



SDJ INTERNATIONAL
COLLEGE

Master of Science (Computer Application) (M.Sc. (CA)) Programme

Project Report

M.Sc.(CA) Sem II
AY 2024-25

Project Title: Elite Assist

By

Exam No	Name of Student
32	Harshal Jariwala

Project Guide by :
Prof. Bhumika Patel

Acknowledgement

The success and outcome of this project required a lot of guidance and assistance from many people, and I am extremely fortunate to have gotten this all along with the completion of my project work. Whatever I have done is only due to such guidance and assistance.

I would not forget to thank Principal Dr. Aditi Bhatt, IQAC coordinator and trust representative Dr. Vaibhav Desai, IT Incharge Dr. Vimal Vaiwala, Head of BCA Department Prof. Nainesh Gathiyawala and Project guide **Prof. Bhumika Patel** and all other assistant professors, who took a keen interest in my project work and guided me till the completion of my project work by providing all the necessary information for developing a good system.

I am extremely grateful to her for providing such nice support and guidance, though she had a busy schedule managing the college dealings.

I am thankful and fortunate enough to have gotten support and guidance from all the teaching staff in the Bachelor of Computer Application Department, which helped me successfully complete our project work. Also, I would like to extend my sincere regards to all the non-teaching staff of the Bachelor of Computer Application Department for their timely support.

Harshal Jariwala

INDEX

Sr. No	Description	Page No.
1	Introduction	1 – 4
	1.1 Project description	1
	1.2 Project Profile	4
2	Environment Description	5 – 10
	2.1 Hardware and Software Requirements	5
	2.2 Technologies Used	6
3	System Analysis and Planning	11 – 19
	3.1 Existing System and its Drawbacks	11
	3.2 Feasibility Study	15
	3.3 Requirement Gathering and Analysis	17
4	Proposed System	20 – 24
	4.1 Scope	20
	4.2 Project Modules Functionalities	22
5	Detail Planning	25 – 31
	5.1 Data Flow Diagram / UML	25
	5.2 Process Specification / Activity Flow Diagram	27
	5.3 Entity-Relationship Diagram / Class Diagram	29
6	System Design	32 – 63
	6.1 Database Design	32
	6.2 Input Design	35
	6.3 Output Design	47
7	Software Testing	64 – 75
	7.1 Unit Testing	64
	7.2 Integration Testing	69
	7.3 System Testing	73
8	Future Scope of Enhancements	76
9	Bibliography & Reference	79

1. Introduction

1.1 Project Description

Elite Assist is an integrated platform that connects service providers, customers, and administrators to manage and streamline the process of service booking, execution, and management. The platform consists of three distinct interfaces:

- I. Admin Interface (Built using Laravel)
- II. Customer Interface (Built using Flutter)
- III. Partner Interface (Built using Flutter)

The system is powered by MongoDB, a NoSQL database, which is used for handling dynamic and unstructured data such as customer orders, services, partner details, and more.

The core objective of Elite Assist is to make the service booking process more efficient by offering a well-structured backend for admins, a user-friendly front-end for customers, and an easy-to-navigate partner portal for service providers. The platform ensures seamless communication between all parties involved and offers an efficient way to manage and track services, orders, and payments.

- **Admin:** The admin manages the backend of the platform, including adding and maintaining services, creating discount coupons, managing partners, and assigning customer orders to service providers (partners).
- **Customer:** The customer browses available services, selects categories and sub-categories, books services, applies available coupons, and tracks order history.
- **Partner:** The partner (service provider) receives service orders, views customer details, completes services, and updates the status of bookings.

The platform is designed to ensure smooth operation across all three interfaces, empowering each stakeholder with the necessary tools for managing their respective roles.

I. Admin Site:

The Admin is the central authority of the platform. The admin has the ability to manage all aspects of the service ecosystem, including defining services, managing categories, and handling orders. Admins can create and manage services, assign partners to services, configure promotional coupons, and manage the service execution lifecycle by assigning orders to partners. The admin also has the ability to monitor and modify the platform's content to ensure it aligns with business goals.

II. Customer Site:

The Customer interface is a mobile app developed using Flutter, where users can browse and book services offered by different partners. Customers can view services based on different categories and sub-categories, apply available coupons for discounts, and manage their order history. The platform provides an intuitive and streamlined experience for browsing services, selecting available time slots, and interacting with service providers. Customers also have the ability to see detailed partner profiles and ratings to help them make informed decisions.

III. Partner Site:

The Partner interface is another mobile app built using Flutter, where service providers can receive and manage orders. Partners can view assigned orders, access customer details, and update the service status once it is completed. The partner site ensures that service providers have all the necessary information to fulfill customer requests while enabling them to track and manage their work efficiently.

The system uses MongoDB, a NoSQL database, which is particularly well-suited for managing unstructured data. This choice allows for greater flexibility in storing diverse data types such as service details, customer orders, partner profiles, and service categories. MongoDB's scalability ensures that the platform can handle increasing amounts of data and traffic as the user base grows.

➤ **Key data entities stored in MongoDB include:**

- **Users:** Customers, partners, and admins, each with relevant details (name, contact, etc.).
- **Services:** Service categories, sub-categories, and specific services available to customers.
- **Orders:** Orders placed by customers, including the services selected, assigned partners, and status.
- **Coupons:** Discount codes that customers can apply to reduce the cost of services.
- **Partners:** Profiles of service providers, including their offered services, contact details, and location.

The system architecture is designed to allow all components (Admin, Customer, and Partner interfaces) to communicate with each other via a central API, ensuring seamless data flow and updates across all interfaces.

1. Admin Workflow:

The Admin begins by creating services, defining categories, and adding sub-categories. Admin can also upload banners to promote services and configure discount coupons.

Admin assigns service providers (partners) to specific services and manages their availability.

When customers place an order, the admin assigns the order to the appropriate partner for fulfillment.

The admin also manages the overall platform by monitoring service execution, reviewing order statuses, and ensuring that partners are meeting performance standards.

2. Customer Workflow:

The Customer browses available services based on categories and sub-categories. After selecting a service, the customer chooses an available time slot for the service.

Customers can apply coupons for discounts during the booking process and then finalize their booking.

The customer can track the status of their orders and view past service history at any time.

In addition, customers can view profiles of service providers (partners) to choose the right provider for their needs based on factors such as ratings, location, and expertise.

3. Partner Workflow:

The Partner receives service orders assigned by the admin and views all necessary details, including customer information and service requirements.

The partner can communicate with the customer if needed, and once the service is completed, the partner marks the order as finished, allowing the platform to update the order status.

The partner has the ability to view their order history, customer feedback, and ratings for each completed service, helping them improve service delivery.

Elite Assist offers a comprehensive and scalable solution for managing service-based businesses, providing a seamless experience for customers, service providers (partners), and administrators. The integration of Laravel and Flutter ensures a robust and user-friendly platform, while MongoDB allows for flexible and efficient data management. This makes Elite Assist an ideal choice for businesses looking to enhance their service delivery and booking processes across multiple platforms.

1.2 Project Profile

Project Title :	Elite Assist
Objective :	The objective of Elite Assist is to streamline service management by connecting admins, customers, and partners. It allows admins to manage services and orders, customers to book and track services, and partners to fulfill orders efficiently. Built with Laravel, Flutter, and MongoDB, it enhances user experience and communication across all users.
Name Of Institute :	SDJ International College
Project Guide :	Prof. Bhumika Patel
Prepared By :	Harshal Jariwala
Tools :	Front-End : Flutter Back-End : Laravel Database : MongoDB Development Tools : VS Code, Android Studio Server : Xampp Server Technology : HTML, CSS, Bootstrap, Dart, PHP

2. Environment Description

2.1 Hardware and Software Requirements

➤ **Admin :**

- **Processor :** minimum Intel Dual-Core (Recommended: Intel Core I3)
- **Ram :** minimum 4 GB (Recommended: 8 GB)
- **Storage :** At least 1 GB of free storage space (Recommended: 10 GB or more)
- **Operating System :** Window 10 or Higher
- **Browser :** Chrome, Edge, Brave, Firefox (Recommended: Chrome)
- **Internet Connection :** Wi-Fi (Recommended: 5Ghz Network)

➤ **Customer :**

- **Processor :** minimum Qualcomm Snapdragon 6 Series (Recommended: Qualcomm Snapdragon 8 Series) Or minimum Apple A12 Bionic Chip (Recommended: Apple A14 Bionic Chip)
- **Ram :** minimum 4 GB (Recommended: 8 GB)
- **Storage :** At least 1 GB of free storage space (Recommended: 10 GB or more)
- **Operating System :** Android 10 or Higher (Recommended: Android 13) Or IOS 14 (Recommended: IOS 16)
- **Internet Connection :** Wi-Fi Or Mobile Network (Recommended: 5G Network)

➤ **Partner :**

- **Processor :** minimum Qualcomm Snapdragon 6 Series (Recommended: Qualcomm Snapdragon 8 Series) Or minimum Apple A12 Bionic Chip (Recommended: Apple A14 Bionic Chip)
- **Ram :** minimum 4 GB (Recommended: 8 GB)
- **Storage :** At least 1 GB of free storage space (Recommended: 10 GB or more)
- **Operating System :** Android 10 or Higher (Recommended: Android 13) Or IOS 14 (Recommended: IOS 16)
- **Internet Connection :** Mobile Network (Recommended: 5G Network)

2.2 Technologies Used

I. Backend Framework: Laravel

Laravel is a powerful, open-source PHP framework used for building the Admin Site of the Elite Assist platform. It provides a clean and elegant syntax that facilitates rapid development and maintenance of web applications. Here's why Laravel was chosen for this project:

Key Features of Laravel:

➤ MVC Architecture:

- Model-View-Controller (MVC) is the architectural pattern that Laravel follows. It divides the application into three interconnected components:
 - **Model:** Handles data and database interaction.
 - **View:** Responsible for displaying the user interface.
 - **Controller:** Manages the business logic and connects the Model and View.
- This structure promotes a clean separation of concerns, making the application easier to maintain, test, and scale.

➤ Eloquent ORM (Object-Relational Mapping):

- Laravel's Eloquent ORM allows developers to interact with the database using PHP syntax, making database operations more intuitive and readable. It abstracts complex SQL queries into simple PHP code, enabling faster development.
- Eloquent also supports relationships such as one-to-many, many-to-many, and polymorphic, which is perfect for handling complex relationships between entities like users, services, orders, and partners.

➤ Blade Templating Engine:

- Blade is Laravel's templating engine, allowing developers to write HTML views with embedded PHP. Blade helps to create dynamic and reusable components while keeping the views clean and easy to read.
- Blade supports control structures (loops, if-else) and includes features such as template inheritance, making it ideal for rendering the Admin site's UI.

➤ RESTful API Support:

- Laravel is well-equipped to create RESTful APIs, allowing smooth communication between the backend and mobile apps (Customer and

Partner apps). Laravel's routing system simplifies the creation of REST APIs to handle user requests and serve JSON responses.

➤ **Security:**

- Laravel offers various security features, such as password hashing, data encryption, CSRF protection, and SQL injection prevention. These built-in features make the platform secure and protect sensitive data like user credentials and service details.

➤ **Task Scheduling:**

- Laravel provides a task scheduling feature that helps automate repetitive tasks such as sending reminders, processing payments, or generating reports, making the platform more efficient.

➤ **Routing and Middleware:**

- Laravel's routing system simplifies handling HTTP requests. Developers can easily define routes for specific resources such as services, orders, or partners.
- Middleware is used to filter HTTP requests entering the application. It can be used for tasks like authentication, logging, and input validation.

➤ **Artisan Command Line Interface:**

- Laravel's Artisan CLI helps developers with common tasks such as database migration, seeding, running tests, and managing application logs. Artisan makes the development process faster and more streamlined.

Why Laravel for Elite Assist:

- Laravel's robust features, including Eloquent ORM, RESTful API support, and security features, make it an ideal choice for building a backend that requires scalability, security, and ease of maintenance. Additionally, Laravel's developer-friendly tools, such as Artisan and Blade, speed up the development process.

II. Mobile App Framework: Flutter

Flutter is an open-source UI framework developed by Google, used for building the Customer and Partner mobile applications in Elite Assist. It allows developers to create beautiful, natively compiled applications for mobile (iOS and Android) from a single codebase. Here's why Flutter is the right choice for mobile app development in this project:

Key Features of Flutter:

➤ Cross-Platform Development:

- Flutter enables developers to build a single application for both iOS and Android using a single codebase. This is a significant time-saver and reduces development costs as there's no need to write separate code for different platforms.
- Platform-specific Integration: Even though Flutter uses a single codebase, it still provides deep integration with iOS and Android through platform channels. This allows developers to access platform-specific APIs and features when needed.

➤ Fast Development with Hot Reload:

- Hot Reload is one of Flutter's standout features. It allows developers to see changes in the code immediately in the app without restarting it, significantly speeding up the development and debugging process.

➤ High Performance:

- Flutter apps are compiled to native code, which means they run directly on the device without an intermediary bridge. This results in high performance, similar to native apps.
- Dart, the programming language used by Flutter, ensures high performance through its Ahead-of-Time (AOT) compilation, which optimizes the code before execution.

➤ Rich and Customizable UI:

- Flutter provides a wide range of pre-designed widgets to create a smooth, responsive, and attractive user interface. It offers flexible and customizable widgets that match the design guidelines of both Material Design (for Android) and Cupertino (for iOS).
- With Flutter's widget tree architecture, developers can easily compose complex UIs that can scale to different screen sizes and orientations.

➤ Unified Codebase:

- Since both Customer and Partner apps are built with Flutter, maintaining a unified codebase simplifies future updates and bug fixes. Any feature or bug fix can be applied to both apps simultaneously, reducing the time and effort involved in maintenance.

➤ **Integration with Firebase and Other Services:**

- Flutter has excellent integration with Firebase, a cloud platform that provides services such as authentication, real-time databases, cloud storage, and push notifications. These services are essential for features like real-time order updates, user authentication, and notifications.

Why Flutter for Elite Assist:

- The ability to write code once and deploy it across both iOS and Android saves considerable development time and effort. Flutter's high-performance rendering engine and native-like experience ensure that Elite Assist delivers a smooth, fast, and responsive user experience for customers and partners.

III. Database: MongoDB

MongoDB is a popular NoSQL database used to store data in Elite Assist. Unlike traditional relational databases, MongoDB stores data in a flexible, JSON-like format called BSON (Binary JSON). It is ideal for handling large amounts of dynamic and unstructured data such as user profiles, services, orders, and more.

Key Features of MongoDB:

➤ **Flexible Schema:**

- MongoDB is a schema-less database, meaning data can be stored in documents without predefined structures. This flexibility allows the platform to adapt to changing requirements easily and store varied data types, such as customer preferences, service bookings, and partner details.
- This is particularly useful for a service-based application like Elite Assist, where service types, categories, and partner data might change frequently.

➤ **Document-Oriented:**

- Data is stored in documents (BSON), allowing complex data models to be represented naturally. For example, an order can contain nested information such as customer details, service information, and the assigned partner, all within a single document.

➤ **Scalability:**

- MongoDB is designed to scale horizontally by distributing data across multiple machines. This makes it suitable for growing applications with increasing amounts of data and traffic, such as Elite Assist, which may handle a large number of users and services.

➤ **High Availability:**

- MongoDB supports replication to ensure that data is available even if one server fails. This is crucial for maintaining service continuity, especially in applications that need to ensure users can always access their data, such as booking orders or service histories.

➤ **Aggregation Framework:**

- MongoDB's aggregation framework allows for powerful querying and data processing, making it easy to filter, sort, and aggregate large datasets. This is useful for generating reports or insights, such as order history or service popularity.

➤ **Sharding:**

- For larger datasets, MongoDB provides sharding, which distributes data across multiple servers to improve performance and scalability. This is an important feature for handling high volumes of data and ensuring that queries run efficiently even as the dataset grows.

➤ **Real-Time Data:**

- MongoDB's real-time data processing features ensure that updates are quickly reflected across the platform. For instance, when a partner marks a service as completed, the change is instantly visible to customers.

Why MongoDB for Elite Assist:

- MongoDB's flexible schema and scalability make it a great choice for an application like Elite Assist, which needs to handle a variety of data types, from customer orders to service details, and scale as the platform grows. Its ability to store complex, nested data within a single document also aligns with the dynamic needs of the platform.

3. System Analysis and Planning

3.1 Existing System and its Drawbacks

In many traditional service-based businesses, service management and booking systems are either manual or powered by legacy software. These systems often rely on separate software solutions for managing different aspects of the business, such as customer bookings, partner coordination, and order tracking. While they may be functional, they come with various limitations and challenges that make them less efficient, scalable, and user-friendly.

I. Traditional System for Service Providers:

➤ Existing System:

- **Manual Processes:** Many service providers still use manual systems, such as paper logs or spreadsheets, to manage customer bookings, service assignments, and order history. These systems require a significant amount of human intervention, are prone to errors, and lack centralization.
- **Multiple Disparate Systems:** Service providers may rely on different software tools for managing bookings, invoices, customer support, partner coordination, and more. These tools often don't integrate well with each other, requiring manual data entry and reconciliation.
- **Phone or Email Communication:** Many service providers still use phone calls or emails for customer communication and service bookings. This process is inefficient and prone to misunderstandings, especially in high-demand situations.

➤ Drawbacks:

- **Inefficiency:** The manual handling of data leads to time-consuming processes and higher chances of human error, such as booking conflicts, missed appointments, or incorrect service details.
- **Lack of Centralization:** Using multiple tools for different tasks creates data silos, which makes it difficult to access critical information, such as service availability, customer details, and service history, from a single source of truth.
- **Poor Customer Experience:** With communication happening via phone calls or emails, it's easy for customers to experience delays or misunderstandings. There's also no streamlined process for booking, tracking, or managing orders.
- **Limited Reporting:** Legacy systems often lack advanced reporting features or require manual effort to generate insights, which limits the ability to analyze business performance or optimize operations.

II. Existing Customer Booking Systems

➤ Existing System:

- **Website or Basic App-based Booking:** Many businesses have a website or a basic mobile application that allows customers to book services. These systems often allow customers to select services, but the process is often limited in terms of flexibility, scheduling options, and user experience.
- **Limited Service Categories:** Customers may have limited options to filter and select services according to their needs, as service categorization may be rigid, requiring manual updates by admins.
- **Coupon and Discount Management:** Most traditional systems do not allow customers to easily apply coupons or track promotions, and the process often requires admin intervention for discounts or approvals.
- **No Partner Interaction:** Traditional systems often lack direct communication between customers and service providers (partners), meaning that customers don't know much about their service provider's availability or capabilities until after the booking.

➤ Drawbacks:

- **Complex Booking Process:** Customers often face a lengthy or complicated process to book services, which can deter them from completing bookings. Additionally, without real-time updates on service availability, customers may encounter booking conflicts or disappointment.
- **Limited Interaction with Partners:** In traditional systems, there's a lack of direct communication or transparency between customers and service providers. This can result in misunderstandings regarding the service, expectations, or scheduling.
- **No Real-time Updates:** If services are canceled or rescheduled, customers are often not notified in real-time, leading to confusion and dissatisfaction.
- **Difficulty with Coupon Application:** In many legacy systems, the process of applying and managing coupons or discounts is either cumbersome or unavailable, leaving customers without any easy way to save on services.

III. Partner or Service Provider Management Systems

➤ Existing System:

- **Manual Assignment of Orders:** In traditional systems, orders are assigned manually by admins or managers. Service providers (partners) typically receive orders via phone calls, emails, or paper-based communication, which creates inefficiency and potential for errors.

- **Lack of Integration:** Many partner systems do not integrate with the main booking or customer management system, making it difficult for partners to track their orders, view customer information, or manage their own schedule.
- **Service Completion Tracking:** Partners often lack tools to easily update service statuses. Instead, they might rely on external communication to notify the admin when a service is completed or delayed.

➤ **Drawbacks:**

- **Inefficient Order Assignment:** Manual assignment of orders can be time-consuming and prone to mistakes, such as assigning the wrong service provider or forgetting to assign an order altogether.
- **Poor Communication:** Partners do not have a central platform to manage customer interactions, track orders, or see real-time updates on their work schedule. They often rely on outdated communication methods (e.g., calls or texts) to receive information.
- **Lack of Visibility:** Service providers may not be able to access complete customer details, service requirements, or order history, leading to service delays, lower quality, and unsatisfied customers.
- **Difficulty in Performance Tracking:** Without integrated reporting tools, service providers find it hard to track performance metrics, such as service completion times, customer feedback, or earnings.

IV. System for Admins (Backend System)

➤ **Existing System:**

- **Manual Management of Services:** Admins often handle service addition, categorization, and partner management manually through internal tools or spreadsheets. Any updates to services, categories, or promotions require manual entry in the backend.
- **No Centralized Platform:** Admins may struggle to have a centralized view of all customer orders, partner performance, and real-time updates about bookings and service statuses.
- **Manual Order Assignments:** Admins manually assign orders to partners and often don't have automated tools for tracking whether orders are completed, delayed, or pending.

➤ **Drawbacks:**

- **Time-Consuming Management:** Admins spend a lot of time manually managing services, categories, and order assignments. This limits their ability to focus on more critical tasks like strategic decision-making and growth.
- **Lack of Real-Time Data:** Admins often do not have access to real-time data on service bookings, order statuses, or partner availability. This makes it difficult to handle issues promptly and improve customer satisfaction.
- **Error-Prone:** Since much of the process is manual, it is more prone to human error. For example, an admin might assign an order to the wrong partner or forget to update a service status.
- **Limited Analytics:** Legacy systems typically offer basic reporting features, making it difficult for admins to gather meaningful insights about business performance, customer behavior, or service efficiency.

V. General Drawbacks of the Existing System

➤ **Overall Problems with Legacy Systems:**

- **Fragmented System:** Existing systems often involve the use of separate software for different tasks (e.g., service booking, order management, customer communication). This creates data silos and makes it difficult for admins, customers, and partners to collaborate or access information in real-time.
- **Limited Scalability:** Legacy systems are often not built to handle the growth of the business or accommodate future features like new services, geographic expansion, or increased customer demand.
- **Poor User Experience:** Many legacy systems offer limited functionality or outdated user interfaces, leading to friction for customers, partners, and admins. This results in lower user adoption and dissatisfaction.
- **Lack of Automation:** Many aspects of the existing system still require manual intervention (e.g., order assignment, service tracking), leading to inefficiency and slower response times.

3.2 Feasibility Study

A Feasibility Study is a critical evaluation of the Elite Assist platform's practicality, cost-effectiveness, and potential challenges. This study covers four main areas: Technical Feasibility, Operational Feasibility, Economic Feasibility, and Legal Feasibility.

Technical Feasibility focuses on assessing whether the proposed system can be developed and implemented with the available technology and infrastructure. The project is based on a robust technology stack. The backend framework is Laravel, a widely used PHP framework known for its scalability, security, and easy-to-use features. Laravel's built-in support for Eloquent ORM, RESTful APIs, and Blade templating make it an ideal choice for the Admin site, providing efficient service management and handling user requests seamlessly. The mobile apps for customers and partners are developed using Flutter, a cross-platform framework that enables the creation of high-performance apps for both Android and iOS from a single codebase. This ensures consistency in user experience across platforms and speeds up development time. The choice of MongoDB as the database is strategic, as it allows for scalable and flexible data management, suitable for the dynamic nature of the data, such as customer profiles, service orders, and partner details. To ensure high availability and scalability, cloud hosting services like AWS or DigitalOcean will be used, with MongoDB hosted on MongoDB Atlas for automatic scaling and backup management. The system architecture will be built on RESTful APIs to allow smooth integration between the backend and the mobile apps, enabling users to perform actions like booking services, viewing available services, and updating order statuses.

From a technical standpoint, the system is feasible as all chosen technologies are reliable, scalable, and well-supported by the developer community. Additionally, they allow for easy integration and future upgrades, ensuring the system can grow with the business's needs.

Operational Feasibility assesses how well the proposed system will operate in the real-world environment and whether it meets the needs of users. For the Admin site, Laravel's backend will provide an intuitive dashboard that allows easy management of services, customer orders, partner assignments, and coupon promotions. The Customer mobile app will offer a smooth and straightforward interface to browse services, book appointments, and apply coupons, while providing real-time updates on the status of their orders. The Partner mobile app will allow service providers to view assigned orders, track customer details, and update the status of completed services, ensuring seamless communication between customers and service providers. The platform will automate many manual processes, such as order assignments, tracking service completion, and notifying users, reducing the workload on admins and improving service efficiency.

The user experience is a critical aspect of operational feasibility. The design of the platform is intended to be user-friendly, with easy navigation for customers and partners and efficient service management for admins. The system will include automated notifications, reminders, and updates, enhancing the experience for all users involved. Additionally, training for admins, customers, and partners will be provided to ensure they are equipped to use the platform effectively. Customer support will also be available to address any technical issues or inquiries, ensuring smooth operations and quick problem resolution.

In terms of operational feasibility, the project is well-positioned to streamline service provider management, enhance communication, and improve the overall efficiency of the service booking process. By centralizing service management, communication, and order tracking into one platform, the project will reduce manual intervention and ensure real-time information flow between customers, service providers, and admins.

Economic Feasibility evaluates the financial aspects of the project, including the costs of development, deployment, maintenance, and expected returns. Development costs include building the backend using Laravel, designing and developing the mobile apps with Flutter, setting up cloud hosting, and integrating third-party services such as payment gateways and push notifications. The costs also include ongoing maintenance, such as bug fixes, performance monitoring, and updates to the platform.

The economic benefits of the platform are substantial. By automating manual processes, the system reduces operational costs for the business, such as administrative overhead and order management. The platform also enhances customer satisfaction through better service delivery, potentially increasing customer retention and repeat bookings. By providing partners with a streamlined system for managing orders and interacting with customers, the platform boosts service efficiency, reducing service completion times and increasing the number of orders completed. These improvements can directly lead to higher revenues for the service provider and partners alike. Moreover, the mobile apps' user-friendly interface encourages adoption, resulting in a broader user base.

In terms of financial returns, the project is expected to generate a significant return on investment (ROI) through increased customer bookings, better service management, and expanded partner participation. The cost of development is justified by the efficiencies gained, both for the service provider and the end-users, ultimately enhancing profitability.

From an economic perspective, the project is financially viable and can lead to substantial cost savings and revenue generation. With the right marketing and user adoption strategy, the platform has the potential to scale significantly.

Legal Feasibility examines the legal aspects of the project, such as data protection, privacy laws, and regulatory compliance. Since the platform will handle sensitive user information, including personal details and payment data, compliance with data protection regulations, such as the General Data Protection Regulation (GDPR) in Europe, is crucial. The system will implement data encryption and other security measures to ensure that customer and partner data is securely stored and transmitted.

The platform will also comply with applicable payment gateway regulations, ensuring that financial transactions are processed securely and in accordance with industry standards. For partners, the system will include features that allow for contract management and service agreements, ensuring legal compliance in terms of service provision and payment terms.

3.3 Requirement Gathering and Analysis

The process of Requirement Gathering and Analysis for the Elite Assist platform is essential for ensuring that the system meets the needs of all stakeholders involved, including Admin, Customer, and Partner. This process involves understanding the functional and non-functional requirements, business goals, and constraints of the project. By gathering requirements from different sources, we can accurately define the features, workflows, and technical specifications that will guide the development of the platform.

The first step in requirement gathering involves identifying and understanding the needs of the key stakeholders: Admins, Customers, and Partners. Each of these user groups has distinct needs that must be addressed by the system.

For Admins, the platform must provide a central management dashboard to control and monitor various aspects of the service provider system. This includes managing services, assigning orders to partners, viewing customer data, tracking service history, and offering promotions through coupons. Admins also require an efficient system for managing the onboarding process for partners and handling customer queries. The Admin's primary focus is on control, visibility, and decision-making, so the platform must allow for real-time updates, automated workflows, and comprehensive reporting tools.

Customers require a streamlined and intuitive experience when browsing services, booking appointments, and interacting with partners. They need a system that provides clear information about available services, their pricing, and real-time availability of partners. The ability to filter services by categories, apply coupons for discounts, and view order history are essential features. Customers should also receive timely updates and notifications regarding their bookings, and they must have easy access to partner details to make informed choices. Most importantly, they want a hassle-free booking process and responsive communication from both the platform and service providers.

For Partners, the system must allow for easy access to assigned orders, the ability to view customer information, and the functionality to update service statuses. Partners need a platform that helps them efficiently manage their appointments and communicate with customers. They also need real-time notifications about new assignments, cancellations, and customer feedback. The system should support service completion tracking, enabling partners to mark tasks as completed, which will be reflected on the customer's order history.

➤ Functional Requirements

Functional requirements define the specific actions the system must support to meet the needs of the stakeholders. These include the various services, processes, and interactions that the platform should facilitate.

For the Admin site, the system must allow for the management of categories and subcategories for different services, the creation and editing of service listings, and the assignment of partners to customer orders. Admins must be able to apply and manage coupons, monitor order statuses, and generate detailed reports on the

business's performance. Additionally, the system must include the ability to review customer feedback and ratings for partners to maintain service quality.

For the Customer mobile app, the system should enable customers to browse services by category or subcategory, view available time slots, and book appointments with partners. Customers must be able to apply promotional coupons, view past orders, and track their order status. Notifications regarding appointment confirmations, cancellations, or service updates are crucial for keeping customers informed. Customers should also be able to view the profiles of partners, which will include ratings, reviews, and service details.

For the Partner mobile app, partners must have the ability to view and manage their assigned orders, update service statuses (e.g., service completed), and communicate with customers regarding specific order details. The platform should allow partners to access customer feedback, allowing them to improve service delivery. Partners should also be able to manage their availability, which will be synced with the customer booking system to prevent overbooking.

➤ **Non-Functional Requirements**

Non-functional requirements define the system's performance, usability, scalability, and reliability.

The system must be scalable to accommodate future growth, including the addition of new services, partners, and users. Given that the platform is designed to serve multiple user types (Admins, Customers, and Partners), it needs to handle a large volume of concurrent users and requests. This scalability should be supported by the cloud hosting infrastructure, ensuring high availability and load balancing, particularly during peak times.

In terms of performance, the platform should provide real-time updates for both customers and partners. For example, when a customer books a service, the system should immediately notify the relevant partner. The platform should also provide fast response times for service searches, bookings, and updates, ensuring a smooth and efficient user experience. Push notifications and in-app updates must be delivered without delay.

The system must ensure security and data privacy, especially since it deals with personal information, payment details, and service data. Encryption must be used for sensitive information, and the platform must comply with relevant data protection regulations, such as GDPR, ensuring user data is handled securely. Additionally, the system must have robust user authentication processes, including secure login and account management for customers, partners, and admins.

The platform should have usability at its core, with a simple and intuitive interface for all types of users. The mobile apps must be user-friendly, allowing customers and partners to navigate the platform easily without a steep learning curve. The Admin panel should also be designed for ease of use, enabling Admins to manage services, track orders, and view reports without requiring technical expertise.

The system should be designed for reliability, ensuring minimal downtime and quick recovery in case of any failures. Regular backups and system monitoring will be necessary to maintain smooth operations. Error handling mechanisms should be in place to ensure that users are informed of any issues that arise, such as a failed booking or an issue with a payment.

➤ **System Constraints**

Several constraints must be considered during the development of the Elite Assist platform. These include technical constraints such as ensuring the platform can work seamlessly across different devices and operating systems. Since the mobile apps will be developed using Flutter, compatibility with both iOS and Android is crucial. The backend system must be capable of managing large volumes of data, especially customer and partner information, and must integrate well with external services such as payment gateways and notifications.

From a business perspective, the platform must support the different roles within the organization and provide flexibility in assigning and managing services. Admins need to have complete control over the platform, but at the same time, partners should have sufficient autonomy to manage their work schedules. The platform should also ensure that the customer experience is optimized, making it easy to discover and book services.

Budget constraints must also be considered. While the platform needs to be feature-rich, the development and ongoing maintenance costs should align with the business's financial capabilities. This requires balancing the desired functionality with development resources and ongoing operational costs, ensuring the platform is cost-effective in the long run.

The requirement gathering and analysis phase for the Elite Assist platform involves identifying the needs and expectations of all user groups, translating those needs into detailed functional and non-functional requirements, and ensuring that the project's constraints are considered. By understanding these requirements in detail, the development team can create a platform that delivers value to admins, customers, and partners, providing a seamless service management experience that is efficient, scalable, and user-friendly.

4. Proposed System

4.1 Scope

The scope of the Elite Assist project defines the boundaries and extent of the platform's functionalities, focusing on what will be developed, implemented, and delivered. The platform is a comprehensive service provider management system designed to streamline and improve the interaction between three primary user types: Admins, Customers, and Partners.

The primary objective of the Elite Assist platform is to provide an efficient, user-friendly environment for managing services, bookings, and communication between customers and service providers. It aims to centralize the processes involved in service bookings, order management, partner assignments, and customer interactions. Through a well-structured Admin panel, mobile apps for customers and partners, and integration with a MongoDB database, the platform is designed to cater to the diverse needs of the stakeholders involved.

For Admins, the system will allow for comprehensive management capabilities. Admins will be able to create, modify, and categorize services, assign partners to specific orders, manage customer data, and oversee order processing and completion. They will also be responsible for overseeing promotions and coupons, ensuring that discounts and offers are applied accurately to customer bookings. The Admin will be equipped with powerful tools to monitor performance metrics, manage reports, and gain insights into the platform's operations, providing data-driven decision-making tools.

The Customer side of the platform will be designed to offer a seamless and intuitive service discovery and booking experience. Customers will be able to browse available services, categorized and sub-categorized for easy navigation, and book appointments based on service availability. They will also have the ability to apply coupons for discounts on services and keep track of their order history, giving them full visibility over their past interactions with the platform. Customer notifications will be a key feature, alerting users about booking confirmations, status updates, and other service-related events in real time. The mobile app will ensure that customers can interact with the platform anytime and from anywhere.

For the Partners, the platform will enable easy management of assigned tasks, viewing customer details, and marking service tasks as completed. Partners will be notified of new orders, cancellations, or updates to their existing tasks, and they will be able to communicate directly with customers to coordinate appointments or address service-related questions. The partner interface will be optimized for efficient service delivery, allowing partners to manage their schedules, mark services as completed, and provide feedback to improve the customer experience.

The system will be underpinned by MongoDB, a NoSQL database that offers flexibility and scalability for handling large amounts of dynamic and diverse data. The database will store critical information such as customer profiles, order history, service details, and partner information. The Laravel backend framework will serve as the core of the Admin panel, providing secure, scalable, and efficient data processing and management.

capabilities. The backend will support RESTful APIs, enabling smooth interaction between the backend system and the mobile applications built with Flutter.

The platform's scope will also encompass features like user authentication, ensuring that only registered and authorized users can access certain parts of the platform, and push notifications to keep customers and partners updated on service status changes or new bookings. The system will also provide secure payment gateways for financial transactions related to service bookings, ensuring compliance with industry standards for financial data security.

The project will focus on ensuring that the system is scalable and can grow with the needs of the business. As more partners and customers join the platform, the system will be able to handle the increased load without compromising performance or reliability. Regular updates and maintenance will be built into the development lifecycle to ensure that the platform remains secure and operational as new features are added.

The scope of Elite Assist also includes training for Admins, Customers, and Partners on how to use the platform effectively. This will ensure that users are well-prepared to leverage the system to its full potential. Alongside the development, testing and user feedback will be incorporated to ensure the platform meets the expectations of all user groups.

However, there are certain limitations to the project scope as well. While the platform will support multiple categories of services, it will initially focus on a select set of services to ensure that the system is fully optimized before expanding to more diverse service offerings. Additionally, while the platform will be designed for scalability, the project will be launched in phases, with key features being prioritized for the initial rollout, while additional features and enhancements may be introduced over time based on user feedback and business needs.

In conclusion, the Elite Assist project will deliver a comprehensive platform for managing service bookings, partner assignments, and customer interactions, all while ensuring ease of use, scalability, and efficient management. The scope clearly defines the functionalities and user requirements to be met, setting the stage for successful development and deployment of the system.

4.2 Project Modules Functionalities

The Elite Assist platform consists of several interconnected modules, each designed to provide specific functionalities for the Admin, Customer, and Partner user types. These modules collectively form the core of the platform and enable smooth interaction, service booking, and order management.

I. Admin Panel (Laravel Backend)

- **Dashboard Management:** The Admin panel provides a comprehensive dashboard where admins can monitor overall system activity. It displays key metrics like the number of active orders, new bookings, service performance, and revenue statistics. Admins can get real-time insights into how the platform is performing.
- **Service Management:** Admins can add, edit, or remove services available on the platform. They can categorize services under specific categories and subcategories, ensuring that the offerings are well-organized and easy to browse for customers. This module also includes setting up service details like pricing, descriptions, and availability.
- **Partner Management:** Admins have full control over the onboarding and management of service partners. They can approve or reject partner registrations, assign partners to specific services, and view detailed profiles of all active partners. Admins can also track the performance of each partner by reviewing completed tasks and customer feedback.
- **Order Management:** Admins have the ability to manage customer bookings. They can assign orders to partners based on availability, track order statuses (pending, in progress, completed), and update customers and partners about order progress. Admins also have the ability to cancel or modify orders when needed.
- **Coupon & Discount Management:** Admins can create and manage promotional coupons, offering discounts to customers for specific services. They can set coupon validity periods, define eligible services, and track the use of coupons. This module allows the platform to run promotional campaigns effectively.
- **Reporting & Analytics:** Admins can generate detailed reports on various aspects of the platform, including financial performance, service usage statistics, customer behavior, and partner performance. These reports help in strategic decision-making and understanding the growth and efficiency of the platform.

II. Customer Mobile App (Flutter)

- **Service Discovery:** Customers can browse a wide range of services that are categorized and subcategorized. They can filter services based on parameters like location, price range, and availability. This feature helps customers find services that match their specific needs quickly.
- **Service Booking:** Once a customer selects a service, they can view available time slots, select a preferred time, and book an appointment. The system will automatically assign the service to an available partner based on location and availability.
- **Coupon Application:** Customers can apply available coupons to their bookings to receive discounts. The app will display the applicable coupons for each service, and customers can choose the one that provides the best benefit.
- **Order History:** Customers can view their past orders and services booked through the platform. This history provides details such as the service provider's name, service status, and payment details. Customers can also access order-specific feedback and ratings.
- **Partner Profiles:** Customers can view detailed profiles of the partners assigned to their orders, including ratings, reviews, and the partner's service history. This feature helps customers choose trusted partners and manage expectations.
- **In-App Payments:** Customers can pay for services directly through the app. The payment module integrates with third-party payment gateways, ensuring secure and seamless transactions.

III. Partner Mobile App (Flutter)

- **Order Assignment:** Partners receive notifications about new orders assigned to them, including customer details, service type, and booking time. Partners can accept or reject orders based on their availability.
- **Order Management:** Partners can manage their assigned orders by marking them as "In Progress" or "Completed." This feature allows customers to see the real-time status of their service bookings. Partners can also update the order status if there are any delays or issues with service delivery.
- **Customer Details:** Partners can view customer details related to each order, including name, contact information, and service preferences. This helps partners prepare for the appointment and tailor their service to the customer's needs.
- **Service Completion:** Once a service is completed, partners can mark it as finished. The system will notify the customer and update the order status. The completion status will also trigger a request for feedback from the customer.

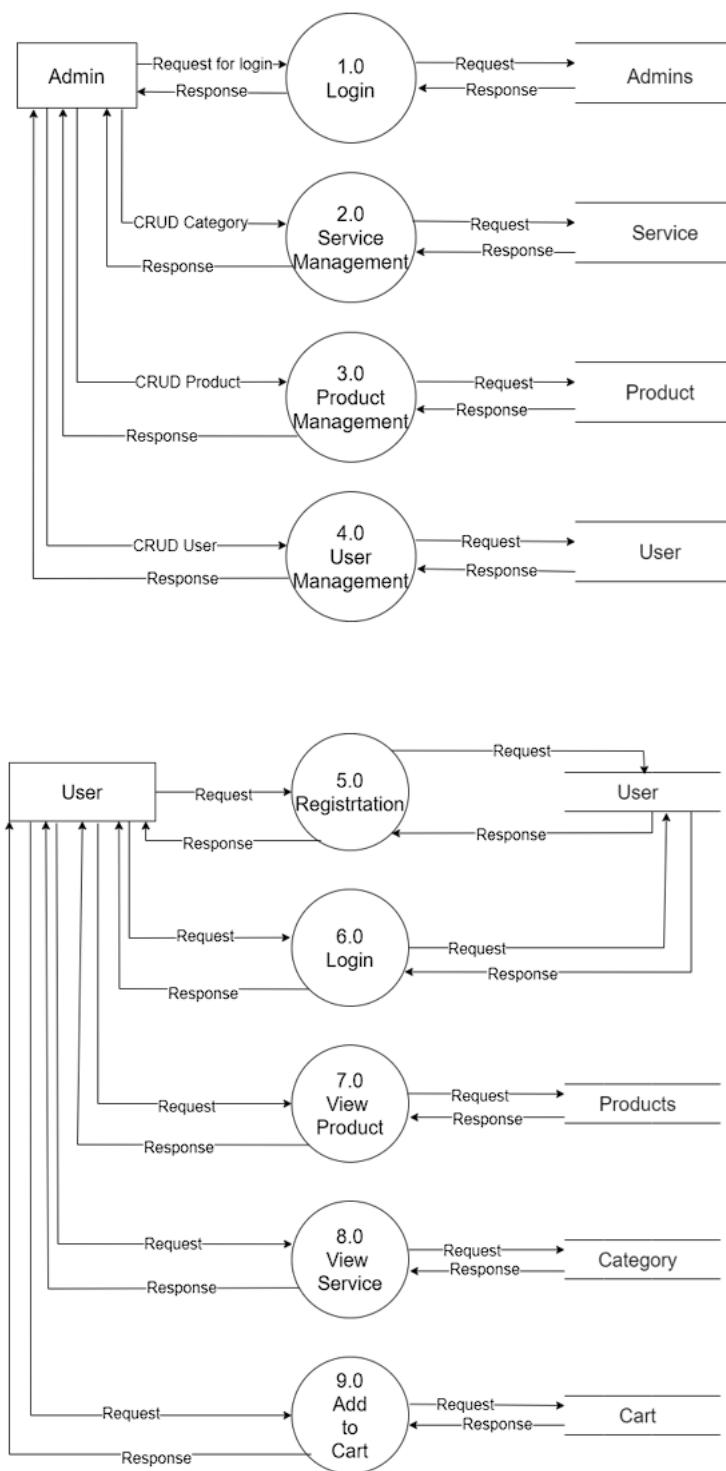
- **Partner Availability:** Partners can manage their availability by updating their working hours. This ensures that they only receive orders during times they are available to provide services. The system automatically syncs available time slots with customer booking requests.

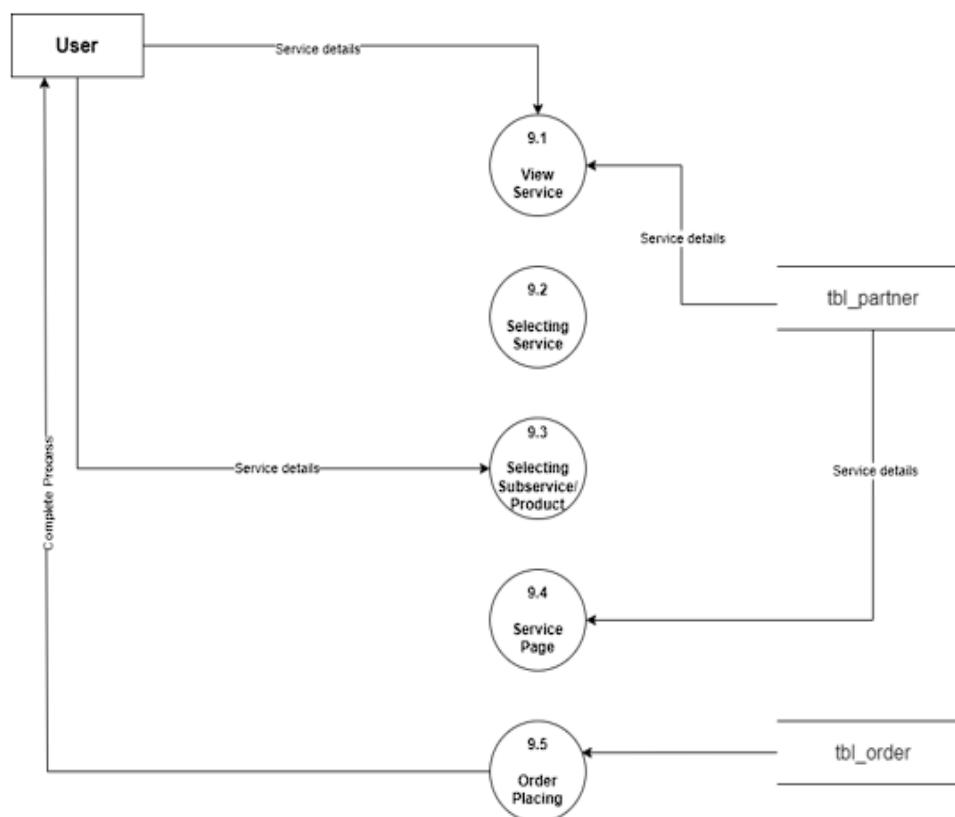
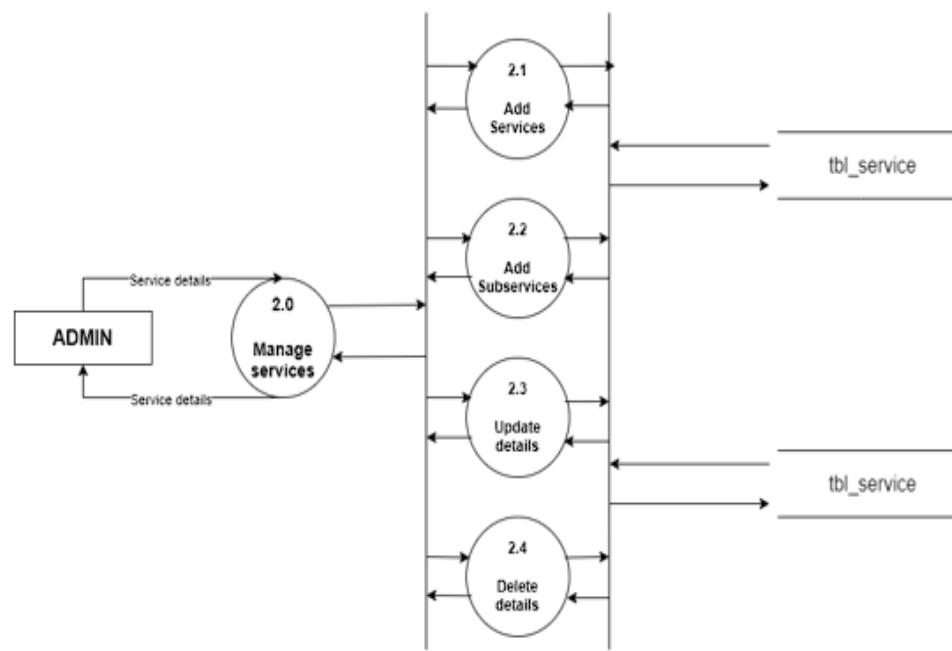
IV. Database (MongoDB)

- **Data Storage:** MongoDB stores a wide variety of structured and unstructured data, including customer preferences, service categories, pricing information, and partner details. The NoSQL nature of MongoDB allows for flexibility in how data is stored and retrieved.
- **Scalability:** MongoDB's flexible schema ensures that as the platform grows and more services, partners, and customers are added, the database can scale accordingly without sacrificing performance.
- **Real-Time Data Access:** MongoDB supports real-time data access, which is critical for providing immediate updates to customers, partners, and admins. This is particularly useful for features such as order status updates, new bookings, and notification triggers.
- **Data Integrity & Security:** MongoDB ensures that the data is securely stored, offering built-in encryption features and compliance with data protection standards like GDPR. User credentials, financial transactions, and personal information are protected using encryption mechanisms and secure access controls.

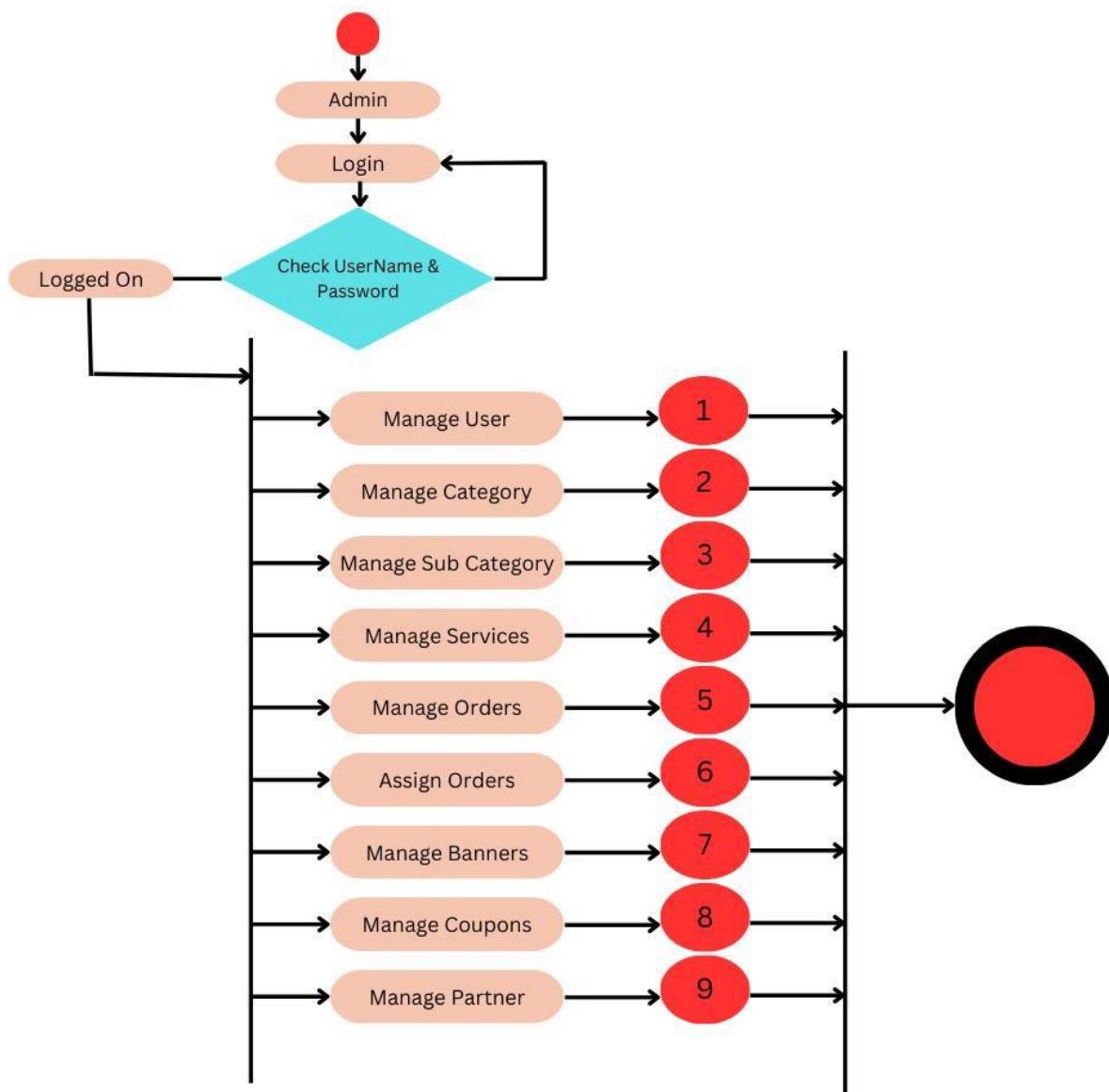
5. Detail Planning

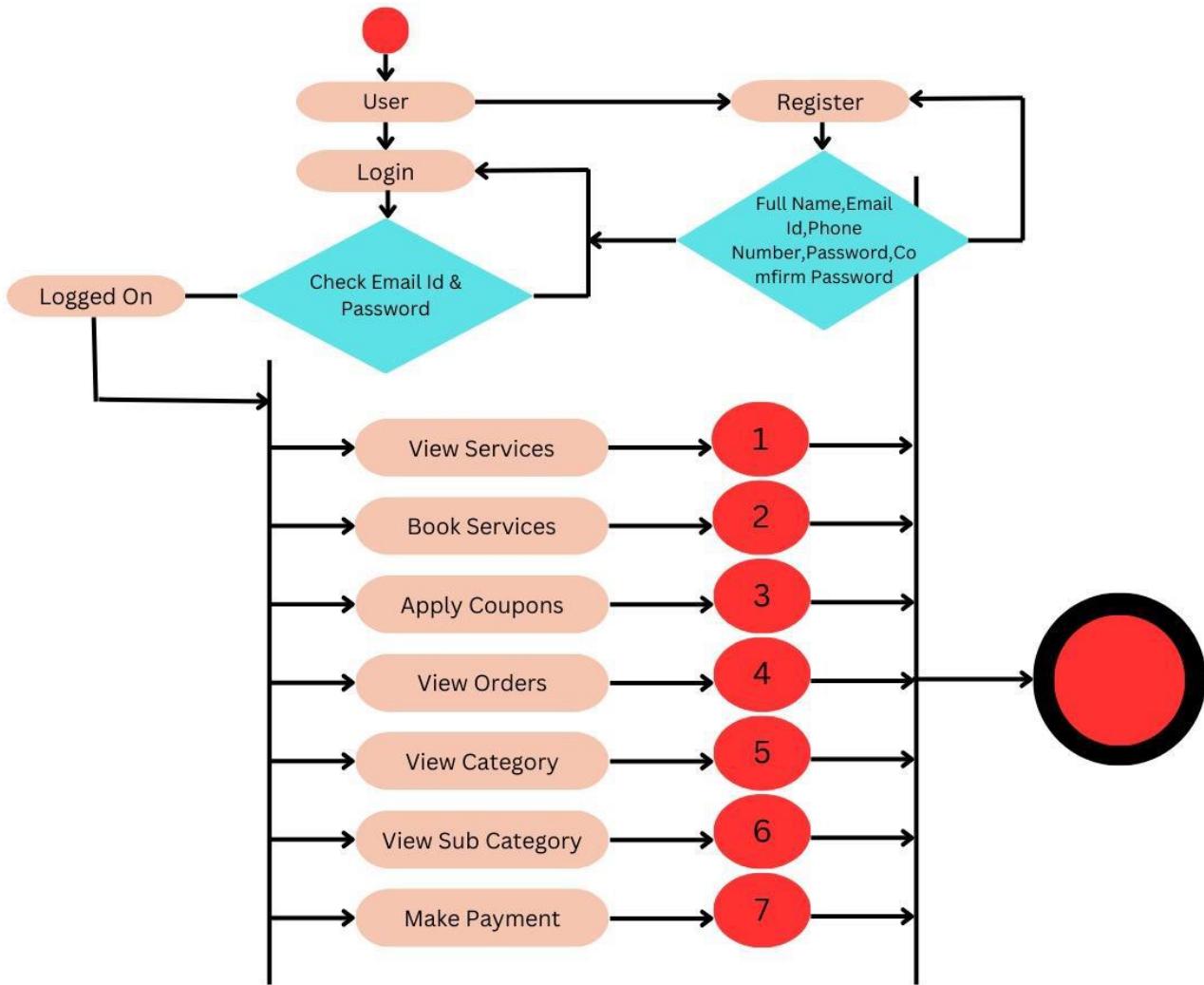
5.1 Data Flow Diagram



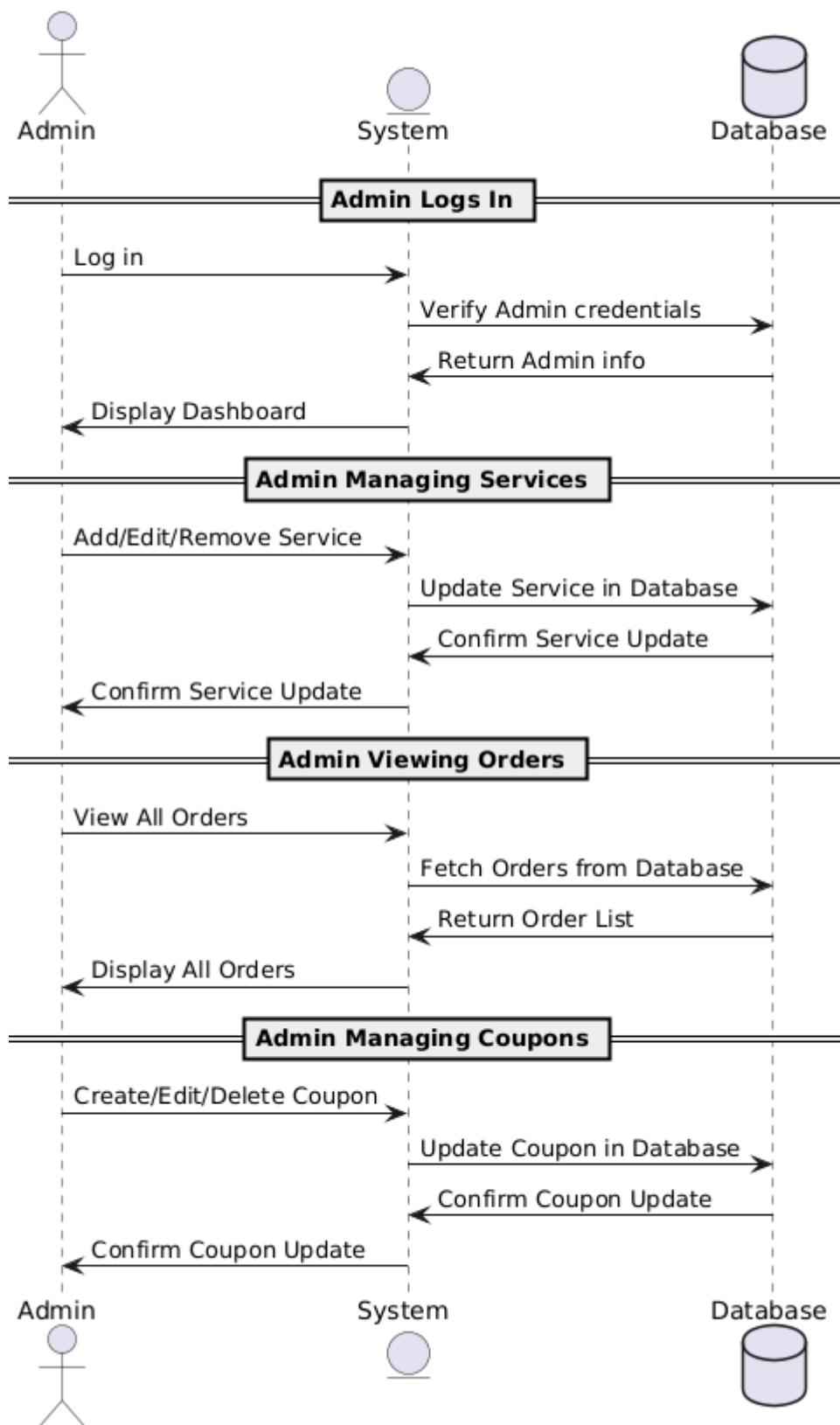


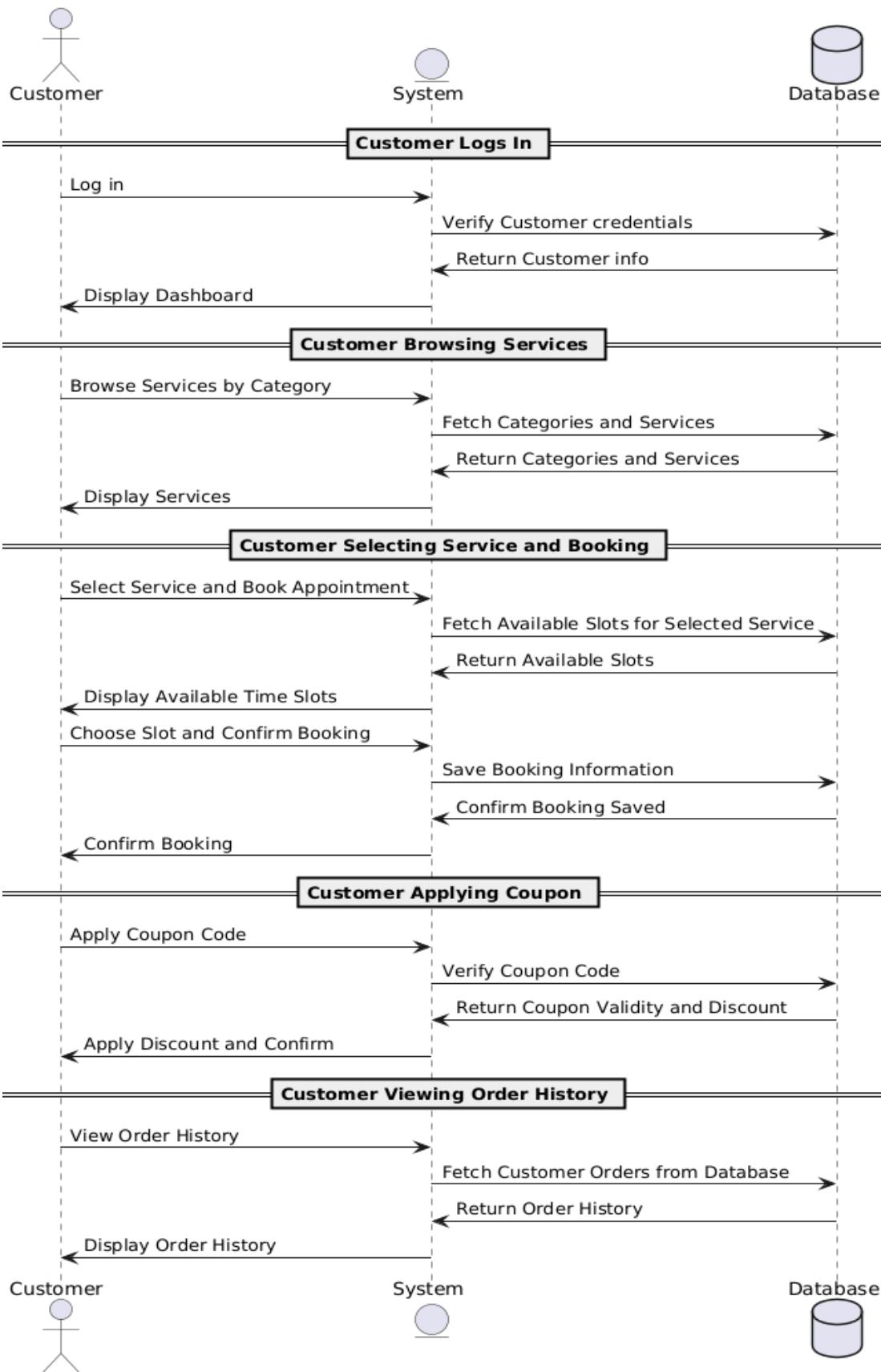
5.2 Activity Flow Diagram

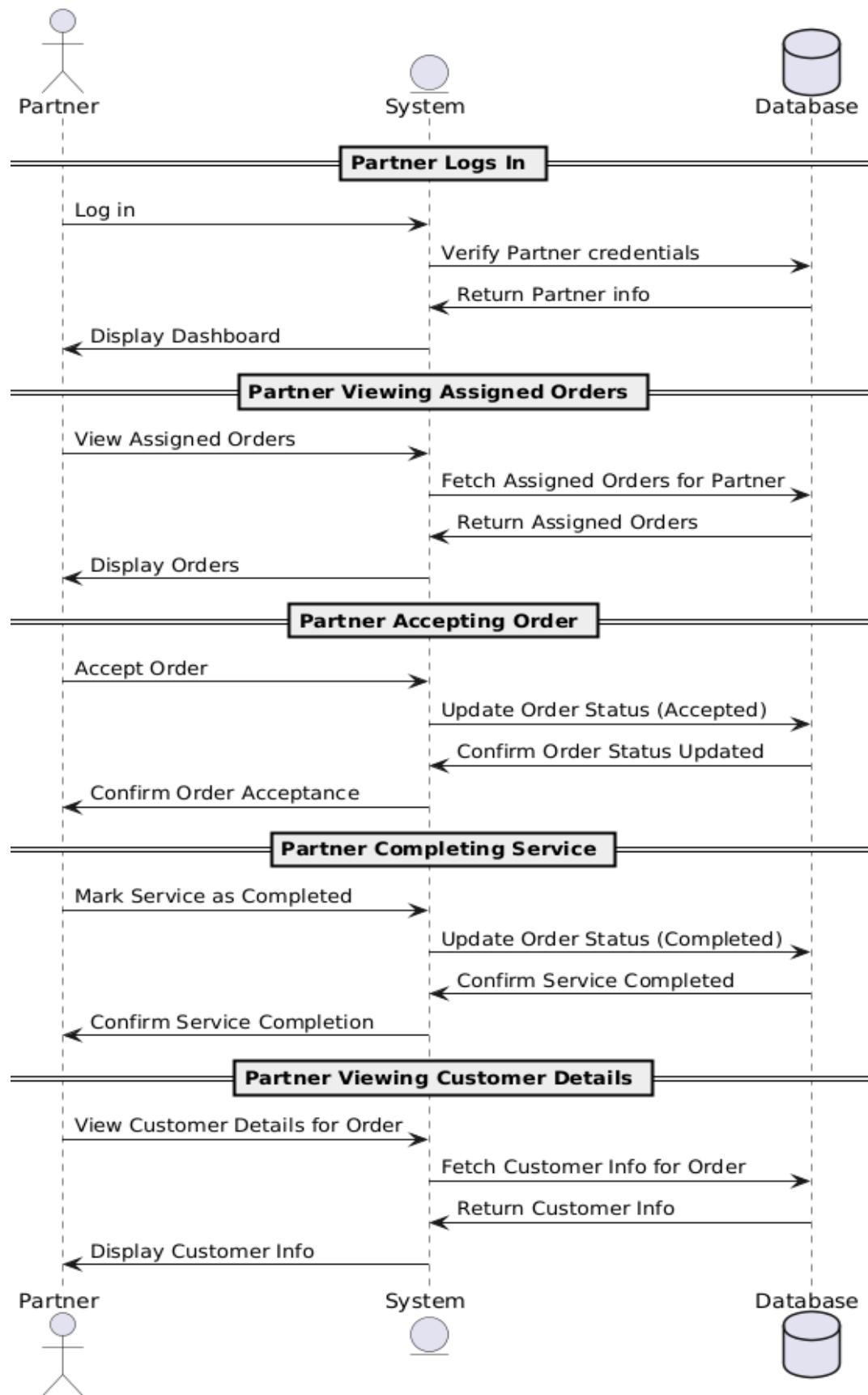




5.3 Entity-Relationship Diagram







6. System Design

6.1 Database Design

➤ Admin Table

No	Field Name	Data Type	Constraints
1.	Id	String	Primary Key Auto Generate
2.	Name	String	Not Null
3.	Email	String	Not Null
4.	Password	String	Not Null
5.	Photo	String	Not Null
6.	Status	Integer	Not Null

➤ User Table

No	Field Name	Data Type	Constraints
1.	Id	String	Primary Key Auto Generate
2.	Name	String	Not Null
3.	Email	String	Not Null
4.	Phone	String	Not Null

➤ Banner Table

No	Field Name	Data Type	Constraints
1.	Id	String	Primary Key Auto Generate
2.	Title	String	Not Null
3.	Status	Boolean	Not Null

➤ Category Table

No	Field Name	Data Type	Constraints
1.	Id	String	Primary Key Auto Generate
2.	Name	String	Not Null
3.	Photo	String	Not Null

➤ Sub-Category Table

No	Field Name	Data Type	Constraints
1.	Id	String	Primary Key Auto Generate
2.	Name	String	Not Null
3.	Photo	String	Not Null
4.	Category_Id	String	Foreign Key

➤ **Service Table**

No	Field Name	Data Type	Constraints
1.	Id	String	Primary Key Auto Generate
2.	Name	String	Not Null
3.	Price	Integer	Not Null
4.	Detail	String	Not Null
5.	Photo1	String	Not Null
6.	Photo2	String	Not Null
7.	Photo3	String	Not Null
8.	Video	String	Not Null
9.	Sub_Category_Id	String	Foreign Key

➤ **User Address Table**

No	Field Name	Data Type	Constraints
1.	Id	String	Primary Key Auto Generate
2.	House_No	String	Not Null
3.	Street	String	Not Null
4.	Landmark	String	Not Null
5.	Area	String	Not Null
6.	City	String	Not Null
7.	State	String	Not Null
8.	Pincode	Integer	Not Null
9.	Type	String	Not Null
10.	User_Id	String	Foreign Key

➤ **Coupon Table**

No	Field Name	Data Type	Constraints
1.	Id	String	Primary Key Auto Generate
2.	Code	String	Not Null
3.	Description	String	Not Null
4.	Discount	Integer	Not Null
5.	Photo	String	Not Null
6.	Status	Boolean	Not Null

➤ **Partner Table**

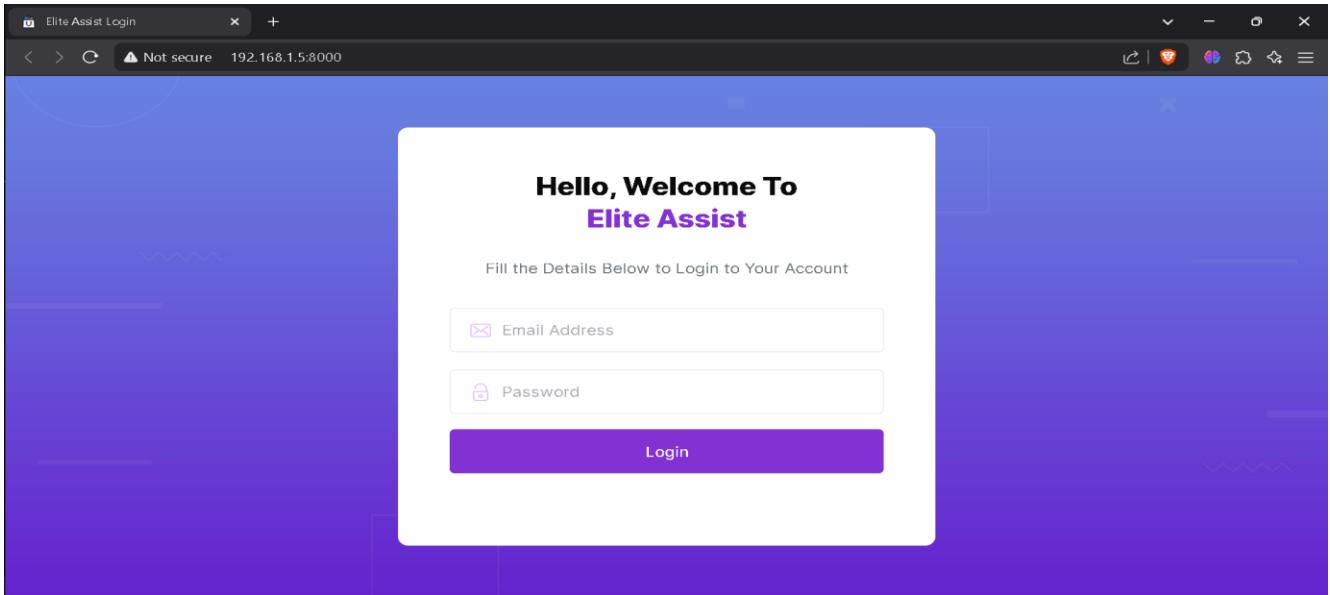
No	Field Name	Data Type	Constraints
1.	Id	String	Primary Key Auto Generate
2.	Name	String	Not Null
3.	Mobile_No	String	Not Null
4.	Email	String	Not Null
5.	Photo	String	Not Null
6.	Aadhar_No	String	Not Null
7.	Aadhar_Photo	String	Not Null
8.	Service_Id	String	Foreign Key

➤ Order Table

No	Field Name	Data Type	Constraints
1.	Id	String	Primary Key Auto Generate
2.	User_Id	String	Not Null
3.	Service_Id	String	Not Null
4.	Amount	Integer	Not Null
5.	Gst_Amount	Integer	Not Null
6.	Discount_Amount	Integer	Not Null
7.	Fees	Integer	Not Null
8.	Total_Amount	Integer	Not Null
9.	Coupon_Code	String	Nullable
10.	Order_Date	String	Not Null
11.	Order_Time	String	Not Null
12.	Service_Date	String	Not Null
13.	Service_Time	String	Not Null
14.	End_Date	String	Not Null
15.	End_Time	String	Not Null
16.	Payment_Type	String	Not Null
17.	Address	String	Not Null
18.	Status	Integer	Not Null
19.	Is_Assign	Boolean	Not Null

6.2 Input Design

➤ Login



➤ Dashboard

The screenshot shows a web browser window titled "Elite Assist Admin" with the URL "192.168.1.5:8000/Dashboards". On the left is a sidebar menu with the "Elite Assist" logo at the top. The menu items include: Dashboard, Banners, Category, Services, Orders, Coupons, Partner, Users Data, Profile, and Logout. The main dashboard area has a light blue header with a profile picture and a three-line menu icon. Below the header, there is a section titled "Dashboard" with three cards: "Total Customers 1", "Total Orders 4", and "Total Revenue 1415". Underneath is a section titled "Newest Orders" with a table:

ID	Product Name	Date	Price	Assigned
1	Ac Service	25-03-2025	299	Pending

➤ Banner

The screenshot shows the 'Banners' section of the Elite Assist Admin interface. On the left, a sidebar menu includes 'Dashboard', 'Banners' (which is selected and highlighted in purple), 'Category', 'Services', 'Orders', 'Coupons', 'Partner', 'Users Data', 'Profile', and 'Logout'. The main content area displays a table titled 'Banners' with columns: Id, Pic, Banner Name, Status, Update, and Delete. Two banners are listed:

Id	Pic	Banner Name	Status	Update	Delete
1		Banner 4	On Air	<button>Edit</button>	<button>Delete</button>
2		Banner 3	On Air	<button>Edit</button>	<button>Delete</button>

A green button labeled '+Add Banner' is located at the top left of the main content area.

➤ Add/Edit Banner

The screenshot shows the 'Add Banner' form. The sidebar menu is identical to the previous screenshot. The main content area has a title 'Add Banners' and contains the following fields:

- Banner Title**: A text input field with placeholder text 'Enter Service Name'.
- Banner Pic**: A file upload field with a 'Choose File' button and a message 'No file chosen'.
- Active Status**: A toggle switch that is currently turned on (blue).
- Add**: A large purple button at the bottom of the form.

➤ Category

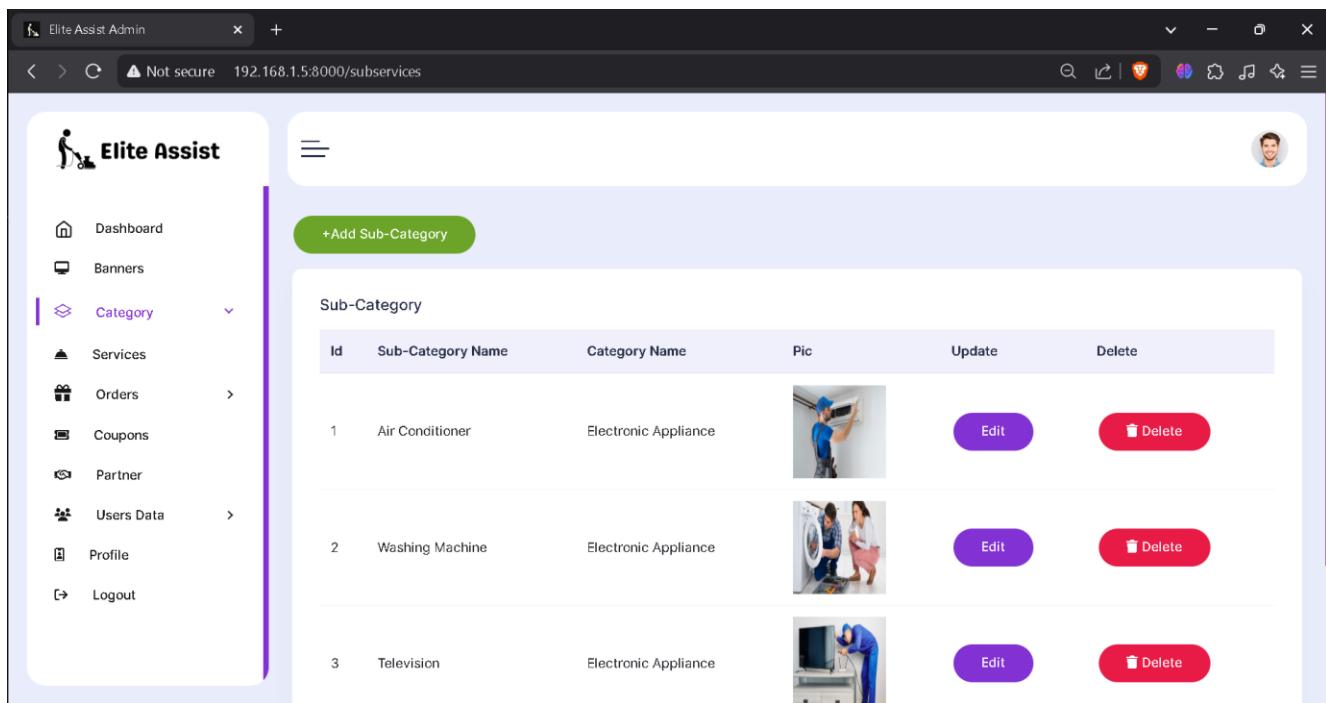
The screenshot shows the 'Category' section of the Elite Assist Admin interface. On the left, a sidebar menu includes 'Dashboard', 'Banners', 'Category' (selected), 'Services', 'Orders', 'Coupons', 'Partner', 'Users Data', 'Profile', and 'Logout'. The main content area displays a table titled 'Category' with columns: Id, Category Name, Pic, Update, and Delete. Three categories are listed:

Id	Category Name	Pic	Update	Delete
1	Electronic Appliance		<button>Edit</button>	<button>Delete</button>
2	Man Salon		<button>Edit</button>	<button>Delete</button>
3	Woman Salon		<button>Edit</button>	<button>Delete</button>

➤ Add/Edit Category

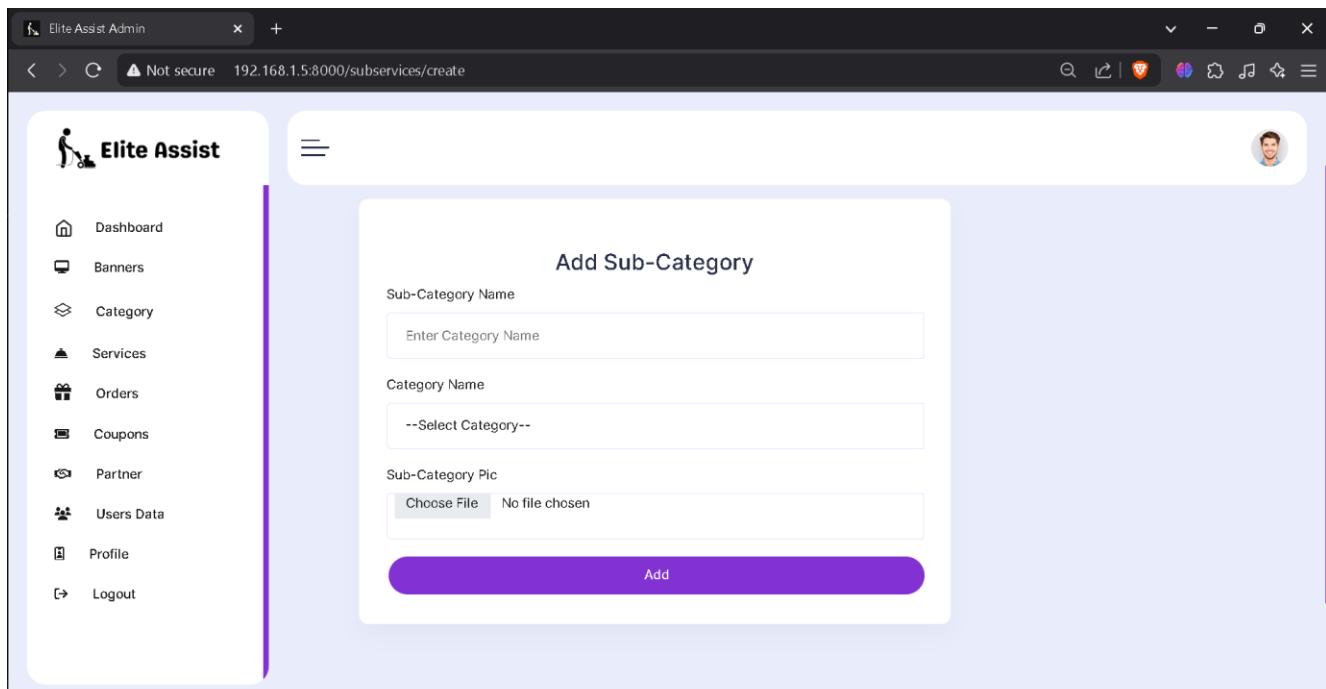
The screenshot shows the 'Add Category' form. The sidebar menu is identical to the previous screenshot. The main content area has a title 'Add Category' and two input fields: 'Category Name' (with placeholder 'Enter Category Name') and 'Category Pic' (with a file upload button 'Choose File' and message 'No file chosen'). A large purple 'Add' button is at the bottom.

➤ Sub – Category



ID	Sub-Category Name	Category Name	Pic	Update	Delete
1	Air Conditioner	Electronic Appliance		<button>Edit</button>	<button>Delete</button>
2	Washing Machine	Electronic Appliance		<button>Edit</button>	<button>Delete</button>
3	Television	Electronic Appliance		<button>Edit</button>	<button>Delete</button>

➤ Add/Edit Sub – Category



Add Sub-Category

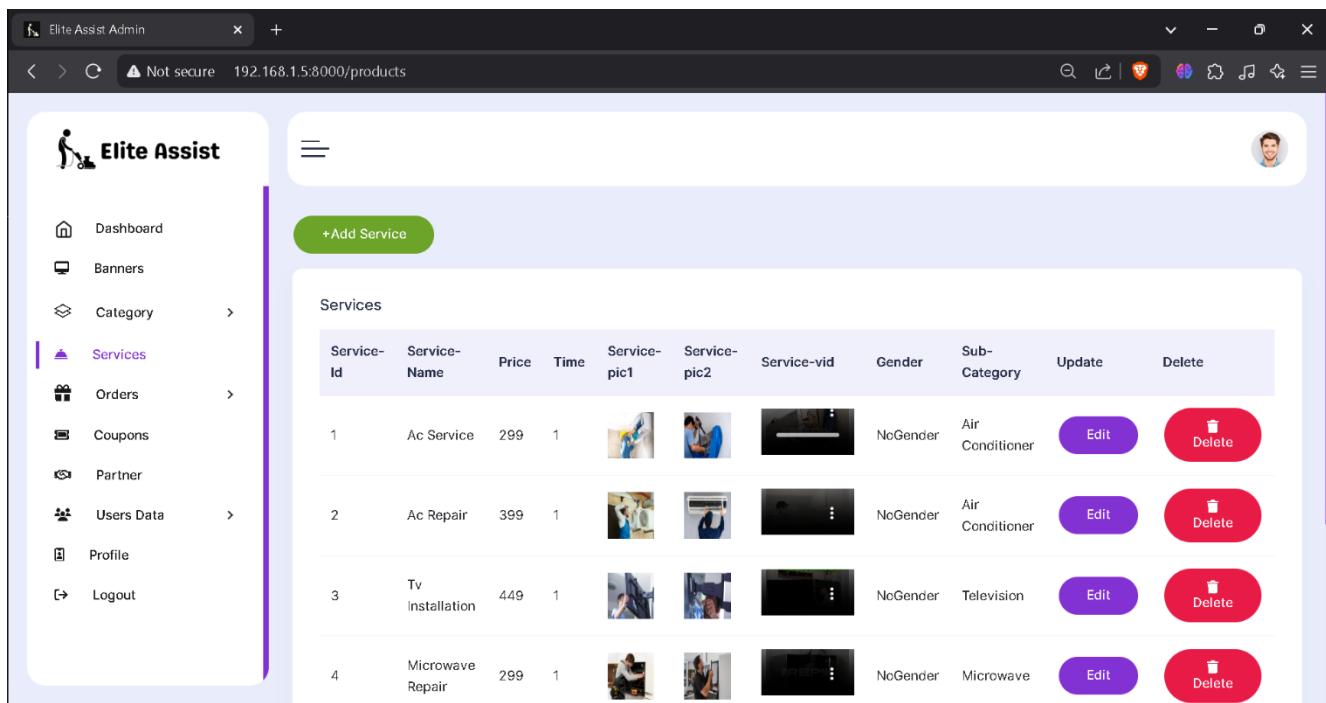
Sub-Category Name

Category Name

Sub-Category Pic
 No file chosen

Add

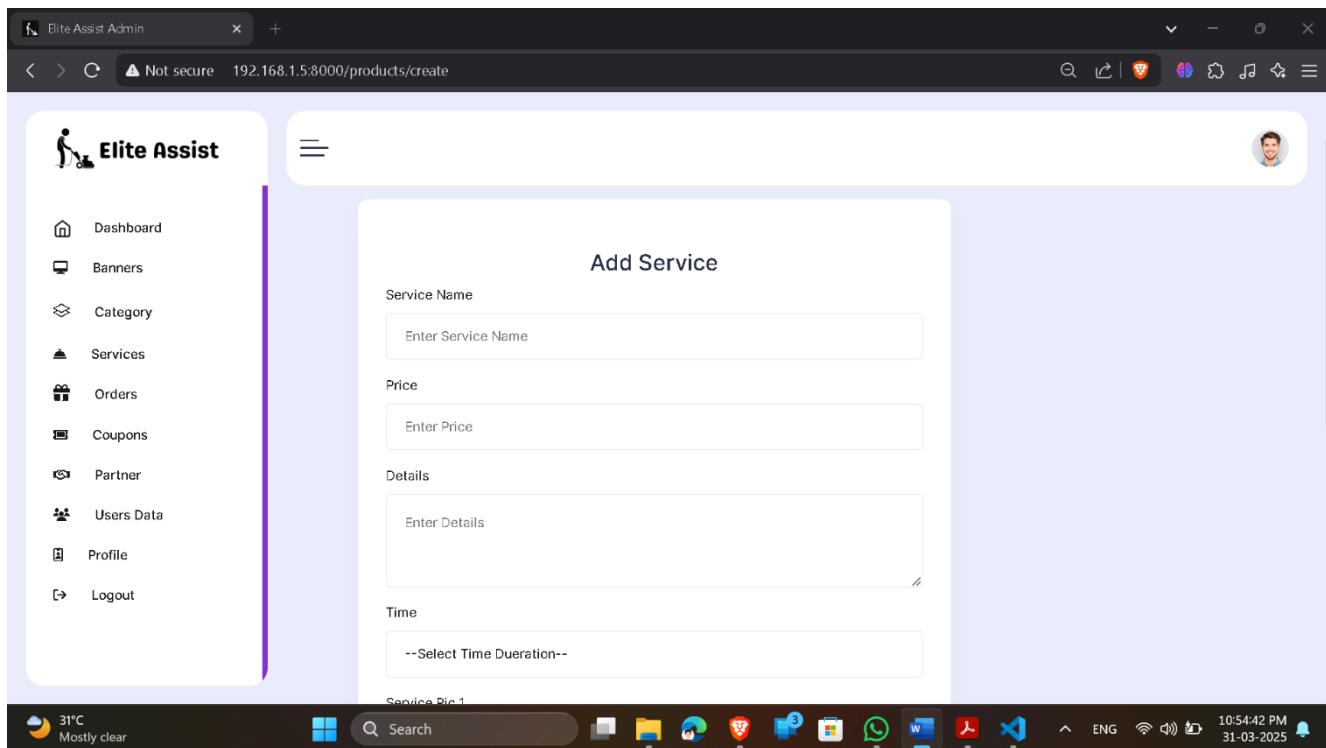
➤ Service



The screenshot shows the 'Services' section of the Elite Assist Admin interface. On the left is a sidebar with navigation links: Dashboard, Banners, Category, Services (selected), Orders, Coupons, Partner, Users Data, Profile, and Logout. The main area displays a table of services with columns: Service-ID, Service-Name, Price, Time, Service-pic1, Service-pic2, Service-vid, Gender, Sub-Category, Update, and Delete. There are four entries in the table:

Service-ID	Service-Name	Price	Time	Service-pic1	Service-pic2	Service-vid	Gender	Sub-Category	Update	Delete
1	Ac Service	299	1				NoGender	Air Conditioner	<button>Edit</button>	<button>Delete</button>
2	Ac Repair	399	1				NoGender	Air Conditioner	<button>Edit</button>	<button>Delete</button>
3	Tv Installation	449	1				NoGender	Television	<button>Edit</button>	<button>Delete</button>
4	Microwave Repair	299	1				NoGender	Microwave	<button>Edit</button>	<button>Delete</button>

➤ Add/Edit Service



The screenshot shows the 'Add Service' form in the Elite Assist Admin interface. The sidebar on the left is identical to the previous screenshot. The main area is titled 'Add Service' and contains four input fields: 'Service Name' (with placeholder 'Enter Service Name'), 'Price' (with placeholder 'Enter Price'), 'Details' (with placeholder 'Enter Details'), and 'Time' (with placeholder '--Select Time Duration--'). At the bottom of the screen, there is a taskbar showing system status: 31°C Mostly clear, a search bar, and various application icons.

➤ Coupon

The screenshot shows the 'Coupons' section of the Elite Assist Admin dashboard. On the left, a sidebar menu includes 'Dashboard', 'Banners', 'Category', 'Services', 'Orders', 'Coupons' (which is highlighted in purple), 'Partner', 'Users Data', 'Profile', and 'Logout'. A green button labeled '+Add Coupon' is at the top right of the main content area. Below it, a table lists a single coupon entry:

Coupon-Id	Coupon-Image	Coupon-Code	Coupon-description	Coupon-Discount	Status	Edit	Delete
1		NEW50	50rs discount	50 RS	On Air	<button>Edit</button>	<button>Delete</button>

Copyright 2025 © EliteAssist All Rights Reserved.

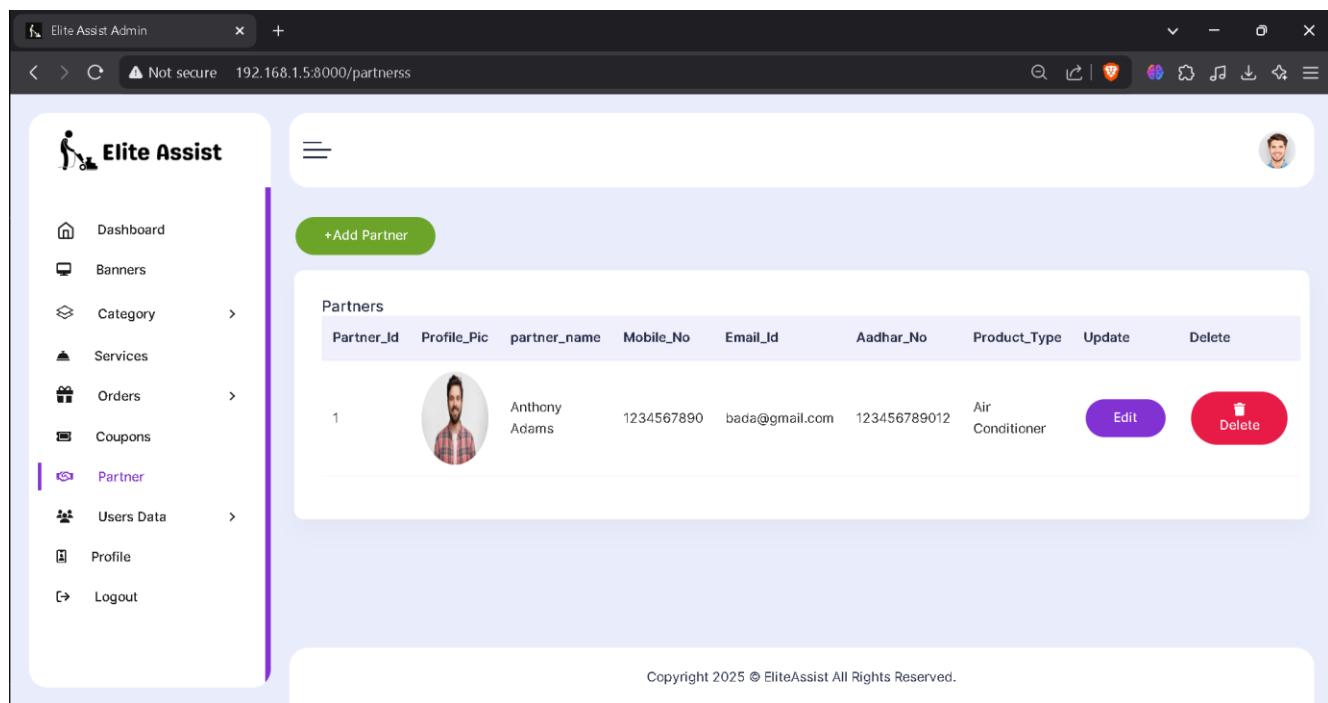
➤ Add/Edit Coupon

The screenshot shows the 'Add Coupon' form. The sidebar menu is identical to the previous screenshot. The main form has a title 'Add Coupon' and four input fields:

- Coupon-Code: A text input field with placeholder 'Enter Coupon-Code'.
- Coupon-Description: A text input field with placeholder 'Enter Coupon-Description'.
- Coupon-Discount: A text input field with placeholder 'Enter Coupon-Discount'.
- Coupon Image: A file upload input field with placeholder 'Choose File' and message 'No file chosen'.

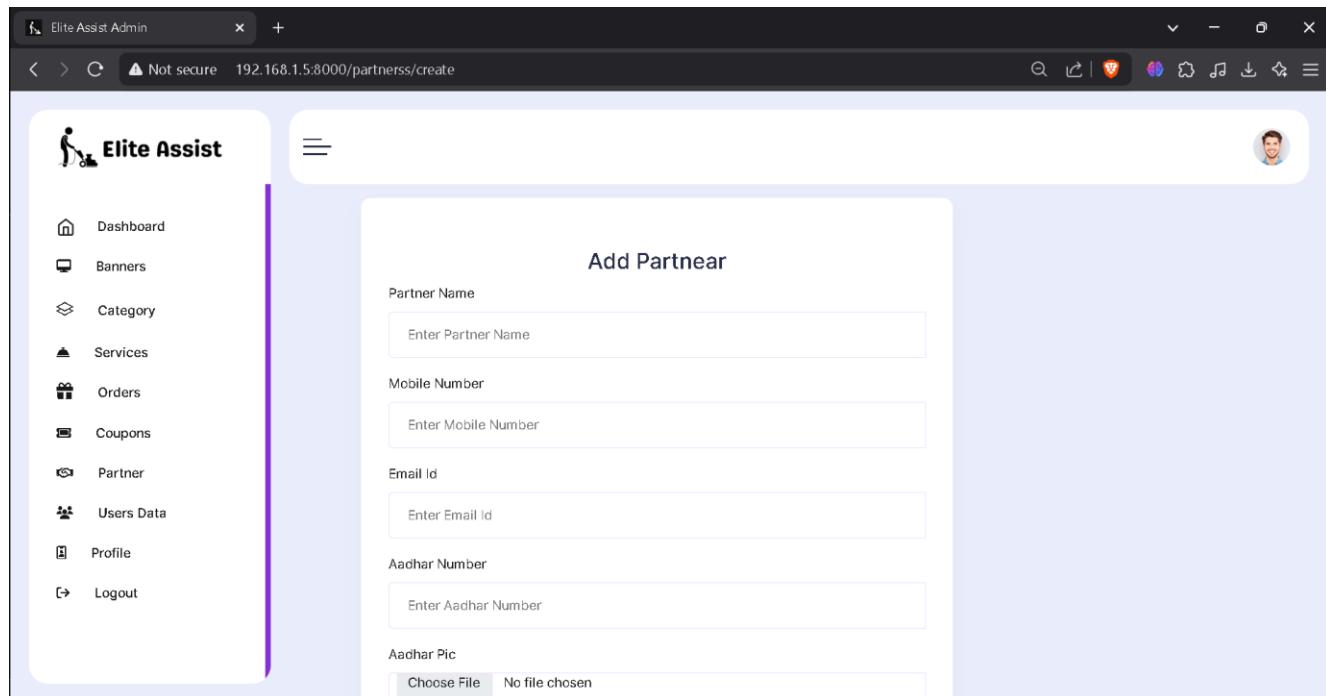
At the bottom of the form is a checkbox labeled 'Active Status'.

➤ Partner



The screenshot shows the 'Partners' section of the Elite Assist Admin dashboard. On the left, there's a sidebar with icons for Dashboard, Banners, Category, Services, Orders, Coupons, Partner (which is selected and highlighted in purple), Users Data, Profile, and Logout. The main area has a header with a user profile icon and a '+Add Partner' button. Below is a table titled 'Partners' with columns: Partner_Id, Profile_Pic, partner_name, Mobile_No, Email_Id, Aadhar_No, Product_Type, Update, and Delete. One row is visible for 'Anthony Adams' with ID 1, mobile number 1234567890, email bada@gmail.com, Aadhar number 123456789012, and product type Air Conditioner. There are 'Edit' and 'Delete' buttons for this row. At the bottom, it says 'Copyright 2025 © EliteAssist All Rights Reserved.'

➤ Add/Edit Partner



The screenshot shows the 'Add Partner' form. The sidebar on the left is identical to the previous screenshot. The main area has a header with a user profile icon and a title 'Add Partnear'. The form contains fields for Partner Name (with placeholder 'Enter Partner Name'), Mobile Number (placeholder 'Enter Mobile Number'), Email Id (placeholder 'Enter Email Id'), Aadhar Number (placeholder 'Enter Aadhar Number'), and Aadhar Pic (a file input field with 'Choose File' and 'No file chosen' labels).

➤ Pending Assign

The screenshot shows the Elite Assist Admin dashboard. On the left, a sidebar menu includes options like Dashboard, Banners, Category, Services, Orders (selected), Coupons, Partner, Users Data, Profile, and Logout. The main content area displays a table titled 'Orders' with one row:

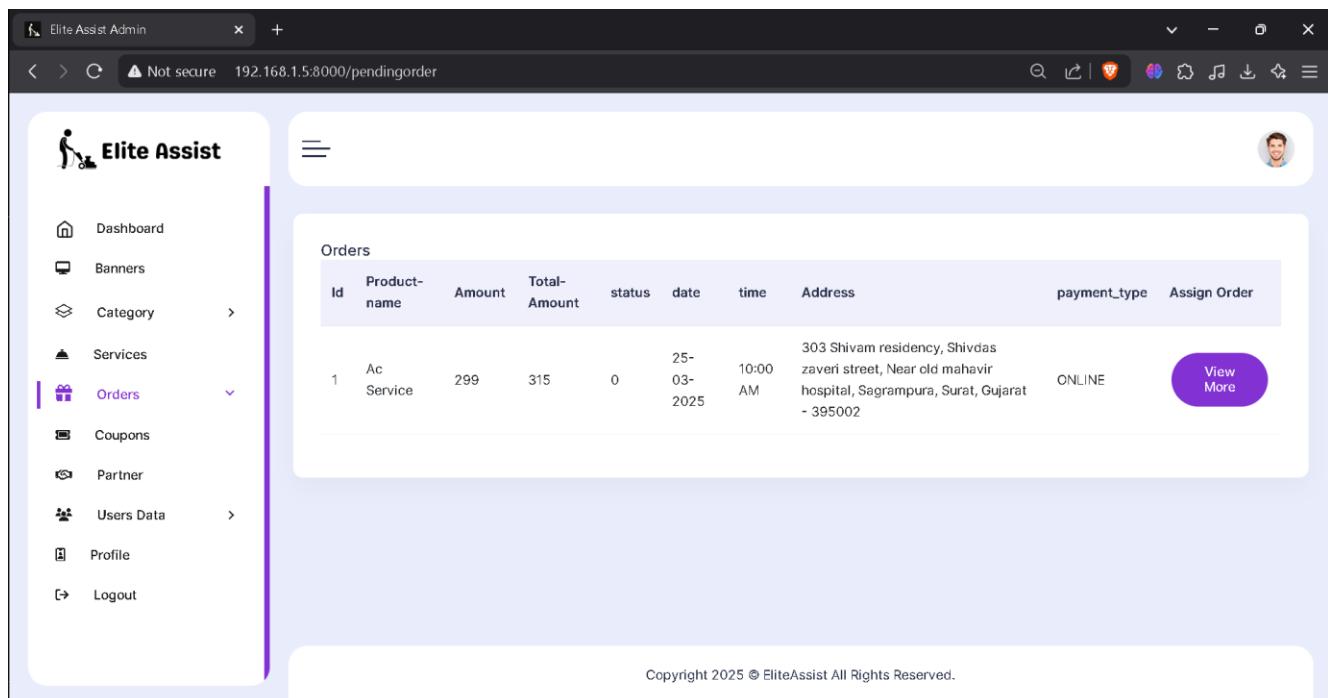
ID	Service-id	Service-name	Amount	status	Date	Time	payment_type	Assign Order
1	67c06e4edd8951a8ed0318f3	Microwave Repair	315	0	31-03-2025	10:00 AM	ONLINE	<button>Assign</button>

At the bottom right of the main area, there is a copyright notice: "Copyright 2025 © EliteAssist All Rights Reserved."

➤ Assign

The screenshot shows the Elite Assist Admin interface with a modal dialog titled 'Assign Partner'. The dialog has a 'Select Partner' label and a dropdown menu containing the placeholder text '--Select Partner--'. At the bottom of the dialog is a large purple 'Assign' button.

➤ Pending Order

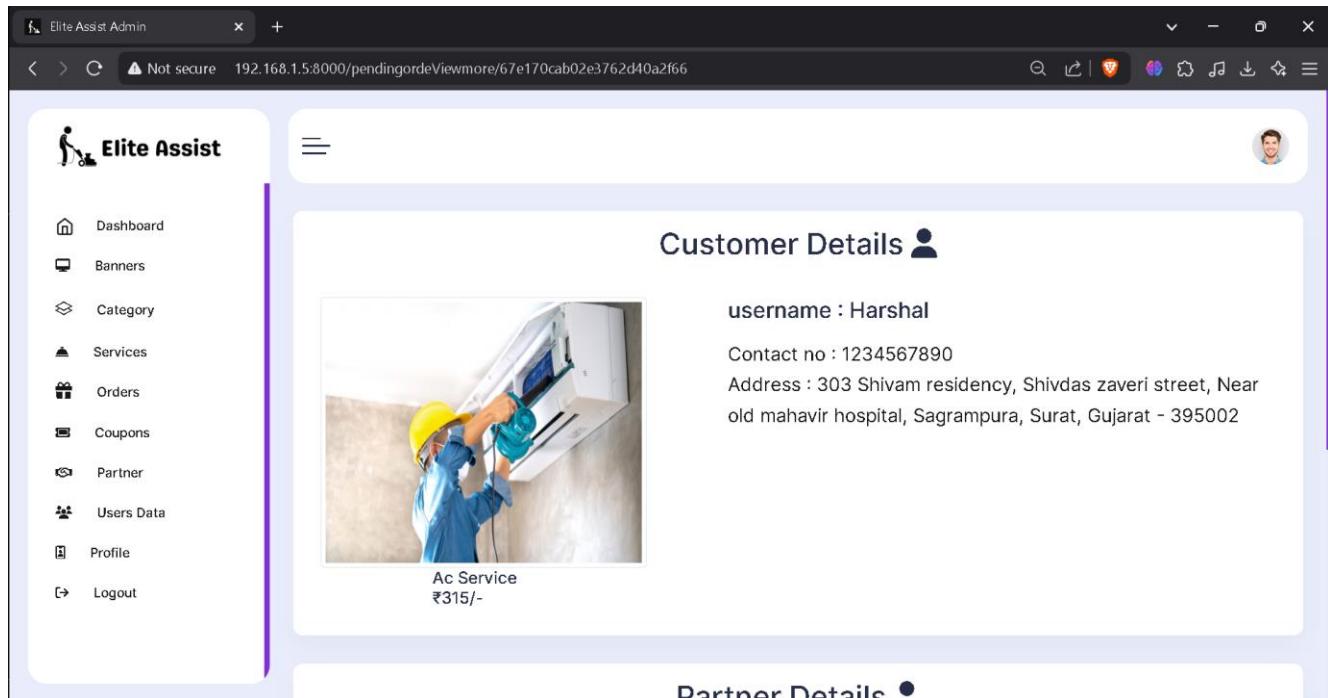


The screenshot shows the 'Orders' section of the Elite Assist Admin dashboard. A single pending order is listed:

ID	Product-name	Amount	Total-Amount	Status	Date	Time	Address	payment_type	Action
1	Ac Service	299	315	0	25-03-2025	10:00 AM	303 Shivam residency, Shvdas zaveri street, Near old mahavir hospital, Sagrampura, Surat, Gujarat - 395002	ONLINE	View More

Copyright 2025 © EliteAssist All Rights Reserved.

➤ Pending Order Details



The screenshot shows the 'Customer Details' section for a pending order. It includes a thumbnail image of a technician working on an AC unit and the service details:

Customer Details 

username : Harshal
 Contact no : 1234567890
 Address : 303 Shivam residency, Shvdas zaveri street, Near old mahavir hospital, Sagrampura, Surat, Gujarat - 395002

Service Details 

Ac Service
 ₹315/-

Partner Details 

➤ Completed Order

The screenshot shows the 'Completed Order' section of the Elite Assist Admin interface. On the left, a sidebar menu includes 'Dashboard', 'Banners', 'Category', 'Services', 'Orders' (selected), 'Coupons', 'Partner', 'Users Data', 'Profile', and 'Logout'. The main content area displays a table titled 'Orders' with columns: Id, Product-name, Amount, Total-Amount, status, date, time, Address, payment_type, and Order Detail. Two completed orders are listed:

ID	Product-name	Amount	Total-Amount	status	Date	Time	Address	payment_type	Order Detail
1	Ac Repair	399	470	1	29-03-2025	6:00 PM	303 Shivam residency, Shivedas zaveri street, Near old mahavir hospital, Sagrampura, Surat, Gujarat - 395002	COD	<button>View</button>
2	Ac Service	299	315	1	27-03-2025	10:00 AM	303 Shivam residency, Shivedas zaveri street, Near old mahavir hospital, Sagrampura, Surat, Gujarat - 395002	COD	<button>View</button>

Copyright 2025 © EliteAssist All Rights Reserved.

➤ Completed Order Details

The screenshot shows the 'Completed Order Details' page for order ID 67e174016688f035ef020c92. The sidebar menu is identical to the previous screenshot. The main content area is divided into two sections: 'Customer Details' and 'Partner Details'.

Customer Details

username : Harshal
Contact no : 1234567890
Address : 303 Shivam residency, Shivedas zaveri street, Near old mahavir hospital, Sagrampura, Surat, Gujarat - 395002


Ac Repair
₹470/-

Partner Details

Partner Name : Bada Bhai
Contact no : 1234567890
Date : 29-03-2025 | Time : 6:00 PM
Payment : COD



➤ Users

The screenshot shows the 'Users' page of the Elite Assist Admin application. The left sidebar contains a navigation menu with items like Dashboard, Banners, Category, Services, Orders, Coupons, Partner, and Users Data (which is currently selected). The main content area displays a table titled 'Users' with columns for User-Id, Username, Email ID, and Mobile No. One row is visible, showing User-Id 67e16a3cb02e3762d40a2f64, Username Harshal, Email ID h@gmail.com, and Mobile No 1234567890. A copyright notice at the bottom right reads 'Copyright 2025 © EliteAssist All Rights Reserved.'

User-Id	Username	Email ID	Mobile No
67e16a3cb02e3762d40a2f64	Harshal	h@gmail.com	1234567890

➤ User Address

The screenshot shows the 'User Address' page of the Elite Assist Admin application. The left sidebar contains a navigation menu with items like Dashboard, Banners, Category, Services, Orders, Coupons, Partner, and Users Data (which is currently selected). The main content area displays a table titled 'User Address' with columns for Address-Id, User Id, Type, and Address. Two rows are visible, both categorized as 'Home'. The first row has Address-Id 67c498fc4139b9a2e01adf3, User Id 67c9d08db0df60628202cd33, and Address 303 shivam residency, shivdas zaveri street, near old Mahavir hospital, sagrampura, surat, gujarat - 395002. The second row has Address-Id 67e16f7fb02e3762d40a2f65, User Id 67e16a3cb02e3762d40a2f64, and Address 303 Shivam residency, Shvidas zaveri street, Near old mahavir hospital, Sagrampura, Surat, Gujarat - 395002. A copyright notice at the bottom right reads 'Copyright 2025 © EliteAssist All Rights Reserved.'

Address-Id	User Id	Type	Address
67c498fc4139b9a2e01adf3	67c9d08db0df60628202cd33	Home	303 shivam residency, shivdas zaveri street, near old Mahavir hospital, sagrampura, surat, gujarat - 395002
67e16f7fb02e3762d40a2f65	67e16a3cb02e3762d40a2f64	Home	303 Shivam residency, Shvidas zaveri street, Near old mahavir hospital, Sagrampura, Surat, Gujarat - 395002

➤ Profile

The screenshot shows the 'Profile' section of the Elite Assist Admin interface. On the left is a sidebar with a logo and navigation links: Dashboard, Banners, Category, Services, Orders, Coupons, Partner, Users Data, Profile (which is selected and highlighted in purple), and Logout. The main content area has a header 'Profile' and a sub-header 'Home https://kamleshyadav.com/ Profile'. It features a circular profile picture of a man with a beard, labeled 'admin Admin'. Below the picture is a table with profile details:

Overview	Edit Profile	Change Password
Profile Details		
Full Name	admin	
email	admin@gmail.com	
Job	Admin	
Country	India	
Username	admin	

At the bottom right of the main content area is a copyright notice: 'Copyright 2025 © EliteAssist All Rights Reserved.'

➤ Edit Profile

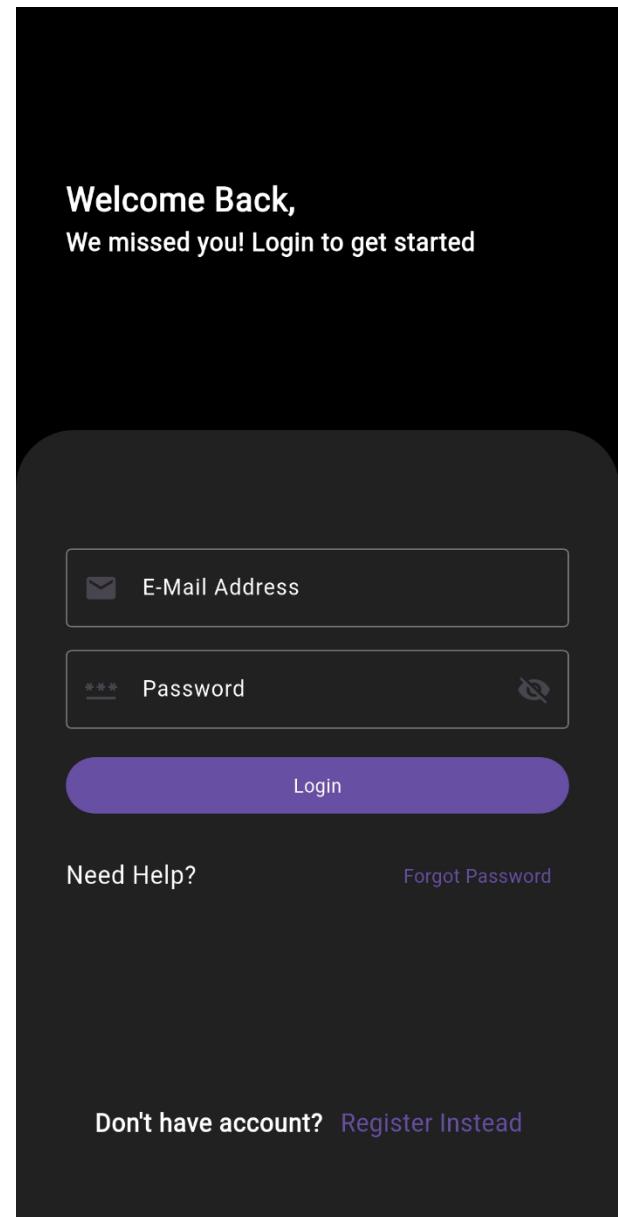
The screenshot shows the 'Edit Profile' section of the Elite Assist Admin interface. The sidebar and main content area are identical to the 'Profile' page, except for the 'Edit Profile' tab being active (highlighted in blue). The main content area includes a 'Change Password' tab. The profile picture and user information remain the same. A file input field for 'Profile-Pic' is present, showing 'Choose File' and 'No file chosen'. The form fields for 'Full Name', 'Email', and 'username' all contain the value 'admin'. At the bottom right is a 'Save Changes' button.

6.3 Output Design

➤ Customer Application



Splash Screen



Welcome Back,
We missed you! Login to get started

E-Mail Address

Password 

Login

Need Help? [Forgot Password](#)

Don't have account? [Register Instead](#)

Login Screen

REGISTER

**Register Now,
Create your account**

User Name

Email Email Address

Phone Contact Number
0/10

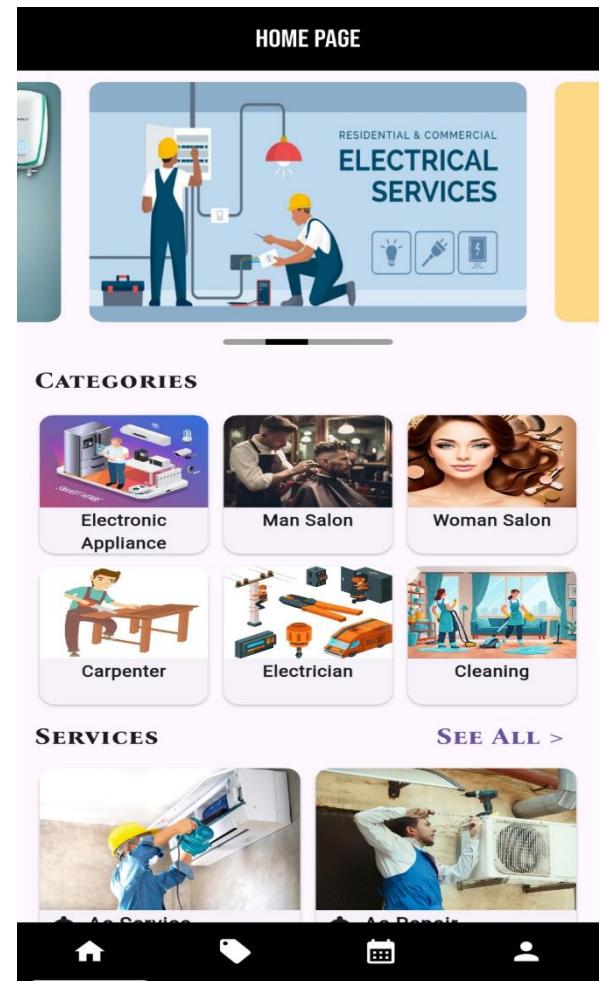
Pass Password

Pass Confirm Password

Register

Already have an account?
Login

Register Screen



Home Screen

OFFER ZONE

**50₹
OFF**

50rs discount

Coupon Code
NEW50

[Click To Copy Coupon Code](#)

Coupon Screen

BOOKINGS



AC SERVICE
25-03-2025 | 10:00 AM
PENDING



AC REPAIR
29-03-2025 | 6:00 PM
COMPLETED

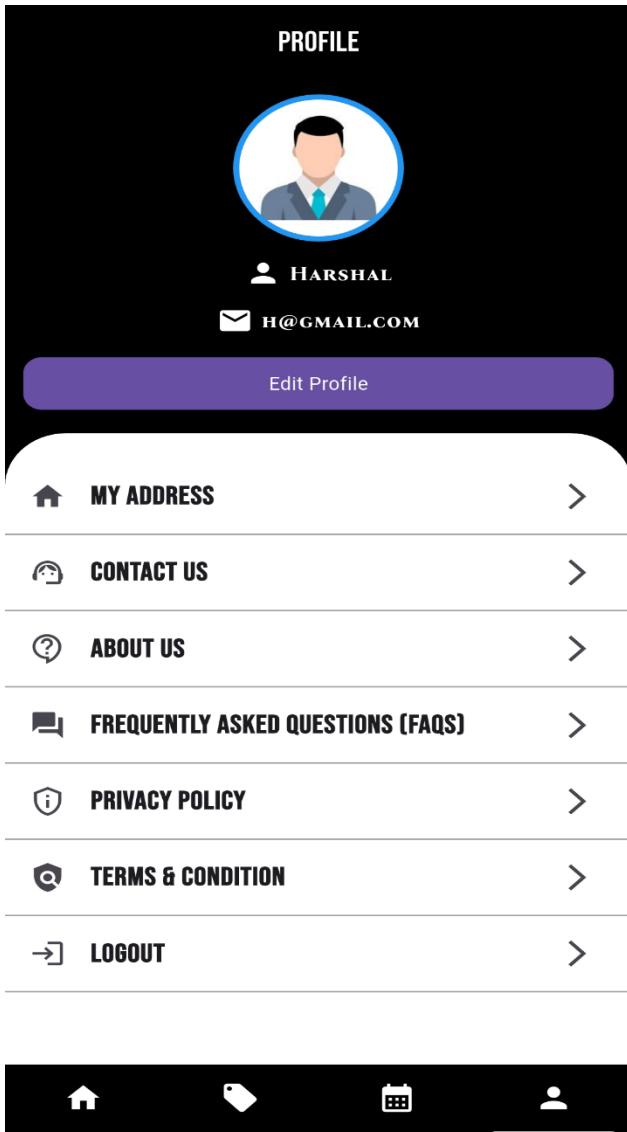


MICROWAVE REPAIR
31-03-2025 | 10:00 AM
PENDING



AC SERVICE
27-03-2025 | 10:00 AM
COMPLETED

Booking Screen



Profile Screen



Category Wise Service Screen

AIR CONDITIONER



SERVICE DETAILS

AC SERVICE

₹ 299

4.64

Less Then 1 Hour



SERVICE DESCRIPTION

Book Now

Service Detail Screen

AC SERVICE BOOKING



AC SERVICE

₹ 299

SELECT SLOTS

Select Date

Select Time

SELECT ADDRESS

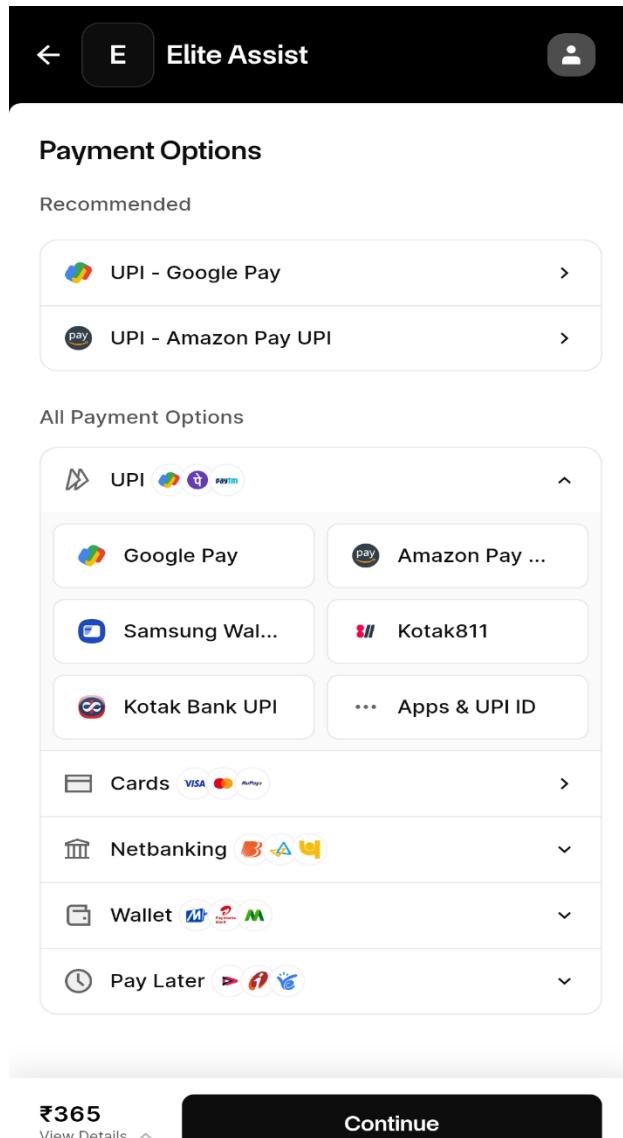
+ ADD

303 Shivam residency, Shivdas zaveri street,
Near old mahavir hospital, Sagrampura, Surat,

PAYMENT SUMMARY

SERVICE AMOUNT	299 ₹
DISCOUNT	- 0 ₹
GST (5%)	17 ₹
CONVENIENCE FEES	49 ₹
TOTAL AMOUNT	365 ₹

Service Booking Screen



Elite Assist

Payment Options

Recommended

- UPI - Google Pay
- UPI - Amazon Pay UPI

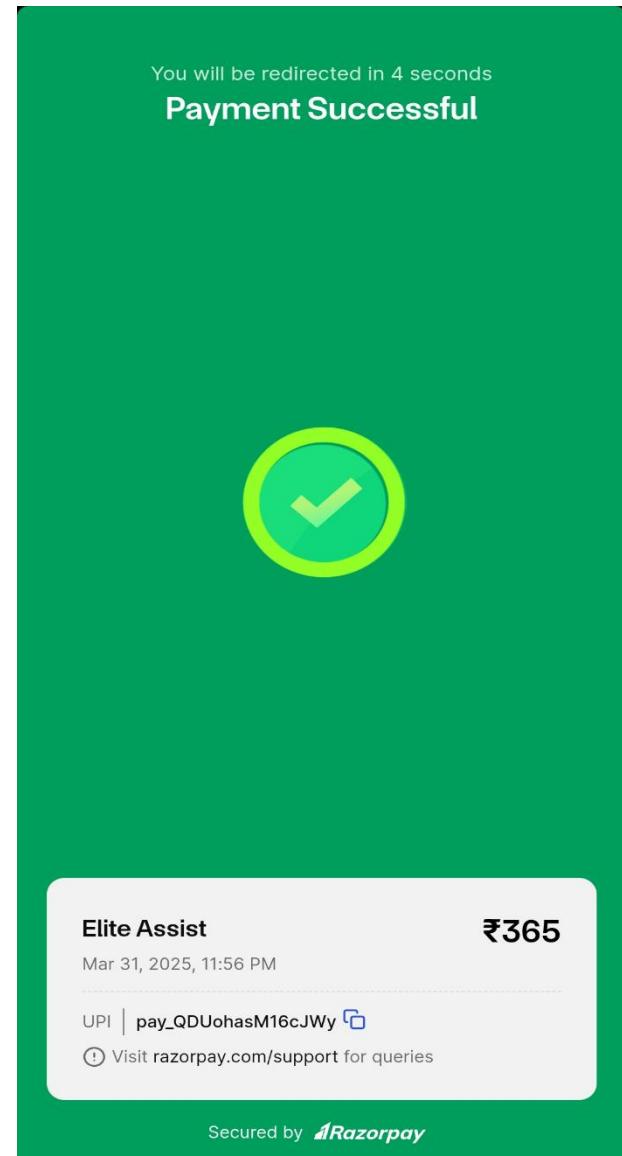
All Payment Options

- UPI (Google Pay, Amazon Pay, Samsung Wallet, Kotak811, Kotak Bank UPI, Apps & UPI ID)
- Cards (Visa, Mastercard, American Express)
- Netbanking (Bank of Baroda, Axis Bank, Kotak Mahindra Bank)
- Wallet (Paytm, PhonePe, MobiKwik)
- Pay Later (PhonePe, Paytm, SBI Credit Card)

₹365
View Details ▾

Continue

Online Payment



You will be redirected in 4 seconds

Payment Successful

A large green circle with a white checkmark inside.

Elite Assist ₹365
Mar 31, 2025, 11:56 PM

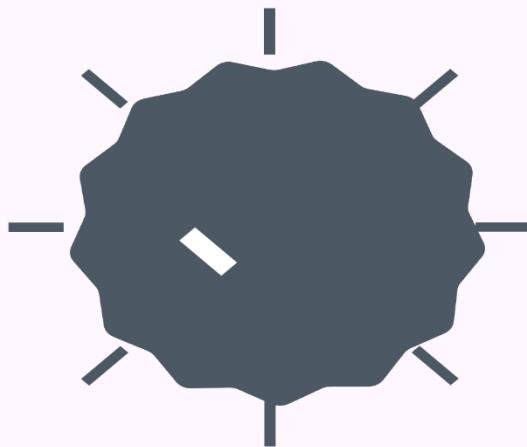
UPI | pay_QDUohasM16cJWY [Copy]
Visit razorpay.com/support for queries

Secured by 

Payment Success Screen

Payment Successful

Your Service Booked



Redirect To Home in 3 Sec

← AC REPAIR



SERVICE DETAILS

BOOKED ON: 24-03-2025 | 08:32 PM

PICKED SLOT: 29-03-2025 | 6:00 PM

COMPLETED ON: 25-03-2025 | 12:00 AM

PARTNER DETAILS



 BADA BHAI

PAYMENT SUMMARY

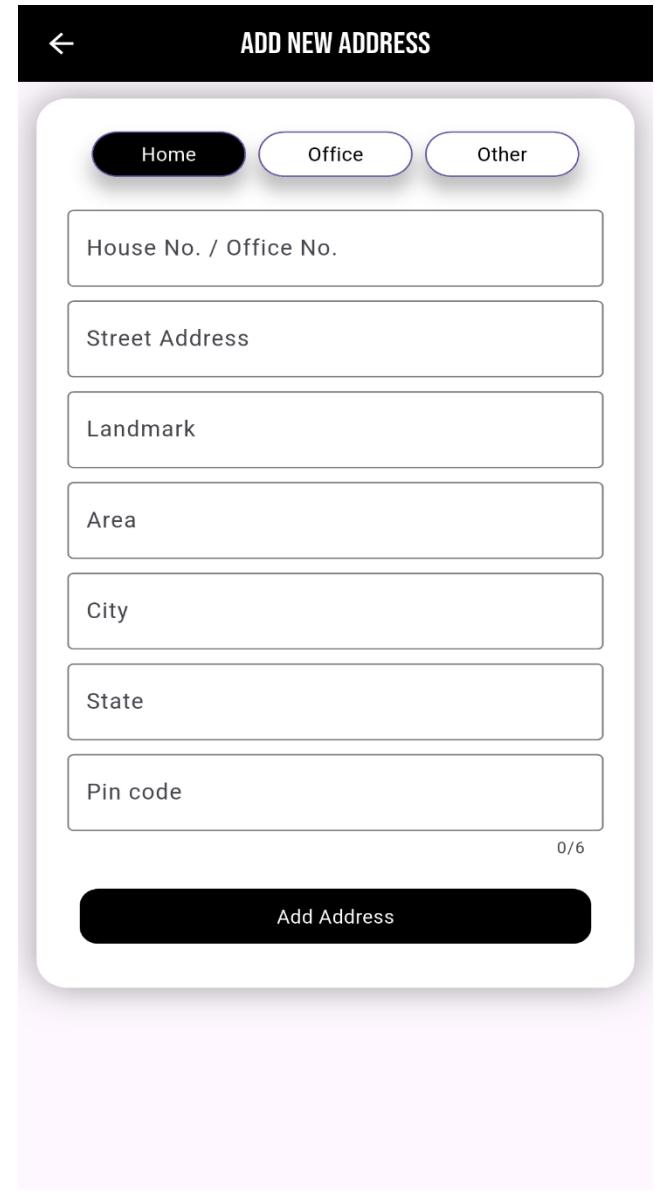
INODE. COD

Thank you Screen

Order History Screen



User Address Screen



The Add/Edit Address Screen is a form for entering new address details. At the top left is a back arrow icon, and at the top center is the title "ADD NEW ADDRESS". Below the title are three circular buttons: "Home" (selected), "Office", and "Other". The form consists of seven input fields: "House No. / Office No.", "Street Address", "Landmark", "Area", "City", "State", and "Pin code". At the bottom right of the form is a black rectangular button labeled "Add Address". Below the "Add Address" button is the text "0/6".

Add/Edit Address Screen

CONTACT US



SUPPORT CONTACT: +91 12345 67890

SUPPORT E-MAIL: SUPPORT@ELITEASSIST.COM

DEVELOP BY: HARSHAL JARIWALA

Contact Us Screen

ABOUT US

At Elite Assist, we understand the value of your time and the importance of hassle-free service experiences. Whether you're looking for home maintenance, beauty services, wellness treatments, or any other assistance, Elite Assist is here to cater to your needs.

Our Mission :

Our mission at Elite Assist is to revolutionize the way you access services by providing a seamless platform that connects you with trusted partners who deliver exceptional service experiences. We strive to make your life easier by offering a wide range of services that can be booked at your convenience, all through our user-friendly mobile app.

Why Choose Elite Assist? :

Convenience: With Elite Assist, you can book services anytime, anywhere, with just a few taps on your phone. Say goodbye to lengthy phone calls and waiting times.

Quality Assurance: We handpick our partners to ensure that they meet our high standards.

About Us Screen


FREQUENTLY ASKED QUESTIONS
1. What services does Elite Assist offer?

Elite Assist provides a wide range of professional services, including [list your services like virtual assistance, personal concierge, administrative support, tech assistance, etc.]. We tailor our services to meet your specific needs for both personal and business requirements.

2. How can I book a service with Elite Assist?

You can book a service easily through our website or by contacting our customer support team. Simply choose the service you need, and our team will get in touch to confirm the details and schedule.

3. How much do your services cost?
4. Is there a minimum commitment for your services?
5. Are your services available 24/7?
6. How do I know if Elite Assist is the right fit for my needs?
7. What happens if I need to cancel or reschedule a service?

PRIVACY POLICY

This Privacy Policy outlines how Elite Assist collects, uses, maintains, and discloses information obtained from users of the Elite Assist mobile application. By using the App, you agree to the terms outlined in this Privacy Policy.

Information We Collect :

Personal Information: When you register an account or use our services, we may collect personal information such as your name, email address, phone number, and address.

Usage Information: We may collect information about how you interact with the App, including pages visited, features used, and preferences selected.

Device Information: We may collect information about the device you use to access the App, including device type, operating system, and unique device identifiers.

Location Information: With your consent, we may collect location information to provide location-based services such as finding

F.A.Q Screen
Privacy Policy Screen

TERMS & CONDITIONS

These Terms and Conditions govern your use of the Elite Assist mobile application and the services provided therein. By accessing or using the App, you agree to be bound by these Terms. If you do not agree with any part of these Terms, you may not use the App.

Use of the App :

Registration: In order to access certain features of the App, you may be required to register an account. You agree to provide accurate and complete information during the registration process and to keep your account credentials confidential.

User Conduct: You agree to use the App in accordance with all applicable laws and regulations. You further agree not to:

- Use the App for any unlawful purpose or in any way that violates these Terms.
- Interfere with the operation of the App or attempt to gain unauthorized access to any portion of the App.
- Upload or transmit any content that is harmful, offensive, or violates the rights of others.

Terms & Condition Screen

EDIT PROFILE



E-Mail Address –

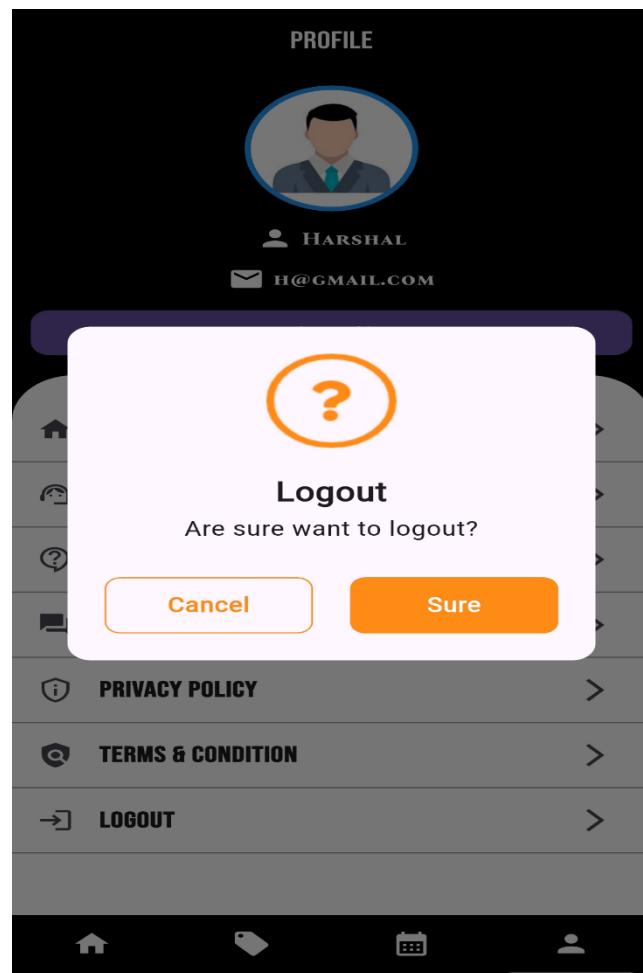
Name –

Contact Number –

10/10

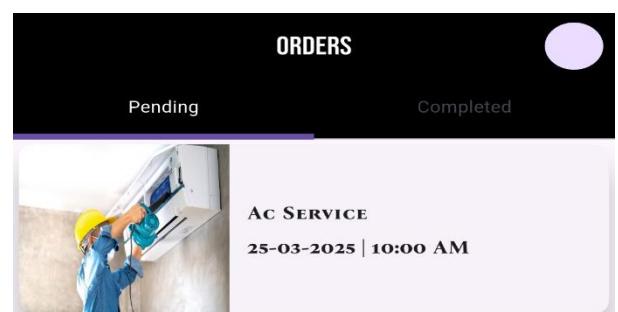
[Edit Details](#)

Edit Profile Screen



Logout Popup

➤ Partner Application



Mobile Number

Login

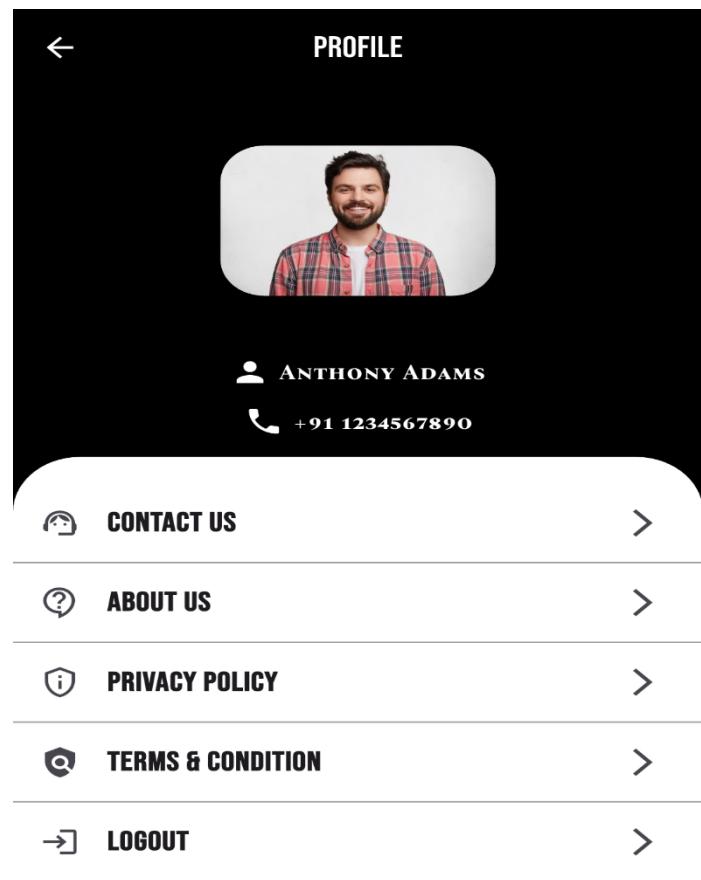


Login Screen

Pending Order Screen



Completed Order Screen



Profile Screen

CONTACT US



SUPPORT CONTACT: +91 12345 67890
SUPPORT E-MAIL: SUPPORT@ELITEASSIST.COM

DEVELOP BY: HARSHAL JARIWALA

ABOUT US

At Elite Assist, we understand the value of your time and the importance of hassle-free service experiences. Whether you're looking for home maintenance, beauty services, wellness treatments, or any other assistance, Elite Assist is here to cater to your needs.

Our Mission :

Our mission at Elite Assist is to revolutionize the way you access services by providing a seamless platform that connects you with trusted partners who deliver exceptional service experiences. We strive to make your life easier by offering a wide range of services that can be booked at your convenience, all through our user-friendly mobile app.

Why Choose Elite Assist? :

Convenience: With Elite Assist, you can book services anytime, anywhere, with just a few taps on your phone. Say goodbye to lengthy phone calls and waiting times.

Quality Assurance: We handpick our partners to ensure that they meet our high standards.

Contact Us Screen

About Us Screen



PRIVACY POLICY

This Privacy Policy outlines how Elite Assist we, collects, uses, maintains, and discloses information obtained from users of the Elite Assist mobile application. By using the App, you agree to the terms outlined in this Privacy Policy.

Information We Collect :

Personal Information: When you register an account or use our services, we may collect personal information such as your name, email address, phone number, and address.

Usage Information: We may collect information about how you interact with the App, including pages visited, features used, and preferences selected.

Device Information: We may collect information about the device you use to access the App, including device type, operating system, and unique device identifiers.

Location Information: With your consent, we may collect location information to provide location-based services such as finding

Privacy Policy Screen



TERMS & CONDITIONS

These Terms and Conditions govern your use of the Elite Assist mobile application and the services provided therein. By accessing or using the App, you agree to be bound by these Terms. If you do not agree with any part of these Terms, you may not use the App.

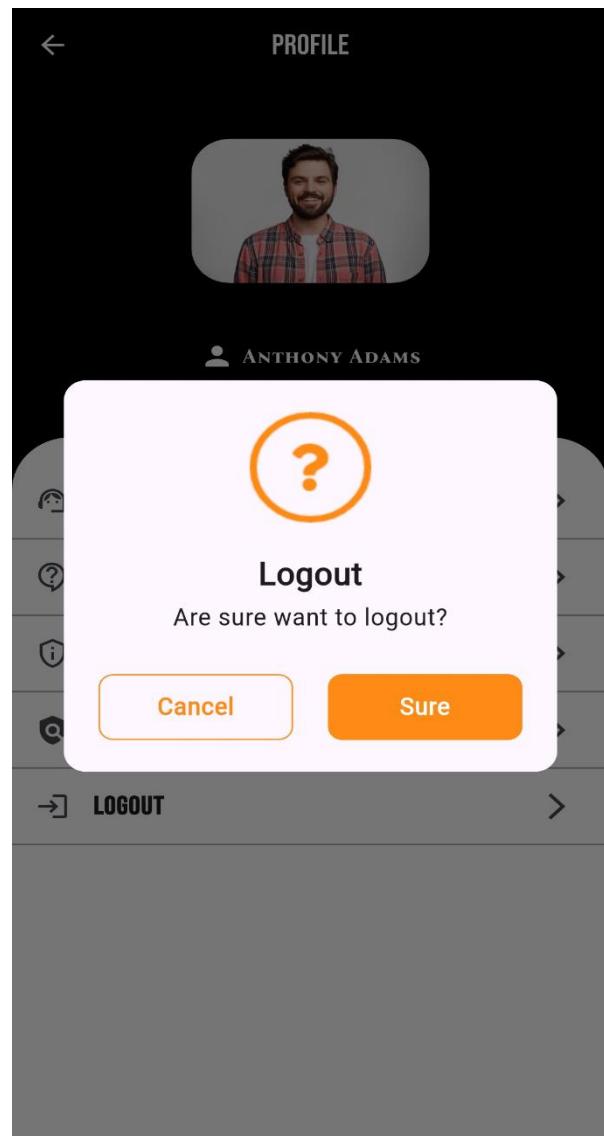
Use of the App :

Registration: In order to access certain features of the App, you may be required to register an account. You agree to provide accurate and complete information during the registration process and to keep your account credentials confidential.

User Conduct: You agree to use the App in accordance with all applicable laws and regulations. You further agree not to:

Use the App for any unlawful purpose or in any way that violates these Terms.
Interfere with the operation of the App or attempt to gain unauthorized access to any portion of the App.
Upload or transmit any content that is harmful, offensive, or violates the rights of others.

Terms & Condition Screen



Logout Popup

7. Software Testing

7.1 Unit Testing

➤ **User Table**

SR NO	NAME	VALUE	VALID / INVALID	DESCRIPTION
1.	User_Id	Auto Generate		
2.	User_Name	BLANK	INVALID	Not Allow Blank
		123	INVALID	Allow Only Text
		ABC	VALID	
3.	Email_Id	BLANK	INVALID	Not Allow Blank
		abc@gmail.com	VALID	
4.	Mobile_No	BLANK	INVALID	Not Allow Blank
		123	INVALID	Allow 10 Digit
		1234567890	VALID	
5.	Password	BLANK	INVALID	Not Allow Blank
		123ABC	VALID	
		ABC@123	VALID	

➤ **Category Table**

SR NO	NAME	VALUE	VALID / INVALID	DESCRIPTION
1.	Service_Id	Auto Generate		
2.	Service_Name	BLANK	INVALID	Not Allow Blank
		123	INVLAID	Allow Only Text
		ABC	VALID	
3.	Service_Pic	BLANK	INVALID	Not Allow Blank
		Pic.pdf	INVALID	Allow Only Image
		Pic.jpg	VALID	

➤ **Sub – Category Table**

SR NO	NAME	VALUE	VALID / INVALID	DESCRIPTION
1.	SubService_Name	Auto Generate		
2.		BLANK	INVALID	Not Allow Blank
3.		123	INVLAID	Allow Only Text
4.	SubService_Pic	ABC	VALID	
5.		BLANK	INVLAID	Not Allow Blank
6.		Pic.pdf	INVLAID	Allow Only Images
7.		Pic.jpg	VALID	

➤ **Partner Table**

SR NO	NAME	VALUE	VALID / INVALID	DESCRIPTION
1.	Partner_Id	Auto Generate		
2.	Partner_Name	BLANK	INVALID	Not Allow Blank
3.		123	INVALID	Allow Only Text
4.		ABC	VALID	
5.	Mobile_No	BLANK	INVALID	Not Allow Blank
6.		123	INVALID	Allow 10 Digit
7.		1234567890	VALID	
8.	Email_Id	BLANK	INVALID	Not Allow Blank
9.		abc@gmail.com	VALID	
10.	Aadhar_NO	BLANK	INVALID	Not Allow Blank
11.		123	INVALID	Allow 12 Digit
12.		123456789012	VALID	
13.	Product_Id	Auto Generate		

➤ Service Table

SR NO	NAME	VALUE	VALID / INVALID	DESCRIPTION
1.	Product_Name	Auto Generate		
2.		BLANK	INVALID	Not Allow Blank
		123	INVALID	Allow Only Text
		ABC	VALID	
3.	Price	BLANK	INVALID	Not Allow Blank
		ABC	INVALID	Allow Only Numeric
		123	VALID	
4.	Details	BLANK	INVALID	Not Allow Blank
		ABC123	VALID	
5.	Type	BLANK	INVALID	Not Allow Blank
		123	INVLAID	Allow Only Text
		ABC	VALID	
6.	Time	BLANK	INVALID	Not Allow Blank
		ABC123	VALID	
7.	Product_Pic1	BLANK	INVALID	Not Allow Blank
		Pic.pdf	INVALID	Allow Only Image
		Pic.jpg	VALID	
8.	Product_Pic2	BLANK	INVALID	Not Allow Blank
		Pic.pdf	INVALID	Allow Only Image
		Pic.jpg	VALID	
9.	Product_Vid	BLANK	INVALID	Not Allow Blank
		Video.pdf	INVALID	Allow Only Video
		Video.mp4	VALID	
10.	Gender	BLANK	INVALID	Not Allow Blank
		'Male' Or 'Female'	VALID	
11.	SubService_Id	Auto Generate		

➤ **Booking Table**

SR NO	NAME	VALUE	VALID / INVALID	DESCRIPTION
1.	Booking_Id	Auto Generate		
2.	User_Id	Auto Generate		
3.	Product_Id	Auto Generate		
4.	Price	Auto Generate		
5.	Status	BLANK	INVALID	Not Allow Blank
		'0' Or '1'	VALID	
6.	Date	BLANK	INVALID	Not Allow Blank
		DD/MM/YYYY	VALID	
7.	Time	BLANK	INVALID	Not Allow Blank
		HH AM/PM	VALID	

➤ **Coupon Table**

SR NO	NAME	VALUE	VALID / INVALID	DESCRIPTION
1.	Coupon_Id	Auto Generate		
2.	Coupon_Code	BLANK	INVALID	Not Allow Blank
		ABC123	VALID	
3.	Coupon_Description	BLANK	INVALID	Not Allow Blank
		ABC123	VALID	
4.	Coupon_Discount	BLANK	INVALID	Not Allow Blank
		ABC	INVALID	Not Allow Text
		123 %	VALID	
5.	Status	BLANK	INVALID	Not Allow Blank
		'0' Or '1'	VALID	

➤ **Banner Table**

SR NO	NAME	VALUE	VALID / INVALID	DESCRIPTION
1.	Banner_Id	Auto Generate		
2.	Banner_Pic	BLANK	INVALID	Not Allow Blank
		Pic.pdf	INVALID	Allow Only Image
		Pic.jpg	VALID	
3.	Status	BLANK	INVALID	Not Allow Blank
		'0' Or '1'	VALID	

7.2 Integration Testing

Integration testing is essential for ensuring that different parts of your system work together as expected. Since you have a Laravel backend, Flutter frontends for Customer and Partner sites, and MongoDB as your database, integration testing will involve verifying the communication between your services, data consistency, and overall functionality across the different platforms.

Integration Testing Strategy

I. Test Environment Setup:

- **Backend (Laravel):** Ensure that your Laravel application is running in an environment where all dependencies are properly configured, especially for database connections to MongoDB.
- **Frontend (Flutter):** Make sure that both the Customer and Partner Flutter apps are connected to the correct backend API.
- **Database (MongoDB):** Use a test database or mock database for testing purposes to ensure that real data is not modified during tests.

II. Test Flow Definition Identify key user flows across the Admin, Customer, and Partner sites. Below are the primary use cases to cover in the tests:

- Admin manages categories, sub-categories, services, and orders.
- Customer can view and book services, apply coupons, and view order history.
- Partner can view assigned orders, see customer details, and mark services as complete.

Step-by-Step Integration Testing Breakdown

I. Admin Site (Laravel) Tests

➤ Manage Categories & Sub-categories:

- **Test Case:** Admin adds a new category and sub-category.
- **Expected Result:** Category and sub-category should be successfully added to MongoDB. Verify if they appear in the Customer and Partner apps.
- **Test Steps:**
 - Send a request to the POST /categories endpoint with valid data.
 - Verify the response is success (status 200).

- Check if the new category appears in MongoDB.
- Ensure that the new category is visible on the Customer and Partner apps.

➤ **Manage Services:**

- **Test Case:** Admin adds a new service under a category.
- **Expected Result:** The service should be added and linked to the appropriate category and sub-category.
- **Test Steps:**
 - Admin sends a request to add a service.
 - Verify if the service is correctly linked with the category and sub-category.
 - Check if the service appears on the Customer site.

➤ **Assign Orders to Partners:**

- **Test Case:** Admin assigns an order to a partner.
- **Expected Result:** The assigned partner should be able to see the order in their system.
- **Test Steps:**
 - Admin assigns an order to a partner.
 - Verify that the partner's system receives the correct order.
 - Confirm that the order status changes appropriately in the system.

II. Customer Site (Flutter) Tests

➤ **View Services by Category/Sub-category:**

- **Test Case:** Customer selects a category and sees the relevant services.
- **Expected Result:** Services are filtered and displayed correctly based on the selected category/sub-category.
- **Test Steps:**
 - Customer selects a category.

- Flutter app requests services for that category.
- Verify that the correct services are returned and displayed in the app.

➤ **Book a Service with Slots:**

- **Test Case:** Customer selects a service and books it for a specific time slot.
- **Expected Result:** The booking is registered correctly, and the order is created in MongoDB.
- **Test Steps:**
 - Customer selects a service and time slot.
 - The app sends a booking request to the backend.
 - Verify that the order is created in MongoDB with the correct details.
 - Ensure that the order is assigned to a partner.
 - Verify that the booking appears in the customer's order history.

➤ **Apply Coupon:**

- **Test Case:** Customer applies a valid coupon code during service booking.
- **Expected Result:** The discount is applied correctly, and the final price is updated.
- **Test Steps:**
 - Customer applies a coupon code.
 - Verify that the correct discount is applied to the service cost.
 - Check if the discount is reflected in the final order price.

III. Partner Site (Flutter) Tests

➤ **View Assigned Orders:**

- **Test Case:** Partner views their assigned orders.
- **Expected Result:** The assigned orders should be visible on the Partner site.
- **Test Steps:**
 - Partner logs in.
 - The partner should see a list of orders assigned to them.

- Verify that the order details (e.g., customer info, service info) are correct.

➤ **Finish Service:**

- **Test Case:** Partner marks an order as completed.
- **Expected Result:** The order status should change to "Completed" and this should be reflected in the Customer's order history.
- **Test Steps:**
 - Partner marks the service as completed.
 - Verify that the order status changes to "Completed" in MongoDB.
 - Verify that the Customer app shows the updated order status.

7.3 System Testing

System testing is a critical phase where the entire system is evaluated as a complete unit to ensure it functions as intended. In the context of Elite Assist, your system involves an Admin site built with Laravel, two Flutter apps for the Customer and Partner sites, and MongoDB as the database. The purpose of system testing is to validate that all components interact correctly and that the system meets the functional and non-functional requirements specified in the design phase.

➤ Functional Testing

Functional testing ensures that all system features work according to the specifications and requirements. For the Admin Site, the primary goal is to test the ability of the admin to manage categories, sub-categories, services, orders, partners, and coupons. The admin should be able to add, edit, or delete categories and sub-categories. These changes should be reflected accurately in MongoDB and should also be visible in both the Customer and Partner apps. If the admin modifies a service or creates a new one, the service should be properly linked to the appropriate category and sub-category, and the Customer site should display it correctly.

Admin order management is another critical area. The admin should be able to assign or change the status of orders. For example, when an order is assigned to a partner, the partner should be notified, and the order should reflect this status change both in MongoDB and in the Partner app. When the customer views their order history, the data displayed should align with what is in the database. Admin actions should trigger corresponding changes across all system components, including the Customer and Partner sites.

On the Customer Site, the focus of functional testing is to verify that the customer can browse services based on categories and sub-categories. The system should filter and display only the relevant services based on the customer's selection. If the customer books a service, the system should correctly create an order in MongoDB with the right service details, time slot, and customer information. Moreover, the booking system should allow for the application of valid coupons, adjusting the order total accordingly. After booking a service, the customer should be able to see the order confirmation and the status of their booking.

In addition, the customer must be able to view their order history. Each order should reflect the correct status, such as "Pending," "Completed," or "In Progress," based on the progress of the associated service. These statuses should also be updated in MongoDB and visible across all sites, so the customer can track the progress of their orders in real time.

For the Partner Site, testing should focus on whether the partner can view the orders assigned to them. When the admin assigns an order, it should appear on the Partner's site with all the relevant details, such as the customer's name, service booked, and time slot. The partner should also be able to see and manage their assigned orders, including the ability to mark an order as completed once the

service has been rendered. After marking an order as completed, the order's status should automatically update in MongoDB, and this change should reflect in the Customer app as well, providing real-time updates to the customer.

➤ **Non-functional Testing**

Non-functional testing focuses on testing the system's behavior and performance under various conditions, beyond the basic functional requirements. One of the first non-functional tests to perform is performance testing. It evaluates how the system performs under load. For example, during peak usage, such as when multiple customers are browsing services or booking orders simultaneously, the backend should not experience slowdowns, and the Flutter apps should remain responsive. The system should be able to handle a significant number of concurrent users without crashing or slowing down. You can simulate such load using performance testing tools like Apache JMeter or Loader.io, testing for response times, throughput, and scalability. The system should also maintain acceptable response times even during high traffic.

Security testing is another important aspect. In this phase, the system is tested to ensure that all sensitive data, such as customer details, passwords, payment information, and service booking data, is securely stored and transmitted. The backend should use secure protocols (HTTPS) to protect data in transit. The system should also enforce proper authentication and authorization mechanisms, ensuring that only authorized users can access specific resources. For example, only admins should have access to manage categories or services, and only partners should be able to view their assigned orders. Additionally, testing for security vulnerabilities such as SQL injection, Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF) is crucial. These vulnerabilities can expose sensitive information or allow attackers to manipulate system data. Proper validation of user inputs and robust session management are essential for maintaining the security of the application.

➤ **Data Integrity Testing**

Data integrity testing ensures that the system's data remains accurate and consistent across the database, backend, and frontend components. Since the Elite Assist system relies heavily on MongoDB as its database, it is essential to ensure that data stored in MongoDB is correctly handled and consistent across the Admin, Customer, and Partner sites. If the admin creates or modifies a category, sub-category, or service, this data should be saved correctly in MongoDB, and the changes should be immediately reflected on the Customer and Partner apps. Similarly, if a customer books a service or applies a coupon, the booking should be recorded correctly in MongoDB with the accurate details, such as the service chosen, time slot, customer name, and coupon details.

Data integrity testing will also include verifying that any changes made on one site (e.g., a customer booking a service) are correctly propagated to the other sites (e.g., the Partner sees the booking in their assigned orders). If there is a change in order status (e.g., from "Pending" to "Completed"), the system should update all sites accordingly, so the customer, admin, and partner are always in sync. The database should maintain consistency throughout these processes, and no data should be lost or corrupted during operations.

➤ **User Interface (UI) Testing**

UI testing ensures that the frontend is intuitive, user-friendly, and visually functional across various devices. Since your frontend is built with Flutter for both the Customer and Partner apps, you will need to check if the UI components like buttons, text fields, images, lists, and navigation menus are displayed correctly. The user interface should be designed for easy navigation, with clear labeling of elements, appropriate color schemes, and responsive layouts that adjust based on screen size and resolution.

For example, when a customer browses through services by category, the app should present the available services in a clean and structured manner. The booking process should be simple and easy to understand, with clearly defined steps for service selection, time slot selection, and coupon application. Any errors or validation messages should be clearly visible, and users should not face difficulties while interacting with the app.

Similarly, the Partner app should display assigned orders in an organized manner, with easy-to-read customer details, service details, and status updates. The interface should also be responsive and display correctly on different devices. UI testing ensures that the system is not only functional but also provides a pleasant experience for the users.

➤ **End-to-End Workflow Testing**

End-to-end testing is a comprehensive type of testing where the entire system is tested as a whole. The goal is to simulate real-world usage and ensure that all components, including the Admin site, Customer site, Partner site, and MongoDB, work seamlessly together throughout the entire workflow. For example, testing the entire process from service creation by the admin to service booking by the customer and completion by the partner is critical.

In this flow, the admin first creates and assigns a service to a partner. The customer then browses the available services and selects one to book, applying a coupon for a discount. The order is saved in MongoDB, and the partner sees the order assigned to them. Once the service is completed, the partner marks the order as "Completed," and the system updates the status. The customer is notified of the completed service, and the order history is updated on both the Customer and Partner apps.

During end-to-end testing, you will verify that all system interactions are functioning as expected, from the Admin creating services to the customer booking them, and the partner completing them. The system should maintain data consistency across all components, and no part of the system should be out of sync. This testing ensures that the flow of data between components is smooth and error-free, and that users (whether admins, customers, or partners) are able to complete their tasks without issues.

8. Future Scope of Enhancements

The future scope of enhancements for Elite Assist holds significant potential for growth and innovation. As technology continues to evolve and user needs become more sophisticated, the platform must adapt to ensure it remains competitive and continues to deliver an exceptional experience to all users—Admins, Customers, and Partners alike.

One of the most crucial future directions is the improvement of real-time communication and collaboration within the platform. Currently, communication between the Admin, Customers, and Partners may be limited to notifications or email. However, enhancing this with in-app messaging and live chat functionality can improve the speed and quality of interactions. Admins could directly communicate with customers or partners to resolve issues quickly, and partners could coordinate better with customers for last-minute adjustments or clarifications about services. For instance, real-time chat with a customer could allow a partner to get immediate feedback on the service being delivered, improving the quality and customer satisfaction.

The mobile app experience can also be significantly enhanced. While the existing Flutter-based apps for Customers and Partners provide essential functionality, as the user base grows, there may be opportunities for further optimization. For example, offline functionality could be incorporated into the apps, allowing Partners to view and update their assigned orders even when they are not connected to the internet. This would be particularly useful in regions where network connectivity is unstable. Additionally, adding push notifications for critical updates, such as service status changes or new service assignments, can keep users engaged and informed in real-time.

Expanding on customer engagement, there's an opportunity to introduce loyalty programs to incentivize repeat bookings. A point system where customers earn points for every service they book, which can later be redeemed for discounts or free services, would be an excellent way to foster customer retention. Moreover, creating seasonal campaigns and flash sales can be an engaging way to keep customers coming back and offer partners the opportunity to showcase their services during high-demand periods.

To enhance service delivery, introducing augmented reality (AR) or virtual reality (VR) could revolutionize the way services are showcased or booked. For example, if the platform offers home cleaning or interior design services, AR could allow customers to visualize the outcome of the service in their own space before booking. This could help customers make more informed decisions and increase conversion rates. Additionally, VR could be used for training partners to handle complex tasks or provide better customer service.

From a back-end perspective, integrating artificial intelligence (AI) into the system can greatly improve the efficiency and personalization of services. AI-powered algorithms could analyze customer behavior and booking patterns to predict which services are likely to be in demand in a particular region at a given time. This predictive modeling could help the platform suggest the right services to customers even before they search

for them, making the process much smoother. AI can also be used for smart scheduling to optimize service delivery times based on partner availability, location, and customer preferences.

On the operational side, incorporating blockchain technology could be a game-changer, particularly for transactional transparency and contract management between partners and customers. Using blockchain for payment verification would ensure that all transactions are secure, immutable, and transparent, which could help foster trust, especially when dealing with multiple partners and customers in different regions. Blockchain can also be used for smart contracts to streamline the agreement processes between partners and customers, ensuring all terms are clearly defined and automatically enforced.

The platform's data privacy and compliance are also key areas for improvement as the system expands. With an increasing number of users and data being generated, ensuring that the system complies with global data protection regulations like GDPR, CCPA, and others will be crucial. Enhancements in data anonymization and tokenization can help ensure sensitive user information is protected while also enabling the platform to scale globally without running into legal issues. Additionally, data retention policies can be automated to comply with legal requirements, ensuring that data is only kept for the necessary duration.

There's also potential for cross-platform integration with external services that could provide added value to customers and partners. For instance, integrating with calendar applications (like Google Calendar or Outlook) would allow customers and partners to automatically sync their appointments and availability. For customers, integration with home service providers (such as smart home systems) could enable them to automate bookings for home-related services. These integrations could enhance the platform's utility and help it expand into new verticals.

The use of IoT (Internet of Things) in the platform could also be explored. For example, for service providers in industries like home repairs, installation, or maintenance, IoT devices could automatically detect issues in a home (such as a leak, temperature anomaly, or faulty appliance) and trigger an automatic service request. This could streamline the booking process and improve the efficiency of services provided. Similarly, for fitness or wellness services, wearable devices could integrate with the platform, tracking customer health and automatically recommending services or follow-up appointments based on data such as steps taken, heart rate, or sleep quality.

Customer support is another area that can benefit from advanced technology. While chatbots are a good start, introducing AI-driven virtual assistants could take customer service to the next level. These assistants could understand complex customer queries, offer solutions, and even schedule appointments based on the customer's available times. By integrating machine learning into the virtual assistant, the system could continuously learn from past interactions, offering increasingly sophisticated and accurate solutions over time. Moreover, multilingual support could make the platform more accessible to non-English-speaking users, especially as the service expands to new geographical locations.

The partner ecosystem can also be enhanced by improving the tools available to service providers. Introducing training modules and certifications for partners could elevate the

quality of services provided. These modules could cover everything from soft skills (like customer service) to technical skills (such as using specific tools or following best practices for specific services). Partners could earn badges or certifications, which could be displayed on their profiles, building trust and credibility with customers. This not only enhances the partner experience but also ensures that customers receive high-quality services consistently.

As the platform grows and evolves, it could also explore new business models and service offerings. For example, creating a marketplace where partners can offer their services directly to customers without going through the admin for assignment would be a significant expansion. This peer-to-peer model could open up new revenue streams and allow customers to book services based on ratings and reviews, much like a gig economy platform. Another opportunity is white-labeling the platform for other businesses. For example, local businesses or franchises could adopt Elite Assist as their own platform to provide services under a branded experience.

Finally, as Elite Assist scales, the platform will need to evolve its analytics and reporting features. More advanced analytics could provide Admins with deeper insights into customer trends, service performance, revenue generation, and partner activity. With data-driven insights, Admins can make better decisions about where to expand services, which services to prioritize, and how to manage customer and partner relationships effectively.

9. Bibliography & Reference

- <https://www.urbancompany.com/surat>
- <https://www.youtube.com/>
- <https://docs.flutter.dev/get-started/install>
- <https://laravel.com/>
- <https://pub.dev/>
- <https://fluttergems.dev/>