Industrial Internship Report on

"Multi-client website offering client services"

Prepared by

Surbhi Sharma

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 **Multi-client website offering client services,** On this website merchants are the primary clients . Merchants will be able to sign up at the site and create a page for themselves that display a list of their services and the pricing. The users who are customers to clients will be able to sign up as users and purchase goods or services from the merchants. There will be a standard checkout process throughout which is integrated into a payment gateway 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 the internship, I worked on developing a multi-client website that offers various services, showcasing my full stack development skills. The project involved creating a user-friendly frontend where clients could browse and select services, complemented by a responsive design to ensure accessibility across devices. On the backend, I implemented a robust API using Node.js and Express, managing client requests and handling service data from a MongoDB database. The application featured user authentication to manage client profiles and service orders securely. I focused on seamless integration between the frontend and backend, ensuring efficient data retrieval and submission. Overall, this project not only reinforced my technical skills but also demonstrated my ability to create a functional, real-world application that meets diverse client needs.

This internship played a crucial role in career development for several reasons. First, provided practical, hands-on experience that enhances theoretical knowledge gained in the classroom, helping apply what I've learned in real-world scenarios. This experience is invaluable in developing specific skills that are often sought after by employers, such as teamwork, problem-solving, and technical proficiency.

Brief about project/problem statement:

On this website merchants are primary customers. Merchants will be able to sign up at the site and create a page for themselves that display a list of their services and the pricing.

The users who are customers to merchants will be able to sign up as users and purchase goods or services from the merchants. There will be a standard checkout process throughout which is integrated into a payment gateway system.

The site could offer different categories of services, like home, beauty, pet, etc. Merchants can select their category when signing up. Customers can browse and search for merchants by category. Merchants can customize their pages with images, descriptions, availability, and other service details.

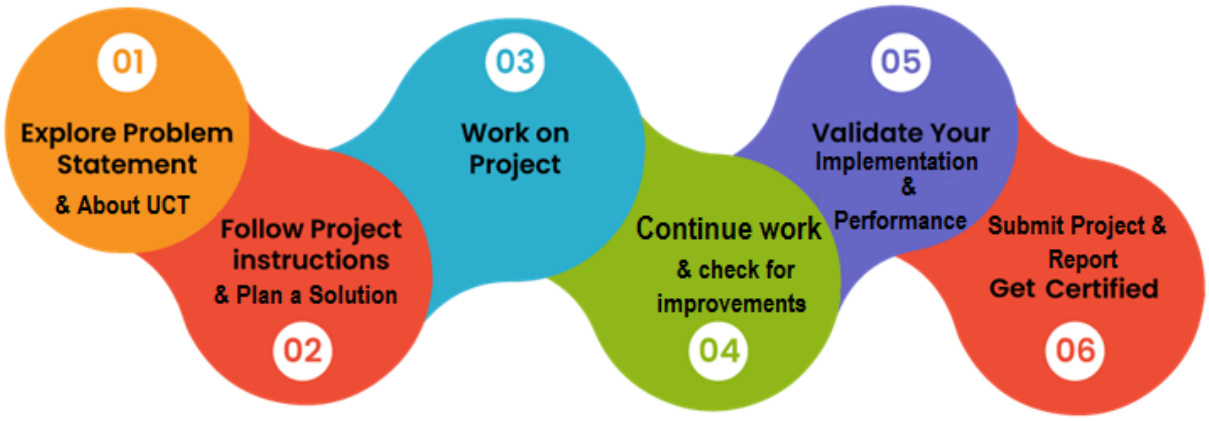
The checkout process will securely collect customer and payment information. Integrating a payment gateway allows for secure credit card processing. Email receipts can be sent to the customer and merchant after purchase. Merchants should have a dashboard to view orders and manage their accounts. Customers can leave reviews and ratings for merchants. Additional

features could include appointment scheduling, notifications, customer accounts and order history.

The multi-tenant structure allows easy scaling as more merchants and customers join the platform. Robust admin tools help manage users, categories, disputes, and platform growth over time. The core functionality connects service providers to customers in a seamless online marketplace.

This opportunity given by **USC/UCT** can helped me explore different career paths and clarify my career goals. By experiencing various roles and company cultures, I can better understand what I enjoy and where I excel, leading to more informed decisions about my future.

How Program was planned



Learnings and overall experience:

This experience deepened my understanding of web development by integrating IoT concepts into the application. I learned to create responsive and user-friendly interfaces, manage backend services using technologies like Node.js and Express, and interact with databases for data storage and retrieval. Additionally, I gained insights into user authentication and real-time data handling, enhancing your ability to build dynamic applications.

Thank to all (USC/UCT seniors), who have helped me directly or indirectly.

Message to juniors and peers:

I encourage you to seek out internship that align with your interests. This kind of internships offer invaluable opportunities to learn, network, and grow as a professional. Don't hesitate to

take the plunge—it's a fantastic way to gain insights and skills that will serve you well in your future careers!

Best of luck, and make the most of your opportunities!

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.

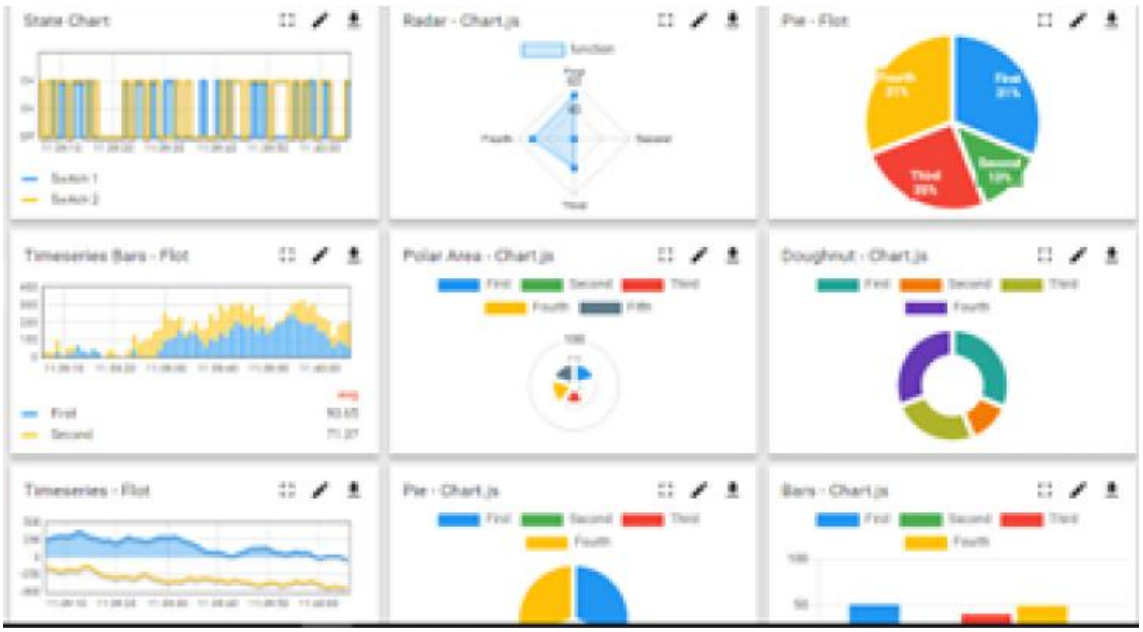
i. UCT IoT Platform (UCT Insight)

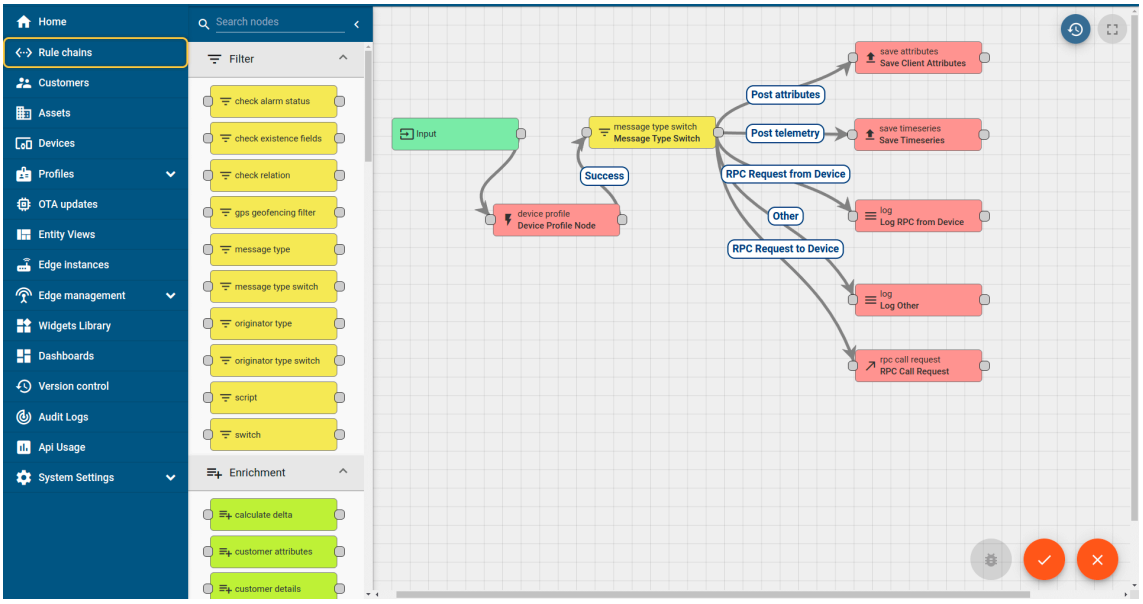
UCT Insight is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable "insight" for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

- 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.

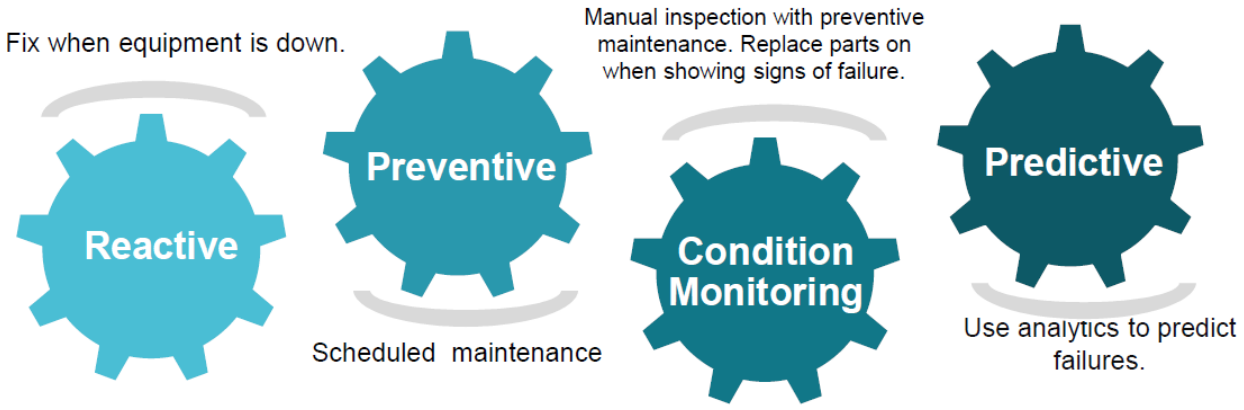
Machine	Operator	Work Order ID	Job ID	Job Performance	Job Progress		Output		Rejection	Time (mins)				Job Status	End Customer
					Start Time	End Time	Planned	Actual		Setup	Pred	Downtime	Idle		
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i
CNC_S7_81	Operator 1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

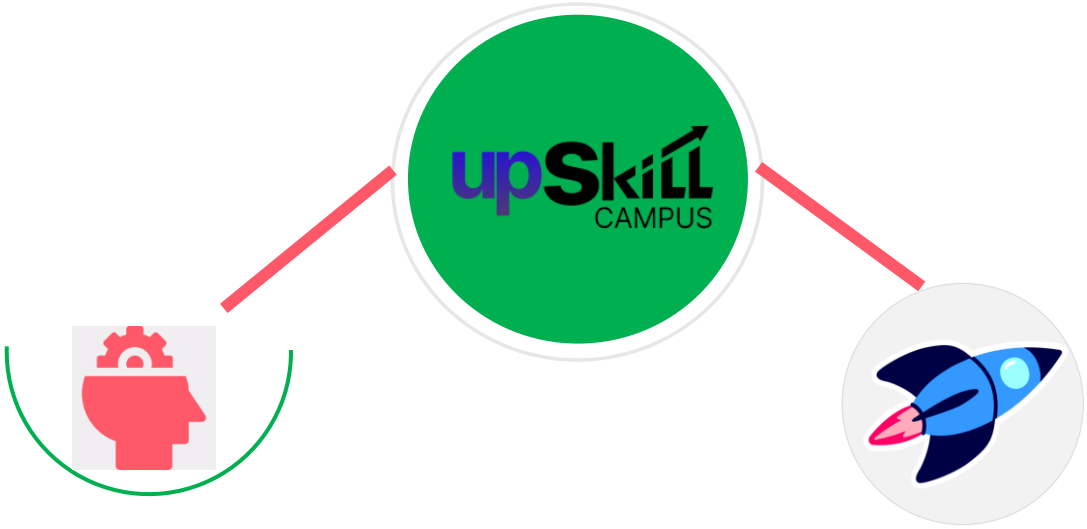
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



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.

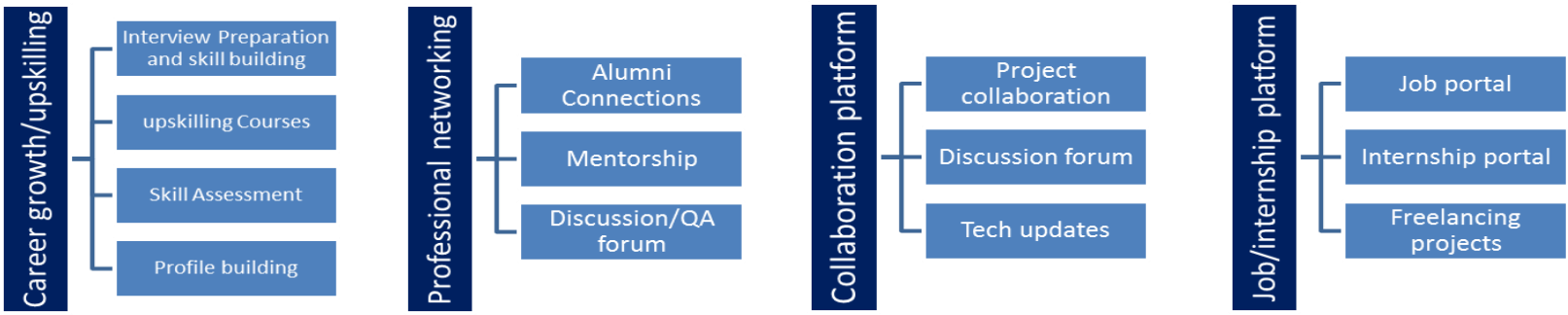
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



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.

3. Problem Statement

Multi-client website offering client services

Here merchants are your primary customers. Merchants must be able to sign up at the site and create a page for themselves that display a list of their services and the pricing.

The users who are customers to your customers should be able to sign up as users and purchase goods or services from the merchants. There should be a standard checkout process throughout which is integrated into a payment gateway system.

The site could offer different categories of services, like home, beauty, pet, etc. Merchants can select their category when signing up. Customers can browse and search for merchants by category. Merchants can customize their pages with images, descriptions, availability, and other service details.

The checkout process should securely collect customer and payment information. Integrating a payment gateway allows for secure credit card processing. Email receipts can be sent to the customer and merchant after purchase. Merchants should have a dashboard to view orders and manage their accounts. Customers can leave reviews and ratings for merchants. Additional features could include appointment scheduling, notifications, customer accounts and order history.

The multi-tenant structure allows easy scaling as more merchants and customers join the platform. Robust admin tools help manage users, categories, disputes, and platform growth over time. The core functionality connects service providers to customers in a seamless online marketplace.

4 Existing and Proposed solution

Summary of existing solutions provided by others and their limitations:

Many multi-client solutions, such as SaaS platforms (e.g., Salesforce, HubSpot) or custom MERN stack applications, use a **multi-tenant architecture**, where a single system serves multiple clients (tenants). Each client may have their own data, user management, and branding requirements.

However, these solutions face several challenges:

1. **Data Isolation & Security**: Ensuring complete separation of client data is crucial. If not handled correctly, it could lead to security risks or data breaches.
2. **Scalability**: As the number of clients grows, performance can degrade. Multi-client platforms need to scale horizontally and manage traffic efficiently.
3. **Customization & Branding**: Clients often require unique branding and features, which can complicate UI/UX development and increase maintenance complexity.
4. **Performance Issues**: Heavy traffic or complex queries from multiple clients can lead to slow response times or downtime, especially if the system isn't optimized for concurrency.
5. **Maintenance & Upgrades**: Rolling out updates or bug fixes without affecting one client's setup can be challenging.
6. **Client-Specific Features**: Handling varying feature sets, permissions, and integrations for different clients (e.g., different payment gateways or third-party services) can be complex.

Proposed Solution to overcome above mentioned challenges:

1. Data Isolation & Security

- Use separate databases or tenant-specific collections to ensure data isolation.
- Implement Role-Based Access Control (RBAC) to manage permissions securely.
    - 2. Scalability & Performance
- Use MongoDB sharding for horizontal scaling and load balancing to manage high traffic.
- Implement caching (Redis) for frequently accessed data and use a CDN for static content.
- Apply API rate limiting to prevent overuse of resources.

3. Customization & Branding

- Allow dynamic branding and theming via React components, pulling data from the database.
- Use feature flagging (e.g., LaunchDarkly) to enable/disable features per client.
- Offer tenant-specific integrations (e.g., payment gateways, CRMs) via APIs.

4. Maintenance & Upgrades

- Use a staging environment for each tenant and blue/green deployment for seamless updates.
- Manage database schema changes with migration tools to ensure compatibility.

5. Client-Specific Features & Permissions

- Implement a modular feature set**,** where each client can toggle features on/off.
- Allow custom user roles and permissions for granular control over access.

6. Cost Management

- Implement usage-based billing for API calls, storage, or users, with flexible pricing tiers.

Value additions for future purpose:

- **Dynamic Customization**: Allow clients to easily customize the look, feel, and functionality of their instance through a no-code admin panel. This empowers users to modify their UI themes, workflows, and features without technical intervention.
- **Multilingual Support**: Add **internationalization (i18n)** support to allow the platform to cater to clients from different regions and languages, improving accessibility for global audiences.
- **AI-Driven Insights**: Integrate AI/ML models to provide clients with intelligent data analysis, predictive analytics, or automated recommendations to enhance decision-making.

4.1 Code submission (Github link)

https://github.com/surbhisharma1408/upskillcampus/tree/3337df33387a2cae5c2e318d6bdcb6f0caac10b6/Multi-Client%20Website

4.2 **Report submission (Github link) :**

https://github.com/surbhisharma1408/upskillcampus/blob/main/Multi-Client%20Website_Surbhi_USC_UCT.pdf

5 Proposed Design/ Model

Multi-Client MERN Stack Solution Design Flow

The proposed solution for a multi-client platform built using the **MERN stack (MongoDB, Express, React, Node.js)** focuses on key elements like **data isolation**, **scalability**, **customization**, and **performance**.

1. **User Flow**: Each tenant (client) gets a **customized experience**. Upon signing up, a new tenant is created, and they can personalize their settings (branding, features). Users log in via **JWT authentication**, with roles defining access levels.
2. **Backend Architecture**:
    - The **Node.js/Express backend** processes incoming API requests, checks authentication, and retrieves data specific to the tenant.
    - **MongoDB** stores tenant data, with **isolated data models** either in separate collections or databases per tenant.
3. **Customization**: Tenants can modify their branding, toggle features on/off via **feature flagging**, and access their own data through dynamic, tenant-specific UIs in React.
4. **Security & Access Control**: Authentication and **RBAC (Role-Based Access Control)** ensure secure access to tenant-specific data. **JWT tokens** and **tenant IDs** are used to identify and authorize requests.
5. **Scalability**: The system can **scale horizontally** with load balancers, **database sharding** in MongoDB, and **auto-scaling** via cloud services to accommodate increasing client demands.
6. **Performance Optimization**: **Caching** (e.g., with Redis) and **CDN** usage ensure fast data retrieval and content delivery, while optimized database queries maintain responsiveness.
7. **Maintenance & Upgrades**: The platform supports **CI/CD pipelines**, **blue/green deployments**, and **auto-scaling** to ensure smooth updates and maintenance.

This design ensures a **secure, customizable, scalable** multi-tenant application with optimized performance and minimal client impact during upgrades.

6 Performance Test

6.1 Test Plan/ Test Cases

Test Plan Overview:

- **Objective**: Ensure the multi-client MERN stack solution is secure, functional, scalable, and performs as expected.
- **Scope**: Includes frontend, backend, database, security, performance, and integration testing.

6.2 Test Procedure

Key Testing Areas:

1. **User Authentication**:
    - **Valid/Invalid Login**: Ensure users can log in with correct credentials and are denied access with incorrect ones.
    - **JWT Expiry**: Verify expired tokens are rejected.
2. **Tenant Isolation & Data Security**:
    - **Data Access Control**: Ensure tenants can only access their own data, no cross-tenant data leakage.
    - **Subdomain/Database Isolation**: Test tenant data isolation via subdomains or separate databases/collections.
3. **Role-Based Access Control (RBAC)**:
    - **Admin & User Roles**: Verify users with the **Admin** role can access admin features, while **regular users** are restricted.
    - **Tenant-Specific Permissions**: Ensure each tenant's roles and permissions are respected.
4. **Functional Testing**:
    - **Dashboard Data**: Ensure tenant-specific data is fetched and displayed correctly.
    - **Customization**: Test tenant ability to customize branding and toggle features.
5. **Performance Testing**:
    - **Load & Stress Testing**: Verify the system can handle high traffic and degrade gracefully under extreme load.
    - **Response Times**: Ensure the platform maintains acceptable performance under different loads.
6. **Security Testing**:
    - **SQL Injection & XSS Prevention**: Ensure the app is secure from common attacks.

- **Data Encryption**: Verify that sensitive data is encrypted during transmission.

6.3 Performance Outcome

The test plan ensures comprehensive coverage of functional, security, and performance aspects of the application. By testing user flows, data isolation, role management, and security vulnerabilities, the platform will be robust, secure, and scalable.

7 My learnings

Over the course of building and refining the Full stack application, I've gained extensive hands-on experience in key areas of software development, architecture, and scalability, which will be immensely valuable for my career growth. Here's a breakdown of what I've learned and how it contributes to my professional development:

- **MERN Stack Proficiency**:

  - Deepened my understanding of **MongoDB**, **Express.js**, **React**, and **Node.js**. I became proficient in creating full-stack applications, managing RESTful APIs, and building dynamic UIs using React.
  - Learned how to efficiently handle **client-side rendering**, **state management** with React, and building reusable components for better code maintainability.

- **Multi-Tenant Architecture**:

  - Worked with **multi-client architecture**, designing a platform that can support multiple clients while ensuring data isolation and customization per client.
  - Gained experience in handling various design patterns like **tenant-specific data models**, **role-based access control (RBAC)**, and the concept of **data partitioning** (separate databases or collections for tenants).

Impact on Career Growth:

- **Enhanced Problem-Solving Abilities**:

  - By designing a solution that handles multiple tenants and requires deep integration across different layers of the tech stack, I've sharpened my problem-solving skills, particularly when it comes to **scalability**, **data integrity**, and **user experience**.
  - These problem-solving skills will be valuable as I take on more complex projects or work in leadership roles that require making architectural decisions.

- **Full-Stack Development Expertise**:

  - This project has solidified my position as a full-stack developer, capable of managing both the frontend and backend of a web application, and has equipped me with the ability to **build scalable, secure, and maintainable applications**.

8 Future work scope

- **AI-Powered Analytics and Insights**

Idea: Integrating **AI-driven analytics** to provide **tenant-specific insights** based on usage patterns, user behaviors, or sales data (e.g., using machine learning algorithms to predict user churn, recommend features, or optimize performance).

Future Benefit: Offering insights and recommendations could help tenants optimize their operations and decision-making, adding a competitive edge to the platform.

- **Automated Data Backup and Disaster Recovery**

Idea: Implementing **automated data backups** and **disaster recovery** processes to ensure tenant data is always secure and recoverable in case of an emergency or system failure.

Future Benefit: This would add an additional layer of security and trust, ensuring that the platform meets higher standards for data protection and business continuity.

- **Personalization using User Behavior Analytics**

Idea: Implementing a **personalized user experience** based on **user behavior**—e.g., custom dashboards or personalized content recommendations—by analyzing user actions and preferences.

Future Benefit: This would create a more engaging user experience, making the platform feel more intuitive and tailored to each user's needs.