[Your College Logo]

# Industrial Internship Report on

# “Console-based expense tracker application”

## Prepared by

## Mahima Anade


| *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).<br><br>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.<br><br>My project was Console-based expense tracker application<br><br>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

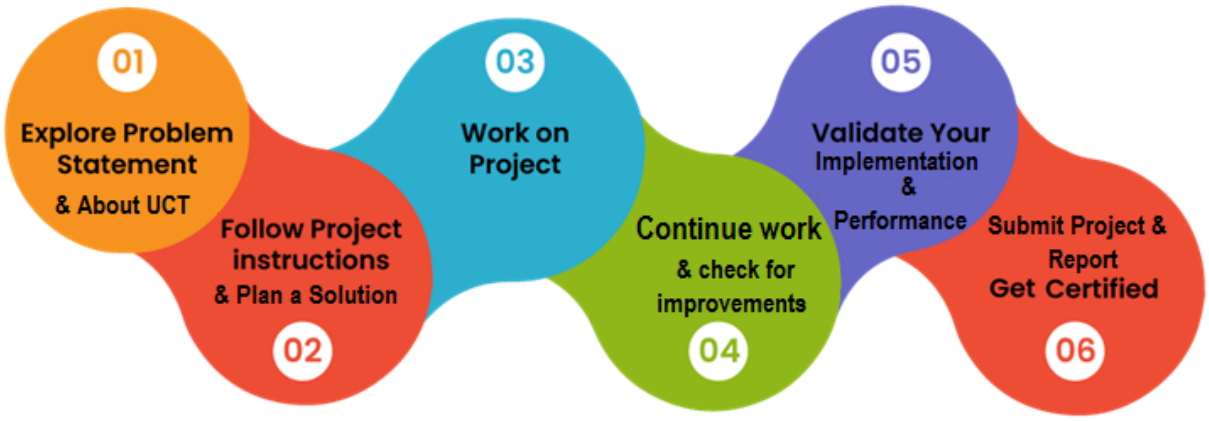
Summary of the whole 6 weeks' work.

About need of relevant Internship in career development.

Brief about Your project/problem statement.

Opportunity given by USC/UCT.

How Program was planned




1.Structured Learning Paths: Upskill Campus likely offers structured learning paths or courses designed to help individuals enhance their skills in specific areas such as programming, data science, web development, etc. Engaging with these courses can provide you with a clear roadmap for learning and skill development.

2.Quality Content: Platforms like Upskill Campus often curate or create high-quality educational content, including tutorials, videos, quizzes, and projects. Engaging with such content can deepen your understanding of key concepts and techniques relevant to your field of interest.

3.Continous Learning: Learning is a lifelong journey, and platforms like UpSkill Campus provide opportunities for continuous learning and skill development. Whether you're just starting out in your career or looking to advance to the next level, there are likely courses and resources available to help you achieve your goals.

# 2 Introduction

## 2.1 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.

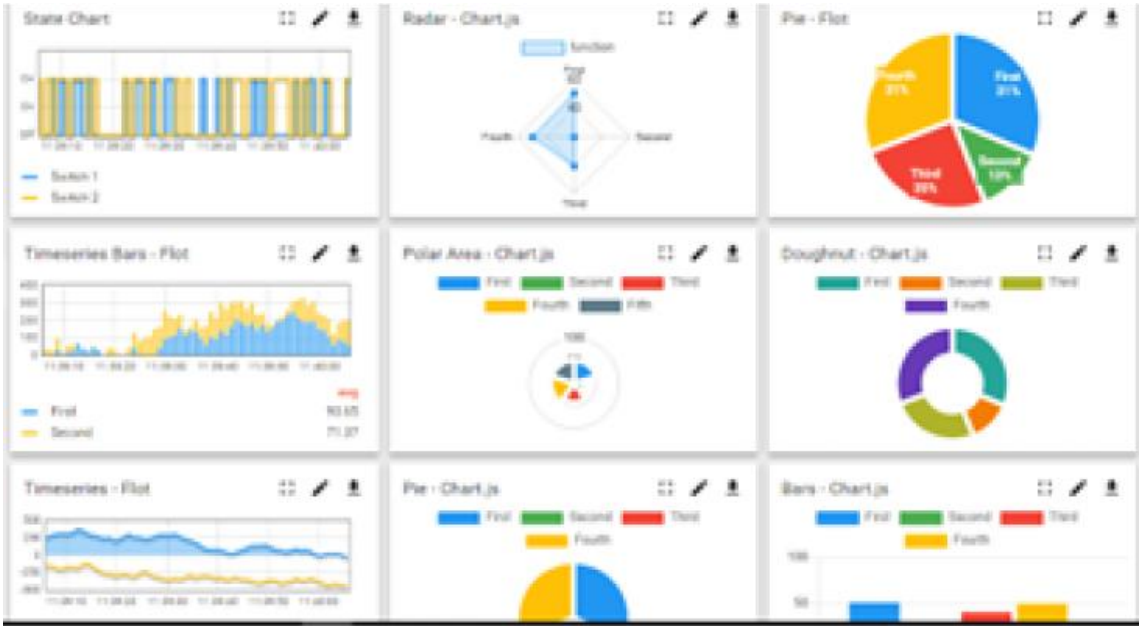

## i. UCT IoT Platform (UCT Insight)

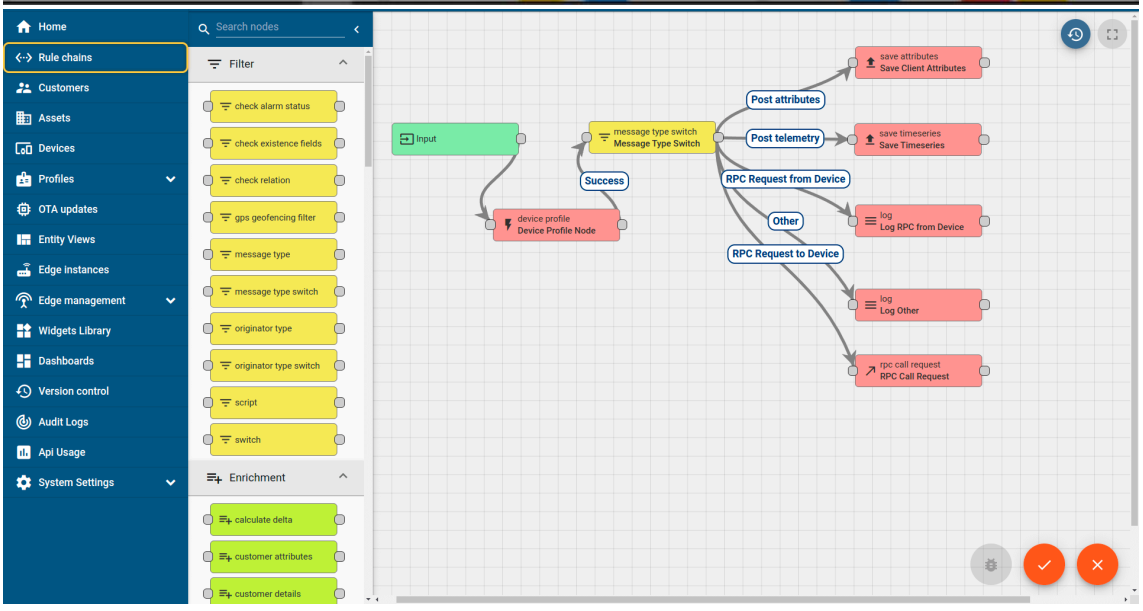
**UCT Insight** is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable "insight" for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

- 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



## ii. Smart Factory Platform ( FACTORY WATCH )

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.

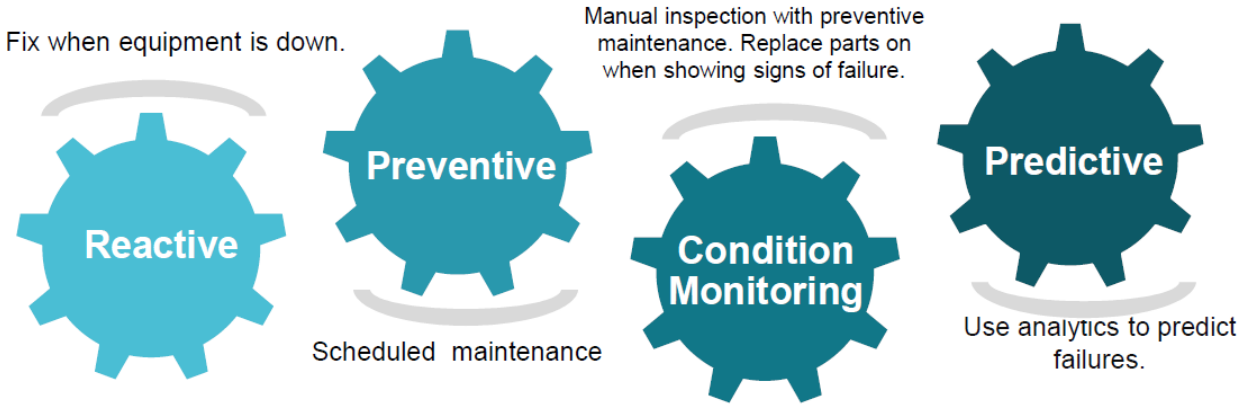

| Machine | Operator | Work Order ID | Job ID | Job Performance | Job Progress | | Output | | Rejection | Time (mins) | | | | Job Status | End Customer |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | Start Time | End Time | Planned | Actual | | Setup | Pred | Downtime | Idle | | |
| CNC_S7_81 | Operator 1 | WO0405200001 | 4168 | 58% | 10:30 AM | | 55 | 41 | 0 | 80 | 215 | 0 | 45 | In Progress | i |
| CNC_S7_81 | Operator 1 | WO0405200001 | 4168 | 58% | 10:30 AM | | 55 | 41 | 0 | 80 | 215 | 0 | 45 | In Progress | i |

### iii. LoRaWAN 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.

### iv. Predictive Maintenance

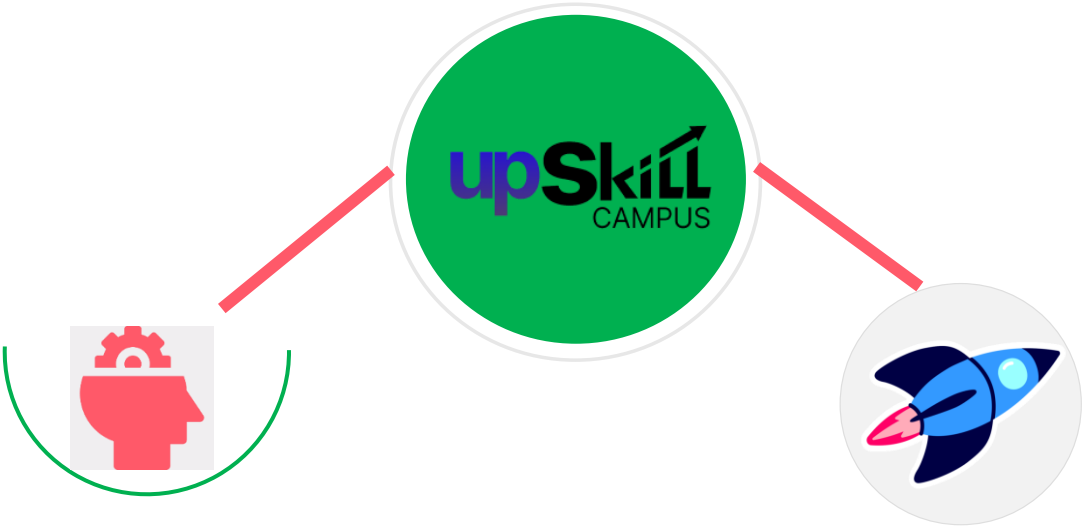
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.




## 2.2 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.

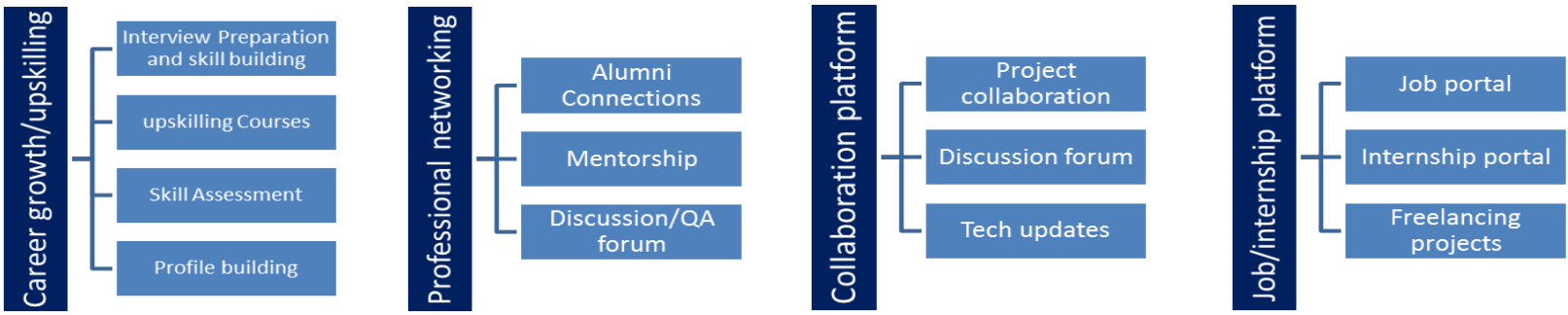
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



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

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

https://www.upskillcampus.com/



## 2.3 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.

## 2.4 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.

## 2.5 Reference

**1)Java Documentation:** The official Java documentation is an invaluable resource for learning about Java syntax, standard libraries, and best practices. You can find it at docs.oracle.com.

**2)Java Tutorial:** Websites like javatpoint and TutorialsPoint offer comprehensive tutorials on Java programming, including topics like file handling, data structures, and GUI programming.

**3)Books:** There are many excellent books on Java programming that cover topics relevant to building applications, including expense trackers. Some popular ones include "Head First Java" by Kathy Sierra and Bert Bates, "Effective Java" by Joshua Bloch, and "Java: A Beginner's Guide" by Herbert Schildt.

## 2.6 Glossary

| Terms | Acronym |
|---|---|
| Expense tracker | A tool or application used to record, manage, and analyze personal or business expenses. |
| CRUD | Create, Read, Update, Delete. The four basic functions of persistent storage, often used in database applications. |
| GUI | Graphical User Interface. A visual way for users to interact with software through graphical icons and visual indicators. |
| CUI | Command-Line Interface. A text-based interface for interacting with software by entering commands into a terminal or console. |
| Validation | The process of ensuring that inputs meet certain criteria or constraints, such as date formats, amount ranges, etc. |

# 3 Problem Statement

In the assigned problem statement :

Create a console-based expense tracker application that allows user to manage their personal expenses. The application should provide functionalities to record and track expenses, view spending summaries, and manage categories for better organization.

**Features:**

1. **Expense Recording:** Allow users to enter details of their expenses, including the date, amount, category, and description.
2. **Expense Categories:** Enable users to create and manage expense categories. Users can assign expenses to specific categories to classify their spending.
3. **Expense Tracking:** Implement features to track and display the total expenses for a specified time period or by category.
4. **Expense Filtering:** Provide options for users to filter and view expenses based on specific criteria such as date range, category, or amount.
5. **Expense Modification and Deletion:** Allow users to modify or delete recorded expenses if needed. Users should be able to edit expense details such as date, amount, category, or description.
6. **Data Persistence:** Ensure that the recorded expenses and category information are stored persistently so that users can access and manage their expenses even after restarting the application.
7. **Reports Generation:** Implement the ability to generate reports that summarize expenses, such as monthly expense reports or category-wise expense reports.
8. **User-Friendly Interface:** Design a user-friendly console interface that provides clear instructions, prompts for user inputs, and formatted displays for expense summaries, category management, and reports.
9. **Error Handling:** Implement error handling mechanisms to handle invalid inputs, errors, and exceptions gracefully, providing informative error messages when necessary.

This project provides a basic foundation for an expense tracking application, allowing users to record, manage, and analyze their expenses through a console-based interface. You can enhance and expand upon these features based on your project requirements and desired complexity.

# 4 Existing and Proposed solution

**Existing solution:**

The existing solution might involve manual tracking of expenses using methods like writing down expenses in a notebook or using spreadsheet software like Excel.

Users have to manually input each expense along with its details such as date, amount, category, and description.

Data persistence might rely on manual backup methods like saving files or sheets on a computer.

**Proposed Solution:**

The proposed solution is a console-based expense tracker application implemented in Java.

It provides a structured system for recording, managing, and analysing expenses.

Users can easily record expenses through a user-friendly console interface, with validation for inputs like date formats and amount formats.

Features like expense categorization, tracking, filtering, modification, deletion, and reporting are built-in, providing users with comprehensive tools for expense management.

## 3.1 Code submission (Github link):

https://github.com/MahimaAnade/upskillcampus/blob/main/Console-basedexpensetrackerapplication.java

## 3.2 Report submission (Github link):

# 4 Proposed Design/ Model

For the proposed design/model of the expense tracker application, we can consider the following components:

**1.Expense class:** This class represents an individual expense and contains attributes such as date, amount, category, and description. It also includes methods for getting and setting these attributes.

**2.ExpenseTrackerApp Class:** This class serves as the main entry point for the application. It contains the main method and handles user interactions through the console interface.

**3.ExpenseManager Class:** This class manages the recording, modification, and deletion of expenses. It maintains a list of expenses and provides methods to perform operations on them.

**4.CategoryManager Class:** This class handles the management of expense categories. It allows users to create, modify, and delete categories, and assigns expenses to specific categories.

# 6 Performance Test

For performance testing of the expense tracker application, you can focus on several key aspects to ensure that it can handle various loads and scenarios efficiently. Here are some performance tests you can conduct:

1.Recording Expenses: Test the application's ability to handle a large number of concurrent expense recording operations. Measure the time taken to record expenses individually and in batches.

2.Retrive Expenses: Measure the time taken to retrieve expenses by category, date range, or other criteria. Test the application's performance when retrieving a large volume of data.

3.Modifying Expenses: Test the performance of modifying expense records, including updating dates, amounts, categories, and descriptions.

4.Deleting Expenses: Measure the time taken to delete individual expenses and bulk deletion of expenses. Evaluate the application's performance when deleting large numbers of records.

5.Category Expenses: Test the application's performance when creating, modifying, and deleting expense categories. Measure the time taken to update category assignments for expenses.

6.Filtering Expenses: Evaluate the performance of the expense filtering feature by testing various filtering criteria such as date range, category, and amount. Measure the time taken to apply filters to a large dataset.

7.Report Generation: Test the performance of generating different types of reports, such as monthly expense reports or category-wise expense reports. Measure the time taken to generate reports for varying data sizes.

8.Concurrency and Scalability: Assess the application's ability to handle concurrent user requests and scale to accommodate increased loads. Test performance under different levels of concurrent users and data volume.

9.Data Persistence: Evaluate the performance of data persistence operations, such as saving and loading expense records and category information from a database or file system. Measure the time taken for data storage and retrieval operations.

10.Error Handling: Assess how the application performs under error conditions, such as handling invalid inputs, errors, and exceptions. Measure the response time and resource utilization during error scenarios.

# 7 My learnings

Reflecting on this project, there are several valuable lessons you can carry forward for future endeavors:

1.Modular Design: Emphasize modular design principles to break down complex systems into smaller, manageable components. This approach enhances flexibility, maintainability, and scalability.

2.User-Centric Development: Prioritize user needs and usability throughout the development process. Design intuitive user interfaces and focus on delivering features that provide value and enhance user experience.

3.Testing and Quality Assurance: Invest time in comprehensive testing and quality assurance to ensure the reliability, performance, and correctness of your software. Implement unit tests, integration tests, and end-to-end tests to catch bugs early and validate functionality.

4.Documentation and Communication: Maintain clear and concise documentation to facilitate understanding, collaboration, and future maintenance of the project. Effective communication with stakeholders and team members is also essential for project success.

5.Project Management Skills: Develop project management skills to effectively plan, organize, and execute software projects. Prioritize tasks, manage timelines and resources, and adapt to changing requirements and constraints.