

Of course, here is a breakdown of all the pages, their required functionalities, and the specific technologies that will be used to build them.

1. Customer-Facing Portal

This is the main website that your guests will see and use.

Pages & Functionalities:

- **Homepage**
 - **Functionality:** Display dynamic lists of trending destinations and hotels.
 - **Front-End Tech:** **React** will build the UI components (like hotel cards), and **Tailwind CSS** will style the page. Data will be fetched from the backend to make the lists dynamic.
 - **Back-End Tech:** **Java/Spring Boot** will create an API endpoint (e.g., `/api/hotels/trending`) that provides the list of hotels to the front-end.
- **Login & Signup Page**
 - **Functionality:** Allow users to create a new account and log in securely, including OTP (One-Time Password) logic.
 - **Front-End Tech:** **React** will be used to create the interactive forms.
 - **Back-End Tech:** **Java/Spring Boot** will handle the security, process user details, and manage login/signup requests via the `/api/auth` endpoint.
- **Hotel & Room Details Page**
 - **Functionality:** Show details for a selected hotel, check room availability for specific dates, and allow users to select a room.
 - **Front-End Tech:** **React** will display all hotel and room information dynamically. An interactive calendar or date-picker component will be used for checking availability.
 - **Back-End Tech:** **Java/Spring Boot** will provide API endpoints like `/api/hotels/{hotelId}` and `/api/rooms/hotel/{hotelId}` to send hotel and room data to the front-end.
- **Booking Page**
 - **Functionality:** Finalize the booking process by confirming dates and user details to create a booking.
 - **Front-End Tech:** **React** forms will capture all necessary information for the booking.
 - **Back-End Tech:** **Java/Spring Boot** will have an endpoint like `/api/bookings` to process the request and save the new booking to the **MySQL** database.
- **Community Blog Page**
 - **Functionality:** Display blog posts from other travelers and provide a form for verified guests to submit their own travel stories.
 - **Front-End Tech:** **React** will be used to show the posts and the submission form.
 - **Back-End Tech:** **Java/Spring Boot** will manage fetching and submitting posts through the `/api/blog` endpoint.

2. Hotel Owner Portal

This is the private dashboard for hotel managers.

Page & Functionalities:

- **Hotel Management Dashboard**
 - **Functionality:**
 - Display charts for revenue and occupancy.
 - Manage room status (e.g., available, cleaning).
 - View a list of all current and past bookings.
 - Read and manage guest reviews.
 - Manage property, employees, and billing/POS integration.
 - **Front-End Tech: React** will be used to build the dashboard. The UI will be styled with **Bootstrap** (from the admin template). **Recharts** or **Chart.js** will be used to create the data visualization charts.
 - **Back-End Tech: Java/Spring Boot** will provide various API endpoints to supply all the necessary data for bookings, revenue, and room statuses.

3. Admin Portal

This is the master control panel for the platform administrators.

Page & Functionalities:

- **Admin Dashboard**
 - **Functionality:**
 - Manage all users on the platform.
 - View and approve or reject new hotel registrations.
 - Manage and resolve user complaints.
 - Review and approve pending blog posts from the community blog.
 - **Front-End Tech: React** and a **Bootstrap**-based admin template will be used to create a professional and functional UI.
 - **Back-End Tech: Java/Spring Boot** will create secure `/api/admin` endpoints that provide the necessary functions, such as fetching pending hotels or approving blog posts.