Industrial Internship Report on

”Grocery_Delivery_App”

Prepared by

Vedant Shigwan

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.

During my 6-week internship, I developed a **Grocery Delivery Web Application** that simulates an online grocery shopping experience. The platform allows users to browse products, add items to a cart, select payment methods, schedule delivery slots, and submit feedback after order placement.

This full-stack web application was built using **Python Flask** for backend routing and session management, **SQLite** for data storage, and **Tailwind CSS** for a modern responsive interface. The app supports essential e-commerce features such as cart management, flash messaging, user feedback, and delivery coordination.

Through this project, I gained hands-on experience in building real-world web applications from scratch, integrating frontend and backend components, and ensuring smooth user interaction with well-structured routes and templates. This internship helped enhance my technical skills in web development and taught me how to design user-friendly, efficient, and scalable web systems.

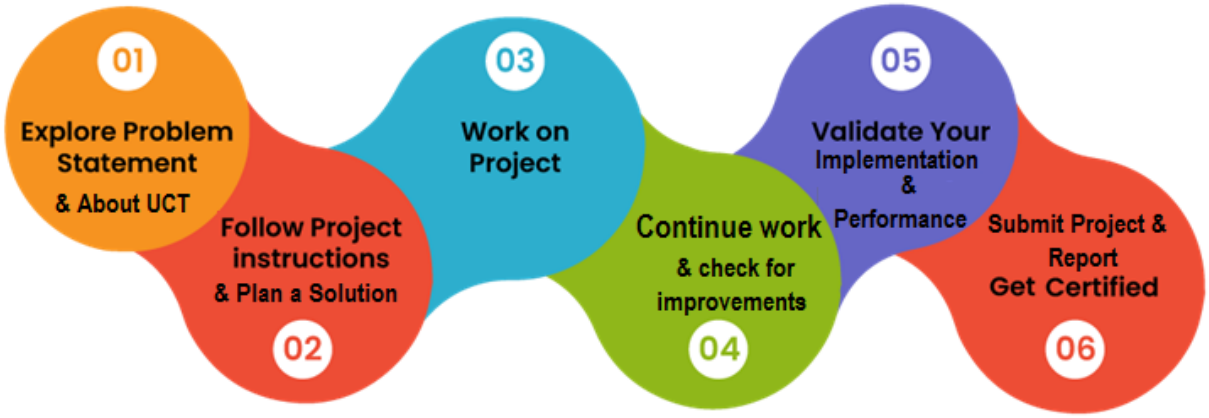
This internship gave me a very good opportunity to get exposure to Industrial problems and design/implement solution for that. It was an overall great experience to have this internship.

TABLE OF CONTENTS

1 Preface

During my 6-week internship at UniConverge Technologies Pvt Ltd (UCT), in association with upskill Campus (USC) and The IoT Academy, I worked on developing a Grocery Delivery Web Application using Python Flask. This internship offered hands-on exposure to real-world full-stack development and aligned well with my domain of study.

The program was structured over six weeks. In the first week, I explored UCT's work domains and selected the grocery app project. In the second week, I studied the technologies and proposed a design. Weeks 3 and 4 were dedicated to implementation, followed by performance testing in Week 5. Week 6 focused on report preparation and project submission.



I sincerely thank all mentors who guided me. I'm also grateful to my faculty and peers for their support.

To my juniors: Take such internships seriously—they're a great opportunity to build practical skills and industry-ready confidence.

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.

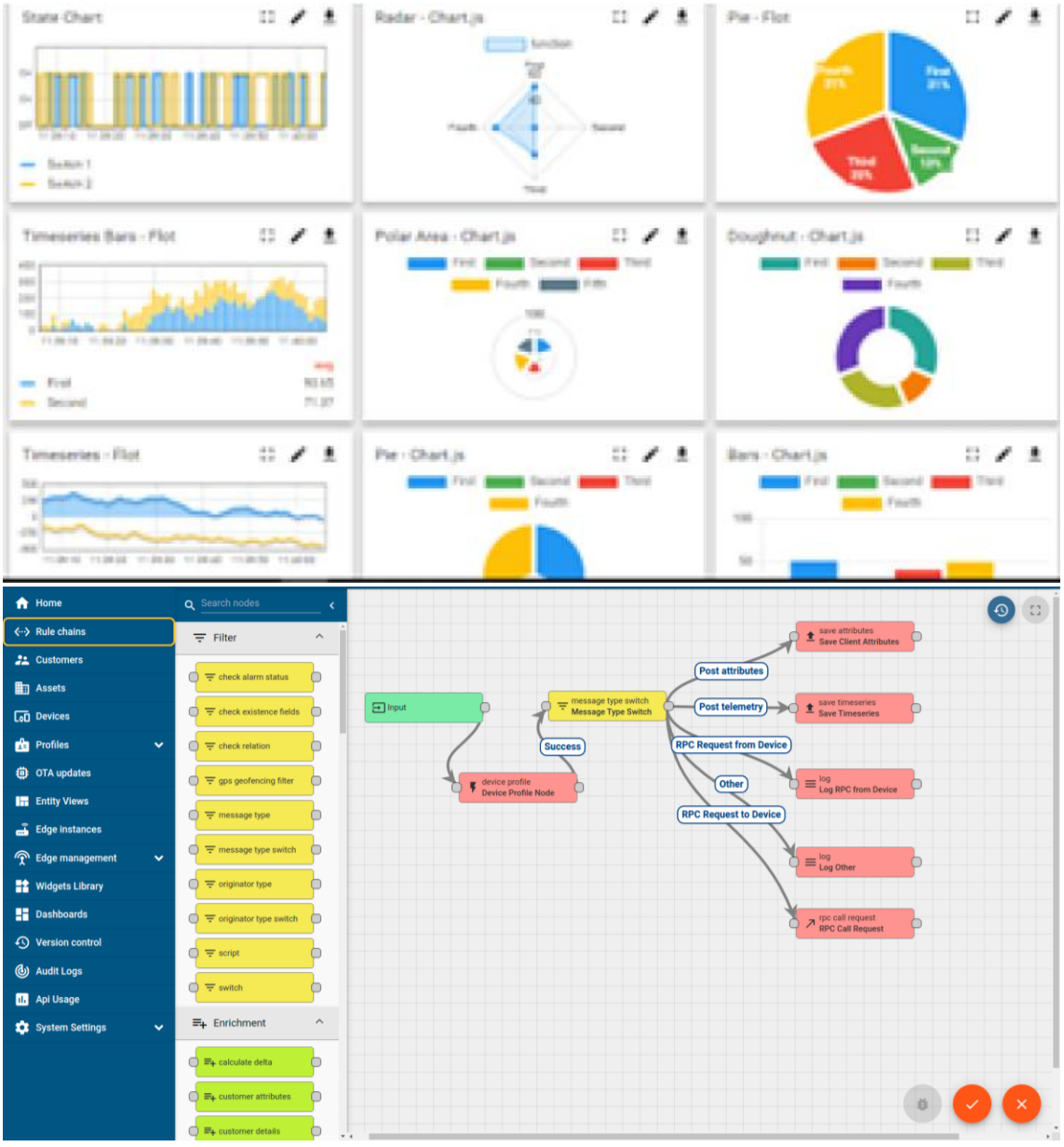
i. UCT IoT Platform (uct Insight)

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.

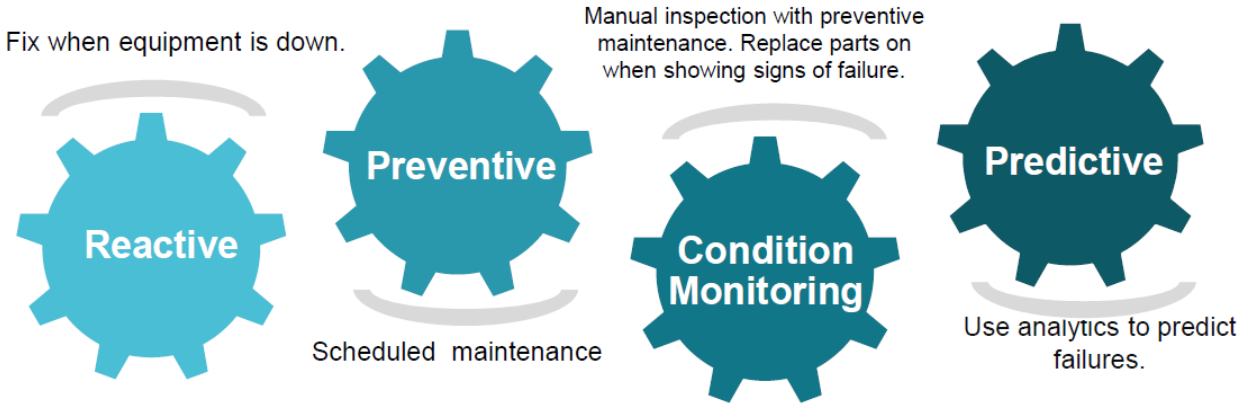
Machine	Operator	Work Order ID	Job ID	Job Performance	Job Progress		Output		Rejection	Time (mins)				Job Status	End Customer
					Start Time	End Time	Planned	Actual		Setup	Pred	Downtime	Idle		
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i
CNC_S7_81	Operator 1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

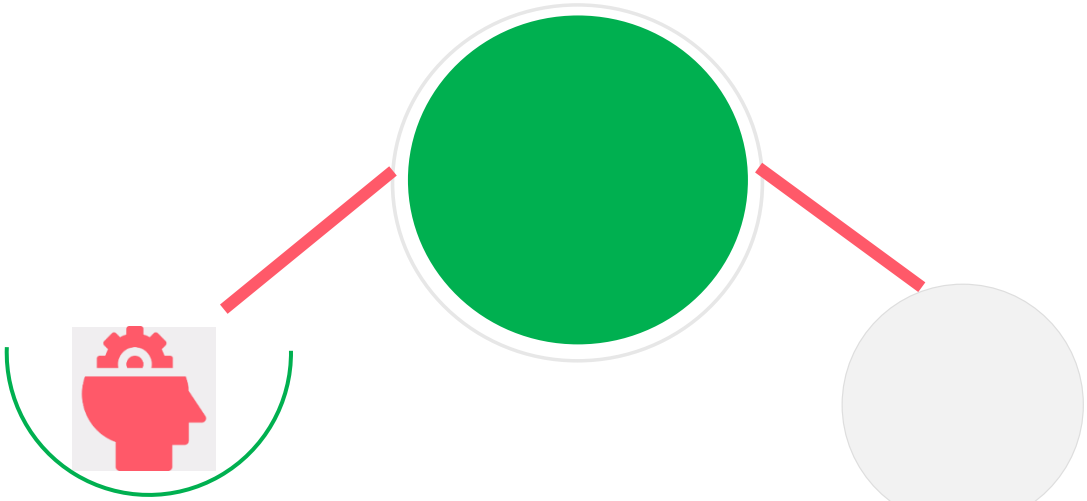
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



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.

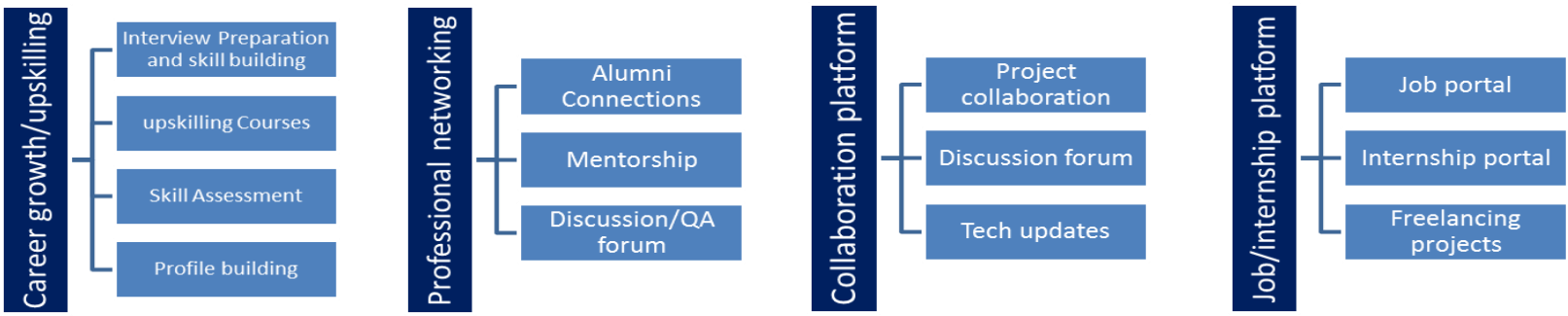
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



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

The assigned project was to develop a Grocery Delivery Web Application for a large-scale online departmental store. The application is intended to manage a vast inventory of grocery items stored in a central godown. Each item must be listed on the website with details such as name, quantity, and price.

The platform should allow users to log in, browse available groceries, add items to a cart, and proceed to checkout. Before finalizing the order, the user must be able to select a convenient delivery slot and choose from various payment methods such as UPI, cards, or cash on delivery.

In addition to core e-commerce functionality, the app can be extended with features like personalized item suggestions, order history tracking, ratings and reviews, loyalty points, and admin dashboards. The focus is on delivering a smooth, intuitive, and reliable user experience for both customers and administrators.

This project simulates a real-world grocery platform, giving insight into how online grocery systems work from both the customer and backend management perspective.

4 Existing and Proposed solution

Popular platforms like **BigBasket**, **Blinkit**, and **Amazon Fresh** already provide online grocery delivery services. These platforms are feature-rich and robust but often come with the following limitations:

- High development and maintenance cost
- Require large teams and complex infrastructure
- Difficult for small vendors or startups to replicate
- Less customizable for specific business models or regions

Proposed Solution:

My project proposes a **lightweight, easy-to-deploy Grocery Delivery Web Application** built using **Flask (Python)** and **SQLite**. It is tailored for small to mid-scale vendors who want to manage product listings, customer orders, and deliveries from a single dashboard. The system includes features like:

- Product catalog with quantity and pricing
- Cart and checkout flow
- Delivery slot selection
- Payment option selection
- Customer feedback collection

Value Addition:

- Clean and user-friendly interface using Tailwind CSS
- Session-based cart and flash messaging system
- Easy to extend (e.g., loyalty program, custom packaging, admin dashboard)
- Beginner-friendly codebase for academic and prototype use
- Fully offline/localhost deployable for demonstration or internal use

4.1 Code submission (Github link)

https://github.com/Vedant-Shigwan11/upskillcampus/blob/main/Grocery_Delivery_App.zip

4.2 Report submission (Github link) :

https://github.com/Vedant-Shigwan11/upskillcampus/blob/main/Grocery_Delivery_App_Vedant_USC_UCT.pdf

5 Proposed Design/ Model

The Grocery Delivery Web App follows a structured multi-stage design approach. It starts from user interaction and moves through cart management, payment, and delivery — ultimately leading to feedback and order completion.

5.1 High Level Diagram

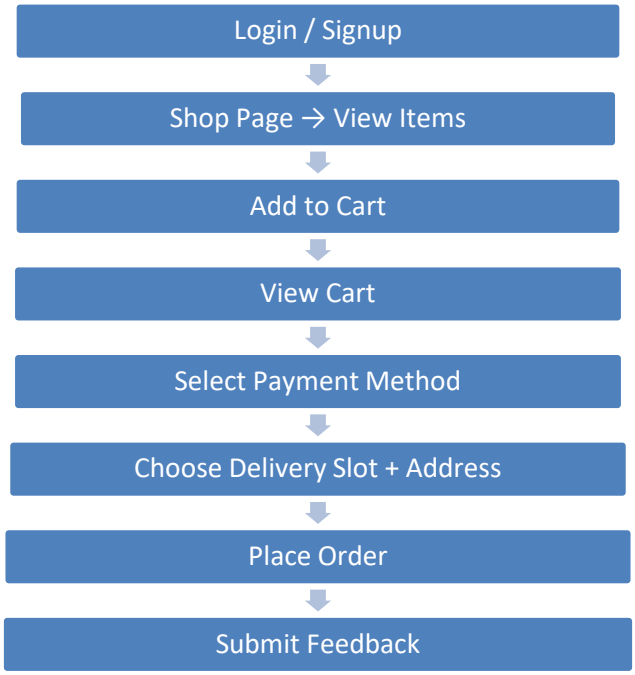


Figure 1: HIGH LEVEL DIAGRAM OF THE SYSTEM

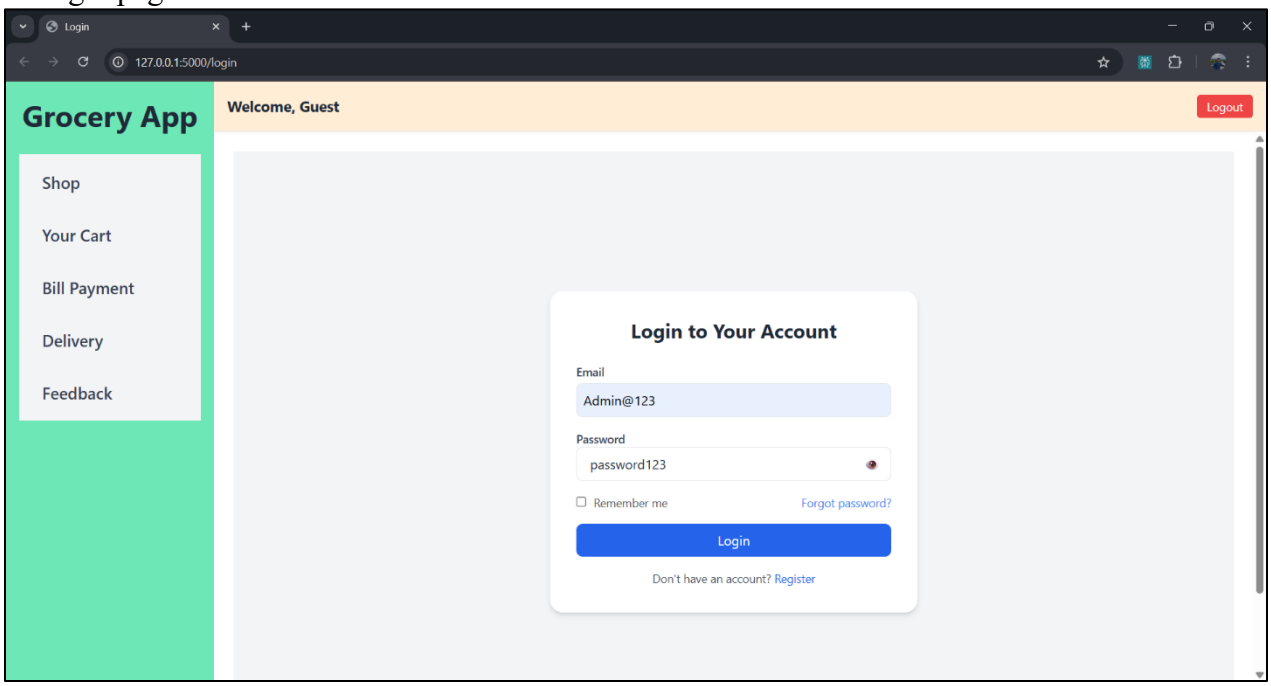
5.2 Low Level Diagram

- **User Session Management**
- **Product Model (name, price, quantity)**
- **Cart Logic (session-based)**
- **Order Data Processing**
- **Delivery Time Handling**
- **Feedback Submission**

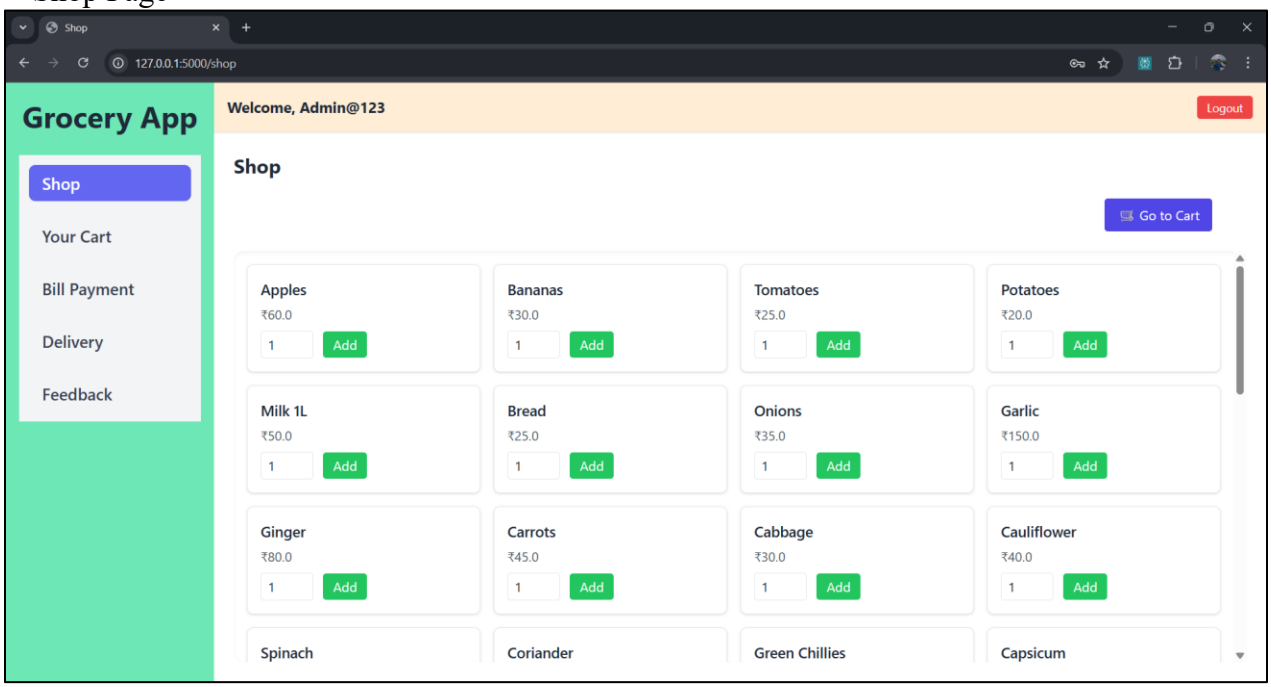
Each module interacts via Flask routes and form POST requests, storing or retrieving data from a SQLite database using SQLAlchemy ORM.

5.3 Interfaces

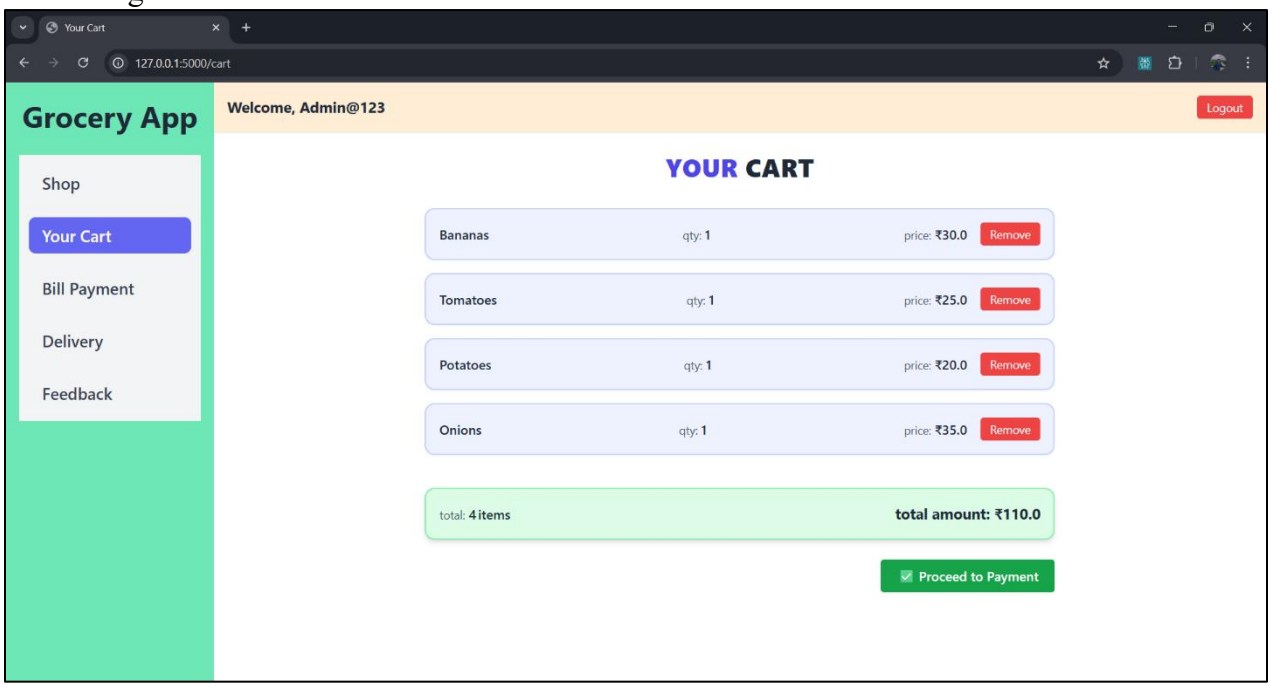
- Login page



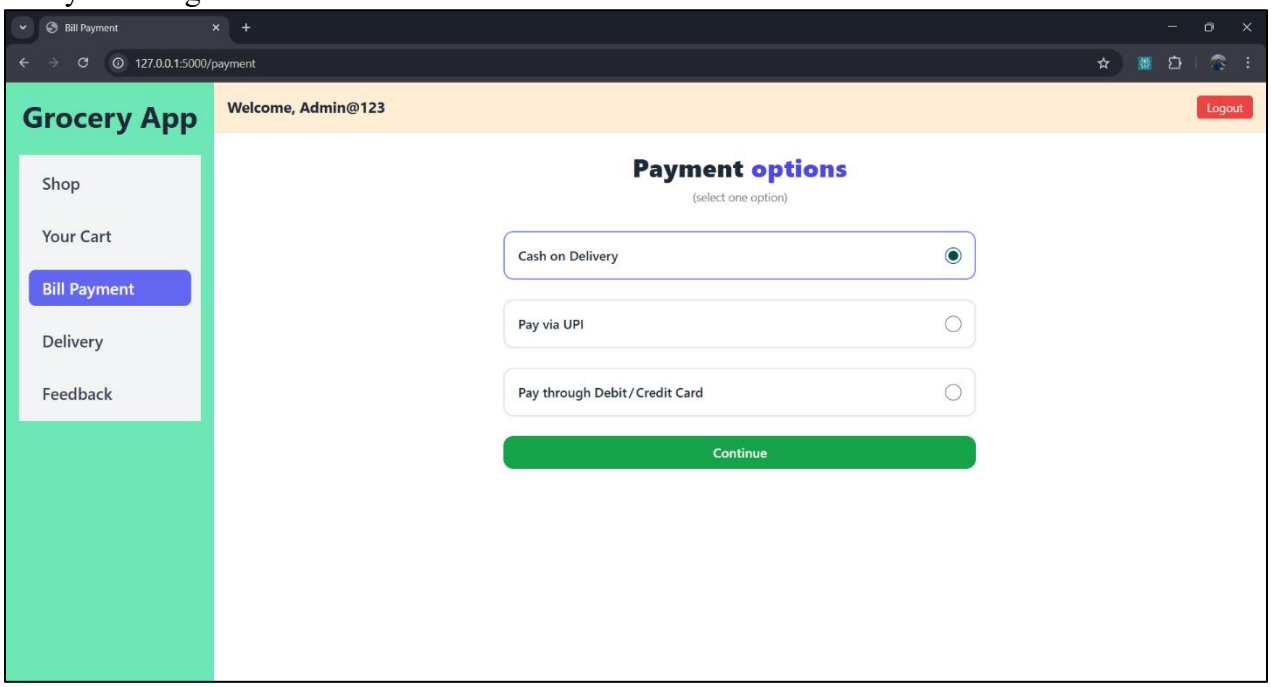
- Shop Page



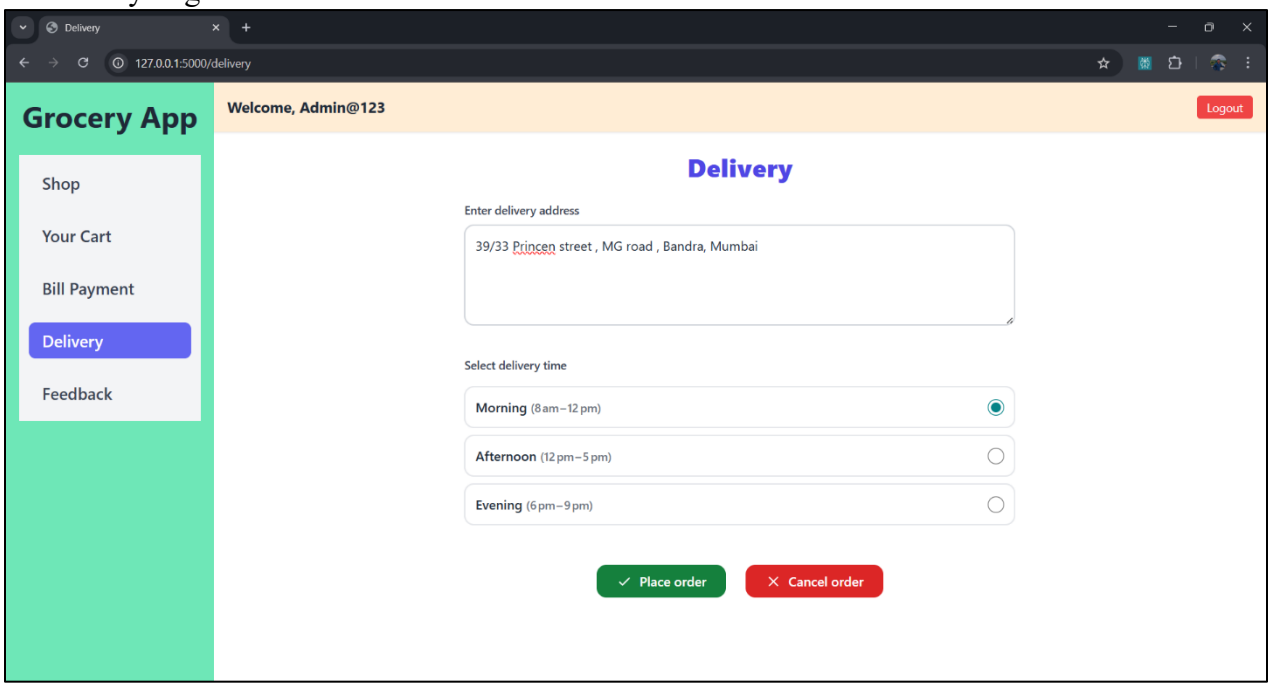
- Cart Page



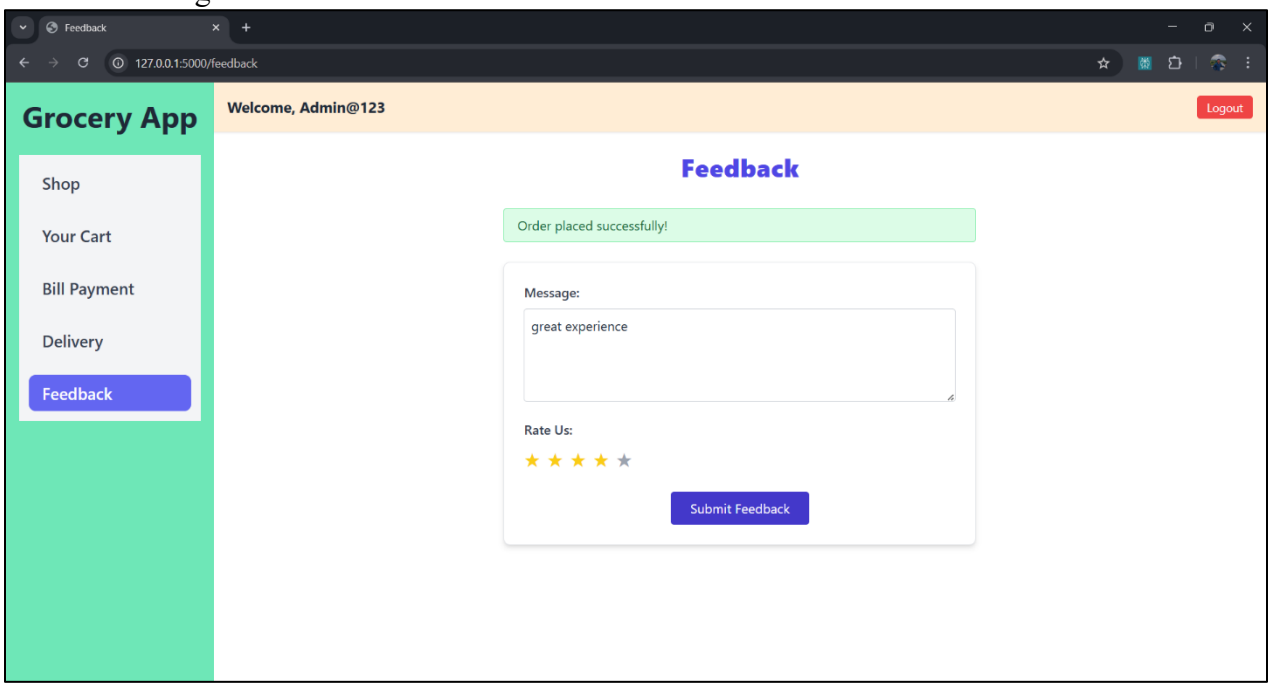
- Payment Page



- Delivery Page



- Feedback Page



6 Performance Test

This project was designed to simulate a real-world online grocery system. While academic in nature, the project considered practical constraints and testing scenarios to ensure the application could scale or evolve into a production-grade solution.

🔶 **Key Constraints Considered:**

Constraint	Design Handling
Session Handling	Used Flask's built-in session for tracking cart items across pages.
Data Storage	SQLite used for lightweight, fast access with ORM (SQLAlchemy) abstraction.
User Feedback	Flash messages implemented to ensure users get proper status after each action.
Form Validation	All forms include required fields and backend-side checks for robustness.
Page Responsiveness	Tailwind CSS used to keep the UI responsive across screen sizes.

Although the app wasn't stress-tested under load, the design allows future scaling via:

- Switching to PostgreSQL/MySQL for heavier DB loads.
- Adding login-based session management for better user tracking.

6.1 Test Plan/ Test Cases

Test Case ID	Feature	Action	Expected Result
TC01	Add to Cart	Add multiple items	Items are added and total updated
TC02	Remove from Cart	Remove a selected item	Item removed, total recalculated

Test Case ID	Feature	Action	Expected Result
TC03	Place Order	Fill address + select slot	Redirects to Feedback page with flash
TC04	Cancel Order	Click cancel	Cart cleared, redirected to shop
TC05	Submit Feedback	Rate and comment	Message shown, stays on same page

6.2 Test Procedure

1. Launch the app on localhost.
2. Login using test credentials.
3. Add and remove items from the cart.
4. Navigate through each page (shop → cart → payment → delivery).
5. Try valid and invalid inputs in forms (e.g., blank address).
6. Submit feedback multiple times with different ratings.
7. Observe flash messages and session behavior.

6.3 Performance Outcome

- All key functionalities worked without errors.
- Flash messages provided instant user feedback on actions.
- Forms handled empty or invalid inputs gracefully.
- Application flow was smooth and intuitive.
- No unexpected data loss or crashes occurred during testing.

7 My learnings

This internship has been a transformative learning experience for me. Over the course of six weeks, I gained deep insight into the process of designing, building, and deploying a full-stack web application using real-world technologies.

On the technical front, I learned how to:

- Structure a Flask-based application with clear routing and modular design.
- Work with relational databases using SQLAlchemy ORM and perform CRUD operations efficiently.
- Use **Tailwind CSS** to create responsive and accessible user interfaces.
- Manage user sessions, handle form submissions, and implement server-side validations.
- Implement flash messaging and page redirection to improve user experience.
- Design user flows and test them to ensure consistency and performance.

Apart from technical skills, I also learned how to:

- Break down a problem statement into phases and milestones.
- Translate client-side requirements into backend logic.
- Write clean, maintainable, and reusable code.
- Debug effectively and fix layout/data handling issues during testing.
- Document project flow, challenges, and outcomes professionally.

This experience gave me a taste of how industry-level software projects are planned and executed. It also improved my time management, focus, and confidence in working independently on complex systems.

I believe this internship has significantly contributed to my career development. The skills and experience I gained will help me pursue opportunities in **full-stack development**, **backend engineering**, or even **product prototyping** roles in the tech industry.

I now feel more equipped to contribute meaningfully to real-world software teams and tackle larger, more challenging projects in the future.

8 Future work scope

While the core features of the Grocery Delivery Web App were successfully implemented during the internship, there are several areas that can be enhanced or expanded in future phases:

- **User Registration and Authentication:**
  Currently, login is handled with a test user. Adding full user account management (signup, password reset, profiles) would improve personalization and security.
- **Real Payment Gateway Integration:**
  Integration of secure payment services like Razorpay, Stripe, or UPI APIs would make the app transaction-ready for production use.
- **Order History and Tracking:**
  Users could be allowed to view past orders, track delivery status, and reorder from previous purchases.
- **Admin Panel Enhancements:**
  A full-featured dashboard for administrators could allow them to manage inventory, view orders, analyze customer feedback, and generate reports.
- **Recommendation Engine:**
  Implementing a basic ML-based recommendation system could help suggest frequently bought items or show personalized deals.
- **Deployment to Cloud Platform:**
  The app can be containerized and deployed to platforms like Heroku, Render, or AWS to make it accessible online.
- **Mobile-Responsive Layout or PWA Support:**
  Making the app mobile-first or a progressive web app (PWA) would improve usability across devices.

These enhancements would not only add more value to the application but also make it production-ready for real businesses.