Home Page

Service Booking Portal

Home Admin About

Electrician

Expert electrical repair and installation services.

Plumbing

Professional plumbing and pipe fixing.

Cleaning

Residential and commercial cleaning solutions.

AC Repair

Air conditioner service and maintenance.

Book a Service

Your Name

Phone Number

Select Service

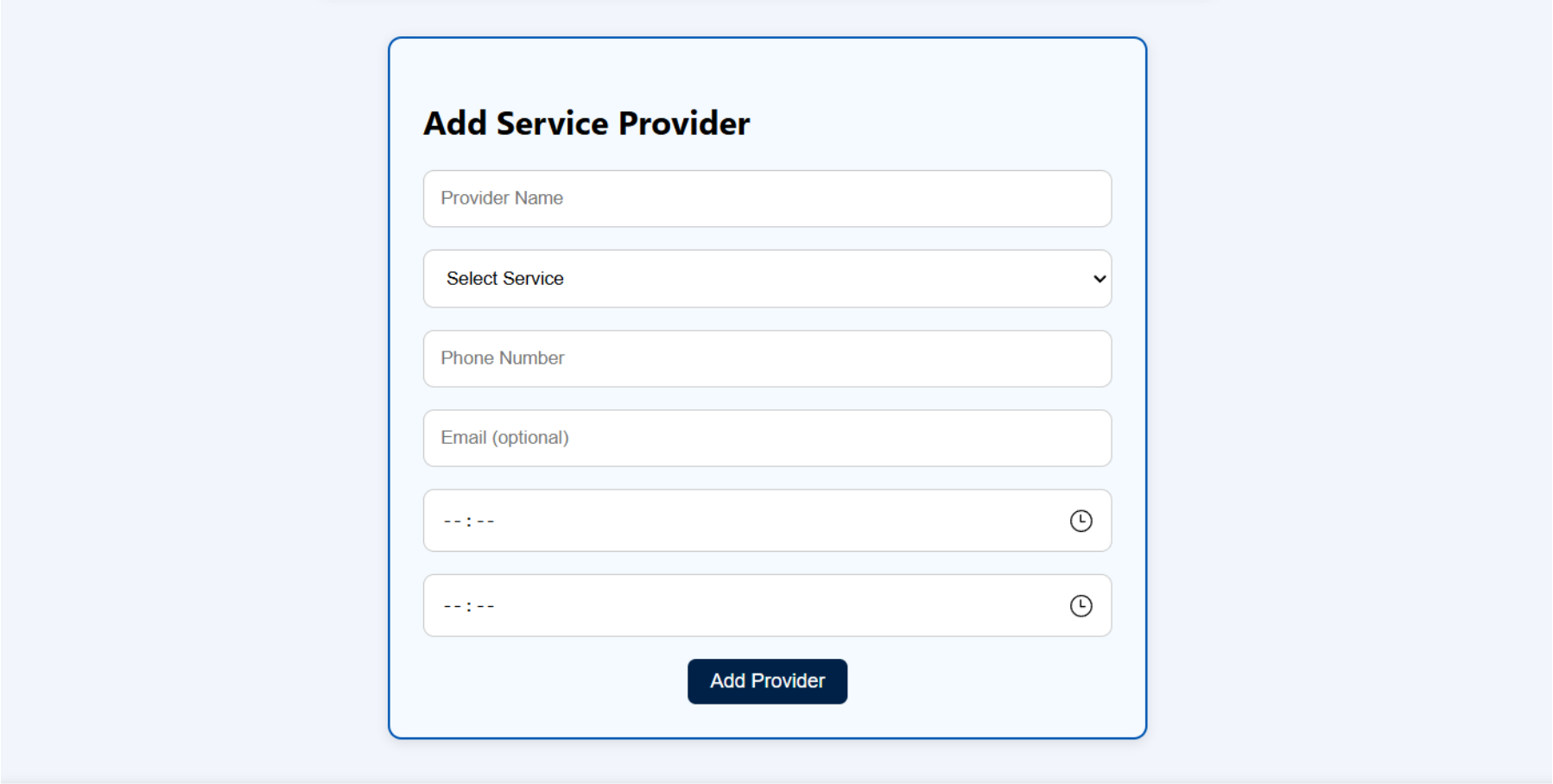
dd-mm-yyyy

--:--

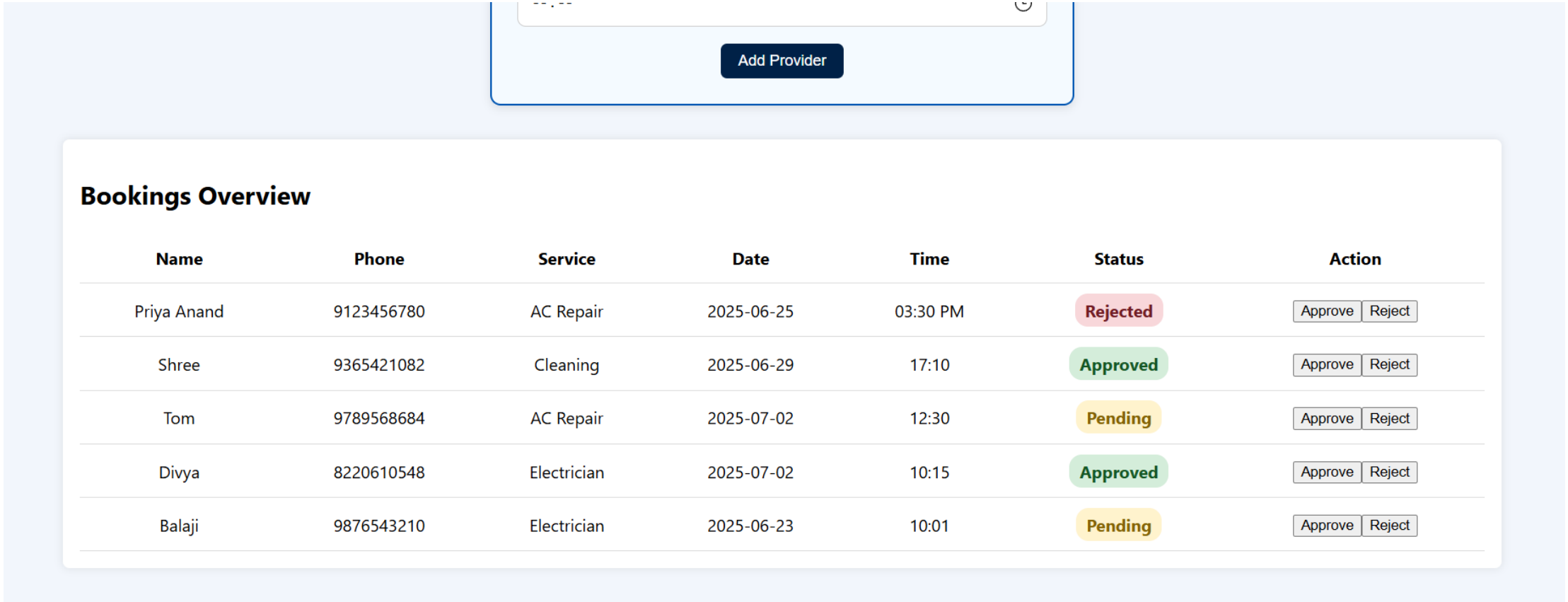
Submit

Admin Page (Passcode verification)

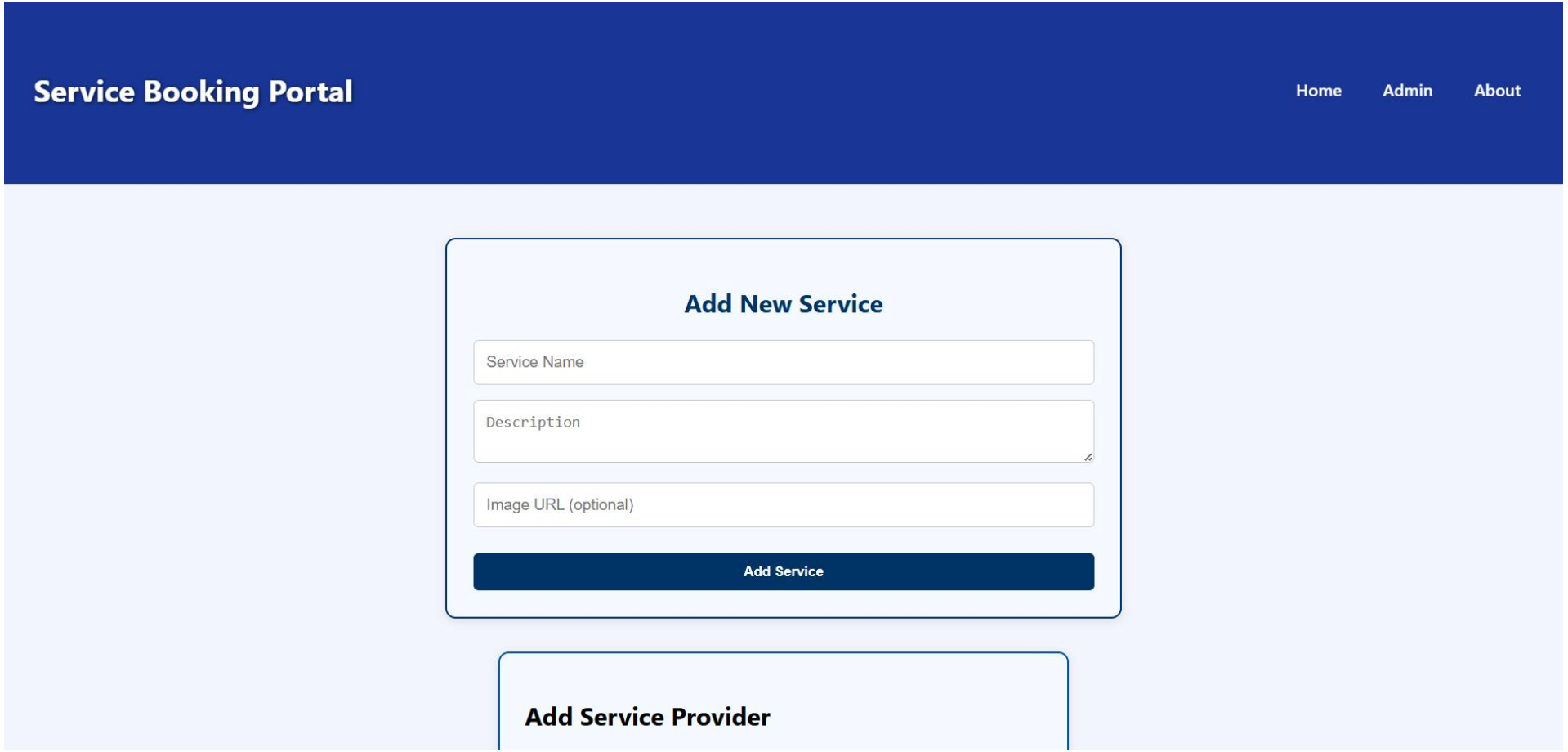
Admin page: (Add service provider)



Admin page: (Bookings dashboard)



Admin page: (Add new service)



About page:

About This Portal

This is a professional service booking platform for managing and scheduling services.

🔧 SERVER-SIDE ENGINEERING FUNDAMENTALS

Your backend is built using Node.js, Express.js, and MongoDB, applying key server-side concepts such as:

1. **Server & Framework: Node.js with Express.js**
   Node.js is a JavaScript runtime used for building scalable, event-driven servers.
   Express.js is a lightweight web application framework for Node.js that simplifies routing and middleware management.
   Key concepts used: Middleware functions for parsing request bodies and handling errors. Routing to define API endpoints like /services, /bookings, /providers. Controllers to separate logic from route definitions.
2. **RESTful API Architecture**
   Your backend exposes RESTful API endpoints, which means:
   - Stateless interactions: Each request is independent.
   - HTTP methods:
     GET → fetch data (services/bookings/providers)
     POST → create new entries (add service, book a service)
     PATCH → update existing data (e.g., update booking status)

2. **RESTful API Architecture**
   Your backend exposes RESTful API endpoints, which means:
   - Stateless interactions: Each request is independent.
   - HTTP methods:
     GET → fetch data (services/bookings/providers)
     POST → create new entries (add service, book a service)
     PATCH → update existing data (e.g., update booking status)
     DELETE (optional) → remove data (future scope)
3. **MongoDB (NoSQL Database)**
   MongoDB stores data as documents (JSON-like format) inside collections.
   Used collections: services, bookings, providers.
   Features: Schema-less structure, optional Mongoose ODM.
4. **Modular Architecture**
   Folder structure:
   - routes/: API route definitions
   - controllers/: Business logic
   - models/: MongoDB schemas
   - config/db.js: MongoDB connection logic
5. **Database Connection Handling**
   Async connection to MongoDB with reusable instance setup.
6. **Validation & Error Handling**
   Input validation (e.g., required fields), async error catching, user-friendly frontend messages.
7. **Cross-Origin Resource Sharing (CORS)**
   Enables frontend-backend communication using cors middleware.
8. **Port & Server Initialization**
   Backend listens on a specific port using app.listen and logs success messages.
9. **Security (Basic Level)**
   Simple admin passcode check. No advanced authentication implemented yet.
10. **Scalability Considerations**
    Modular code enables future enhancements like user auth, service categories, deployment to production-grade infrastructure.

Database [Bookings]


```
_id: ObjectId('6857ea9c304b8c872157715e')
customerName : "Priya Anand"
phone : "9123456780"
serviceId : ObjectId('6857e86b304b8c872157714f')
date : "2025-06-25"
time : "03:30"
status : "rejected"
```

```
_id: ObjectId('6857eb7b4323e715f227d5a2')
customerName : "Shree"
serviceId : ObjectId('6857e7f4304b8c872157714d')
date : "2025-06-29"
time : "17:10"
status : "approved"
__v : 0
phone : "9365421082"
```

```
_id: ObjectId('6857eba04323e715f227d5a8')
customerName : "Tom"
serviceId : ObjectId('6857e86b304b8c872157714f')
date : "2025-07-02"
time : "12:30"
status : "pending"
__v : 0
phone : "9789568684"
```


2025-07-02

Database [Providers]

```
_id: ObjectId('6857c52f304b8c8721577145')
name : "Ravi Kumar"
service : "Electrician"
phone : "9876543210"
email : "ravi@example.com"
startTime : "09:00"
endTime : "17:00"
```

```
_id: ObjectId('6857e905304b8c8721577152')
name : "Arun Das"
service : "Plumbing"
phone : "9123456789"
email : "arun.plumber@example.com"
startTime : "08:30"
endTime : "17:30"
```

```
_id: ObjectId('6857e917304b8c8721577154')
name : "Meena Raj"
service : "Plumbing"
phone : "9001234567"
email : "meena.plumbwork@example.com"
startTime : "10:00"
endTime : "18:00"
```

Database [Services]

```
_id: ObjectId('6857e740304b8c8721577149')
name : "Electrician"
description : "Expert electrical repair and installation services."
image : "/images/electrician.jpg"
```

```
_id: ObjectId('6857e78c304b8c872157714b')
name : "Plumbing"
description : "Professional plumbing and pipe fixing."
image : "/images/plumber.jpg"
```

```
_id: ObjectId('6857e7f4304b8c872157714d')
name : "Cleaning"
description : "Residential and commercial cleaning solutions."
image : "/images/cleaning.jpg"
```

```
_id: ObjectId('6857e86b304b8c872157714f')
name : "AC Repair"
description : "Air conditioner service and maintenance."
image : "/images/ac.jpg"
```