

---

## AllServ: A Comprehensive Service Provider Platform

---

### 1. Project Overview

**AllServ** is an innovative web application designed to connect users with trusted service providers for a wide range of home and office services. The platform streamlines the process of booking services such as plumbing, electrical work, cleaning, and repairs, ensuring a smooth, efficient experience for both customers and service providers. The solution is built using MongoDB, Django, HTML, CSS, and JavaScript.

---

## 2. Key Features

### For Users:

- **Landing Page (index.html):**
  - A visually appealing landing page with a prominent search bar, popular service categories, and a “Get Started” call-to-action.
- **User Dashboard:**
  - Personalized area displaying current bookings, recommended services, and notifications.
- **Service Listing & Booking:**
  - Dynamic service search with filters, detailed service cards, and a booking process that captures date, time, and location.
- **Payment Integration:**
  - Secure payment gateway options (UPI, cards, net banking) along with an order summary and coupon code application.
- **Booking History & Support:**
  - Detailed log of past bookings with options to download invoices and leave feedback; support page for raising issues or initiating live chat.

### For Service Providers:

- **Login & Registration:**
  - Multi-step registration with document uploads and OTP verification to ensure authenticity.
- **Provider Dashboard:**
  - Overview of new requests, accepted bookings, earnings, and performance analytics.
- **Service Request Management:**
  - Easy-to-use interface to accept/reject booking requests and mark services as completed.
- **Service Management:**
  - CRUD operations for managing services offered, including adding new services, editing existing ones, and image uploads.
- **Earnings & Transaction Tracking:**
  - Graph-based earnings visualization, transaction history, and withdrawal options.
- **Profile & Settings:**

- Update personal and business information, along with account and notification settings.

## **For Administrators:**

- **Admin Dashboard:**
    - Comprehensive overview with key metrics such as total users, service providers, active bookings, and revenue trends.
  - **Provider Verification:**
    - Interface for reviewing and approving or rejecting new service provider applications.
  - **User & Provider Management:**
    - Tools for filtering and managing registered users and providers.
  - **Reports & Analytics:**
    - Interactive charts for monitoring platform performance and exporting reports.
  - **Content Management & Support:**
    - Manage service categories, FAQs, announcements, and resolve support tickets.
-

### 3. Technology Stack

- **Frontend:** HTML, CSS, JavaScript, Bootstrap
  - **Backend:** Django (Python)
  - **Database:** MongoDB
  - **Payment Integration:** Razorpay/Stripe (API-based integration)
  - **Additional Libraries:** Chart.js or D3.js for interactive data visualizations
-

## 4. Front-End Design Details (User-Focused)

### Index Page (index.html)

- **Purpose:**
  - Serve as the first point of interaction for users.
- **Components & Workflow:**
  - **Header & Navigation:** Contains the logo, navigation links, and login/register buttons.
  - **Hero Section:** Features a large banner with an engaging tagline and a search bar prompting “What service do you need?”.
  - **Popular Services:** Displays service categories using icon-based cards.
  - **How It Works:** Step-by-step visual guide explaining the service process.
  - **Footer:** Provides quick links, contact info, and social media buttons.
- **User Journey:**
  - Users arrive, explore the services through search or category selection, and proceed to either register or log in.

### Dashboard (dashboard.html)

- **Purpose:**
  - Offer a personalized space for logged-in users.
- **Components:**
  - **Sidebar Navigation:** Links to Home, My Bookings, Payment History, and Support.
  - **Main Area:** Displays a welcome banner, summary cards for bookings, and recommended services.
- **User Journey:**
  - After login, users immediately see a snapshot of their activity and can quickly navigate to other sections.

### Service Listing Page (services.html)

- **Purpose:**
  - Enable users to search and filter through available services.
- **Components:**
  - **Search & Filter Bar:** Input fields and dropdowns for narrowing down options by price, rating, and location.
  - **Service Cards:** Each card shows the provider’s details, service description, pricing, and a “Book Now” button.
- **User Journey:**

- Users input search criteria, view the filtered results, and select a service to book.

## Booking Page (booking.html)

- **Purpose:**
  - Capture and confirm booking details.
- **Components:**
  - **Service Details:** Displays provider info, service breakdown, and pricing.
  - **Booking Form:** Fields for selecting date, time, and service location.
  - **Payment Method:** Options for online payment or cash on delivery.
- **User Journey:**
  - Users review details, fill in booking information, and proceed to the payment page.

## Payment Page (payment.html)

- **Purpose:**
  - Facilitate secure payment for booked services.
- **Components:**
  - **Order Summary:** Details of the service, provider, and cost breakdown.
  - **Payment Options:** Multiple choices including UPI, cards, or net banking.
  - **Coupon Input & Confirm Button:** For discount codes and final payment confirmation.
- **User Journey:**
  - Users verify their order, select a payment method, and complete the transaction.

## Booking History Page (history.html)

- **Purpose:**
  - Maintain a record of all past and ongoing bookings.
- **Components:**
  - **Listing:** Accordion or table view displaying service name, provider, date, and status.
  - **Actions:** Options to download invoices or provide ratings and reviews.
- **User Journey:**
  - Users review historical data, access detailed information on each booking, and manage invoices.

## Support & Complaint Page (support.html)

- **Purpose:**

- Provide assistance and handle user queries.
  - **Components:**
    - **Support Form:** Fields for selecting issue type, subject, and message.
    - **Live Chat Widget:** Optional integration for real-time support.
  - **User Journey:**
    - Users submit their queries through the form and can use live chat for immediate assistance.
-

## 5. User Workflow Summary

### 1. Landing at index.html:

- Users are greeted with an engaging interface and a clear call to action.

### 2. Service Search & Selection:

- They use the search bar or click on popular services to navigate to detailed listings.

### 3. Booking & Payment:

- After selecting a service, users fill in the booking form and complete the payment process.

### 4. Post-Booking Management:

- Users can review their booking history, download invoices, and leave feedback.

### 5. Support Access:

- Should any issues arise, users can quickly access the support page for assistance.
-



## 6. How to Run the Front-End Locally

### 1. Clone the Repository:

```
git clone https://github.com/your-repo/allserv.git  
cd allserv
```

2.

### 3. Ensure Front-End Dependencies:

- Confirm Bootstrap and any required JavaScript libraries are available.

### 4. Start the Django Development Server:

```
python manage.py runserver
```

5.

### 6. Access the User Interface:

- Open your browser and navigate to <http://127.0.0.1:8000/> to interact with the index.html page.
-

## **7. Conclusion**

The AllServ user front-end is crafted to deliver an intuitive, responsive, and engaging user experience. From the moment a user lands on the index page to booking services, making payments, and accessing support, every step is designed for clarity and ease of use. This document outlines the key pages and workflows that form the backbone of the user experience on AllServ.