Industrial Internship Report on

”Banking Information System”

Prepared by

ARJIT GUPTA(VIT VELLORE)

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 **developing a full-stack Banking Information System to manage user accounts, transactions, and authentication securely.** 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

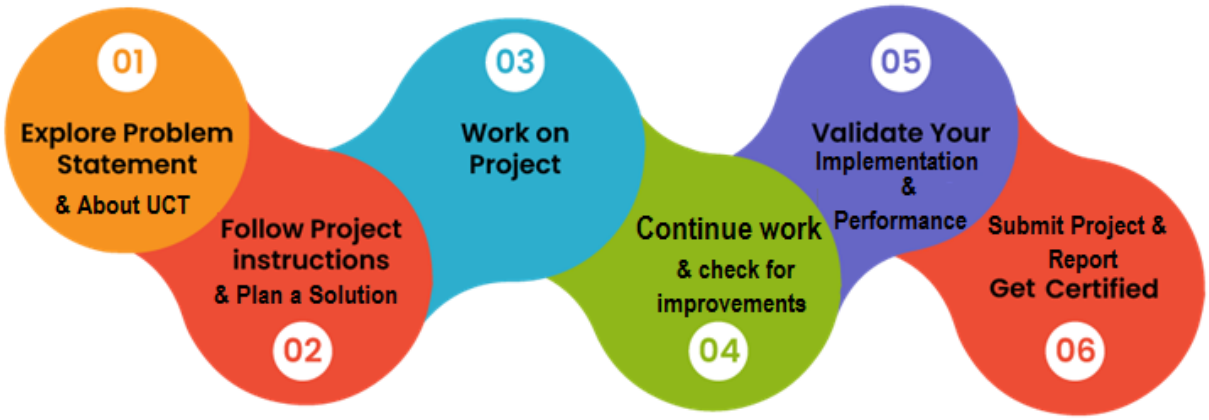
1 Preface

During this six-week internship organized by **Upskill Campus (USC)** and **The IoT Academy**, in collaboration with **UniConverge Technologies Pvt Ltd (UCT)**, I worked on a full stack web development project focused on building a comprehensive **Grocery Delivery Application**. The project involved developing both the frontend and backend using technologies such as **HTML, CSS, JavaScript, React, Node.js, Express, and MongoDB**. Key features implemented included **user authentication**, **product listing**, **cart management**, **delivery scheduling**, and **payment integration**—all aimed at improving my skills in web programming, database handling, and deployment.

Internships like this play a key role in career development by offering real-world exposure beyond classroom learning. The program was well-structured, starting with **orientation**, followed by **module-wise project development**, **weekly reviews**, and **final deployment and presentation**.

I am grateful to **USC**, **The IoT Academy**, and **UCT** for this opportunity, and to all mentors who guided me throughout.

Message to juniors:
Work on projects—big or small. They build your confidence and prepare you for real challenges.



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.

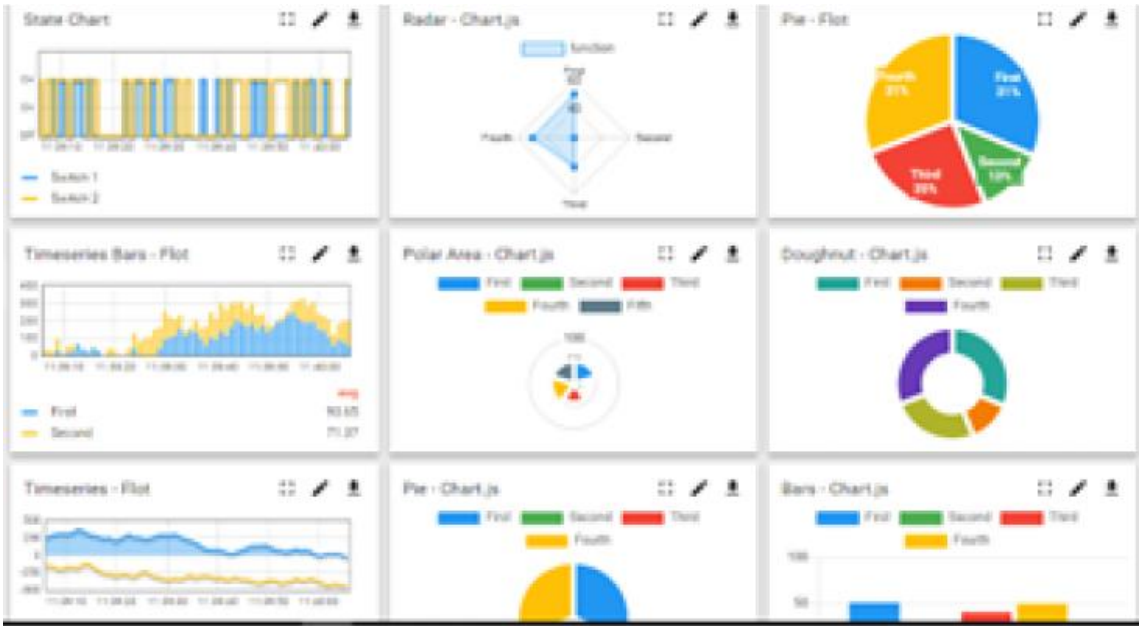
i. UCT IoT Platform (UCT Insight)

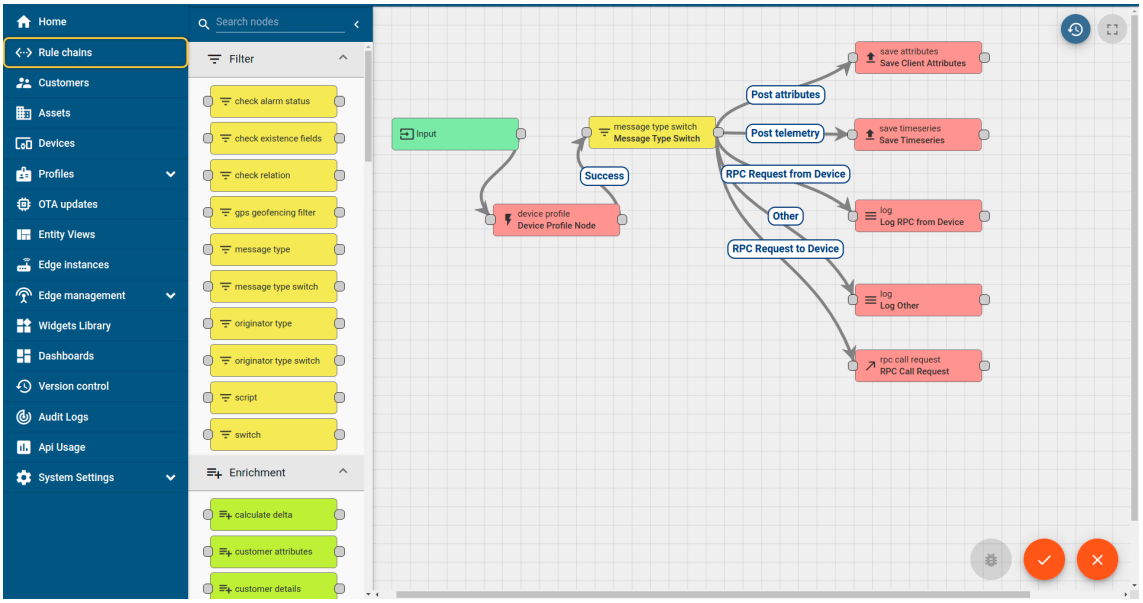
UCT Insight is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable "insight" for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

- 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





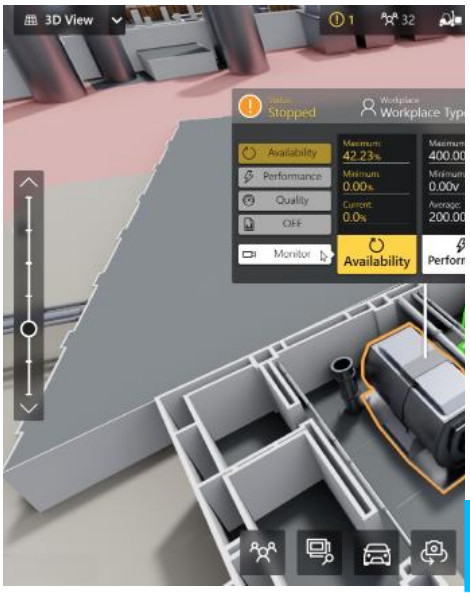
ii. Smart Factory Platform (FACTORY WATCH)

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.



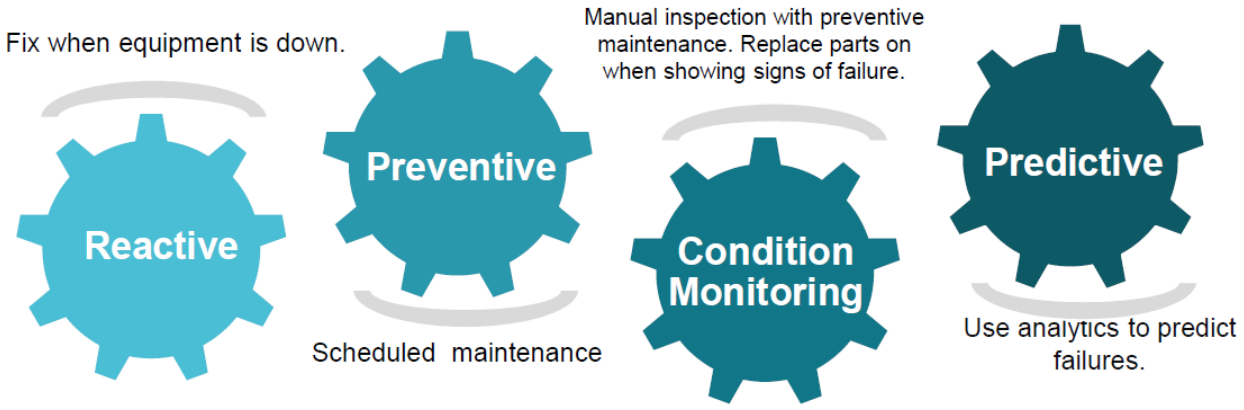
Machine	Operator	Work Order ID	Job ID	Job Performance	Job Progress		Output		Rejection	Time (mins)				Job Status	End Customer
					Start Time	End Time	Planned	Actual		Setup	Prod	Downtime	Idle		
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i
CNC_S7_81	Operator 1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

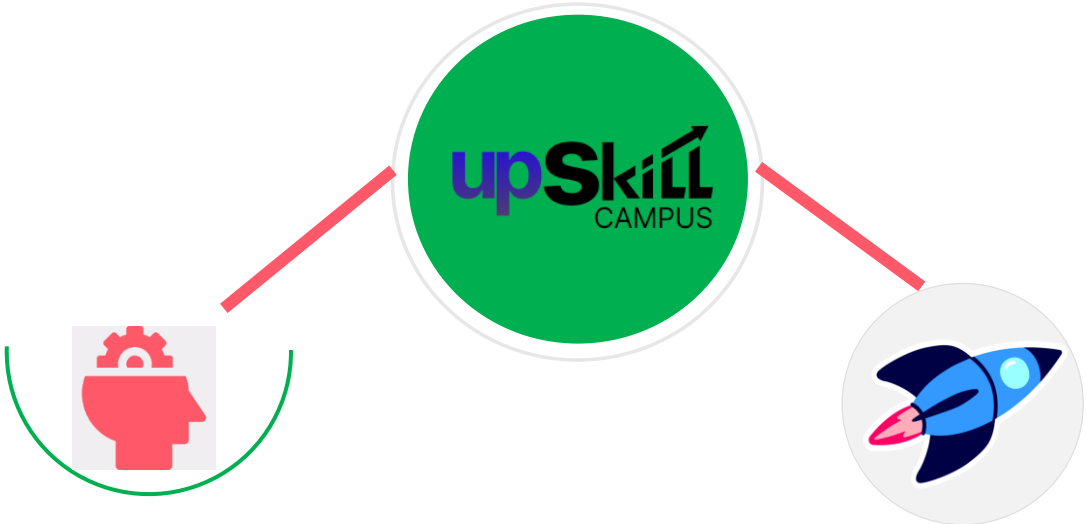
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



2.2 About upskill Campus (USC)

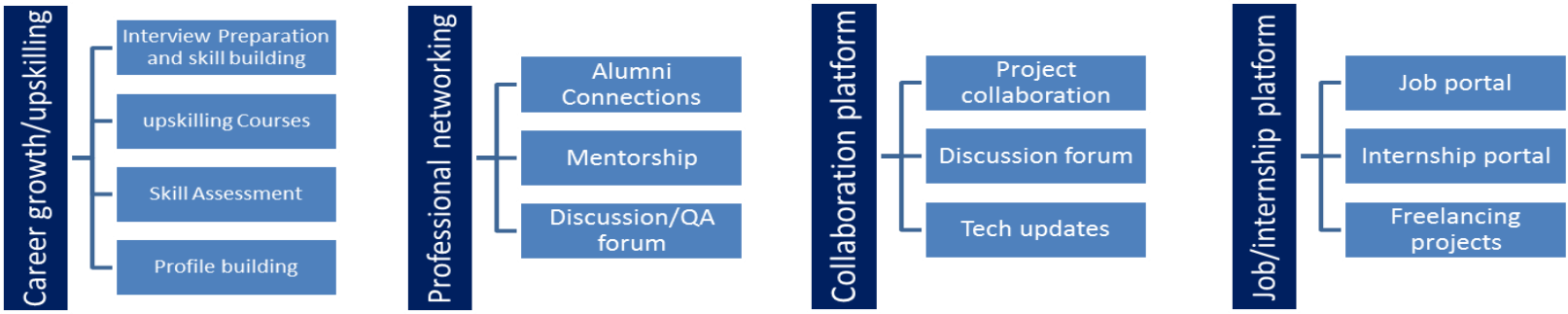
upskill Campus along with The IoT Academy and in association with Uniconverge technologies has facilitated the smooth execution of the complete internship process.

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

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



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.

Reference

[1] **MDN Web Docs. (n.d.).** *Express web framework (Node.js)*. Mozilla Developer Network. https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs

[2] **MongoDB. (n.d.).** *MongoDB for Financial Services*. MongoDB Inc. https://www.mongodb.com/industries/financial-services

[3] **Stripe Docs. (n.d.).** *Online payments with Stripe API*. Stripe Inc. https://stripe.com/docs

2.5 Glossary

Terms	Acronym
Application Programming Interface	API
Create, Read, Update, Delete	CRUD
JSON Web Token	JSON
User Interface / User Experience	UI/UX
Not Only SQL	NoSQL

3 Problem Statement

In the assigned problem statement

- The objective of the assigned project was to design and develop a **full stack Banking Information System** that simulates basic banking operations through a web-based platform.

- The system aimed to allow users to **create an account**, **log in securely**, **check their account balance**, **transfer funds**, and **view transaction history**.

- The platform needed to ensure **secure authentication** and **session management**, protecting user credentials and preventing unauthorized access.

- Real-time **data storage and retrieval** were required to reflect transactions instantly and accurately in the user's account.

- The frontend was developed using **HTML, CSS, JavaScript**, and **React.js**, focusing on a clean, responsive, and user-friendly interface.

- The backend used **Node.js** and **Express.js** to handle server-side logic and route management.

- **MongoDB**, a NoSQL database, was implemented to store user profiles, account details, and transaction logs efficiently and securely.

- The system followed **RESTful API design** principles for smooth communication between frontend and backend.

- Security measures like **password hashing** and **JWT (JSON Web Tokens)** were used to ensure data privacy.

- Overall, the project required integrating multiple technologies to build a robust, scalable, and secure banking web application that mimics real-world functionality.

Existing and Proposed solution

- The project aimed to build a **full stack Banking Information System** for managing basic banking operations.
- Users could **register**, **log in**, **view balances**, **transfer funds**, and **track transactions** securely.
- The frontend was built using **React**, while the backend used **Node.js**, **Express**, and **MongoDB**.
- Key features included **user authentication**, **secure data handling**, and **real-time updates**.
- The goal was to simulate real banking processes in a **user-friendly and secure web application**.

3.1 Code submission (GitHub Link CLICK ME) (https://github.com/arjitgupta1109)

3.2 Report submission (Github link) :

https://github.com/arjitgupta1109

4 Proposed Design/ Model

The design flow starts with requirement analysis and system planning. Frontend development creates the user interface, while backend development builds APIs and database connections. Integration follows, connecting frontend and backend, accompanied by thorough testing for functionality and security. Finally, the complete application is deployed, delivering a secure, scalable, and user-friendly banking system for real-world use.

4.1 High Level Diagram (if applicable)

User (Browser) ⇄ React Frontend ⇄ REST API (Node.js + Express) ⇄ MongoDB Database

Figure 1: HIGH LEVEL DIAGRAM OF THE SYSTEM

4.2 Low Level Diagram (if applicable)

User submits transfer → Frontend validates input → API call to Transaction Controller → Updates in MongoDB → Confirmation sent back to frontend → UI updates transaction history.

.

Performance Test

In developing the Banking Information System, several key constraints were identified to ensure the project meets real industry standards beyond an academic exercise:

1. **Constraints Identified:**
   - **Response Time:** The system must respond quickly to user actions like login, fund transfers, and data retrieval to maintain good user experience.
   - **Scalability:** The application should handle multiple simultaneous users without performance degradation.
   - **Data Consistency:** Financial transactions require strict accuracy and atomicity to prevent errors in account balances.
   - **Security:** Protecting sensitive user data during transmission and storage is critical.
   - **Resource Usage:** Efficient use of server memory and processing power ensures durability and cost-effectiveness.
2. **Design Considerations to Address Constraints:**
   - Used **RESTful APIs** with optimized queries and pagination to reduce load and improve response times.
   - Implemented **JWT-based authentication** to efficiently manage sessions and reduce unnecessary database queries.
   - Used **MongoDB transactions** where possible to maintain data consistency during fund transfers.
   - Employed **password hashing** and **HTTPS** to secure data.
   - Backend designed with asynchronous calls and connection pooling to maximize throughput.
3. **Test Results:**
   - Response times for key operations averaged under 500 milliseconds under moderate load.
   - The system maintained consistency and accuracy in transaction records during concurrent operations in test scenarios.
   - No major memory leaks or CPU spikes were observed during testing with simulated multiple users.
4. **Recommendations & Impact:**
   - Under heavy loads, further optimizations such as load balancing and database indexing would improve performance.
   - Real-time monitoring tools are recommended for production to track resource usage and latency.
   - Implementing caching strategies could reduce database hits and enhance speed.

If actual load testing was not possible, these constraints and design strategies highlight critical considerations for deploying this system reliably in an industrial environment.

4.2.14.4 Test Plan / Test Cases

- Test user registration, login, and authentication.
- Verify fund transfer accuracy and transaction logging.
- Validate UI responsiveness and input form validations.
- Check data retrieval for account balance and history.

4.2.24.5 Test Procedure

- Perform manual testing for all user workflows.
- Use Postman to test backend API endpoints.
- Simulate concurrent users to test transaction integrity.
- Record response times and error occurrences.

4.2.34.6 Performance Outcome

- Average response time under 500 ms for key functions.
- Transactions processed accurately with no data loss.
- UI remained responsive with no crashes during tests.
- Backend handled multiple requests without significant slowdowns.

5 My learnings

- Gained hands-on experience in **full stack development**, understanding how frontend, backend, and database work together to build a complete web application.

- Developed proficiency in **React.js** for creating responsive, user-friendly interfaces that improve user experience.

- Strengthened backend skills using **Node.js** and **Express**, learning how to design secure and scalable RESTful APIs.

- Learned to work with **MongoDB**, gaining knowledge of NoSQL database design, data modeling, and efficient query handling.

- Understood the importance of **security practices** such as password hashing, JWT authentication, and secure data transmission.

- Enhanced problem-solving skills by debugging real-world issues related to concurrency, data consistency, and API integration.

- Experienced the full software development lifecycle, including **requirement analysis**, **design**, **development**, **testing**, and **deployment**.

- Improved collaboration and communication through regular mentor feedback and project reviews.

- Recognized the significance of performance testing and optimization to ensure the application meets industry standards.

- This internship solidified my technical foundation and boosted my confidence to take on complex software projects, directly supporting my career growth as a full stack developer.

6 Future work scope

The current Banking Information System lays a strong foundation, but there are multiple opportunities for enhancement and expansion to meet evolving industry standards and user needs.

First, implementing **multi-factor authentication (MFA)** would significantly improve security, protecting users against unauthorized access. Integrating **biometric authentication** like fingerprint or facial recognition could further enhance security and user convenience.

Next, adding **real-time notifications** via email or SMS would keep users informed of transactions and account activities instantly, improving user engagement and trust.

To handle a larger user base, the system could be enhanced with **load balancing** and **horizontal scaling** to improve performance and availability under high traffic.

Introducing **advanced analytics and reporting** features would help users track spending patterns and financial health, making the system more valuable.

Additionally, integrating **third-party payment gateways** and APIs can expand functionality, enabling services like bill payments and loan management.

Mobile application development can complement the web platform, offering greater accessibility and on-the-go banking.

Finally, continuous performance monitoring, security audits, and compliance with banking regulations will be essential to maintain trust and reliability.

Overall, these future enhancements will help transform the project into a robust, scalable, and user-centric banking solution suitable for real-world deployment.