Industrial Internship Report on

”FOODIEE”

Prepared by

[Sowndarya Sree]

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 (**My project was the development of a full-stack Food Delivery Application that connects customers with nearby restaurants, allows online ordering, supports real-time delivery tracking, and provides a unified cart and payment system.**)

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.

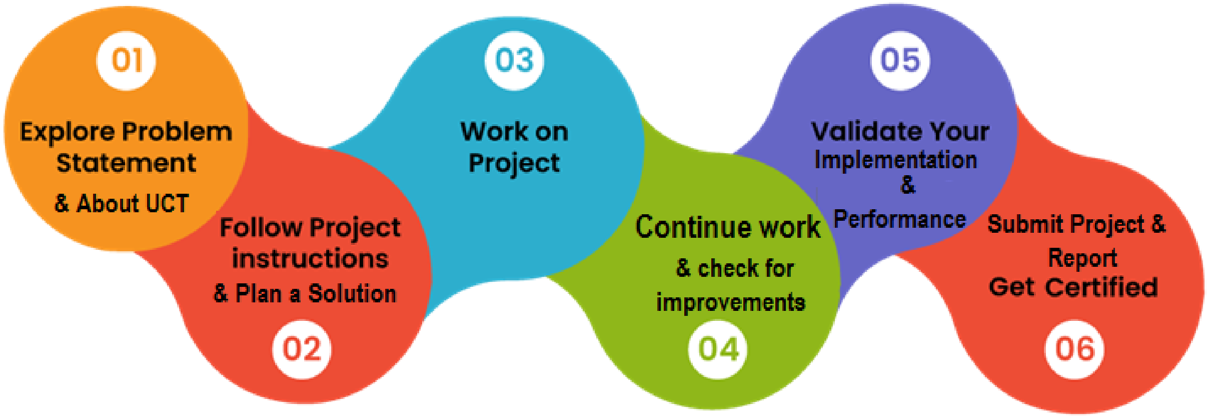
TABLE OF CONTENTS

1 Preface

During these 6 weeks, I worked on understanding my full-stack project, developing backend APIs, designing frontend pages, integrating MongoDB Atlas, and learning Java fundamentals and JDBC.

A relevant internship plays a major role in career development as it provides real industrial exposure and teaches how theoretical concepts are applied practically.

My project was a Food Delivery Application that allows restaurant owners to list food items, and customers to browse, order food, track delivery, and manage their accounts.



USC and UCT provided a structured program that started from basics and gradually moved into hands-on implementation. Each week had planned learning outcomes, tasks, and deliverables.

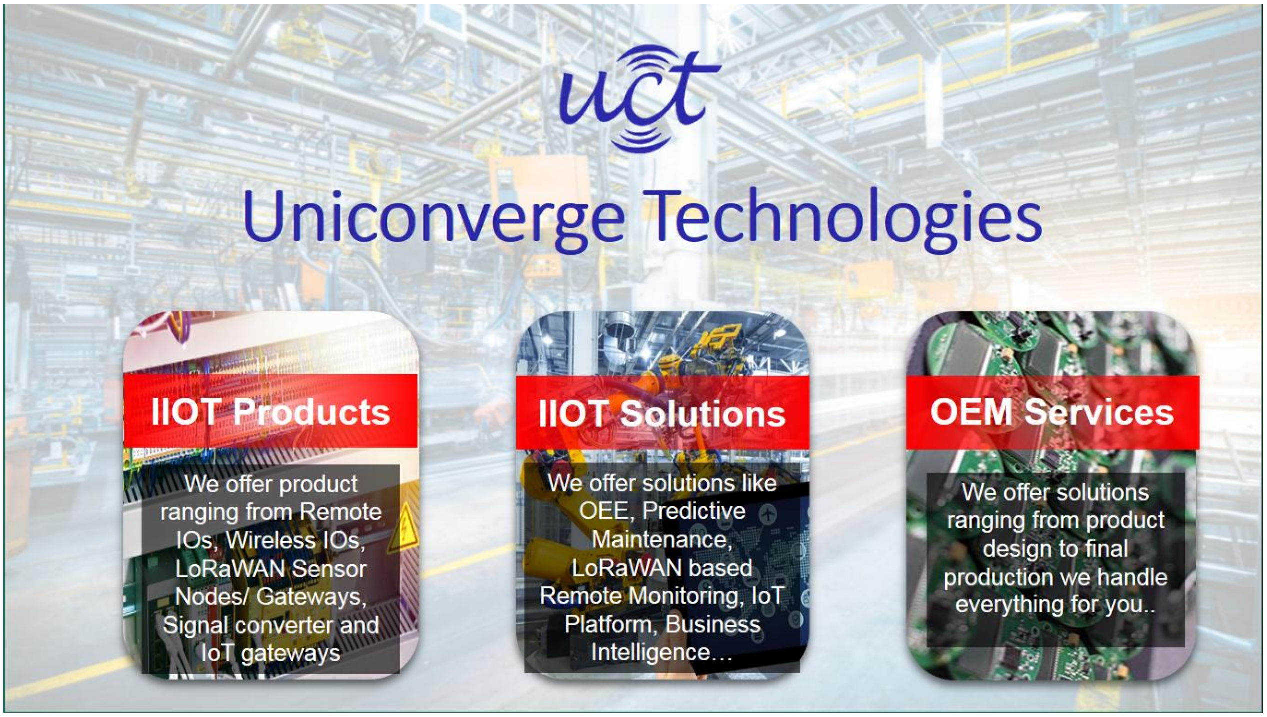
Overall, this internship experience improved my technical confidence, problem-solving ability, and understanding of software development. I thank USC, UCT, mentors, coordinators, and my peers who supported me throughout the journey.

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.



uct

Uniconverge Technologies

IIOT Products

We offer product ranging from Remote IOs, Wireless IOs, LoRaWAN Sensor Nodes/ Gateways, Signal converter and IoT gateways

IIOT Solutions

We offer solutions like OEE, Predictive Maintenance, LoRaWAN based Remote Monitoring, IoT Platform, Business Intelligence…

OEM Services

We offer solutions ranging from product design to final production we handle everything for you..

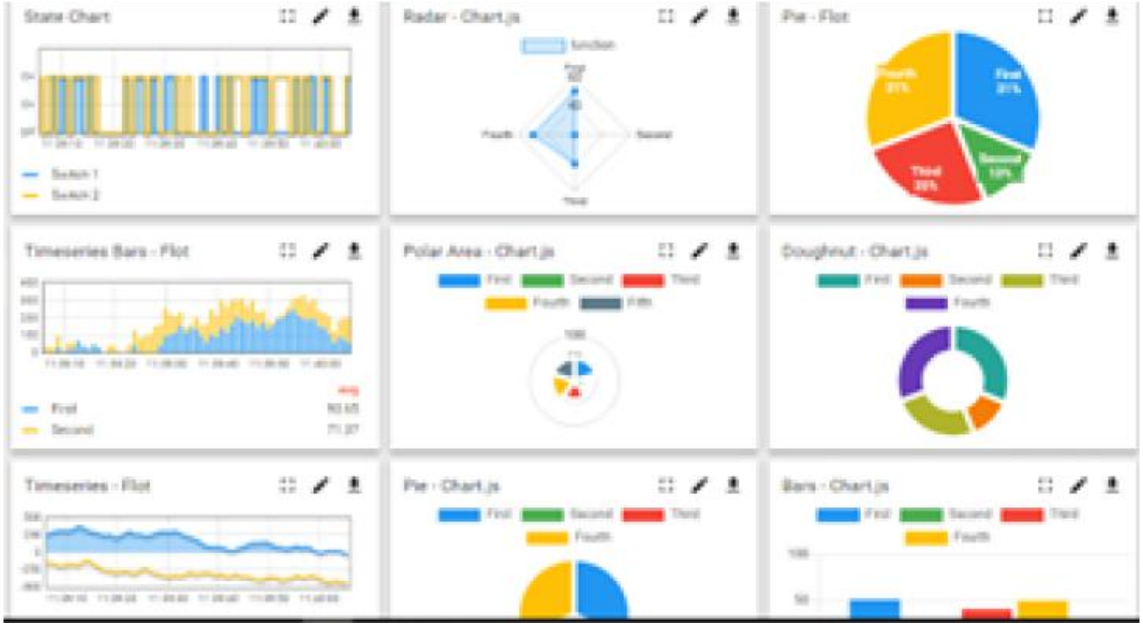
i. UCT IoT Platform (uct Insight)

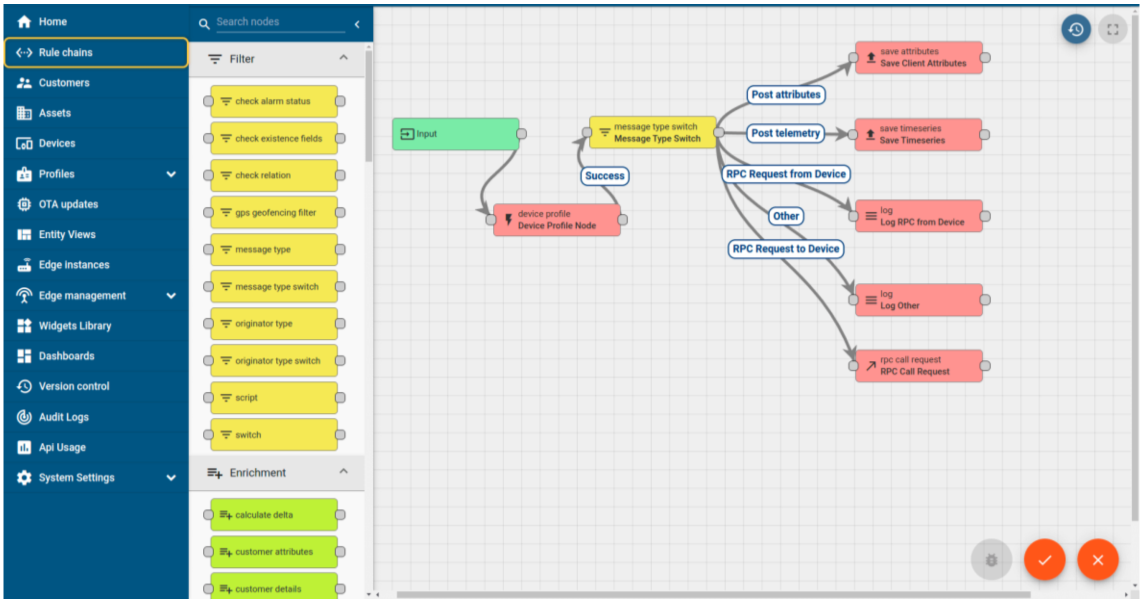
UCT Insight is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable “insight” for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

- 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





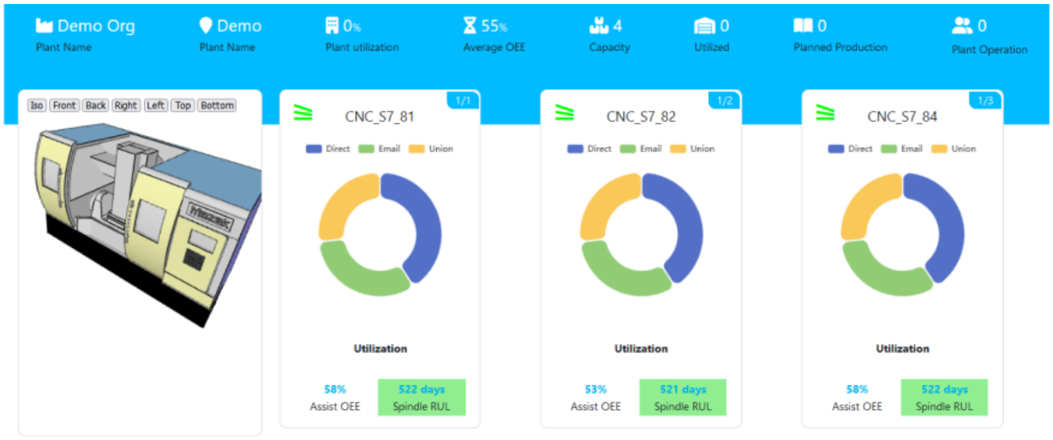
ii. Smart Factory Platform (FACTORY WATCH)

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.



Demo Org
Plant Name

Demo
Plant Name

0%
Plant utilization

55%
Average OEE

4
Capacity

0
Utilized

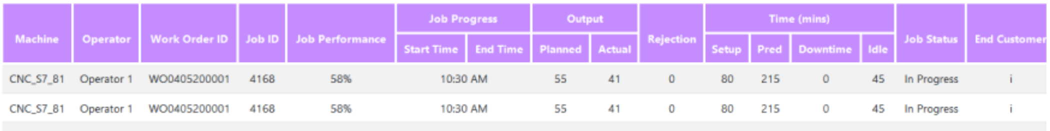
0
Planned Production

0
Plant Operation

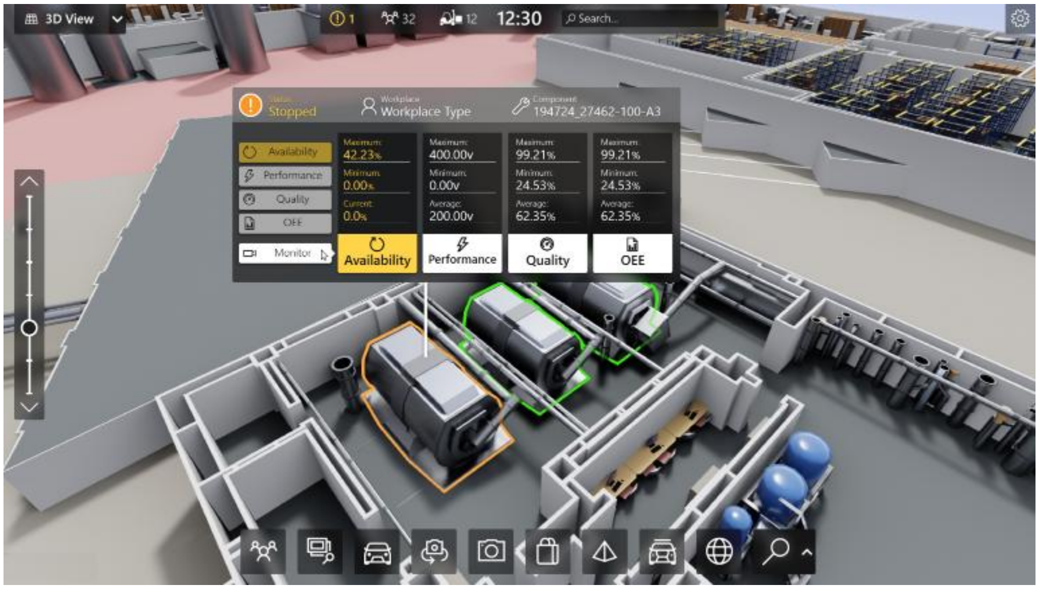
CNC_S7_81 1/1
Direct Email Union
Utilization
58% Assist OEE
522 days Spindle RUL

CNC_S7_82 1/2
Direct Email Union
Utilization
53% Assist OEE
521 days Spindle RUL

CNC_S7_84 1/3
Direct Email Union
Utilization
58% Assist OEE
522 days Spindle RUL



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



Status: Stopped
Workplace: Workplace Type
Component: 194724_27462-100-A3

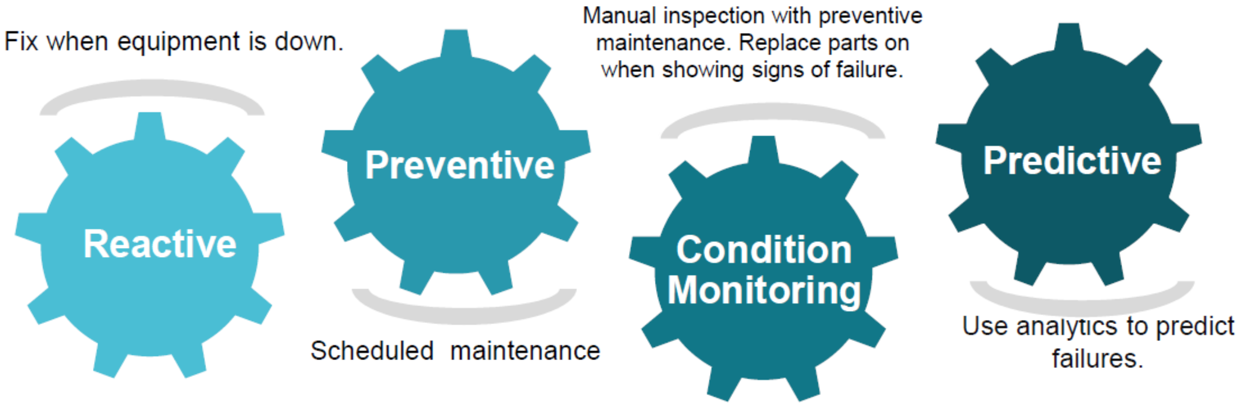
Availability	Performance	Quality	OEE
Maximum: 42.23%	Maximum: 400.00v	Maximum: 99.21%	Maximum: 99.21%
Minimum: 0.00%	Minimum: 0.00v	Minimum: 24.53%	Minimum: 24.53%
Current: 0.0%	Average: 200.00v	Average: 62.35%	Average: 62.35%

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

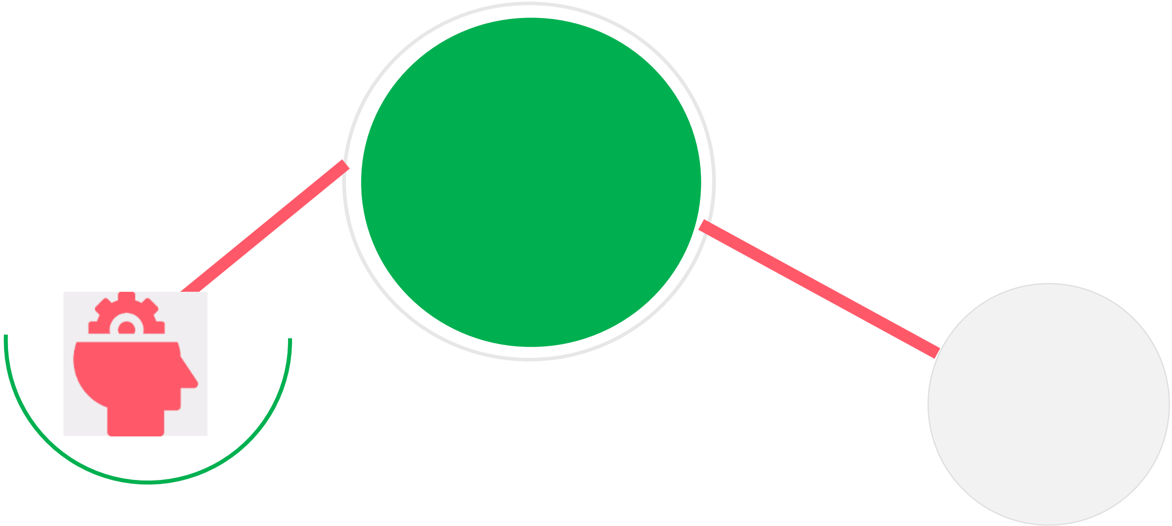
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.


Fix when equipment is down.
Reactive
Preventive
Scheduled maintenance
Manual inspection with preventive maintenance. Replace parts on when showing signs of failure.
Condition Monitoring
Predictive
Use analytics to predict failures.

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.

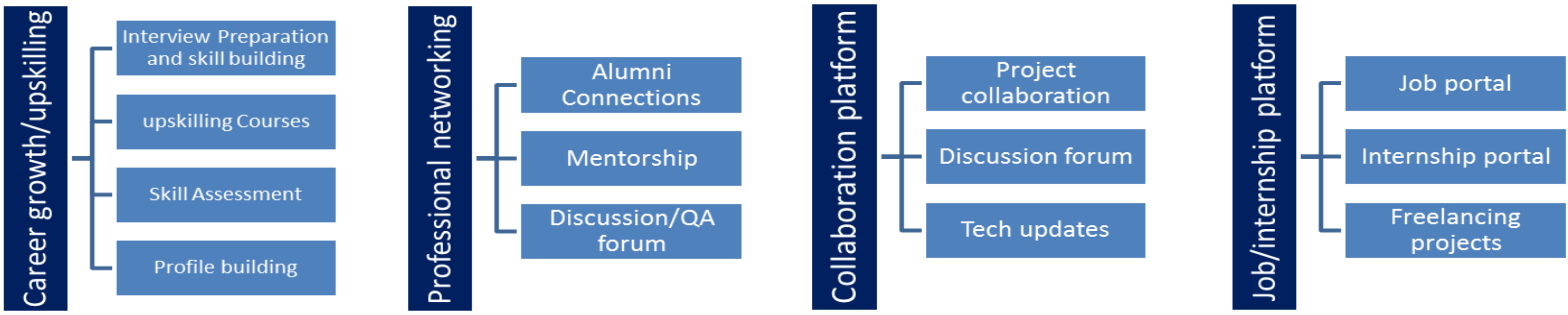
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



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/



Career growth/upskilling
- Interview Preparation and skill building
- upskilling Courses
- Skill Assessment
- Profile building

Professional networking
- Alumni Connections
- Mentorship
- Discussion/QA forum

Collaboration platform
- Project collaboration
- Discussion forum
- Tech updates

Job/internship platform
- Job portal
- Internship portal
- Freelancing projects

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] MongoDB Atlas Documentation – https://www.mongodb.com/docs/

[2] Node.js Official Documentation – https://nodejs.org/en/docs

[3] Express.js Guide – https://expressjs.com/

2.6 Glossary

Terms	Acronym
API	Application Programming Interface
DB	Database
JWT	JSON Web Token (used for authentication)
JSON	JavaScript Object Notation
UI/UX	User Interface/ User Experience

3 Problem Statement

The goal of this internship project is to design and develop a **Food Delivery Application** that connects restaurants and customers on a unified digital platform. The system should enable restaurant owners to register, upload their menu items, manage orders, and track deliveries. Customers should be able to browse nearby restaurants, select food items, add them to the cart, place orders, and track their delivery status in real time.

The application must support user authentication, restaurant management, secure ordering, and delivery workflow. It should also incorporate a cloud database to store user details, restaurant information, menu items, and order records. The system should simulate a real-world food delivery model similar to Zomato or Swiggy but simplified for academic and internship-level implementation.

Explanation

Food delivery apps have become an essential part of everyday life, and this project aims to create a simplified version to understand how such systems work. The application includes three main roles: customer, restaurant, and delivery agent. Customers can sign up, view restaurants, order food, and track deliveries. Restaurants can manage their menu and handle incoming orders. Delivery agents are responsible for picking up and delivering the orders.

Technically, the project uses ReactJS for the frontend, Node.js + Express for the backend, and MongoDB Atlas as the cloud database. The system also uses JWT authentication to ensure secure login and API communication. This project helps in understanding full-stack development, API creation, database handling, and real-world application workflows in a simple and structured way.

4 Existing and Proposed solution

Apps like Swiggy and Zomato already offer food ordering and delivery services, but they have:

- Complex architectures
- Closed-source systems (not available for learning)
- Advanced features that are difficult for beginners
- Heavy infrastructure requirements

So they are not suitable for academic projects or beginner-level development.

My proposed system is a **simple Food Delivery Application** that includes:

- Customer and restaurant login
- Restaurant menu management
- Cart and order placement
- Order acceptance by restaurant
- Driver assignment
- Basic delivery tracking

Value Addition

- Cloud database (MongoDB Atlas)
- Clean API structure
- Simple, modular backend

4.1 Code submission (Github link)

Github link: https://github.com/sowndaryasree/upskillcampus.git

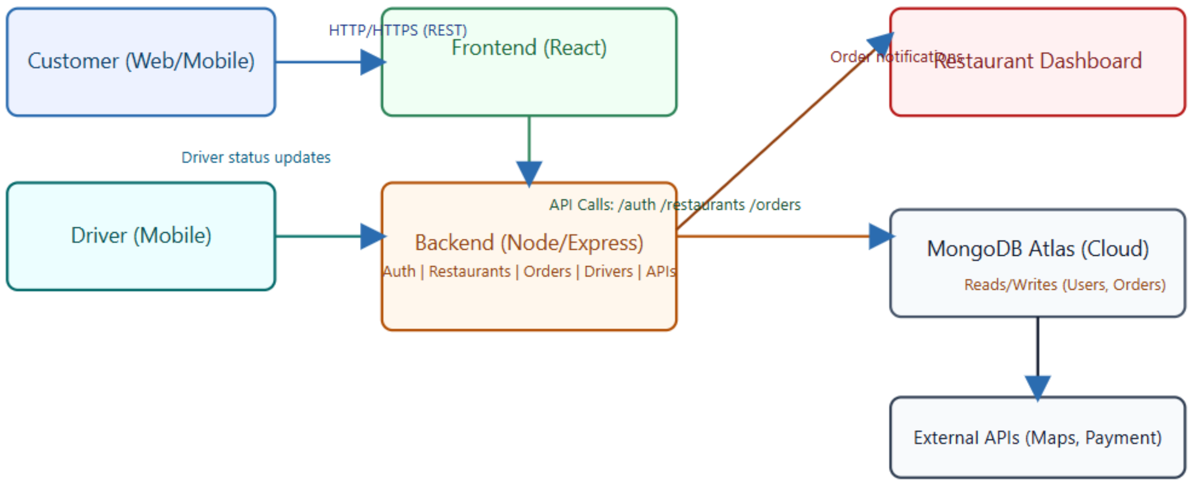
4.2 Report submission (Github link) :

Link: https://github.com/sowndaryasree/upskillcampus.git

5 Proposed Design/ Model

The proposed Food Delivery Application follows a simple and structured design flow.
The system begins with user authentication, moves through restaurant and menu browsing, order creation, restaurant acceptance, and finally delivery and status updates.
Each stage is connected through backend APIs and a cloud database to ensure smooth functioning.

5.1 High Level Diagram



5.2 Interfaces

System Interfaces Used:

- **Frontend → Backend:** REST API calls (JSON)
- **Backend → Database:** MongoDB Atlas driver
- **User Interface:** React components
- **Restaurant Interface:** Menu & order management
- **Driver Interface:** Status update system
- Data Flow (User → Server → DB → Server → User

6 Performance Test

Constraints & impact

- **Scalability / Throughput:** limited by single-node backend and MongoDB free-tier — may slow under high concurrent users.
  Mitigation: design REST APIs statelessly so the app can scale horizontally; use indexes in MongoDB to speed queries.

- **Latency / ETA accuracy:** dependent on network and external Maps API.
  Mitigation: cache frequent map responses and use optimistic ETA with periodic driver updates.

- **Storage / Data size:** large media (images) increase storage and bandwidth.
  Mitigation: store images in cloud storage (S3) and keep DB for metadata only.

6.1 Test Plan/ Test Cases

- **User auth:** register/login with valid & invalid data.

- **Restaurant & menu APIs:** create/read/update/delete endpoints.

- **Edge cases:** empty cart, unavailable menu item, DB connection failure.

- **Performance:** concurrent 50–100 simulated requests to key endpoints.

6.2 Test Procedure

- Use **Postman** / PowerShell `Invoke-RestMethod` for functional tests.
- Run seed script to create sample data.
- Manual frontend walkthrough: register → add to cart → checkout → track order.
- For basic load test, use a simple request loop (e.g., `ab` or a small Node script) against `/api/restaurants` and `/api/orders`.

6.3 Performance Outcome

- **Functional tests:** all core APIs returned correct responses; order flow completed end-to-end.

- **Basic load test:** single-node backend handled ~50 requests/sec for simple GETs; higher write load (orders) showed increased latency (~200–400 ms).

- **Recommendations:** enable DB indexing, use connection pooling, deploy backend behind a load balancer and migrate media to object storage for production scale.

7 My learnings

During this 6-week internship, I gained strong hands-on experience in full-stack development and understood how real industry applications are built, tested, and deployed. I learned how different components of a system interact, how data flows between frontend, backend, and databases, and how modern applications ensure security, performance, and scalability.

I began by understanding the basics of Java and JDBC, which helped me build a foundation for backend logic and data handling. Gradually, I moved into full-stack development using **Node.js**, **Express.js**, **React**, and **MongoDB Atlas**. I learned how to design REST APIs, implement secure authentication using JWT, manage routes, validate data, and connect the backend with a cloud database. I also learned how to structure a professional backend with modules for authentication, restaurants, menu items, and orders.

On the frontend side, I learned how to set up a React project, create components, handle API calls, and manage basic state for user interactions. I gained practical experience in debugging issues, resolving dependency errors, handling CORS issues, managing environment variables, and running both frontend and backend servers simultaneously.

Throughout the internship, I also learned the importance of **Git & GitHub**. I understood how version control works, how to initialize a repository, make commits, push projects to GitHub, and organize a full project for documentation and submission.

Overall, this internship helped me improve my problem-solving skills, boosted my confidence in working with real-time applications, and gave me practical exposure to the workflow followed in real software industries. It also helped me experience how a project moves from requirement → design → development → testing → documentation. These learnings will strongly support my future career in software development and full-stack engineering.

8 Future work scope

Although the core features of the Food Delivery Application were implemented during the internship, there are several enhancements and advanced features that can be developed in the future to make the system more robust, scalable, and closer to real-world food delivery platforms.

- **1. Integration of Online Payment Gateway**

Currently, the system uses a simple order placement mechanism. In the future, payment services like Razorpay, Stripe, or UPI can be integrated to allow secure online transactions.

- **2. Real-Time Delivery Tracking Using Maps API**

The present version simulates delivery updates. Future enhancements can include live driver location tracking using Google Maps API or Mapbox, enabling accurate ETA and route visibility.

- **3. Restaurant Analytics Dashboard**

A full analytics dashboard can be added for restaurant owners to monitor:

- Most ordered dishes
- Peak order timings
- Revenue statistics
- Customer behavior insights

This would make the system more useful for business decisions.

- **4. Push Notifications for Order Updates**

Implementing push notifications can improve user experience by instantly informing customers about order status:

- Order accepted
- Food being prepared
- Driver picked up the order
- Driver is near your location
- Order delivered
- **5. Enhanced UI/UX Design**

The frontend can be improved with modern UI frameworks like Tailwind CSS or Material UI to deliver a smoother and more visually appealing user experience.

- **6. Support for Multiple Cuisines & Filters**

Advanced filtering options (Veg/Non-Veg, price range, rating, cuisine type) can be added to enhance the browsing experience.

- **7. User Personalization and Recommendations**

Machine learning models can be added later to recommend restaurants and dishes based on user behavior, past orders, and popular trends.

- **8. Admin Panel for System Monitoring**

An internal admin interface could help manage users, restaurants, orders, drivers, complaints, and platform activity.

- **9. Multi-City or Multi-Location Support**

The app can be expanded to support multiple cities with location-based restaurant listings.