**Industrial Internship Report on**

**”comsole based expense tracker application”**

**Prepared by**

**Aakriti soni**

|  |
| --- |
| Executive Summary |

|  |
| --- |
| This report provides details of the Industrial Internship provided by upskill Campus and The IoT Academy in collaboration with Industrial Partner UniConverge Technologies Pvt Ltd (UCT).  This internship was focused on a project/problem statement provided by UCT. We had to finish the project including the report in 6 weeks’ time.  A console-based expense tracker application in Java allows users to record, manage, and analyze their expenses through a simple text-based interface. It provides users with essential functionalities like adding new expenses, viewing existing expenses, deleting expenses, calculating total expenses, and generating summary reports.  This internship gave me a very good opportunity to get exposure to Industrial problems and design/implement solution for that. It was an overall great experience to have this internship. |

**TABLE OF CONTENTS**

[1 Preface 4](#_Toc139702806)

[2 Introduction 14](#_Toc139702807)

[2.1 About UniConverge Technologies Pvt Ltd 14](#_Toc139702808)

[2.2 About upskill Campus 18](#_Toc139702809)

[2.3 Objective 19](#_Toc139702810)

[2.4 Reference 19](#_Toc139702811)

[2.5 Glossary **Error! Bookmark not defined.**](#_Toc139702812)

[3 Problem Statement 20](#_Toc139702813)

[4 Existing and Proposed solution 22](#_Toc139702814)

[5 Proposed Design/ Model 23](#_Toc139702815)

[5.1 High Level Diagram (if applicable) 23](#_Toc139702816)

[5.2 Low Level Diagram (if applicable) **Error! Bookmark not defined.**](#_Toc139702817)

[5.3 Interfaces (if applicable) **Error! Bookmark not defined.**](#_Toc139702818)

[6 Performance Test 24](#_Toc139702819)

[6.1 Test Plan/ Test Cases 24](#_Toc139702820)

[6.2 Test Procedure 25](#_Toc139702821)

[6.3 Performance Outcome 26](#_Toc139702822)

[7 My learnings 27](#_Toc139702823)

[8 Future work scope 28](#_Toc139702824)

# Preface

**Summary of the whole 6 weeks’ work:-**

**In week one, I got to know about the java. Here I have the detailed description about**

Java is a high-level, object-oriented programming language developed by James Gosling and his team at Sun Microsystems (later acquired by Oracle). It was released in 1995 and has since become one of the most popular programming languages in the world. Java was designed to be platform-independent, meaning it can run on various operating systems without modification, thanks to its "write once, run anywhere" (WORA) philosophy.

The topic which I have learned is History of java. History of java : James Gosling, the father of Java, was intent on building a low cost, hardwareindependent software platform using C++. James Gosling initiated the Java language project in June 1991 for use in one of his many set-top Box projects. After knowing all about the history of the java will moved further where we will execute our first java program For executing fist program I have used tool called eclipse , eclipse is an IDE which is used to run the program in written in java . After that they provide a document inside that document contain all information about this internship and the details about the project

**In week two, I got to know about the various conditional statements in java. Below is the detailed description about it:**

Java continues to be one of the most widely used programming languages worldwide. It is useful in mobile development, backend development, and cloud-based solutions. Even trendy tech niches like IoT and Big Data use it. As a result, being a Java developer is a lucrative career. It is used by millions of developers worldwide to create software applications. As a Java developer, you will be responsible for designing, developing, and maintaining software applications. You will also be involved in debugging and testing these applications. You need to ensure they meet customer requirements. You must also know object-oriented programming principles. Have experience with databases like SQL Server, Oracle and MySQL. With the right skill set and experience, you can become a successful Java developer.

**Java Developer Skills:**

As a Java developer, it is essential to possess the right skills to be successful. Java developers need to have a good understanding of objectoriented programming concepts and design patterns.

**Career Growth Of Java Developer**:

The Java platform has seen several improvements, modifications, and additions. One thing is for certain: it will remain popular for at least the next ten years. Although it may be a bold claim given the number of technologies that have appeared just to slowly go away. Java has withstood the test of time and is still loved by developers. The fact that number of Java programmers is growing quickly. It suggests that to advance your career as a Java developer and remain relevant, you must constantly learn new skills.

Career Opportunities in Java You'll need to search through several job listings after acquiring the necessary equipment to launch your Java profession: • Junior Developer • Senior Developer • Java Web Developer • Architect • Java Android Developer • Java EE Developer

**In week three, I got to learn about how python help in SEO and the must known topics for data science:**

Introduction For developing incredible websites, applications, operating systems, or digital technology, software engineers deserve praise. Before beginning their professions, software engineers typically learn to write in a few different programming languages. Every software engineer who wants to succeed in their job must have a solid understanding of Java, one of the most crucial programming languages. Software engineers can only be full-stack Java developers if they are skilled in both front-end and back-end tasks.

**About who is full stack java developer:**

The full list of full-stack java developer abilities must be mastered for someone to possess them. These abilities cover every Java-compatible framework and utility. The entire collection of Java technologies, including web architecture, REST APIs, front-end frameworks, and others, that are necessary for creating applications and websites are included in the full-stack Java developer skill set

**StringBuilder** in Java is a class used to build mutable, or changeable, strings of characters. The StringBuilder class acts as a replacement for the String Class in Java. It creates a mutable sequence of characters instead of an immutable one as the String Class does. The functions of the StringBuilder and StringBuffer classes are extremely similar because both create changeable sequences of characters as an alternative to the String Class. However, synchronization is where the StringBuilder class and StringBuffer class differ. Contrary to the StringBuffer class, the StringBuilder class does not guarantee synchronization

**In week four I got to learn about the Roadmap for learning python and relation between the NumPy and pandas.**

Here is the short summary about topic I had learned this week:

1.Interface in java

2.Abstract class in java

3.Method Overloading in java

**INTERFACE IN JAVA:**

In Java, an interface is a reference type that defines a contract or a set of methods that a class must implement. It is a blueprint for a class and defines the methods that the class must provide.

An interface is declared using the interface keyword in Java. It can contain method declarations, constants (static final variables), and default methods (methods with implementations) since Java 8. However, interfaces cannot have instance variables or constructors.

**ABSTRACT CLASS IN JAVA:**

In Java, an abstract class is a class that cannot be instantiated and is designed to be subclassed. It serves as a blueprint for derived classes, providing common implementation details and defining abstract methods that must be implemented by its subclasses.

**METHOD OVERLOADING IN JAVA:**

Method overloading in Java refers to the ability to define multiple methods in a class with the same name but different parameters. It allows you to have multiple methods with the same name but different argument lists, enabling you to perform similar operations with different types of inputs

**In week five I got to learn about the NumPy and Pandas and about its operations in detail.**

# Success ladder to the corporate world

**To create Awareness of Placement & Selection Process in the Campuses to the Students**

To prepare Students for Placement Activities

To Understand Importance of

a) Preparation before Placement

b) Performance during Placement

**Before Campus Placement Starts...**

Collect Information on

• Industry

• Company

• Future Growth

• Visit Web Sites

Get information through Friends Current happenings in the company

**What Do Companies Expect From You?**

Ability to learn.

•Ability to work in a team.

• Positive attitude and passion towards work.

Aptitude that meets the job description.

**A Typical Campus Recruitment Day...**

• Pre-placement Talk (PPT)

• Scrutiny of CV's • Aptitude Test

• Group Discussion Personal Interview

• Final Review by Interviewing

Committee

• Final Selection – Offer

**STEPS**

1.Aptitude test

2.Group Discussion

3.Interview

**During Personal Interview...**

Points to be noted:

Creativity

Confidence

Temperament

Physical Setup

Psychological Setup Presence of Mind

Satisfying Ego

Facial/Verbal Expressions

Problem Solving Capability

Liveliness/Cheerfulness How do you ask questions

**Questions related to interview**

1.Tell me about yourself?

2.Why do you want to work at our company?

3.What are your strengths?

4.What is your Weakness?

5.Why should we hire you?

6.Tell me what you know about this company?

7.what are your career goal?

8.Finally, do you have any question to ask me?

**Post Interview**

Points to be reviewed:

Review the entire selection process

Find out improvement areas

Letter of thanks

Practice & be ready for next opportunity

**Other Frequently Asked Questions**

Tell me about yourself.

Why weren't your grades better?

What is your favorite subject? Tell me about your project?

What's your greatest achievement till date?

How do you usually go about solving a problem?

Why do you want to work in this industry?

**Campus Interview Don'ts**

Don't be late to the interview.

Don't carry an outdated CV.

Don't make negative comments about professors or project mates.

Don't falsify application materials or answers to interview questions.

Don't exhibit frustrations or a negative attitude in an interview. Don't take a cell phone call. Don't look at/reply to a text message.

Don't make excuses. Take responsibility of your actions.

Don't try to be someone you are not!

So, in this weak we have gain all the knowledge about the company placement and interview questions.

**About need of relevant Internship in career development:**

1. Hands-on Experience: Internships provide an opportunity to gain practical, real-world experience in your chosen field. It allows you to apply the knowledge and skills acquired in your academic studies to actual projects and tasks.

2. Skill Development: Internships help you develop and refine essential skills specific to your industry. Whether it's technical skills like programming, data analysis, or soft skills like communication and teamwork, internships offer a valuable platform for growth.

3. Networking Opportunities: Internships provide a chance to build connections with professionals and experts in your field. Networking during an internship can lead to mentorship, future job opportunities, and recommendations.

4. Industry Insight: Working in a real work environment provides valuable insights into the industry's dynamics, culture, and best practices. It helps you understand the day-to-day operations and challenges faced by professionals in the field.

5. Resume Building: Internship experiences enhance your resume and make you stand out to potential employers. Employers often value candidates with practical experience, and having relevant internships can significantly boost your chances of landing a full-time job.

6. Confidence Building: Successfully completing an internship can boost your confidence in your abilities and make you feel more prepared for future career challenges.

7. Learning Company Culture: Internships allow you to experience the culture of different organizations, helping you determine which work environment aligns best with your values and career goals.

8. Exploring Career Paths: Internships can help you clarify your career goals and interests. They provide an opportunity to try out different roles and industries before committing to a long-term career path.

9. References and Recommendations: A successful internship can lead to strong references and recommendations from supervisors and colleagues, which can be valuable assets when applying for jobs in the future.

10. Transition to Full-Time Job: Some internships can lead to full-time job offers if the employer is impressed with your performance during the internship period. It can be a direct pathway to starting your career with the same company.

11. Personal Growth: Internships offer valuable opportunities for personal growth and self-discovery. They challenge you to adapt to new situations, take on responsibilities, and develop a sense of professionalism.

**Brief about Quiz Game:-**

A console-based expense tracker application in Java is a simple program that allows users to manage their expenses through a text-based interface in the console or terminal. Users can add, view, and delete expenses, calculate total expenses, and generate summary reports. This type of application is a great project for learning Java and practicing fundamental programming concepts.

Key Features of the Console-Based Expense Tracker Application:

**User Interface:** The application presents a menu-driven interface with options for users to interact with the expense tracker. Users can choose the desired operation by entering corresponding numbers or characters.

**Data Storage:** The application uses a data structure (e.g., ArrayList or HashMap) to store expense records during its runtime. The data may also be saved to and loaded from a text file for persistence between application sessions.

**Add and Manage Expenses:** Users can add new expenses by providing details such as amount, description, and date. They can also view all existing expenses in a list format and delete specific expenses if needed.

**Calculate Total Expenses:** The application calculates and displays the total amount of all expenses entered by the user.

**Generate Reports:** Users can generate summary reports based on certain criteria, such as expenses within a specific time range or expenses for a particular category.

**Error Handling:** Proper error handling is implemented to handle invalid user inputs and prevent application crashes. For example, the application may request users to re-enter data if they input incorrect values.

High-level Overview of Implementation Steps:

**Expense Class:** Design a class to represent an expense entity with attributes like amount, description, and date.

**Data Storag**e: Choose an appropriate data structure (e.g., ArrayList<Expense>) to store the expense records during runtime.

**User Interface:** Implement a menu-driven user interface that displays options for adding, viewing, and deleting expenses, calculating total expenses, and generating reports.

**Add Expense:** Develop the logic to allow users to add new expenses by capturing relevant details and adding an Expense object to the data structure.

**View and Delete Expenses:** Implement functionalities to display all expenses and allow users to delete specific expenses based on their selection.

**Calculate Total Expenses:** Calculate the total expenses by iterating through the expense data structure and summing up the amounts.

**Generate Reports:** Implement logic to generate summary reports based on user preferences, such as date ranges or categories.

**Data Persistence:** Provide methods to save and load the expense data to and from a text file to maintain data between different application sessions.

**Error Handling:** Ensure proper validation of user inputs to handle errors gracefully and prevent unexpected application behavior.

**Testing and Refinement:** Thoroughly test the application and refine it based on user feedback and bug fixes.

By building a console-based expense tracker application in Java, you can gain practical experience in object-oriented programming, data handling, user input validation, and file handling, all while creating a useful tool to manage personal finances.

**Opportunity given by USC/UCT:**

I just want to Thank to each and everyone, who have helped you directly or indirectly for successfully completing this project as well as the report.

Also I want to thank UpSkill Campus and The IOT Academy for providing me such an interesting internship opportunity and also I want to thank the mentors who guided and helped throughout the learning duration.

# Introduction

## About UniConverge Technologies Pvt Ltd

A company established in 2013 and working in Digital Transformation domain and providing Industrial solutions with prime focus on sustainability and RoI.

For developing its products and solutions it is leveraging various**Cutting Edge Technologies e.g. Internet of Things (IoT), Cyber Security, Cloud computing (AWS, Azure), Machine Learning, Communication Technologies (4G/5G/LoRaWAN), Java Full Stack, Python, Front end**etc.



1. UCT IoT Platform **(****)**

**UCT Insight** is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable “insight” for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

* It enables device connectivity via industry standard IoT protocols - MQTT, CoAP, HTTP, Modbus TCP, OPC UA
* It supports both cloud and on-premises deployments.

It has features to  
• Build Your own dashboard  
• Analytics and Reporting  
• Alert and Notification  
• Integration with third party application(Power BI, SAP, ERP)  
• Rule Engine

 

1. **Smart Factory Platform (****)**

Factory watch is a platform for smart factory needs.

It provides Users/ Factory

* with a scalable solution for their Production and asset monitoring
* OEE and predictive maintenance solution scaling up to digital twin for your assets.
* to unleased the true potential of the data that their machines are generating and helps to identify the KPIs and also improve them.
* A modular architecture that allows users to choose the service that they what to start and then can scale to more complex solutions as per their demands.

Its unique SaaS model helps users to save time, cost and money.

 

1.  based Solution

UCT is one of the early adopters of LoRAWAN teschnology and providing solution in Agritech, Smart cities, Industrial Monitoring, Smart Street Light, Smart Water/ Gas/ Electricity metering solutions etc.

1. Predictive Maintenance

UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



## About upskill Campus (USC)

upskill Campus along with The IoT Academy and in association with Uniconverge technologies has facilitated the smooth execution of the complete internship process.

USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.





Seeing need of upskilling in self paced manner along-with additional support services e.g. Internship, projects, interaction with Industry experts, Career growth Services

<https://www.upskillcampus.com/>

upSkill Campus aiming to upskill 1 million learners in next 5 year



## The IoT Academy

The IoT academy is EdTech Division of UCT that is running long executive certification programs in collaboration with EICT Academy, IITK, IITR and IITG in multiple domains.

## Objectives of this Internship program

The objective for this internship program was to

 ☛ get practical experience of working in the industry.

 ☛ to solve real world problems.

 ☛ to have improved job prospects.

 ☛ to have Improved understanding of our field and its applications.

 ☛ to have Personal growth like better communication and problem solving.

## Reference

[1] Google:- https://www.google.com/

[2] eclipse

[3] Chrome:- <https://www.google.com/intl/en_in/chrome/>

[4] vs code

# Problem Statement:-

Given below is the explanation of the project Console Based Tracker Application

Java implementation of a console-based expense tracker application. It allows users to manage their expenses by recording, viewing, and generating reports. Below is a brief overview of the key components and functionalities of the application:

**Classes:**

**Expense:** Represents an individual expense entry with attributes like date, amount, category, and description.

**ExpenseCategory:** Represents a category for organizing expenses. It contains a list of expenses associated with that category.

**ExpenseManager:** Acts as the central manager for handling expenses and categories. It maintains lists of expenses and categories and provides methods to add or remove them.

**ExpenseValidator:** Provides static methods for validating expense details, such as amount and date.

**ConsoleHelper:** A utility class that handles console input/output operations and user interactions. It provides methods for reading various types of input and displaying messages/errors.

**Functionality:**

**Record Expense:** Users can add new expenses by entering the date, amount, category, and description. The application validates the input, and if valid, creates an Expense object and adds it to the expense list.

**Manage Categories:** Users can create new categories and delete existing ones. Each category can contain related expenses.

**View Expenses:** Users can view all recorded expenses along with their details, such as date, amount, category, and description.

**Generate Reports:** Users can generate two types of reports:

**Monthly Expense Report:** Calculates and displays the total expenses for a specific month and year.

**Category-wise Expense Report:** Calculates and displays the total expenses for a selected category.

**Error Handling:** The application performs error handling to ensure that users provide valid input for dates, amounts, and other fields.

**Main Method:**

The main method is the entry point of the application. It displays a welcome message and presents a menu-driven interface to the user. Users can choose various options, and the application responds accordingly.

**Usage:**

To use the application, you can run the main method in your Java IDE or command-line environment. The application will display the main menu, and you can follow the prompts to perform the desired operations.

**Note:**

The code provided is a basic implementation of a console-based expense tracker. Depending on your requirements, you can further enhance the application by adding more features, error handling, and data persistence (e.g., saving expenses to a file). Additionally, you may consider improving the user interface to make it more intuitive and user-friendly.

# Existing and Proposed solution

Provide summary of existing solutions provided by others, what are their limitations?

What is your proposed solution?

What value addition are you planning?

## Code submission (Github link) : <https://github.com/Aakriti-soni/UpSkill-Project>

## Report submission (Github link) : <https://github.com/Aakriti-soni/UpSkill-Project>

# Proposed Design/ Model

Given more details about design flow of your solution. This is applicable for all domains. DS/ML Students can cover it after they have their algorithm implementation. There is always a start, intermediate stages and then final outcome.

## High Level Diagram:-



Figure 1: HIGH LEVEL DIAGRAM OF CONSOLE BASED TRACKER APPLICATION

# Performance Test:-

Performance tests for the console-based expense tracker application.

**Performance Testing Considerations**:

Performance testing aims to evaluate how well an application performs under various conditions. For the console-based application, some key aspects to consider for performance testing are:

Response Time: Measure the time taken by the application to respond to user input and display results.

Throughput: Evaluate the number of operations the application can handle in a given time.

Memory Usage: Monitor the memory consumption of the application during different operations.

Scalability: Test the application's ability to handle an increasing number of expenses and categories without significant performance degradation.

## Test Plan/ Test Cases:-

Test cases for the console-based expense tracker application will help ensure its reliability, functionality, and accuracy. Below is an outline of the test plan, along with some sample test cases:

Test Plan for Console-Based Expense Tracker Application:

Introduction:

Overview of the application and its purpose.

Testing objectives and scope.

Roles and responsibilities of the testing team.

Test Environment:

Description of the testing environment (e.g., operating system, Java version).

Tools used for testing (e.g., JUnit, TestNG).

Test Strategy:

Overview of the testing approach (e.g., functional testing, user acceptance testing).

Testing methodologies (e.g., manual testing, automated testing).

Test Scenarios:

List of test scenarios covering various aspects of the application.

Test Cases:

Detailed test cases for each test scenario, including steps, expected results, and test data.

Test Data:

Sample test data used for testing different scenarios.

Defect Reporting:

Guidelines for reporting defects, including severity and priority levels.

Test Schedule:

Timeline for executing the test cases.

Risk and Mitigation:

Identification of potential risks and strategies to mitigate them.

Acceptance Criteria:

Criteria for determining when the application is ready for release.

Sample Test Scenarios and Test Cases:

Test Scenario: Record a new expense.

Test Case 1: Enter valid date, amount, category, and description.

Test Case 2: Enter an invalid date (e.g., future date) and verify error handling.

Test Case 3: Enter an invalid amount (e.g., negative amount) and verify error handling.

Test Scenario: View expenses.

Test Case 4: Verify if all recorded expenses are displayed correctly.

Test Case 5: Check for correct formatting and presentation of expense details.

Test Scenario: Generate a monthly expense report.

Test Case 6: Enter a valid year and month and verify the total expense for that period.

Test Case 7: Enter an invalid month (e.g., 13) and verify error handling.

Test Scenario: Create a new category.

Test Case 8: Enter a valid category name and verify successful creation.

Test Case 9: Enter a category name that already exists and verify error handling.

Test Scenario: Delete a category.

Test Case 10: Select a valid category and verify successful deletion.

Test Case 11: Select an invalid category index and verify error handling.

Test Scenario: Generate a category-wise expense report.

Test Case 12: Select a valid category and verify the total expense for that category.

Test Case 13: Select an invalid category index and verify error handling.

Test Scenario: Verify data persistence.

Test Case 14: Add expenses, close the application, and re-open to check if data is retained.

Test Scenario: Stress testing.

Test Case 15: Perform load testing with a large number of expenses to check performance.

## Test Procedure:-

Test Environment Setup:

Set up the testing environment with the required configurations, including the operating system and Java version.

Ensure that the necessary libraries and dependencies are installed and available.

Test Data Preparation:

Prepare sample test data for different test scenarios, including valid and invalid data.

Test Case Execution:

Execute each test case based on the test scenarios defined in the test plan.

Follow the test steps and provide the test data as input as specified in the test cases.

Observations and Result Recording:

Observe the application's behavior during test case execution.

Record the actual results for each test case, including any errors or deviations from the expected outcome.

Defect Reporting:

If any defects or issues are identified during the test case execution, report them in a defect tracking tool or document.

Include details such as steps to reproduce, expected and actual results, and severity/priority of the defects.

Regression Testing:

After fixing defects, perform regression testing to ensure that the changes do not introduce new issues.

Data Persistence Testing:

Verify the data persistence feature by adding expenses, closing the application, and then re-opening it to check if the data is retained.

Stress Testing (Optional):

If stress testing is part of the testing plan, conduct load testing to check the application's performance under heavy workloads.

Test Completion Criteria:

Evaluate whether all test cases have been executed and verified.

Review the test results and ensure that the application meets the predefined acceptance criteria.

Test Summary Report:

Prepare a test summary report that includes the test execution results, defects identified, and overall assessment of the application's functionality.

Summarize the testing effort and any outstanding issues.

User Acceptance Testing (UAT):

If applicable, involve end-users to perform user acceptance testing on the application.

Gather feedback and make necessary adjustments based on user feedback.

Final Review and Sign-Off:

Conduct a final review of the application and test results with stakeholders.

Obtain sign-off from relevant parties to indicate that the application is ready for release.

## Performance Outcome:-

**Response Time:** The response time of the application should be relatively fast for various operations, such as recording expenses, generating reports, and managing categories. Users should not experience significant delays when interacting with the application.

**Throughput:** The application's throughput refers to its ability to handle a certain number of operations or transactions per unit of time. For a console-based application, the throughput might not be a major concern unless there is a need for simultaneous multi-user access.

**Memory Usage:** The memory consumption of the application should be reasonable and should not lead to memory leaks or excessive resource usage. Memory profiling should be performed to identify any potential memory-related issues.

**Scalability:** The application should be able to handle a reasonable amount of data, including a substantial number of expenses and categories, without significant performance degradation. Performance testing with large datasets will help assess its scalability.

**Load Testing and Stress Testing:** Load testing involves simulating multiple users and executing multiple operations concurrently to assess the application's performance under various loads. Stress testing pushes the application to its limits to determine how it performs under extreme conditions. The outcome of these tests will reveal how well the application can handle different levels of demand.

# My learnings:-

My overall experience during the internship:

• Practical Experience: Internships provide hands-on experience in a real-world work environment, allowing you to apply theoretical knowledge from your studies to practical tasks.

• Skill Development: During an internship, I had the opportunity to develop and refine various skills, including technical skills, problem-solving, and time management.

• Project Experience: Working on real projects during an internship gave me a portfolio of work that can showcase to potential employers.

• Problem Solving Skills: Internships expose me to various challenges and problem-solving opportunities, enabling me to develop critical thinking and analytical skills.

• Feedback and Mentorship: Internships often provide feedback and mentorship from experienced professionals, helping me to identify areas for improvement and learn from seasoned experts.

• Confidence Building: Successfully completing an internship boosts your self-confidence and reassures me that I can excel in your chosen field.

• Continuous Learning: Internships often shows the areas where I need to improve or gain more knowledge, motivating me to pursue continuous learning and professional development.

# Future work scope:-

The console-based expense tracker application provides a solid foundation for managing personal expenses. There are several potential areas for future work and enhancements to make the application even more useful and user-friendly. Here are some future work scopes for the project:

**User Authentication and Security**: Implement user authentication to allow multiple users to have their individual expense data. Add password protection to secure user data.

**Data Persistence and Database Integration:** Currently, the application stores data in memory, which means the data is lost when the application is closed. Integrate the application with a database for persistent storage of expense records and categories.

**User Interface Improvement:** Enhance the user interface to make it more intuitive and visually appealing. Consider using a library or framework to build a graphical user interface (GUI) for better user experience.

**Data Visualization and Analytics:** Provide graphical charts and graphs to visualize expense trends and spending patterns over time. Add analytics features to analyze user expenses and provide insights.

**Expense Filtering and Sorting**: Implement filtering and sorting options for expenses, allowing users to view expenses based on specific criteria, such as date range or category.

**Backup and Restore:** Add functionality to allow users to back up their expense data and restore it when needed.

**Expense Reminders and Notifications**: Incorporate reminders or notifications to alert users about upcoming bill payments or expense due dates.

**Export and Import**: Allow users to export their expense data to a file (e.g., CSV) for backup or import data from external sources.

**Currency Conversion:** If users deal with multiple currencies, provide an option to convert expenses into a base currency for easier tracking.

**Expense Splitting and Sharing:** Enable users to split expenses among multiple participants and share expense reports with others.