

HOTEL MANAGEMENT SYSTEM USING DJANGO

1. Create new django project

- Create a new django project on the desktop: `django-admin startproject hms_prj`
- Create new app `userauths`, `hotel`, `room`, `addon`
- Install Apps in [settings.py](#)
- Install the required libraries, including Taggit, Crispy_forms, Import_Export, Pillow, and Shortuuid.
- Add installed packages to `requirements.txt` and update it.
- Configure [Settings.py](#) to Install Libraries
- Run Python [Manage.py](#) Runserver

2. Setup Django Admin

- Install Django Jazzmin
- Add Jazzmin to [settings.py](#) INSTALLED apps Section
- Configure Jazzmin Settings
- Configure Jazzmin UI Tweaks
- Create Superuser
- Migrate Database
- Login to the admin section.

3. Configure Templates and Static Files

- Configure template configuration in TEMPLATES Section [settings.py](#)
- Create a new folder under Templates.
- Configure the static and media files in [settings.py](#) and [urls.py](#)
- Create media and static folders.
- Copy All Static Files and Add to static folder

4. Custom User and Profile Model

- Create Custom user and profile models in `core.models`
- Register Model in Admin and Add filters, Displays, etc.
- Run Migration Commands
- Test Model in Django Admin Panel

5. Register User Authentication

- In the user authentication app, create a new [forms.py](#) file.
- Create UserRegisterForm in the new [forms.py](#)
- Import In Views
- Create new Register View to sign up a new user
- Configure the URL for the view
- Configure userauths.urls in facebook_prj.urls
- Create a new folder, userauths > sign-up.html
- Copy the login_register.html and paste in the sign-up.html
- Edit the template to display the forms
- Create the error check div
- Test Registration System

6. Login User Authentication

- In the user auth.views, create new view for login users
- Configure Url
- Add Names to template email and password input
- Test Login Functionality

7. Logout User Authentication

- In the user auth.views, create new view for login users
- Configure Url
- Add Names to template email and password input
- Test Logout Functionality

8. Template Inheritance in Django

- Create partials folder > base.html
- Copy and paste feed code in index.html
- Cut the header and footer and it in base.html
- Explain the concept of the blocks, extends, load e.t.c

9. Creating Model for HMS

- Create and register models for hms

10. CKEDITOR in Django

- Install CKEditor 5 pip Install django-ckeditor
- Add CKEditor to installed apps in [settings.py](#) 'django_ckeditor_5',
- Add CKEditor Url in main [urls.py](#) path("ckeditor5/", include('django_ckeditor_5.urls')),
- Add CKEditor snippets

11. Hotel List Page

- In hotel views.py, create a new function to list all the hotels from the database to the frontend

12. Hotel Detail Page

- In hotel views.py, create a new function to get a hotel from the database to the frontend

13. Hotel Room Types Detail Page

- In hotel views.py, create a new function to get the types of rooms in a particular hotel and list it in the frontend

14. Hotel Room Detail Page

- In hotel views.py, create a new function to get a hotel from the database to the frontend

15. Check Room Availability

- Create new app called booking and write a function to check room availability

16. Add Room to Selection 1

- In booking view, write a new function that add rooms to selection session

17. Add Room to Selection 2

- Configure the HTML page to communicate with the JQuery

18. Add Room to Selection 3

- Create a new custom.js file and link it to the base.html
- Write a new JQuery ajax function to add room to selection dynamically

19. Selected Room Detail View

- Create a new function to list all the selected rooms from the session

20. Form to save booking data

- Create a new function to list all the selected rooms from the session

21. Delete Rooms 1

- Create a new django function to delete items from selection

22. Delete Rooms 2

- Configure the html template to communicate with ajax

23. Delete Rooms 3

- Write jquery code to get and send data to django server

24. Checkout Page

- Write code to display checkout details

25. Coupon and Discount

- Write code in checkout view to apply coupon feature

26. Django Environment Variables

- Install environs package - pip install environs
- In settings.py, import environs. - from environs import Env
- Initiate the Env method

```
env = Env()  
env.read_env()
```

27. Stripe Payment Integrations

- Install stripe payment package - pip install stripe
- Import stripe in views.py
- Create a new checkout session.
- Configure the url
- Configure template
- Test Payment

28. Payment Success and Failed Pages

- Create pages for payment success and failure.

29. Notification System

- Create a new notification system object

30. Invoice Pages

- Create a new view to get the order details and create invoice.html page
- Populate html elements with data from database

31. PayPal Payment Integration

- Import the PayPal cdn
- Write the code to render the PayPal button and process payments.
- Redirect to success page
- Test payment integration.

32. Flutterwave Payment Integration

- Import the PayPal cdn
- Write the code to render the PayPal button and process payments.
- Redirect to success page

- Test payment integration.

33. Send an email after payment.

- Install the anymail library: `pip install django-anymail`
- Register the package in the settings.py install app section
- Configure the email backend

```
EMAIL_BACKEND = "anymail.backends.sendgrid.EmailBackend"
```

- Configure the default from email

```
DEFAULT_FROM_EMAIL = env("DJANGO_DEFAULT_FROM_EMAIL")
```

- Configure the AnyMail settings

```
ANYMAIL = { "SENDGRID_API_KEY": env('SENDGRID_API_KEY'), }
```

- Configure Email Template
- Test Email

34. User Dashboard

- Create view to display user dashboard
- Create URLs
- Configure template

35. List All Booking and Stats

- In dashboard views write a function to show some stats on the dashboard
- Display the stats on the dashboar page

36. Create Booking Detail Page

- Create a new function to get the booking detail
- Create a url for the new view
- Configure the template
- List all the booking details on the new page that was created

37. Create All Booking Page

- Create a new function to get all the booking of a user
- Create url for the view
- Create a new template and list out all the booking from the database

38. Create Notification Pages

- Create a new notification page
- Create noti query in the context processor

- Show notifications in the dashboard base

39. Create Wallet Page

- Create a new wallet page
- Create new URLs
- Display wallet related informations

40. Create Bookmark Page

- Create a new view for the bookmark
- Configure the url
- List all the bookmarks from the database

41. Add to Bookmark Page

- Create new view to get data from the client side, create or delete bookmarks.
- Configure the URL
- Configure the HTML to communicate with the js
- Write JS code to send data to the server.

42. Create Profile Page

- Create new view display and update all the profile informations
- Create the profile and user update forms
- Configure the template and the urls

43. Change Password

- Create new url to change password
- Create the password changed success page
- Configure the url

44. Reviews and Ratings

- Write function to create review in user_dashboard views.py
- Configure html to communicate with JS
- Write JS code to get and send data to the server
- Test Feature