**Industrial Internship Report on**

**”Expense Tracker”**

**Prepared by**

**Abhinav Kumar**

|  |
| --- |
| *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.  My project was an expense tracker app to keep track of my expenses.  This internship gave me a very good opportunity to get exposure to Industrial problems and design/implement solution for that. I had an overall great experience in this internship. |

**TABLE OF CONTENTS**

[1 Preface 3](#_Toc139702806)

[2 Introduction 4](#_Toc139702807)

[2.1 About UniConverge Technologies Pvt Ltd 4](#_Toc139702808)

[2.2 About upskill Campus 8](#_Toc139702809)

[2.3 Objective 10](#_Toc139702810)

[2.4 Reference 10](#_Toc139702811)

[2.5 Glossary 10](#_Toc139702812)

[3 Problem Statement 11](#_Toc139702813)

[4 Existing and Proposed solution 12](#_Toc139702814)

[5 Proposed Design/ Model 1](#_Toc139702815)4

[6 Performance Test 1](#_Toc139702819)5

[6.1 Test Plan/ Test Cases 1](#_Toc139702820)5

[6.2 Test Procedure 1](#_Toc139702821)6

[6.3 Performance Outcome 1](#_Toc139702822)6

[7 My learnings 1](#_Toc139702823)7

[8 Future work scope 1](#_Toc139702824)8

# Preface

Summary of the whole 6 weeks’ work.

An internship is a crucial component in today’s career development because of the skills needed. It provides essential knowledge and experience of industry and acts as a transitioning bridge between education and career.

In this internship, I got introduced to full stack full stack development process and created a simple expense tracker app using MERN stack.

Special thanks to USC/UCT for providing me this internship opportunity along with the material and doubt sessions.

How Program was planned:



In this internship, I got introduced to MERN stack and learn how different tech work together to serve a functioning website. It was overall a great experience.

Thanks to all mentors from the company USC/UCT and my friends, who have helped me directly indirectly.

There is no learning unless we are able to apply what we learnt, so keep applying the learning to real world as you learn. This also helps broaden he understanding.

# 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 technology and providing solution in Agri-tech, 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]

[2]

[3]

## Glossary

|  |  |
| --- | --- |
| Terms | Acronym |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Problem Statement

In the assigned problem statement

Managing personal finances is a common challenge for individuals, especially when expenses occur frequently and across multiple categories. Many people struggle to maintain accurate records of their daily spending, which often leads to poor financial planning, overspending, or difficulty in identifying saving opportunities. Traditional methods such as manual note-taking or spreadsheets are time-consuming, prone to errors, and inconvenient for regular use.

While there are existing applications for expense tracking, many are either too complex, require paid subscriptions, or do not provide a user-friendly interface.

To address this issue, develop a full-stack expense tracking app. This app should be simple, easy to use and lightweight so it can be quick, load fast with quick response, and free.

# Existing and Proposed solution

**Existing Solutions and Their Limitations**

Several expense tracking applications are already available in the market, such as Mint, YNAB (You Need a Budget), and Walnut. These tools provide features like budget creation, categorization of expenses, and financial insights. However, they have the following limitations:

* **Complexity:** Many existing apps have too many advanced features that the user simply does not want.
* **Cost:** Several applications require paid subscriptions to unlock essential features, which may not be affordable for all users.
* **Customization Issues:** Most applications follow a fixed structure and do not allow users to customize categories or features according to their needs.
* **Platform Dependency:** Some solutions are available only on mobile or require native installation, limiting accessibility across devices.
* **Privacy Concerns:** A number of existing apps demand sensitive data access, such as bank account or credit card details, raising security concerns.
* **Ads:** Most free applications contain ads. which can be annoying at times.

**Proposed Solution**

To address these limitations, I developed a **web-based Expense Tracker application** using the **MERN (MongoDB, Express.js, React.js, Node.js) stack**. The solution focuses on simplicity, usability, and accessibility while ensuring data security.

The key features of the proposed system include:

* Easy expense or income entry.
* Categorization of expenses for better organization.
* Real-time updates and fast response through a MERN stack architecture.
* Secure data storage in MongoDB.
* Cross-platform availability via a responsive web interface.

**Value**

The value addition of this project lies in providing a **lightweight, customizable, and user-friendly** expense management system that caters to individuals who want a straightforward tool without unnecessary complexity. The application adds value in the following ways:

* **Simplicity:** Focuses on essential feature only that is keep log of expenses only.
* **Focused:** Having simple expense log and entry only removes unwanted distraction.
* **Cost-Effective:** A free solution that eliminates need of subscriptions for basic service.
* **Accessibility:** Works on any device with internet access through a web browser.
* **Data Privacy:** Ensures that users have full control over their data without linking sensitive bank details.

## Code submission (Github link) :

https://github.com/Unknown56860/upskillCampus/tree/main/expense\_tracker

## Report submission (Github link) :

https://github.com/Unknown56860/upskillCampus/blob/main/report.docx

# Proposed Design/ Model

The application is built on the **MERN stack (MongoDB, Express.js, React.js, Node.js)**, where each layer plays role in the design. The solution can be divided into the following stages:

**1. User Interaction Layer (Frontend)**

Users can interact with the frontend, developed using **react**.

**2. Request Handling Layer (Backend API)**

* When a user performs an action, the request is sent to the backend, developed using **express** and **node**.
* This layer creates routing of requests and validation of input data.
* It acts as a secure middleware between the frontend and the database.

**3. Data Storage Layer (Database)**

* Expense details are stored in **MongoDB**, which provides flexibility in handling unstructured or semi-structured data.
* Data is saved in an online data cluster.
* Data is preserved even if the server changes as data is stored in cloud.

**4. Data Processing**

User enter their expense and title in one input field, it is separated in the backend.

# Performance Test

The performance testing of this Expense Tracker Application was carried out to evaluate its efficiency, reliability, and suitability for real-world usage. Performance testing is essential to ensure that the system can handle multiple users, operate within memory and time constraints, and provide accurate and timely responses.

**Constraints Considered**

1. **Response Time:** The system should respond quickly to user actions (e.g., adding an expense should reflect instantly).
2. **Scalability:** The database should handle an increasing number of expense records without performance degradation.
3. **Memory Utilization:** The application should not consume excessive memory on the client or server side.
4. **Data Accuracy:** Expense records must remain in order and correctly stored/retrieved.
5. **Durability:** Data should remain intact even after server restarts.

These constraints were addressed in the design through:

* Use of **React.js** for fast rendering and real-time updates without page reloads.
* **MongoDB** for efficient handling of large, unstructured datasets.
* Implementation of **REST APIs** in Node.js/Express.js for lightweight communication.

## Test Plan/ Test Cases

The following test cases were executed to validate the constraints:

|  | **Description** |  |  | **Status** |
| --- | --- | --- | --- | --- |
|  | Add a new expense entry |  |  | Pass |
|  | Update an existing expense |  |  | Pass |
|  | Delete an expense |  |  | Pass |
|  | Multiple users adding expenses simultaneously (5–10 users) |  |  | Pass |
|  | System restart and data persistence |  |  | Pass |

## Test Procedure

**Environment Setup:**

* Local machine and test server with Node.js, MongoDB, and React.js environment.

**Execution Steps:**

* Insert expense records into the database and measure retrieval time.
* Monitor server performance using logging tools to measure memory usage.
* Perform CRUD operations (Create, Read, Update, Delete) repeatedly under normal and stress conditions.
* Restart the server and confirm persistence of user data.

**Monitoring Tools Used:**

* **MongoDB Compass** for database monitoring.
* **Browser DevTools (Network Tab)** for measuring response times.

## Performance Outcome

* **Response Time:** Average response time for operations was **0.8 seconds,** meeting the constraint.
* **Scalability:** The application successfully handled multiple **expense records** without noticeable delays.
* **Memory Usage:** The server maintained stable memory usage, averaging **150–200 MB**.
* **Accuracy:** No data loss or inconsistency was found.
* **Durability:** Data was persistent even after server restarts, thus ensuring reliability.

# My learnings

During this internship, I gained hands-on experience in developing a full-stack web application using the MERN stack. I enhanced my skills in building responsive user interfaces with React, designing and integrating RESTful APIs with Node.js and Express.js, and managing data efficiently with MongoDB. I also learned the importance of system design, performance testing, and ensuring scalability and data security in real-world applications.

Beyond technical growth, this project improved my problem-solving, debugging, and project documentation skills. It gave me exposure to industry-level practices such as structured development workflows and testing strategies. These learnings have strengthened my confidence as a developer and will serve as a solid foundation for my career, particularly in pursuing opportunities in web development and full-stack engineering.

# Future work scope

While the current version of this app full fills its purpose of keeping record of expenses, there are several areas for enhancement in the future:

* **User Authentication & Security:** Implement authentication methods (OAuth, JWT with refresh tokens, multi-factor authentication) for data security.
* **Budgeting & Alerts:** Introduce budget-setting features where users can set monthly spending limits and receive alerts when approaching or exceeding them.
* **Advanced Analytics:** Provide AI-driven insights into spending patterns and personalized saving suggestions.
* **Mobile Application:** Extend the system by developing a dedicated mobile app (React Native/Flutter) for on-the-go accessibility.
* **Multi-User/Family Accounts:** Allow shared expense tracking for households or groups with role-based access control.
* **Cloud Deployment & Scalability:** Host the application on cloud platforms like AWS or Azure with load balancing to support larger user bases.