

QuickStay: Hotel Booking Application

Abstract:

The emergence of online booking systems for hotels, lodges, motels, and resorts has completely changed the hospitality booking situation. Innovative and user-centric solutions are still needed in the industry, even with the presence of well-established businesses. By creating a booking website with the Next.js 13, App Router, React, Tailwind CSS, Prisma, MongoDB, NextAuth this project seeks to close this gap by addressing major customer pain points. The main goal of this project is to develop a dependable and user-friendly website for resort/hotel reservations. This website will enable service providers to manage reservations more effectively and will provide users with seamless booking experiences. With the help of several modern technologies and an in-depth analysis of Next.js 13 App Router, this project creates a highly functional and aesthetically pleasing online application. The fundamental framework is Next.js, which works in unison with React to provide effective client-side rendering and dynamic content creation. With the use of Tailwind CSS, styling becomes more simplified, facilitating quick prototyping and simple tweaking of the application's visual elements. Furthermore, the project's scalability and data management capabilities are improved using Prisma and MongoDB as the backend database options. MongoDB provides flexibility and scalability for managing a variety of data formats, while Prisma's user-friendly ORM layer streamlines database interactions. In addition, the use of NextAuth enhances the application's authentication and authorization features, guaranteeing safe access control and user administration. By utilizing NextAuth's smooth authentication processes and versatility in supporting many authentication providers, the project attains a safe and seamless user experience—a crucial aspect of contemporary web applications. Through the integration of these technologies, the project demonstrates Next.js's synergistic potential as a flexible and strong framework for creating complex online applications. The project demonstrates how Next.js, and its ecosystem can handle the changing needs of web development in the current digital context by providing secure registration, efficient data management, dynamic content generation, and seamless routing. User login, extensive service listings based on different categories available, availability management, reservations, thorough information, and images are some of the key features.

Keywords: Next.js, Tailwind CSS, React, MongoDB, Prisma, NextAuth

1. Introduction

In the dynamic field of web development, creating dynamic and user-focused applications requires incorporating cutting-edge technologies. In order to create a reliable and feature-rich web application, this report summarizes a thorough investigation and deployment of cutting-edge technologies, such as Next.js 13 App Router, React, Tailwind CSS, Prisma, MongoDB, and NextAuth. At the forefront of this project is the Next.js 13 App Router, which provides the fundamental architecture for smooth navigation and dynamic content rendering. When combined with React, Next.js gives developers the ability to design responsive and interactive user interfaces that make for a seamless and enjoyable user experience. Tailwind CSS is a styling technique that prioritizes functionality over design, allowing for quick prototyping and adaptable customization of the application's visual components in harmony with the frontend architecture. Through this integration, the design process is expedited, and application-wide consistency and scalability are guaranteed. Prisma and MongoDB, which combine the strength of an easy-to-use ORM layer with the adaptability of a document-based database, support the backend architecture. While MongoDB has strong data storage and retrieval capabilities which are necessary for managing complex data structures in a scalable manner, Prisma streamlines database interfaces and schema management. Moreover, NextAuth gives the application an additional layer of authorization and authentication, guaranteeing safe access control and smooth user authentication across a range of authentication providers. This integration improves the security posture of the application and offers a seamless authentication process for users. In order to develop a full web application, we explore the implementation details, difficulties, and advantages of combining different technologies in this report. Using Next.js, React, Tailwind CSS, Prisma, MongoDB, and NextAuth, I hope to demonstrate how these technologies can propel innovation and provide outstanding user experiences in the online development space.

2. Problem Statement

Hotel management in this industry has many obstacles to overcome to effectively manage properties and provide flawless visitor experiences. Conventional systems frequently lack the adaptability and user-centered design needed to satisfy contemporary needs. The rise of websites such as Trivago, Booking.com, Agoda, Expedia, Kayak highlights the demand for all-inclusive, easy-to-use hotel management systems. Still have technical drawbacks such as user interface issues, slow loading times, limited search options, mobile responsiveness problems. Therefore, the main goal of the project is to create a strong hotel management website while addressing several important issues. They include making sure reservation systems are dependable to avoid multiple bookings, empowering property management with diverse listing options and pricing structures, maximizing the user experience through user-friendly interfaces. The website seeks to rethink hotel management by tackling these issues, improving visitor satisfaction and operational efficiency in the dynamic hospitality industry.

2.1 Project Goals

Creation of a User-Centric Interface: The main objective is to create and execute a user-centric interface that improves usability and encourages a simple and intuitive experience for both hotel staff and customers.

Effective Property administration: Provide various pricing options, thorough amenity descriptions, and strong property administration tools that let administrators add, alter update property listings quickly and effectively .

Easy to use reservation system: Enable guests to effortlessly browse for available properties, examine detailed information, and make reservations with ease.

Scalability and Performance Optimization: Build the system with scalability in mind, making sure it can accommodate growing traffic and data volumes without experiencing performance issues. Then, put optimization techniques into place to keep the system responsive under fluctuating loads.

Flexibility and Customization: To improve user satisfaction and meet a range of needs, give administrators the ability to customize the platform to meet their unique requirements. Examples of these customization possibilities include property classification, branding, and reporting functions.

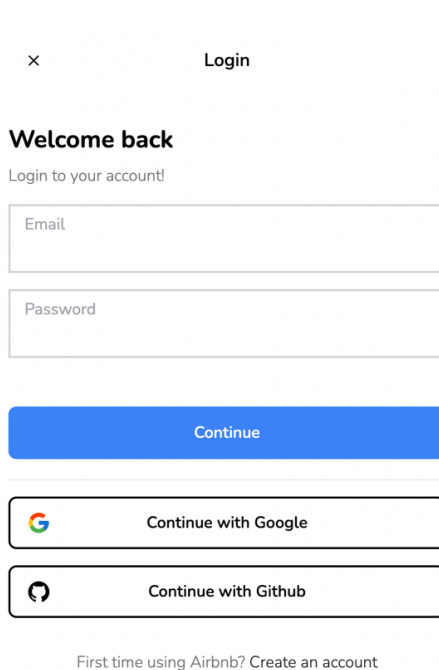
3. Methodology

The methodology of the project is divided into the following phrases:

Phase 1: Defining requirements

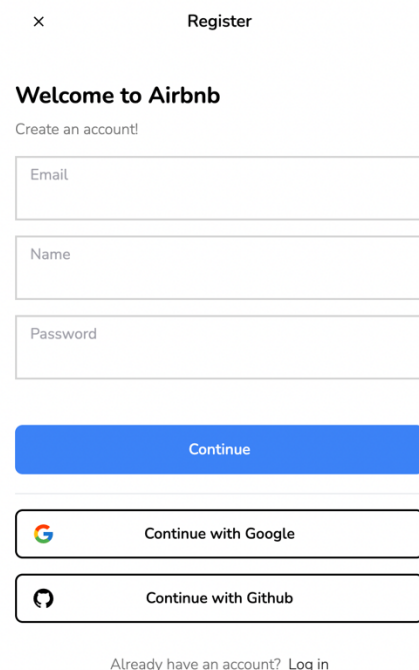
Authentication and User Registration:

- Users should be able to safely create accounts by entering the required data, which may include a password, email address.
- It should be possible for users to securely and quickly log in and logout with their credentials.
- Users can also login using google or GitHub credentials.



The login form is titled "Login" and features a "Welcome back" message. It prompts the user to "Login to your account!" and includes input fields for "Email" and "Password". A blue "Continue" button is positioned below the fields. At the bottom, there are two social login options: "Continue with Google" and "Continue with GitHub". A link at the bottom reads "First time using Airbnb? Create an account".

Fig 1 : User login



The registration form is titled "Register" and features a "Welcome to Airbnb" message. It prompts the user to "Create an account!" and includes input fields for "Email", "Name", and "Password". A blue "Continue" button is positioned below the fields. At the bottom, there are two social login options: "Continue with Google" and "Continue with GitHub". A link at the bottom reads "Already have an account? Log in".

Fig 2: User registration

Search and filtering

- The website has an extensive search function that enables customers to look for lodging using a range of parameters, such as:
 - **Location:** It is possible for users to look for places in their desired countries
 - **Dates:** To identify available options, users should be able to select the dates of their choice for check-in and check-out.
 - **Price range:** It should be possible for users to select a budget or to filter lodging options based on price.
- Each entry should have pertinent information and photos shown, and the search results are presented in an easy-to-read and well-organized format.

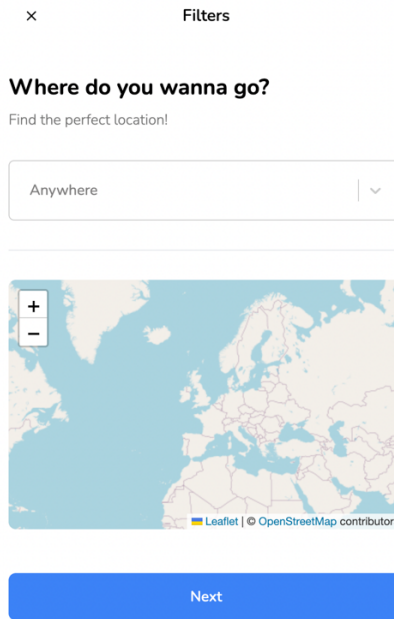


Fig 3: Searching based on location

Reservation Administration:

- Users have access to comprehensive details regarding every lodging listing, such as: a thorough explanation of the property's characteristics, facilities, and neighboring points of interest.
- **Availability:** Customers can check the availability of rooms on the dates they have chosen.
- **Cost:** Users can examine the whole cost of their bookings before confirming them, and the website clearly discloses the price information

Hosted by **mithila nigudkar** 

5 guests 5 rooms 5 bathrooms



Modern

This property is modern!

A stay at Millennium Hotel New York Downtown places you in the heart of the Financial District with unobstructed panoramas of major New York landmarks. Plenty of attractions are just nearby: ✓Amazing shows at Broadway Theatre ✓Tours at the Statue of Liberty ✓Stunning 360-degree views of New York at the Empire State Building ✓Climb 102 floors in 47 seconds to One World Observatory ✓Commemoration, exhibitions, and educational programs, the National September 11 Memorial & Museum

Fig 4: Description of the property

\$ 2000 night

◀ April 2024 ▶

Sun	Mon	Tue	Wed	Thu	Fri	Sat
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4

Reserve

Total \$ 6000

Fig 5: Pricing and dates information

Reviews and ratings

- Consumers can rank and review the lodgings they've stayed in, including the following options: Overall rating: Consumers can rate their experience on a scale of one to five stars.
- Users post thorough evaluations of their experiences that highlight both their positive and negative aspects.
- When perusing the lodging listings on the website, guests need to have the ability to observe and consider these evaluations and reviews, which will aid them in making well-informed reservations

Reviews

mithila nigudkar

1 minute ago

Literally one of the best place I have stayed in! Great for price

Rating: ★★★★★

Leave a review

Your review

Rating

5 ▾

Submit Review

Fig 6 : Reviews and ratings

Phase 2: Designing the database

- By determining the primary entities that are pertinent to the hotel management system. These entities usually consist of properties, users , reservations/Bookings, reviews and ratings, user profiles, property categories. Determine the dependencies and associations between these items by analyzing their relationships.
- Determining the characteristics that best describe each entity. For instance:
Users: Name, email address, password
- **Location:** include location name, description, cost, and availability.
Reservations/Bookings: Total Price, Check-in and Check-out Dates, Number of Guests
Evaluations/Ratings: Review and Submission Date
- Based on their dependencies and relationships, determining the links between the entities. Typical kinds of relationships are:
One-to-one Every record in one entity is linked to precisely one record in another.
One-to-many : One record in one entity is linked to several records in another entity
Many-to-Many: A number of records in one entity are linked to a number of records in another.
- Applying normalization techniques to remove redundancy in the database

Phase 3: The backend

Select the Backend Framework: Next.js

- Next.js is a great option for developing the backend of a React-based application since it offers server-side rendering (SSR) and API routes. It provides smooth interaction with React components and facilitates effective routing and data fetching.
- The server is started by using : npm run dev

Architecture of an application.

- Using Next.js as the backend and React components as the frontend and by implementing a monolithic architecture. This enables future scaling, if necessary, while also simplifying development and deployment.
- To define backend endpoints for data processing, database interactions, and retrieval, leverage Next.js API routes.
- Tailwind CSS is used for styling the webpage
- To install the tailwind CSS: -npm install -D tailwind postcss autoprefixer
- Npx tailwindcss init -p

Designing data storage for MongoDB and prisma

- For versatile data storage, MongoDB NoSQL database is used. It's especially good for storing unstructured data like user profiles, property details and reviews
- Prisma is an ORM (Object-Relational Mapping) library that can be used to streamline database operations such as data validation, querying, and schema building.
- Model data entities and relationships in MongoDB schemas by defining them using Prisma's schema definition language (SDL).

Commands used:

- ***npm install world-countries***: this helps in installing the world-countries package from npm, providing information about the countries worldwide.
- ***npm install react-select***: this installs the react-select package which is a component for the react applications.
- ***npm install leaflet***: this is a common open-source JavaScript library used for interactive maps
- ***npm install -D @types/leaflet*** : this installs typescript type definitions for the leaflet package
- ***npm install react-leaflet***: this provides a react wrapper for the leaflet library , enabling the integration of leaflet maps with react components.
- ***npm install next-cloudinary***: used for media management
- ***npm install date-fns***: used for dates, it is a JavaScript library
- ***npm install -D @types/react-date-range***: Installs the react-date-range package's TypeScript type definitions as a development dependency.
- ***npm install react-date-range***: Installs the react-date-range package, which gives react applications access to a variety of adaptable and configurable date range picker components.

- ***npx prisma db push***: applies database schema modifications and migrates the database to conform to the most recent Prisma schema specified in the project by running the Prisma CLI command.

Phase 4: Media management using Cloudinary

A whole range of tools and services are available for organizing, enhancing, and distributing photos, videos, and other media assets on the web and in mobile applications through Cloudinary, a cloud-based media management platform. By providing developers with a centralized method for distributing, manipulating, and storing media files, it lessens the overhead and complexity of managing media assets in their applications. Cloudinary is frequently utilized in image management projects for the following reasons:

Media Storage and Organization:

- Developers may upload and store photographs and other data in one central spot with Cloudinary's scalable and secure cloud storage for media assets. With its powerful organizing features—folders, tags, and metadata, for example—it makes managing and retrieving media files easy.

Image Optimization and Transformation:

- With Cloudinary's extensive array of potent image transformation features, developers may quickly and easily resize, crop, rotate, and otherwise work with images. To guarantee optimal performance and a positive user experience across devices and screen sizes, it also offers capabilities for optimizing image quality, format conversion, and responsive image delivery.

Content Delivery Network (CDN) Integration:

- Cloudinary's smooth integration with international CDNs guarantees the quick and dependable distribution of media assets to end users around the globe. Through media file caching on edge servers dispersed throughout several regions, Cloudinary lowers latency and speeds up online and mobile application page loads.

Phase 5: The front-end

Describe the User Interface needs:

- Determine the intended style and look of the website by gathering needs from customers.
- Determine the essential navigational processes, visual components, and user interactions needed to provide a flawless user experience.

Selecting the front-end technologies

- Based on project requirements, familiarity, and scalability considerations, select the right front-end technology.
- Use Tailwind CSS for styling and layout, react for creating interactive UI components, and Next.js as the frontend framework for this project.

Responsiveness

- Create layouts that are responsive to different screen sizes and devices so that users of PCs, tablets, and smartphones will always have the same experience.
- Make use of CSS media queries and responsive design concepts to modify layout and styling according to viewport dimensions.

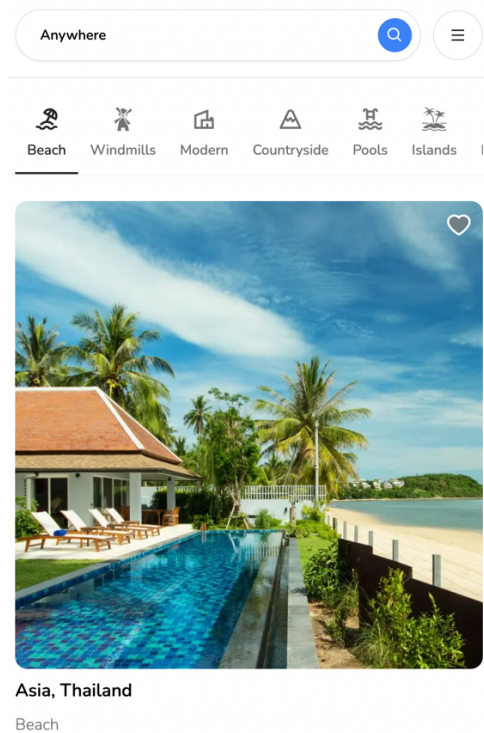


Fig 9: mobile responsiveness

Implementing UI components

- With React, create reusable user interface components that are arranged into a modular component architecture for scalability and easy maintenance.
- Include typical user interface components like search bars, booking forms, property listings, ratings, and reviews in your navigation menus.

Integrating the front-end with the backend API's

- To get and show dynamic data, including user profiles, bookings, reviews, and property listings, integrate frontend components with backend APIs.
- For data retrieval and processing, make use of external Next.js API routes, making sure to properly handle errors and validate the data.

Cross browser compatibility

- To guarantee compatibility and consistent rendering, test the website on several browsers (such as Chrome, Firefox, and Safari) and devices (such as desktop, tablet, and smartphone).
- To find and fix any layout or functionality problems, use responsive design testing frameworks and browser testing tools.

Limitations

Real-Time Updates:

- Features like live availability updates and immediate messaging between users are hampered by the current implementation's lack of real-time connectivity between clients and servers.

Scalability:

- Although the project has been developed to fulfill the existing requirements, as the user base increases, its scalability may become constrained. Upgrades to the infrastructure and additional optimization may be necessary to provide optimal performance and scalability under high loads.

Localization:

- The project's attractiveness to a worldwide audience may be limited if it does not provide multi-language or multi-region capability out of the box. More development work would be needed to add localization functionality.

Accessibility:

- The project may not adhere to all online accessibility guidelines, which could prevent people with impairments from using the website. In order to guarantee inclusivity, accessibility features would need to be improved.

Security concerns:

- Despite the implementation of fundamental security measures, there is a possibility that the project may be subject to different types of security attacks, including data breaches, SQL injection, and cross-site scripting (XSS). To reduce these threats, regular security assessments and updates are necessary.

Future scope of the project:

Real-Time Communication:

- To provide real-time updates for features like live availability, chat messaging, and notifications, integrate WebSocket technology or a real-time data streaming platform like Pusher or Socket.io.

Mobile Application:

- To increase the hotel management platform's accessibility and give users on-the-go access, create a companion mobile application for the iOS and Android operating systems.

Personalization and Recommendation:

- To improve the user experience and boost user engagement, apply personalized recommendations based on user preferences, previous reservations, and browsing history.

Advanced Search and Filtering:

- To improve search efficiency, include geolocation-based suggestions, predictive search ideas, and advanced filtering choices. Users locate accommodations more quickly and appropriately.

Social Media Integration:

- Use social media integration to boost user engagement and viral marketing by allowing users to connect with friends and family, share their experiences, and sign in using their accounts.

Integration with Third-Party Services:

- To increase the platform's value proposition and broaden its variety of offers, consider forming alliances and integrating with third-party services like payment processors, booking aggregators, and trip planning tools.

Multi-Language Support:

- Allow users to access the website in their favorite language and improve accessibility for non-English speakers by implementing multi-language support to cater to a diverse global audience.

Conclusion

In summary, the Next.js, Tailwind CSS, React, MongoDB, Prisma, and NextAuth-developed hotel management website project offers a complete solution for travelers looking for lodging and property owners managing their listings. The project identifies areas for development and future expansion even as it highlights strong features including user registration, property search, booking administration, and review submissions. The limits of the project, such as the lack of real-time updates and scaling issues, highlight areas that should be improved. Through the incorporation of real-time communication technologies and the resolution of scalability issues, the platform may enhance its functionality and offer consumers a smooth and captivating experience. The project's future scope is wide-ranging and includes anything from adding more multilingual support and creating a companion mobile application to integrating personalized suggestions and sophisticated search features. The measures are consistent with the project's overall objective of augmenting user contentment, cultivating involvement, and enabling expansion within a dynamic hospitality domain. Essentially, even while the concept for a hotel management website provides a strong base, its real potential is found in its capacity to adjust and react to user input and market conditions. With a dedication to innovation and constant improvement, the project is well-positioned to become a market leader in the field of online hotel reservations, meeting the various demands and tastes of both guests and property owners.