limits, generating reports, and visualizing the data using charts and graphs.

The application also implements security features such as user authentication and encryption. The project demonstrates the use of DSA concepts such as arrays, lists, stacks, queues, trees, graphs, hashing, sorting, and searching in developing a practical and useful application. The project also evaluates the performance of the application in terms of time and space complexity. The project concludes that the application is effective and efficient in helping users to manage their daily income and expenses, and improve their financial literacy.

INTERNSHIP OBJECTIVES

- Internships are generally thought of to be reserved for college students looking to gain experience in a particular field.
- However, a wide array of people can benefit from Training Internships in order to receive real world experience and develop their skills.
- An objective for this position should emphasize the skills you already possess in the area and your interest in learning more.
- Internships are utilized in a number of different career fields, including architecture, engineering, healthcare, economics, advertising and many more
- . Some internship is used to allow individuals to perform scientific research while others are specifically designed to allow people to gain first-hand experience working.
- Utilizing internships is a great way to build your resume and develop skills that can be emphasized in your resume for future jobs.
- When you are applying for a Training Internship, make sure to highlight any special skills or talents that can make you stand apart from the rest of the applicants so that you have an improved chance of landing the positions.

PROBLEM STATEMENT

Managing personal finances is a challenging task for many people, especially when they have multiple sources of income and expenses. It is difficult to keep track of how much money is spent and saved, and on what categories. Moreover, it is hard to analyze the spending patterns and trends, and to plan a budget accordingly. Therefore, there is a need for a system based on Data Structures and Algorithms using Java that can help users to track and manage their expenses in an easy and efficient way.

SOLUTION

The solution is to develop an expense tracker system using Java as the programming language and DSA as the core concept. The system will allow users to record their income and expenses, categorize them into different types, and store them in a database. The system will also provide features such as:

- Generating reports and charts to visualize the income and expenses data, and to show the spending habits and trends of the user.
- Setting a monthly budget and sending notifications to the user when the budget is exceeded or close to the limit.

The system will use various data structures and algorithms to implement these features, such as:

- Arrays, lists, stacks, queues, and hash tables to store and manipulate the income and expenses data.
- Trees and graphs to represent the hierarchical and relational structure of the data, and to perform operations such as traversal, search, insertion, deletion, and sorting.

The system will use Java as the programming language, as it is an object-oriented, platform-independent, and widely used language. Java also provides various libraries and frameworks that can facilitate the development of the system

The system will be beneficial for the users, as it will help them to:

- Track and manage their income and expenses in a convenient and organized way.

MODULES

➢ EXPENSE

- DEFINES THE EXPENSE CLASS WHICH REPRESENTS THE SINGLE EXPENSE RECORD

➢ EXPENSETRACKER

- DEFINES THE EXPENSETRACKER CLASS, WHICH CONTAINS THE METHODS TO PERFORM VARIOUS OPERATIONS ON THE EXPENSES.
- VARIOUS OPERATIONS ARE
- ADD EXPENSE
- VIEW EXPENSE
- DELETE EXPENSE
- EDIT EXPENSE
- VIEWALL EXPENSES
- VIEW EXPENSE SUMMARY

➢ MAIN

- CONTAINS THE MAIN METHOD, WHICH RUNS THE APPLICATION.

SOFTWARE REQUIREMENTS

Operating system:

Windows 11 Ultimate

Coding Language: **JAVA**

BASIC SQL QUERIES

BASIC KNOWLWDGE ON SQLITE

IDE Used:

Eclipse (IDE) Professional.

TECHNOLOGIES

JAVA LANGUAGE:

Java is a versatile, high-level programming language widely used in a variety of applications, Java is an object-oriented, class-based programming language renowned for its platform independence, reliability, and extensive libraries. Developed by Sun Microsystems (now owned by Oracle), Java has earned its popularity due to its "Write Once, Run Anywhere".

ECLIPSE(IDE):

An Integrated Development Environment (IDE) in Java is a software application

1.**Code Editing**: Java IDEs offer advanced code editing features such as syntax highlighting, auto-completion, and code formatting to enhance productivity and code quality

2**. Project Management**: IDEs help developers organize their projects by providing project templates, source code version control integration, and build tools for managing dependencies.

3. **Debugging**: Debugging tools in an IDE allow developers to identify and fix issues in their Java code. Developers can set breakpoints, step through code, inspect variables, and view stack traces.

4. **Compiler and Build Tools**: Java IDEs often come with integrated compilers and build tools like Ant, Maven, or Gradle, streamlining the process of building and running Java application

CODING

EXPENSE MODULE:

```
private static class Expense {
    private String date;
    private String description;
    private double amount;
    private String category;

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

    // Getters and setters for the Expense class

    public String getDate() {
       return date;
    }

    public void setDate(String date) {
       this.date = date;
    }
```

```
public String getDescription() {
    return description;
}

public void setDescription(String description) {
    this.description = description;
}

public double getAmount() {
    return amount;
}

public void setAmount(double amount) {
    this.amount = amount;
}

public String getCategory() {
    return category;
}

public void setCategory(String category) {
    this.category = category;
}
```

```
    }
```

EXPENSETRACKER MODULE:

```
import java.util.*;


class ExpenseTracker {
    private HashMap<String, List<Expense>> expenses;
    private HashMap<String, Double> categories;

    public ExpenseTracker() {
        this.expenses = new HashMap<>();
        this.categories = new HashMap<>();
    }

    public void addExpense(String date, String description, double amount, String
category) {
        Expense expense = new Expense(date, description, amount, category);
        expenses.computeIfAbsent(date, k -> new ArrayList<>()).add(expense);
        categories.put(category, categories.getOrDefault(category, 0.0) + amount);
        System.out.println("Expense added successfully!");
    }

    public void editExpense(String date, int index, String newDescription, double
newAmount, String newCategory) {
        List<Expense> expenseList = expenses.get(date);
```

```
        if (expenseList != null && index >= 0 && index < expenseList.size()) {
            Expense expense = expenseList.get(index);
            categories.put(expense.getCategory(), categories.get(expense.getCategory()) -
expense.getAmount());
            expense.setDescription(newDescription);
            expense.setAmount(newAmount);
            expense.setCategory(newCategory);
            categories.put(newCategory, categories.getOrDefault(newCategory, 0.0) +
newAmount);
            System.out.println("Expense edited successfully!");
        } else {
            System.out.println("Invalid date or index!");
        }
    }

    public void deleteExpense(String date, int index) {
        List<Expense> expenseList = expenses.get(date);
        if (expenseList != null && index >= 0 && index < expenseList.size()) {
            Expense expense = expenseList.get(index);
            categories.put(expense.getCategory(), categories.get(expense.getCategory()) -
expense.getAmount());
            expenseList.remove(index);
            System.out.println("Expense deleted successfully!");
        } else {
            System.out.println("Invalid date or index!");
```

```
        }
    }

    public void viewExpenses(String date) {
        if (expenses.containsKey(date)) {
            double totalAmount = 0;
            System.out.println("Expenses for " + date + ":");
            System.out.printf("%-5s %-20s %-30s %-10s %-10s%n", "Index", "Date",
"Description", "Amount", "Category");
            List<Expense> expenseList = expenses.get(date);
            for (int i = 0; i < expenseList.size(); i++) {
                Expense expense = expenseList.get(i);
                System.out.printf("%-5d %-20s %-30s RS %-9.2f %-10s%n", i,
expense.getDate(), expense.getDescription(),
                        expense.getAmount(), expense.getCategory());
                totalAmount += expense.getAmount();
            }
            System.out.println("Total expenses for " + date + " =  RS " +
String.format("%.2f", totalAmount));
        } else {
            System.out.println("No expenses recorded for " + date);
        }
    }

    public void viewAllExpenses() {
```

```
        if (expenses.isEmpty()) {
            System.out.println("No expenses recorded yet.");
        } else {
            System.out.println("All Expenses:");
            System.out.printf("%-5s %-20s %-30s %-10s %-10s%n", "Index", "Date",
"Description", "Amount", "Category");
            int index = 0;
            for (List<Expense> expenseList : expenses.values()) {
                for (Expense expense : expenseList) {
                    System.out.printf("%-5d %-20s %-30s  RS  %-9.2f %-10s%n", index,
expense.getDate(),
                            expense.getDescription(), expense.getAmount(),
expense.getCategory());
                    index++;
                }
            }
        }
    }

    public void viewExpenseSummary() {
        if (categories.isEmpty()) {
            System.out.println("No expenses recorded yet.");
        } else {
            System.out.println("Expense Summary:");
            System.out.printf("%-20s %-10s%n", "Category", "Total Amount");
```

```
        for (Map.Entry<String, Double> entry : categories.entrySet()) {
            System.out.printf("%-20s RS %-9.2f%n", entry.getKey(), entry.getValue());
        }
    }
}
```

MAIN CLASS MODULE:

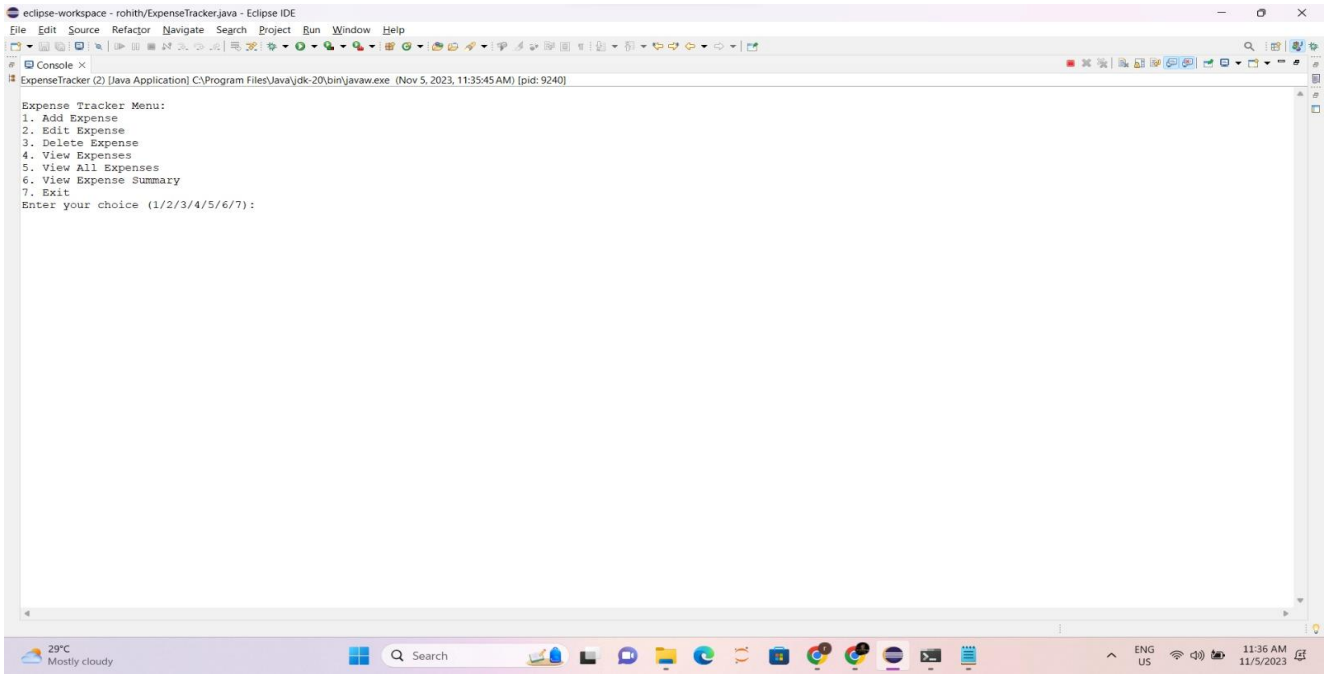
```
public static void main(String[] args) {
    ExpenseTracker expenseTracker = new ExpenseTracker();
    Scanner scanner = new Scanner(System.in);

    while (true) {
        System.out.println("\nExpense Tracker Menu:");
        System.out.println("1. Add Expense");
        System.out.println("2. Edit Expense");
        System.out.println("3. Delete Expense");
        System.out.println("4. View Expenses");
        System.out.println("5. View All Expenses");
        System.out.println("6. View Expense Summary");
        System.out.println("7. Exit");

        System.out.print("Enter your choice (1/2/3/4/5/6/7): ");
        int choice = scanner.nextInt();
```

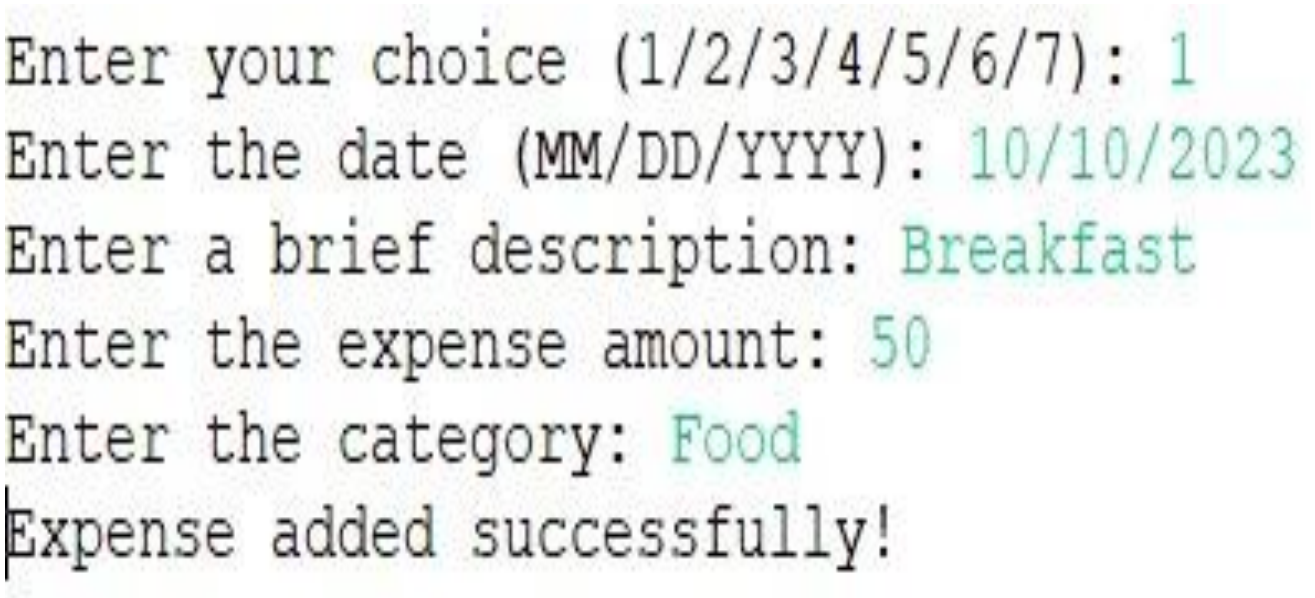
```
switch (choice) {
    case 1:
        System.out.print("Enter the date (MM/DD/YYYY): ");
        String date = scanner.next();
        System.out.print("Enter a brief description: ");
        scanner.nextLine();
        String description = scanner.nextLine();
        System.out.print("Enter the expense amount: ");
        double amount = scanner.nextDouble();
        System.out.print("Enter the category: ");
        scanner.nextLine();
        String category = scanner.nextLine();
        expenseTracker.addExpense(date, description, amount, category);
        break;
    case 2:
        System.out.print("Enter the date (MM/DD/YYYY): ");
        date = scanner.next();
        System.out.print("Enter the index of the expense to edit: ");
        int index = scanner.nextInt();
        System.out.print("Enter a new description: ");
        scanner.nextLine();
        description = scanner.nextLine();
        System.out.print("Enter a new expense amount: ");
        amount = scanner.nextDouble();
```

```
            System.out.print("Enter a new category: ");
            scanner.nextLine();
            category = scanner.nextLine();
            expenseTracker.editExpense(date, index, description, amount, category);
            break;
        case 3:
            System.out.print("Enter the date (MM/DD/YYYY): ");
            date = scanner.next();
            System.out.print("Enter the index of the expense to delete: ");
            index = scanner.nextInt();
            expenseTracker.deleteExpense(date, index);
            break;
        case 4:
            System.out.print("Enter the date (MM/DD/YYYY): ");
            date = scanner.next();
            expenseTracker.viewExpenses(date);
            break;
        case 5:
            expenseTracker.viewAllExpenses();
            break;
        case 6:
            expenseTracker.viewExpenseSummary();
            break;
        case 7:
```

```
                System.out.println("Exiting Expense Tracker...");
                scanner.close();
                System.exit(0);
            default:
                System.out.println("Invalid choice. Please try again.");
        }
    }
  }
}
```

OUTPUT SCREENSHOTS





```
Enter your choice (1/2/3/4/5/6/7): 2
Enter the date (MM/DD/YYYY): 10/10/2023
Enter the index of the expense to edit: 1
Enter a new description: Xerox
Enter a new expense amount: 110
Enter a new category: Education
Expense edited successfully!

Expense Tracker Menu:
1. Add Expense
2. Edit Expense
3. Delete Expense
4. View Expenses
5. View All Expenses
6. View Expense Summary
7. Exit
Enter your choice (1/2/3/4/5/6/7): 4
Enter the date (MM/DD/YYYY): 10/10/2023
Expenses for 10/10/2023:
Index Date                 Description                  Amount     Category
0     10/10/2023           Breakfast                    RS 50.00     Food
1     10/10/2023           Xerox                        RS 110.00    Education
2     10/10/2023           Lunch                        RS 100.00    Food
3     10/10/2023           Bus                          RS 70.00     Transportation
Total expenses for 10/10/2023 =  RS 330.00
```

```
Enter your choice (1/2/3/4/5/6/7): 3
Enter the date (MM/DD/YYYY): 10/10/2023
Enter the index of the expense to delete: 3
Expense deleted successfully!

Expense Tracker Menu:
1. Add Expense
2. Edit Expense
3. Delete Expense
4. View Expenses
5. View All Expenses
6. View Expense Summary
7. Exit
Enter your choice (1/2/3/4/5/6/7): 4
Enter the date (MM/DD/YYYY): 10/10/2023
Expenses for 10/10/2023:
Index Date                 Description                  Amount     Category
0     10/10/2023           Breakfast                    RS 50.00     Food
1     10/10/2023           Xerox                        RS 110.00    Education
2     10/10/2023           Lunch                        RS 100.00    Food
Total expenses for 10/10/2023 =  RS 260.00
```

```
Enter your choice (1/2/3/4/5/6/7): 4
Enter the date (MM/DD/YYYY): 10/10/2023
Expenses for 10/10/2023:
Index Date                Description                Amount     Category
0     10/10/2023          Breakfast                  RS 50.00     Food
1     10/10/2023          Notes                      RS 100.00    Education
2     10/10/2023          Lunch                      RS 100.00    Food
3     10/10/2023          Bus                        RS 70.00     Transportation
Total expenses for 10/10/2023 =  RS 320.00
```

```
Enter your choice (1/2/3/4/5/6/7): 5
All Expenses:
Index Date                Description                Amount     Category
0     18/10/2023          petrol                      RS  200.00   Transport
1     12/10/2023          Birthday Party              RS  4500.00  Entertainment
2     15/10/2023          Movie                       RS  150.00   Entertainment
3     10/10/2023          Breakfast                   RS  50.00    Food
4     10/10/2023          Notes                       RS  100.00   Education
5     10/10/2023          Lunch                       RS  100.00   Food
6     10/10/2023          Bus                         RS  70.00    Transportation
```

```
Enter your choice (1/2/3/4/5/6/7): 6
Expense Summary:
Category                  Total Amount
Entertainment             RS 4650.00
Education                 RS 100.00
Transportation            RS 70.00
Transport                 RS 200.00
Food                      RS 150.00
```

```
Enter your choice (1/2/3/4/5/6/7): 7
Exiting Expense Tracker...
```

CONCLUSION

In this project, we developed an expense tracker system in Java using DSA which allows users to manage their daily expenses and keep track of their spending habits. We used various data structures and algorithms, such as arrays, linked lists, stacks, queues, trees, graphs, heaps, multisets, sorting algorithms, and search algorithms, to implement the core functionalities and features of the system. We also used a MySQL database to store and retrieve the user data.

Our system provides an easy, fast, and smooth tracking system between the expense and the income of the user. It also offers some opportunities that help the user to sustain all financial activities, such as setting budgets, generating reports, analysing trends, and giving suggestions. Our system is based on the existing knowledge and practice in the field of expense tracking and management, and it aims to enhance the user experience and satisfaction.

To conclude, our project demonstrates the use and application of data structures and algorithms in Java to create an effective and efficient expense tracker system. Our project contributes to the field of expense tracking and management, and it provides a useful and practical tool for the user to manage their daily expenses and improve their financial well-being. Our project also shows the importance and relevance of data structures and algorithms in computer science and programming, and how they can be used to solve complex problems and create high-performance applications.