

Project 4: Paying Guest Accommodation

Project objective:

Build an application to simulate Paying Guest Accommodation System

Background of the problem statement:

You, as a Full Stack Developer, has been assigned a project to create an application that can assist users in finding the desirable PG accommodation near the workplace. The application can store all information pertaining to the user's personal information and locality to find out the best and nearby accommodation in reasonable rental amounts.

The tasks that need to be performed are as follows:

- Create a database using MySQL
- Create the necessary code for the various layers of the application
 - Entity Layer: Create tenant, owner, and admin entities with appropriate relationship mappings
 - Repository Layer: Create a layer for all database operations
 - Service Layer: Demonstrate dependency injection and follow interface segregation principle
 - Controller Layer: Expose necessary APIs such as:
 - Registration of tenant on the application
 - Retrieve all the places available for PGs in a specific city
 - Retrieve a profile of a specific PG accommodation place
 - Search for an accommodation place in a specific locality
 - Retrieve the owner details like name and contact information for a specific PG accommodation place
 - Add a new place available for PG
 - Change availability status once the place has been occupied by any tenant

The flow and features of the application:

- The APIs should be used for user interface where the tenant can:
 - List out all the places available for accommodation in a specific city.
 - List all the places available for accommodation in a specific locality (They can search the data by entering the location).
 - Opt for a specific accommodation place for details.
 - View the details like name, mobile number, and email address of the place owner for further communication.
- The APIs should be used for user interface where the owner can:
 - Add a new place and post it on the website along with the place details and the status as available. The accommodation place can be posted with the factors like registration number, built-up area, rent amount, city, locality, availability, and so on.
 - Edit the details of added place.
 - Change the status from available to occupied and vice versa.
 - View the count of visitors to his post of accommodation place.

Technology Specification:

The following technologies are used in the development of this application:

- Spring Tool Suite
- Java
- MySQL
- Spring Data JPA
- Spring Rest
- Postman

Validation (Frontend/Backend):

The following validations must have to be implemented in this application:

- Users can retrieve the data of any PG accommodation place only if the valid ID has been provided
- Users can inquire or get access to the contact information of the PG place owner only if the place is available (status is available). If the status is occupied, then the user will have access only to the place details and not the owner's contact details.
- Users can search the PG accommodation by entering the locality where the locality must have to be a text input.

Business logic/Exception scenarios:

- To implement the application's proper workflow, the database must contain information about the accommodation place, city, and locality.
- The tenant and owner both parties must have to be 18 years old to register or post anything on the portal. The student is required to implement the code to handle this scenario.

The following requirements should be met:

- To claim the project is completed, the student must ensure that all the details, such as a list of posts, search results, and update or delete operations are present in a valid format.
- The student must perform all CRUD operations on the entire data.

API Endpoints:

Sr. No.	API	Endpoint
1	Retrieve available PG places in a specific city	http://localhost:8085/pg/{city_id}
2	List out available PG places in a specific locality	http://localhost:8085/pg/{locality}
3	Retrieve a specific PG place detail by id	http://localhost:8085/pg/details/{id}
4	Retrieve the owner details of a specific PG	http://localhost:8085/pg/owner/{id}
5	Add a new PG place	http://localhost:8085/owner/places/add
6	Retrieve all PG places added by the owner	http://localhost:8085/owner/places
7	Change the status of PG place (Available or Not Available)	http://localhost:8085/owner/places/{id}
8	Edit Place details	http://localhost:8085/owner/places/edit
9	Delete Place Details	http://localhost:8085/owner/places/delete

Note:

1. Do the APIs Functional Testing using Postman.
2. Create the documentation for these APIs using Swagger.