


FRONTEND (React Application)

- Role: Provides interactive UI for hotel bookings using React (SPA).**
- Handles user authentication, booking management, and backend API health.**

Layer	Description	Key Files
UI Components	Visual elements and booking forms for user interaction	BookingList.js, BookingForm.js, BookingDetails.js, Auth.js
State Management	Uses React Hooks (useState, useEffect) to handle data, authentication, and UI state	App.js
API Communication	Axios service layer for backend API calls	src/services/api.js
Routing / View Handling	Controlled by currentView state to switch between List, Details, Create, and Auth screens	App.js
Authentication Logic	Handles token in localStorage, shows  Authenticated badge	App.js, Auth.js
Health Monitoring	Checks backend API status ("□ Online"/"● Offline")	App.js, healthAPI.ping()

REST API Calls (Axios)



Backend Layer Table (NODE.JS + EXPRESS)

- Handles business logic via **Controllers**
- | • Maps API endpoints via **Routes**
- Middleware for auth, validation, and error handling

Layer	Description	Key Files
Controllers	Handle business logic and process HTTP requests	authController.js, bookingController.js, healthController.js
Routes	Map API endpoints to controller functions	authRoutes.js, bookingRoutes.js, healthRoutes.js
Middleware	Handle authentication, validation, error handling	authMiddleware.js, validationMiddleware.js
Models	Mongoose schemas representing database collections	Booking.js
Config / Database	Setup database connection, environment variables	config/database.js
Server	Main entry point, starts Express server	server.js

Mongoose Connection



MongoDB Database

- Users Collection: stores user accounts
- Bookings Collection: stores booking info

Layer	Description	Key Files
Users	Stores registered users for authentication	username, password (hashed), role
Bookings	Stores hotel booking information	userId, hotelName, checkInDate, checkOutDate, guests, status