

Assignment_ID: assignment_category_0004

Project Overview [video](#)

We are seeking passionate and talented individuals to join our team and contribute to the development of the Hotel Booking Web Application. As a team member, you will play a crucial role in shaping the future of hotel accommodation booking, enhancing user experiences, and ensuring the success of this innovative platform.

🚩: 0 [If we have any update we will mention it here]

Main Features

Home Page Design (implement any of 4 sections)

1. **Stunning Imagery:** Use high-quality images and videos of hotel rooms and amenities to create an enticing visual experience for users. Consider using parallax scrolling for added depth.
2. **Map:** Include a map that displays the hotel's location, nearby attractions, and points of interest.
3. **Special Offers and Promotions:** Showcase special offers, discounts, and promotions prominently on the home page. Use eye-catching banners and call-to-action buttons.
4. **User Testimonials:** Display authentic user testimonials and ratings to build trust. Consider using a testimonial carousel for variety.
5. **Newsletter Signup:** Encourage users to subscribe to your newsletter for updates, deals, and exclusive offers.
6. **Featured Rooms:** Highlight a selection of featured rooms or suites with appealing visuals and descriptions. Use a "Book Now" button for quick access.

User Authentication

- Users can create an account with email and password.
- Users can log in using Google authentication via Firebase.

JWT Authentication

- Upon login, you will create a JWT token and store it on the client side. You will send the token with the call and verify the user. Implementing 401 and 403 is optional. Ensure you have implemented the JWT token create a token and store it on the client side for both email/password-based authentication and social login. You must implement JWT on your private routes.

Navigation Bar

- The navigation bar should have links to the "Rooms" page and "My Bookings" page.
- The "My Bookings" page should be accessible only to authenticated users.

Rooms Page

- Display a list of available rooms.
- There will be an image gallery with pictures of rooms. After clicking an individual picture, (it will redirect the user to the room details page).
- Implement a filter system to filter rooms by price range. (filter can be implemented on the server side or client side)
- Users can view room review counts.
- Users can post reviews for rooms after booking. [See details](#)

Room Details Page

- Room Description
- Price per Night
- Room Size
- Availability
- Room Images
- Special Offers (if available)
- There will be a **Book Now**. On clicking the Book Now page user can book a room. Make sure the room will be **unavailable** when someone books the room.
- Reviews (if available else show any meaningful message) [to know how users will add reviews, follow [this](#)]
- Users can select a room and choose the booking duration (single day) using a date picker.
- The system should ensure that users can only book available rooms. Initially, there will be some available seats. On booking the available seats will be decremented.
- Users can view a room summary of their booking before confirming. [room summary means what they choose like price, date, room description, etc.]

Booking Cancellation

- Users can cancel a booking before 1 day from the booking day. Suppose the user booked a room on 9/11/2023. if he wants to cancel the booking then he can cancel within 7/11/2023.
- Users should have the option to view a list of their current bookings on the **My Bookings** page.
- Next to each booking, provide a **Delete** button.
- Upon clicking the **Delete** button, prompt the user to confirm the deletion with a confirmation modal.
- If the user confirms, remove the booking from the database and update the user's booking list.

Hints: you can use [momentjs](#) to compare the time.

Update Booking Date

- Users should be able to view a list of their current bookings on the "My Bookings" page.
- Next to each booking, provide an "Update Date" button.
- Clicking the "Update Date" button should allow the user to select a new date for the booking, similar to the process for updating bookings in the "Rooms" section.
- Upon successful modification, update the booking details in the database and notify the user of the update using modal or toast.

Review System

- Users can post reviews for rooms only they have booked.
- Reviews should include a username, rating, comment, and timestamp.
- Reviews should be shown on the room details page for other users.

Access Control

- Users who are not logged in cannot book a room. If the user clicks the book button, they will be redirected to the login page.
- Users who are not logged in can only view basic room details but cannot post reviews.

404 Page

- Create a 404 page with an exciting jpg/gif and a "Back to home" button that redirects the user to the home page.

Toast

- For all CRUD operations Show relevant toast/sweetalert/anything with a meaningful message.

Package

- Implement any 2 packages from given below. Don't use any 3rd party package for them)
 - Toast (react-hot-toast, sweet-alert, react-toastify)
 - Update the browser tab title, meta-data ([Helmet](#))
 - Tab ([react-tabs](#))

Bonus Requirements 🌟

Commits & Readme 📝

- Minimum 20 meaningful git commits on the client-side repository.
- Minimum 10 meaningful commits on the server-side repository.
- Create a readme for the client side and write about your project (at least 5 bullet points). Remember to add your client-side live link to your website here.

[implement a feature and then commit them]

Reload 🔄

- If you reload the protected/private routes (after login), this page will not redirect the user to the login page. Instead, it will keep the logged-in user on the protected route.

Responsive Design 📱💻

- Make the homepage of your website mobile, tablet & desktop responsive.

Environment Variables 🔒

- Use the Environment variable to hide the Firebase config keys and MongoDB credentials.

Animation 🎉

- Use animation in the home page sections.

Optional

Extra Pages Recommendations

1. **About Us:** Create an "About Us" page that tells the story of your hotel, its history, mission, and values. Include information about the team and management.
2. **Contact Us:** Develop a "Contact Us" page with a contact form, phone numbers, email addresses, and a physical address. Ensure users can reach out easily.
3. **Events and Activities:** Provide information about upcoming events, activities, and local attractions near the hotel. Include a calendar and booking options for these events.
4. **Gallery:** Create a gallery page with high-resolution images of the hotel, rooms, amenities, and events. Users can explore and visualize their stay.
5. **FAQ:** Compile a comprehensive FAQ page that addresses common questions and concerns users may have. Make it easy for users to find answers.
6. **Privacy Policy and Terms of Service:** Add legal pages with privacy policy, terms of service, and other legal disclaimers to ensure compliance.
7. **Testimonials:** Dedicate a page solely for user testimonials. This builds trust and provides potential guests with authentic reviews.
8. **Local Area Guide:** Offer an informative guide to the local area, including restaurants, attractions, and things to do.
9. **Career Opportunities:** If applicable, create a page for job listings and career opportunities at your hotel.

Remember to maintain a consistent design and user experience throughout your website. Tailor these additional features and recommendations to suit the unique characteristics and goals of your hotel booking platform.

Guidelines

1. Save time on the website idea. Just spend 15-20 minutes deciding, find a sample website, and start working on it.
2. Do not waste much time finding the right image. You can always start with a simple idea. Make the website and then add different images.
3. Don't look at the overall task list. Just take one task at a time and do it. Once it's done, pick the next task. If you get stuck on a particular task, move on to the next task.
4. Stay calm, think before coding, and work sequentially. You will make it.
5. Be strategic about the electricity issue.
6. Use ChatGPT to generate JSON data. You can use ChatGPT for other purposes as well.
7. Initially, you can add data to the MongoDB. Then you can use the data from the fetch call. Other collections will depend on you.

No Pain, No Gain

"The most beautiful moments in life come after going through hardships and challenges."