TELESOM ACADEMY
Est 2020

Course:Full-Stack Web Development

Group Project

Group Name: Innovators

Project Members

No	Name	Role
1.	Isxaaq Cabdiqani	Web Backend
2.	Guuleed Maxamed	Web Frontend
3.	Naciima Haruun	UI/UX & Documentation
4.	Cabdiraxmaan Ibrahim	Web Backend
5.	Hibo Sulieman Amen	UI/UX & Documentation
6.	Cabdiraxman Mustafe	Web Frontend

Lecturer:Eng. Zakaria Mohamed

ServiLink: A Local Service Marketplace Platform

Abstract— In today's digital era, many essential daily services such as plumbing, electrical work, cleaning, and technical maintenance are still accessed through informal and inefficient methods. There is currently a significant lack of dedicated software platforms that effectively connect skilled service providers with customers in a structured, reliable, and transparent way. As a result, customers struggle to find trustworthy professionals, while skilled workers face difficulties in reaching potential clients and growing their businesses.ServiLink is a web-based Local Service Provider System designed to bridge this gap by creating a centralized digital marketplace that connects customers with verified and skilled service providers. The platform enables users to easily discover services, compare providers, make bookings, track service progress, and provide feedback through ratings and reviews. At the same time, service providers can manage their profiles, showcase their skills, receive job requests, and build professional reputations.By digitalizing the entire service interaction process, ServiLink improves accessibility, efficiency, and trust for both customers and providers. The system not only simplifies service discovery and management but also contributes to economic empowerment by giving skilled workers greater visibility and opportunities in the digital economy.

1. INTRODUCTION

In many parts of the world, accessing reliable local services such as plumbing, electrical repairs, cleaning, and technical maintenance remains a major challenge. Most people still depend on informal methods like asking friends, walking around neighborhoods, or using social media posts to find skilled workers. These methods are time-consuming, unreliable, and often lead to poor service quality due to lack of proper verification and accountability.

ServiLink is a web-based Local Service Provider System developed to address this problem by creating a digital platform that connects customers with skilled service providers in an organized and efficient manner. The system acts as a bridge between people who need services and professionals who are capable of delivering them, all within a single, user-friendly environment.

The platform is needed in real life because there is currently no widely adopted software solution that focuses specifically on local skilled services in a structured way. Customers struggle with trust issues, unclear pricing, inconsistent availability, and lack of communication. On the other hand, many skilled workers such as plumbers, cleaners, and technicians rely heavily on word-of-mouth and physical presence to find jobs, which limits their reach and income potential.

ServiLink solves these problems by digitalizing the entire service interaction process. Customers can search for services, view verified provider profiles, compare ratings, and make bookings with confidence. Service providers can create professional profiles, list their services, receive booking requests, and build reputations through reviews. The system also improves communication by enabling direct interaction between users and providers, reducing misunderstandings and improving service efficiency.

Overall, ServiLink aims to transform the traditional and fragmented local service market into a transparent, reliable, and scalable digital ecosystem that benefits both customers and service providers.

2. SYSTEM OVERVIEW

ServiLink is designed as a comprehensive web-based platform that bridges the gap between customers seeking local services and skilled service providers offering them. The system supports role-based access, accommodating three main types of users: customers, service providers, and administrators, each with distinct capabilities. Customers can explore a wide variety of service categories, review provider profiles, make bookings, track the status of their requests, and provide ratings and feedback. Service providers, on the other hand, can register, create detailed profiles, manage service listings, handle booking requests, and mark jobs as completed. Administrators oversee the platform's operations, managing user accounts, verifying and approving providers, maintaining service categories, and monitoring overall system activity. This structure ensures that every participant experiences a workflow tailored to their role while maintaining the integrity and reliability of the platform.

The system employs a secure authentication mechanism to protect user data and control access. When users register or log in, the frontend communicates with the backend, which validates credentials and generates a secure session token. This token enables users to access features specific to their roles. Providers undergo an additional verification process during onboarding: after creating a profile with services, pricing, and location, their account remains pending until reviewed and approved by an administrator. This verification ensures that only trusted, qualified providers are visible to customers, fostering confidence and maintaining high service quality standards.

Service discovery in ServiLink is intuitive and efficient. Customers can browse categorized service listings, select a category of interest, and view providers offering relevant services. Each provider profile presents detailed information, including services offered, location, pricing, and previous customer reviews, allowing customers to make informed choices. The booking process is streamlined: once a customer selects a provider and submits the booking request, the provider is notified and can accept or reject the request. Upon completion, providers update the booking status, enabling customers to leave reviews and ratings. This flow ensures clarity, accountability, and transparency throughout the service engagement.

The platform's backend is supported by a well-structured database that maintains all critical information. The User collection stores authentication and role data for all participants. Providers' services, pricing, locations, descriptions, ratings, and approval status are stored in the ProviderProfile collection. Service categories are managed in a dedicated collection to enable easy browsing and filtering. Bookings and reviews are recorded with relational references to customers and providers, ensuring data consistency and enabling the system to generate meaningful insights, such as provider performance or booking trends.

ServiLink is built using a modern web architecture. The frontend, implemented with React, delivers a responsive and interactive user interface, while the backend, powered by Express.js, handles business logic, data processing, and API endpoints. MongoDB is used for data storage, providing flexibility and scalability for a growing number of users, providers, and service records. APIs are organized according to user roles: customer APIs support browsing, booking, and reviewing services; provider APIs manage profile creation, bookings, and status updates; and admin APIs allow oversight, approvals, and system monitoring. This modular structure ensures maintainability, easy integration of new features, and efficient communication between components.

The development of ServiLink follows a phased roadmap to ensure a structured approach. Initial planning defines system requirements, user roles, features, and database schema. Backend development focuses on establishing the server environment, authentication, and core APIs. The frontend phase creates responsive interfaces and dashboards for customers, providers, and administrators, linking them seamlessly with backend services. This phased methodology ensures systematic development, quality control, and a stable final platform.

In essence, ServiLink transforms the fragmented and informal local services market into a transparent, reliable, and organized digital ecosystem. By integrating verified providers, streamlined booking processes, real-time updates, and feedback mechanisms, the platform enhances trust, efficiency, and accessibility for both customers and service providers. Through its thoughtful design, robust architecture, and user-focused features, ServiLink delivers a modern solution that modernizes how local services are accessed and managed.

3. SYSTEM MODULES

3.1. User Modules

The **User Modules** in ServiLink are designed to provide customers with a seamless and intuitive experience when accessing local services. These modules encompass **Authentication, Service Browsing, Booking, Reviews, and Profile Management**, **Login/Register** each playing a crucial role in enabling smooth interaction between customers and service providers.

3.1.1 Authentication Module

The Authentication module ensures secure and controlled access to the platform. Customers are required to register using their personal details and login credentials, which are verified against the backend database. This module generates a secure session token upon successful login, enabling users to access features based on their role. By safeguarding personal and transactional information, the Authentication module provides the foundation for trust and reliability within the platform.

3.1.2 Service Browsing Module

The Service Browsing module allows customers to explore available services in an organized and efficient manner. Users can view categorized listings of services such as plumbing, cleaning, electrical repairs, and technical maintenance. Each category leads to a list of verified providers with detailed profiles, including service offerings, pricing, location, and ratings. This module empowers customers to make informed decisions by providing comprehensive and transparent information about each provider.

3.1.3 Booking Module

The Booking module is the core of customer-provider interaction. Once a customer selects a provider, they can submit a booking request specifying details such as the date, time, and any special requirements. The system records the booking and notifies the provider, who can accept or reject the request. The module tracks the status of each booking pending, accepted, rejected, or completed—ensuring clarity and accountability. By streamlining the booking process, this module reduces delays, miscommunication, and uncertainty, creating a smooth service experience for both parties.

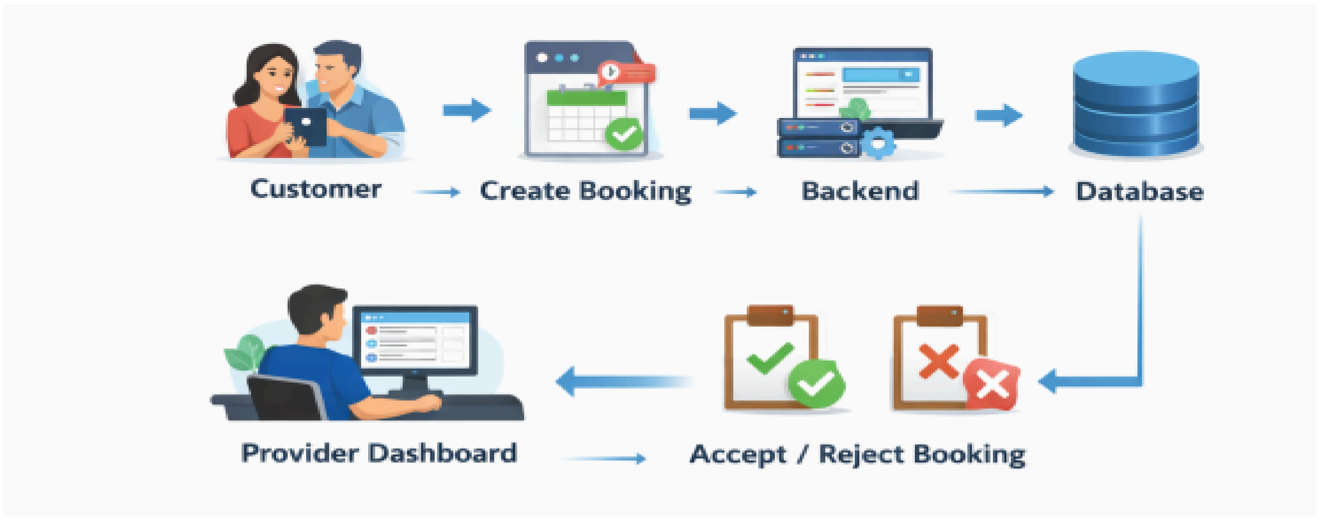


Figure : Booking flow chart

3.1.4 Provider Onboarding & Verification Module

The Provider Profile & Verification Module manages the onboarding and authentication of service providers within the ServiLink ecosystem. The process begins with the provider submitting a detailed profile for registration. This submission is routed to the Admin Dashboard for verification and approval. Once approved by the administrator, the provider's account is activated, granting them access to the Provider Dashboard. Here, they can view, manage, and respond to customer bookings and service orders associated with their profile.

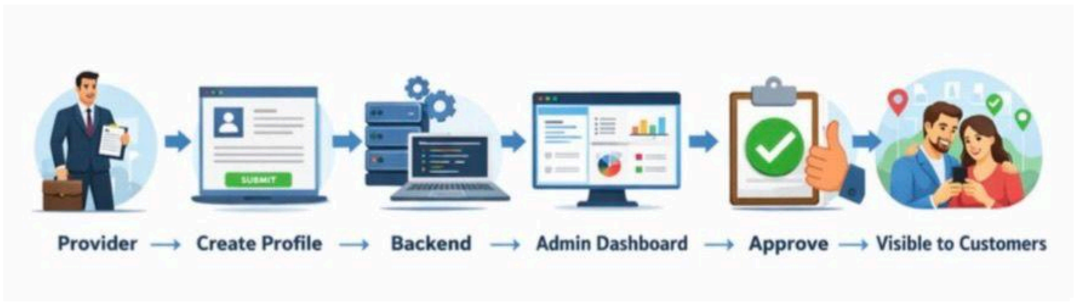


Figure : Provider flow chart

3.1.5 Reviews Module

The Reviews module enables customers to provide feedback on the services they receive. After a booking is completed, customers can submit ratings and comments that are stored in the system and reflected on the provider's profile. This module serves as a quality assurance mechanism, encouraging providers to maintain high standards and helping future customers make informed choices. By fostering transparency and trust, the Reviews module enhances the overall reliability and credibility of the platform.

3.1.5 Profile Module

The Profile module allows customers to manage their personal information and preferences. Users can update contact details, view past bookings, track ongoing services, and manage their account settings. This module ensures that the platform maintains personalized interactions and a smooth, user-friendly experience. It also provides a historical record of user activity, which can be leveraged for insights and improving service delivery.

These User Modules collectively form the backbone of the ***customer experience in ServiLink****, ensuring that interactions with the platform are secure, informative, efficient, and transparent. By integrating authentication, browsing, booking, reviews, and profile management, the system provides a cohesive and intuitive environment for customers to access local services with confidence.*

4. Use Case Diagram(Actors & Actions)

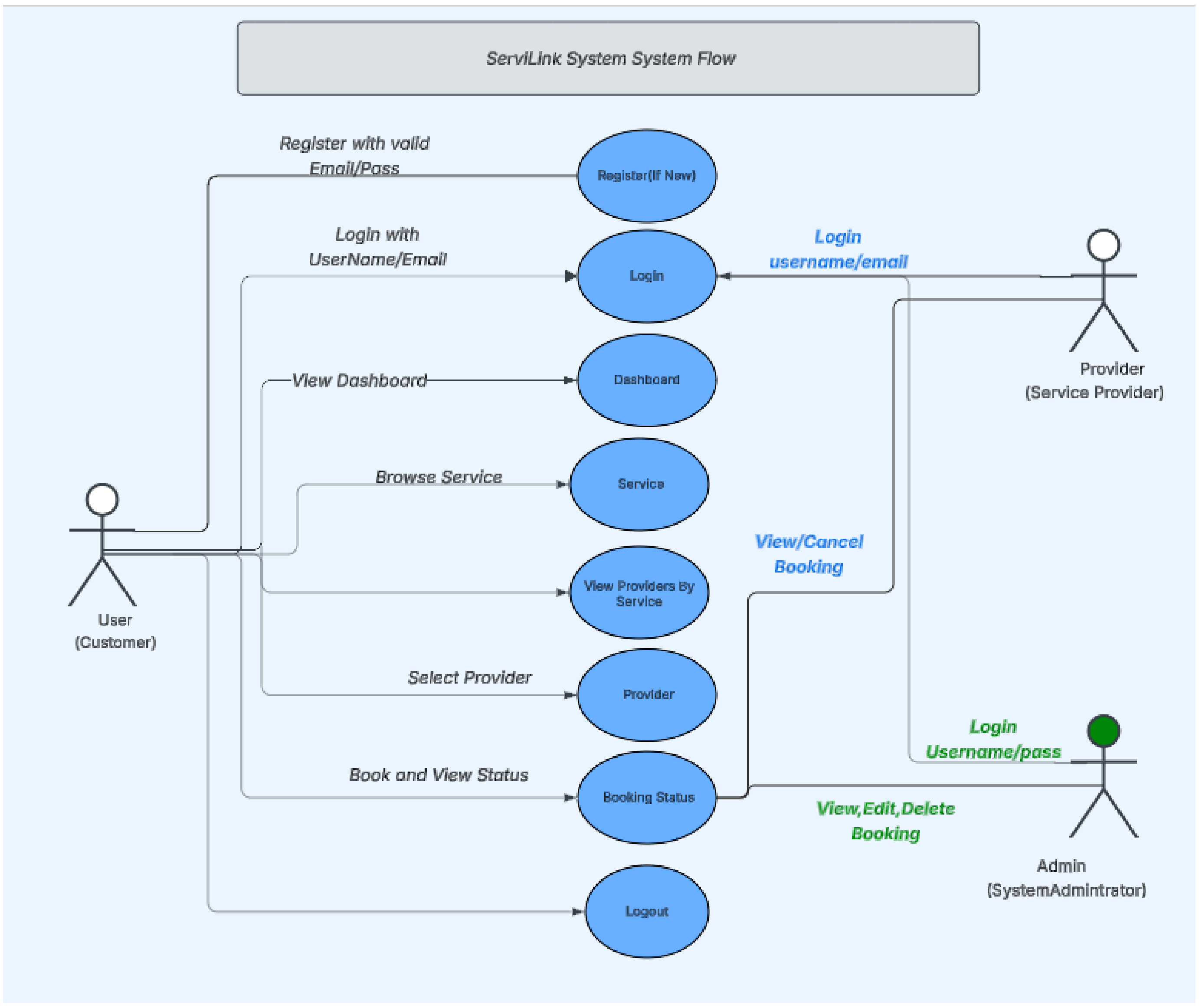


Figure : User Use Cases Diagram

Use Case Diagram Description

Actors and System Interactions in ServiLink

The ServiLink system is designed around three primary actors: User (Customer), Provider (Service Provider), and Admin (System Administrator). Each actor interacts with the system based on their specific role and access permissions. These interactions define the functional behavior of the system and ensure clear separation of responsibilities.

User (Customer)

The User represents individuals who seek services through the ServiLink platform. Users begin their interaction with the system by registering if they are new, after which they can log in using their credentials. Upon successful authentication, users are redirected to the User Dashboard, which serves as their main control panel within the system.

From the dashboard, users can browse available services and view a list of service providers based on the selected service category. Users can then book a provider of their choice and monitor the status of their booking through the system. The user session ends when the user logs out of the platform.

This flow ensures that users can efficiently discover services, make informed selections, and manage their service requests in a structured and user-friendly environment.

Provider (Service Provider)

The Provider represents individuals or organizations that offer services on the ServiLink platform. Providers do not have the ability to self-register; instead, they must be registered by the system administrator. Once registered, providers can log in to access their Provider Dashboard.

Through the dashboard, providers can view incoming booking requests submitted by users. Providers have the authority to either accept or cancel bookings based on their availability or service capacity. These actions directly affect the booking status visible to users. Providers can end their session by logging out of the system.

This interaction model ensures controlled onboarding of providers and maintains service quality and accountability within the platform.

Admin (System Administrator)

The Admin is responsible for managing the overall operation of the ServiLink system. The administrator begins by logging into the system using secure credentials. Once authenticated, the admin has full access to manage both users and providers.

The admin can add new users and providers, update their personal or service-related data, and delete accounts when necessary. Additionally, the admin can view all registered users and providers to monitor system activity and ensure data integrity.

The administrator’s role ensures proper system governance, user management, and operational control across the entire platform.

Actor	Core Responsibilities
User	Register, login, browse services, book providers, track bookings
Provider	Login, view bookings, accept or cancel service requests
Admin	Manage users and providers, update data, maintain system control

(User ↔ System ↔ Provider)

5.System Development Methodology

Agile Scrum Methodology for ServiLink

5.1 Overview of the Methodology

The ServiLink platform adopts the **Agile Scrum methodology** as its system development approach. Agile Scrum is an iterative and incremental software development framework that emphasizes continuous delivery, collaboration, adaptability to change, and high-quality system output.

This methodology enables ServiLink to be developed as a **dynamic, user-centered service platform**, where system features are continuously refined based on real user needs and operational feedback. Rather than delivering the system as a single final product, Scrum supports the development of ServiLink in multiple small, functional releases known as *sprints*.

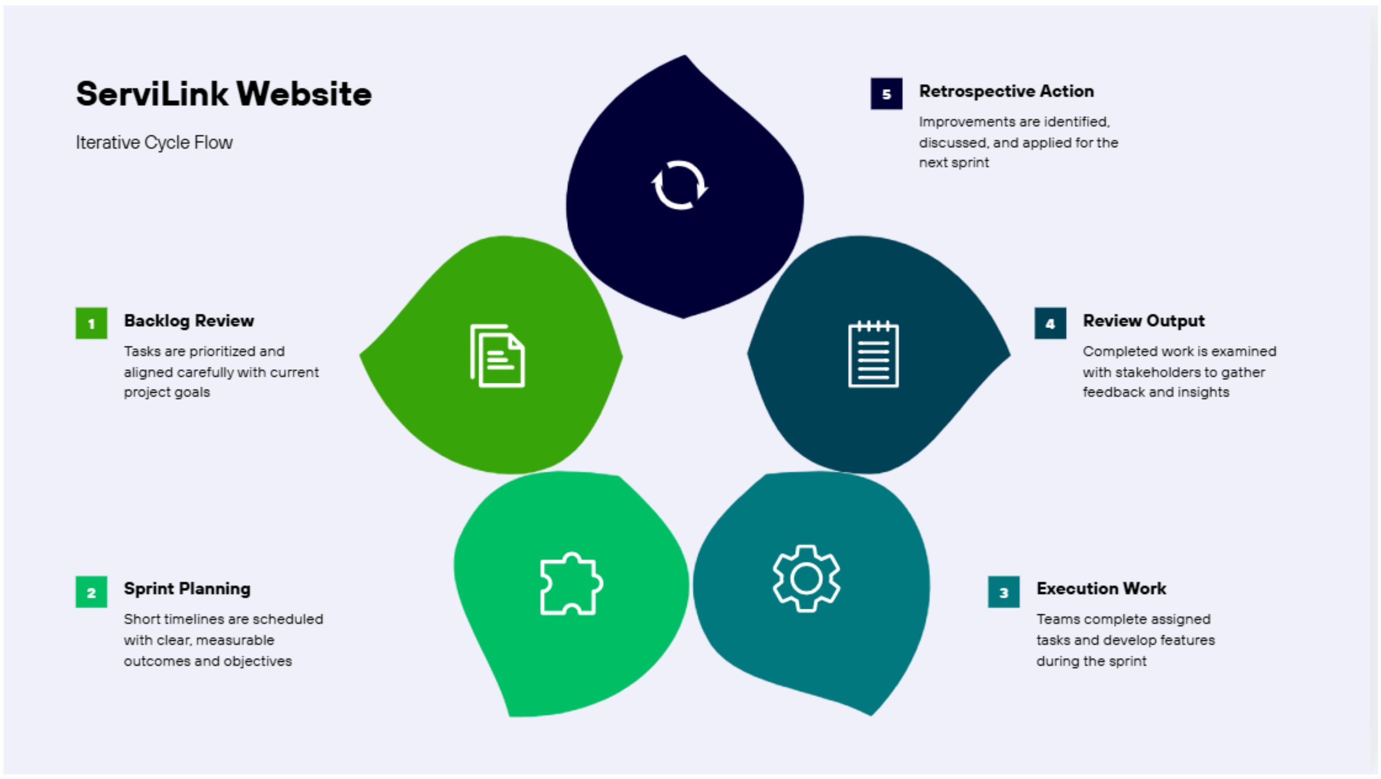

ServiLink Website
Iterative Cycle Flow
1 Backlog Review
Tasks are prioritized and aligned carefully with current project goals
2 Sprint Planning
Short timelines are scheduled with clear, measurable outcomes and objectives
3 Execution Work
Teams complete assigned tasks and develop features during the sprint
4 Review Output
Completed work is examined with stakeholders to gather feedback and insights
5 Retrospective Action
Improvements are identified, discussed, and applied for the next sprint

Figure : Iterative Cycle ServiLink Website Diagram

5.2 Rationale for Selecting Agile Scrum

The Agile Scrum methodology was selected for ServiLink due to several strategic and technical advantages:

5.2.1 Flexibility and Adaptability

ServiLink is a multi-role platform involving users, service providers, and administrators. Each actor has evolving requirements that cannot be fully predicted at the initial stage. Scrum allows requirements to change throughout the development lifecycle without negatively impacting system progress.

5.2.2 User-Centered Development

The success of ServiLink depends heavily on usability and user experience. Scrum supports early prototyping, frequent testing, and continuous user feedback, ensuring the platform remains aligned with stakeholder expectations.

5.2.3 Risk Mitigation

By delivering system features incrementally, critical technical risks such as integration errors, performance issues, and design flaws are identified early. This significantly reduces the cost of failure and improves overall system reliability.

5.2.4 Faster Time to Market

Scrum allows ServiLink to release usable system components early, enabling the team to evaluate functionality while development continues.

5.3 Agile Scrum Process Applied to ServiLink

The Agile Scrum process for ServiLink follows a continuous iterative cycle consisting of the following stages:

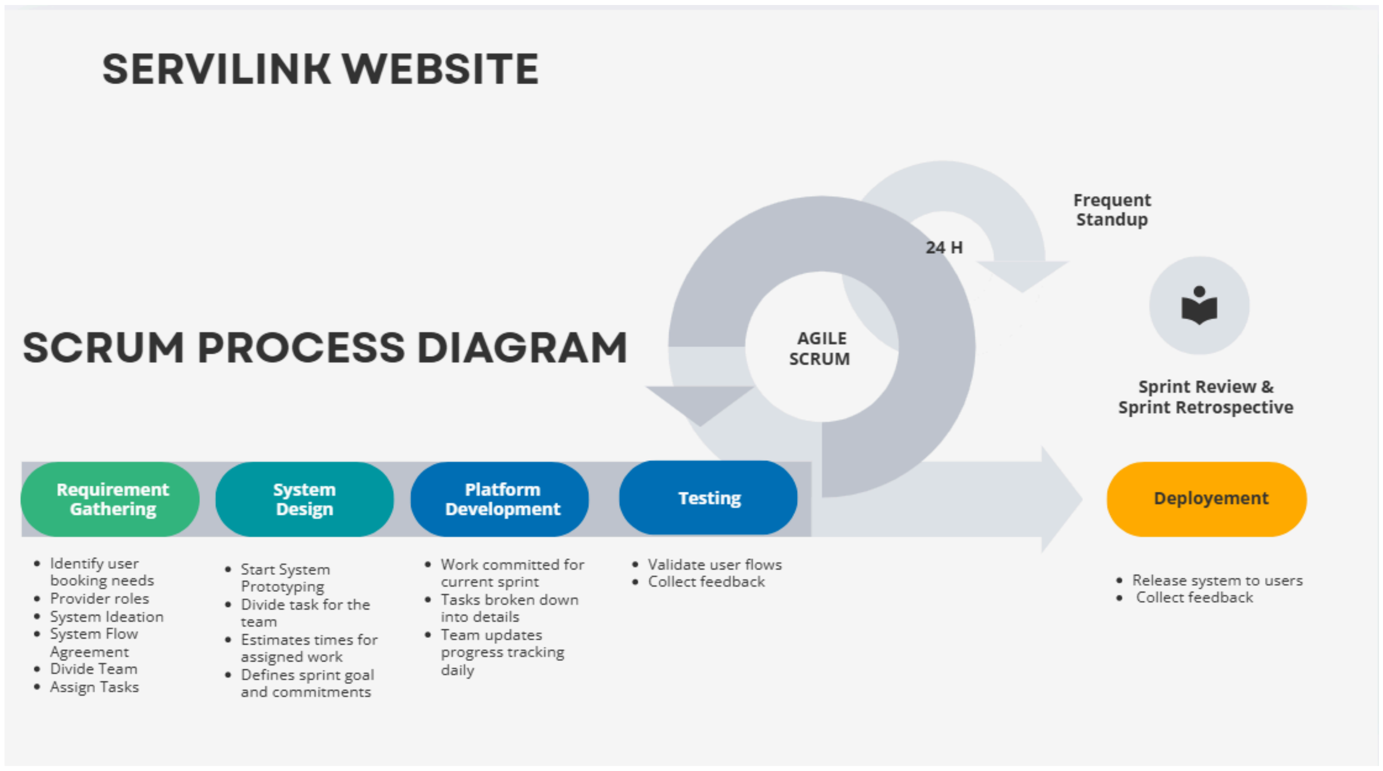


Figure : Agile Scrum Servilink Diagram

5.3.1 Requirement Gathering

This phase focuses on identifying system needs and stakeholder expectations. For ServiLink, this includes:

- User booking requirements
- Provider role definitions
- Administrative control features
- System workflows and functional boundaries

All requirements are documented as **user stories** and stored in the product backlog.

5.3.2 System Design

During this phase, system architecture and interface prototypes are developed. Key activities include:

- Designing dashboards for users, providers, and admins

- Defining system modules and data flow
- Assigning tasks to the development team
- Estimating development time and sprint goals

This ensures a clear technical blueprint before implementation begins.

5.3.3 Platform Development

This phase involves the actual implementation of system features. Tasks are broken down into smaller units and developed within short sprint cycles. Daily stand-up meetings are conducted to:

- Track progress
- Identify obstacles
- Ensure alignment with sprint objectives

Each sprint produces a functional system increment.

5.3.4 Testing

Testing is performed continuously to validate system behavior and quality. For ServiLink, this includes:

- Testing booking workflows
- Validating user roles and permissions
- Collecting usability feedback

Testing ensures that each sprint output is stable and reliable.

5.3.5 Deployment

The deployment phase releases the completed system features to users. This includes:

- System installation
- Final validation
- Stakeholder review
- User feedback collection

The deployed version becomes the foundation for the next development cycle.

5.4 Continuous Improvement through Scrum Events

Scrum incorporates structured feedback mechanisms such as:

- **Daily stand-ups** – Monitor progress and resolve issues
- **Sprint reviews** – Demonstrate completed work
- **Sprint retrospectives** – Reflect and improve team processes

These events enable ServiLink to evolve continuously and remain aligned with stakeholder needs.

5.5 Strategic Benefits of Agile Scrum for ServiLink

Aspect	Benefit
Development	Incremental feature delivery
Quality	Continuous testing and validation
Management	High transparency and accountability
Users	Better usability and satisfaction
System	Scalable and maintainable architecture

Last but not least,ServiLink utilizes the Agile Scrum methodology to enable flexible, iterative, and user-centered system development, ensuring continuous improvement, early delivery, and long-term system sustainability.

6.System Frontend

6.1ServiLinkUser Dashboard

6.1.1 Overview of the UI/UX Design

The ServiLink website is designed using a **user-centered design approach**, focusing on simplicity, clarity, and efficiency. The interface aims to provide users with a seamless experience from their first interaction with the system, enabling easy navigation, quick access to services, and intuitive task completion.

ServiLink delivers a clean, user-centered UI/UX design that ensures easy navigation, fast access to services, and a seamless digital experience across all pages

The overall design follows modern web usability principles, ensuring consistency across all pages and maintaining a strong visual identity.

6.1.2 Login Page

Purpose

The Login page allows registered users and providers to securely access the ServiLink system.

UI/UX Features

- Minimalist layout with clear input fields
- Prominent login button
- Error feedback for incorrect credentials
- Option to navigate to registration

UX Principles Applied

- **Simplicity** – Only essential fields are displayed
- **Clarity** – Labels and placeholders guide user input
- **Feedback** – Validation messages enhance error recovery

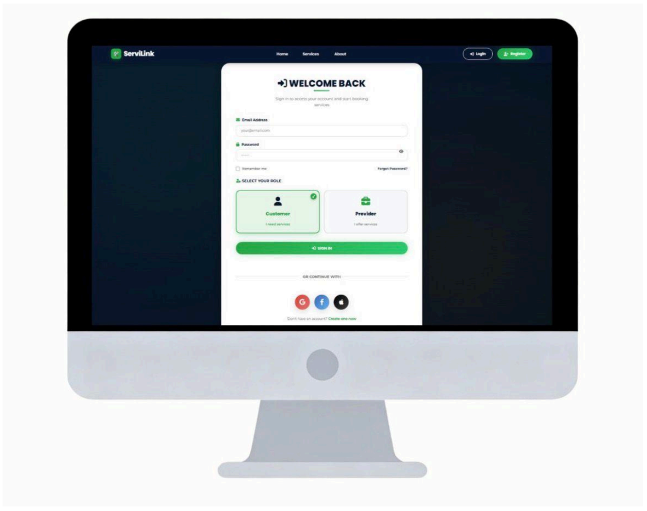


Figure:Login Page UI

6.1.3 Registration Page

Purpose

The Registration page enables new users to create an account and join the ServiLink platform.

UI/UX Features

- Structured form layout
- Logical grouping of personal details
- Clear submission flow
- Confirmation feedback after successful registration

UX Principles Applied

- **Progressive disclosure** – Users only see relevant fields
- **Error prevention** – Required fields are validated
- **User confidence** – Clear instructions reduce friction

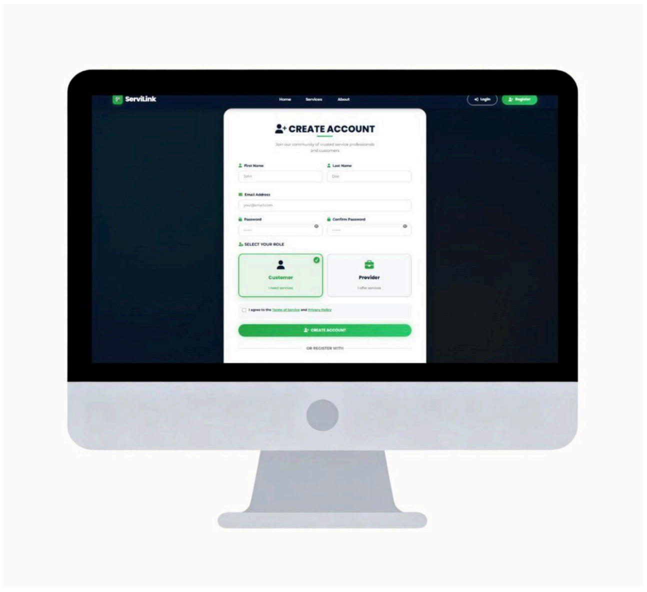


Figure :Registration Page UI

6.1.4 Home Page

Purpose

The Home page serves as the **entry point** to the ServiLink platform and introduces users to the system's core purpose.

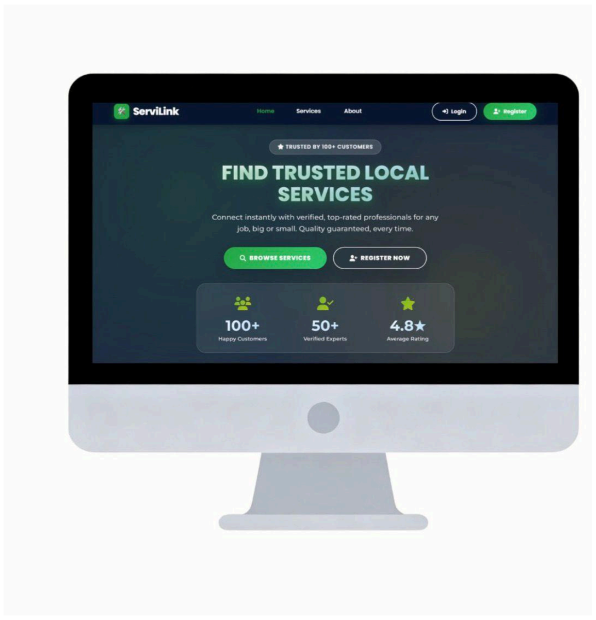
UI/UX Features

- Hero section with system overview
- Call-to-action buttons (e.g., "Get Started")
- Navigation menu for quick access

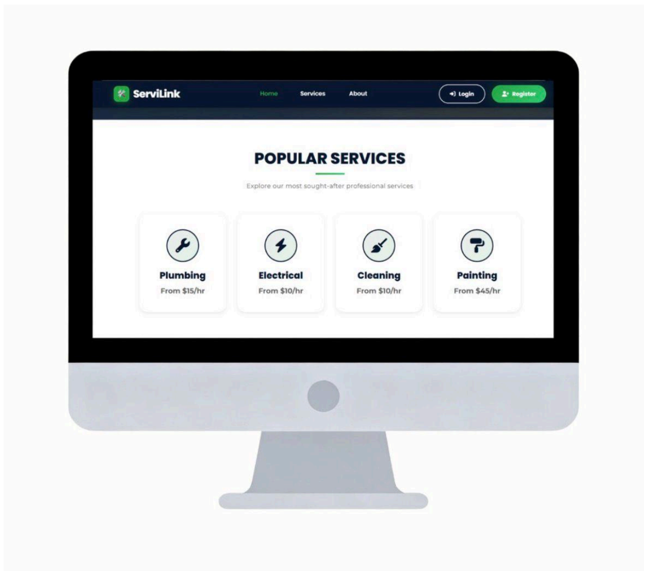
- Clean content layout

UX Principles Applied

- **First impression design** – Builds trust and credibility
- **Visual hierarchy** – Important information appears first
- **Guided navigation** – Users are directed toward key actions



Connect instantly with verified, top-rated professionals for any job, big or small. Quality guaranteed, every time.



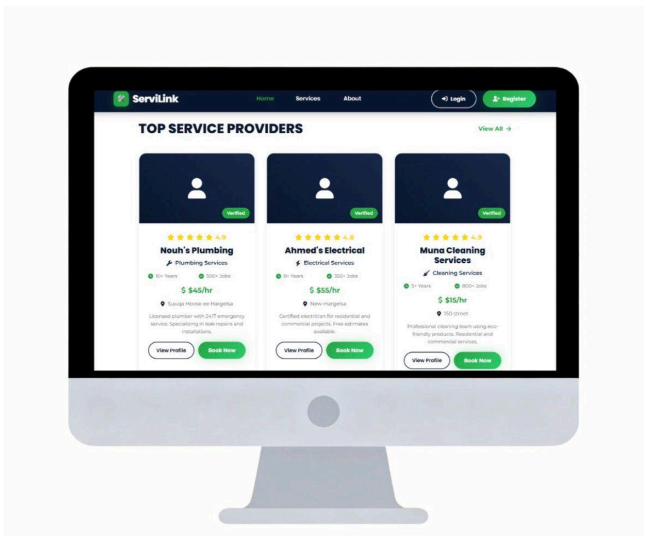


Figure:Home Page UI

6.1.5 Service Page

Purpose

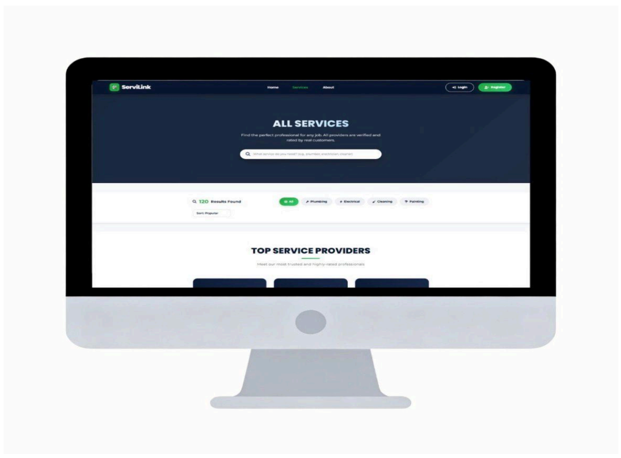
The Service page allows users to explore available services and select the type of service they need.

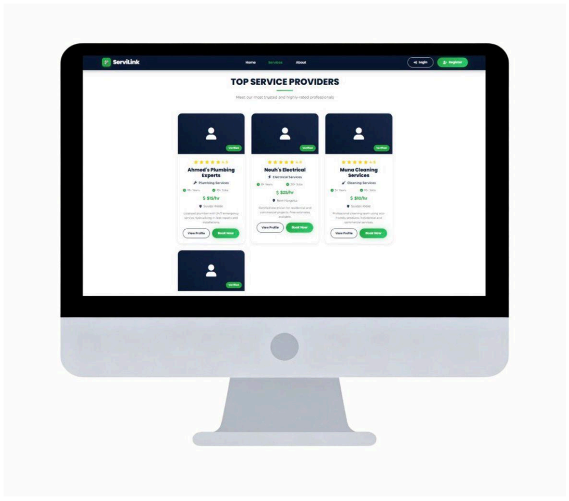
UI/UX Features

- Service categories displayed as cards or lists
- Clickable elements for interaction
- Clear service descriptions
- Easy transition to provider listings

UX Principles Applied

- **Discoverability** – Services are easy to browse
- **Recognition over recall** – Users choose visually instead of memorizing
- **Efficiency** – Minimal steps to reach booking





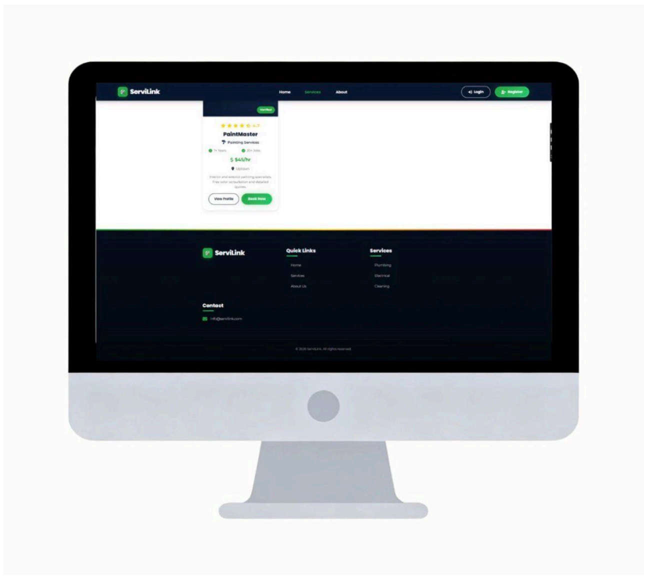


Figure:Service Page UI

6.1.6 About Page

Purpose

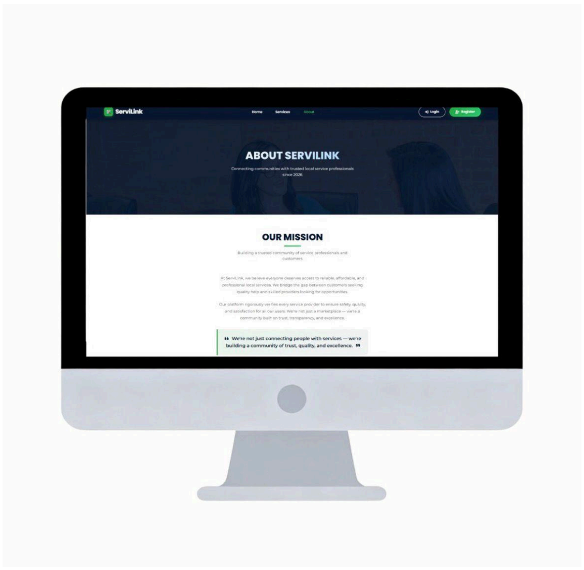
The About page provides background information about ServiLink, its mission, and its objectives.

UI/UX Features

- Informative content sections
- Clean typography for readability
- Visual elements supporting brand story

UX Principles Applied

- **Trust building** – Enhances system credibility
- **Transparency** – Users understand the platform's purpose
- **Brand consistency** – Reinforces identity



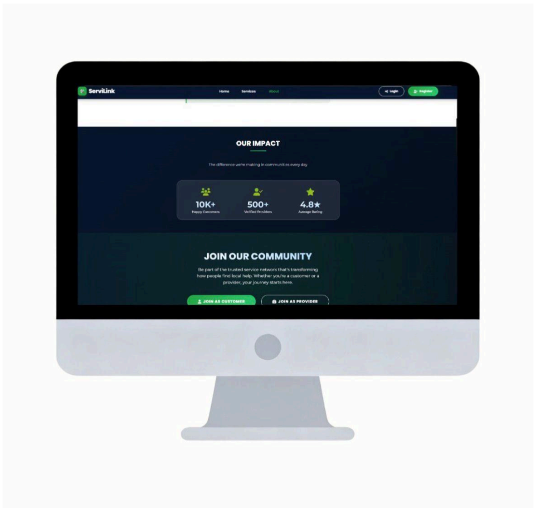
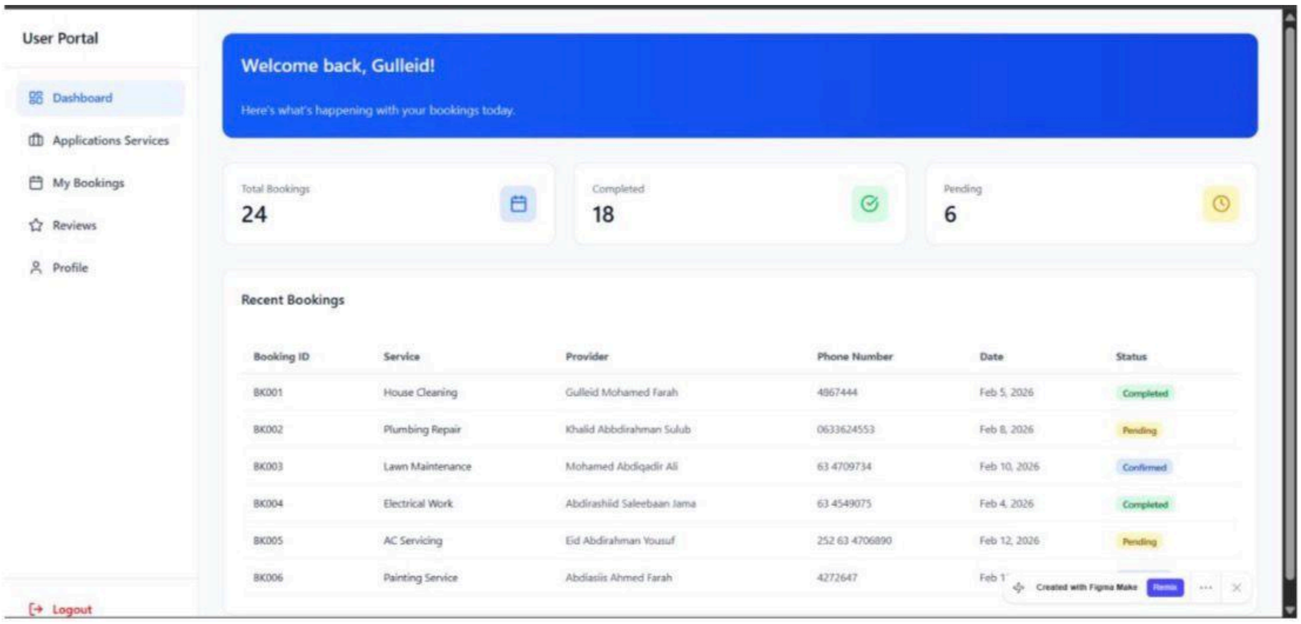


Figure:ServiLink About Page UI

6.2 Provider Dashboard UI/UX


User Portal

- Dashboard
- Applications Services
- My Bookings
- Reviews
- Profile

Logout

Welcome back, Gulleid!

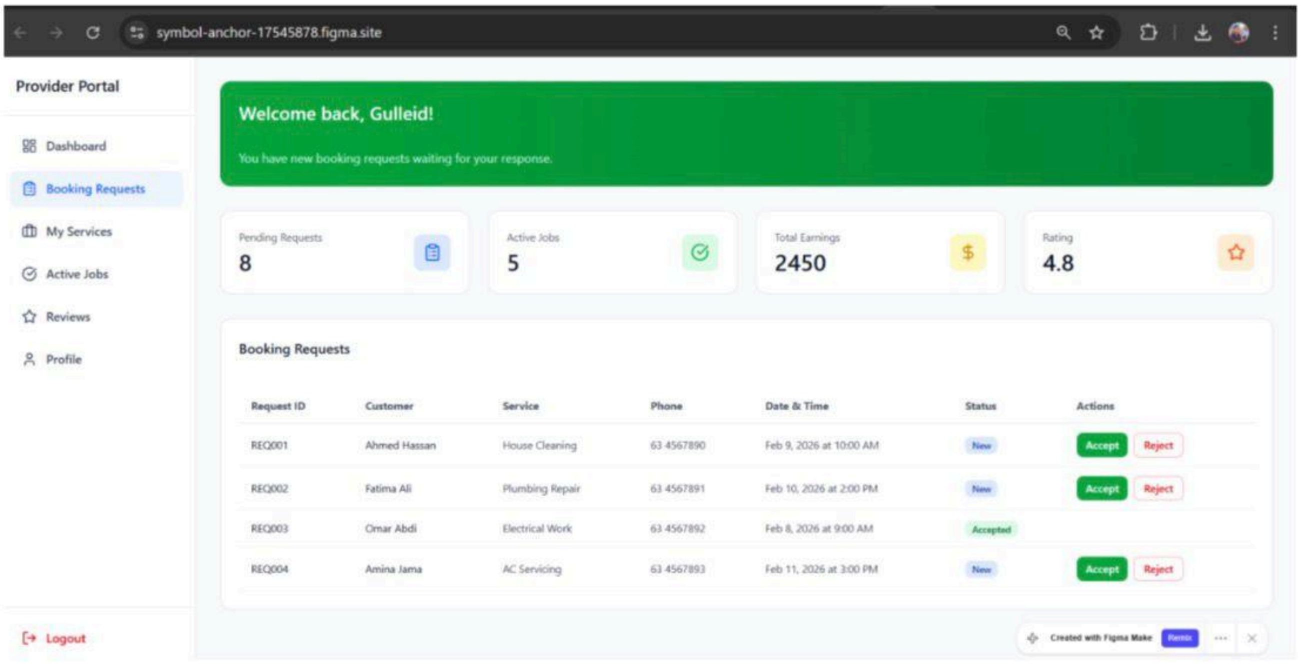
Here's what's happening with your bookings today.

Total Bookings	Completed	Pending
24	18	6

Recent Bookings

Booking ID	Service	Provider	Phone Number	Date	Status
BK001	House Cleaning	Gulleid Mohamed Farah	4867444	Feb 5, 2026	Completed
BK002	Plumbing Repair	Khalid Abbdirahman Sulub	0633624553	Feb 8, 2026	Pending
BK003	Lawn Maintenance	Mohamed Abdiqadir Ali	63 4709734	Feb 10, 2026	Confirmed
BK004	Electrical Work	Abdirashiid Saleebaan Jama	63 4549075	Feb 4, 2026	Completed
BK005	AC Servicing	Eid Abdirahman Yousuf	252 63 4706890	Feb 12, 2026	Pending
BK006	Painting Service	Abdiasiis Ahmed Farah	4272647	Feb 1	

Created with Figma Make


symbol-anchor-17545878.figma.site

Provider Portal

- Dashboard
- Booking Requests
- My Services
- Active Jobs
- Reviews
- Profile

Logout

Welcome back, Gulleid!

You have new booking requests waiting for your response.

Pending Requests	Active Jobs	Total Earnings	Rating
8	5	2450	4.8

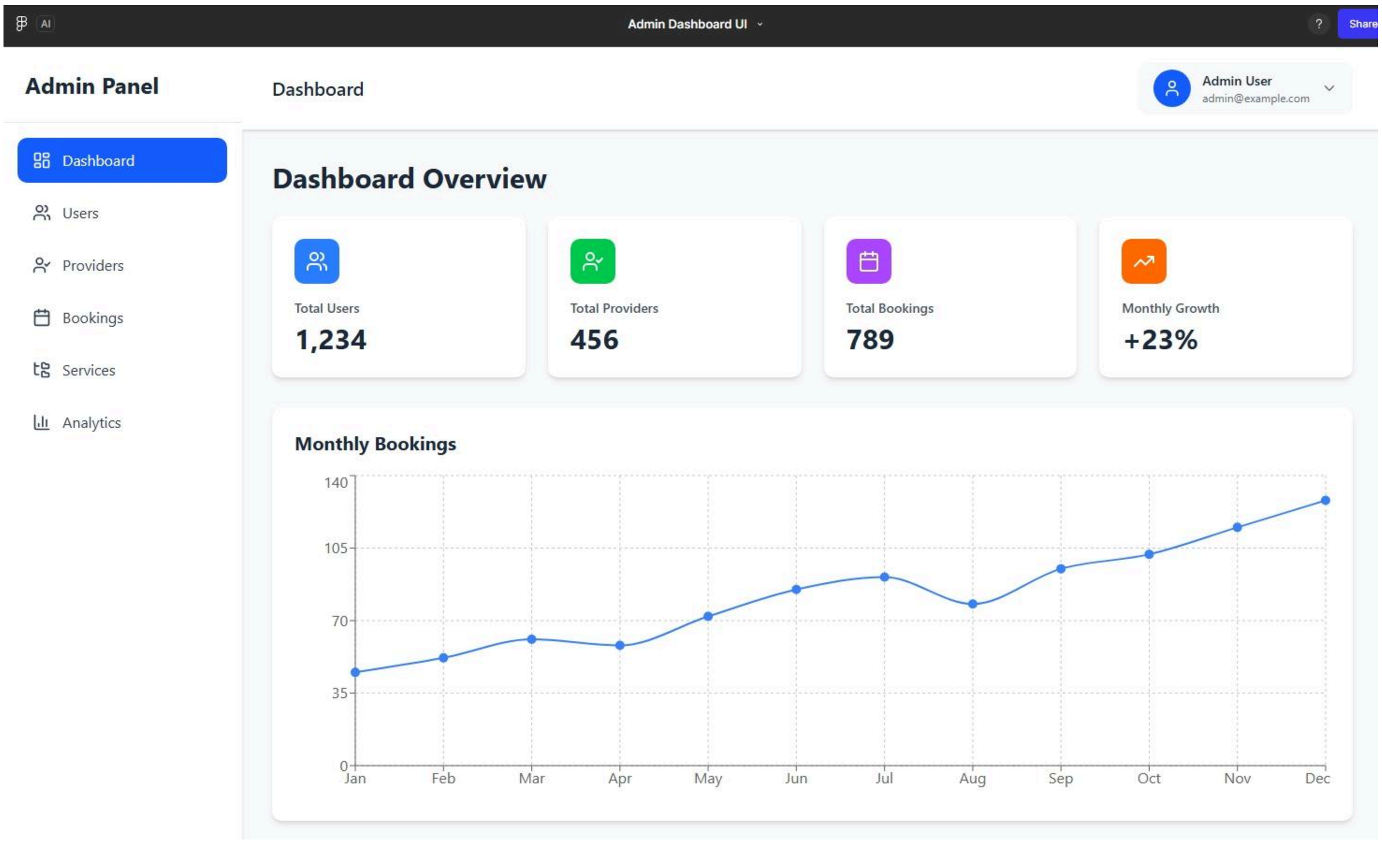
Booking Requests

Request ID	Customer	Service	Phone	Date & Time	Status	Actions
REQ001	Ahmed Hassan	House Cleaning	63 4567890	Feb 9, 2026 at 10:00 AM	New	Accept Reject
REQ002	Fatima Ali	Plumbing Repair	63 4567891	Feb 10, 2026 at 2:00 PM	New	Accept Reject
REQ003	Omar Abdi	Electrical Work	63 4567892	Feb 8, 2026 at 9:00 AM	Accepted	
REQ004	Amina Jama	AC Servicing	63 4567893	Feb 11, 2026 at 3:00 PM	New	Accept Reject

Created with Figma Make

Figure:Provider Dashboard

6.3.Admin Dashboard UI/UX



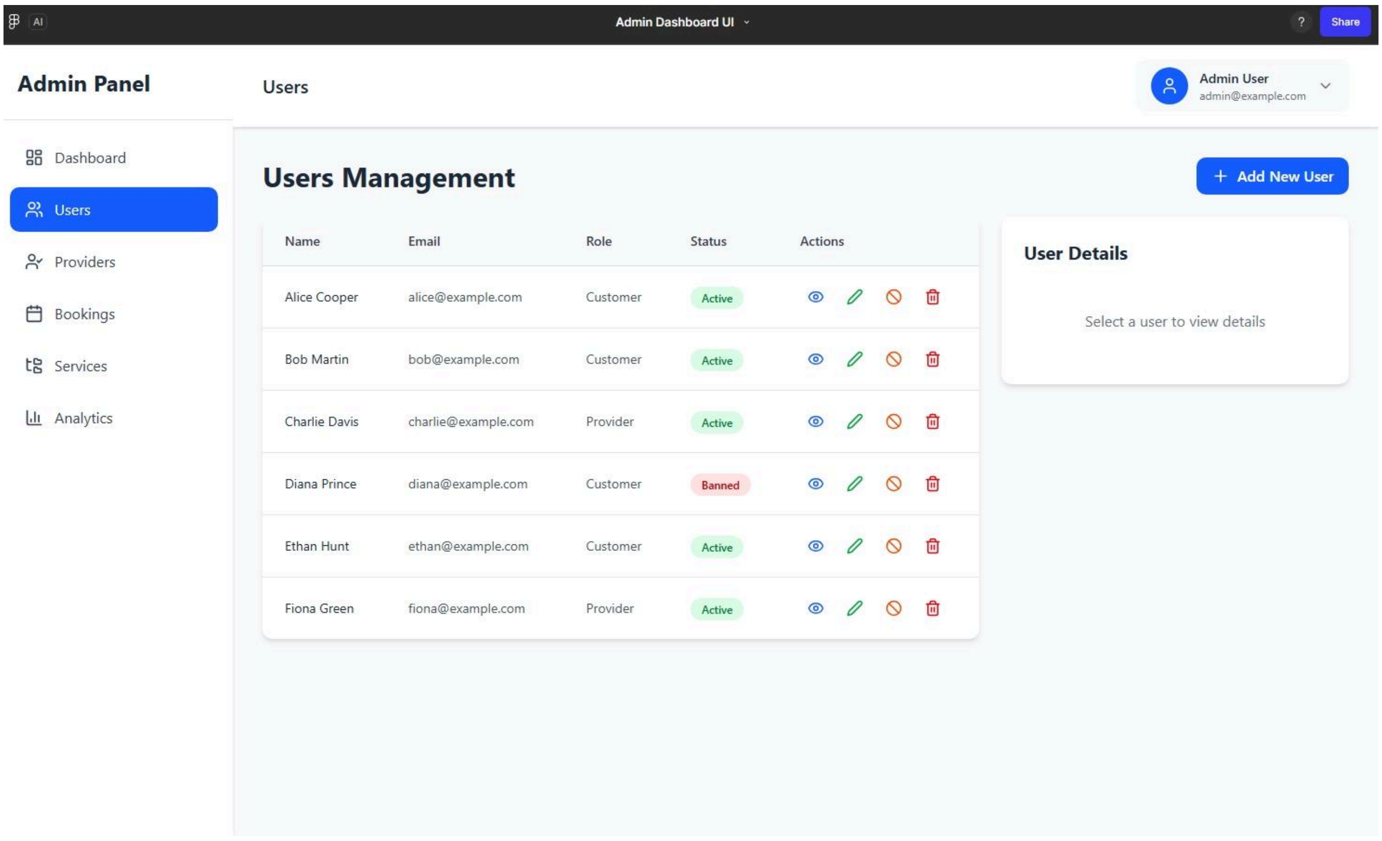
Dashboard Overview

Total Users 1,234

Total Providers 456

Total Bookings 789

Monthly Growth +23%



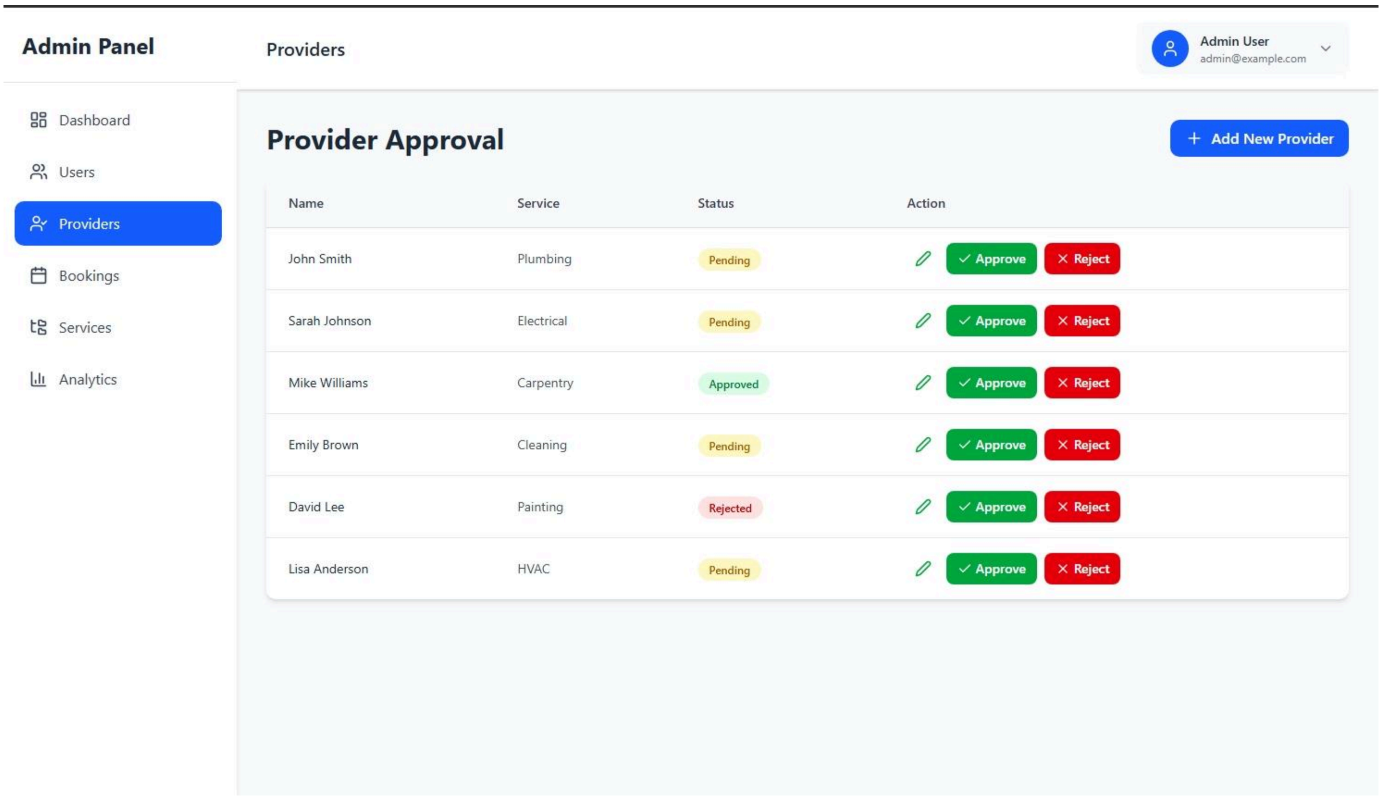
Users

Users Management

Name	Email	Role	Status	Actions
Alice Cooper	alice@example.com	Customer	Active	
Bob Martin	bob@example.com	Customer	Active	
Charlie Davis	charlie@example.com	Provider	Active	
Diana Prince	diana@example.com	Customer	Banned	
Ethan Hunt	ethan@example.com	Customer	Active	
Fiona Green	fiona@example.com	Provider	Active	

User Details

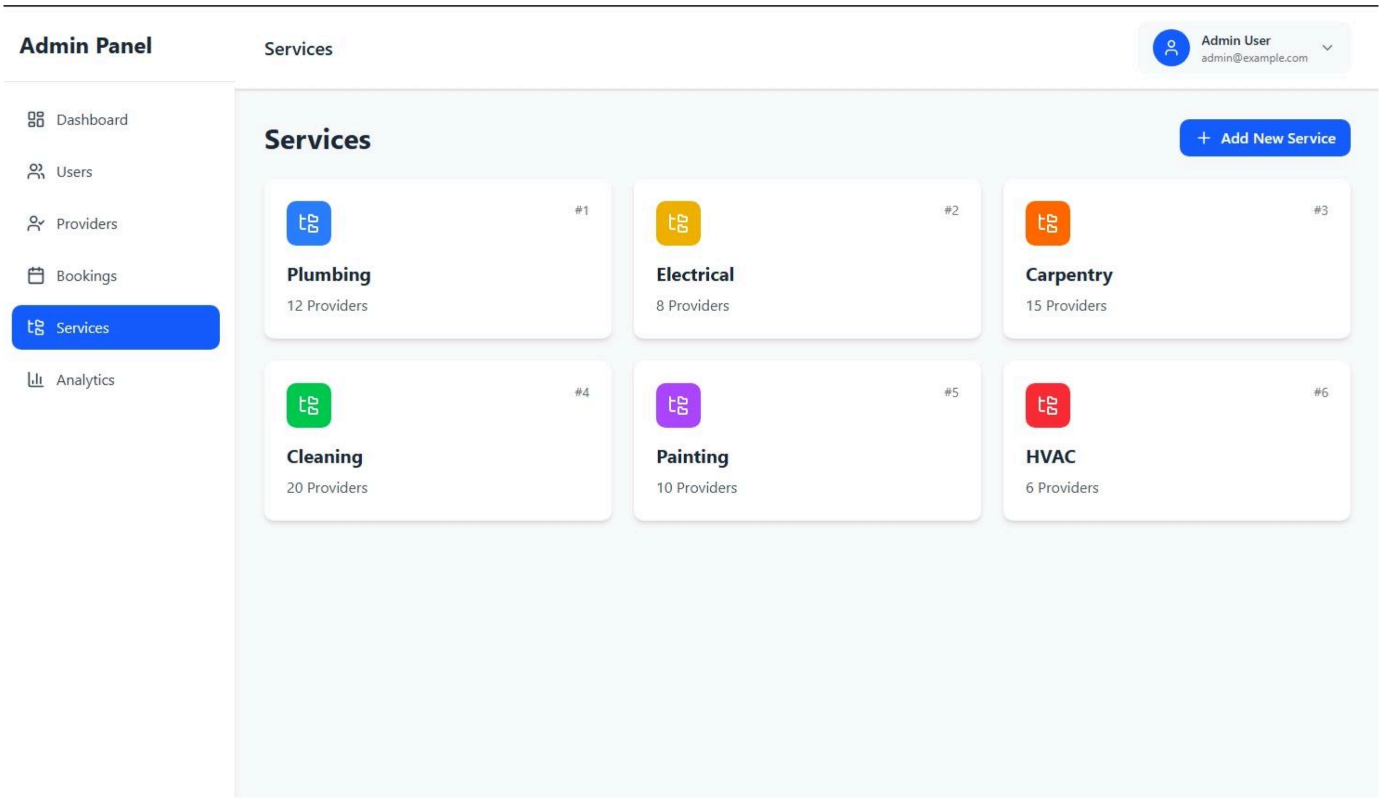
Select a user to view details



Providers

Provider Approval

Name	Service	Status	Action
John Smith	Plumbing	Pending	Approve Reject
Sarah Johnson	Electrical	Pending	Approve Reject
Mike Williams	Carpentry	Approved	Approve Reject
Emily Brown	Cleaning	Pending	Approve Reject
David Lee	Painting	Rejected	Approve Reject
Lisa Anderson	HVAC	Pending	Approve Reject



Admin Panel

Services

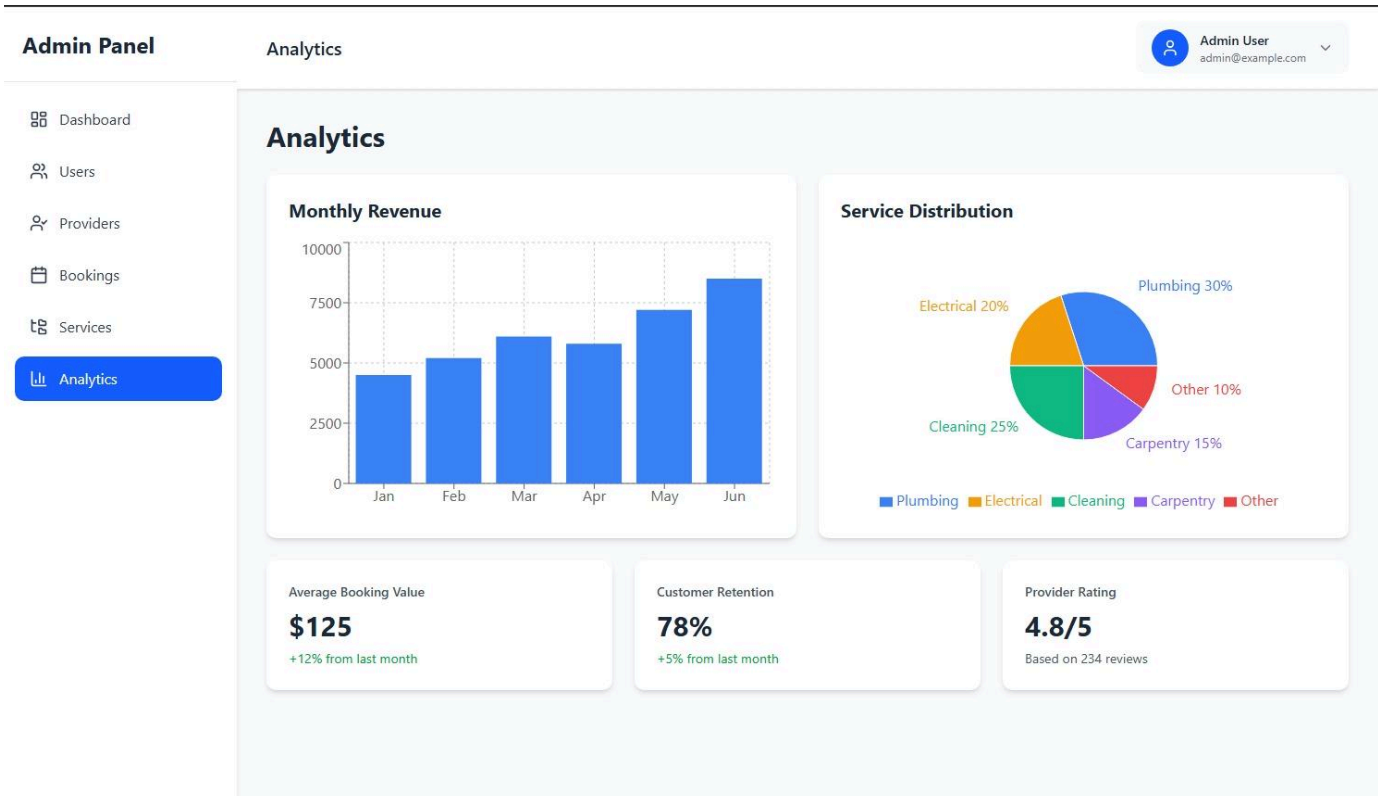
Admin User
admin@example.com

- Dashboard
- Users
- Providers
- Bookings
- Services
- Analytics

Services

+ Add New Service

#1 Plumbing 12 Providers	#2 Electrical 8 Providers	#3 Carpentry 15 Providers
#4 Cleaning 20 Providers	#5 Painting 10 Providers	#6 HVAC 6 Providers



Admin Panel

Analytics

Admin User
admin@example.com

- Dashboard
- Users
- Providers
- Bookings
- Services
- Analytics

Analytics

Monthly Revenue

Service Distribution

Average Booking Value
$125
+12% from last month

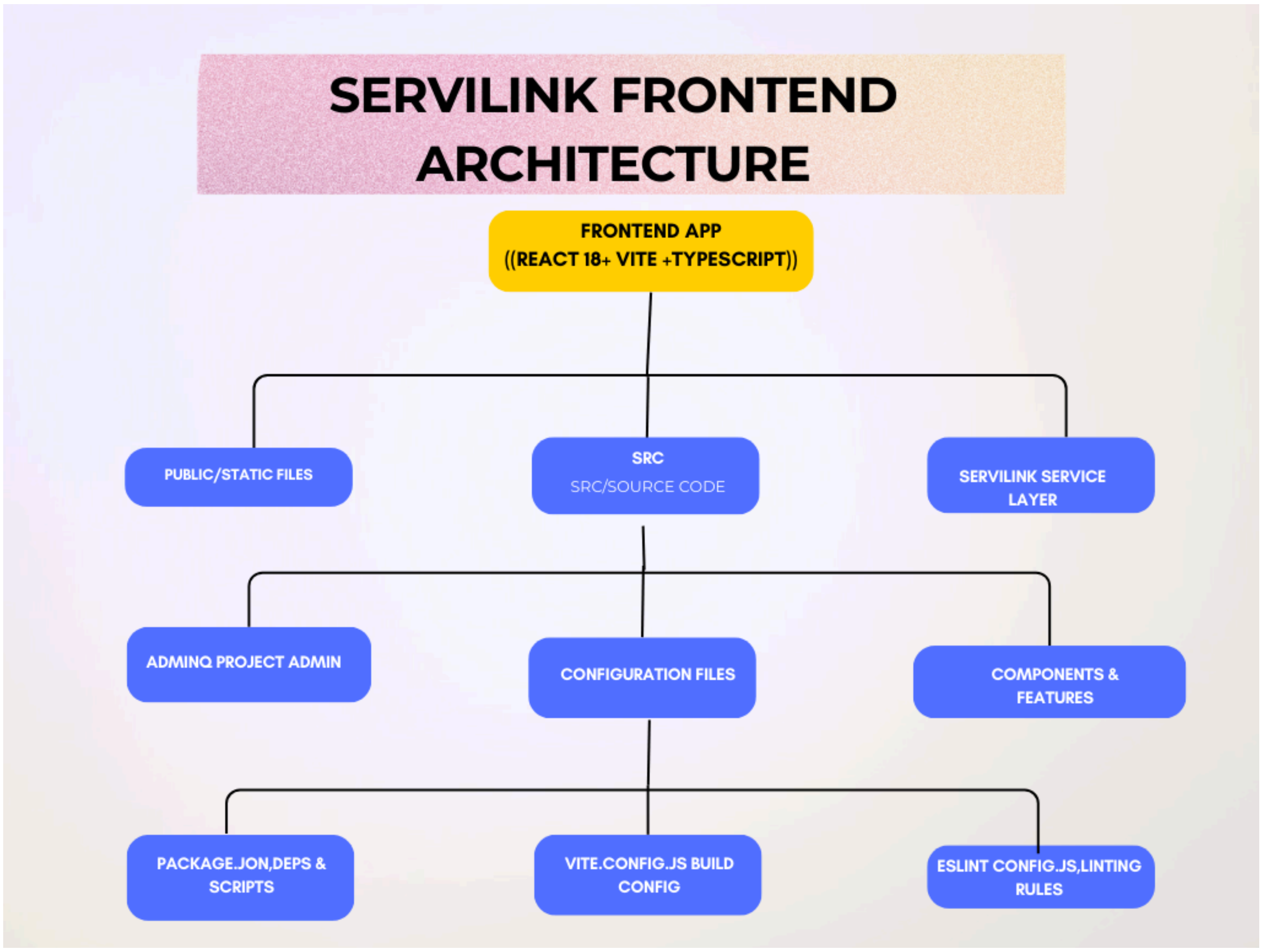
Customer Retention
78%
+5% from last month

Provider Rating
4.8/5
Based on 234 reviews

Figure:Admin Dashboard

6.4.Detailed Frontend Architecture

This frontend application is built with React 18 and Vite, providing a fast, modern development experience with optimized build performance. The architecture follows a modular structure as shown below:



SERVILINK FRONTEND ARCHITECTURE
FRONTEND APP ((REACT 18+ VITE +TYPESCRIPT))
PUBLIC/STATIC FILES
SRC SRC/SOURCE CODE
SERVILINK SERVICE LAYER
ADMINQ PROJECT ADMIN
CONFIGURATION FILES
COMPONENTS & FEATURES
PACKAGE.JON,DEPS & SCRIPTS
VITE.CONFIG.JS BUILD CONFIG
ESLINT CONFIG.JS,LINTING RULES

Figure:Frontend Architecture Diagram

State management is handled through React Context API for global state sharing across components, while React Router DOM manages client-side navigation and routing between different views. The Axios HTTP client is integrated within the servilink service layer to handle API communications, ensuring clean separation of concerns between business logic and UI components.

7.Detailed Backend Architecture

This backend application is built with Node.js and Express, following a clean MVC (Model-View-Controller) architecture pattern for scalable API development. The architecture is organized into modular layers as shown below:

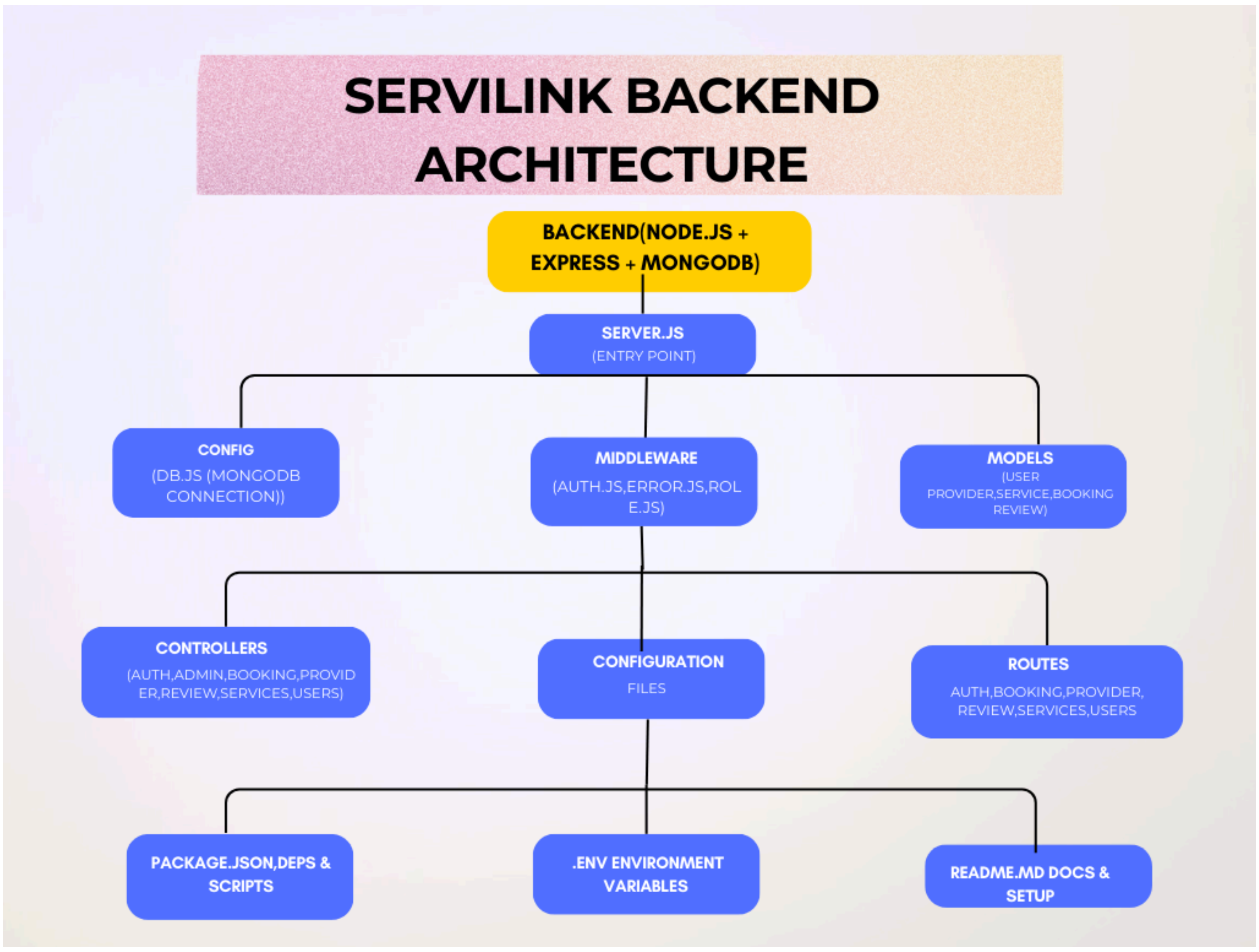

SERVILINK BACKEND ARCHITECTURE
BACKEND(NODE.JS + EXPRESS + MONGODB)
SERVER.JS (ENTRY POINT)
CONFIG (DB.JS (MONGODB CONNECTION))
MIDDLEWARE (AUTH.JS,ERROR.JS,ROLE.JS)
MODELS (USER PROVIDER,SERVICE,BOOKING REVIEW)
CONTROLLERS (AUTH,ADMIN,BOOKING,PROVIDER,REVIEW,SERVICES,USERS)
CONFIGURATION FILES
ROUTES AUTH,BOOKING,PROVIDER, REVIEW,SERVICES,USERS
PACKAGE.JSON,DEPS & SCRIPTS
.ENV ENVIRONMENT VARIABLES
README.MD DOCS & SETUP

Figure:Backend Architecture Diagram

State management is handled through React Context API for global state sharing across components, while React Router DOM manages client-side navigation and routing between different views. The Axios HTTP client is integrated within the servilink service layer to handle API communications, ensuring clean separation of concerns between business logic and UI components.

9.Api EndPoint Documentation

9.1. Auth APIs

Handles user authentication and session management.

Method	Endpoint	Description	Request Body (JSON)	Permissions
POST	/Auth/register	Register a new user	{"name","email","password","role	No
POST	/Auth/login	Login to get token	{"name","email","password","role	No
GET	/Auth/me	Get current logged – in use	–	Yes
POST	/auth/logout	Logout user (invalidate token)	Logout user (invalidate token)	Yes

9.2. User APIs

Handles user profile management and administration.

Method	*Endpoint*	*Description*	*Request Body (JSON)*	*Permissions*
GET	/users/profile	*Get logged–in user's profile*	–	User
POST	/users/profile	*Create/Update user profile*	*Profile Data*	User
GET	/users	*Get all users*	-	Admin
GET	/users/:id	*Get user by ID*	-	*Authenticated*
PUT	/users/:id	*Update user*	*Update Data*	*Authenticated*
DELETE	/users/:id	*Delete user*	-	Admin

9.3. Provider APIs

Handles service provider profiles.

Method	***Endpoint***	***Description***	***Request Body (JSON)***	***Permissions***
POST	/providers/profile	*Create provider profile*	*Profile Data*	*Provider*

GET	/providers	Get list of all providers	Profile Data	Public
GET	/providers/:id	Get provider by ID	-	Public
PUT	/providers/profile	Update provider profile	{"bio", "location", "experience"}	Provider

9.4. Service APIs

Handles services offered by providers.

Method	Endpoint	Description	Request Body (JSON)	Permissions
POST	/services	Create a new service	{"title", "price", "description", "category"}	Provider / Admin
GET	/services	Get list of all services	-	Admin
GET	/services/:id	Get service by ID	-	Public
PUT	/services/:id	Update service	Update Data	Owner / Admin
DELET	/services/:id	Delete service	-	Owner / Admin

9.5. Booking APIs

Handles service booking lifecycle.

Method	Endpoint	Description	Request Body (JSON)	Permissions
POST	/bookings	Create a new booking	{"service Id", "provider Id", "date"}	User
GET	/bookings/customer	Get customer bookings	-	User
GET	/bookings/my	Get my	-	Authenticated

		bookings		
PUT	/bookings/:id/accept	Accept a booking	-	Provider
PUT	/bookings/:id/reject	Reject a booking	-	Provider
PUT	/bookings/:id/cancel	Cancel a booking	-	User / Provider
PUT	/bookings/:id/complete	Mark booking complete	-	Provider

9.6. Review APIs

Handles feedback and ratings.

Method	***Endpoint***	***Description***	***Request Body (JSON)***	***Permissions***
POST	/reviews	Create a review	{ "service Id", "rating", "comment" }	User
GET	/reviews/service/:serviceI	Get reviews for a service	-	Public
DELETE	/reviews/: id	Delete a review	-	Owner / Admin

9.7. Admin APIs

Administrative control over users, providers, and services.

Method	***Endpoint***	***Description***	***Permissions***
GET	/admin/users	Get all users	Admin
DELETE	/admin/users/:id	Delete a user	Admin
PUT	/admin/users/:id	Update a user	Admin
GET	/admin/bookings	Get all bookings	

			Admin
GET	/admin/providers	Get all providers	Admin
PUT	/admin/providers/:id/approve	Approve a provider	Admin
PUT	/admin/providers/:id/reject	Reject a provider	Admin
GET	/admin/services	Get all services	Admin
POST	/admin/services	Create service	Admin
PUT	/admin/services/:id	Update service	Admin
DELET	/admin/services/:id	Delete service	Admin

9.8. Common HTTP Status Codes & Errors

Code	**Status**	**Description**	**Solution**
401	Unauthorized	Token is missing, invalid, or expired.	Ensure the **Authorization** header is included with a valid **Bearer TOKEN**.
403	Forbidden	The user's role does not have permission for this action.	Check user role (User vs Admin vs Provider) against the required permissions.
404	Not Found	The requested resource or route does not exist.	Verify the ID parameter and the API endpoint spelling.
500	Server Error	An unexpected error occurred on the backend.	Check server logs or contact the system administrator.

9.**System Future Enhancement**

-Based on our development roadmap and stakeholder feedback, the following enhancements are planned for future versions of ServiLink:

-AI-Powered Chatbot: Integrate an intelligent chatbot to assist customers in finding the right service providers based on natural language queries and automated support.

-Real-Time Chat: Implement Socket.io for direct messaging between customers and providers for better coordination.

-Payment Integration: Integrate local payment solutions like Telesom ZAAD or Stripe for secure in-app transactions.

-Push Notifications: Add email and SMS notifications using Nodemailer/Twilio to alert users about booking updates and approvals.

-Map & Location Services: Integrate Google Maps or Mapbox to show provider locations and enable location-based search.

-Mobile Application: Develop a React Native mobile app to provide on-the-go access for customers and providers.

-Advanced Search & Filters: Add filters for price range, availability, and rating to improve service discovery.

-Analytics Dashboard: Provide providers and admins with insights on bookings, earnings, and user engagement.

10.Links & Logo

This document serves as a complete guide for anyone that wants to know how Servilink works and has been designed and coded and structured.

Logo:

ServLink

ServLink

Link:

1:User Dashboard UI/UX: **https://tinyurl.com/servilink-user-UI**

2:Admin Dashboard UI/UX: **Admin Dashboard UI – Figma Make**

3:ServiLink Github Link: **the-gulleid/local_services_provider**