# Project Charter: Customizable eCommerce Website Builder

## 1. Project Purpose and Objectives

The objective of this project is to build a multi-shop, customizable eCommerce platform that allows businesses of any domain (e.g., clothing, electronics, groceries) to create their own online store using a single adaptable template. The system will include role-based login (Admin and Customer), product inventory management, and a demo payment gateway integration using Stripe. The ultimate goal is to make online selling accessible to small businesses with no technical expertise.

## Mind Map: Concept Overview



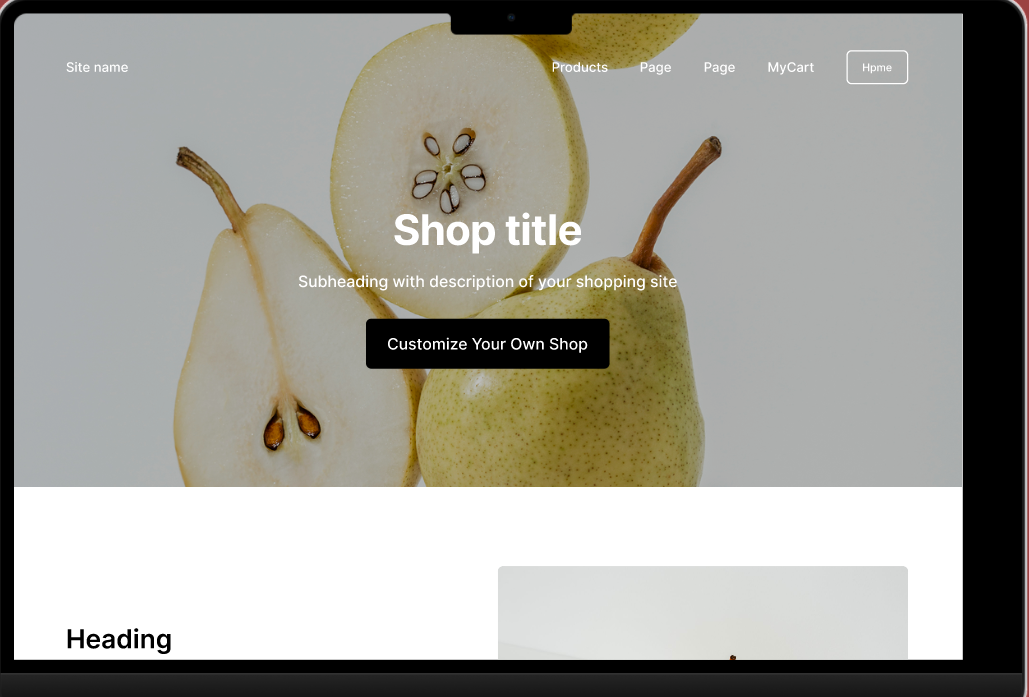
## 2. Justification and Market Demand

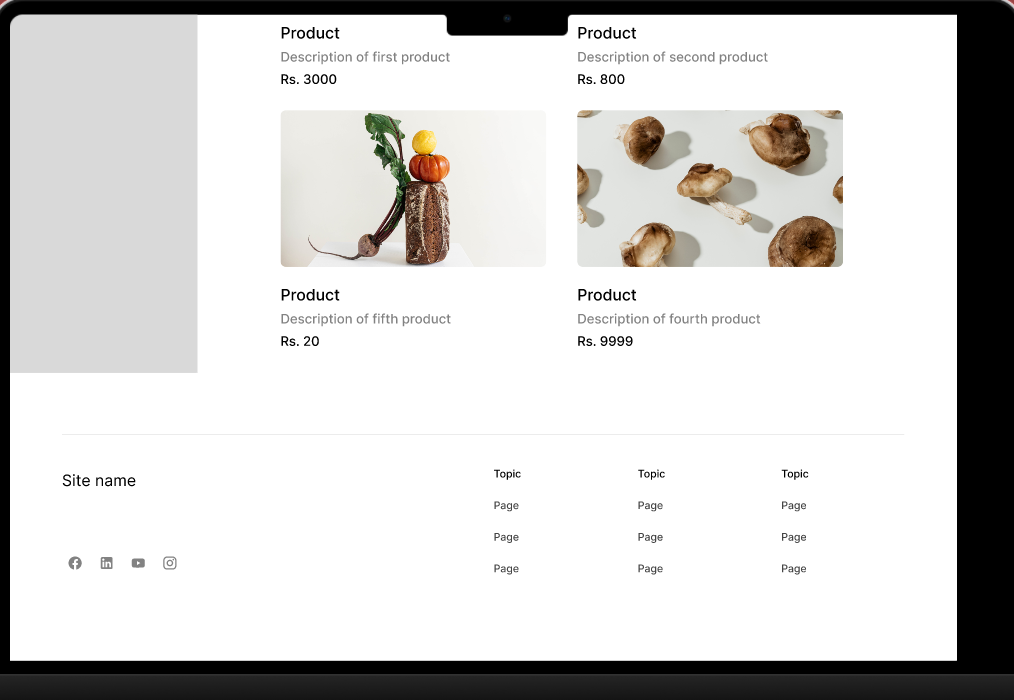
With the exponential rise in online shopping and the increasing digitalization of small businesses, the demand for easy-to-use, affordable online store builders is stronger than ever. Platforms like Shopify and Wix dominate the market, but many small business owners find them complex or expensive. Our platform targets this gap by offering a simplified, ready-to-use solution tailored to micro and small businesses.

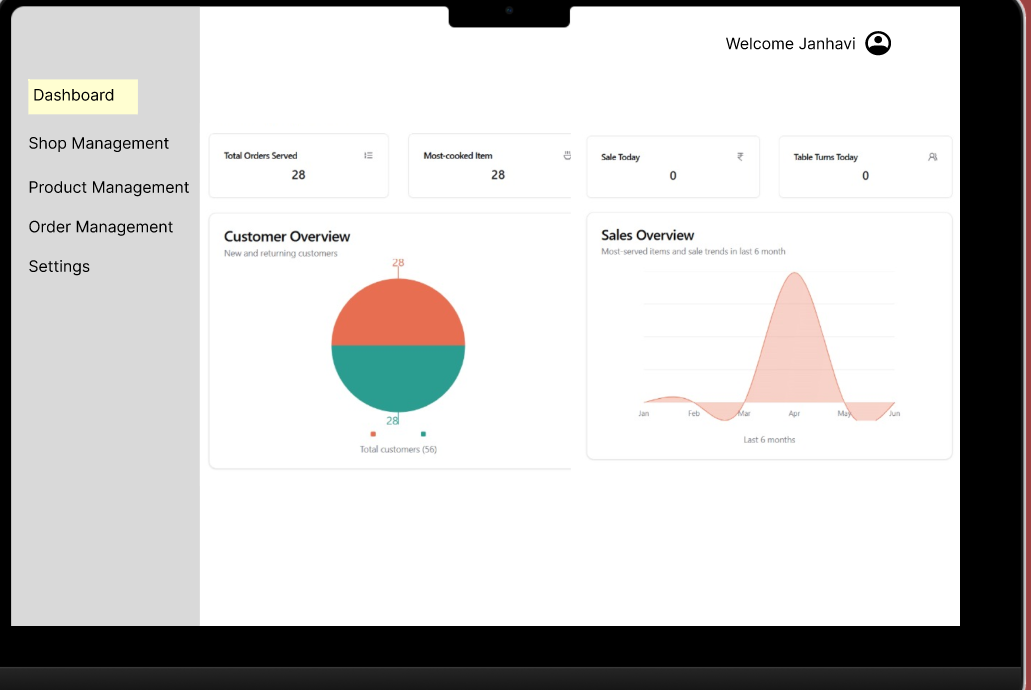
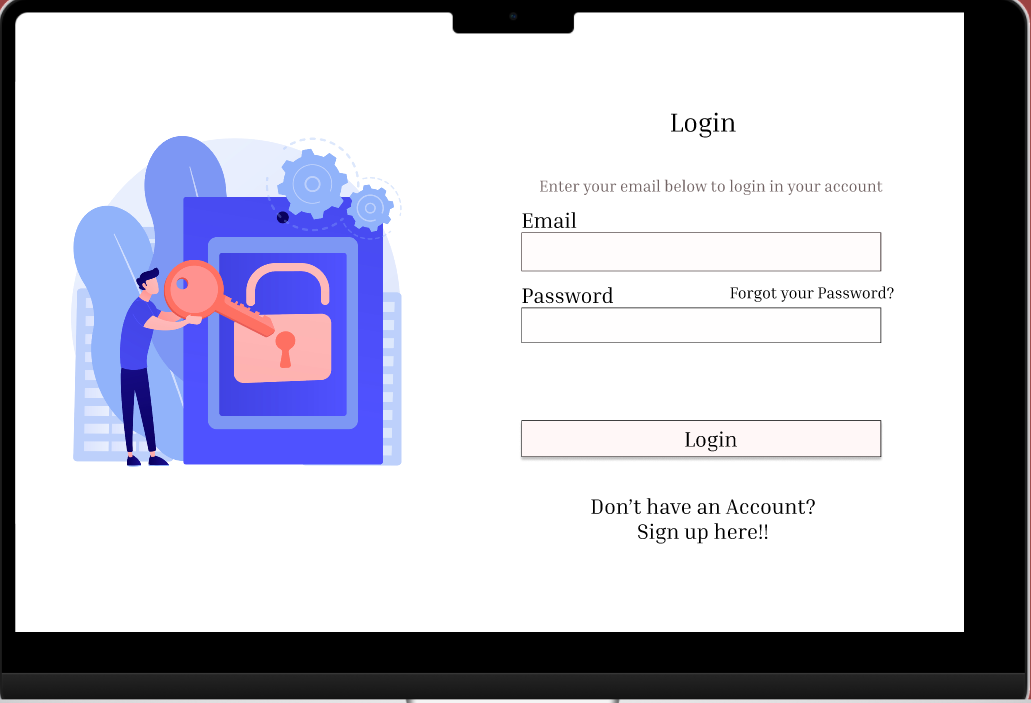
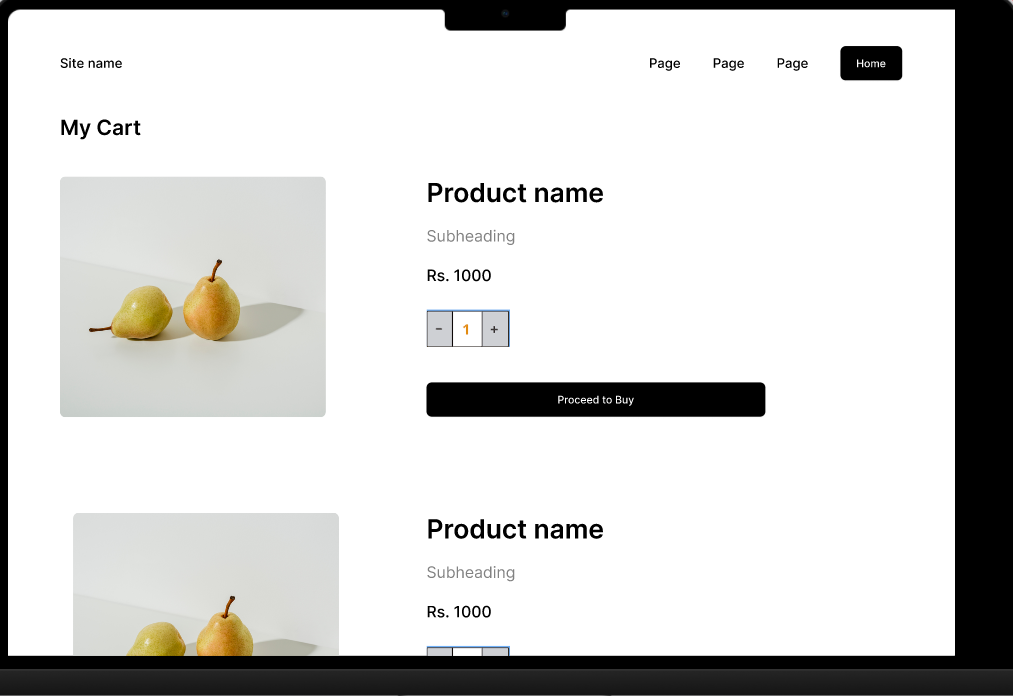
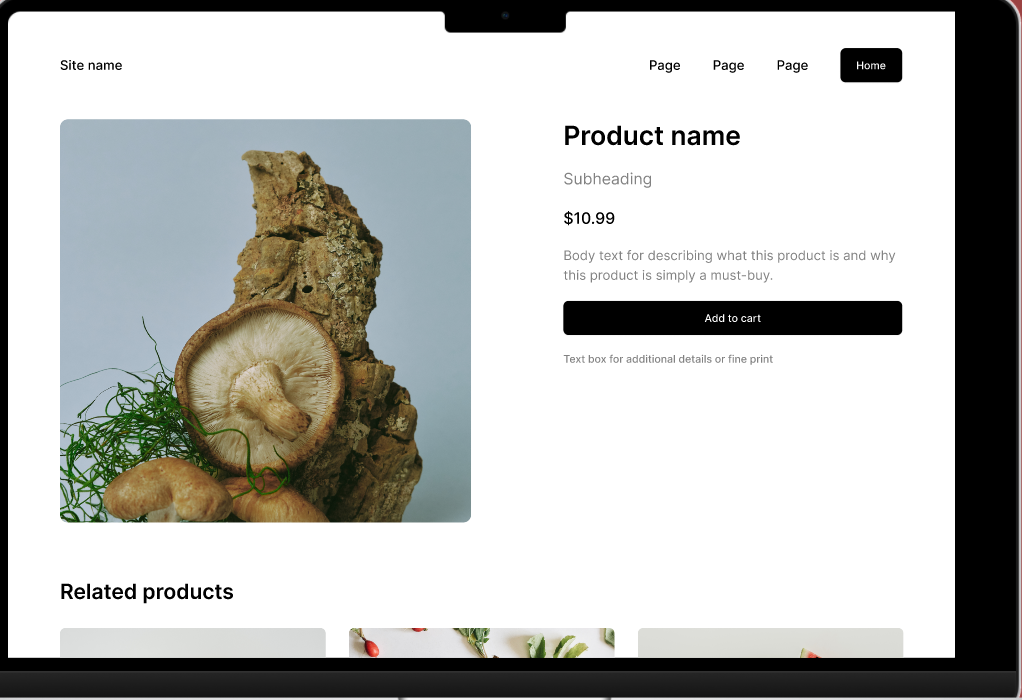
## 3. Business Requirements and Feature Scope

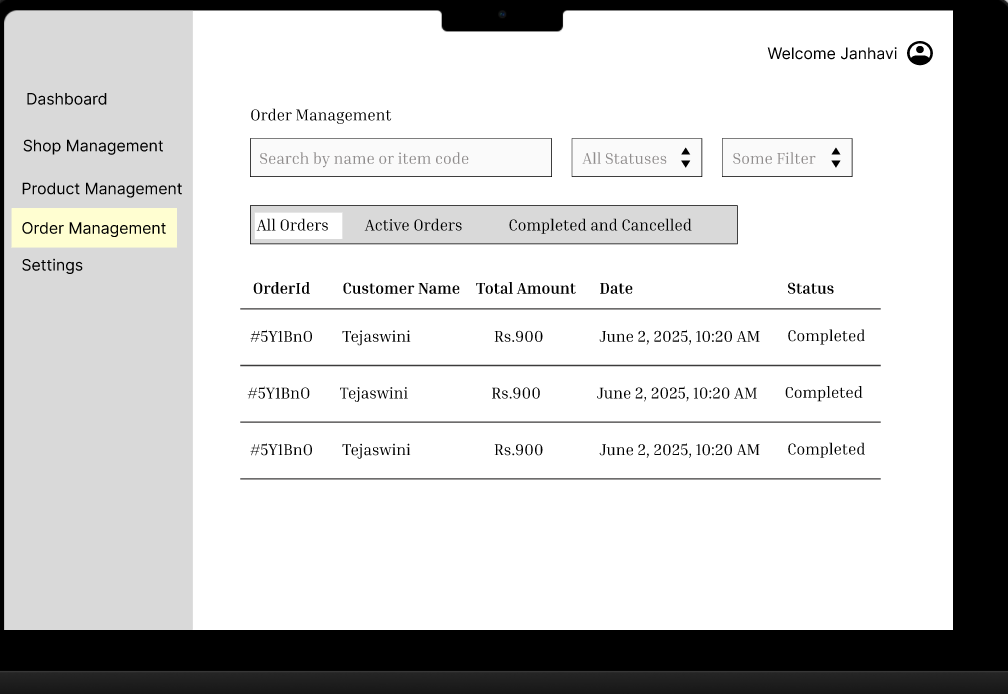
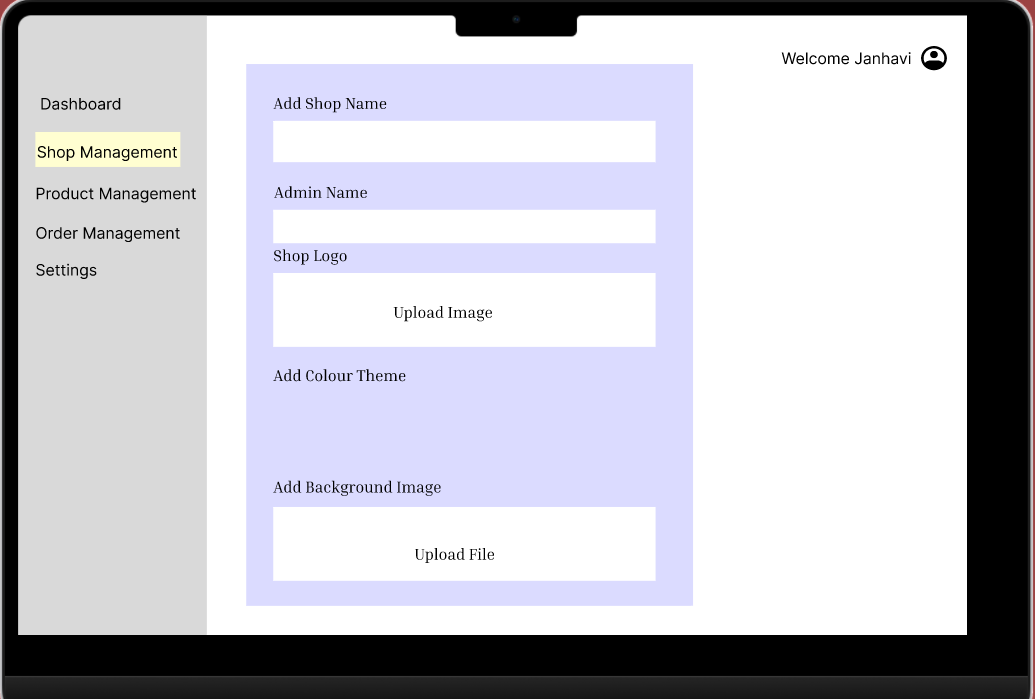
Core Features:  
- Email OTP login system for authentication  
- Role-based access: Admin (shop owner) and Customer (end user)  
- Shop customization (name, logo, color theme)  
- Product inventory management (add/edit/delete products)  
- Customer management  
- Product browsing, add to cart, and checkout for customers  
- Stripe integration for demo payment  
  
Admin Panel Capabilities:  
- Manage shop branding and layout  
- CRUD operations for products  
- View and manage customer details  
  
Customer View:  
- Browse products  
- Add items to cart  
- Checkout with demo payment system

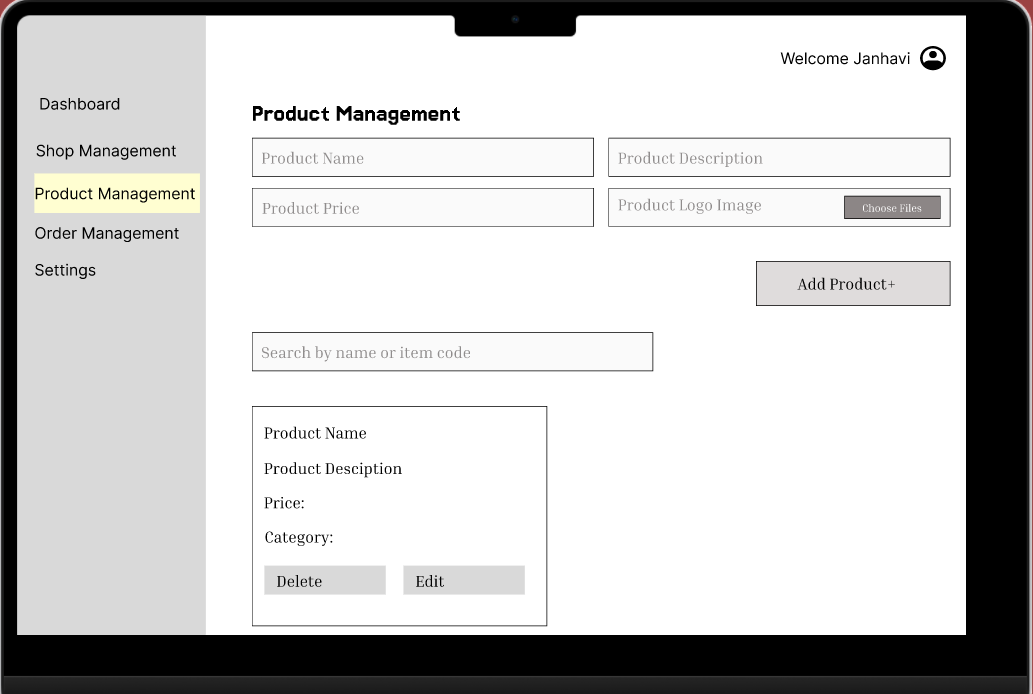
## UI Wireframes: Sample Pages







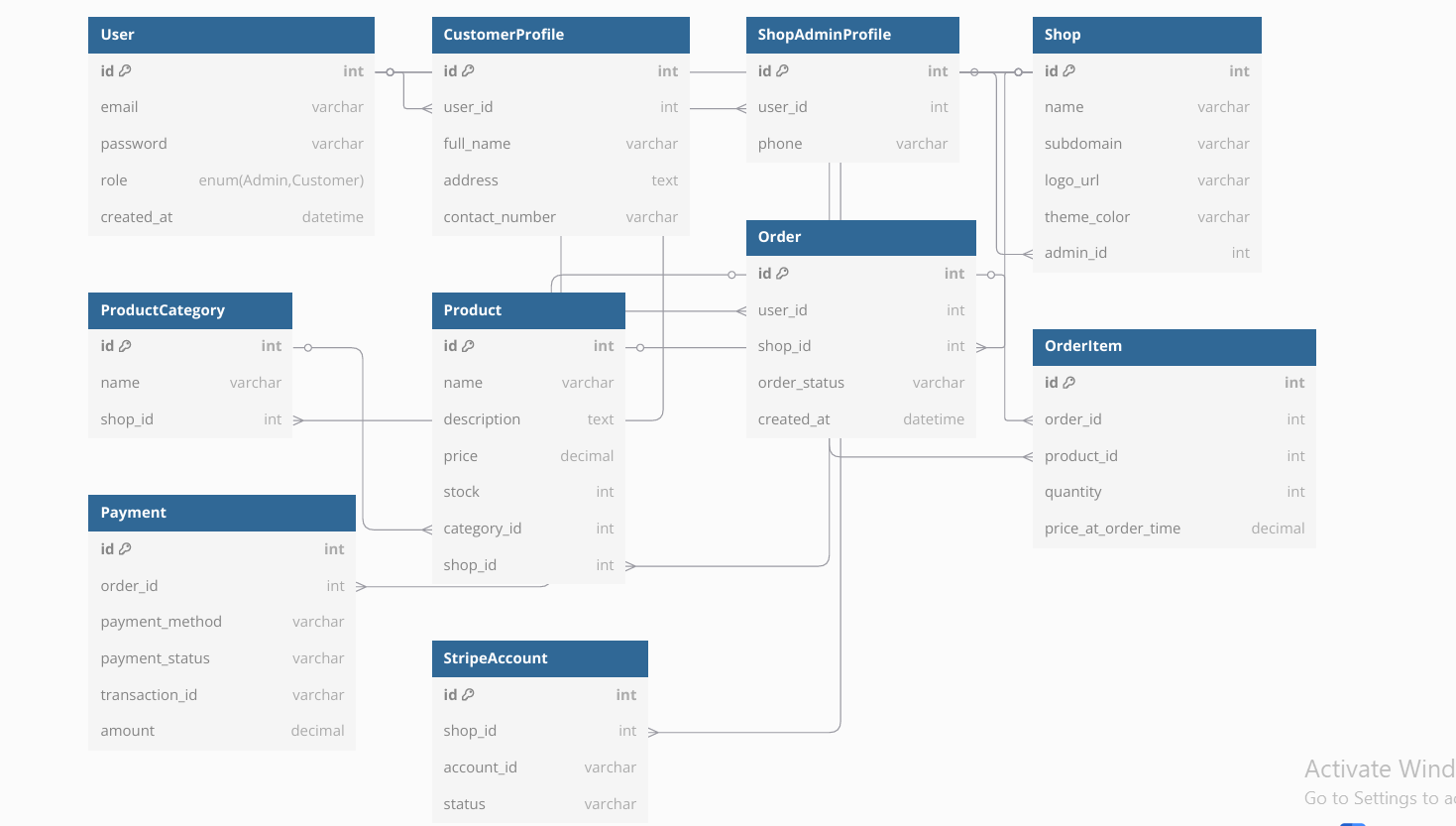


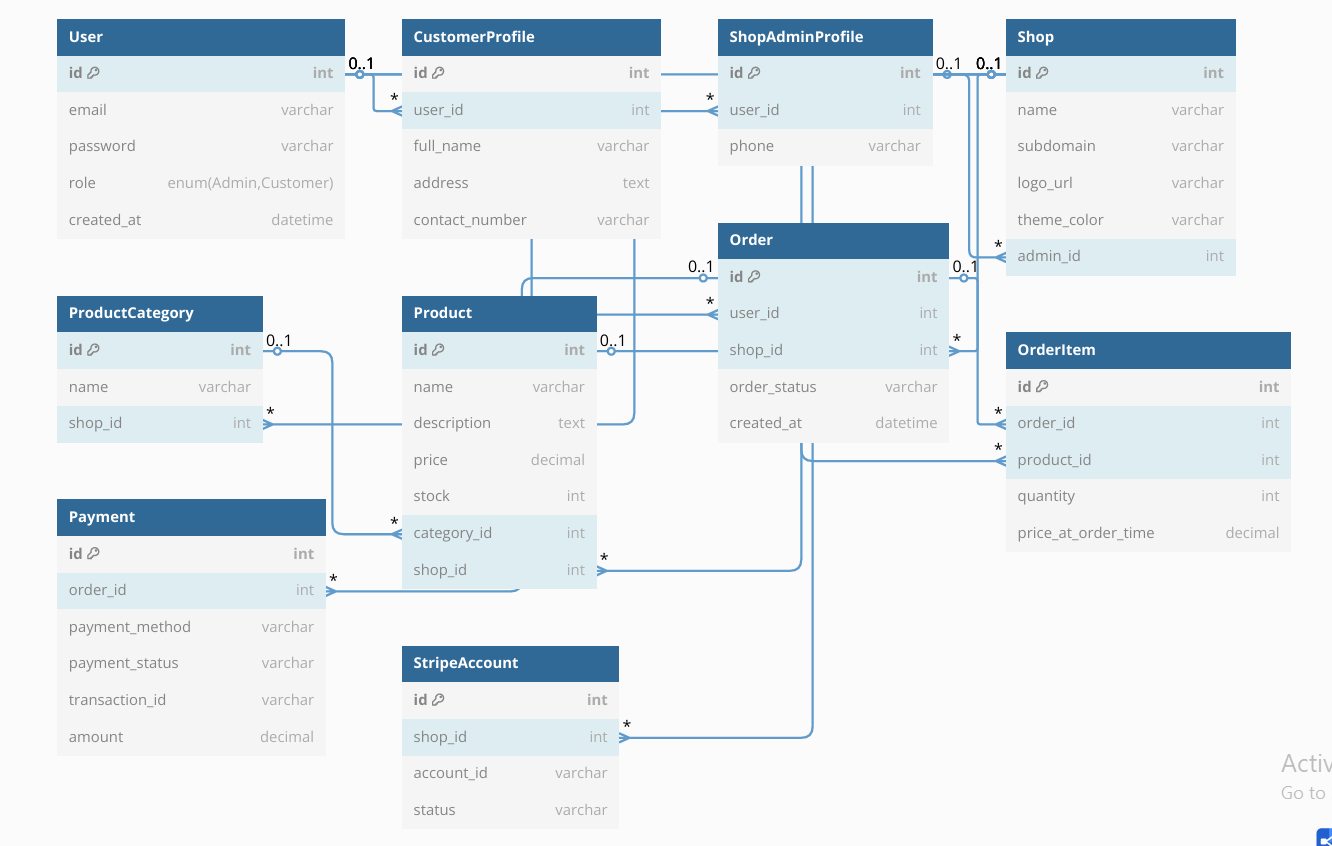


## 4. Technical Requirements

Frontend:  
- React.js  
- JavaScript  
- HTML/CSS  
  
Backend:  
- Node.js  
- Express.js  
  
Database:  
- MongoDB  
  
Authentication:  
- Email-based OTP login  
- Role-based access control  
  
Payment Gateway:  
- Stripe (test/demo mode)  
  
Hosting:  
- To be Finalized

## Entity-Relationship Diagram (Database Design)





## 5. Key Stakeholders

|  |  |
| --- | --- |
| Stakeholder | Role |
| Project Team  (4 Members) | Design, development, integration  & testing |
| End Users | Shop owners (Admins) and Customers |
| Reviewers/Faculty | Project evaluation and guidance |
| Payment Gateway (Stripe) | Demo integration partner (in test mode) |

## 6. Estimated Real-Life Cost

|  |  |
| --- | --- |
| Item | Cost |
| Web Hosting (annual) | ₹4,000 – ₹8,000 |
| Domain name | ₹800 – ₹1,200 |
| SMTP/OTP Email Service | ₹1,500 – ₹3,000/month |
| Stripe Integration (Live use) | 2.9% + ₹3 per transaction |
| Developer Cost (optional) | ₹50,000+ (for a dev team) |

## 7. Resources and Technologies

|  |  |
| --- | --- |
| Resource | Description |
| React.js | Frontend UI framework |
| Node.js | Backend JavaScript runtime |
| Express.js | Backend framework for REST APIs |
| MongoDB | NoSql Database system |
| Git + GitHub | Version control and collaboration |
| Stripe | Demo payment gateway |
| Nodemailer | For sending OTP emails (optional) |
| To be Finalized | Hosting platforms |

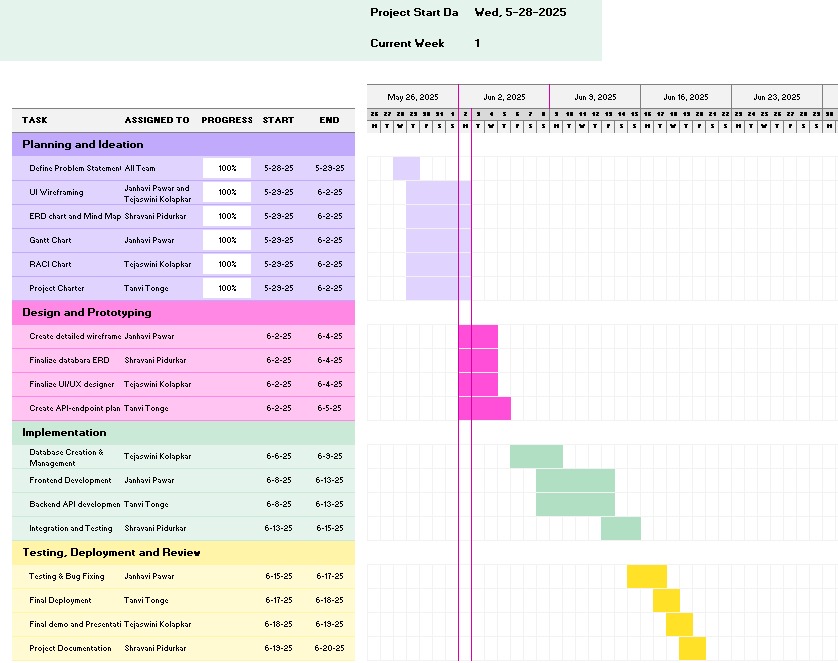
## 8. Risk Assessment

|  |  |
| --- | --- |
| Risk | Mitigation Strategy |
| Time constraints | Prioritize core features, limit scope per review |
| OTP delivery delays | Use test OTP/email library or mock in review |
| Stripe test issues | Integrate using official test cards and keys |
| Team coordination | Use Git or Github to track task ownership |
| Template overload | Stick to 1 reusable template for all shops |

## 9. Timeline & Time Management

|  |  |
| --- | --- |
| Date | Task |
| May 28–May 29 | Define Problem Statement |
| May 29- June 2 | UI Wireframing, ERD Chart, Mind Map, Gantt Chart, RACI Chart, Project Charter |
| June 2- June 5 | Create detailed wireframe, Finalize ERD, Finalize UI/UX designer and Create API-endpoint plan |
| June 6–9 | Database creation & management |
| June 8–13 | Frontend and Backend API development |
| June 13-15 | Integration and testing |
| June 15-20 | Testing & bug fixing, Final deployment, Final demo and Project Documentation |

## Gantt Chart: Project Timeline Overview



## 10. RACI Matrix

