Hindusthan College of Engineering and Technology
An Autonomous Institution, Approved by AICTE, New Delhi, Affiliated to Anna University, Chennai
Valley Campus, Coimbatore – 641 032, Tamil Nadu, INDIA

DEPARTMENT OF INFORMTION TECHNOLOGY

INTERNSHIP – REPORT

2024 - 2025

"JAVA PROGRAMMING INTERN"

JAVA

MOTIONCUT

Submitted By

JEGAN K – 720722110057

DEPARTMENT OF INFORMATION TECHNOLOGY

DECLARATION

This is to certify that the Internship report submitted by Jegan K **(720722110057)** is the Bonafide record of the **Information Technology** student who visited MotionCut for Java Programming Intern - Internship during the Even semester of the academic year 2024-2025**.**

Class advisor

Mr.Praveen Kumar R
Assistant Professor
Department of Information Technology
Hindusthan College of Engineering and Technology,
Coimbatore-641032

Head of the department

Dr. M.Sabrigiriraj
Professor and Head
Department of Information Technology
Hindusthan College of Engineering and Technology,
Coimbatore-641032

DEPARTMENT OF INFORMATION TECHNOLOGY

CHECK LIST

- Acknowledgement
- Internship Offer Letter from Company
- Internship Completion Certificate
- Internship Report

DEPARTMENT OF INFORMATION TECHNOLOGY

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to all the people and organizations that contributed to the success of my internship experience. This internship has been an invaluable learning opportunity, and I would not have been able to accomplish it without the guidance, support, and encouragement from various individuals and organizations.

I would like to thank **MotionCut** for providing me with the opportunity to be part of their team. I am especially grateful to my **internship supervisor,** for their continuous support, mentorship, and for offering valuable insights into the industry. Their expertise and guidance helped me develop both professionally and personally, and I greatly appreciate the time and effort they invested in my learning.

I would also like to extend my gratitude to the entire team at **MotionCut**, whose cooperation and willingness to share their knowledge made my internship experience incredibly enriching. The collaborative work environment allowed me to apply the skills I have learned in my academic journey and gain hands-on experience in the field.

I am deeply thankful to my **college principal**, **Dr.J.Jaya** for their support in granting me the opportunity to pursue this internship and for their continuous encouragement throughout my academic journey. Their leadership and guidance have been crucial in shaping my educational experience.

My heartfelt thanks also go to my **Head of Department, Dr. M.Sabrigiriraj** and my **Class Advisor, Mr.R.Praveen Kumar.R**, for their valuable advice, encouragement, and assistance in making this internship possible. Their belief in the importance of practical experience in enhancing academic learning has been a guiding force in this endeavor.

I would also like to express my sincere appreciation to my **friends** and **family** for their unwavering support throughout my internship. Their encouragement, understanding, and belief in me have been a constant source of motivation.

Finally, I would like to thank everyone who helped and supported me during this internship for making this experience one of the most rewarding and educational chapters of my life.

INTERNSHIP OFFER LETTER

INTERNSHIP COMPLETION CERTIFICATE

INTERNSHIP REPORT

INDEX

INTRODUCTION

1.1 Internship & Its Importance

The internship program is a crucial part of academic learning, bridging the gap between theoretical knowledge and practical application. This report outlines the experience gained during my internship tenure at **MotionCut**, carried out from **5th Jan 2025 to 5th Feb 2025** , as part of the curriculum of **B.Tech Information Technology** at **Hindusthan College of Engineering & Technology, Coimbatore**.

The objective of this internship was to gain practical exposure to real-world work environments, understand industry workflows, and enhance technical and soft skills. It provided a platform to apply academic concepts in a professional setting, contributing to both personal and professional growth.

This report provides a detailed account of the organization, the tasks undertaken during the internship, challenges encountered, skills developed, and overall learning outcomes.

1.2 Objective

- To gain practical experience in Java programming and related technologies.
- To understand the software development life cycle and project workflows.
- To improve problem-solving, debugging, and code optimization skills.
- To develop applications using core and advanced Java features.
- To enhance teamwork and communication skills in a professional setting.

1.3 Period of Training

- **Duration:** 4 Weeks / 1 Month
- **Start Date:** 5th January 2025
- **End Date:** 5th February 2025
- **Organization:** MotionCut

1.4 Outcomes

- Gained hands-on experience in core Java concepts such as OOP, collections, exception handling, and multithreading.
- Learned to build simple console and GUI-based Java applications.
- Understood version control using Git and collaborative coding practices.
- Improved debugging skills and learned to work with IDEs like Eclipse/IntelliJ.
- Gained exposure to real-world software development processes and teamwork dynamics.

COMPANY PROFILE

2.1 Company Overview

MotionCut is an innovative online educational platform dedicated to bridging the gap between academic learning and industry requirements through project-based virtual internships. Established in 2023, MotionCut has rapidly evolved into a prominent provider of hands-on IT training, offering flexible, self-paced programs that emphasize real-world applications.

2.2 Mission & Vision

MotionCut's mission is to empower aspiring professionals by providing practical learning experiences that prepare them for successful careers in the tech industry. The platform envisions a future where education is accessible, practical, and aligned with industry demands, enabling students to transform their passion for technology into thriving careers.

2.3 Key Offerings

- **Virtual Internships**: MotionCut offers a range of online internships in areas such as Java Programming, Web Development, Python Programming, and Frontend Web Development.
- **Project-Based Learning**: The platform emphasizes learning by doing, allowing interns to work on practical projects that simulate real industry challenges.
- **Flexible Durations**: Internship programs range from 2 weeks to 6 months, accommodating various learning needs and schedules.
- **Professional Certification**: Upon completion, interns receive AICTE-verified certificates with unique IDs, recognized by universities and employers, enhancing their professional credibility.

2.4 Contact

- **Website :** Motioncut.in / Motioncut.shop
- **Email :** Motioncut@outlook.com
- **LinkedIn :** www.linkedin.com/company/motioncut/

TECHNOLOGY USED

3.1 Development Tools

- **Programming Language: Java** is used for the entire application logic, including data storage, URL processing, and command-line or optional web interface.
- **Application:**
  - Object-Oriented Programming: Design is structured around classes and methods.
  - Error Handling: Java's try-catch blocks manage exceptions like invalid input.
  - Hashing: Custom hash function for generating short links.
- **Development Platform:** Eclipse or IntelliJ IDEA for Java development.
- **Testing:** Manual Testing
- **Version Control**: Git/GitHub for tracking versions and collaboration (commonly expected in such internships).
- **Submission Platform:** Google Forms.
- **Data Structures & Algorithm:** Hashmap / Dictionary / Similar
- **I/O Interface:** CLI (Command Line Interface) - System in/out, Scanner
- **Project Allotment:** Telegram

MODULE DESCRIPTION

4.1 Work Done during Internship

During the course of my one-month Java Programming Internship at MotionCut, I actively engaged in a structured and immersive learning journey that revolved around real-world Java application development. This internship was designed to strengthen my foundation in programming through a sequence of hands-on, project-based tasks that incrementally introduced more advanced concepts.

I worked on three core projects that gave me exposure to essential programming practices including object-oriented design, error handling, file management, command-line interfaces, and data structuring. Each project helped refine my problem-solving abilities, logic-building skills, and my understanding of Java's capabilities in building practical solutions.

In addition to coding assignments, I became proficient with tools like Git and GitHub, developed habits of writing clean and maintainable code, and improved my ability to independently manage, test, and document projects. Below is a detailed summary of the work carried out during each phase of the internship.

4.1 Modules

4.1.1 Project 1

- **Title:** Daily Expense Tracker
- **Objective:** To build a simple, console-based application that allows users to log and manage their daily expenses.
- **Work Summary:**
  - Designed and implemented an Expense class to encapsulate data such as amount, category, and description.
  - Developed an ExpenseManager class to manage the addition of expenses and generate summaries based on time periods (daily, weekly, monthly).
  - Integrated file handling to persist expense data using Java's FileWriter and BufferedReader classes.
  - Implemented user input handling using loops and conditional logic.
  - Gained experience with object-oriented principles including encapsulation and modular design.
  - Conducted manual testing and debugging to ensure functionality and robustness.
- **Technologies Used:** Java SE, CLI, File I/O, OOP

4.1.2 Project 2

- **Title: Link Shortener Development**
- **Objective:** To develop a Java application that shortens long URLs into unique short codes and expands them back.
- **Work Summary:**
  - Created classes to handle the core logic of URL shortening and expanding.
  - Designed a custom hash function to generate short URLs and implemented logic to ensure uniqueness and handle collisions.
  - Used HashMap to manage and retrieve URL mappings efficiently.

- Implemented basic error handling to manage duplicate entries and invalid URLs.
- Developed a simple command-line interface for user interaction.
- Explored optional features like file persistence for saving mappings across sessions.

- **Technologies Used:** Java SE, Hashing, Data Structures, CLI, Error Handling

4.1.3 Project 3

- **Title: Expense Tracker (Advanced)**
- **Objective:** To build an enhanced version of the expense tracker with additional features like user registration, category-wise expense tracking, and a better user interface.
- **Work Summary:**
  - Implemented user registration functionality to create personalized sessions for multiple users.
  - Enhanced the data model with features like category-wise summation and filtering capabilities.
  - Utilized object-oriented design principles such as inheritance and abstraction for cleaner architecture.Improved error handling mechanisms for better reliability.
  - Incorporated file handling techniques to load and save user-specific data, ensuring persistence across sessions.
  - Designed a more structured CLI with menus and options to navigate various features.
- **Technologies Used:** Java SE, File I/O, OOP (Inheritance, Abstraction), CLI, Data Filtering

4.2 Other Activities

- Learned to use Git and GitHub for version control and code management.
- Submitted each project via public GitHub repositories following best practices for documentation and code organization.
- Improved problem-solving and debugging skills through iterative development and testing.
- Gained confidence in applying core Java concepts to build practical, real-world applications.

4.3 Source Code

Github Repository:

https://github.com/Jegan1412/MotionCut-Java-Internship/tree/main

PROOF OF MATERIAL

5.1 Project Details

JAVA PROGRAMMING INTERNSHIP
PROJECT 1

Project Title: Daily Expense Tracker

Project Overview

The **Daily Expense Tracker** is a practical and user-friendly tool designed to help users manage their finances efficiently. The project allows users to log daily expenses, categorize them (e.g., food, travel, utilities), and view summarized reports for specific time periods such as a day, week, or month. The application also stores all data in a text file, ensuring that users can retrieve and review their expense history at any time.

This project introduces real-world applications of programming concepts and is an excellent stepping stone toward building more advanced applications.

Key Features of the Project

1. **Add Expenses**: Users can log expenses by specifying the amount, category, and a short description.
2. **View Summaries**: The application can display:

- Total expenses for the day.

- Total expenses for the week.

- Total expenses for the month.

3. **Save Data**: All expense records are stored in a text file for future reference.
4. **Concepts Covered**:

- Object-Oriented Programming (OOP): Design classes like Expense and ExpenseManager to encapsulate data and logic.

- File Handling: Use Java's file I/O to store and retrieve expense data.

-Basic Loops and User Input: Implement loops for iterative operations and handle user input seamlessly.

Tips for Success

1. **Plan Your Code Structure**: Identify the necessary classes, methods, and their interactions before starting the implementation. Suggested classes include:

- **Expense** for storing details of individual expenses.

- **ExpenseManager** for managing and processing all expenses.

2. **Start Simple**: Begin by implementing basic functionalities like adding an expense and displaying all expenses before moving to advanced features like summaries and file handling.
3. **Test Incrementally**: Test each feature as you implement it to ensure smooth functionality and identify bugs early.
4. **Organize Your Code**: Use clear, meaningful variable and method names for readability and maintainability.

Project: Link Shortener Development **Duration:** 7 Days

Project Overview

As part of your ongoing Java programming internship, we are excited to introduce the Week 2 project, "Link Shortener Development." This project will provide you with valuable hands-on experience in Java programming and web development.

The main goal of this project is to create a simple Link Shortener application using Java. A Link Shortener is a tool that takes a long URL and generates a short, unique identifier, allowing users to share concise links. This project will give you insights into data structures, algorithms, and basic web application development.

Project Objectives

To accurately outline the scope of work required for a project, it is crucial to first identify its objectives. Pinpointing what the project hopes to accomplish will assist in determining its inclusions and limitations.

- Implement a Link Shortener system in Java.
- Develop functions to shorten long URLs and expand short URLs to their original form.
- Ensure the uniqueness of short URLs and handle potential collisions.
- Explore basic error handling and user feedback.
- Gain practical experience in Java programming, data structures, and algorithmic thinking.

Requirements and Features

- Use Java as the primary programming language.
- Create a class or set of classes that manage the shortening and expanding of URLs.

Project: Expense Tracker **Duration:** 7 Days

Project Overview

As we embark on Week 3 of your internship program, I am excited to introduce the project that will further enhance your skills in Java Programming. The upcoming task is the creation of a "Expense Tracker" as part of your Java Programming internship.

The Expense Tracker project aims to create a robust and user-friendly application that allows users to manage and monitor their expenses efficiently. This program will serve as a hands-on experience in applying Java programming concepts to develop a functional and practical software solution.

Project Objectives

To accurately outline the scope of work required for a project, it is crucial to first identify its objectives. Pinpointing what the project hopes to accomplish will assist in determining its inclusions and limitations.

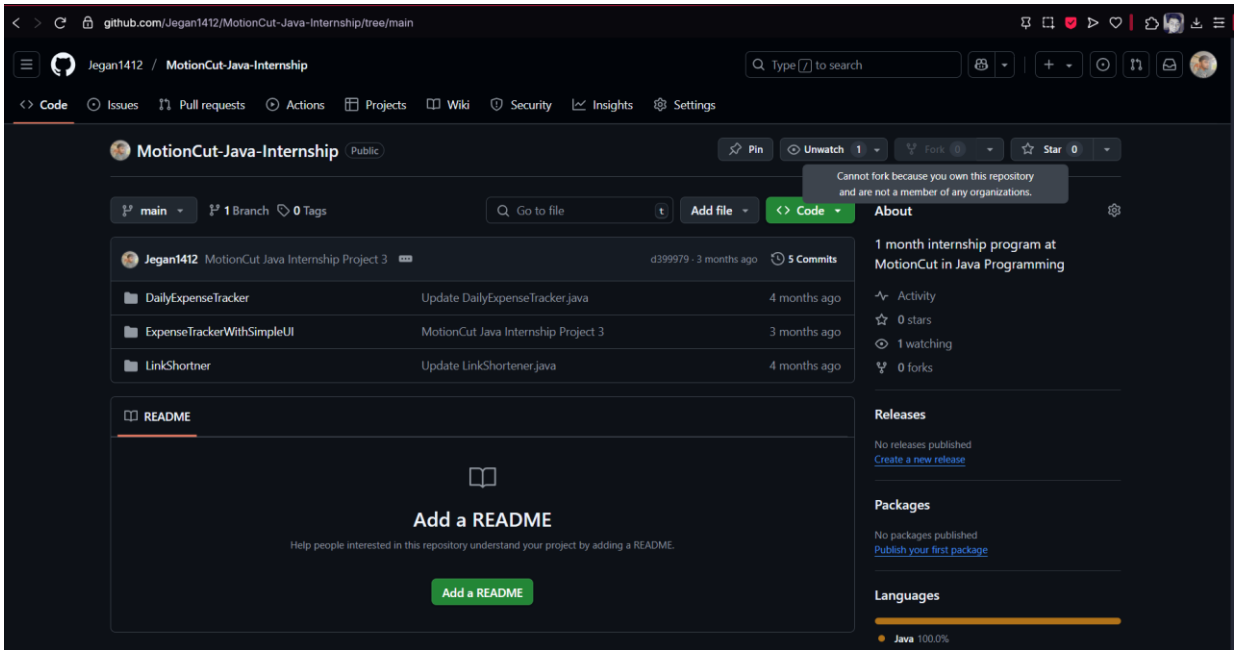
- **Understanding Java Fundamentals:** Strengthen your grasp on core Java concepts such as classes, objects, inheritance, and exception handling.
- **Data Handling:** Implement data structures and algorithms to manage and organize expense data effectively.
- **User Interface Design:** Develop a simple and intuitive user interface for users to input, edit, and view their expenses.
- **File Handling:** Utilize file I/O operations to save and load expense data, ensuring persistence across sessions.
- **Exception Handling:** Implement robust error handling to enhance the reliability of the application.

Requirements and Features

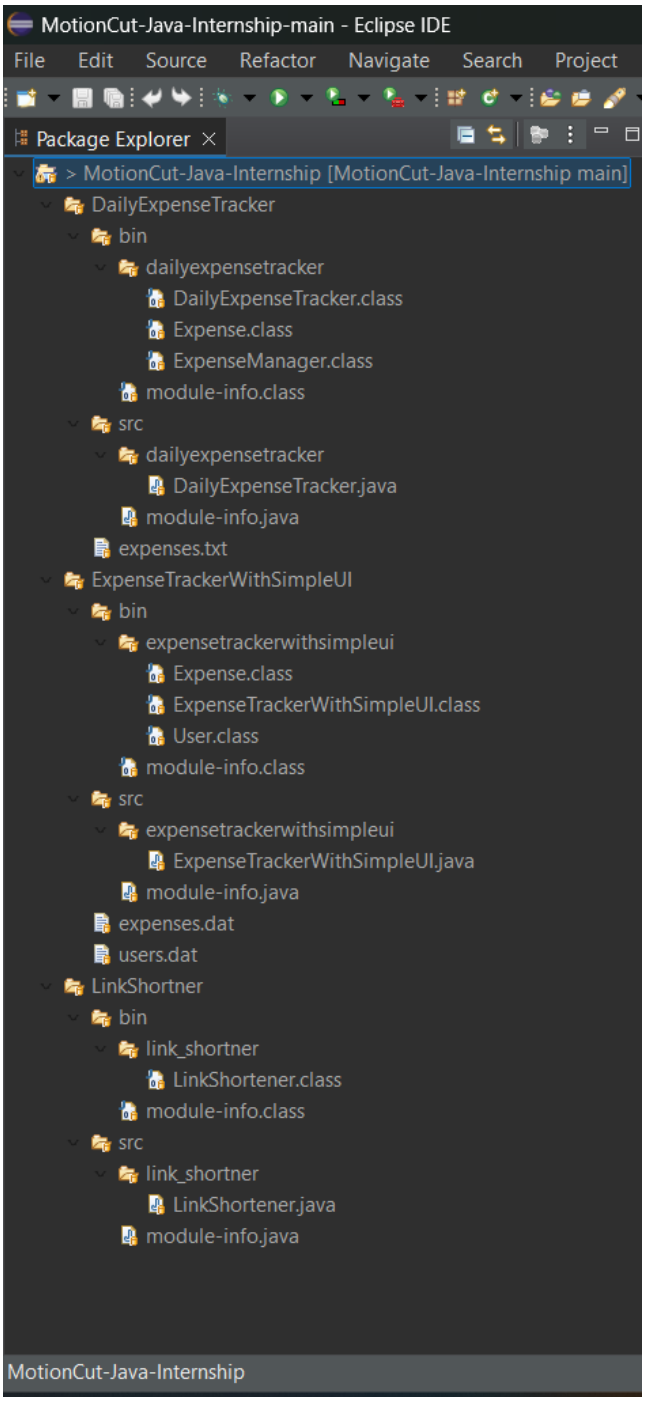
- **User Registration:** Allow users to create accounts to personalize and secure their expense data.
- **Expense Entry:** Enable users to input details of their expenses, including the date, category, and amount.
- **Expense Listing:** Provide a clear and organized view of all entered expenses, with the ability to sort or filter.
- **Category-wise Summation:** Implement functionality to calculate and display the total expenses for each category.
- **Persistence:** Save and load expense data to and from files for a seamless user experience across sessions.

- Implement a basic hash function for generating short URLs.
- Include error handling to address scenarios like duplicate long URLs and invalid short URLs.
- Optionally, consider persisting data to maintain link mappings between sessions.
- Develop a simple command-line interface (CLI) or web-based interface for user interaction.

5.2 Github Details



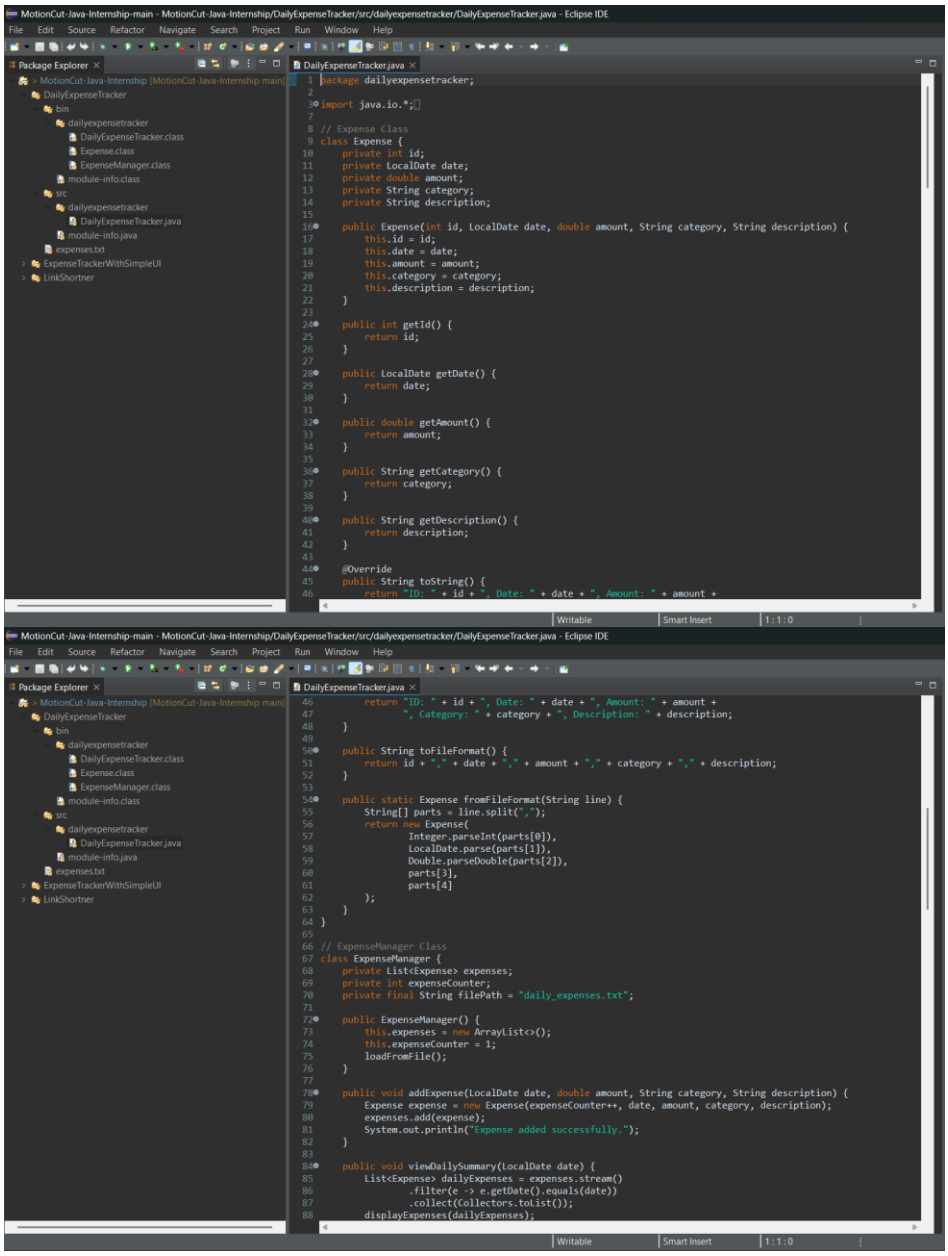
5.3 Project Directory

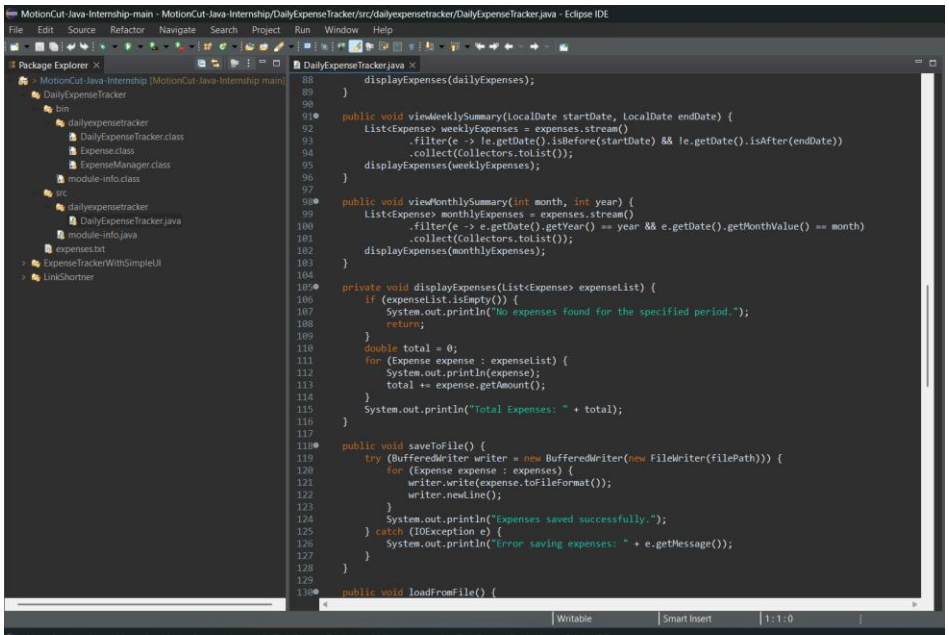


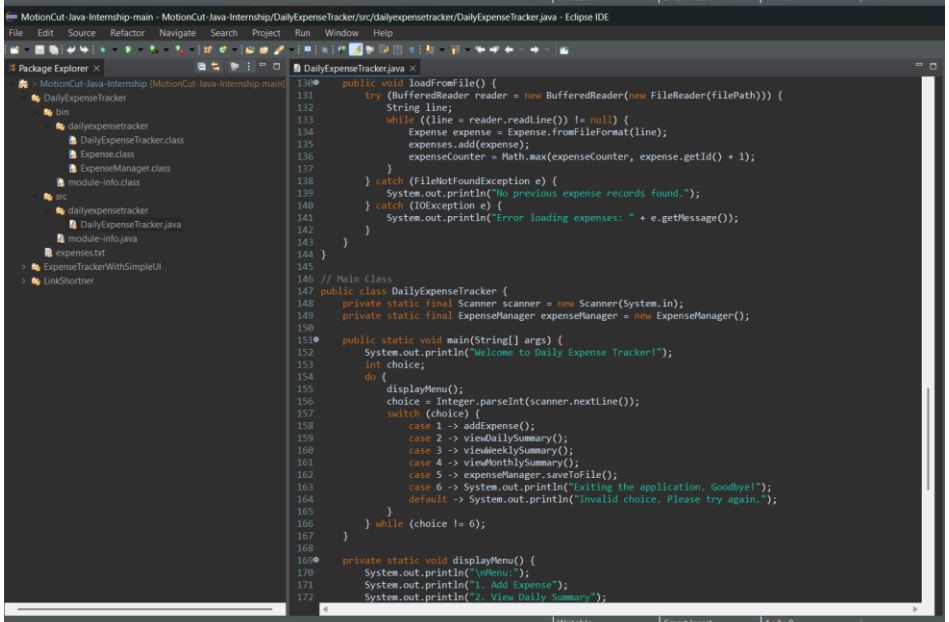
5.4 Code & Output Samples

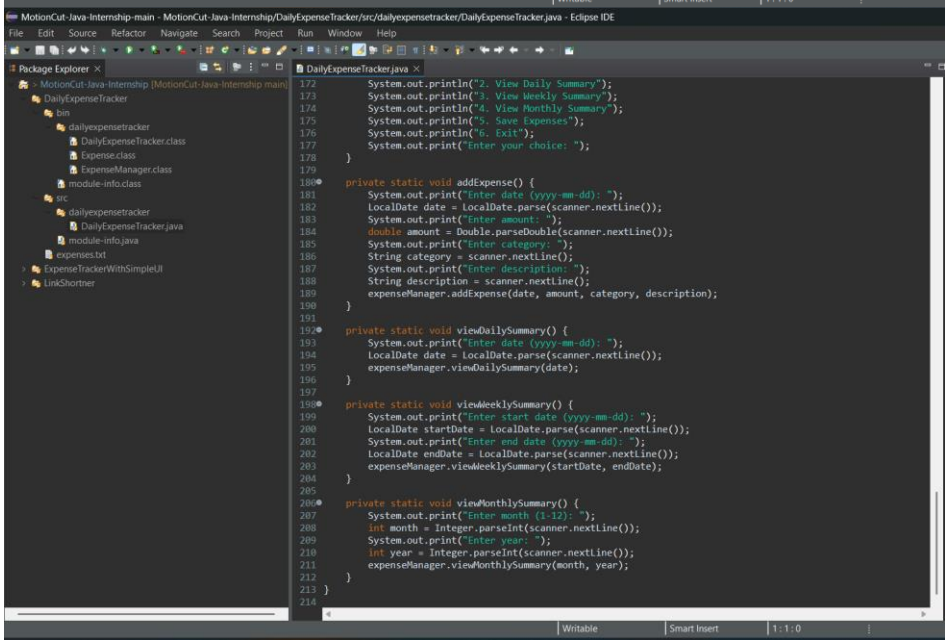
5.4.1 Project 1

Code:

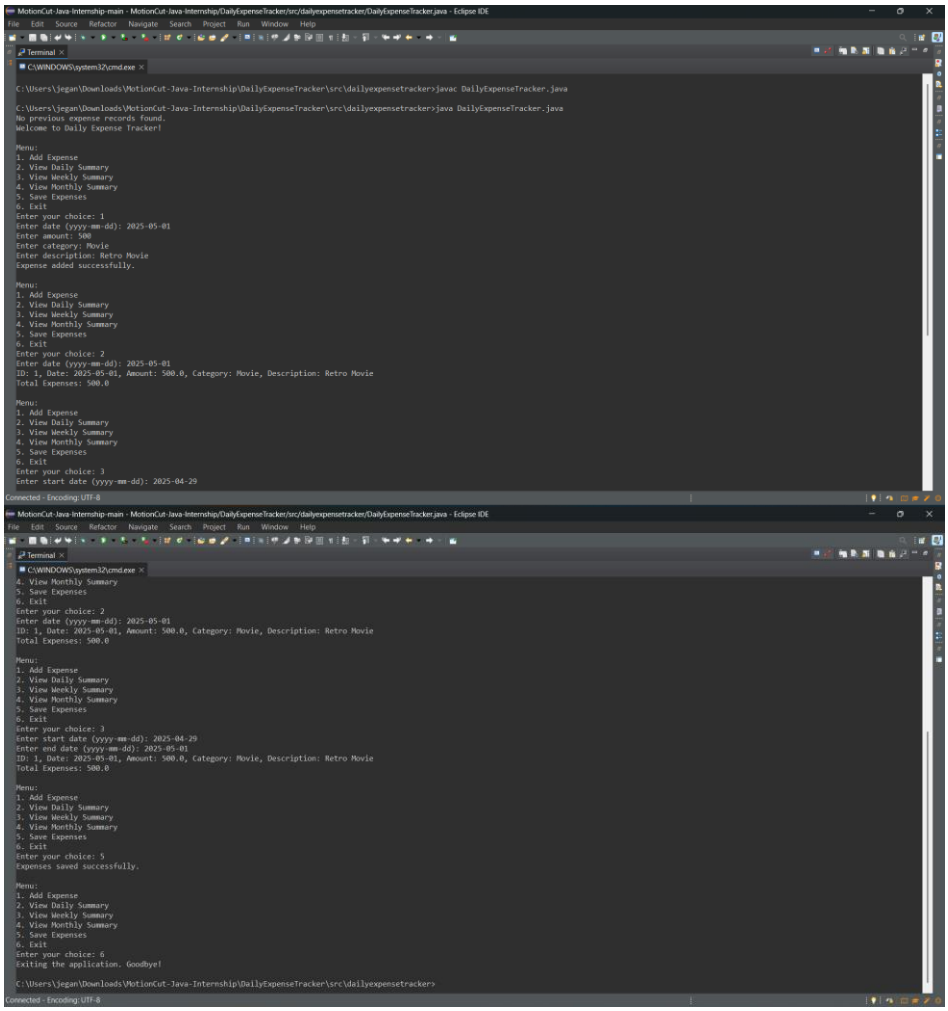



MotionCut-Java-Internship-main - MotionCut-Java-Internship/DailyExpenseTracker/src/dailyexpensetracker/DailyExpenseTracker.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
DailyExpenseTracker.java
displayExpenses(dailyExpenses);
}
public void viewWeeklySummary(LocalDate startDate, LocalDate endDate) {
List<Expense> weeklyExpenses = expenses.stream()
.filter(e -> !e.getDate().isBefore(startDate) && !e.getDate().isAfter(endDate))
.collect(Collectors.toList());
displayExpenses(weeklyExpenses);
}
public void viewMonthlySummary(int month, int year) {
List<Expense> monthlyExpenses = expenses.stream()
.filter(e -> e.getDate().getYear() == year && e.getDate().getMonthValue() == month)
.collect(Collectors.toList());
displayExpenses(monthlyExpenses);
}
private void displayExpenses(List<Expense> expenseList) {
if (expenseList.isEmpty()) {
System.out.println("No expenses found for the specified period.");
return;
}
double total = 0;
for (Expense expense : expenseList) {
System.out.println(expense);
total += expense.getAmount();
}
System.out.println("Total Expenses: " + total);
}
public void saveToFile() {
try (BufferedWriter writer = new BufferedWriter(new FileWriter(filePath))) {
for (Expense expense : expenses) {
writer.write(expense.toFileFormat());
writer.newLine();
}
System.out.println("Expenses saved successfully.");
} catch (IOException e) {
System.out.println("Error saving expenses: " + e.getMessage());
}
}
public void loadFromFile() {
Writable
Smart Insert
1 : 1 : 0


MotionCut-Java-Internship-main - MotionCut-Java-Internship/DailyExpenseTracker/src/dailyexpensetracker/DailyExpenseTracker.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
DailyExpenseTracker.java
public void loadFromFile() {
try (BufferedReader reader = new BufferedReader(new FileReader(filePath))) {
String line;
while ((line = reader.readLine()) != null) {
Expense expense = Expense.fromFileFormat(line);
expenses.add(expense);
expenseCounter = Math.max(expenseCounter, expense.getId() + 1);
}
} catch (FileNotFoundException e) {
System.out.println("No previous expense records found.");
} catch (IOException e) {
System.out.println("Error loading expenses: " + e.getMessage());
}
}
}
// Main Class
public class DailyExpenseTracker {
private static final Scanner scanner = new Scanner(System.in);
private static final ExpenseManager expenseManager = new ExpenseManager();
public static void main(String[] args) {
System.out.println("Welcome to Daily Expense Tracker!");
int choice;
do {
displayMenu();
choice = Integer.parseInt(scanner.nextLine());
switch (choice) {
case 1 -> addExpense();
case 2 -> viewDailySummary();
case 3 -> viewWeeklySummary();
case 4 -> viewMonthlySummary();
case 5 -> expenseManager.saveToFile();
case 6 -> System.out.println("Exiting the application. Goodbye!");
default -> System.out.println("Invalid choice. Please try again.");
}
} while (choice != 6);
}
private static void displayMenu() {
System.out.println("\nMenu:");
System.out.println("1. Add Expense");
System.out.println("2. View Daily Summary");
Writable
Smart Insert
1 : 1 : 0


MotionCut-Java-Internship-main - MotionCut-Java-Internship/DailyExpenseTracker/src/dailyexpensetracker/DailyExpenseTracker.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
DailyExpenseTracker.java
System.out.println("2. View Daily Summary");
System.out.println("3. View Weekly Summary");
System.out.println("4. View Monthly Summary");
System.out.println("5. Save Expenses");
System.out.println("6. Exit");
System.out.print("Enter your choice: ");
}
private static void addExpense() {
System.out.print("Enter date (yyyy-mm-dd): ");
LocalDate date = LocalDate.parse(scanner.nextLine());
System.out.print("Enter amount: ");
double amount = Double.parseDouble(scanner.nextLine());
System.out.print("Enter category: ");
String category = scanner.nextLine();
System.out.print("Enter description: ");
String description = scanner.nextLine();
expenseManager.addExpense(date, amount, category, description);
}
private static void viewDailySummary() {
System.out.print("Enter date (yyyy-mm-dd): ");
LocalDate date = LocalDate.parse(scanner.nextLine());
expenseManager.viewDailySummary(date);
}
private static void viewWeeklySummary() {
System.out.print("Enter start date (yyyy-mm-dd): ");
LocalDate startDate = LocalDate.parse(scanner.nextLine());
System.out.print("Enter end date (yyyy-mm-dd): ");
LocalDate endDate = LocalDate.parse(scanner.nextLine());
expenseManager.viewWeeklySummary(startDate, endDate);
}
private static void viewMonthlySummary() {
System.out.print("Enter month (1-12): ");
int month = Integer.parseInt(scanner.nextLine());
System.out.print("Enter year: ");
int year = Integer.parseInt(scanner.nextLine());
expenseManager.viewMonthlySummary(month, year);
}
}
Writable
Smart Insert
1 : 1 : 0

Output:


MotionCut-Java-Internship-main - MotionCut-Java-Internship/DailyExpenseTracker/src/dailyexpensetracker/DailyExpenseTracker.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Terminal
C:\WINDOWS\system32\cmd.exe
C:\Users\jegan\Downloads\MotionCut-Java-Internship\DailyExpenseTracker\src\dailyexpensetracker>javac DailyExpenseTracker.java
C:\Users\jegan\Downloads\MotionCut-Java-Internship\DailyExpenseTracker\src\dailyexpensetracker>java DailyExpenseTracker.java
No previous expense records found.
Welcome to Daily Expense Tracker!
Menu:
1. Add Expense
2. View Daily Summary
3. View Weekly Summary
4. View Monthly Summary
5. Save Expenses
6. Exit
Enter your choice: 1
Enter date (yyyy-mm-dd): 2025-05-01
Enter amount: 500
Enter category: Movie
Enter description: Retro Movie
Expense added successfully.
Menu:
1. Add Expense
2. View Daily Summary
3. View Weekly Summary
4. View Monthly Summary
5. Save Expenses
6. Exit
Enter your choice: 2
Enter date (yyyy-mm-dd): 2025-05-01
ID: 1, Date: 2025-05-01, Amount: 500.0, Category: Movie, Description: Retro Movie
Total Expenses: 500.0
Menu:
1. Add Expense
2. View Daily Summary
3. View Weekly Summary
4. View Monthly Summary
5. Save Expenses
6. Exit
Enter your choice: 3
Enter start date (yyyy-mm-dd): 2025-04-29
Connected - Encoding: UTF-8
MotionCut-Java-Internship-main - MotionCut-Java-Internship/DailyExpenseTracker/src/dailyexpensetracker/DailyExpenseTracker.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Terminal
C:\WINDOWS\system32\cmd.exe
4. View Monthly Summary
5. Save Expenses
6. Exit
Enter your choice: 2
Enter date (yyyy-mm-dd): 2025-05-01
ID: 1, Date: 2025-05-01, Amount: 500.0, Category: Movie, Description: Retro Movie
Total Expenses: 500.0
Menu:
1. Add Expense
2. View Daily Summary
3. View Weekly Summary
4. View Monthly Summary
5. Save Expenses
6. Exit
Enter your choice: 3
Enter start date (yyyy-mm-dd): 2025-04-29
Enter end date (yyyy-mm-dd): 2025-05-01
ID: 1, Date: 2025-05-01, Amount: 500.0, Category: Movie, Description: Retro Movie
Total Expenses: 500.0
Menu:
1. Add Expense
2. View Daily Summary
3. View Weekly Summary
4. View Monthly Summary
5. Save Expenses
6. Exit
Enter your choice: 5
Expenses saved successfully.
Menu:
1. Add Expense
2. View Daily Summary
3. View Weekly Summary
4. View Monthly Summary
5. Save Expenses
6. Exit
Enter your choice: 6
Exiting the application. Goodbye!
C:\Users\jegan\Downloads\MotionCut-Java-Internship\DailyExpenseTracker\src\dailyexpensetracker>
Connected - Encoding: UTF-8

SAMPLE CODING

6.1 Code & Output Samples

Code: Project 2

File Name: LinkShortner.java

```
package link_shortner;
import java.util.HashMap;
import java.util.Scanner;

public class LinkShortener {

    private final HashMap<String, String> shortToLong = new HashMap<>();
    private final HashMap<String, String> longToShort = new HashMap<>();
    private final String characters =
"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";
    private final int shortURLLength = 6;

    // Method to shorten a URL
    public String shortenURL(String longURL) {
        if (longToShort.containsKey(longURL)) {
            return longToShort.get(longURL); // Return existing short URL
        }

        String shortURL;
        do {
            shortURL = generateShortURL();
        } while (shortToLong.containsKey(shortURL)); // Ensure uniqueness

        shortToLong.put(shortURL, longURL);
        longToShort.put(longURL, shortURL);
        return shortURL;
    }

    // Method to expand a short URL
    public String expandURL(String shortURL) {
        return shortToLong.getOrDefault(shortURL, "Error: Short URL not found");
    }

    // Basic hash function to generate a short URL
    private String generateShortURL() {
        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < shortURLLength; i++) {
            int index = (int) (Math.random() * characters.length());
            sb.append(characters.charAt(index));
        }
        return sb.toString();
    }

    // Command-line interface
```

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

        System.out.println("Welcome to the Link Shortener Application!");
        while (true) {
            System.out.println("\nChoose an option:");
            System.out.println("1) Shorten URL");
            System.out.println("2) Expand URL");
            System.out.println("3) Exit");
            int choice = scanner.nextInt();
            scanner.nextLine(); // Consume newline character

            switch (choice) {
                case 1:
                    System.out.println("Enter the long URL:");
                    String longURL = scanner.nextLine();
                    if (longURL.isEmpty()) {
                        System.out.println("Error: URL cannot be empty!");
                    } else {
                        String shortURL = linkShortener.shortenURL(longURL);
                        System.out.println("Shortened URL: " + shortURL);
                    }
                    break;
                case 2:
                    System.out.println("Enter the short URL:");
                    String shortURL = scanner.nextLine();
                    String expandedURL = linkShortener.expandURL(shortURL);
                    System.out.println("Original URL: " + expandedURL);
                    break;
                case 3:
                    System.out.println("Exiting the application. Goodbye!");
                    scanner.close();
                    return;
                default:
                    System.out.println("Invalid choice! Please try again.");
            }
        }
    }
}
```

Output:

C:\Users\jegan\Downloads\MotionCut-Java-Internship\LinkShortner\src\link_shortner>javac LinkShortener.java

C:\Users\jegan\Downloads\MotionCut-Java-Internship\LinkShortner\src\link_shortner>java LinkShortener.java Welcome to the Link Shortener Application!

Choose an option:

1. Shorten URL
2. Expand URL
3. Exit

1

Enter the long URL: https://github.com/Jegan1412/MotionCut-Java-Internship/tree/main/LinkShortner

Shortened URL: yl9ey4

Choose an option:

1. Shorten URL
2. Expand URL
3. Exit

2

Enter the short URL: yl9ey4

Original URL: https://github.com/Jegan1412/MotionCut-Java-Internship/tree/main/LinkShortner

Choose an option:

1. Shorten UR
2. L
3. Expand URL
4. Exit

3

Exiting the application. Goodbye!

CONCLUSION

The one-month Java Programming Internship at MotionCut has been a transformative journey, significantly enhancing my programming skills and practical understanding of software development. Through a series of structured, real-world projects, I have transitioned from grasping fundamental concepts to applying advanced programming techniques.

Throughout the internship, I also gained proficiency in using development tools such as Eclipse and version control systems like Git and GitHub. These tools were instrumental in managing codebases, tracking changes, and collaborating effectively.

The self-paced, project-driven approach of the MotionCut internship provided a conducive environment for learning and growth. It encouraged independent problem-solving, critical thinking, and continuous improvement. The mentorship and structured feedback further enriched the learning experience, ensuring that each project contributed meaningfully to my skillset.

In conclusion, this internship has not only bolstered my technical competencies in Java programming but also prepared me for future challenges in the software development industry. The projects undertaken have equipped me with a robust foundation to pursue advanced roles and continue learning in the ever-evolving field of technology.