

Lab Project Report on Tourist Database Management System

Submitted by

Aadarsh Lakshmi Narasiman - 211CS201

Praveen Kumar - 211CS238

Rishi Diwaker - 211CS243

Sanchit Sudhakar Sakhare - 211CS251

Submitted to-

Prof.Chetan Naik

Department of Computer Science and Engineering

National Institute of Technology Karnataka,Surathkal

COURSE-

Database Systems Lab - CS254



Introduction

The proposed project is a tourism hotel database management system that aims to provide an efficient and personalized hotel selection experience to users. The system allows users to log in and input their preferences such as location, amenities, and room types to search for hotels. The system also enables users to view user reviews and create a wish list of their favorite hotels. The project's objective is to develop a user-friendly system that can efficiently manage hotel information and provide a personalized hotel selection experience to users. The system's successful implementation can benefit the tourism industry by enhancing the user experience and improving hotel management efficiency.

Literature Survey

Existing websites for tourists, such as Airbnb, TripAdvisor, and Goibibo, offer a variety of features for finding and booking travel accommodations. However, many of these websites lack a wishlist feature, which can be a useful tool for users to save and compare different options for their trip. Additionally, some websites prioritize hotels that pay extra money to them for their promotion, potentially limiting the range of options available to users. Furthermore, excessive advertisements can be frustrating for users and detract from the overall user experience.

To address these issues, our project aims to develop a tourist management system that includes a wishlist feature, allows for unbiased search results, and minimizes advertisements. Our system will enable users to save and compare hotels and activities, and provide recommendations based on user preferences rather than paid placements. By providing a user-friendly and transparent platform, our system aims to enhance the overall travel planning experience for users.

Problem Statement

This project aims at providing different hotels and location suggestions based on the location for tourists. Further, compared to conventional websites, we have added extra features like wishlists for tourists to add their locations/hotels of choice for referencing to in future.

Objectives

- To develop a user-friendly system that allows users to log in and input their preferences such as location, amenities, and room types to search for hotels.
- To create a comprehensive database that can efficiently manage hotel-related information, including hotel location, room type availability, pricing, and amenities.
- To enable users to view user reviews and ratings to make informed decisions when selecting hotels.
- To develop a wish list feature that enables users to save their favorite hotels for future reference.

Description

The backend of this application has 7 tables to implement its functionality. The tables include-

Table 1- Location Table

The following table provides data on the location of the sightseeing or tourist spot and other relevant information. Here loc_id is the Primary Key.

The following table has these attributes-

location_id,location_name,description,address,city,state,country, zip code.

Table 2- Hotel Table

The following table provides details regarding a particular hotel based on the primary key called hotel_id. Here hotel_id is the Primary Key to the table and location_id is Foreign Key referencing Location table.

The following table has these attributes-
hotel_id , location_id, email , website ,description , star_rating ,
total_rooms.

Table 3 - Room Table

The following table provides details about Room details.

The Primary Key includes room_id and the Foreign Keys are hotel_id and amenity_id referencing Hotel and Amenities Table respectively.

The attributes include -

room_id , hotel_id , room_type, am_id, max_occupancy, price_per_night.

Table 4 - User Table (Tourist)

The following table holds the data for the tourist. The primary key is user_id.

The attributes include -

user_id, first_name , last_name, email_id, password,
phone,city,state,country,zipcode.

Table 5 - Reviews Table

The following table holds information on the tourist user's reviews on the hotel.

The Primary Key is review_id and the Foreign Keys are hotel_id and user_id.

The following attributes are included in the table-
review_id, hotel_id,user_id,rating,title,body

Table 6 - Amenities Table

This table provides information about the amenities available in the rooms.

The Primary Key is am_id and the Foreign Keys are r_id.

It includes the following attributes -
amenity_id,room_id,Names,description

Table 7 - Wishlist

This table holds the information about the places the tourist would like to visit by choosing their hotels.

The Primary Key is wishlist_id and the Foreign Keys are user_id and hostel_id.

It has the following attributes-
wishlist_id, user_id , hostel_id, name and description.

Technical Requirements

To make it look good and easy to use, we'll be using HTML, CSS, and JavaScript. We'll also use JavaScript to make the website more interactive. To keep track of all the information about hotels, rooms, bookings, and users, we'll be using a database called SQLite. Finally, we'll be using a framework called Django to put everything together and make the website work.

Methodology

The process of implementing our project went as follows-

1. First we went through different sources to decide on a problem statement.
2. Based on our research, we decided on the features which further helped us develop our schema diagram.

3. The next step was building the front end pages, based on requirement and flow.
4. Following this the different pages were linked and created as views with the help of Django.
5. After this the models were created followed by migrations to implement these changes.
6. Following this records were created into the tables.
7. The final step was rendering these table record data onto the front end side.

The backend of this application has 7 tables to implement its functionality. The tables include-

Table 1- Location Table

The following table provides data on the location of the sightseeing or tourist spot and other relevant information. Here loc_id is the Primary Key.

The following table has these attributes-

location_id,location_name,description,address,city,state,country, zip code.

Table 2- Hotel Table

The following table provides details regarding a particular hotel based on the primary key called hotel_id. Here hotel_id is the Primary Key to the table and location_id is Foreign Key referencing Location table.

The following table has these attributes-

hotel_id , location_id, email , website ,description , star_rating , total_rooms.

Table 3 - Room Table

The following table provides details about Room details.

The Primary Key includes room_id and the Foreign Keys are hotel_id and amenity_id referencing Hotel and Amenities Table respectively.

The attributes include -

room_id , hotel_id , room_type, am_id, max_occupancy, price_per_night.

Table 4 - User Table (Tourist)

The following table holds the data for the tourist. The primary key is user_id.

The attributes include -

user_id, first_name , last_name, email_id, password, phone,city,state,country,zipcode.

Table 5 - Reviews Table

The following table holds information on the tourist user's reviews on the hotel.

The Primary Key is review_id and the Foreign Keys are hotel_id and user_id.

The following attributes are included in the table-
review_id, hotel_id,user_id,rating,title,body

Table 6 - Amenities Table

This table provides information about the amenities available in the rooms.

The Primary Key is am_id and the Foreign Keys are r_id.

It includes the following attributes -

amenity_id,room_id,Names,description

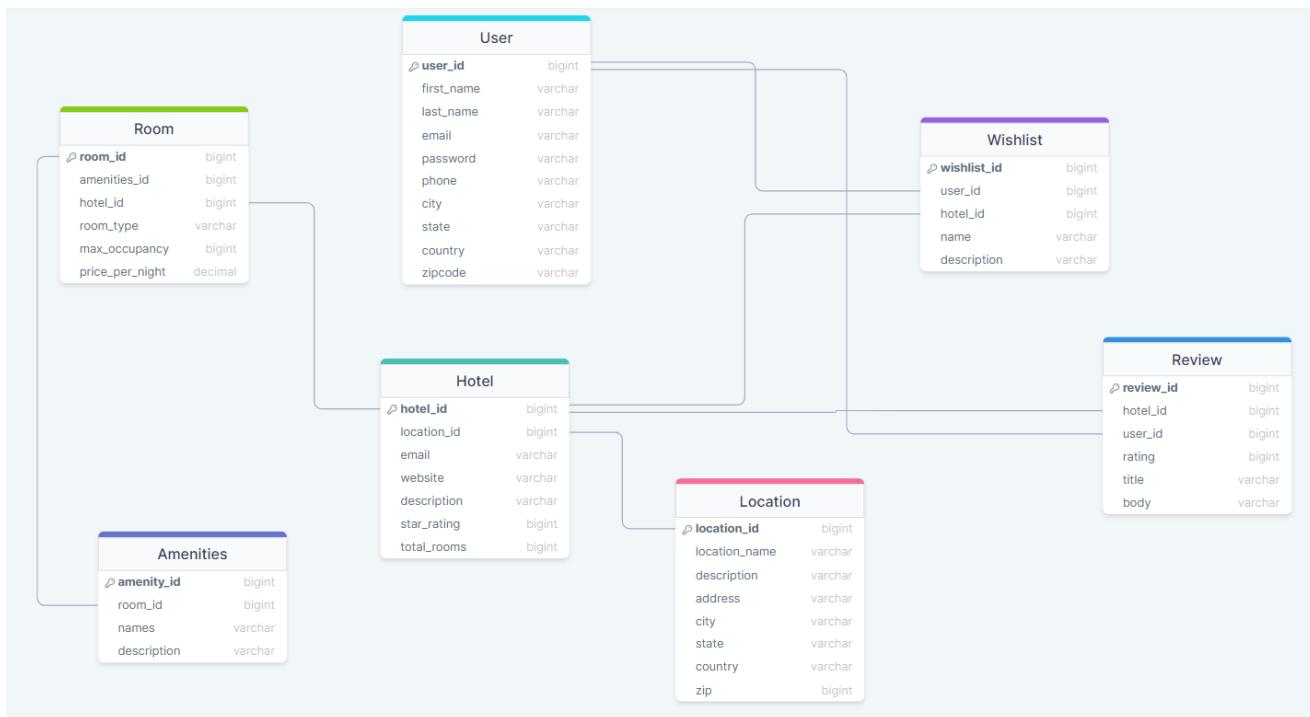
Table 7 - Wishlist

This table holds the information about the places the tourist would like to visit by choosing their hotels.

The Primary Key is wishlist_id and the Foreign Keys are user_id and hostel_id.

It has the following attributes-

wishlist_id , user_id , hostel_id, name and description.

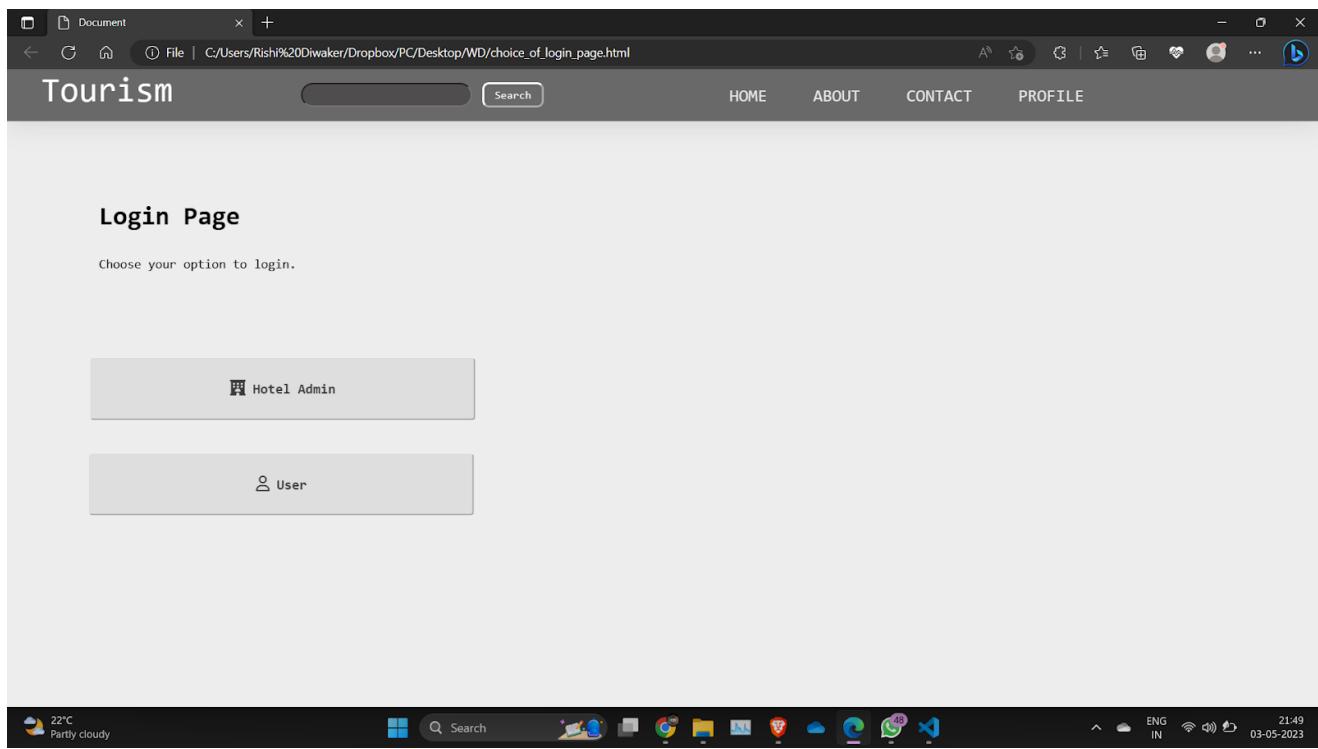


The above diagram is a schema diagram summarizing the same.

Front-end Review

The following pages were created which were later used as templates while developing our websites.

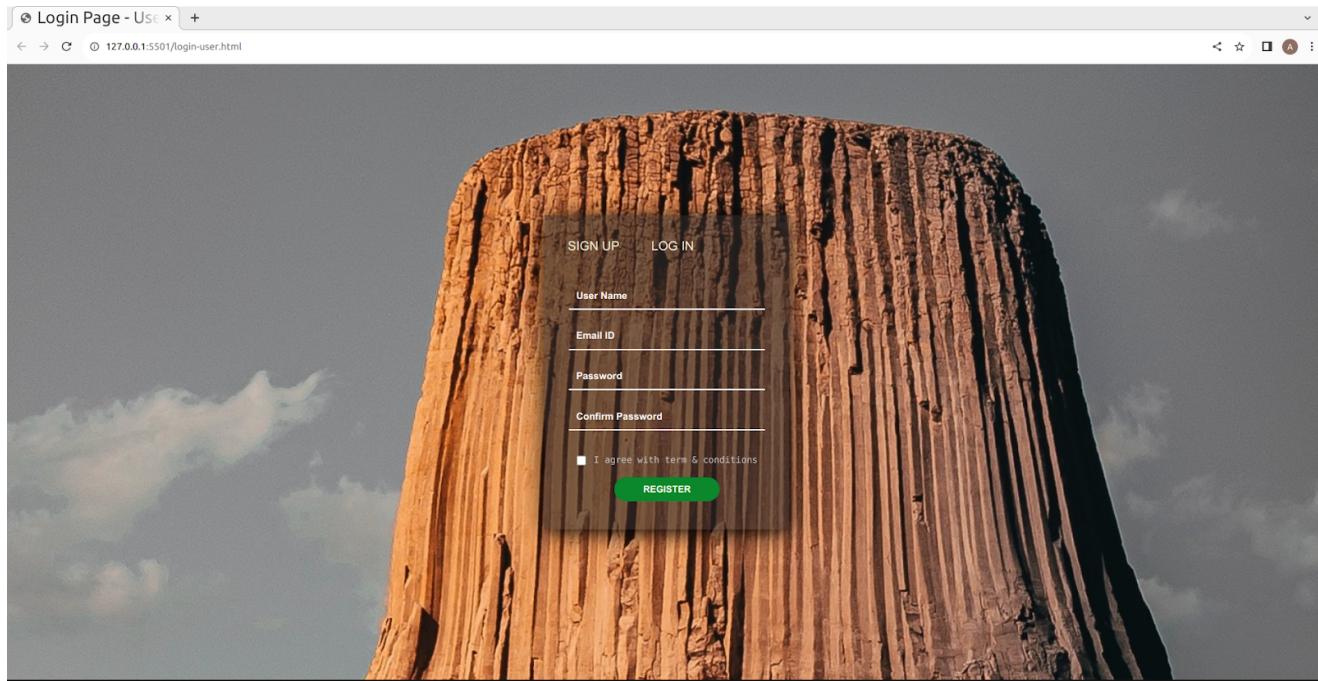
1. Login option page



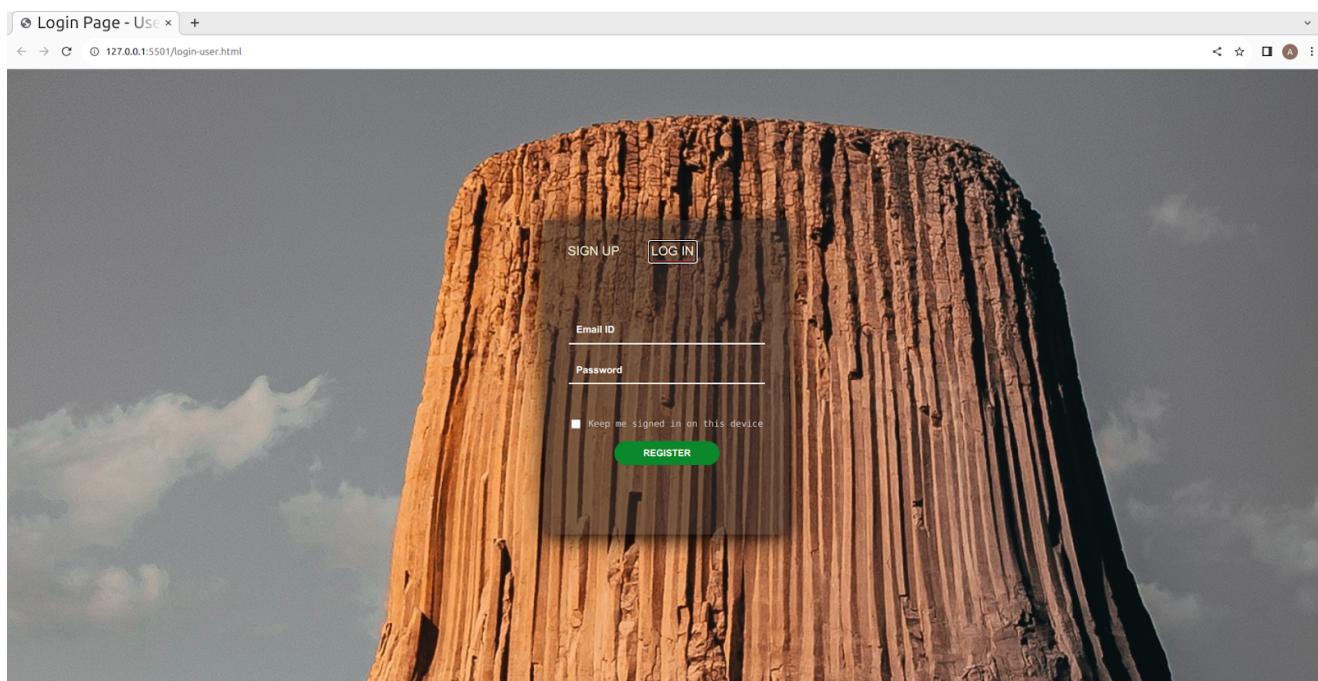
Following is the page to redirect the hotel staff and customers to their specific login page. The design for this page was inspired by the login page of Hackerrank website[6].

2. Login Page

Sign up page as user



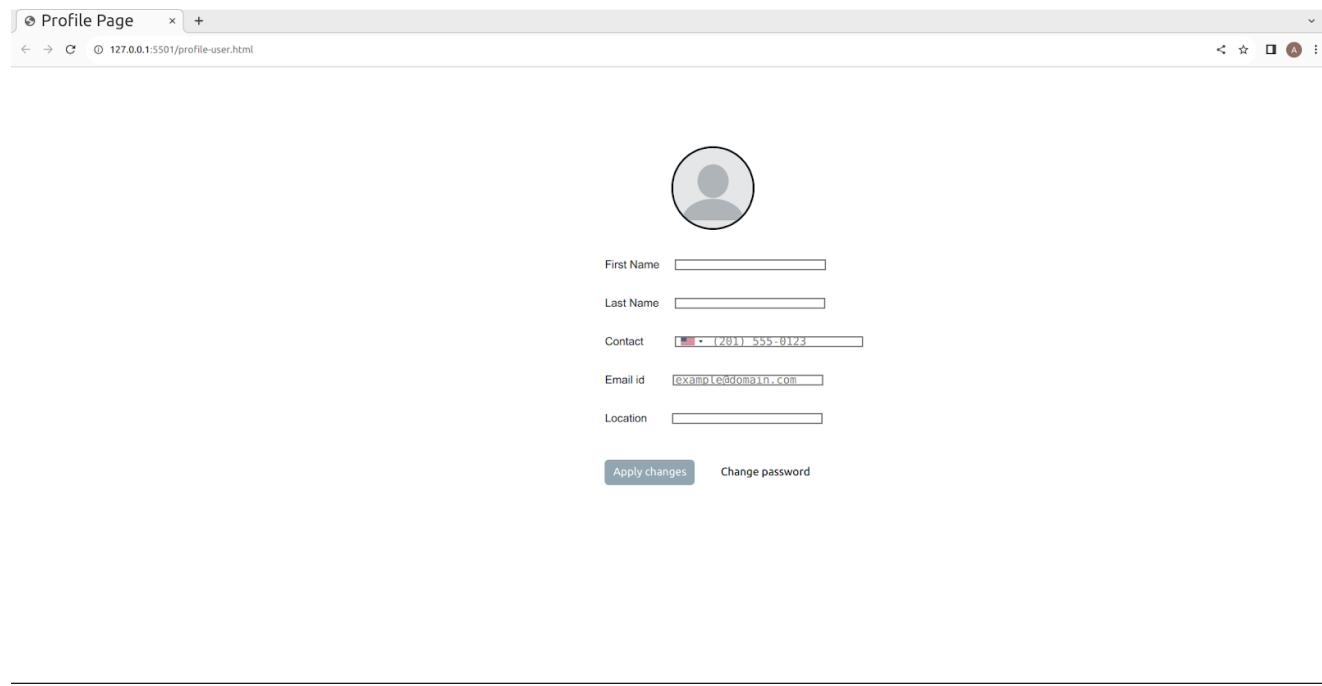
Login page as user



After choosing the mode of login, the user is prompted to login / signup(in the case there is no account)

3. Profile Page

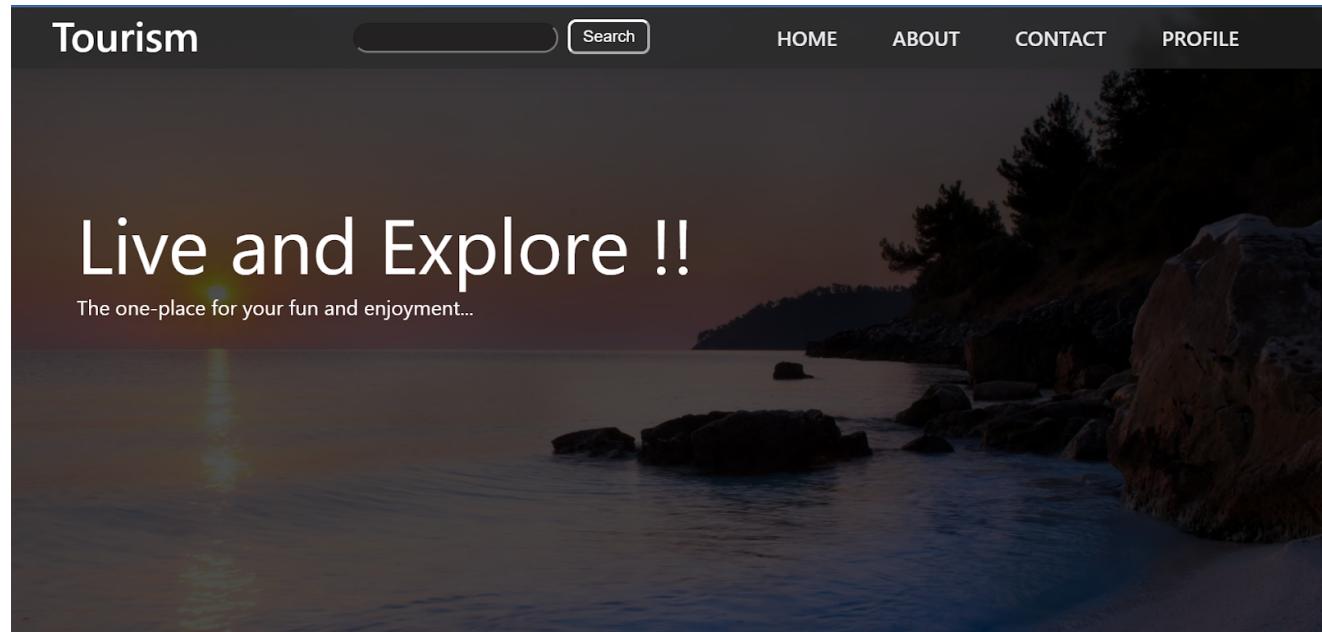
Profile page for user



The screenshot shows a web browser window with the title "Profile Page". The address bar indicates the URL is 127.0.0.1:5501/profile-user.html. The page content includes a large circular placeholder for a profile picture. Below it are five input fields: "First Name" (empty), "Last Name" (empty), "Contact" (containing "(201) 555-0123"), "Email id" (containing "example@domain.com"), and "Location" (empty). At the bottom are two buttons: "Apply changes" and "Change password".

This page is used to view and alter the user / hotel administrator details. In the case of a normal user, the hotel name option will not be displayed.

4. Home Page



This is the main page of the website which emphasizes complete description and functionalities.

6. About page

The screenshot shows the 'About' page of a website titled 'Tourism'. At the top, there is a navigation bar with links for 'HOME', 'ABOUT' (which is highlighted in blue), 'CONTACT', and 'PROFILE'. Below the navigation bar, the main content area features a large heading 'This is Us' in white. To the right of the heading is a photograph of a long, illuminated pier stretching into the ocean under a cloudy sky. On the left side of the content area, there is a block of placeholder text (Lorem ipsum) and a small URL at the bottom left: '127.0.0.1:5500/index.html#about'.

This page of the website shows information about us along with the motive that made us take this initiative.

7. Contact page

The screenshot shows the 'CONTACT' page of the 'Tourism' website. The page layout is identical to the 'About' page, with a navigation bar at the top. Below the navigation bar, there are four contact cards, each featuring a portrait of a man with glasses and a beard, wearing a light blue shirt, with his arms crossed. The first card is for 'Sanchit Sudhakar Sakhare', the second for 'Praveen kumar P', the third for 'Aadarsh Lakshmi Narasimhan', and the fourth for 'Rishi Diwakar'. Each card includes their title ('CEO & Founder, Example Harvard University'), a list of social media icons (Globe, Twitter, LinkedIn, Facebook, Instagram), and a 'Contact' button at the bottom. The URL '127.0.0.1:5500/index.html#contact' is visible at the bottom left of the page.

This is the contact page, which gives the users the facility to contact the developers and provide them with the suggestions regarding the UI/UX inorder to make the website better.

8. Search Result Page

The screenshot shows a search results page for 'Tourism'. At the top, there is a navigation bar with links for HOME, ABOUT, CONTACT, and PROFILE. A search bar is located at the top left. Below the navigation bar, there are three search results, each enclosed in a rounded rectangle:

- Taj Hotel**
\$19.99
Some text about the jeans. Super slim and comfy lorem ipsum lorem jeansum. Lorem jeamsun denim lorem jeansum.
Some text about the jeans. Super slim and comfy lorem ipsum lorem jeansum. Lorem jeamsun denim lorem jeansum.
Some text about the jeans. Super slim and comfy lorem ipsum lorem jeansum. Lorem jeamsun denim lorem jeansum.
Some text about the jeans. Super slim and comfy lorem ipsum lorem jeansum. Lorem jeamsun denim lorem jeansum.
[Add to Cart](#)
- Taj Hotel**
\$19.99
Some text about the jeans. Super slim and comfy lorem ipsum lorem jeansum. Lorem jeamsun denim lorem jeansum.
Some text about the jeans. Super slim and comfy lorem ipsum lorem jeansum. Lorem jeamsun denim lorem jeansum.
Some text about the jeans. Super slim and comfy lorem ipsum lorem jeansum. Lorem jeamsun denim lorem jeansum.
Some text about the jeans. Super slim and comfy lorem ipsum lorem jeansum. Lorem jeamsun denim lorem jeansum.
[Add to Cart](#)
- Taj Hotel**
\$19.99
Some text about the jeans. Super slim and comfy lorem ipsum lorem jeansum. Lorem jeamsun denim lorem jeansum.

This page is used to view hotels based on search results and can be used to filter results based on different parameters.

9. Wish List page and History page

Wish List



Taj Hotel
\$19.99
Some text about the jeans. Super slim and comfy lorem ipsum lorem jeansum. Lorem jeamsun denim lorem jeansum.
Some text about the jeans. Super slim and comfy lorem ipsum lorem jeansum. Lorem jeamsun denim lorem jeansum.
Some text about the jeans. Super slim and comfy lorem ipsum lorem jeansum. Lorem jeamsun denim lorem jeansum.
Some text about the jeans. Super slim and comfy lorem ipsum lorem jeansum. Lorem jeamsun denim lorem jeansum.
[Add to Cart](#)



Taj Hotel
\$19.99
Some text about the jeans. Super slim and comfy lorem ipsum lorem jeansum. Lorem jeamsun denim lorem jeansum.
Some text about the jeans. Super slim and comfy lorem ipsum lorem jeansum. Lorem jeamsun denim lorem jeansum.

This page is used to view hotels which are added in the wish list.

History



Taj Hotel
\$19.99
Some text about the jeans. Super slim and comfy lorem ipsum lorem jeansum. Lorem jeamsun denim lorem jeansum.
Some text about the jeans. Super slim and comfy lorem ipsum lorem jeansum. Lorem jeamsun denim lorem jeansum.
Some text about the jeans. Super slim and comfy lorem ipsum lorem jeansum. Lorem jeamsun denim lorem jeansum.
Some text about the jeans. Super slim and comfy lorem ipsum lorem jeansum. Lorem jeamsun denim lorem jeansum.
[Add to Cart](#)



Taj Hotel
\$19.99
Some text about the jeans. Super slim and comfy lorem ipsum lorem jeansum. Lorem jeamsun denim lorem jeansum.
Some text about the jeans. Super slim and comfy lorem ipsum lorem jeansum. Lorem jeamsun denim lorem jeansum.

This page is used to view hotels which users have already booked.

Back-end Review

The main focus from the backend perspective was the development of databases. The following code below was implemented to create the databases.

```
class Location(models.Model):
    name = models.CharField(max_length=250)
    description = models.TextField(blank=True)
    class Meta:
        ordering = ('name',)
        verbose_name_plural = 'Locations'
    def __str__(self):
        return self.name

class Hotel(models.Model):
    name = models.CharField(max_length=250)
    image = models.ImageField(upload_to='item_images', blank=True)
    description = models.TextField(blank=True)
    is_sold = models.BooleanField(default=False)
    Location = models.ForeignKey(Location, related_name='hotels',
on_delete=models.CASCADE)
    created_by = models.ForeignKey(User, related_name='hotels',
on_delete=models.CASCADE)
    created = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return self.name

class Room(models.Model):
    Hotel = models.ForeignKey(Hotel, on_delete=models.CASCADE)
    type = models.CharField(max_length=250)
    image = models.ImageField(upload_to="products", blank=True)
    description = models.TextField(blank=True)
    price = models.DecimalField(max_digits=15, decimal_places=2, default=0.0)
    max_occupancy = models.IntegerField(default=2,
validators=[MaxValueValidator(100), MinValueValidator(1)])
    created = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return self.type

class Review(models.Model):
    Hotel = models.ForeignKey(Hotel, on_delete=models.CASCADE)
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    rate = models.IntegerField(default=10, validators=[MaxValueValidator(10),
MinValueValidator(1)])
```

```

review = models.TextField(blank=True)
created = models.DateTimeField(auto_now_add=True)

def __str__(self):
    return self.review

class Wishlist(models.Model):
    Hotel = models.ForeignKey(Hotel, on_delete=models.CASCADE)
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    created = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return self.user

```

Then migrations were made to save these changes. Then records were added. Finally with the help of views, these records were rendered to the front end.

Future Scope

This application provides users with the ability to plan and book hotels at the same time with easy wishlist creation. This way while planning users can view different prices, compare them and make decisions. Further more it provides a hassle free service addition and control by Hotel admins.

However we do have some ideas on improving this project in the future like adding Google API for location identification. Further we can expand the database for larger and more realistic data. After implementing these points, our application will be ready to be hosted for global usage.

References

- 1.<https://creately.com/diagram-type/database-design/>
- 2.<https://www.goibibo.com/>
- 3.<https://www.tripadvisor.in/>
- 4.<https://www.unwto.org/tourism-statistics/tourism-statistics-database>

5.<https://tourism.gov.in/market-research-and-statistics>

6.<https://www.hackerrank.com/get-started/>

7.<https://www.w3schools.com/>