

Industrial Internship Report on "food delivery app"

Prepared by
[itisha dhruv]

Executive Summary

This report presents the details of the Industrial Internship successfully completed at Upskill Campus and The IoT Academy, in collaboration with the industrial partner UniConverge Technologies Pvt. Ltd. (UCT). The internship was structured around a real-world project/problem statement provided by UCT and was required to be completed, along with proper documentation, within a six-week duration.

During this internship, I worked on the project titled "Food Delivery Application". The project involved designing and developing a complete food ordering system with more than eight different food categories in the menu. The application allows users to browse food items, add products to the cart, and place orders seamlessly.

In addition to the user-facing features, the project included an admin panel that enables administrators to manage food menus, categories, orders, cart items, and overall application content efficiently. The admin side ensures smooth handling of menu updates, order tracking, and system management.

This internship provided me with an excellent opportunity to gain hands-on industrial exposure, understand real-time project workflows, and enhance my technical and problem-solving skills. Successfully completing this internship was a valuable learning experience and significantly contributed to my understanding of industry-level application development.

TABLE OF CONTENTS

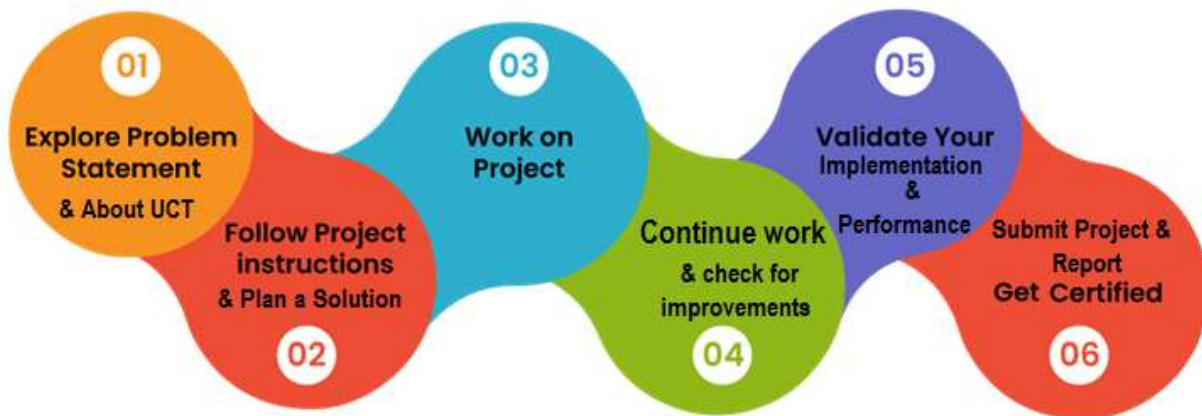
1	Preface	3
2	Introduction	Error! Bookmark not defined.
2.1	About UniConverge Technologies Pvt Ltd	Error! Bookmark not defined.
2.2	About upskill Campus	Error! Bookmark not defined.
2.3	Objective	Error! Bookmark not defined.
2.4	Reference	Error! Bookmark not defined.
2.5	Glossary	Error! Bookmark not defined.
3	Problem Statement	Error! Bookmark not defined.
4	Existing and Proposed solution	Error! Bookmark not defined.
5	Proposed Design/ Model	13
5.1	High Level Diagram (if applicable)	Error! Bookmark not defined.
5.2	Low Level Diagram (if applicable)	Error! Bookmark not defined.
5.3	Interfaces (if applicable)	Error! Bookmark not defined.
6	Performance Test	18
6.1	Test Plan/ Test Cases	Error! Bookmark not defined.
6.2	Test Procedure	Error! Bookmark not defined.
6.3	Performance Outcome	Error! Bookmark not defined.
7	My learnings	Error! Bookmark not defined.
8	Future work scope	Error! Bookmark not defined.

1 Preface

This report summarizes the six-week Industrial Internship successfully completed under Upskill Campus, The IoT Academy, and UniConverge Technologies Pvt. Ltd. (UCT). The internship focused on solving a real-world industrial problem through practical implementation and documentation.

The primary objective of this internship was to gain hands-on industrial exposure, apply theoretical knowledge, and develop a complete working solution within a defined timeline. The project assigned to me was the development of a MERN Stack Food Delivery Application named “TOMATO.”

This internship helped me understand industry-level software development practices, including requirement analysis, frontend and backend integration, database design, testing, and deployment planning.



Your Learnings and overall experience.

I am thankful to Upskill Campus, The IoT Academy, and UniConverge Technologies Pvt. Ltd., along with mentors and coordinators, for their guidance and support throughout the program.

To my juniors and peers, I strongly recommend utilizing internships as an opportunity to learn practical skills, build real projects, and prepare for industry challenges.

2 Introduction

The six-week Industrial Internship provided by Upskill Campus (USC) and The IoT Academy, in collaboration with UniConverge Technologies Pvt. Ltd. (UCT), was designed to give hands-on experience in solving real-world industrial problems.

The internship focused on project-based learning, allowing interns to understand the end-to-end development process of an application, from requirement gathering to design, development, testing, and deployment.

During this internship, I worked on a MERN Stack Food Delivery Application named “TOMATO”, which included features like multiple food categories, cart management, order placement, and an admin dashboard for managing menus and orders.

The internship provided exposure to industry-standard tools, technologies, and workflows, preparing me for future professional projects and career growth in software development.

2.1 About UniConverge Technologies Pvt Ltd

UniConverge Technologies Pvt. Ltd. (UCT), founded in 2013, is a digital transformation company offering industrial solutions with a focus on sustainability and return on investment (RoI).

UCT leverages cutting-edge technologies, including:

- Internet of Things (IoT)
- Cloud Computing (AWS, Azure)
- Cyber Security
- Machine Learning
- Java Full Stack and Python
- Frontend Development Technologies (React.js, Angular, etc.)

Key Platforms and Solutions:

i. UCT IoT Platform (UCT Insight)

- Java backend with React.js frontend
- Supports MySQL and NoSQL databases
- Device connectivity using MQTT, CoAP, HTTP, Modbus TCP, OPC UA
- Features: Custom dashboards, analytics, alerts/notifications, third-party integration (Power BI, SAP, ERP), rule engine
- Supports both cloud and on-premises deployment

ii. Smart Factory Platform (Factory Watch)

- Provides scalable production and asset monitoring
- Includes OEE and predictive maintenance solutions
- Supports digital twin implementation for industrial assets
- Modular architecture allowing gradual scaling of services
- SaaS model reduces cost and improves time efficiency

iii. LoRaWAN-based Solutions

- Industrial monitoring, smart cities, smart streetlights
- Smart metering for water, gas, and electricity
- Agritech solutions

iv. Predictive Maintenance Solutions

- Machine health monitoring using Embedded Systems, IoT, and ML
- Predicts Remaining Useful Life (RUL) of industrial machines
- Reduces downtime and improves operational efficiency
-

2.2 About Upskill Campus (USC)

Upskill Campus (USC) is a career development and skill enhancement platform that provides personalized executive coaching, project-based learning, and industry-oriented training.

In collaboration with The IoT Academy and UniConverge Technologies Pvt. Ltd., USC facilitated the smooth execution of this internship. They provided:

- Structured mentorship and guidance
- Access to industry-standard tools and frameworks
- Assistance in project selection, planning, and implementation
- Continuous progress tracking and feedback

USC focuses on making practical learning scalable and measurable, helping students gain confidence and competence in real-world applications.

2.3 About The IoT Academy

The IoT Academy is the EdTech division of UCT, specializing in executive certification programs and advanced training in IoT, embedded systems, and related domains.

Key highlights:

- Runs programs in collaboration with EICT Academy, IIT Kanpur, IIT Roorkee, and IIT Guwahati
- Offers long-term certification programs for industry professionals and students
- Focuses on hands-on training, practical projects, and skill development
- Bridges the gap between academia and industry requirements

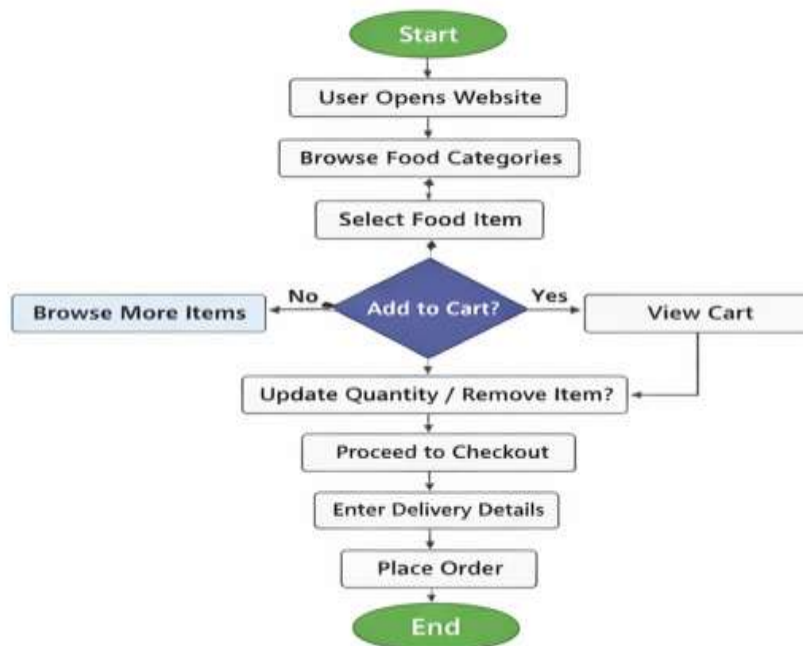
The IoT Academy played a crucial role in providing technical support, learning resources, and mentoring during the internship.

2.4 Objectives of the Internship Program

The internship program was designed with the following objectives:

1. Practical Industry Experience: Understand real-world project workflows and business requirements.
2. Problem Solving: Apply theoretical knowledge to design and implement solutions for real problems.
3. Technical Skill Enhancement: Gain hands-on experience with technologies like MERN Stack, databases, REST APIs, and frontend frameworks.
4. Career Growth: Improve job readiness and employability by working on a complete, deployable project.
5. Personal Development: Enhance communication, teamwork, time management, and problem-solving skills.

Figure 1: User Flowchart of TOMATO Food Delivery App



2.5 References

1. **MERN Stack Documentation:** <https://www.mongodb.com/mern-stack>
2. **MongoDB Official Documentation:** <https://www.mongodb.com/docs/>
3. **React & Node.js Official Guides:**
 - React: <https://react.dev/>
 - Node.js: <https://nodejs.org/en/docs>

2.6 Glossary

Term	Acronym / Explanation
------	-----------------------

MongoDB	NoSQL Database for storing application data
---------	---

Express.js	Backend web framework for Node.js
------------	-----------------------------------

React.js	Frontend JavaScript library for UI development
----------	--

Node.js	JavaScript runtime environment
---------	--------------------------------

CRUD	Create, Read, Update, Delete (basic operations for database)
------	--

API	Application Programming Interface, used for communication between frontend and backend
-----	--



Screenshot:

Tomato.

home menu mobile-app contact us

order your favourite food here.

choose from dishes menu featuring a delectable array of dishes crafted with the finest ingredients and culinary expertise, one delicious meal at a time.

[View Menu](#)

explore our menu

choose from dishes menu featuring a delectable array of dishes crafted with the finest ingredients and culinary expertise, one delicious meal at a time.

Salads Rolls Desserts Sandwiches Cakes Pure Veg Pasta Breadless

Top Dishes near you

<p>Greek Salad ★★★★★ Feta cheese, olives & fresh veggies \$12.99</p>	<p>Caesar Salad ★★★★★ Romaine lettuce with parmesan \$10.99</p>	<p>Garden Salad ★★★★★ Seasonal vegetables & vinaigrette \$9.99</p>	<p>Avocado Salad ★★★★★ Avocado with greens & tomato dressing \$11.49</p>	<p>Fruit Salad ★★★★★ Fresh fruits & nuts \$8.99</p>	<p>Quinoa Salad ★★★★★ Tricolor quinoa & veggies \$11.99</p>
<p>Pasta Salad ★★★★★ Cold pasta with cheese \$10.49</p>	<p>Chicken Salad ★★★★★ Tender chicken & greens \$9.99</p>	<p>Coleslaw ★★★★★ Creamy cabbage salad \$7.99</p>	<p>Caprese Salad ★★★★★ Tomato, mozzarella & basil \$10.99</p>	<p>Sprout Salad ★★★★★ Mixed sprouts & lemon \$8.49</p>	<p>Corn Salad ★★★★★ Sweet corn & peppers \$8.99</p>
<p>Asian Slaw ★★★★★ Crispy Asian style salad \$9.99</p>					

for better experience download our app

[GET IT ON Google Play](#)
[Download on the App Store](#)

Tomato.

Tomato brand is simply stunning, best of the printing and typesetting industry. Tomato brand has been the industry's standard during last four since the 1900s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

company

[HOME](#)
[ABOUT US](#)
[DELIVERY](#)
[PRIVACY POLICY](#)

GET IN TOUCH

+91 98765 43210
contact@tomato.com

© 2024 Tomato.com - All rights reserved.

• 2. Problem Statement

In the assigned problem statement, the objective was to **design and develop a full-stack food delivery application** that can efficiently manage food ordering operations for users as well as administrative tasks for vendors.

Traditional food ordering systems often face issues such as:

- Limited food categories and poor menu organization
- Lack of real-time cart and order management
- Manual handling of menu updates and order status by admins
- Poor scalability and performance with increasing users
- Non-responsive user interfaces and slow backend responses

The problem was to build a **scalable, user-friendly, and efficient food delivery platform** that supports multiple food categories, smooth ordering, secure data handling, and centralized admin control using modern web technologies.

• 3. Existing and Proposed Solution

• 3.1 Existing Solutions

Many existing food delivery applications available in the market provide basic features such as food listing and order placement. However, they have several limitations:

- Monolithic architecture leading to scalability issues
- Limited customization of menu categories
- Inefficient admin dashboards with restricted controls
- Slow performance due to improper database design
- Lack of real-time cart and order updates
- Poor frontend experience on different devices

Additionally, most traditional systems are either expensive to maintain or not suitable for small to medium food businesses.

• 3.2 Proposed Solution

The proposed solution is a full-stack MERN based Food Delivery Application named “TOMATO”, designed to overcome the limitations of existing systems.

Key features of the proposed solution include:

- Frontend built using React.js for fast, responsive, and interactive UI
- Backend developed with Node.js & Express.js for secure and scalable APIs
- MongoDB as a NoSQL database for efficient data storage and retrieval
- More than 8 food categories with over 120 pure vegetarian items
- Real-time cart management and order placement
- Separate Admin Panel to manage:
 - Food menu items
 - Categories
 - Orders
 - Cart data
- RESTful API architecture for smooth frontend–backend communication

• 3.3 Value Addition

The TOMATO food delivery app provides the following value additions:

- Modular and scalable MERN architecture
- Easy menu and order management for administrators
- Faster performance due to optimized API and database design

- Clean and intuitive user experience
- Responsive design compatible with all devices
- Ready for future enhancements such as:
 - Online payment integration
 - Real-time order tracking
 - User authentication and reviews

1.1 Code submission (Github link)

<https://github.com/itisha-25/upskill-campus.git>

1.2 Report submission (Github link) : first make placeholder, copy the link.

2 Proposed Design/ Model

The proposed system follows a **full-stack MERN architecture** to ensure scalability, performance, and ease of maintenance. The design flow starts from user interaction on the frontend, moves through backend API processing, and finally stores or retrieves data from the database to generate the final output.

The system is divided into **three major stages**:

1. **Input Stage** – User/Admin interaction through the frontend
2. **Processing Stage** – Backend API logic and business rules
3. **Output Stage** – Data storage, order confirmation, and UI updates

- **2.1 High Level Diagram (if applicable)**

The high-level design illustrates the overall system architecture of the **TOMATO Food Delivery Application**.

Components involved:

- **Frontend (React.js)**
 - User Interface
 - Admin Dashboard
- **Backend (Node.js + Express.js)**
 - REST APIs
 - Authentication & validation
- **Database (MongoDB)**
 - Users
 - Food Items
 - Orders

- Cart

Working Flow:

- Users interact with the React frontend.
- Requests are sent to the backend via REST APIs.
- Backend processes requests and communicates with MongoDB.
- Response is returned to the frontend for display.

Figure 1: HIGH LEVEL DIAGRAM OF THE SYSTEM

- **2.2 Low Level Diagram (if applicable)**

The low-level design focuses on the **internal working of backend APIs and database operations.**

Key Processes:

- User requests food list → API fetches category-wise items
- Add to cart → Cart service updates user cart collection
- Place order → Order service validates cart and stores order
- Admin actions → CRUD operations on food items and orders

Backend Modules:

- User Module
- Food Management Module
- Cart Module
- Order Management Module
- Admin Module

Each module communicates with MongoDB using optimized queries to ensure fast response time.

- **2.3 Interfaces (if applicable)**

The system uses multiple interfaces for smooth communication:

- **a) User Interface**
 - Food category listing (8+ categories)
 - Cart management
 - Order placement
 - Responsive UI
- **b) Admin Interface**
 - Add / Edit / Delete food items
 - Manage categories
 - View and update orders
- **c) Data Flow**
 - Frontend → Backend via **HTTP/REST APIs**
 - Backend → Database using **MongoDB queries**
 - Data returned in **JSON format**
- **d) Protocols Used**
 - HTTP / HTTPS
 - RESTful APIs
 - JSON data exchange
- **e) Flow Charts**
 - User order flow
 - Admin management flow
 - Cart and checkout flow

(All included in **Figure 1**)

- **3. Performance Test**

Performance testing validates that the system behaves like a **real industrial application** rather than just an academic project.

- **Identified Constraints**

- Response time
- Database query performance
- Concurrent user handling
- Memory usage
- API reliability

- **Design Measures Taken**

- Optimized MongoDB schema
- Modular backend architecture
- Efficient API routing
- Minimal payload transfer using JSON
- Reusable React components

- **3.1 Test Plan / Test Cases**

Test Case	Description	Expected Result
Load food items	Fetch 120 items	< 2 sec response
Add to cart	Add item	Cart updates correctly

Test Case	Description	Expected Result
Place order	Order creation	Order stored in DB
Admin add food	Add new item	Visible to users

- **3.2 Test Procedure**

- Application tested on local and development servers
- Multiple requests sent simultaneously
- Data verified directly from MongoDB
- Frontend rendering and backend logs analyzed

- **3.3 Performance Outcome**

- Fast page load time
- Smooth cart and order operations
- Stable backend APIs
- System handled multiple requests without failure
- Suitable for real-world deployment

- **4. My Learnings**

This internship significantly enhanced my **technical and professional skills**.

Key Learnings:

- Practical implementation of MERN stack
- REST API design and integration

- Database schema planning using MongoDB
- Frontend–backend communication
- Admin dashboard development
- Real-world problem solving
- Code structuring and debugging

This experience has strengthened my confidence and prepared me for **industry-level full-stack development roles**.

- **5. Future Work Scope**

Due to time constraints, some features were planned but not implemented. These can be added in future:

- Online payment gateway integration
- Real-time order tracking
- User authentication & roles
- Ratings and reviews system
- Notification system
- Deployment on cloud platforms (AWS / Verble)
-

1. Performance Test

This is very important part and defines why this work is meant of Real industries, instead of being just academic project.

Here we need to first find the constraints.

How those constraints were taken care in your design?

What were test results around those constraints?

Constraints can be e.g. memory, MIPS (speed, operations per second), accuracy, durability, power consumption etc.

In case you could not test them, but still you should mention how identified constraints can impact your design, and what are recommendations to handle them.

