HOTEL ROOM BOOKING SYSTEM

AIM : TO DESIGN WEB APPLICATION ON HOTEL ROOM BOOKING SYSTEM USING DJANGO FRAMEWORK IN PYTHON

OBJECTIVE :

- ➔ IN This design we are implementing user Registration and user withdrawal from the hotel room
- ➔ Hotel room booking system, allowing users to register for different types of rooms, withdraw registrations, and receive email notifications regarding their registration status. The admin can view statistics and manage registration records
- ➔ We are using HTML,CSS,Bootstrap,javascript for FrontEnd part while BackEnd We are using Database and DTL language

FLOW OF EXECUTION :

STEP-1 : HOME PAGE

STEP-2 : NAVBAR PAGE

- ➢ HOME PAGE
- ➢ ABOUT PAGE
- ➢ REGISTRATION
- ➢ WITHDRAWAL
- ➢ ADMIN-LOGIN

STEP-3 : IF USER WANT TO REGISTER

- ➢ PERFERENCE
- ➢ PAYING AMOUNT

STEP-4 : IF USER WANT TO WITHDRAWAL

- ➢ BY ENTREING REGISTER-ID
- ➢ FINAL WITHDRAWL

STEP-5 : END

PROCESS:

1. **Import Statements**:

- The code begins with import statements that bring in necessary modules and functions from Django and other Python modules.

2. **Views Functions**:

- The code defines several view functions, which are used to handle HTTP requests and return HTTP responses.

- `start(request)`, `home(request)`, and `about(request)` are simple view functions that render HTML templates when the corresponding URLs are accessed.

- `next(request)` is used for handling room selection. It checks the user's choice from a form submission and redirects them to the appropriate registration page based on their choice.

- `acroom(request)`, `nacroom(request)`, and `sacroom(request)` handle the registration of different types of rooms. They create a new registration record in the database with user-provided information.

- `withd(request)` handles room withdrawal requests. It checks if the withdrawal is allowed, updates the registration status, and sends email notifications.

- `adm(request)` renders an admin page with statistics about room registrations, withdrawals, and payments.

- `withlist(request)` renders a page that lists room withdrawal requests.

- `dwl(request, v)` processes a room withdrawal request. It sends an email notification to the user, updates the payment amount, and changes the registration status.

- `popu(request, s)` sends a payment successful email notification to the user after room registration.

- `alre(request)` and `com(request)` render pages to indicate that a room withdrawal request has already been processed or that the room registration process is completed.

- `adnot(request, s)` sends a notification email to the user when a selected room type is not available and updates the registration status accordingly.

- `nor(request)` renders a page indicating that an error occurred (likely for handling exceptions).

3. **Database Models**:

- The code interacts with a database using Django models. The database model used is named `Reg`. The model is not provided in the code snippet, but it likely contains fields for user information, room type, payment, and registration status.

4. **Email Notifications**:

   - The code uses Django's `send_mail` function to send email notifications to users. It sends emails for payment confirmation, withdrawal requests, and room availability notifications.

5. **Aggregate Query**:

   - In the `adm(request)` view, an aggregate query is used to calculate the sum of payments (`x["pay__sum"]`) from all registration records.

6. **HTML Templates**:

   - Throughout the code, the `render` function is used to render HTML templates, passing data to the templates as context.