**1.**

A Project Report On

**BizBook**

Submitted in partial fulfillment of the requirement for the award of the degree

Master of Computer Applications (MCA)

Academic Year 2025 – 26

**Krupal Mehta 92400584065**

**Shyam Humbal 92400584084**

**Divyesh Gondaliya 92400584068**

**Internal Guide**

Dr. Jaypalsinh Gohil



Rajkot-Morbi Road, At &PO :Gauridad, Rajkot 360 003. Gujarat. India.

A certificate of a college degree

AI-generated content may be incorrect.

–

# DECLARATION

We hereby declare that this project work entitled **BizBook** is a record done by us.

We also declare that the matter embodied in this project is genuine work done by us and has not been submitted whether to this University or to any other University / Institute for the fulfillment of the requirement of any course of study.

Place:

Date:

Krupal Mehta (92400584065) Signature: Shyam Humbal (92400584084) Signature: Divyesh Gondaliya (92400584068) Signature:

# ACKNOWLEDGEMENT

It is indeed a great pleasure to express our thanks and gratitude to all those who helped us. No serious and lasting achievement or success one can ever achieve without the help of friendly guidance and co-operation of so many people involved in the work.

We are very thankful to our guide **Dr.Jaypalsinh Gohil** the person who makes us to follow the right steps during our project work. We express our deep sense of gratitude to for his /her guidance, suggestions and expertise at every stage. A part from that his/her valuable and expertise suggestion during documentation of our report indeed help us a lot.

Thanks to our friend and colleague who have been a source of inspiration and motivation that helped to us during our project work.

We are heartily thankful to the Dean of our department **Dr. R. Sridaran** sir and HoD **Dr. Sunil Bajeja** sir for giving us an opportunity to work over this project and for their end-less and great support to all other people who directly or indirectly supported and help us to fulfil our task.

Krupal Mehta (92400584065) Signature: Shyam Humbal (92400584084) Signature: Divyesh Gondaliya (92400584068) Signature:

|  |  |  |
| --- | --- | --- |
| **Chapters** | **Particulars** | **Page No.** |
| **1** | **Introduction to Project Definition** | 1 |
| **2**  2.1 | **PREAMBLE**  Module description | 2 |
| **3** | **REVIEW OF LITERATURE**  Analyze, Study and Compare the similar types of the systems/applications and highlights major findings | 4 |
| **4** | **TECHNICAL DESCRIPTION** |  |
| 4.1 | Hardware Requirement | 5 |
| 4.2 | Software Requirement |  |
| **5** | **SYSTEM DESIGN AND DEVELOPMENT** |  |
|  | Flow Chart | 6 |
| 5.1 | Data Flow Diagram |  |
| 5.2 | Class Diagram |  |
| 5.3 | Use Case Diagram |  |
| 5.4 | Sequential Diagram |  |
| 5.5 | Activity Diagram |  |
| 5.6 | Database Design / File Structure |  |
|  | Menu Design |  |
|  | Screen Design |  |
| **6** | **SYSTEM TESTING** | 26 |
| **7** | **CONCLUSION** | 32 |
| **8**  8.1 | **LEARNING DURING PROJECT WORK**  **Future Enhancement** | 33 |
| **9**  9.1  9.2 | **BIBLIOGRAPHY**  Online References Offline References | 34 |

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Title** | **Page No.** |
| Table 4.1 | Hardware Requirement | 5 |
| Table 4.2 | Software Requirement | 5 |
| Table 5.1 | User Collection | 14 |
| Table 5.2 | Business Collection | 14 |
| Table 5.3 | Category Collection | 14 |
| Table 5.4 | Product/Service Collection | 15 |
| Table 5.5 | Booking Collection | 15 |
| Table 6.1 | Authentication (Client) | 26 |
| Table 6.2 | Business Management(Client) | 27 |
| Table 6.3 | Product/Service Management(Client) | 27 |
| Table 6.4 | Booking System(Client) | 28 |
| Table 6.5 | User Booking History(Client) | 29 |
| Table 6.6 | Authentication (Admin) | 29 |
| Table 6.7 | Business Management (Admin) | 30 |
| Table 6.8 | Product/Service Management (Admin) | 31 |
| Table 6.9 | Admin Dashboard Testing | 32 |

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Title** | **Page No.** |
| Figure 5.1 | Flow Chart – Admin | 6 |
| Figure 5.2 | Flow Chart – Client | 7 |
| Figure 5.3 | Data Flow Diagram – Level 0 | 8 |
| Figure 5.4 | Data Flow Diagram – Level 1 | 8 |
| Figure 5.5 | Class Diagram | 9 |
| Figure 5.6 | Use Case Diagram | 10 |
| Figure 5.7 | Sequential Diagram | 11 |
| Figure 5.8 | Activity Diagram – Client | 12 |
| Figure 5.9 | Activity Diagram – Admin | 13 |
| Figure 5.10 | Business List | 16 |
| Figure 5.11 | Business Product | 16 |
| Figure 5.12 | Business Details and Book Service | 17 |
| Figure 5.13 | My Bookings | 17 |
| Figure 5.14 | BizBook Dashboard | 18 |
| Figure 5.15 | Business Table | 18 |
| Figure 5.16 | Select Product by Business | 19 |
| Figure 5.17 | Admin Login | 19 |
| Figure 5.18 | Admin Home Page | 20 |
| Figure 5.19 | Admin Manage User | 20 |
| Figure 5.20 | Manage Business List | 21 |
| Figure 5.21 | Edit Business | 21 |
| Figure 5.22 | Product List | 22 |
| Figure 5.23 | Add New Product | 22 |
| Figure 5.24 | Manage Booking | 23 |

1. **BizBook**

**Introduction:**

* **Project Title**: BizBook (Vocal for Local)
* **Technology Stack**: MERN (MongoDB, Express.js, React.js, Node.js)
* **Type**: Web-based Appointment & Order Booking Platform
* **Client**: Public users (customers) + Admin Panel (for business management)
* **Domain**: Local Business Service Platform

**Synopsis:**

**BizBook** (also called **Local for Vocal**) is a full-stack web application built using the **MERN stack** that helps local businesses connect with users digitally. The platform allows users to discover nearby businesses like salons, restaurants, cafes, or retail stores and make **appointments, table bookings, or product orders** directly through the app.

Admins can manage businesses, products/services, and categories through a secure backend panel. The system smartly handles different types of businesses based on category logic – such as appointments for salons, table bookings for restaurants, and product orders for retail shops.

**Project Description:**

In today's digital age, many local businesses still rely on walk-in customers or phone calls. BizBook helps modernize that by creating a **smart online presence** for these businesses. It acts like a mini local version of Google Maps + UrbanClap + Zomato, where users can:

* Browse verified businesses
* View their services or products
* Make appointments or place orders
* Track their bookings

Each business belongs to a category such as:

* **Salon / Doctor** – takes **Appointments**
* **Restaurant / Café** – supports **Table Bookings**
* **Retail / Grocery** – handles **Product Orders** (without delivery for now)

1. **PREAMBLE**
   1. **Module Description:**
2. **User Module**
   * **Register/Login** with JWT-based session
   * View all businesses
   * View business details (services/products)
   * Book appointment, table, or order based on category
   * View all their bookings
   * Frontend built in React with axios, routing, and session logic
3. **Admin Module**
   * Secure login for admin (handled on EJS panel)
   * Dashboard view: total businesses, services, appointments
   * Add/manage businesses
   * Add/manage products or services for each business
   * View and manage user appointments
   * Admin routes protected and organized using Express.js
4. **Business Module**
   * Add new business (via Admin)
   * Each business has:
     + Name, Description, Category
     + Address, Phone, Map link, Social Media
   * Category determines how booking works (appointment/order/table)
   * Businesses are shown with clean cards on frontend
5. **Product/Service Module**
   * Each product/service is **linked to a business**
   * Added via admin panel
   * Shown in detail on business page
   * Used during appointment or product order flow
   * Supports price, description, and image fields
6. **Booking/Appointment Module**
   * Users can make appointments/orders/bookings
   * Logic based on business category:
     + **Salon** = time-based appointment
     + **Restaurant** = date + time + number of people (table booking)
     + **Retail** = simple product order (no delivery)
   * Stored in MongoDB with relation to User & Business
   * Admin can view all bookings in panel
7. **Category Module**
   * Predefined business types:
     + Salon, Restaurant, Café, Grocery, Doctor, etc.
   * Each category has a **booking type**:
     + Appointment, Table, or Order
   * Used to drive logic in both backend & frontend
   * Admin can fetch categories for dropdowns
8. **REVIEW OF LITERATURE**

To better understand and position BizBook, a study was conducted on similar platforms that allow users to discover and book local services or products. The following systems were reviewed:

* 1. **UrbanClap**

Functionality: Offers appointment-based services such as salon, repair, cleaning, and more.

Technology: Uses a mobile-first approach with robust backend services.

Learning: Emphasis on category-specific booking flows and service provider profiles.

* 1. **Justdial**

Functionality: Directory-style listing of businesses with user reviews, contact info, and basic inquiry features.

Learning: Strong search engine and categorization helped understand listing dynamics.

* 1. **Zomato**

Functionality: Focused on restaurants with menu display, booking table or ordering food.

Learning: Product-based and table booking logic were well implemented, influencing BizBook's dual booking types.

* 1. **Booksy**

Functionality: Salon and wellness booking app with time slots and user profiles. Learning: Gave insights into appointment scheduling UX and availability logic.

Major Findings

A successful system integrates category-wise logic for booking: appointment, table, or product order.

Admin control over business listings and services is essential for data integrity.

Real-time appointment availability, feedback, and booking confirmation mechanisms are essential features.

1. **TECHNICAL DESCRIPTION**
   1. **Hardware Requirements:**

|  |  |
| --- | --- |
| **Component** | **Specification** |
| **Processor** | Intel Core i5 or above |
| **RAM** | Minimum 8 GB |
| **Hard Disk** | Minimum 256 GB SSD or 500 GB HDD |
| **Display** | 14" or higher, 1366x768 resolution or more |
| **Network** | Internet connection (for API/database) |

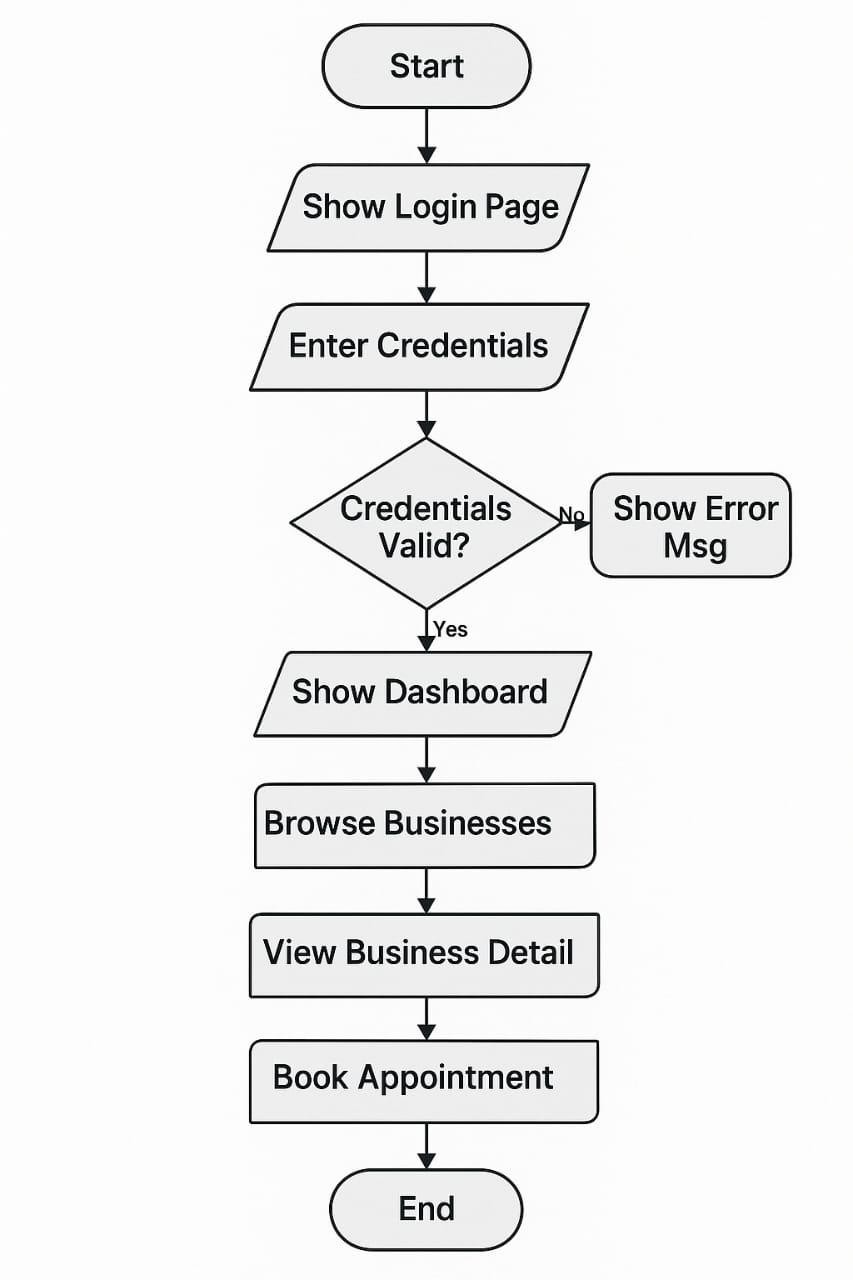
**Table 4.1 Hardware Requirements**

* 1. **Software Requirements:**

|  |  |
| --- | --- |
| **Frontend** | **React.js, Axios, Tailwind CSS** |
| **Backend** | Node.js, Express.js |
| **Database** | MongoDB |
| **Admin Panel** | EJS Templates with Express.js |
| **Authentication** | JWT |
| **Operating System** | Windows 10 / 11 or Linux (Ubuntu) |
| **Version Control** | Git & GitHub |
| **Development Tools** | VS Code, Postman, MongoDB Compass |
| **Hosting** | Render / Vercel / Netlify (Frontend), Render |

**Table 4.2 Software Requirements**

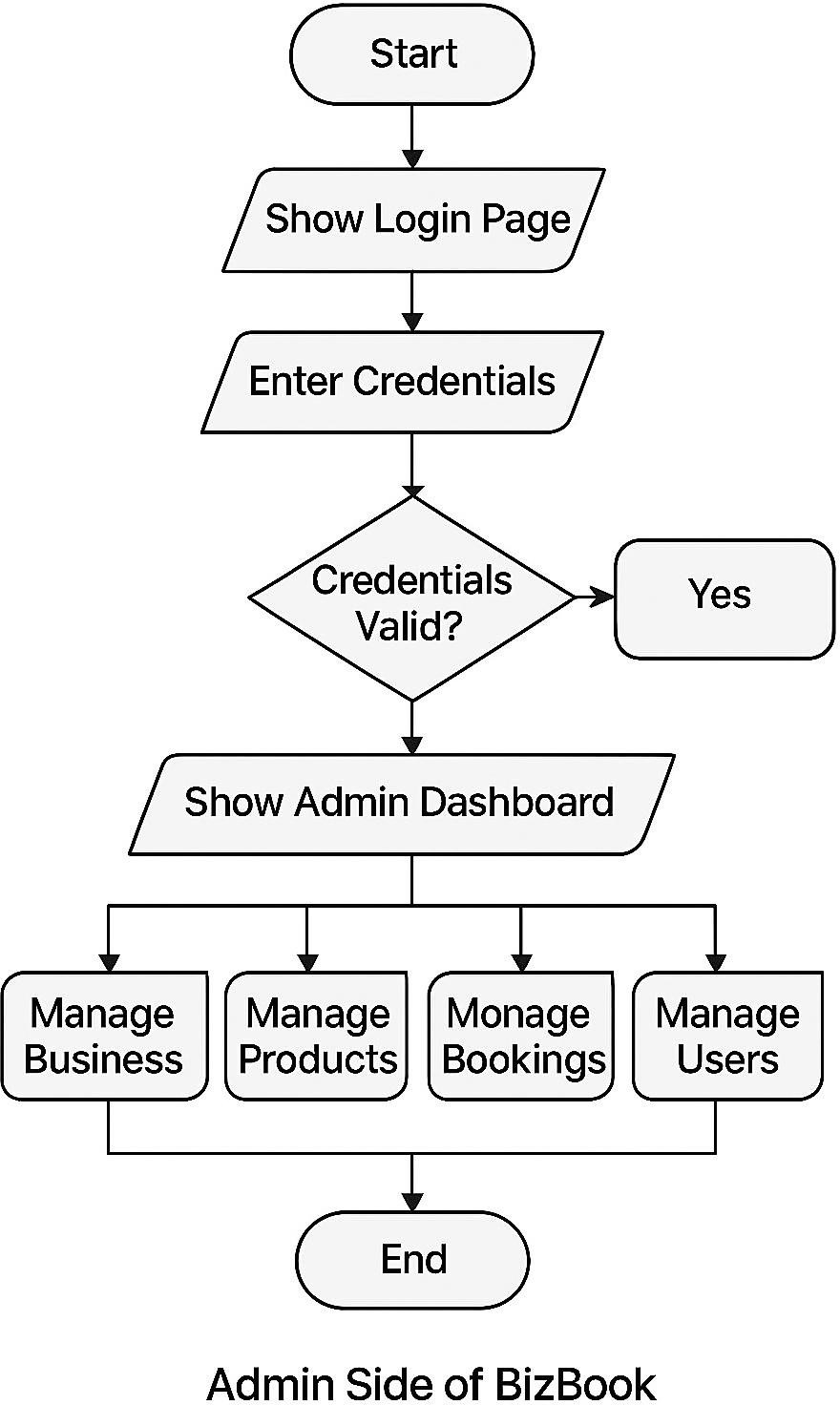
1. **SYSTEM DESIGN AND DEVELOPMENT**



**Diagrams:**

**Flow Chart Diagram: Client**

**Figure 5.1 Flow Chart Diagram-Client**



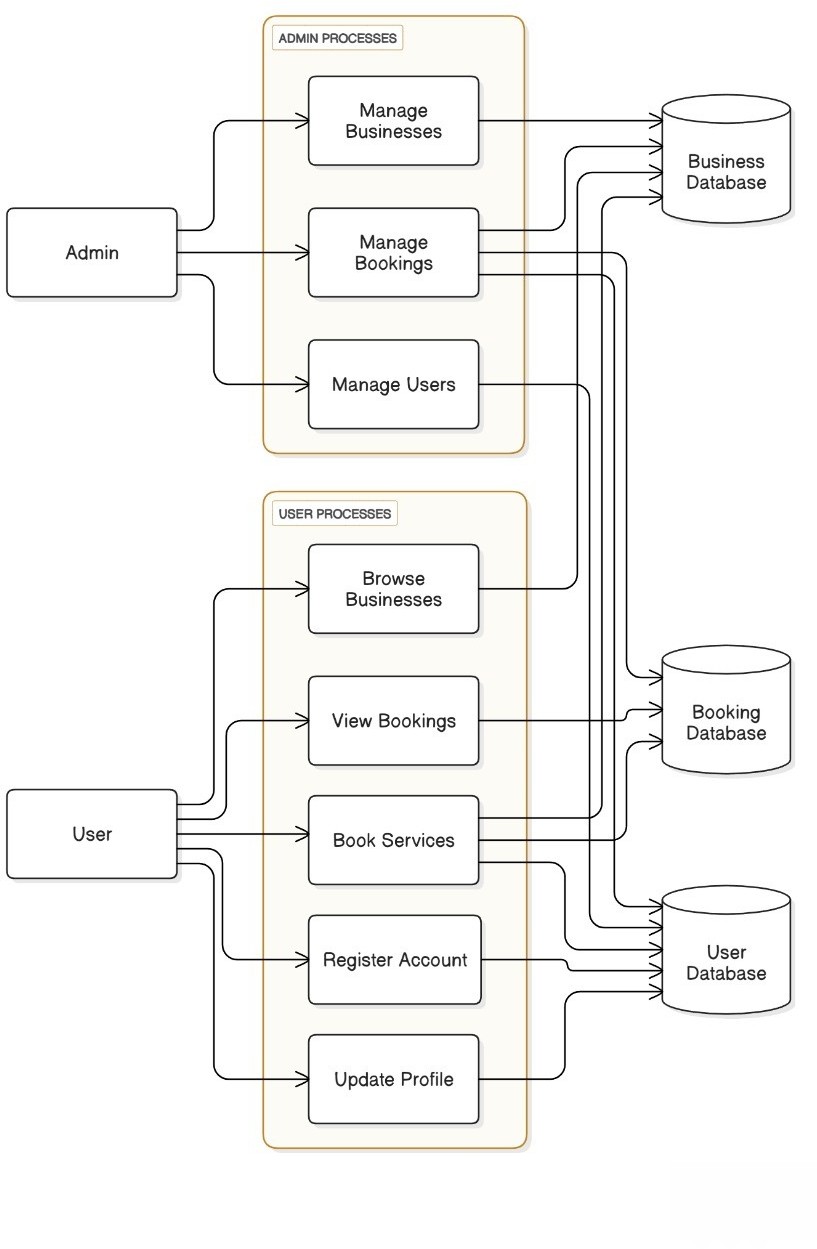
**Flow Chart : Admin**

Error

Yes

**Figure 5.2 Flow Chart Diagram-Admin**

**DFD :**



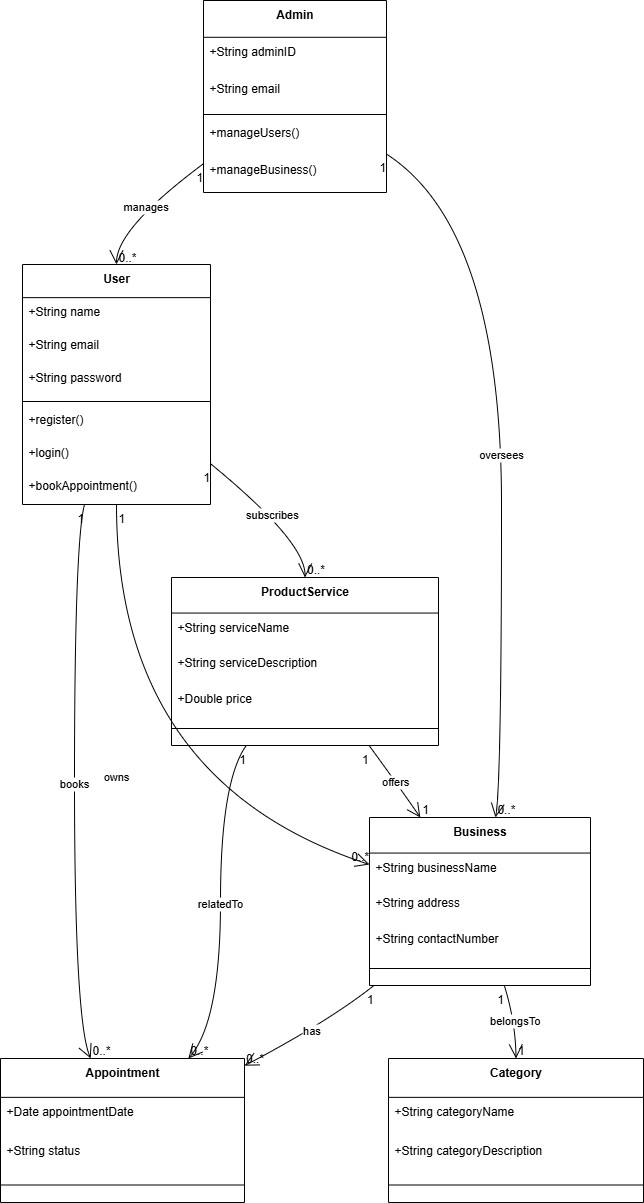
**Level 0:**

**Figure 5.3 Data Flow Diagram – Level 0**

**Level 1 :**

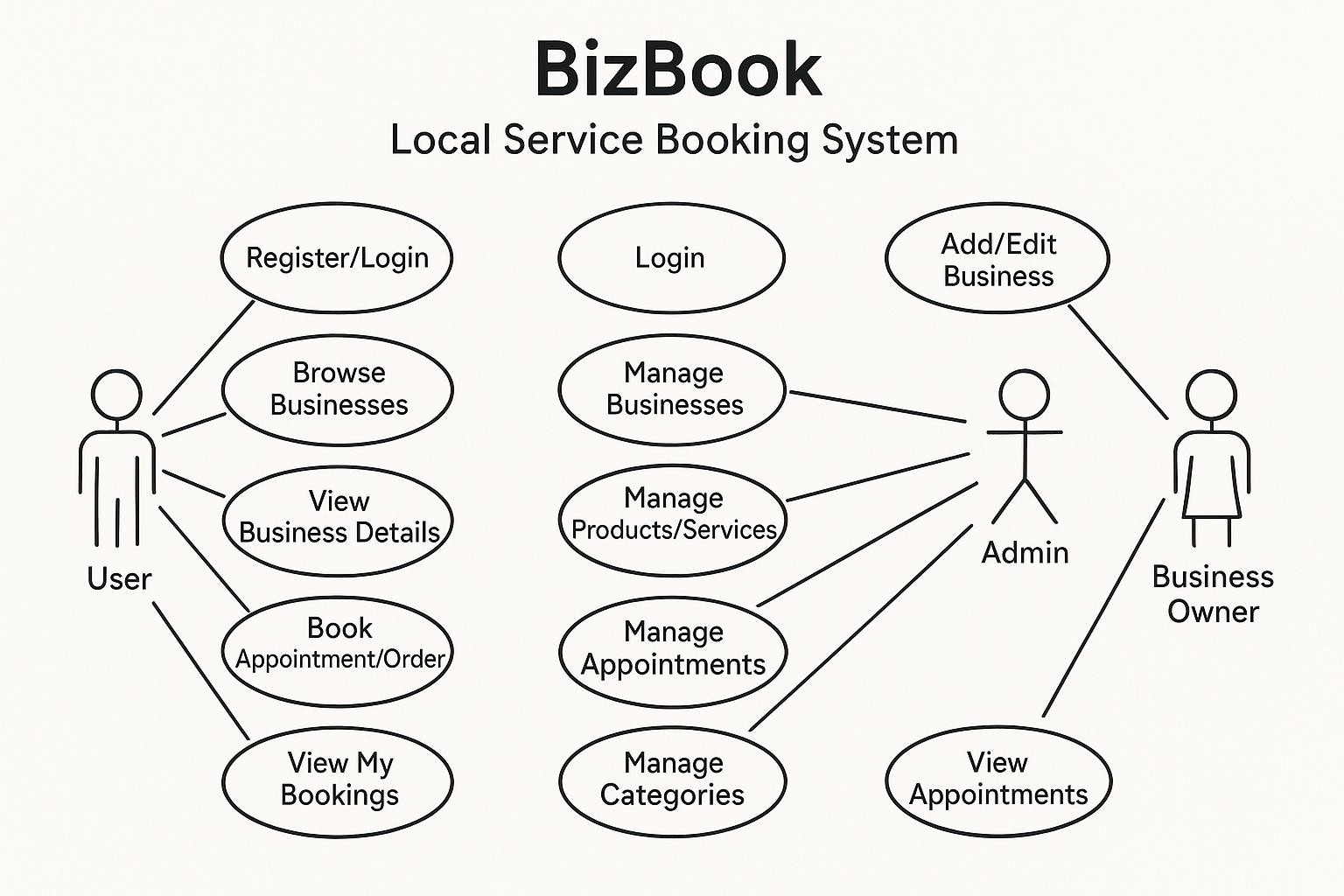
**Figure 5.4 Data Flow Diagram – Level 1**

**Class Diagram:**



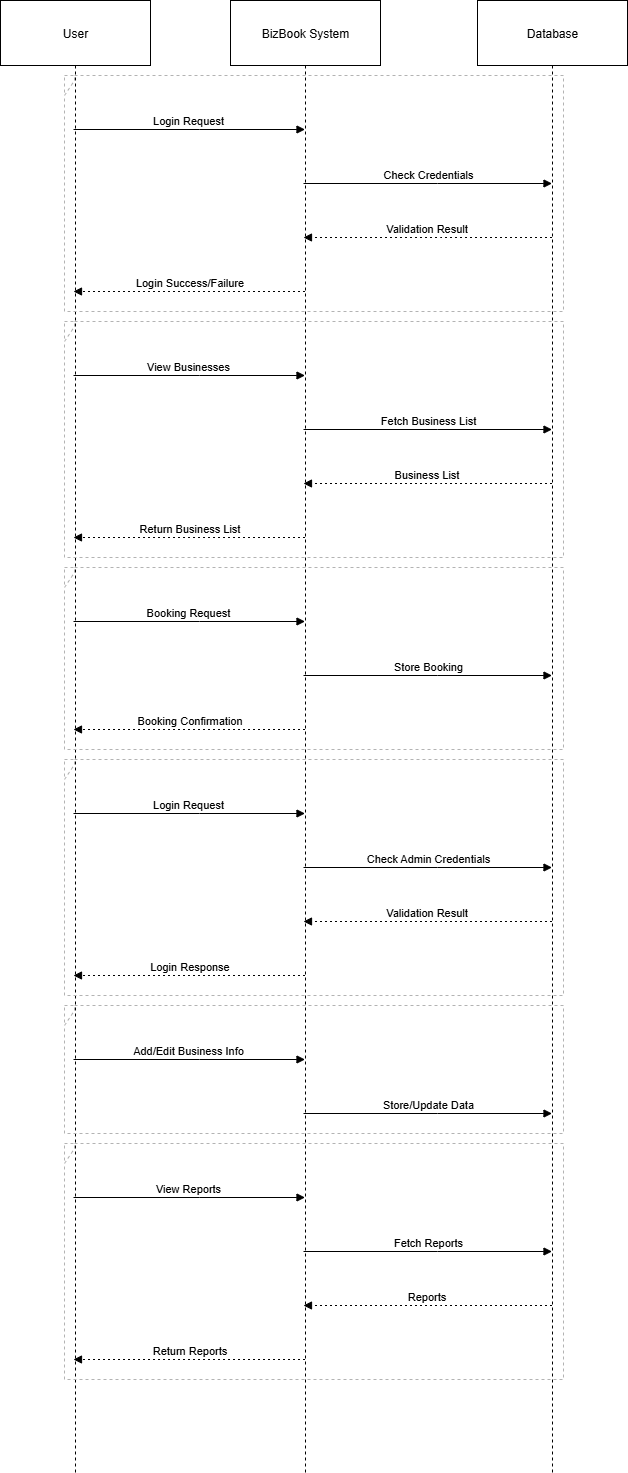
**Figure 5.5 Class Diagram**

**Use Case Diagram:**



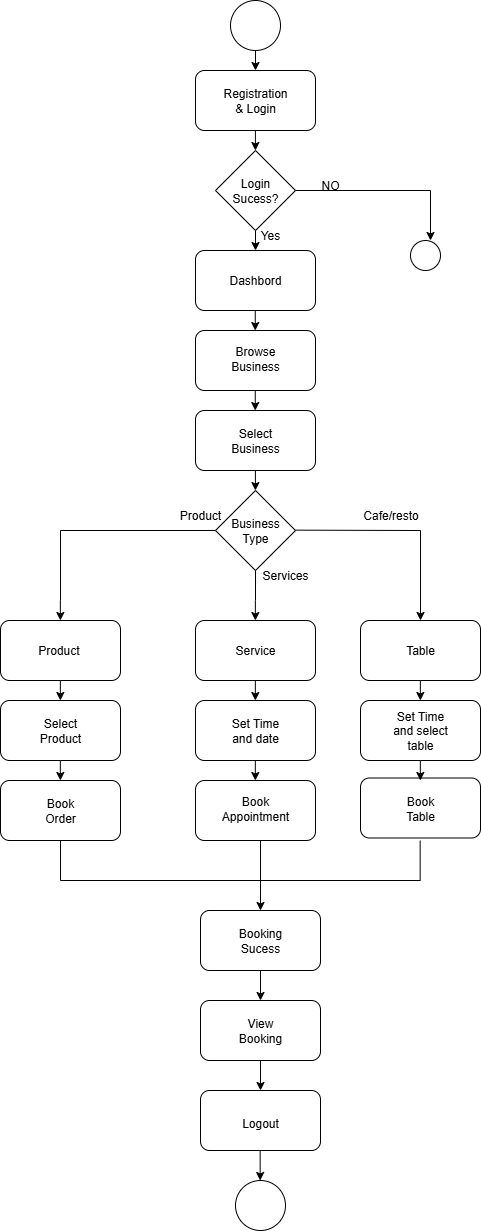
**Figure 5.6 Use Case Diagram**

**Sequential Diagram:**



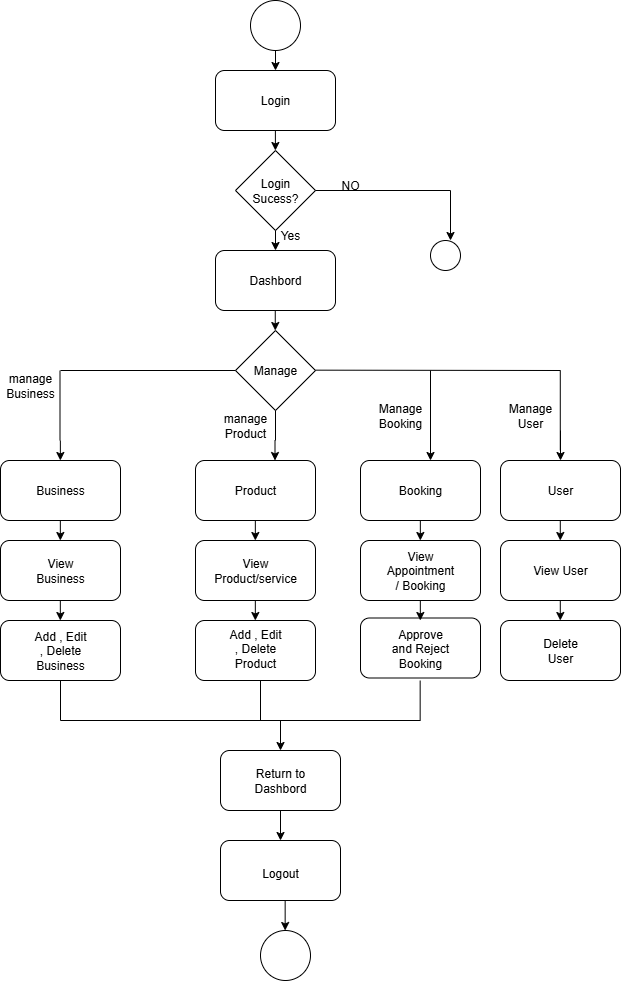
**Figure 5.7 Sequential Diagram**

**Activity Diagram**: **Client side:**



**Figure 5.8 Activity Diagram-Client**

**Admin Side:**



**Figure 5.9 Activity Diagram-Admin**

**Database Design:**

1. **User Collection**

Stores all user details including registered clients and role-based access.

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| **\_id** | ObjectId | Unique identifier (automatically generated) |
| **name** | String | Full name of the user |
| **email** | String | User's email (must be unique) |
| **password** | String | Hashed password |
| **phone** | String | Contact number |
| **role** | String | user or admin |
| **createdAt** | Date | Registration date |

**Table 5.1 User Collection**

1. **Business Collection**

Represents businesses listed on the platform (e.g., salons, restaurants)

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| **\_id** | ObjectId | Unique identifier |
| **name** | String | Business name |
| **email** | String | Business contact email |
| **phone** | String | Business phone number |
| **address** | String | Full address |
| **description** | String | Short description or about section |
| **category** | ObjectId | FK reference to Category collection |
| **createdAt** | Date | Timestamp of addition |

**Table 5.2 Business Collection**

1. **Category Collection**

Predefined categories with booking logic (appointment/order/table).

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| **\_id** | ObjectId | Unique identifier |
| **name** | String | e.g., Salon, Restaurant, Retail Store |
| **type** | String | appointment, table, or order |

**Table 5.3 Category Collection**

1. **Product/Service Collection**

Stores services or products offered by businesses.

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| **\_id** | ObjectId | Unique identifier |
| **name** | String | Name of product/service |
| **description** | String | Details about the product/service |
| **price** | Number | Price in INR or other local currency |
| **businessId** | ObjectId | FK reference to Business collection |
| **createdAt** | Date | Timestamp of creation |

**Table 5.4 Product/Service Collection**

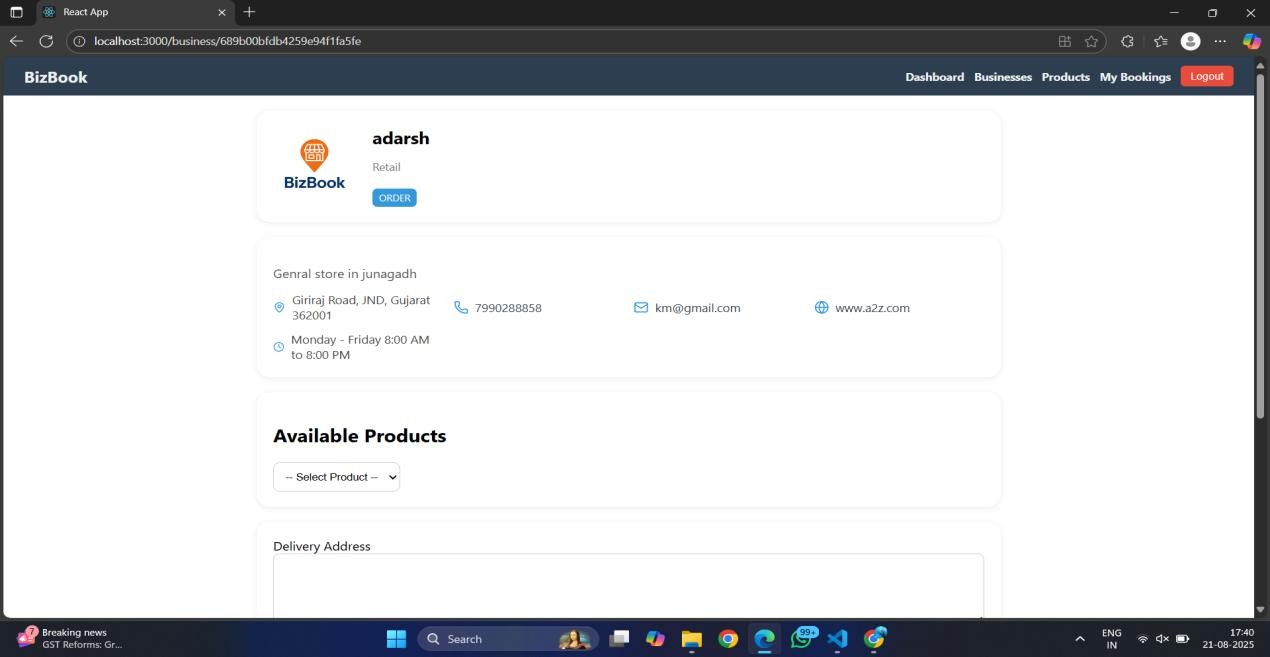
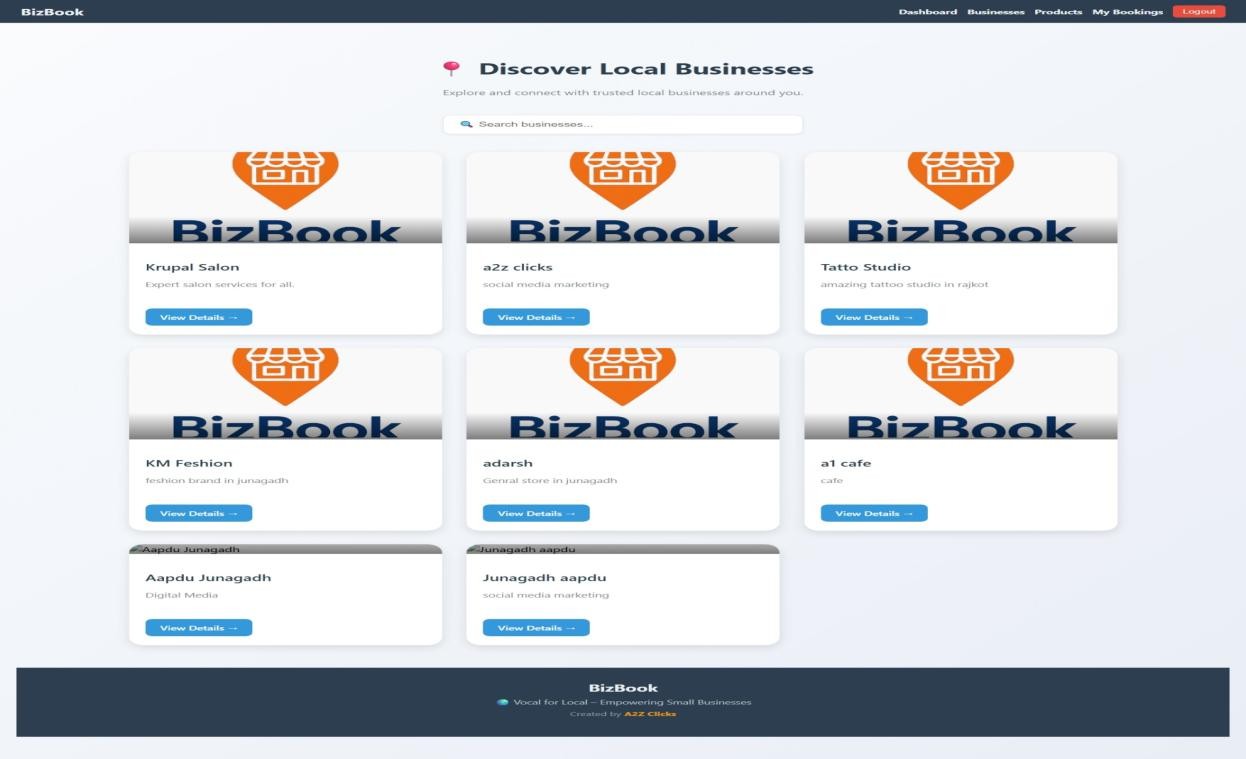
1. **Booking Collection**

Captures user bookings or product orders based on category type.

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| **\_id** | ObjectId | Unique identifier |
| **userId** | ObjectId | FK to User collection |
| **businessId** | ObjectId | FK to Business collection |
| **productId** | ObjectId | FK to Product/Service (optional) |
| **date** | String | Date of booking (YYYY-MM-DD format) |
| **time** | String | Time of appointment (if applicable) |
| **type** | String | appointment, table, or order |
| **status** | String | pending, confirmed, completed |
| **createdAt** | Date | Timestamp of booking |

**Table 5.5 Booking Collection**

**Menu Design:**



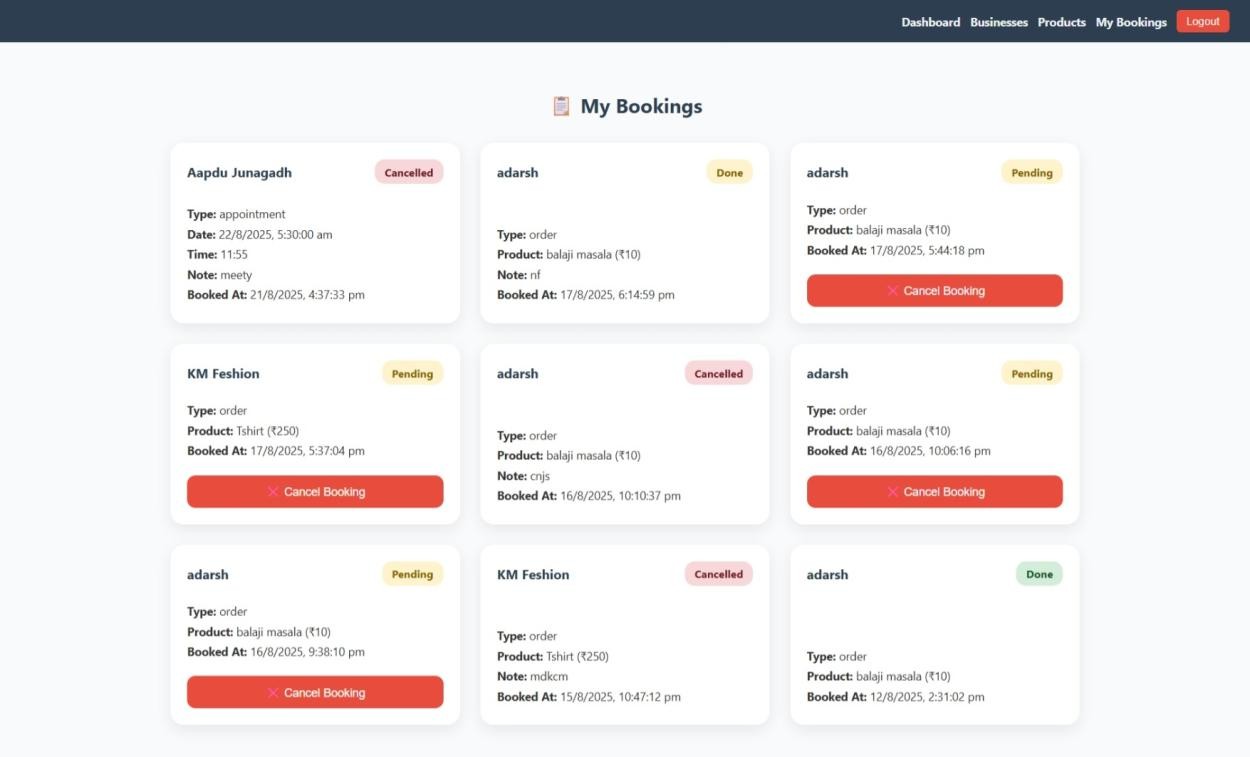
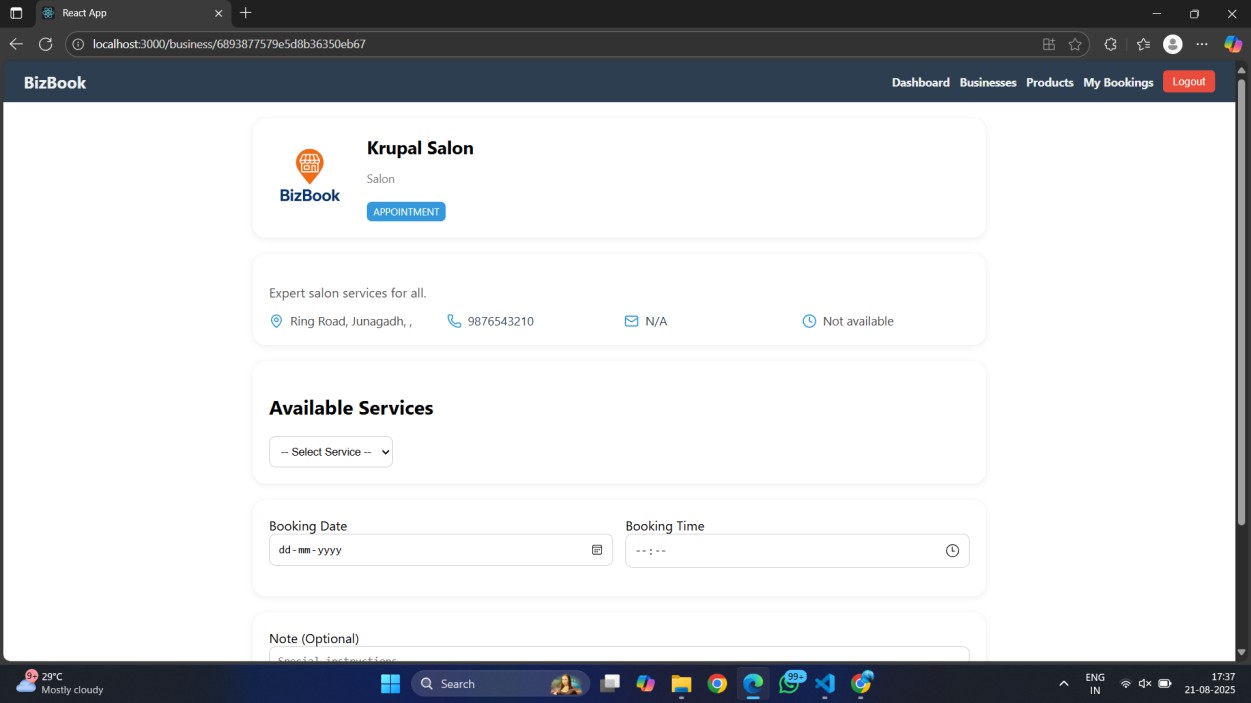
1. **Business List**

**Figure 5.10.1 Business List**

1. **Business Product**

**Figure 5.10.2 Business Product**

1. **Business Details and Book Service**

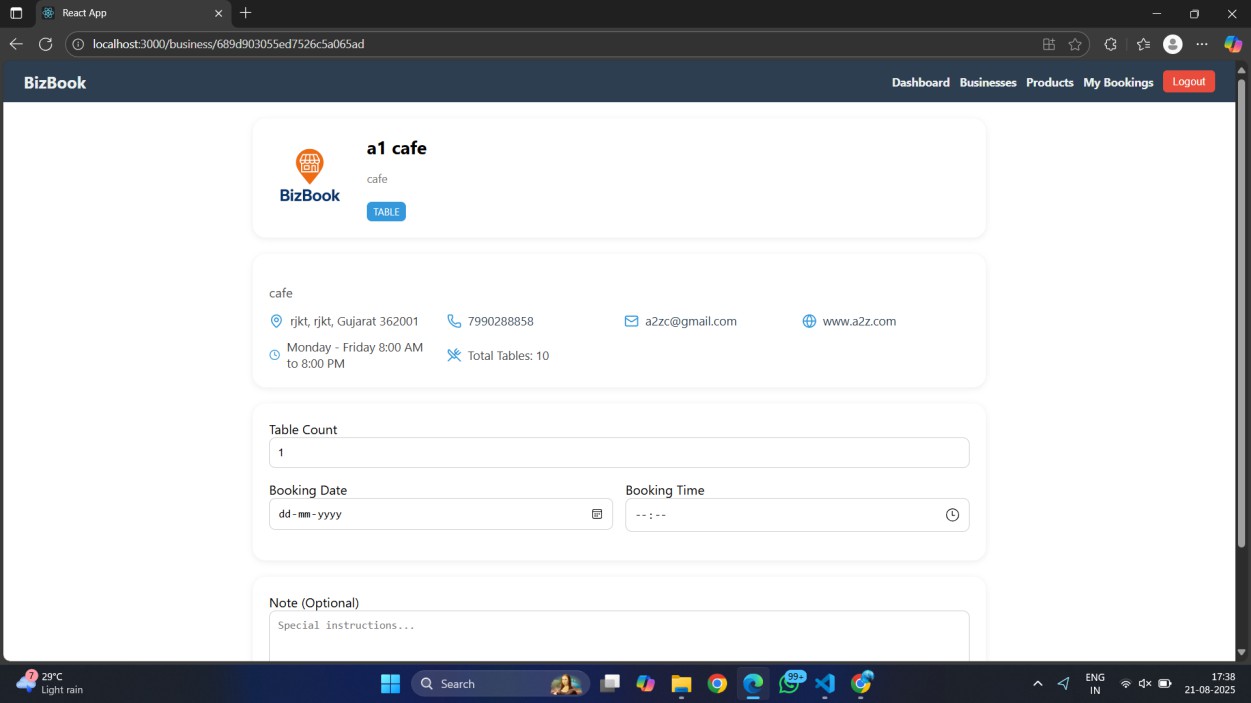
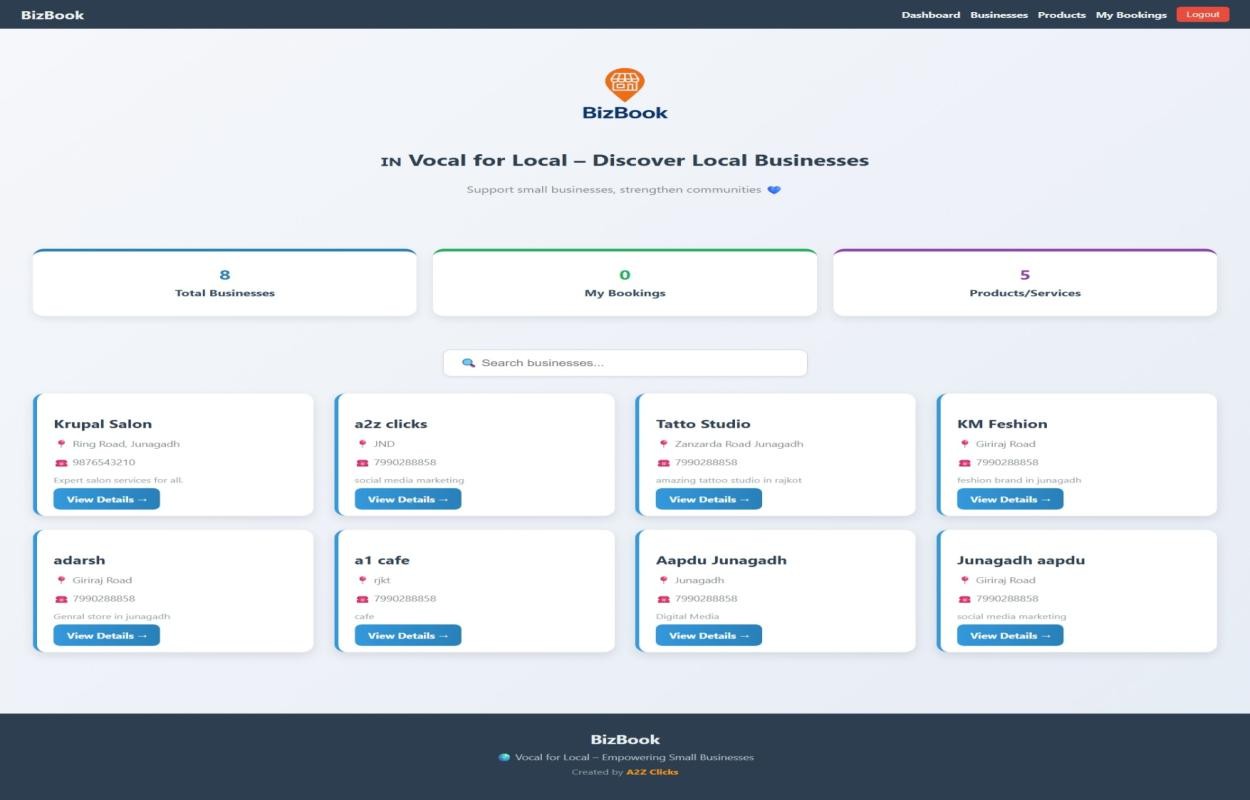


**Figure 5.10.3 Business List**

1. **My Bookings**

**Figure 5.10.4 My Bookings**

1. **BizBook Dashboard**

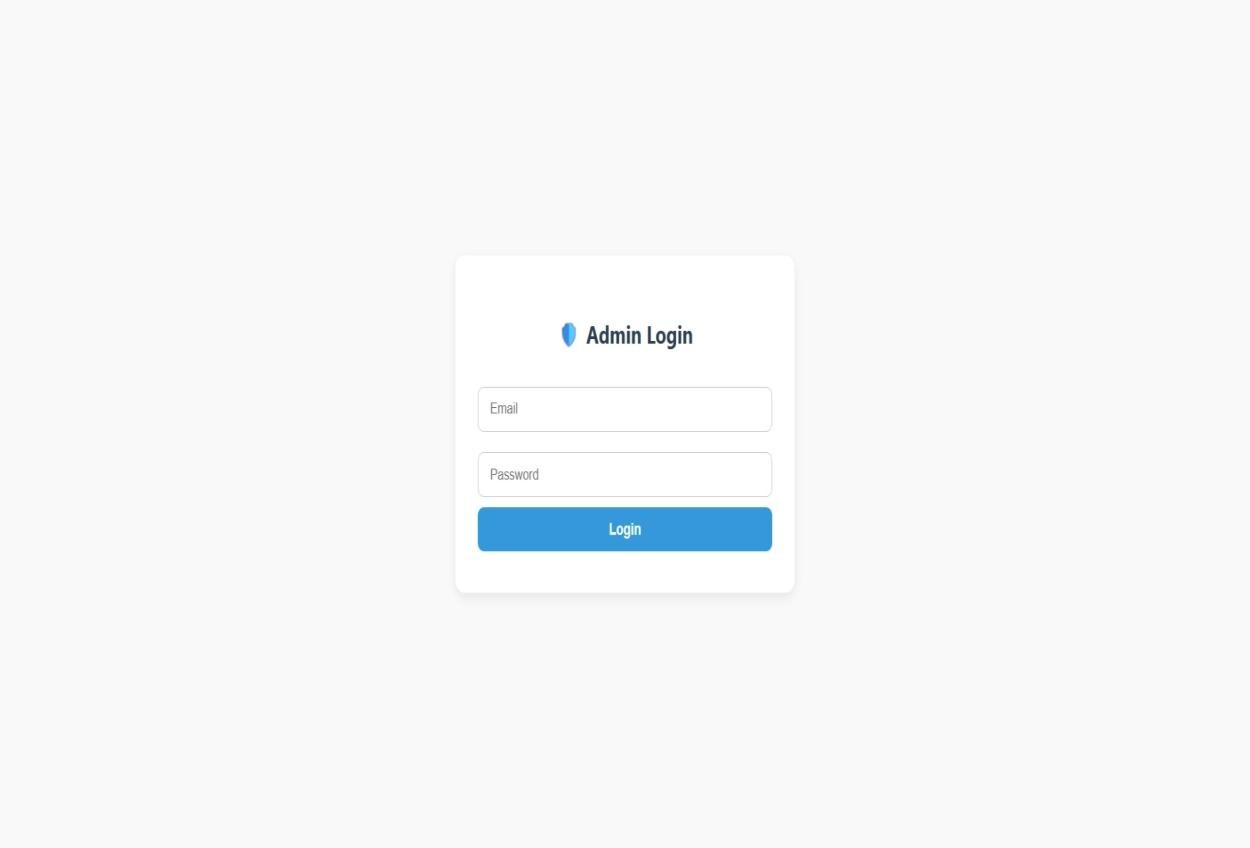
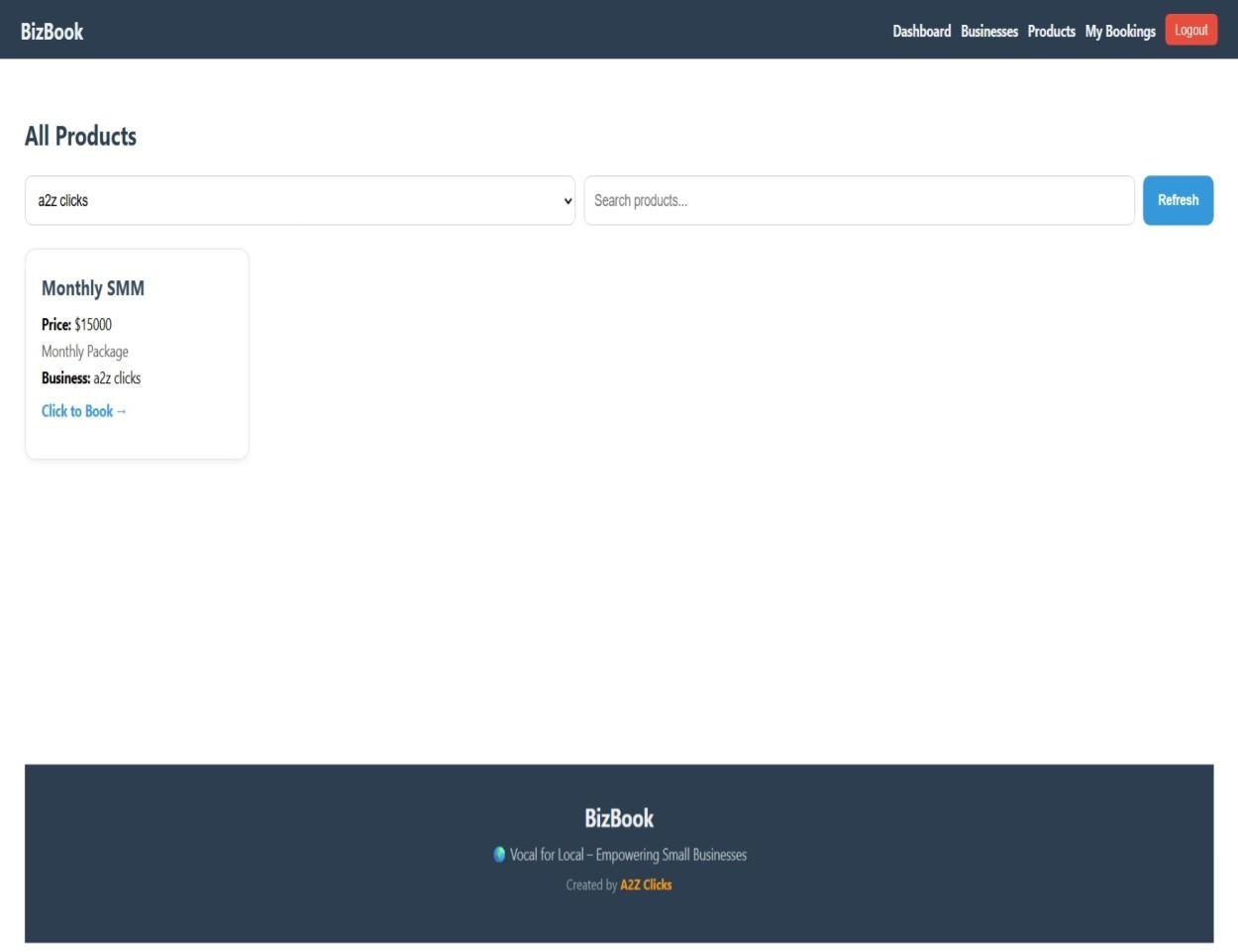


**Figure 5.10.5 BizBook Dashboard**

1. **Business Table**

**Figure 5.10.6 Business Table**

1. **Select Product by business**

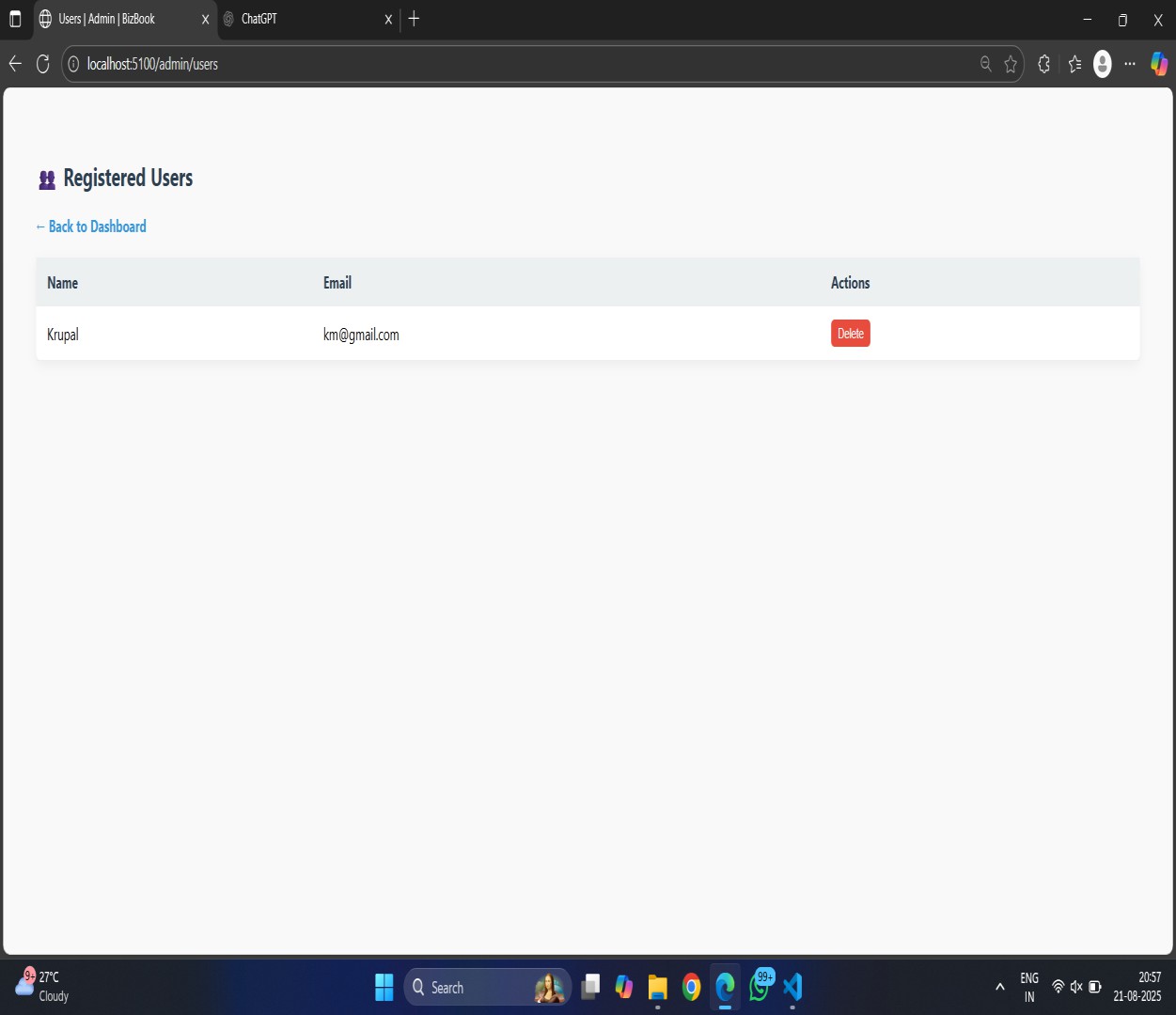
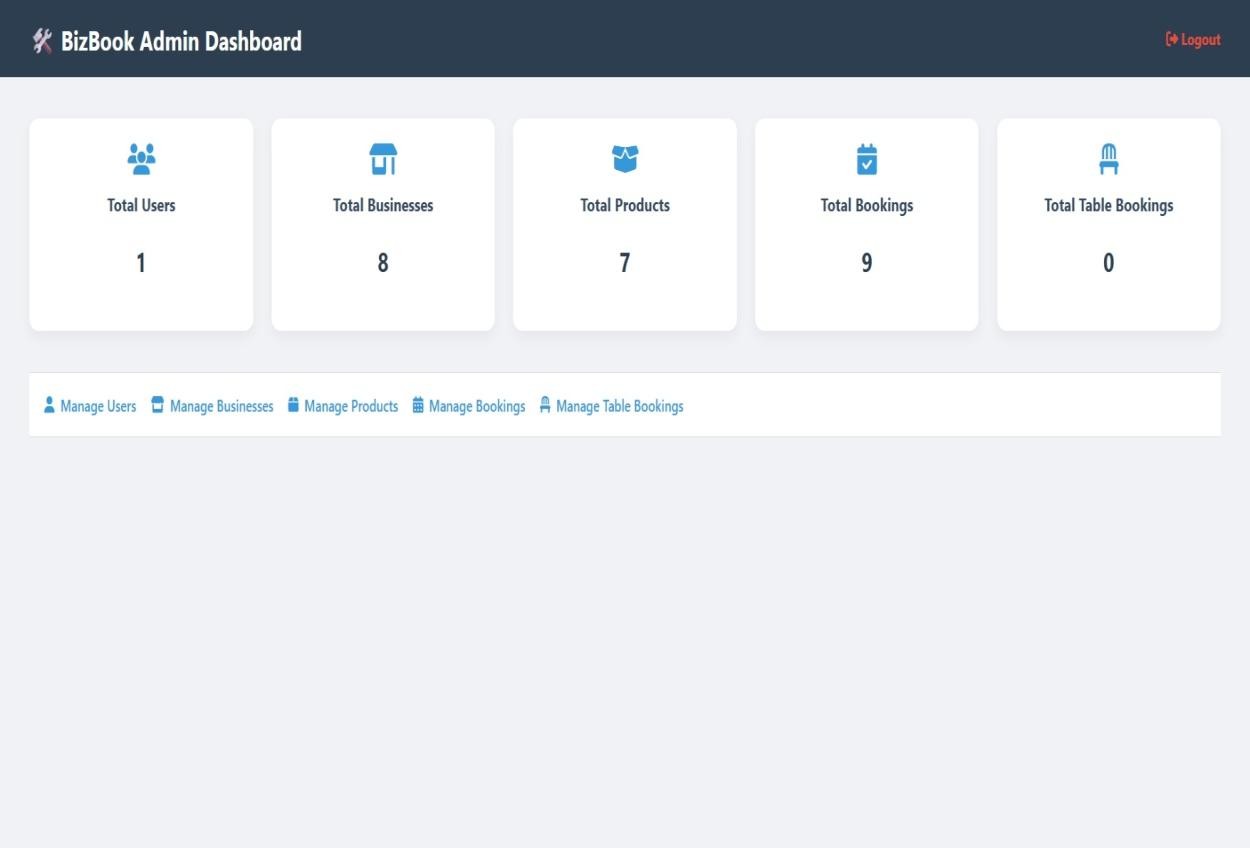


**Figure 5.10.7 Select Product by business**

1. **Admin Login**

**Figure 5.10.8 Admin Login**

1. **Admin Home page**

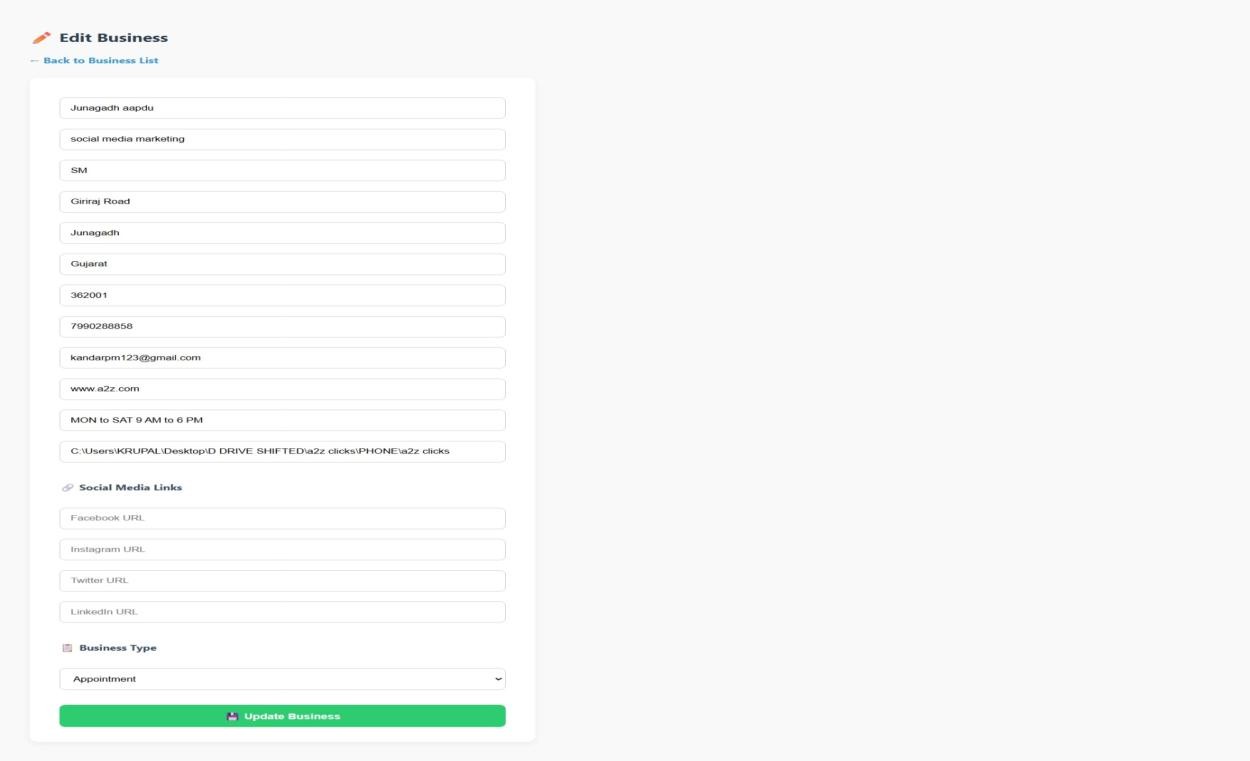
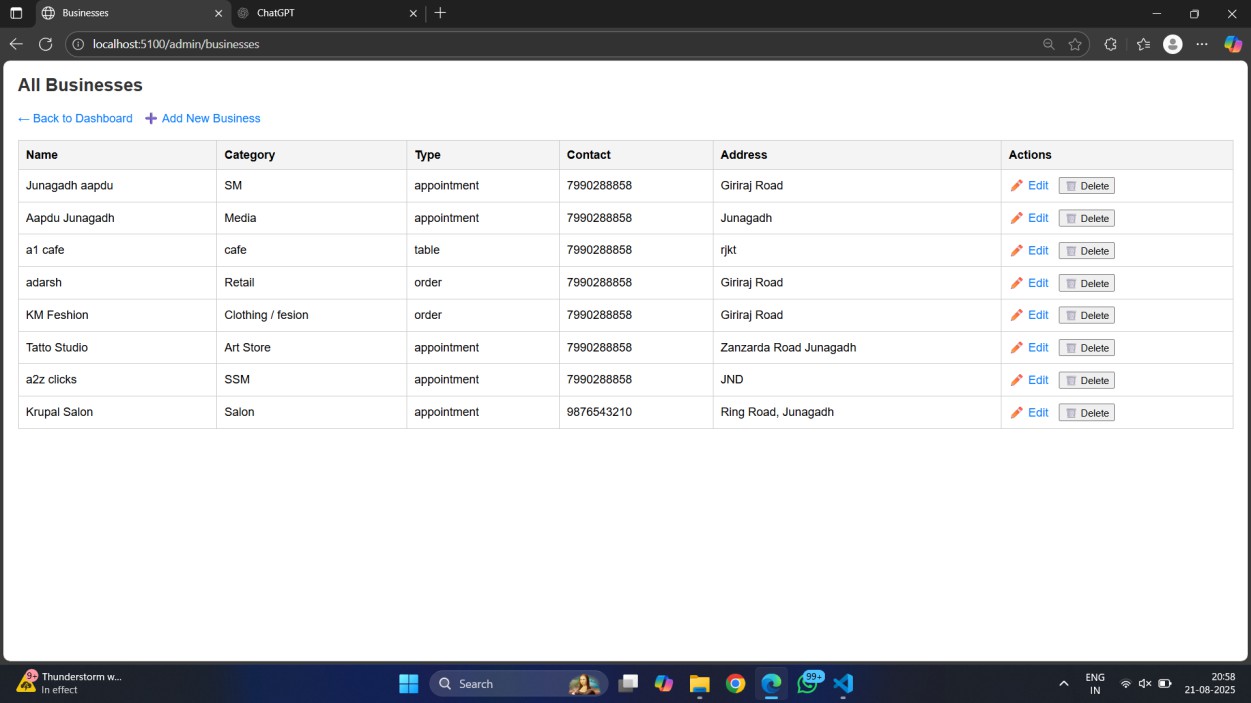


**Figure 5.10.9 Admin Home Page**

1. **Admin Manage User**

**Figure 5.10.10 Admin Manage User**

1. **Manage Business List**

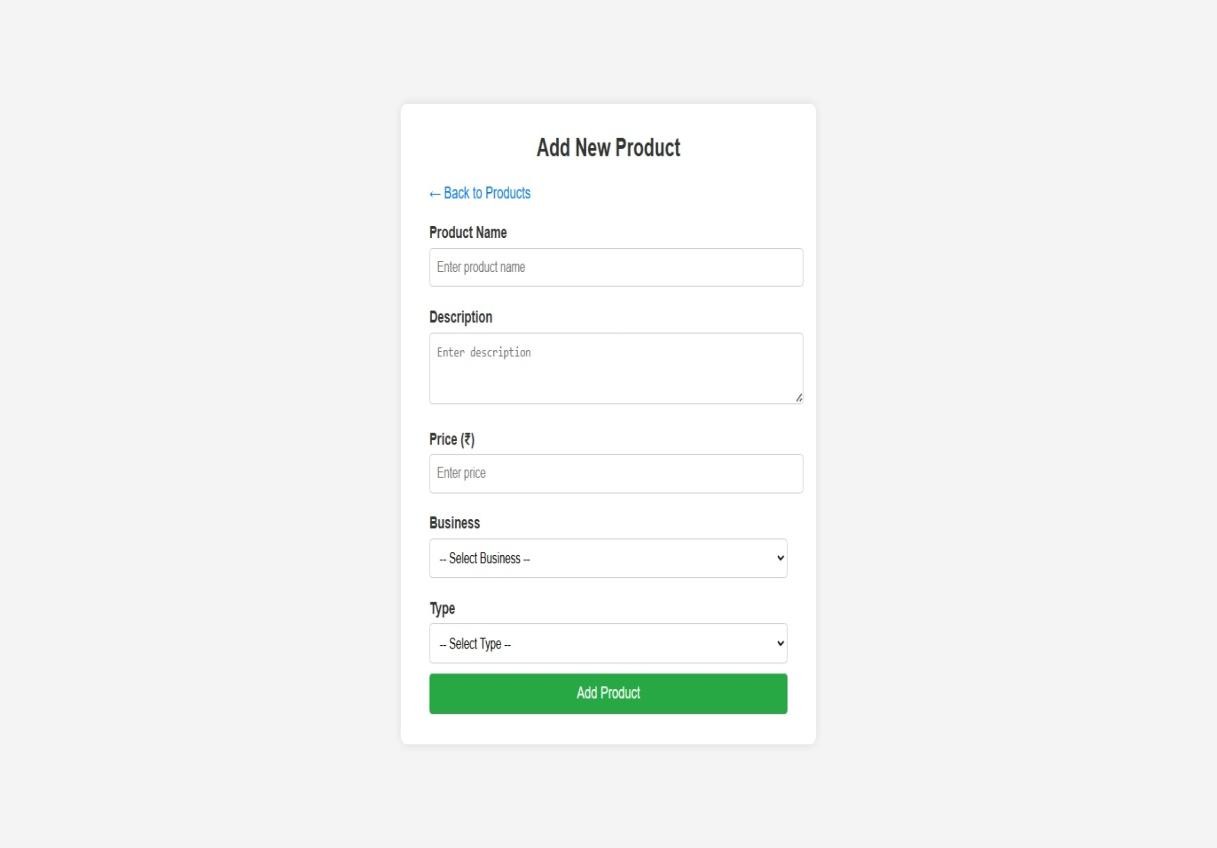
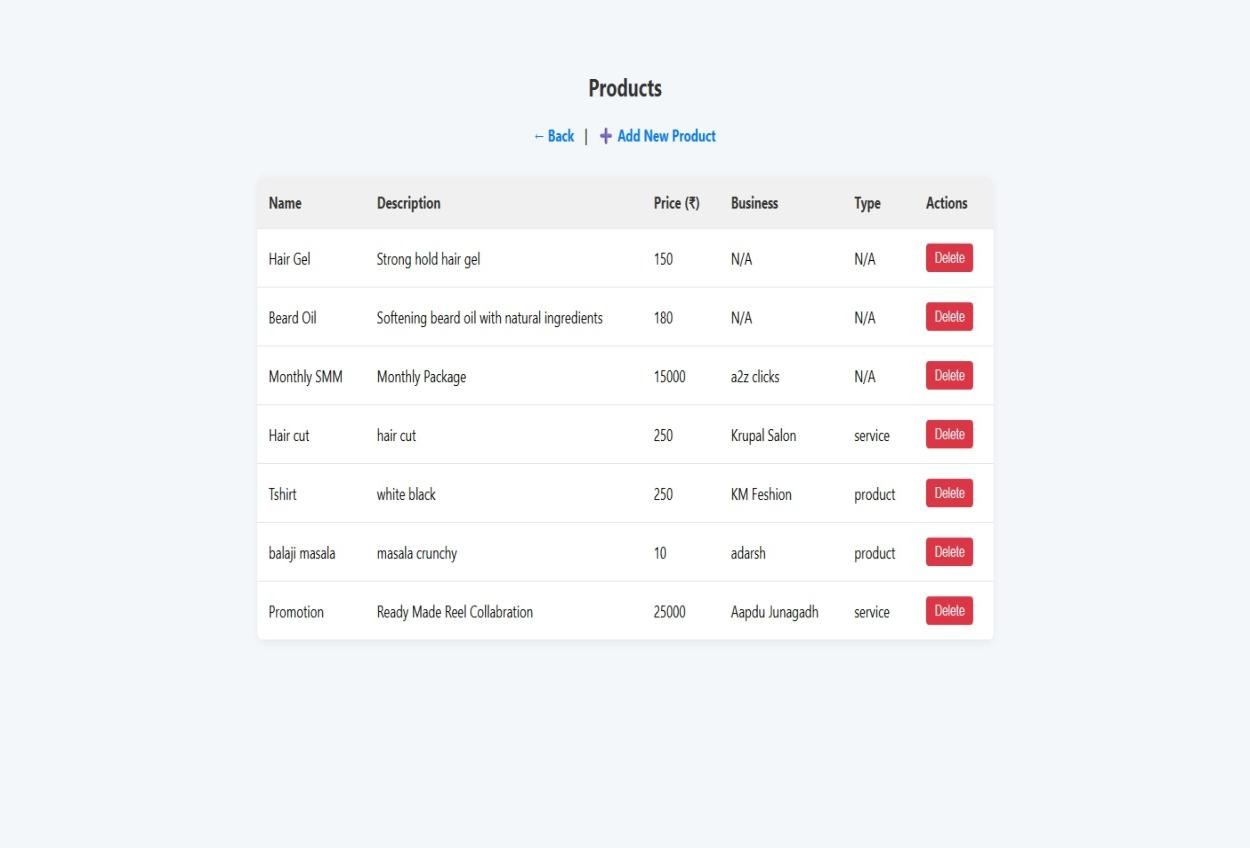


**Figure 5.10.11 Manage Business List**

1. **Edit Business**

**Figure 5.10.12 Edit Business**

1. **Product List**

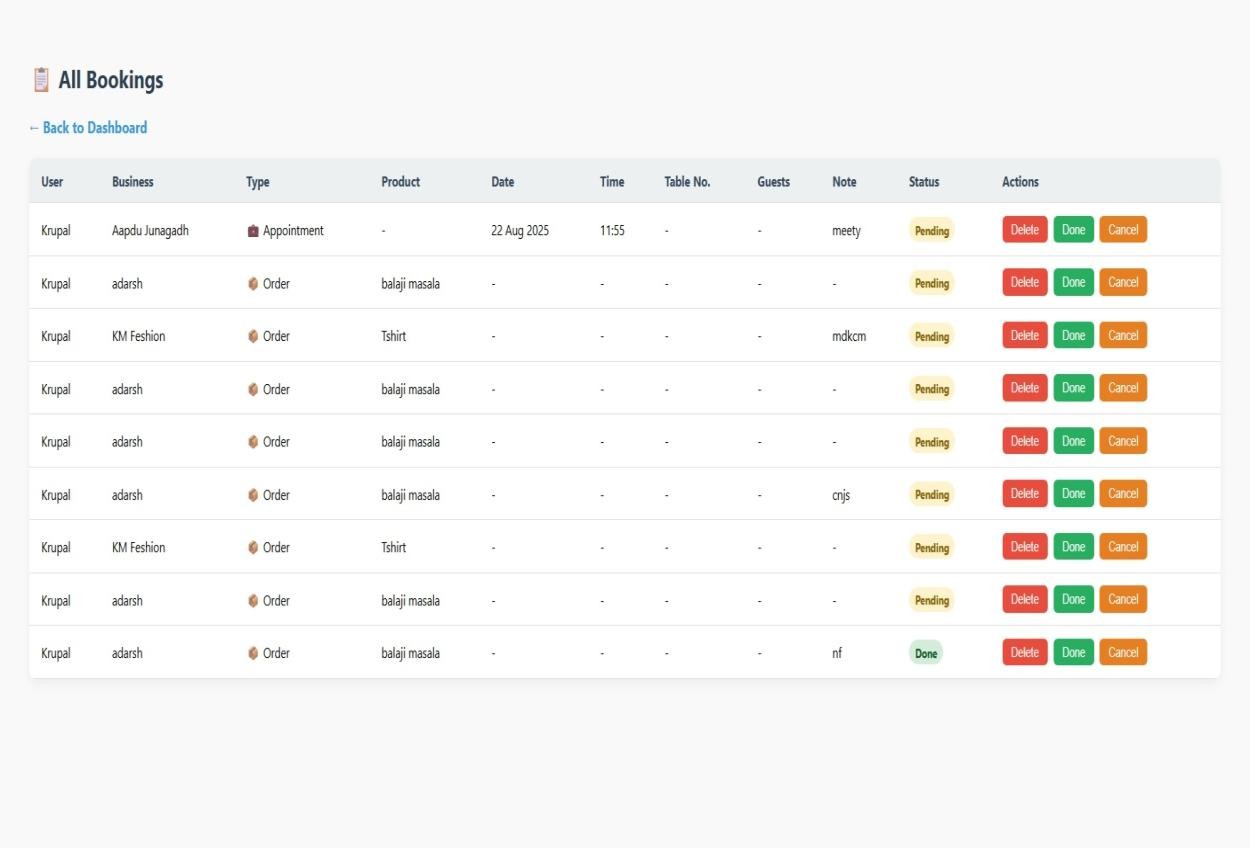


**Figure 5.10.13 Product List**

1. **Add New Product**

**Figure 5.10.14 Add New Product**

1. **Manage Booking**



**Figure 5.10.15 Manage Booking**

**Screen Design :**

1. **Landing Page (Public Home)**
   * **Purpose:** Showcase all businesses with basic filters.
   * **Elements:**
     + Top navbar with login/register
     + Business cards (name, category, location, CTA: “View Details”)
     + Search bar (by name/category)
2. **User Authentication Pages**

➤ **Login Page**

* + Fields: Email, Password
  + Button: Login
  + Link: “New user? Register now”

➤ **Register Page**

* + Fields: Name, Email, Phone, Password
  + Button: Register
  + Link: “Already have an account? Login”

1. **Business Details Page**
   * **Purpose:** Show business info + products/services + booking options
   * **Elements:**
     + Business name, address, contact
     + Category badge (e.g., "Salon" / "Retail")
     + Products/Services list (name, price, description)
     + Booking form:
       - For appointment/table: date & time picker
       - For order: one-click confirm
2. **Book Appointment Page**
   * **Fields:**
     + Auto-filled: Business, Product
     + Input: Date, Time
     + Button: Confirm Booking
3. **My Bookings Page**
   * **List of user’s bookings**
   * **Shows:**
     + Business name
     + Product/service
     + Date, Time, Type
     + Status (Pending/Confirmed)
4. **Admin Panel (EJS UI)**
   * **Admin Login Page**
   * **Dashboard**
     + Total Users, Bookings, Businesses, Products
   * **Business Management**
     + List, Add, Edit, Delete
   * **Product Management**
     + Linked to businesses
   * **User Management**
     + View all registered users
   * **Booking Management**
     + View bookings with filters
5. **SYSTEM TESTING**

**Authentication (User & Admin) :**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test**  **Case ID** | **Test Case Description** | **Test Steps** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Comment** |
| C- 001 | Register new user with valid credentials | Go to register → Enter valid details → Submit | Name, Email, Password | User registered successfully | User registered successfully | Pass | – |
| C- 002 | Prevent duplicate registration | Go to register → Enter already registered  email | Existing Email | Error shown: "Email already registered" | Error shown: "Email already registered" | Pass | – |
| C- 003 | Login with valid credentials | Go to login  → Enter valid email &  password | Registered Email, Password | Login successful | Login successful | Pass | – |
| C- 004 | Login with invalid password | Go to login  → Enter correct email but wrong  password | Email, Wrong Password | Error: "Invalid password" | Login successful | Fail | Password validation not working |
| C- 005 | Unauthorized access | Try to access protected route without  login | – | Access denied | Data accessed without login | Fail | Authorization middleware missing |

**Table 6.1 Authentication (User & Admin)**

**Business Management :**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Case Description** | **Test Steps** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Comment** |
| C- 006 | View all businesses | Login →  Navigate to "All  Businesses" | – | All businesses displayed | All businesses displayed | Pass | – |
| C- 007 | View single business by ID | Login → Select business ID | Business ID | Business details displayed | Error: "Business details not loading" | Fail | API response delay |
| C- 008 | Invalid business ID | Login →  Enter invalid business ID | Invalid ID | Error:  "Business not found" | Error:  "Business not found" | Pass | – |
| C- 009 | Unauthorized add attempt | Try to add a business as user (non- admin) | Business Data | Error: "Unauthorized action" | Business added successfully | Fail | Access control not implemented |

**Table 6.2 Business Management Product/Service Management :**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Case Description** | **Test Steps** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Comment** |
| C- 009 | View products for a business | Login  → Open  products under a business | Business ID | List of products displayed | List of products displayed | Pass | – |
| C- 010 | View products with invalid business ID | Login  → Enter wrong business  ID | Invalid Business ID | Error message shown | Error message shown | Pass | – |
| C- 011 | Product without business FK | Try to add product without linking to  business | Product Data (No FK) | Error: "Business ID  required" | Product added without validation | Fail | Missing FK  validation |
| C- 012 | Incomplete product submission | Try to add product with  missing fields | Product Data (Incomplete) | Error: "All fields required" | Error not shown, product partially saved | Fail | Field validation missing |

**Table 6.3 Product/Service Management**

**Booking System :**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Case Description** | **Test Steps** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Comment** |
| C- 011 | Book appointment (Salon) | Login → Choose salon → Book slot | Salon ID, Date/Time | Appointment booked | Appointment booked | Pass | – |
| C- 012 | Book table (Restaurant) | Login → Choose restaurant  → Select  table | Restaurant ID,  Date/Time | Table booked | Table booked | Pass | – |
| C- 013 | Order product (Retail) | Login → Choose retail shop → Order item | Product ID, Qty | Order placed successfully | Payment gateway error | Fail | Payment integration issue |
| C- 014 | Double booking attempt | Try booking already reserved slot | Existing Slot | Error: "Slot not available" | Booking confirmed for same slot | Fail | Double booking not prevented |
| C- 015 | Invalid booking type | Submit invalid booking type | Invalid Type | Error message shown | Error message shown | Pass | – |

**Table 6.4 Booking System**

**User Booking History :**

**Table 6.5 User Booking History Admin Dashboard Testing :**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Case Description** | **Test Steps** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Comment** |
| C- 016 | View all user bookings | Login → Open "My Bookings" | User ID | List of bookings displayed | No bookings shown even though user has past  bookings | Fail | Booking history fetch not working |
| C- 017 | View booking confirmation | Login → Book → Check confirmation | Booking ID | Confirmation shown | Confirmation shown | Pass | – |
| C- 018 | Unauthorized booking  history access | Open "My Bookings"  without login | – | Access denied | Bookings displayed  without login | Fail | Missing authentication  check |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test**  **Case ID** | **Test Case Description** | **Test Steps** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Comment** |
| A- 001 | Admin login | Go to admin login → Enter valid credentials | Admin Email, Password | Login successful | Login successful | Pass | – |
| A- 002 | Unauthorized admin action | Login as normal user  → Try admin  route | User Token | Access denied | Admin features accessible | Fail | Role-based access  control missing |
| A- 003 | View all businesses | Login as admin → Open "All Businesses" | – | All businesses displayed | No businesses loaded (blank page) | Fail | API  response issue |
| A- 004 | Delete business with cascade | Login as admin → Delete a business | Business ID | Business and related data deleted | Only business deleted, related  products remain | Fail | Cascade delete not working |
| A- 005 | Add business category | Login as admin → Add new  category | Category Data | Category added successfully | Category added successfully | Pass | – |
| A- 006 | View all bookings | Login as admin → Open "All  Bookings" | – | All user bookings displayed | All user bookings displayed | Pass | – |

**Table 6.6 Admin Dashboard Testing**

**Business Management (Admin):**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Cas e ID** | **Test Case Descriptio n** | **Test Steps** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fa il** | **Commen t** |
| A- 003 | Add new business | Login as admin → Enter valid details | Business Data | Business added successfull y | Business added successfull y | Pass | – |
| A- 004 | Add business with missing  fields | Login as admin → Submit incomplet  e data | Incomplet e Data | Error shown | Business added without validation | Fail | Field validatio n missing |
| A- 005 | Get all businesses (Admin) | Login as admin → Fetch all businesse s | – | All businesses displayed | No businesses shown | Fail | API not returning data |
| A- 006 | Get single business (Admin) | Login as admin → Enter valid business ID | Business ID | Business details displayed | Business details displayed | Pass | – |

**Table 6.7 Business Management (Admin)**

**Product/Service Management (Admin)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tes** | **Test Case** | **Test** | **Input** | **Expected** | **Actual** | **Pass/Fa** | **Comme** |
| **t** | **Description** | **Steps** | **Data** | **Result** | **Result** | **il** | **nt** |
| **Cas** |  |  |  |  |  |  |  |
| **e** |  |  |  |  |  |  |  |
| **ID** |  |  |  |  |  |  |  |
| A- | Add | Login | Product | Product | Product | Pass | – |
| 005 | product/servi | as | Data | added | added |  |  |
|  | ce linked to | admin |  | successful | successful |  |  |
|  | business | → |  | ly | ly |  |  |
|  |  | Submit |  |  |  |  |  |
|  |  | valid |  |  |  |  |  |
|  |  | product |  |  |  |  |  |
|  |  | with |  |  |  |  |  |
|  |  | busines |  |  |  |  |  |
|  |  | s FK |  |  |  |  |  |
| A- | Add product | Login | Product | Error: | Product | Fail | FK |
| 006 | without | as | Data (No | "Business | added |  | validatio |
|  | business FK | admin | FK) | required" | without |  | n |
|  |  | → |  |  | validation |  | missing |
|  |  | Submit |  |  |  |  |  |
|  |  | product |  |  |  |  |  |
|  |  | without |  |  |  |  |  |
|  |  | linking |  |  |  |  |  |
| A- | Incomplete | Login | Incomple | Error | Error | Pass | – |
| 007 | product | as | te Data | shown | shown |  |  |
|  | submission | admin |  |  |  |  |  |
|  |  | → |  |  |  |  |  |
|  |  | Submit |  |  |  |  |  |
|  |  | missing |  |  |  |  |  |
|  |  | fields |  |  |  |  |  |
| A- | Get products | Login | Business | List of | No | Fail | API not |
| 008 | for a business | as | ID | products | products |  | fetching |
|  | (Admin) | admin |  | displayed | shown |  | products |
|  |  | → |  |  | despite |  |  |
|  |  | Fetch |  |  | entries |  |  |
|  |  | product |  |  |  |  |  |
|  |  | s by |  |  |  |  |  |
|  |  | Busine |  |  |  |  |  |
|  |  | ss ID |  |  |  |  |  |

**Table 6.8 Product/Service Management (Admin)**

**Admin Dashboard:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Case Description** | **Test Steps** | **Input Data** | **Expected Result** | **Actual Result** | **Pass/Fail** | **Comment** |
| A- 008 | View all businesses | Login as  admin →  Open "All Businesses" | – | List displayed | Blank list shown (data not loading) | Fail | API fetch issue |
| A- 009 | View all bookings | Login as admin → Open "All Bookings" | – | All bookings displayed | All bookings displayed | Pass | – |
| A- 010 | Add business category | Login as admin → Add new  category | Category Data | Category added successfully | Category added successfully | Pass | – |
| A- 011 | Delete business with cascade | Login as admin → Delete business | Business ID | Business + related data deleted | Only business deleted (products remain) | Fail | Cascade delete not working |
| A- 012 | Unauthorized admin action | Try admin action with user token | User Token | Access denied | Admin actions accessible with user  token | Fail | Role-based access missing |

**Table 6.9 Admin Dashboard**

1. **CONCLUSION**

The **BizBook Project** successfully integrates a complete business booking and management system using **Node.js, Express, MongoDB, and EJS**. It provides a **role- based platform** where:

* **Users** can register, log in, and easily make **table bookings, product orders, or appointments**.
* **Business Owners** can manage their own business profiles, products, and bookings through a secure dashboard.
* **Admins** can oversee the entire system, update booking statuses, and manage businesses through an **EJS-based admin panel**.

The project uses a **hybrid authentication system**:

* **JWT tokens** secure API routes for mobile apps and external clients.
* **Session-based authentication** secures the web dashboards for admin, owners, and users.

With its modular design (separate models, controllers, and middleware), the system is

**scalable, secure, and easy to extend**.

In the future, features like **payment integration, notifications, and analytics** can be added without major changes.

In short, this project demonstrates how a **single system** can serve both **API consumers** and **web dashboard users**, making it flexible and practical for real-world business solutions.

1. **LEARNING DURING PROJECT WORK**

* **Backend Development**: Learned how to use **Node.js and Express** for creating REST APIs and handling different routes.
* **Database Handling**: Understood how to design and manage collections in

**MongoDB**, including relationships between users, businesses, and bookings.

* **Authentication**: Implemented both **JWT-based authentication** for APIs and **session-based authentication** for web dashboards, which gave hands-on experience with security.
* **Role Management**: Learned how to separate features for **Admin, Business Owner, and User**, ensuring that each role gets the right access.
* **Error Handling**: Understood the importance of validating data and handling errors properly so the system works smoothly.
* **Teamwork & Documentation**: Improved communication, documentation skills, and ability to work on a large project step by step.
  1. **Future Enhancement:**
     + **Payment Gateway Integration**: Allow users to pay for bookings/orders online using UPI, cards, or wallets.
     + **Notification System**: Add email/SMS/WhatsApp alerts to inform users about booking confirmations or updates.
     + **Analytics Dashboard**: Provide owners and admins with reports about sales, bookings, and customer trends.
     + **Mobile Application**: Extend the system with a mobile app for Android/iOS using the same APIs.
     + **Reviews & Ratings**: Allow customers to give feedback on businesses and services.
     + **Multi-language Support**: Make the platform usable by people from different regions.

These enhancements will make the system more **user-friendly, scalable, and practical** for real business use.

1. **BIBLIOGRAPHY**

# Online References

1. W3Schools, *“Node.js Tutorial,”* W3Schools Online Web Tutorials. [Online]. Available: [https://www.w3schools.com/nodejs/](https://www.w3schools.com/nodejs/?utm_source=chatgpt.com). [Accessed: July 24, 2025].
2. Mozilla Developer Network (MDN), *“Express.js Guide,”* MDN Web Docs. [Online]. Available: [https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express\_Nodejs.](https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs?utm_source=chatgpt.com) [Accessed: July 24, 2025].
3. GeeksforGeeks, *“ReactJS Tutorial,”* GeeksforGeeks. [Online]. Available: [https://www.geeksforgeeks.org/reactjs-tutorial/](https://www.geeksforgeeks.org/reactjs-tutorial/?utm_source=chatgpt.com). [Accessed: July 25, 2025].

# Offline References

1. E. Brown, *Web Development with Node and Express: Leveraging the JavaScript Stack*,

2nd ed. Sebastopol, CA, USA: O’Reilly Media, 2019.

1. A. Banks and E. Porcello, *Learning React: Functional Web Development with React and Redux*, 2nd ed. Sebastopol, CA, USA: O’Reilly Media, 2018.
2. S. Holzner, *MongoDB: The Definitive Guide*, 3rd ed. Sebastopol, CA, USA: O’Reilly

Media, 2020.