Internship Report and Documentation

Tech Stack:

- Frontend: HTML, CSS, Bootstrap
- Backend: Nodejs with Express
- Database: MongoDB Atlas
- Authentication: JWT
- Password Encryption: bcrypt

ATLAS URL
mongodb+srv://gharpayywebsite:Azad1234@cluster0.ij5yv.mongodb.net/gharpayy?retryWrites=true&w=majority&appName=Cluster0

Folder Structure for Gharpayy.in

- Adminpages
  - css
  - fonts
  - js
  - libs
  - html files
- Controllers
  - listingController
  - userController
- Middlewares
- Models
  - Enquiry.js
  - Listing.js
  - User.js
- Public : contains user side page code
  - css
  - fonts
  - gurugram
  - images
  - img
  - js
  - video
  - webfonts
  - html files
- Routes
- Uploads : contains images uploaded in admin side forms
- .env
- Server.js

ADMIN SIDE

➢ **How to access**

- Url : https://admin.gharpayy.in/admin
- Credentials : gharpayywebsite@gmail.com , abc123
- Password can be changed by going to profile page in admin portal

➢ **Features**

- Authentication
- Dashboard
- Add a new Listing
- Update a listing
- Disable / enable listings
- Admin Profile page
- Reset Password

➢ **Files and description**

Page	**File Name**	**JS logic for backend**
Admin Dashboard	admin-index.html	allListing.js
Login Page	login.html	Inside html file
Add a new Listing	addListing.html	addListing.js
See all Listings	allListings.html	fetchListings.js
Update Listing	invoice.html	updateListing.js
Profile Page	profile-setting.html	Inside html file
Reset Password	reset-password.html	Inside html file
Logout functionality	All files	logout.js

DATABASE SCHEMA

Can be found in models folder

- ➢ Listings
- ➢ Users

Listings Schema Fields

name (String, required)
➢ Name of the listing (e.g., "Ace Girls, Koramangala").

location (String, required)
➢ Physical address or city of the property.

description (String)
➢ A short description about the property and its offerings.

photos ([String])
➢ Array of image URLs or filenames for the listing's images.

status (String, enum: ['open', 'not_open'], default: 'open')
➢ Indicates whether the listing is currently available to users.

openDate (String, default: 'NA')
➢ Opening date of the listing (used when status = 'not_open').

gender (String, enum: ['M', 'F', 'Coed'], required)
➢ Type of tenants the property is meant for:

propType (String, enum: ['ADVANTAGE', 'BASIC PLUS', 'CLASSICS', 'LUXE MAX'], required)
➢ Category/tier of the property (helps in filtering/listing types).

amenities (Object)
➢ Various groups of amenities offered by the property:

kitchenEssentials: e.g., microwave, fridge
utilities: e.g., Wi-Fi, geyser
comfort: e.g., AC, mattress
facilities: e.g., laundry, gym
security: e.g., CCTV, guard
additional: e.g., house help, entertainment

disabled (Boolean, default: false)
➤ Flag to soft-delete or disable the listing from being shown.

createdAt (Date, default: Date.now)
➤ Timestamp when the listing was created.

User Schema

name (String, required)
➤ Full name of the user (admin).

email (String, required, unique)
➤ Email address used for login.

password (String, required)
➤ Hashed password of the user

adminType (String, enum: ['super', 'regular'], required)
➤ Defines the level of admin access:

super → Full access to all features (possibly manage users too).
regular → Limited access (maybe can't manage other admins)

BACKEND API ENDPOINTS

GET	api/listings/	Get all listings
GET	api/listings/enabled	Get all enabled listings
GET	api/listings/stats	Get stats for dashboard
POST	api/listings/	Add new listing
GET	api/listings/:id	Edit a listing
PUT	api/listings/:id	Update a listing
PUT	api/listings/:id/disable	Disable a listing
PUT	api/listings/:id/enable	Enable a listing
POST	api/listings/upload-photo	Upload photos for listing
DELETE	api/listings/delete-photo	Delete photos for listing

POST	api/users/login	User Login
GET	api/users/profile/:id	Get Profile Information
POST	api/users/reset-password	Reset password
POST	api/users/logout	Logout
POST	api/users/verify-password	Verify Password

PHOTO UPLOAD AND MANAGEMENT

Middleware Used: Multer

Max File Size allowed (per photo) : 5MB

Accepted Formats: only jpeg, jpg, and png formats

Storage: Photos are stored in ‘Uploads’ folder

Filename: unique filenames (timestamp + original name)

Code/ Logic in folder: backend/routes/listingroutes.js

Upload Endpoint

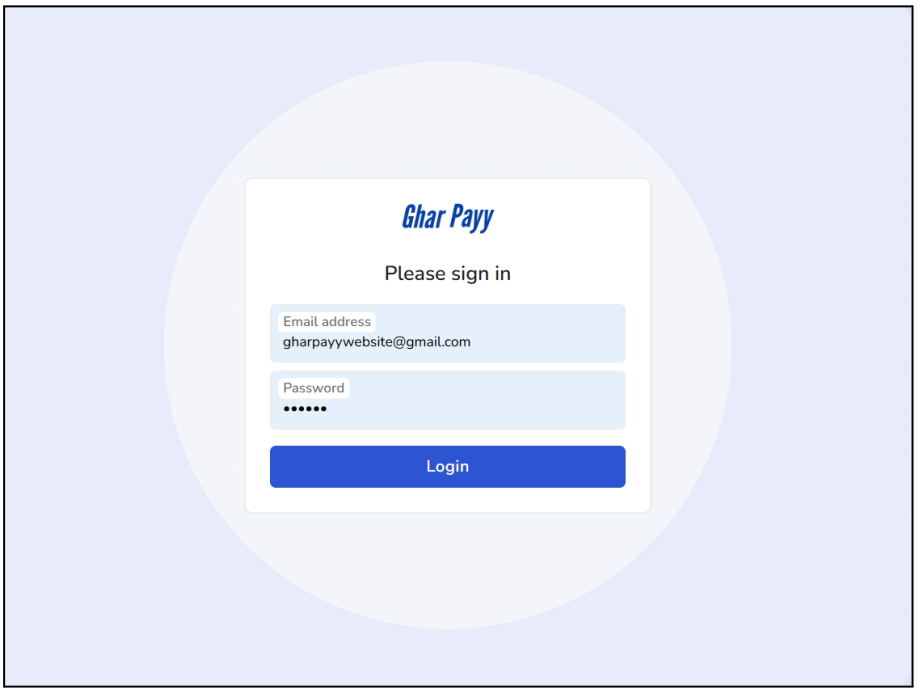
- Route: POST api/listings/upload-photo
- Middleware: upload.single('photo')
- Response:
  On success: { photoPath: "/uploads/filename.jpg" }
  On failure: { error: "No photo uploaded" } or file type error.

Delete Photo Endpoint

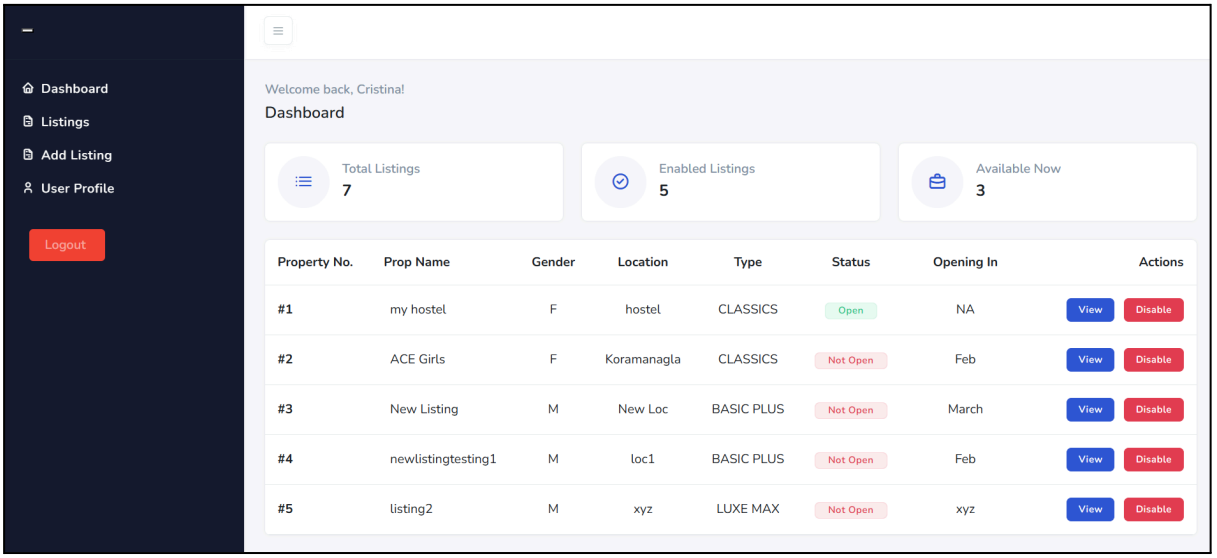
- Route: DELETE api/listings/delete-photo
- Body Parameter: { "photoPath": "/uploads/filename.jpg" }
- Response:
  On success: { success: true, message: "File deleted successfully" }
  On failure: { success: false, message: "Failed to delete file"

SCREENSHOTS OF ADMIN SIDE PAGES

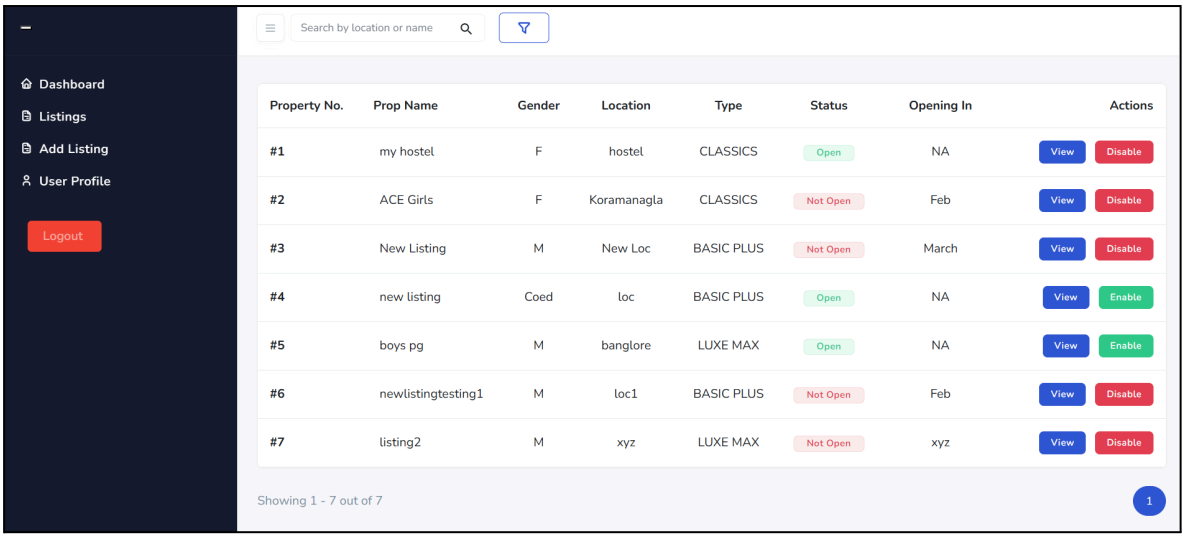
1. Login Page



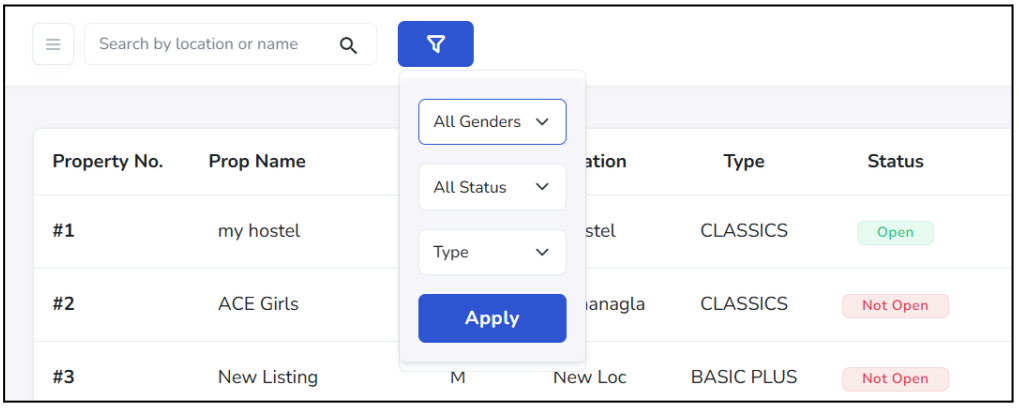
2. Dashboard



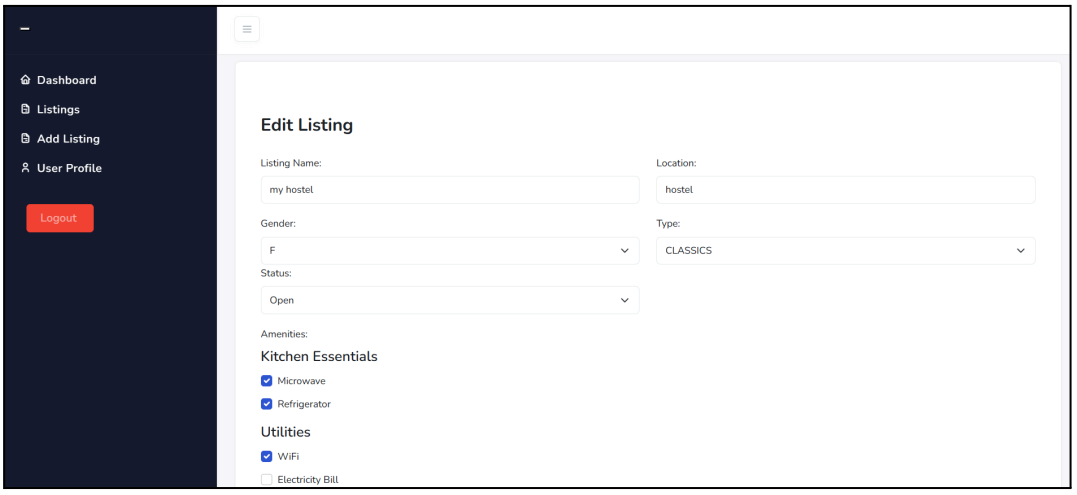
3. All Listings Page



4. Filter and Search Options

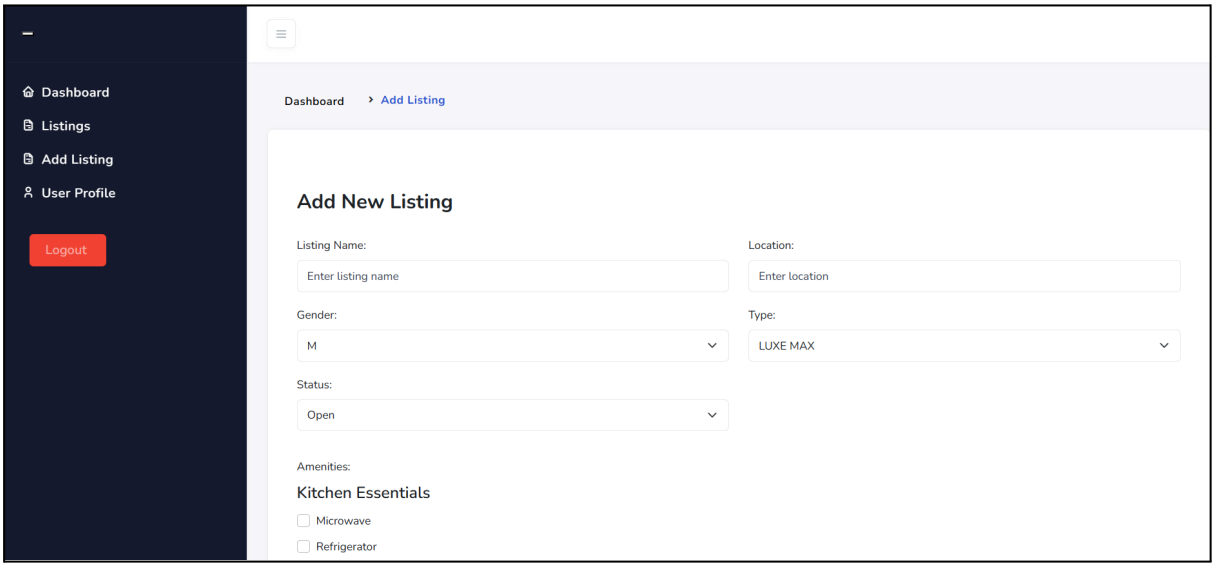


5. Click on view button to edit a listing

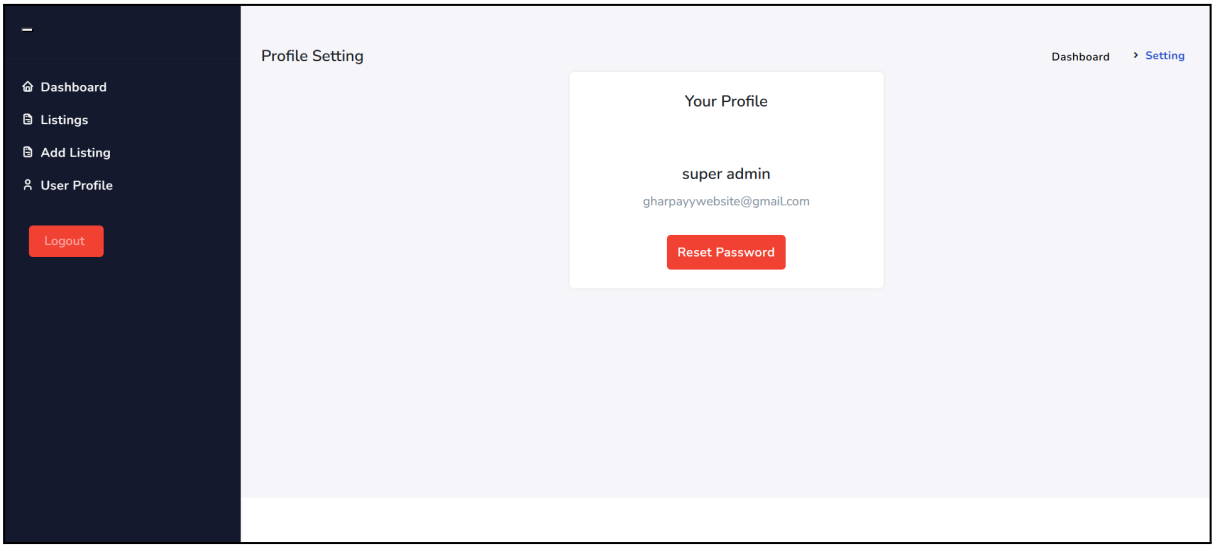


6. Click on cross icon next to every photo in order to delete it and click on upload photo to upload a new photo

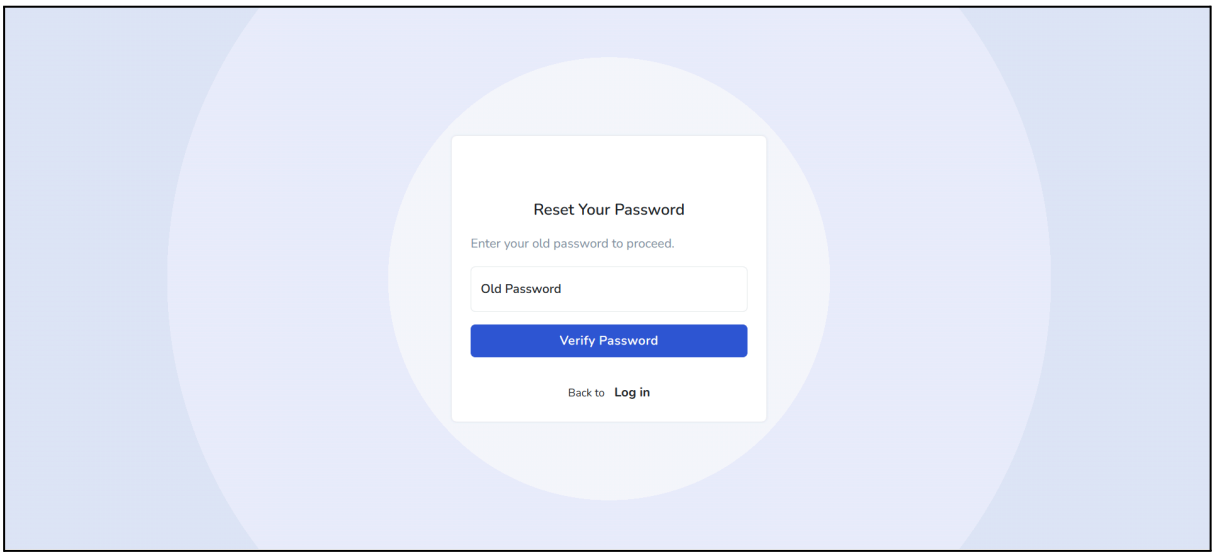
7. Form to Add a new listing



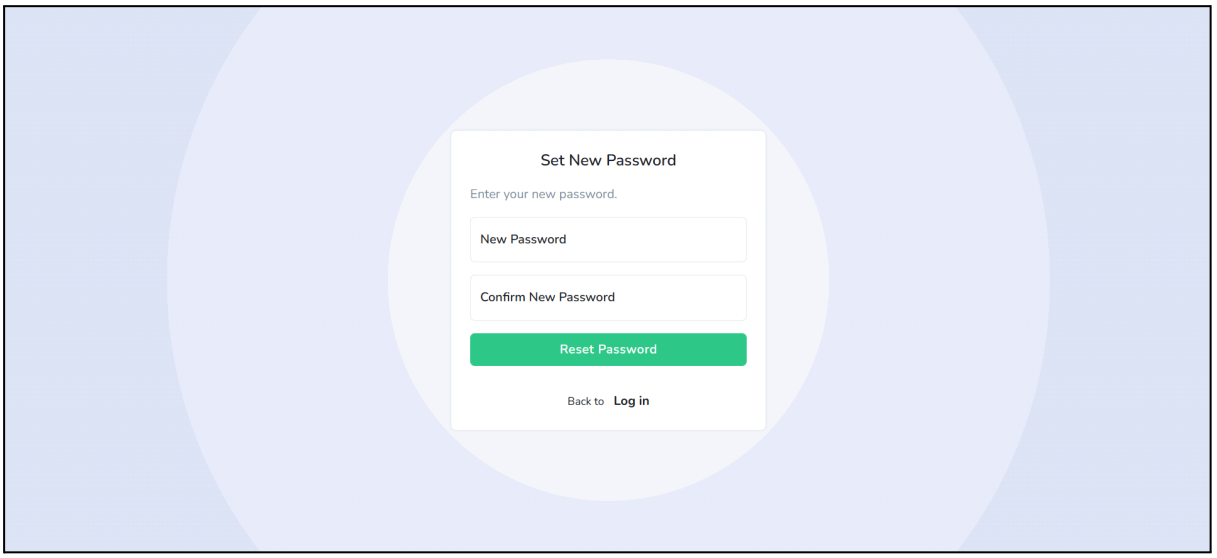
8. User Profile



9. Reset Password Screen

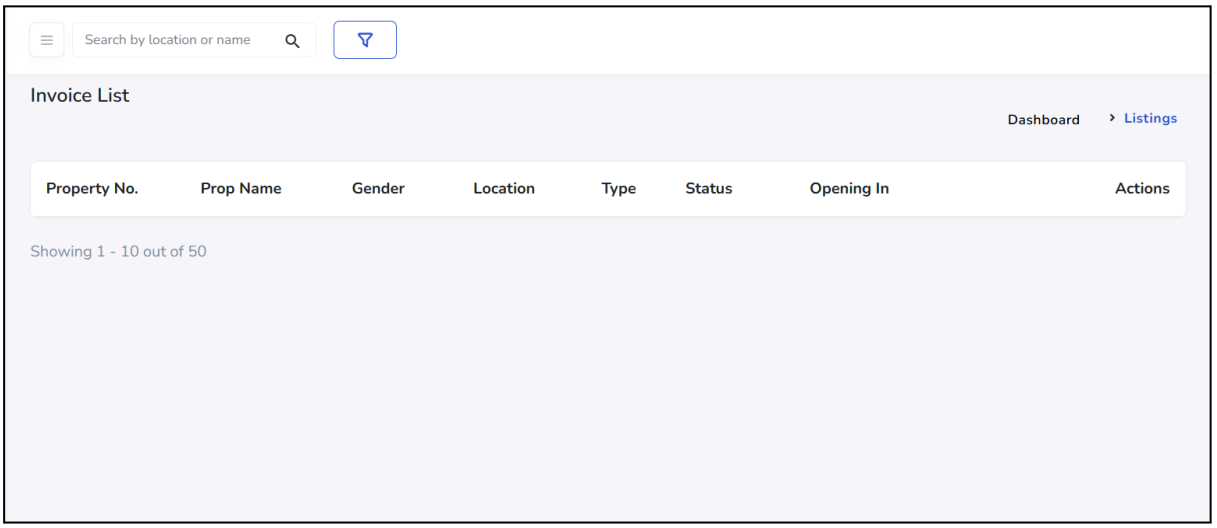


10. Set new password screen

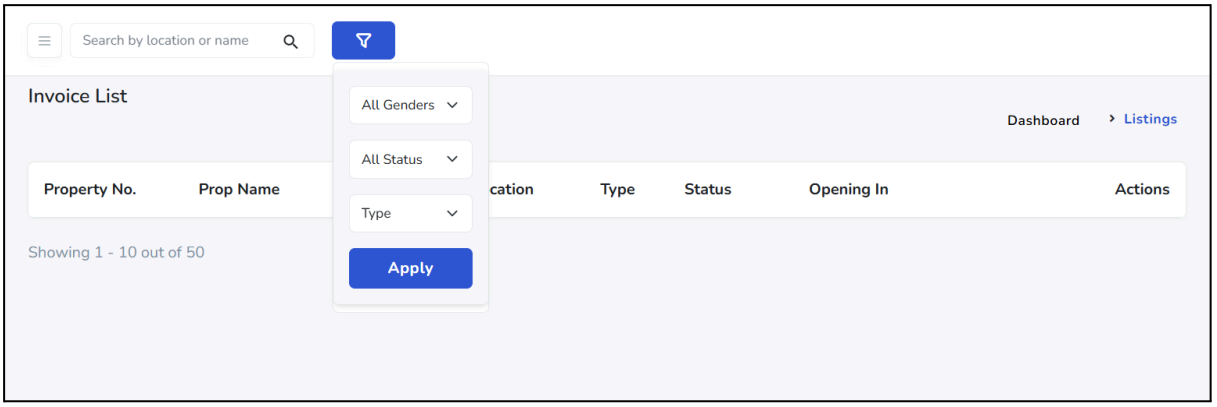


BUG IN VIEW ALL LISTINGS PAGE

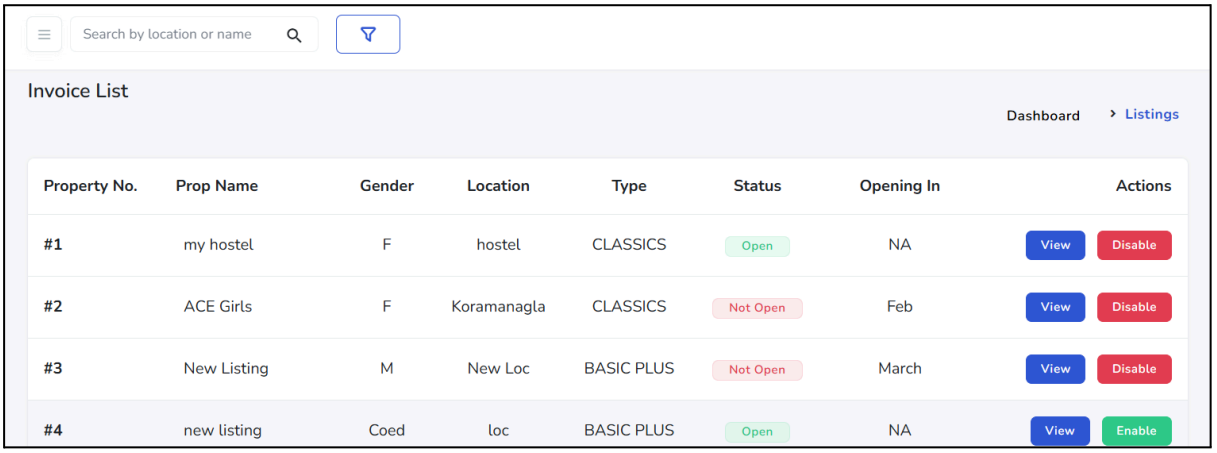
Bug: No listings are visible



Temporary Fix:
Click on the filter icon and just click on apply, no need to select anything

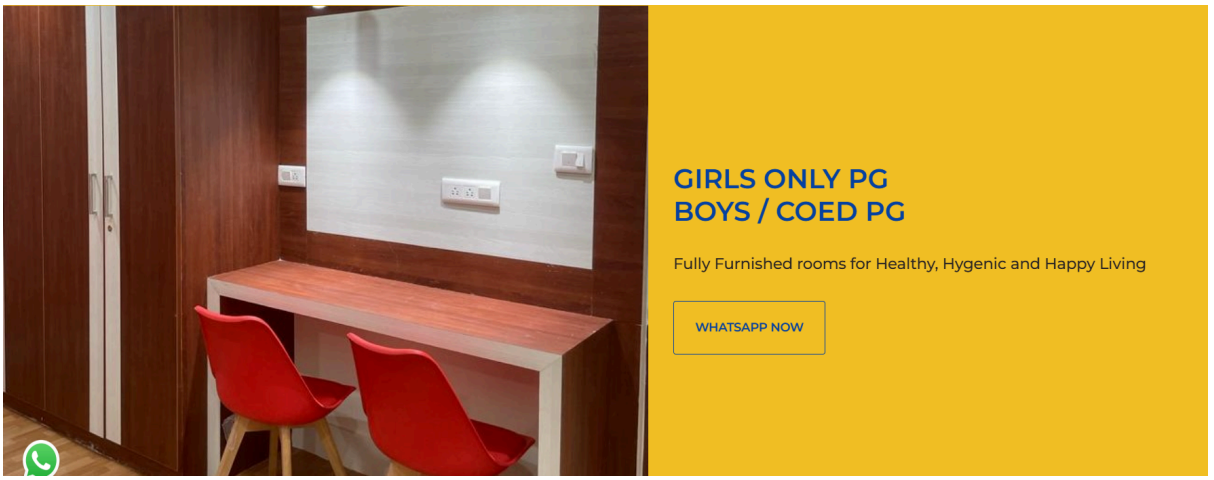


Fixed..



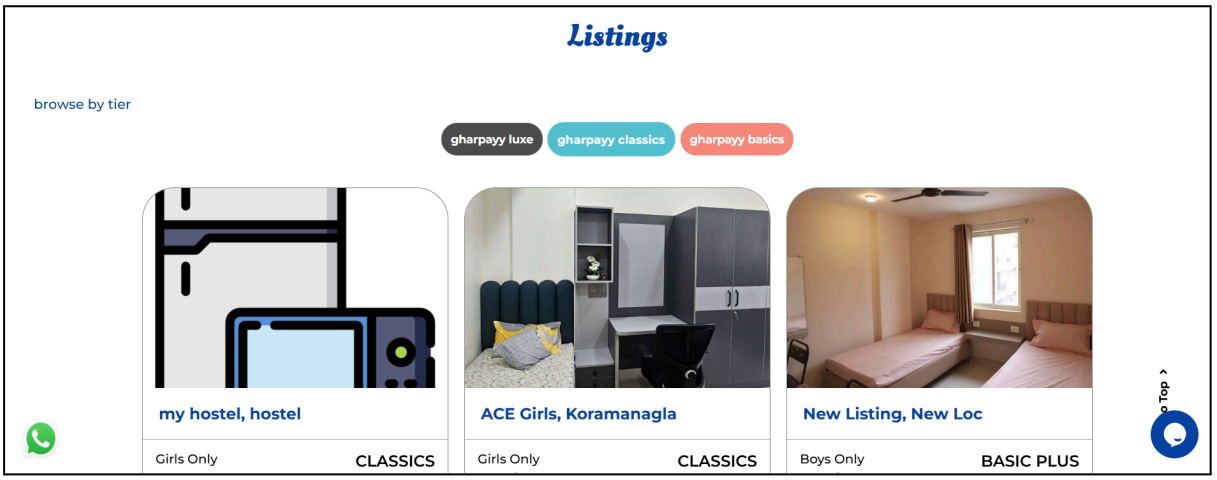
CLIENT SIDE CHANGES

1. New section added



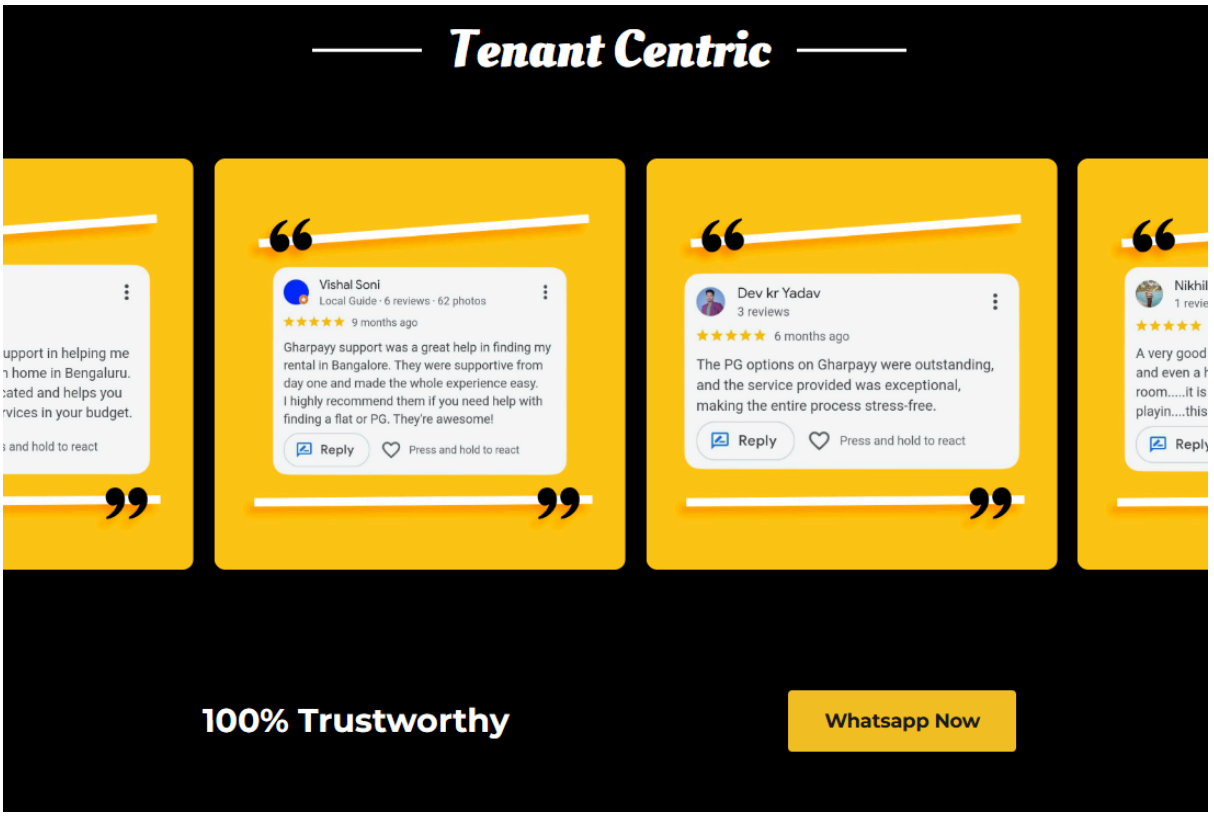
2. New Section 2 - Horizontally scrollable

3. Listings display section (Fetched from backend)



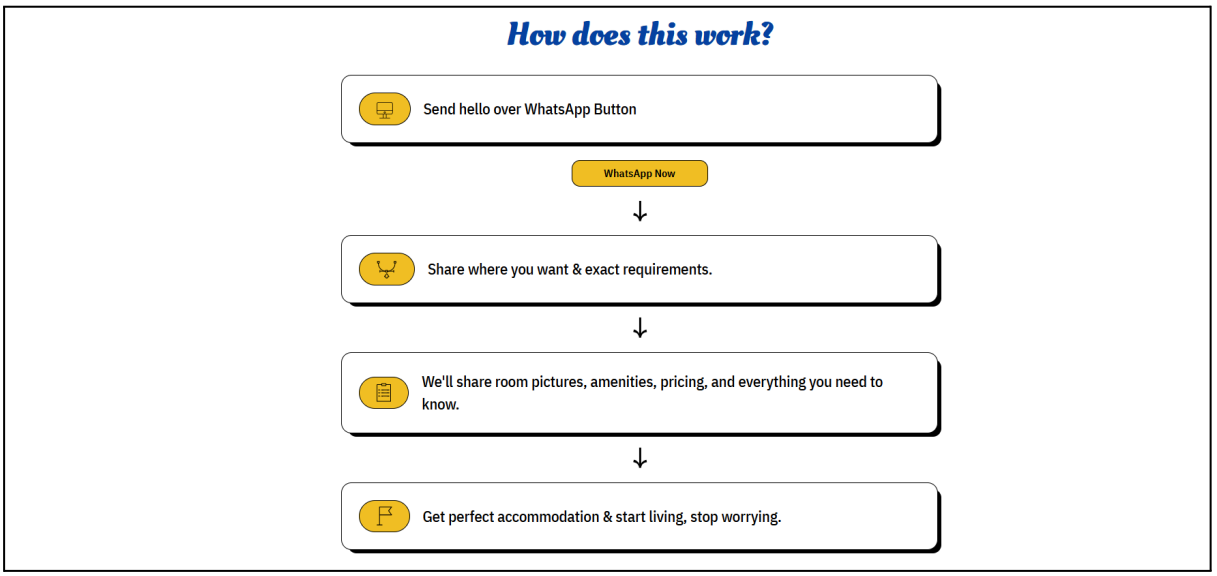
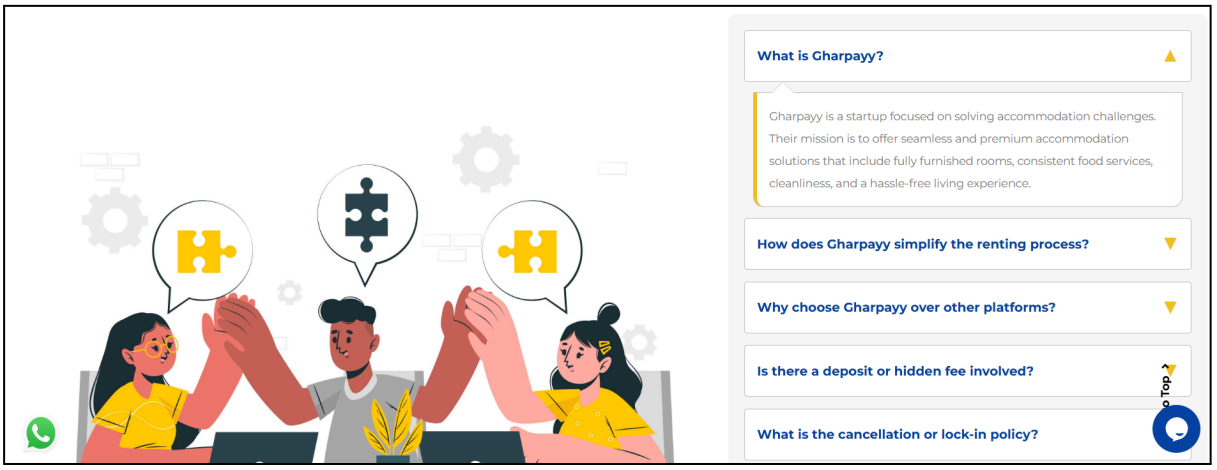
Click on any of the listing property to open a detailed view of that listing (Connected with backend)

4. Review Section - updated



5. Pictograph Section (Horizontally scrollable)

6. How does this work (new section added)



7. FAQ Section - Modified