

Project Name: UpaayaX (v1)

Frontend: Next Js

Backend: Node js

Database: MYSQL

ORM: SEQUELIZER

Tools, Frameworks and Protocols: Websocket, TailwindCSS,

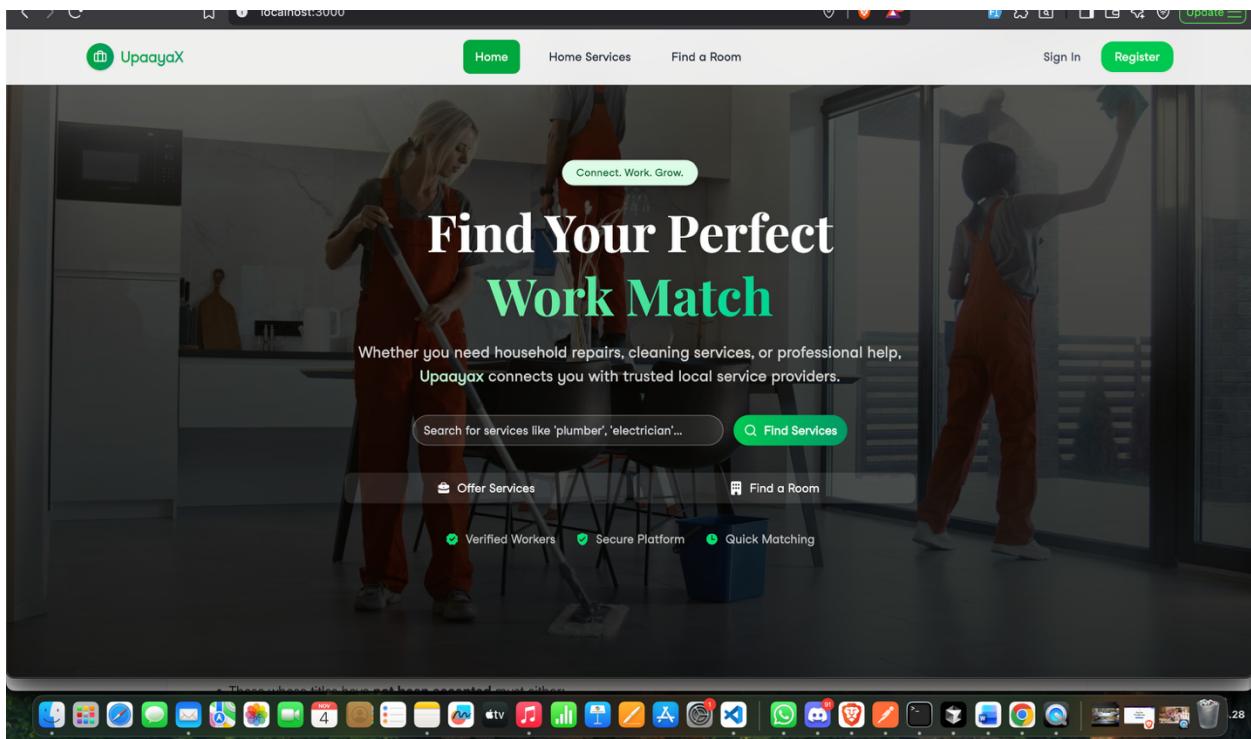
f

## Table of contents

### Table of Contents

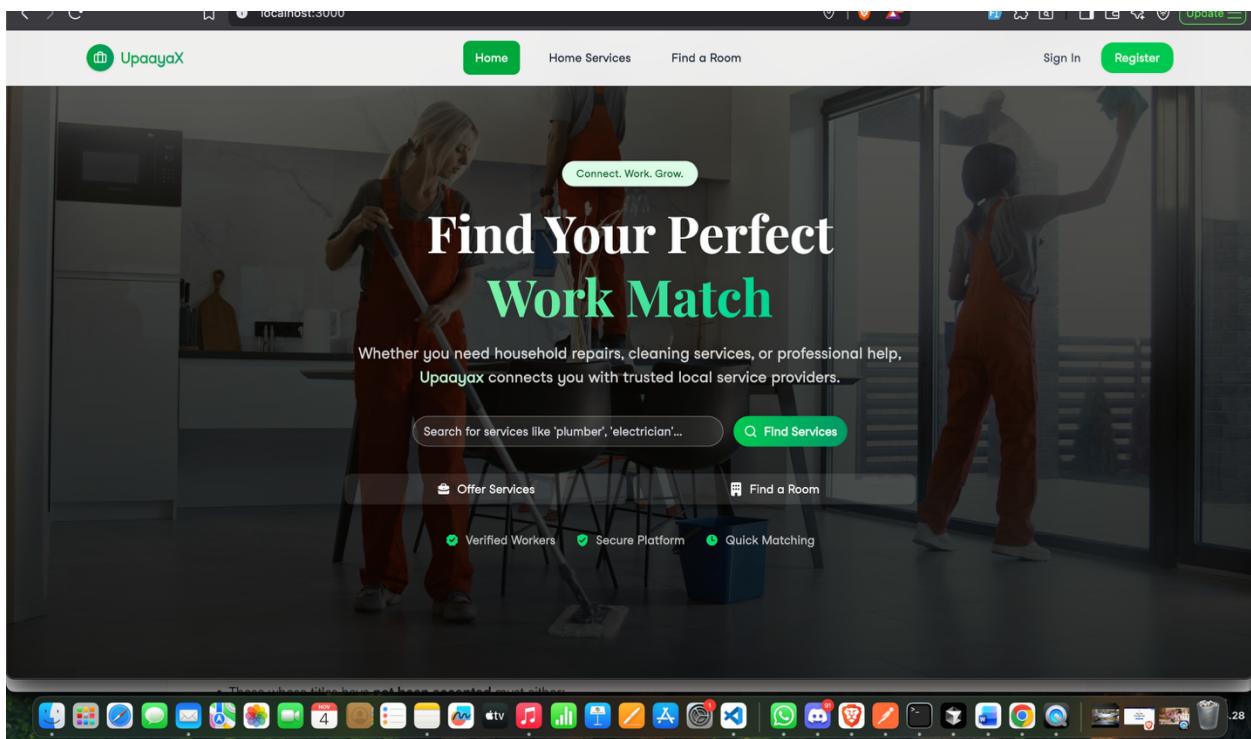
<b>1</b>	<i>Introduction of project</i>	3
<b>2</b>	<i>Fronte pages and its features</i>	4
2.1	Search feature	5
<b>3</b>	<i>Roles of the system</i>	7
<b>4</b>	<i>Registering as a User and then getting other roles flow</i>	7
<b>5</b>	<i>Login Page:</i>	13
5.1	Forget Password	14
<b>6</b>	<i>Getting role service Provider</i>	18
6.1	Adding the services as a service provider	28
<b>7</b>	<i>Profile photo change</i>	36
<b>8</b>	<i>For payment account creation provider</i>	38
<b>9</b>	<i>For booking status in provider part</i>	40
<b>10</b>	<i>For gharbeti Role</i>	41
10.1	Add room	45
10.2	For gharbeti profile	49
<b>11</b>	<i>For booking part from user</i>	51
<b>12</b>	<i>My booking</i>	54
<b>13</b>	<i>Admin Site</i>	61
13.1	manage user part	65
13.2	Manage the kyc	71
13.3	<b>Services</b>	74
13.4	Deactivating the services	75
13.5	Manage rooms	77
13.6	Manage Categories	78
13.7	Payment section	82

# 1 Introduction of project

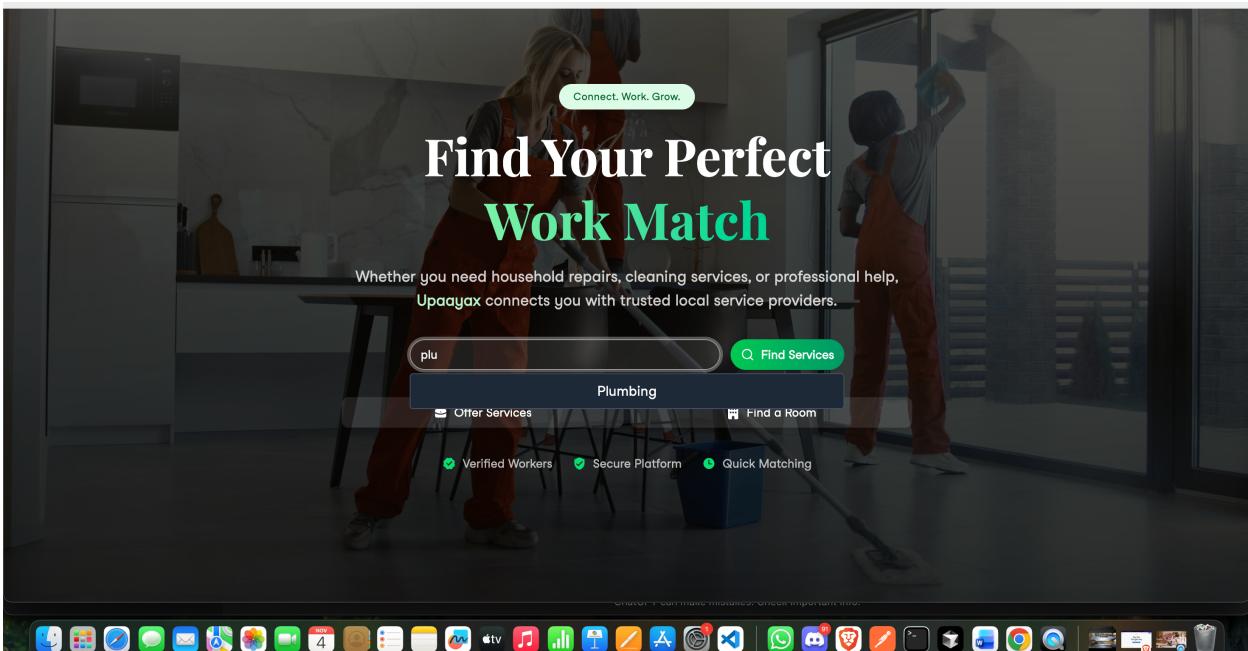


The project fully focusses on home services and room rental system.

## 2 Front pages and its features



## 2.1 Search feature



The search feature gives the suggestion on to the available services. On clicking of that find services it navigates to the home-services section

A screenshot of the Upayax "Home Services" search results page. The URL in the browser bar is "localhost:3000/home-services?search=Plumbing". The page features a header with the Upayax logo, navigation links for "Home", "Home Services" (which is highlighted in green), "Find a Room", "Sign In", and "Register". On the left side, there are three filter boxes: "Category" (set to "Plumbing"), "Price Range" (with "Min Price" and "Max Price" fields), and "Your Location" (showing "Latitude: 26.6619, Longitude: 87.2741" and a "Show Map" button). The main content area is titled "Available Services" and displays a message: "No services available." There are also some small icons at the bottom of the page.

The flow for this search is:

1. The search params: on that URL is what our home-services left side category filters takes
2. If the search params is input random values, then the category section sees if that categories are there on that options or not, if not then it fallbacks to all categories.

Like this:

The screenshot shows a web browser window with the URL `localhost:3000/home-services?search=jdjakjdfklajdfal;kdjfa;lfjka`. The page is titled "UpaayaX". On the left, there are three filter boxes: "Category" (dropdown menu showing "All Categories"), "Price Range" (two input fields for "Min Price" and "Max Price"), and "Your Location" (text field showing "Latitude: 26.6619, Longitude: 87.2741" with a "Show Map" button). To the right, under "Available Services", it says "No services available." There are also navigation links for "Home", "Home Services" (which is highlighted in green), and "Find a Room". At the top right are "Sign In" and "Register" buttons, along with social media sharing icons.

Price range accepts: Only Positive values

Your location filters work on this flow:

If the user had on the geolocation / allow their location then it gets that user location and based on that location the user can search the services till 50km radius .

For more convenient if the service provider is available on that 50km radius then it will show that to the user. BUT

Suppose the service provider on itahari had said that I am in itahari I can give services to (10km radius of my location) and the service seeker suppose is from Urlabari and he/she had used the search filter based on that location, then on 50km radius itahari exists as the distance between itahari to urlabari is 35km (approx) then that service provider from itahari is also shown. But the itahari service provider had mentioned he can only work within the radius of 10km. So, now the service seeker can't book this service.

I will show you the flow for it as well.

Let's be a service provider for now but before that let's understand the roles of our system

## 3 Roles of the system

Our system contains four roles: 1. Service\_seeker, 2. Provider, 3. Gharbeti 4. Admin

```
5 rows in set (0.020 sec)

mysql> select * from roles;
+----+-----+-----+-----+
| id | name      | createdAt        | updatedAt        | deletedAt      |
+----+-----+-----+-----+
| 1  | Admin     | 2025-11-04 04:40:27 | 2025-11-04 04:40:27 | NULL           |
| 2  | Service_provider | 2025-11-04 04:40:27 | 2025-11-04 04:40:27 | NULL           |
| 3  | Service_seeker   | 2025-11-04 04:40:27 | 2025-11-04 04:40:27 | NULL           |
| 4  | Gharbeti    | 2025-11-04 04:40:27 | 2025-11-04 04:40:27 | NULL           |
+----+-----+-----+-----+
4 rows in set (0.003 sec)

mysql> █
```

A normal user can have many roles, even an admin can be a provider, and gharbeti

When the normal user is registered then that user is registered as service-seeker

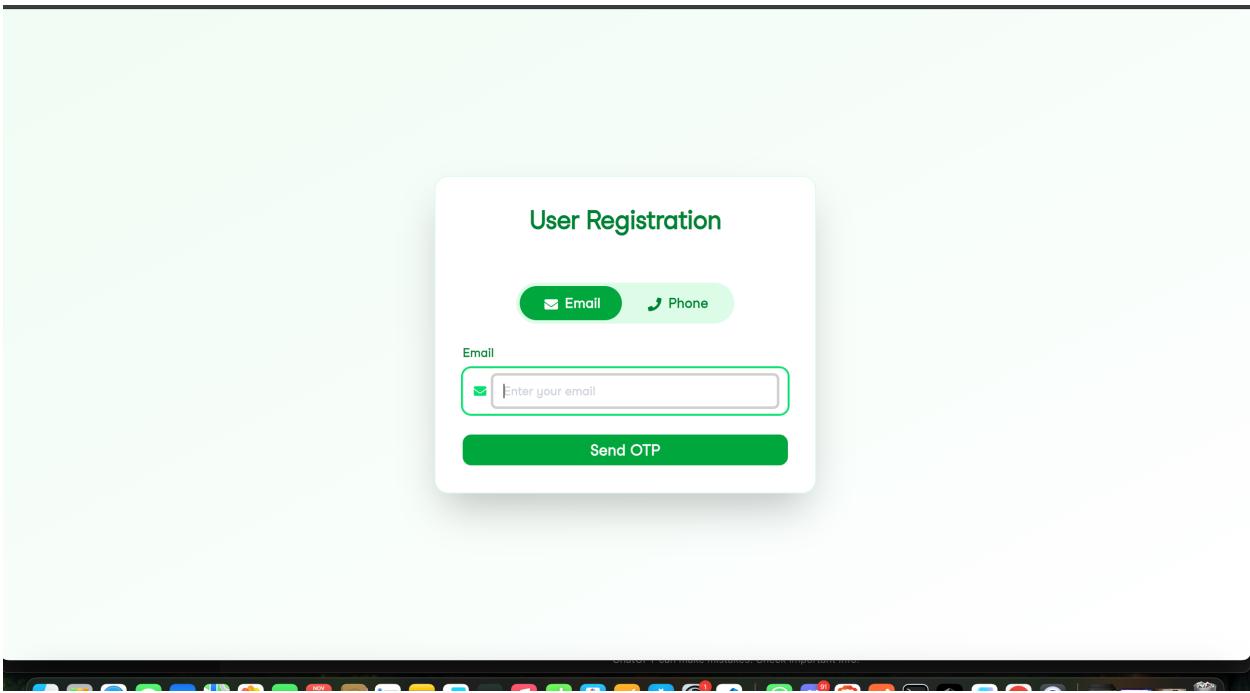
Let's do the registration process for now

## 4 Registering as a User and then getting other roles flow

At first you need to create the account for getting any other roles, that is either service provider or gharbeti.

Click on register button on home page

The registration Page looks like this



**Note: Currently the Phone is not implemented**

Email: Enter the valid email: a valid email contains @ and (dot)

# User Registration

Email

Phone

Email



raghavdahal686@gmail.com

Send OTP

**Note: The validations are shown in the below section topic Validation of system**

Send OTP

## User Registration

Email

Phone

Email

 raghavdahal686@gmail.com

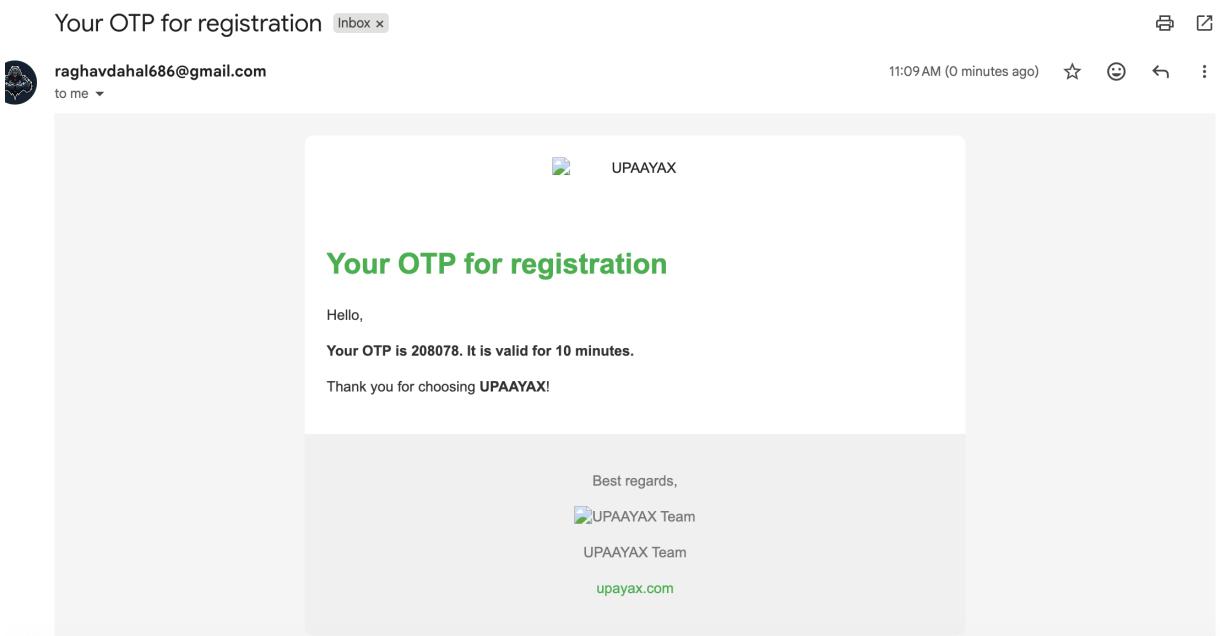
Enter OTP

 Enter OTP

Verify OTP

**Sometimes the OTP takes long time to being sent**

In gmail, the oTP message is sent like this



Enter the OTP in that OTP field and verify-otp after that you will be redirected to this section

# User Registration

Password



Enter your password

Primary Location

Select Province

Select District

Select Municipality

Secondary Location (optional)

Select Province

Select District

Select Municipality

Name



Enter your name

Register

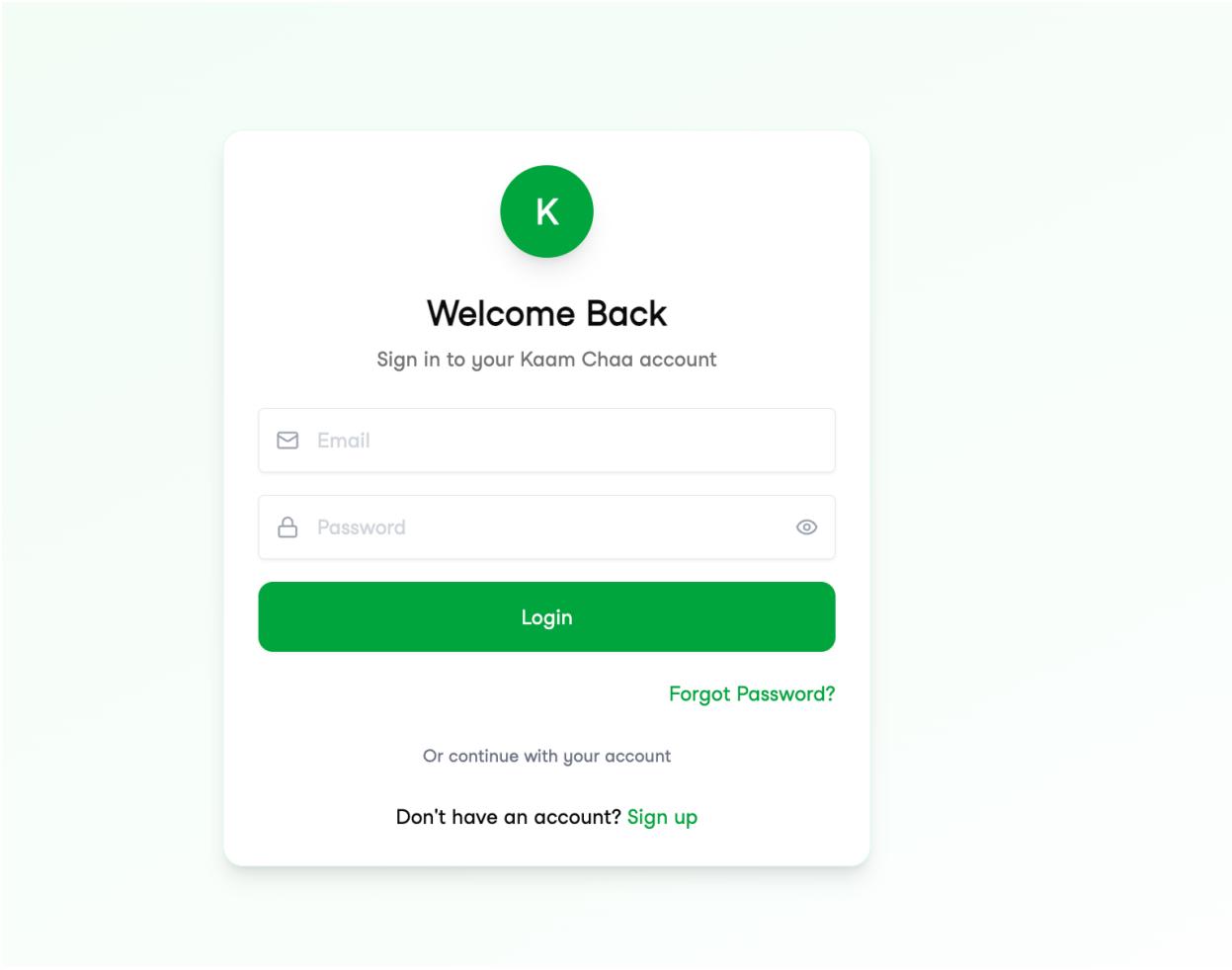
Google

Here: the fields it expects are:

1. Password: (at least 8 characters)
2. Primary Location: (compulsory)
3. Secondary Location: (Optional)
4. Name: (Compulsory) (no @ and (dot) less than 50 characters)

**After registration you are redirected to login Page:**

## 5 Login Page:

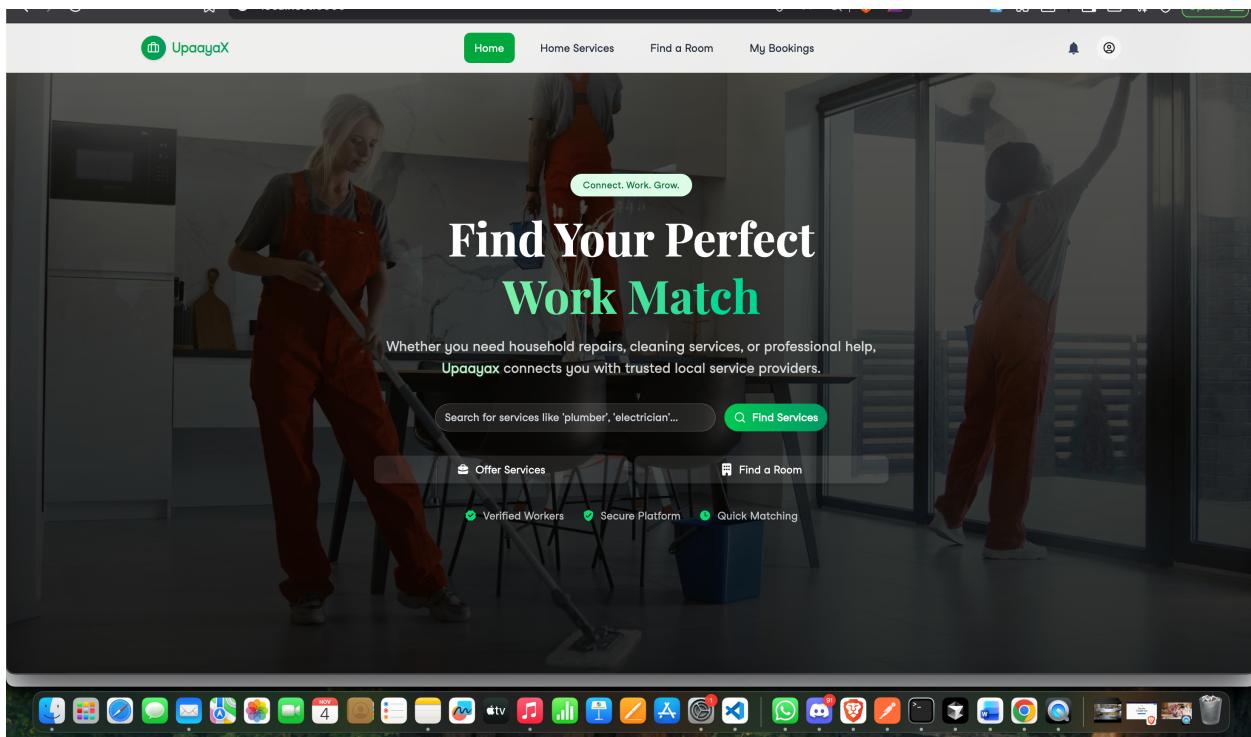


Enter your credentials and login

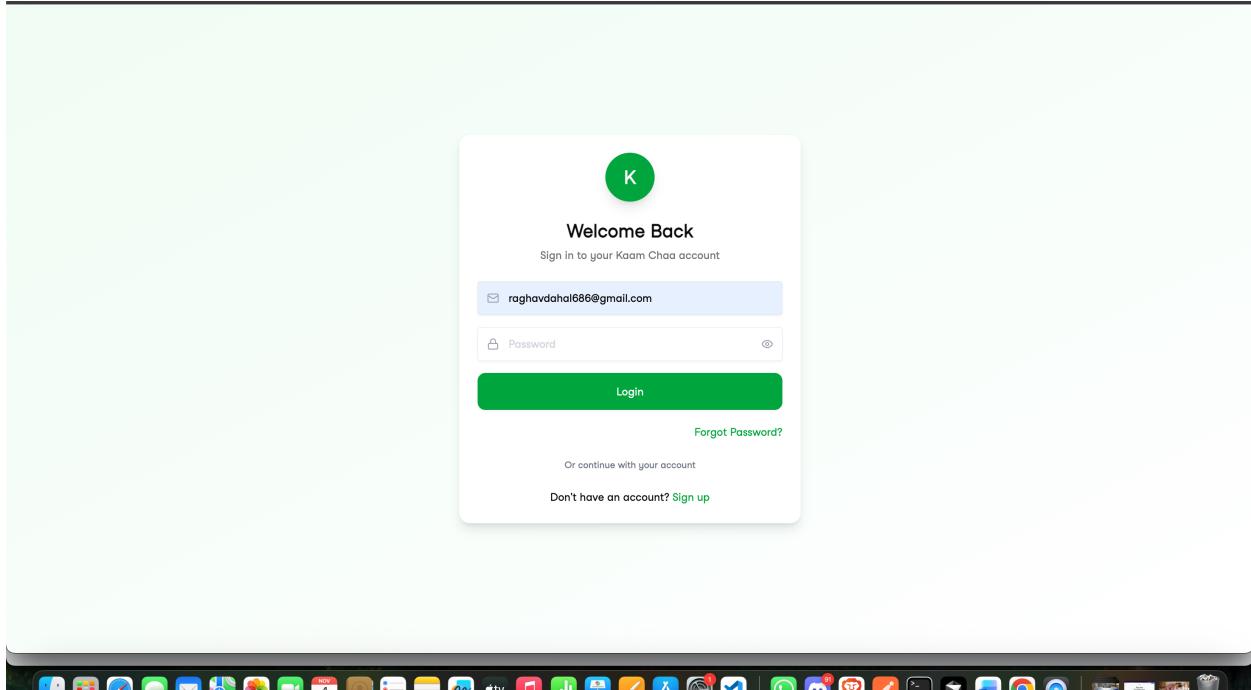
After Login you are redirected to Home page

**Technicality: After login you will get the token, send that token to headers: authorization: Bearer {token} for accessing the protected routes.**

**So, for that make sure to store the token in your state or local storage.**



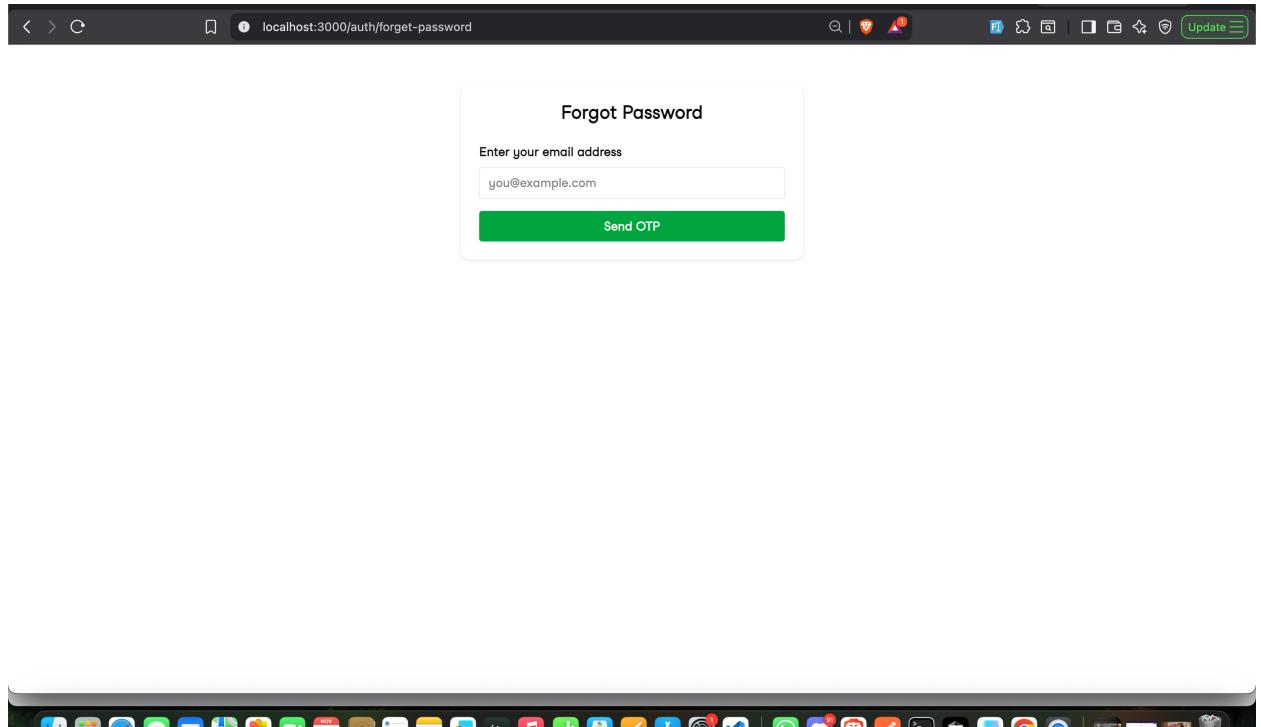
## 5.1 Forget Password



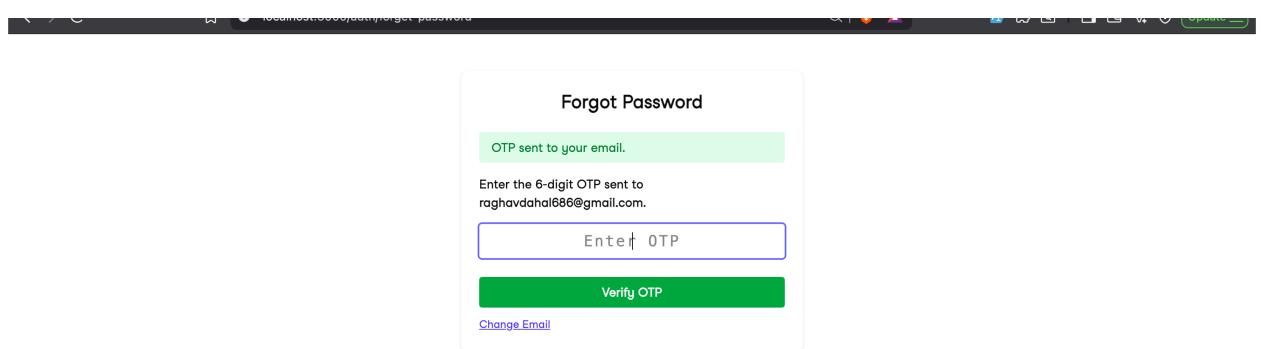
Click on forget Password:

After that you are redirected to this page

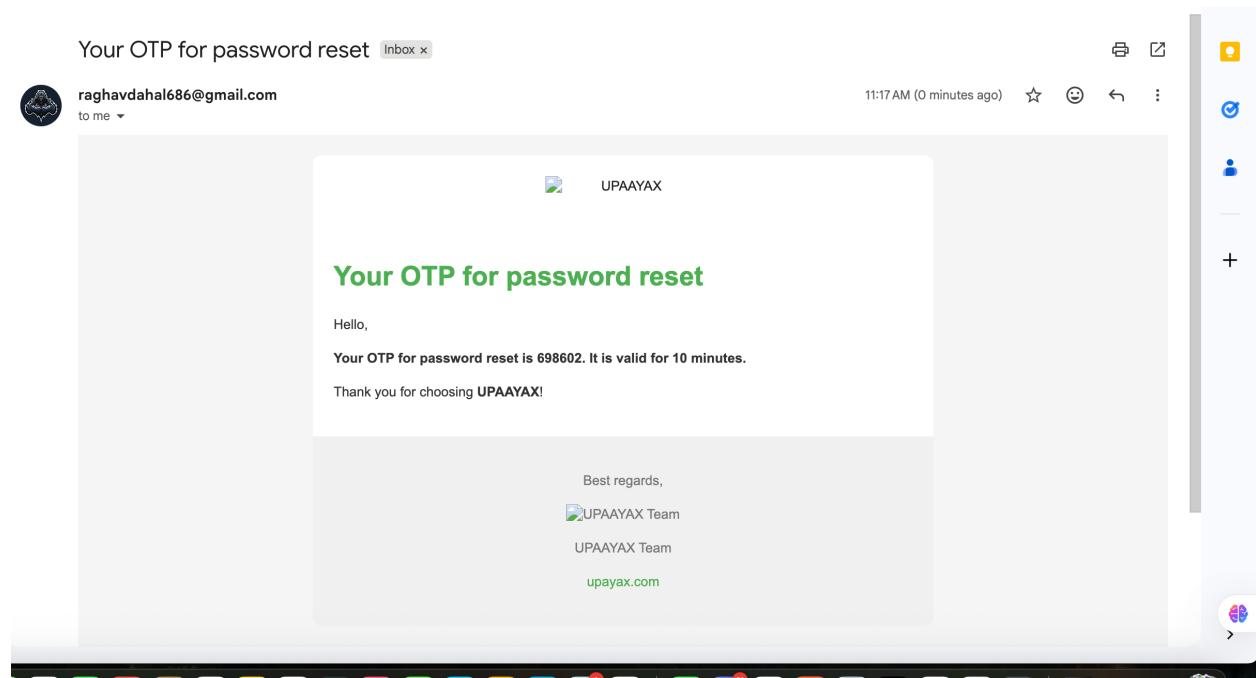
Enter the valid email and send OTP



Once the otp is sent you are redirected to this page

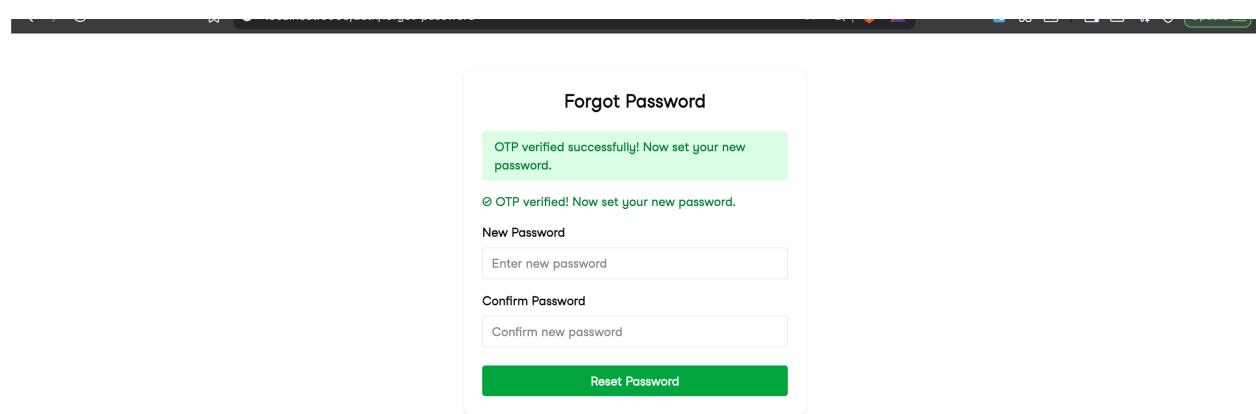


Check your mail:



**Note: Email service takes time sometimes**

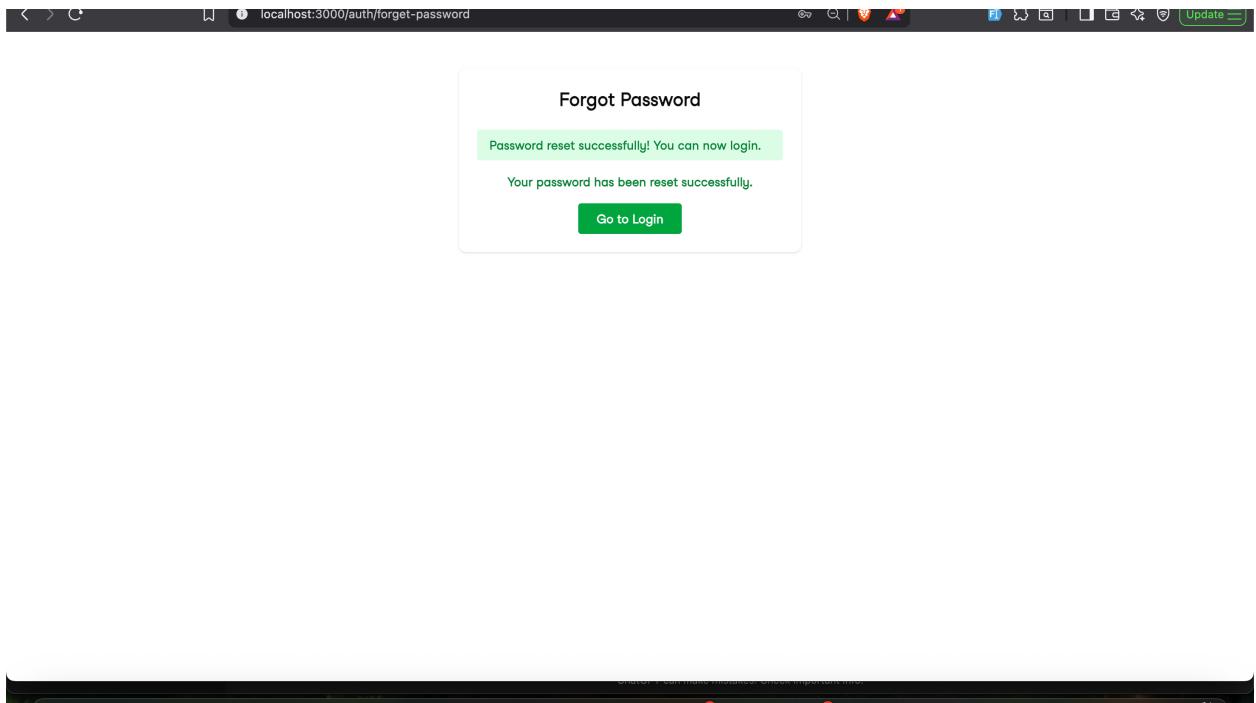
After that OTP verification



You are redirected to this page

The password should be (at least 8 characters)

Once the password is reset

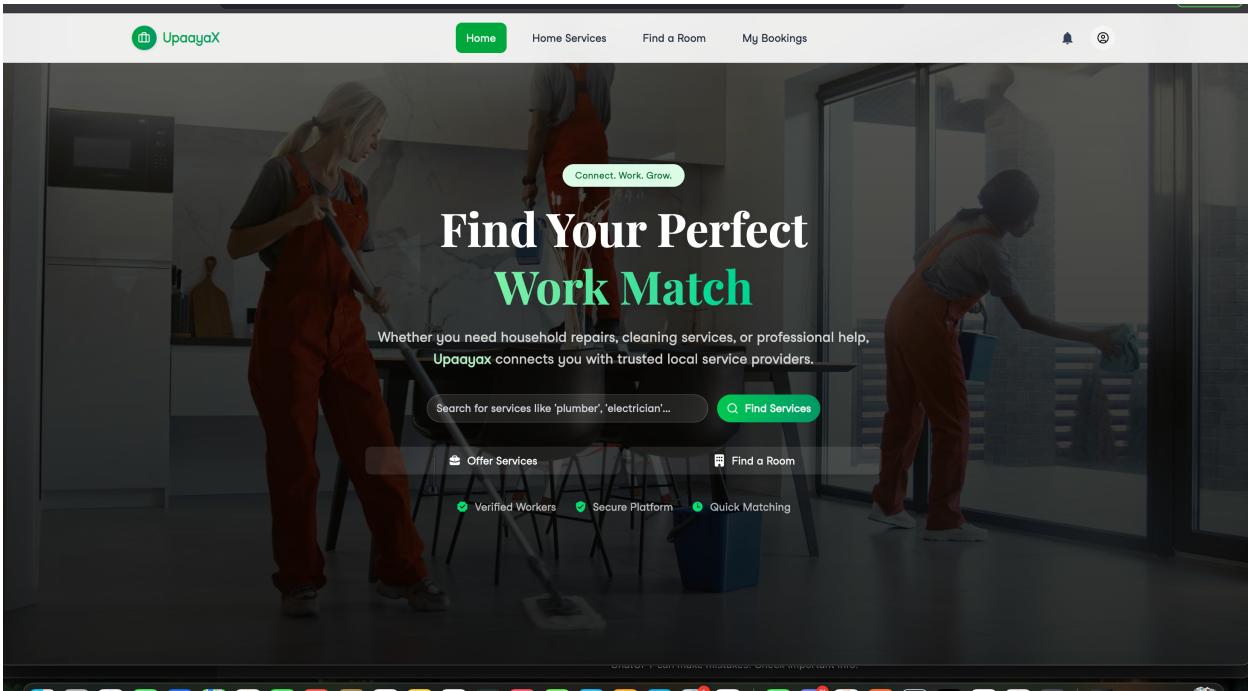


You are redirected to this page and can login from here

For login guide check

5. Login Page

After login you are redirected to home page

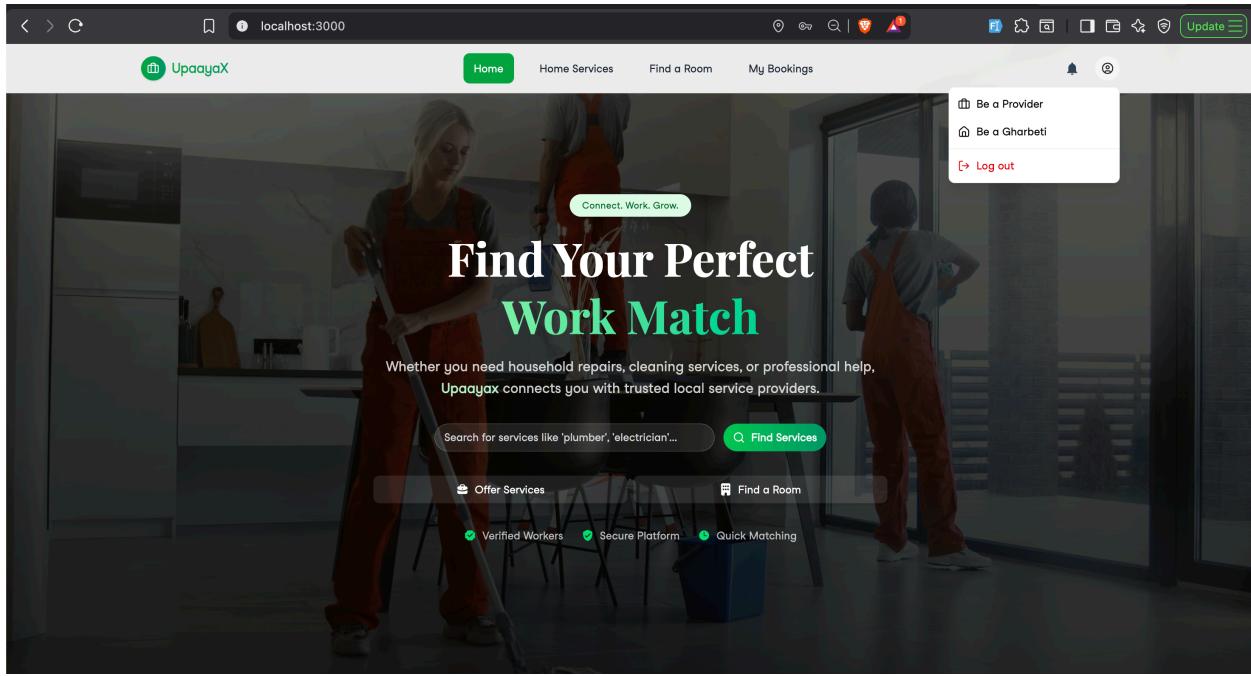


## 6 Getting role service Provider

For getting the roles like service provider and gharbeti at first you must have to be register as a normal user or create an account just like

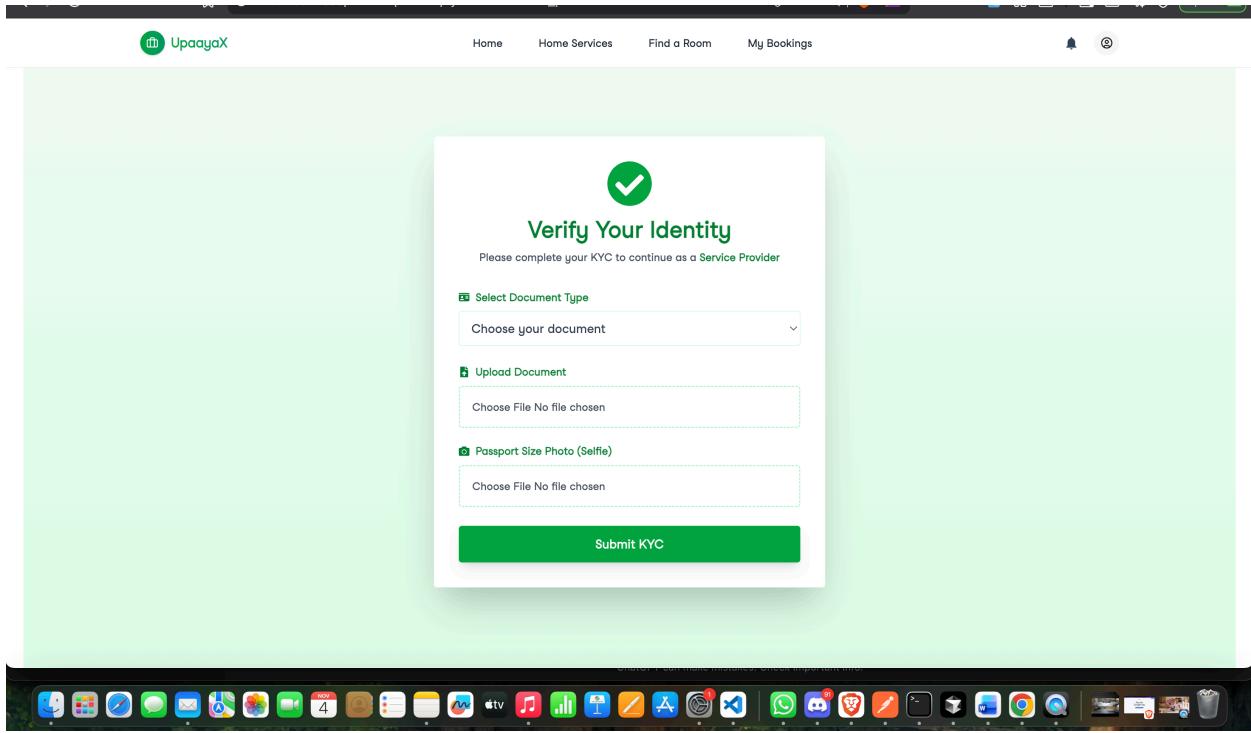
Heading : 4

On the right side of user icon



You get the options for be a provider and be a gharbeti

In this section we will be a provider, for gharbeti please find **the table of contents at top and navigate their accordingly**



**To get access to provider dashboard, you must be a provider at first how, does this work?**

When you login: The backend calls this API for profile

<http://localhost:5000/api/users/profile>

The screenshot shows the Postman interface with a dark theme. At the top, there's a navigation bar with various API endpoints listed. Below the navigation is a header bar with the URL "upaayax / aboutuser" and several action buttons like "Save", "Share", and "Send". The main area contains a "GET" request for the endpoint "{{baseURL}} /api/users/profile". Underneath the request, there are tabs for "Params", "Auth", "Headers (8)", "Body", "Scripts", and "Settings". The "Params" tab is currently selected. A table titled "Query Params" is shown, with columns for "Key", "Value", and "Description". The "Body" tab is also visible, showing a JSON response with a status of "401 Unauthorized". The response body is a JSON object with the following content:

```
1  {
2   "status": "error",
3   "code": 401,
4   "message": "No access token provided. Please log in."
5 }
```

At the bottom of the interface, there are several utility buttons: "Runner", "Start Proxy", "Cookies", "Vault", "Trash", and others.

if no token is provided it gives the error for no access token provided for this you must send the authorization in headers with the value as Bearer token

Once you login

The screenshot shows the Postman application interface. At the top, the URL is set to `http://localhost:5000/api/users/login`. The method is selected as `POST`. The `Body` tab is active, showing the following JSON payload:

```
1 {
2   "email": "raghavdahal686@gmail.com",
3   "password": "raghav@gmail.com"
4 }
```

Below the request, the response is displayed. The status is `200 OK`, and the response body is:

```
1 {
2   "status": "success",
3   "code": 200,
4   "message": "Login successful.",
5   "data": {
6     "user": {
7       "id": 3,
8       "username": "raghavdahal686841",
9       "refresh_token": null,
10      "email": "raghavdahal686@gmail.com",
11      "phone_number": null,
12      "profile_picture": "/images/default-profile.jpg",
13      "is_active": true,
14      ...
15    }
16  }
17}
```

It sends this response at the end there exists access token and refresh token

The screenshot shows the Postman interface with the following details:

- Request URL:** upaayax / loginuser
- Method:** POST
- Endpoint:** http://localhost:5000/api/users/login
- Body (JSON):**

```
1 {
2   "email": "raghavdahal686@gmail.com",
3   "password": "raghav@gmail.com"
4 }
```
- Response Status:** 200 OK
- Response Time:** 91 ms
- Response Size:** 1.31 KB
- Response Content:** A JSON object containing user information and tokens. The token is a long string starting with "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9".

Copy this token: and paste on that profile api

Like this:

HTTP upaayax / aboutuser

Save Share ⌂

GET {{baseURL}} /api/users/profile Send

Params Auth Headers (9) Body Scripts Settings Cookies </>

Headers 8 hidden

Key	Value	Desc...	Bulk Edit	Presets
Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5c...			
Key	Value	Description		

Body 200 OK • 24 ms • 1.09 KB • Save Response

{ } JSON ▾ Preview Visualize

```
1 {  
2   "status": "success",  
3   "code": 200,  
4   "message": "User profile fetched successfully.",  
5   "data": {  
6     "user": [  
7       "id": 3,  
8       "username": "raghavdahal686841",  
9       "refresh_token": null,  
10      "email": "raghavdahal686@gmail.com",  
11      "phone_number": null,  
12      "profile_picture": "/images/default-profile.jpg",  
13      "is_active": true,  
14    ]  
15  }  
16}
```

Runner Start Proxy Cookies Vault Trash ?

The screenshot shows the Postman application interface with a dark theme. At the top, there are tabs for 'Body', 'Cookies (1)', 'Headers (10)', 'Test Results', and a settings gear icon. To the right of these are status indicators: '200 OK', '24 ms', '1.09 KB', and a save response button. Below the tabs is a dropdown menu set to 'JSON'. There are three preview options: 'Preview' (selected), 'Visualize', and 'Raw'. The main area displays a JSON response with line numbers from 1 to 25. The response object has fields: 'status': 'success', 'code': 200, 'message': 'User profile fetched successfully.', and 'data': { ... }. The 'data' field contains a nested 'user' object with various properties like id, username, email, etc., and a large JWT token string. On the far right, there are several small icons for actions like copy, search, and refresh.

```
1 "status": "success",
2 "code": 200,
3 "message": "User profile fetched successfully.",
4 "data": {
5     "user": {
6         "id": 3,
7         "username": "raghavdahal686841",
8         "refresh_token": null,
9         "email": "raghavdahal686@gmail.com",
10        "phone_number": null,
11        "profile_picture": "/images/default-profile.jpg",
12        "is_active": true,
13        "name": "raghavdahal",
14        "municipal_code": 11208,
15        "blocked_reason": null,
16        "createdAt": "2025-11-04T05:29:15.000Z",
17        "updatedAt": "2025-11-04T05:35:22.000Z",
18        "deletedAt": null
19    },
20    "service_provider_id": null,
21    "gharbeti_id": null,
22    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  
eyJpZCI6MywiZW1hawvioiJyWdoYXZkYWhhbDY4NkBnbWFpbC5jb20iLCJyb2xlcI6W10sIm5hbWUiOjJyYWdoYXZkYWhhbCIIsInVzzXJuYW1lIjoicm  
FnaGF2ZGFoYw2ODY4NDEiLCJpXXQiojE3NjIyMzUz0dgsImV4ccCI6MTc2MjIz0dk40H0.ghyKF6069jJb1F9B3bMih0L2v6yMQ9NGtWt5iy-nM2I"
23
24 }
25 }
```

This is the full response of the endpoint

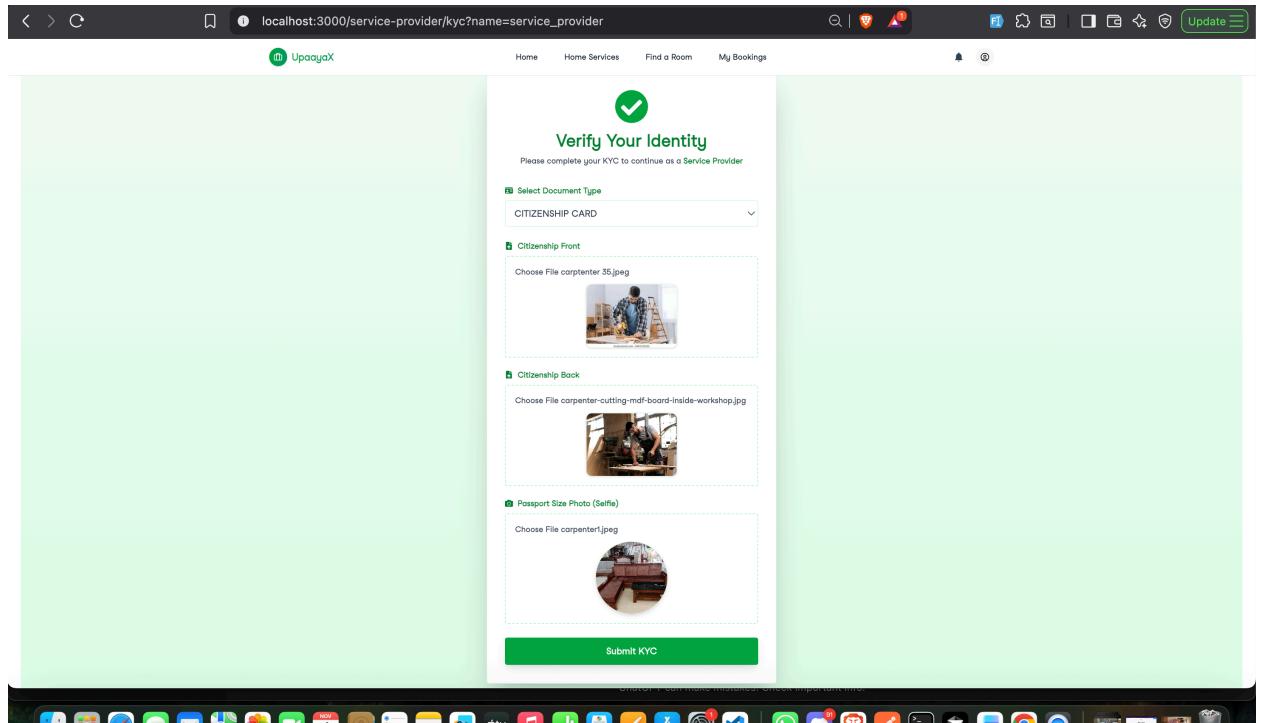
`{{baseUrl}}/api/users/profile`

**Note:** Base Url: your backend url

Based on this response when the user clicks on provider it decides whether to send the user to provider-dashboard or to register as provider (KYC part)

Notice, `service_provider_id: null`?

This decides the actual flow for provider to navigate into provider dashboard or register as provider (kyc part)



## Kyc details: verification part

Document type: (data is coming from backend)

The screenshot shows a Postman request for the endpoint `upayak / documenttype`. The method is `GET`, and the URL is `{{baseURL}} /api/kyc/document-type?`. The `Headers` tab is selected, showing a table with one row for `Authorization` containing the value `Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...`. The `Body` tab is selected, showing a JSON response with status `success`, code `200`, message `"Document types fetched successfully"`, and data containing four types: `passport`, `national_id`, `driver_license`, and `citizenship_card`.

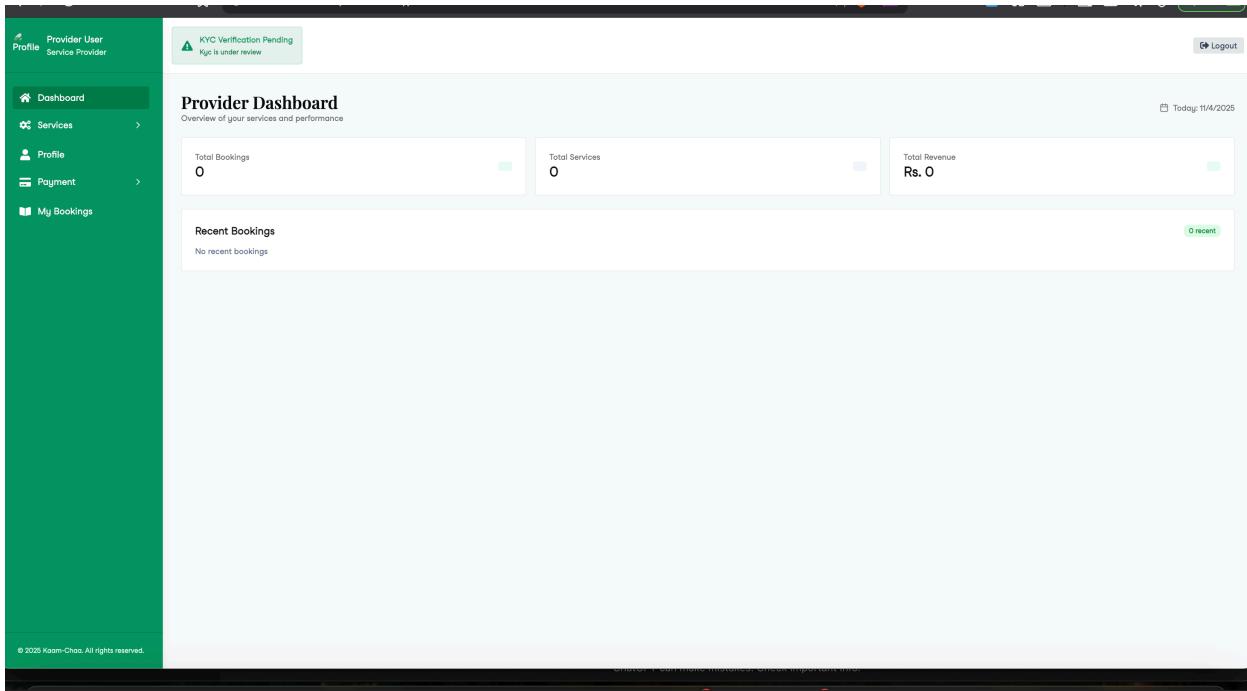
```
1
2   "status": "success",
3   "code": 200,
4   "message": "Document types fetched successfully",
5   "data": {
6     "type": [
7       "passport",
8       "national_id",
9       "driver_license",
10      "citizenship_card"
11    ]
12  }
```

The endpoint for this api is:

[{{baseURL}}/api/kyc/document-type?](#)

You must include authorization in headers as well

Once you submit the kyc it redirects you to provider dashboard page



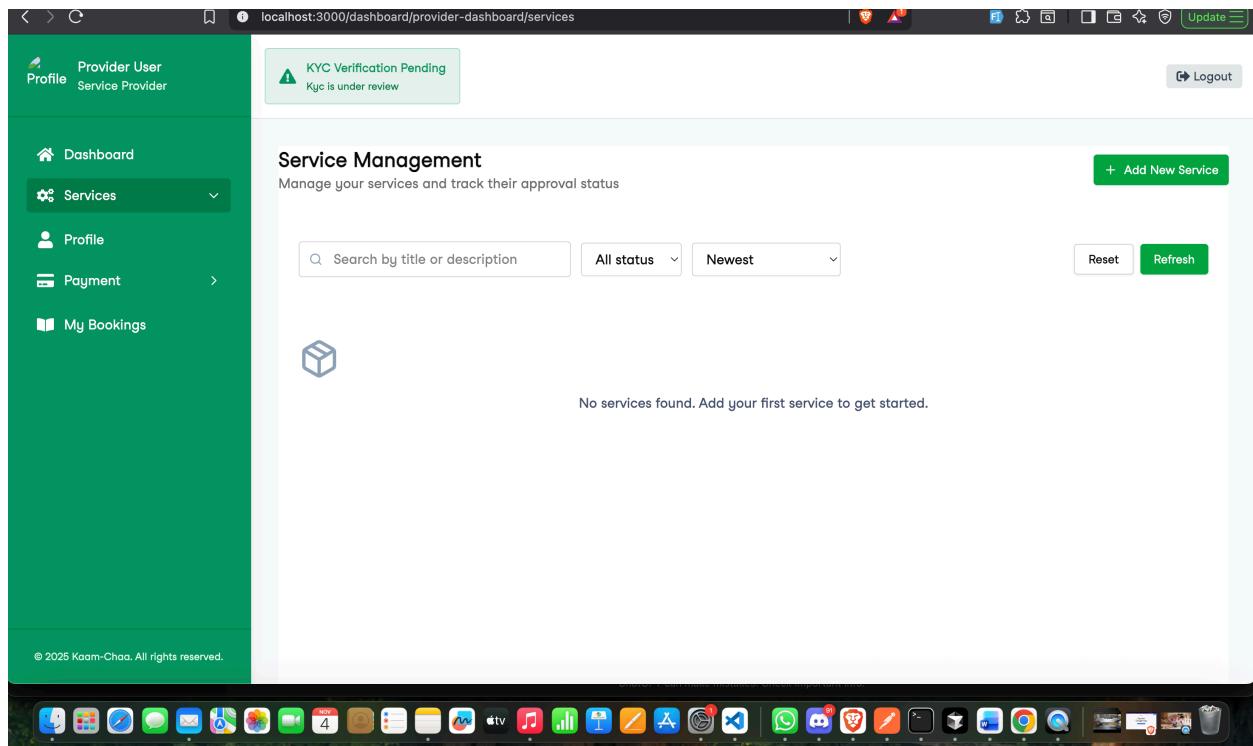
The dashboard looks like this

#### Another things to consider:

**Client said, just after the kyc submission the user should see the dashboard, we shouldn't make them wait for kyc verification. But when the user adds the services, then in that case, the services are sent as pending. Admin needs to verify this. For admin part again see the references at table of contents.**

## 6.1 Adding the services as a service provider

Navigate to services section and click on add services option



Click on add new services, on clicking on add new services will give you this page

For this page, the fields values are:

1. Category: (compulsory)
2. Description:(Compulsory)
3. Rate: (Compulsory)
4. Service Location (Compulsory)
5. Upload documents (Optional)
6. Photos (compulsory at least 1)
7. Weekly Schedule (Compulsory at least one day one schedule, schedule from and to must be greater than 1 hour)
8. What this service includes: (Optional)
9. Notes: (optional)

**Tip: For testing Purpose use fake filler extension.**

User Profile

KYC Verification Pending

Add New Service

Fill in all details for your service, including schedule, packages, and documents.

**Category**

Select category ▾

**Description (max 250 words)**

0 of 250 words

**Rate (NPR)/hour**

**Service Locations (Radius from your current location)**

Select Location

**Upload Documents (optional)**

Click to upload or drag and drop

**Photos (required, up to 8)**

All rights reserved.

**Weekly Schedule**

Day	Status	Action
Monday	<input type="checkbox"/> Available	Apply to all
Tuesday	<input type="checkbox"/> Available	Apply to all
Wednesday	<input type="checkbox"/> Available	Apply to all
Thursday	<input type="checkbox"/> Available	Apply to all
Friday	<input type="checkbox"/> Available	Apply to all
Saturday	<input type="checkbox"/> Available	Apply to all
Sunday	<input type="checkbox"/> Available	Apply to all

Logout

Add New Service

Refresh

### For service location:

**Clarity: When you click on select your service radius, the logic behind the select your service radius is:**

Your preferred location from your current location. How much radius area are you available for getting booked? MAX: 50km MIN:5KM

ce

your s

d

50 wor

d no

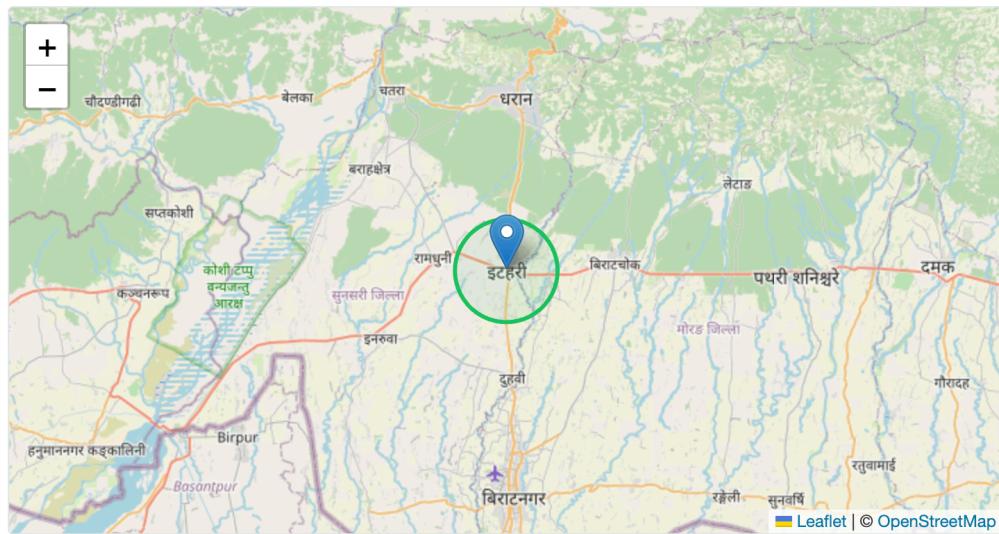
Radius

Lat 26.

: (option

C

## Select Your Service Area



X

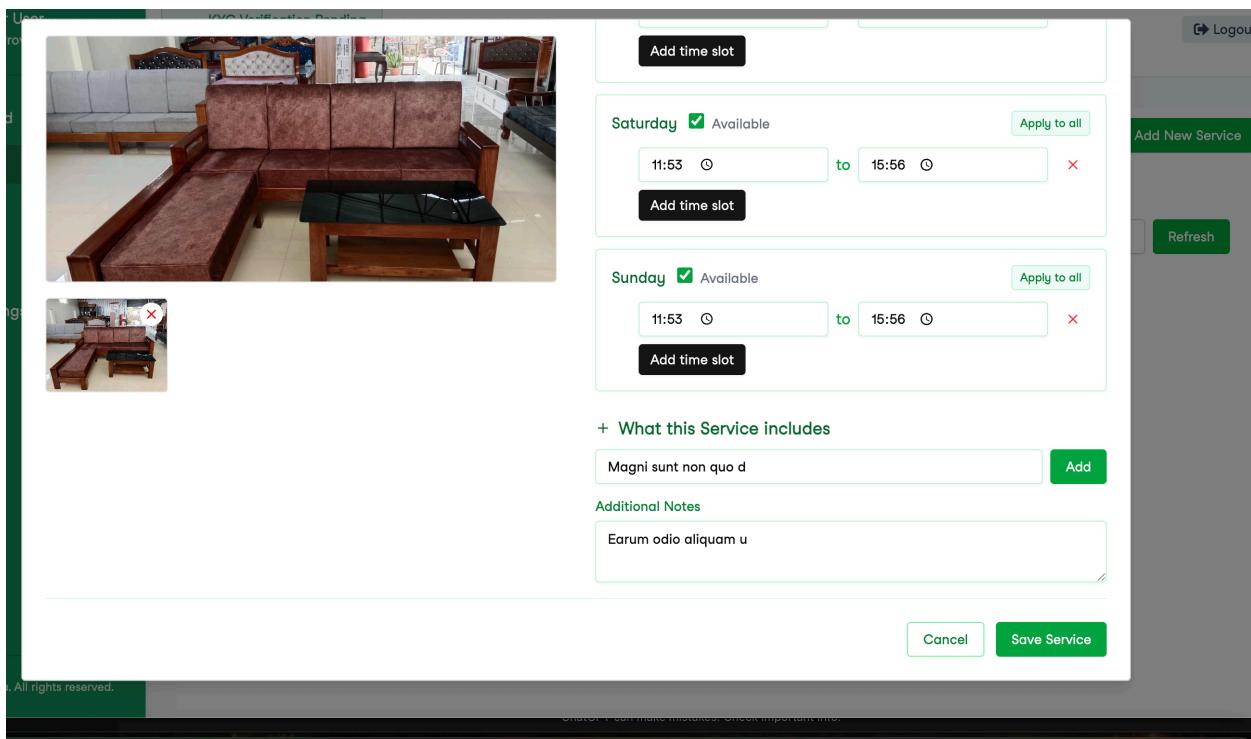
Service Radius

5 km

Save Location

Thursday  Available

After all the data is filed click on save service



This screenshot shows the provider user dashboard. The sidebar on the left includes links for Dashboard, Services (selected), Profile, Payment, and My Bookings. A 'KYC Verification Pending' badge indicates that KYC is under review. The main content area is titled 'Service Management' and shows a list of services. One service, 'Appliance Repair', is listed with a thumbnail image of a sofa set, a price of 'Rs. 42/hr', and status indicators 'pending' and 'Active'. The dashboard also features a search bar, filters for 'All status' and 'Newest', and buttons for 'Reset' and 'Refresh'. At the bottom, there is a standard Mac OS X dock with various application icons.

TO add the service:

The api is:

`{{baseURL}}/api/services/add`

**The api expects the form data make sure to send it as form Data as the api expects the images and files as well**

POST  `{{baseURL}} /api/services/add` Send

Params Authorization Headers (11) Body Scripts Settings Cookies </> ...

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL

Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/> serviceId	Text <input type="button" value="5"/>			
<input checked="" type="checkbox"/> description	Text <input type="button" value="djflafj"/>			
<input checked="" type="checkbox"/> rate	Text <input type="button" value="2300"/>			
<input checked="" type="checkbox"/> notes	Text <input type="button" value="fjadlfa"/>			
<input checked="" type="checkbox"/> locations	Text <input latitude\":26.661945977363402,\"longitude\":87.27402665645185}]"="" type="button" value="[{\"/>			
<input checked="" type="checkbox"/> schedules	Text <input \"15:07\",="" \"16:07\"}]"="" \"end_time\":="" \"monday\",="" \"start_time\":="" day_of_week\":="" type="button" value="[{\"/>			
<input checked="" type="checkbox"/> images	File <input type="button" value="Black and White Simple Minimalist Job..."/> <input type="button" value="Black and White Simple Minimalist Job..."/>			
<input checked="" type="checkbox"/> documents	File <input type="button" value="Black and White Simple Minimalist Job..."/> <input type="button" value="Black and White Simple Minimalist Job..."/>			
Key	Value	Description	...	

Body Cookies (1) Headers (10) Test Results | ... 201 Created • 25 ms • 2.12 KB • ... Save Response • ...

The data that api expects are:

Service Id: number

Description: upto 250 words

Rate: number

Notes: (string, upto 250 words)

Locations: (an array)

Example:

locations:

`[{"latitude":26.661945977363402,"longitude":87.27402665645185,"radius":30}]`

Schedules: JSON

Example: schedules: `[{"day_of_week": "Monday", "start_time": "15:07", "end_time": "16:07"}, {"day_of_week": "Monday", "start_time": "16:07", "end_time": "17:07"}, {"day_of_week": "Monday", "start_time": "17:07", "end_time": "18:07"}]`

Images: (multiple upto 8): data type: File

Documents (multiple): data type: File

Note: The service provider can deactivate their services, but only after the service is accepted

For deactivate:

API: <http://localhost:5000/api/services/toggle-active-status/1>

Here the /1 is the serviceproviderserviceid the params parameter

The HTTP method is PATCH

The payload is

{is\_active: true}

## POSTMAN

The screenshot shows the Postman application interface. At the top, the URL is set to `http://localhost:5000/api/services/toggle-active-status/1`. The method is selected as `PATCH`. The `Body` tab is active, showing the raw JSON payload:

```
1
2   "is_active":true
3 }
```

Below the body, the response is displayed under the `Test Results` tab. The response code is `200 OK`, with a duration of `10 ms` and a size of `817 B`. The response body is shown in JSON format:15 "note": "Earum odio aliquam u",
16 "reappliedAt": null,
17 "is\_active": true,
18 "createdAt": "2025-11-04T06:16:08.000Z",
19 "updatedAt": "2025-11-04T09:27:35.000Z",
20 "deletedAt": null,
21 "serviceProviderId": 1,
22 "serviceId": 6
23 }
24 }
25 }

## 7 Profile photo change

For profile photo change and update:

The screenshot shows a web browser window for 'localhost:3000/dashboard/provider-dashboard/profile'. The left sidebar is green and contains navigation links: Dashboard, Services, Profile (which is highlighted), Payment, and My Bookings. The main content area has a title 'Profile' and displays a placeholder image of a person working at a workshop bench. Below the image, the name 'raghavdahal' and email 'raghavdahal686@gmail.com' are shown, along with a 'Change photo' button. To the right, there are two sections: 'User Details' and 'Location Information'. The 'User Details' section includes fields for Username (raghavdahal686841), Email (raghavdahal686@gmail.com), Phone (N/A), Active (Yes), and Joined (11/4/2025). The 'Location Information' section includes fields for Municipality (Uttarbari Urban Municipality (उत्तर्बारी नगरपालिका)), District (Morang (मोरङ)), and Province (Koshi Pradesh (कोशी प्रदेश)). The bottom of the screen shows a Mac OS X-style dock with various application icons.

User Details	
Username	raghavdahal686841
Email	raghavdahal686@gmail.com
Phone	N/A
Active	Yes
Joined	11/4/2025

Location Information	
Municipality	Uttarbari Urban Municipality (उत्तर्बारी नगरपालिका)
District	Morang (मोरङ)
Province	Koshi Pradesh (कोशी प्रदेश)

For changing the profile Photo:

Hit the api:

<http://localhost:5000/api/users/update-profile>

The screenshot shows the Postman application interface. At the top, there's a header bar with the URL `http://localhost:5000/api/users/update-profile`. Below the header, the method is set to `POST`. The main body of the interface is divided into sections: `Params`, `Authorization`, `Headers (10)`, `Body` (which is currently selected), `Scripts`, and `Settings`. Under the `Body` section, the `form-data` option is selected. A table is present with two rows. The first row has a key `profile_image` and a value that is a file named `Photo on 02-10-2025 at 12.15 #2.jpeg`. The second row has a key `Key` and a value `Value`. On the right side of the interface, there are various toolbars and a sidebar with icons like `</>`, `?`, and `?`.

At the bottom, there's a `Response` tab and a history section. A small illustration of a character with a rocket on their back is centered at the bottom. Below it, the text `Click Send to get a response` is displayed. Along the bottom edge, there are several utility buttons: `Runner`, `Start Proxy`, `Cookies`, `Vault`, `Trash`, and a help icon.

## 8 For payment account creation provider

For now only esewa is accepted:

Esewa Id: use proper number

And click on save payment methods

The screenshot shows a web browser window for 'KAAM CHAA' on a Mac OS X desktop. The URL in the address bar is 'localhost:3000/dashboard/provider-dashboard/payment'. The left sidebar is green and includes links for 'Provider User', 'Service Provider', 'Dashboard', 'Services', 'Profile', 'Payment' (which is selected), and 'My Bookings'. The main content area has a white background and is titled 'Payment Methods'. It says 'Add or update your payout methods. Keep at least one method to receive payments.' A timestamp '11/4/2025' is in the top right. There are two sections: 'eSewa' and 'Khalti'. Under 'eSewa', it says 'Mobile / ID used to receive payments via eSewa' and 'eSewa ID / Mobile' with the value '9807381017'. Under 'Khalti', it says 'Mobile / ID used to receive payments via Khalti' and 'Khalti ID / Mobile' with the value 'e.g., 9801234567'. Below these are fields for a 'Bank Account': 'Bank' (e.g., Nepal Bank) and 'Account Number' (e.g., 1234567890). To the right is an 'Account Name' field (e.g., John Doe). At the bottom, there's a tip: 'Tip: Keep your payment IDs up-to-date. You can use any combination of payment methods.' followed by 'Reset' and 'Save Payment Methods' buttons.

The screenshot shows the Postman application interface. At the top, there's a navigation bar with various icons and a dropdown menu. Below it, the URL is displayed as `http://upaayax / payment-account-creation`. The main area shows a `POST` request to `http://localhost:5000/api/payments/create-payment-account`. The `Body` tab is selected, showing the raw JSON payload:

```
1 {  
2   "esewa_id": "9807381021"  
3 }
```

Below the request, the response details are shown. It indicates a `201 Created` status with a response time of `41 ms` and a size of `641 B`. The response body is displayed in JSON format:{ } JSON ▾ ▶ Preview ▾ Visualize ▾  
1 {  
2 "status": "success",  
3 "code": 201,  
4 "message": {  
5 "id": 3,  
6 "userId": 3,  
7 "esewaId": "43d97b57-b290-4f37-9924-2665a3d107cf",  
8 "khaltiId": null,  
9 "bankId": null,  
10 "status": "active",  
11 "updatedAt": "2025-11-04T09:46:35.722Z",  
12 "createdAt": "2025-11-04T09:46:35.722Z"  
13 },  
14 "data": "Payment account created successfully"  
15 }

## 9 For booking status in provider part

Api:

The status is the params that accepts:

**Completed**

**Pending**

**Confirmed**

**Cancelled**

**Rejected**

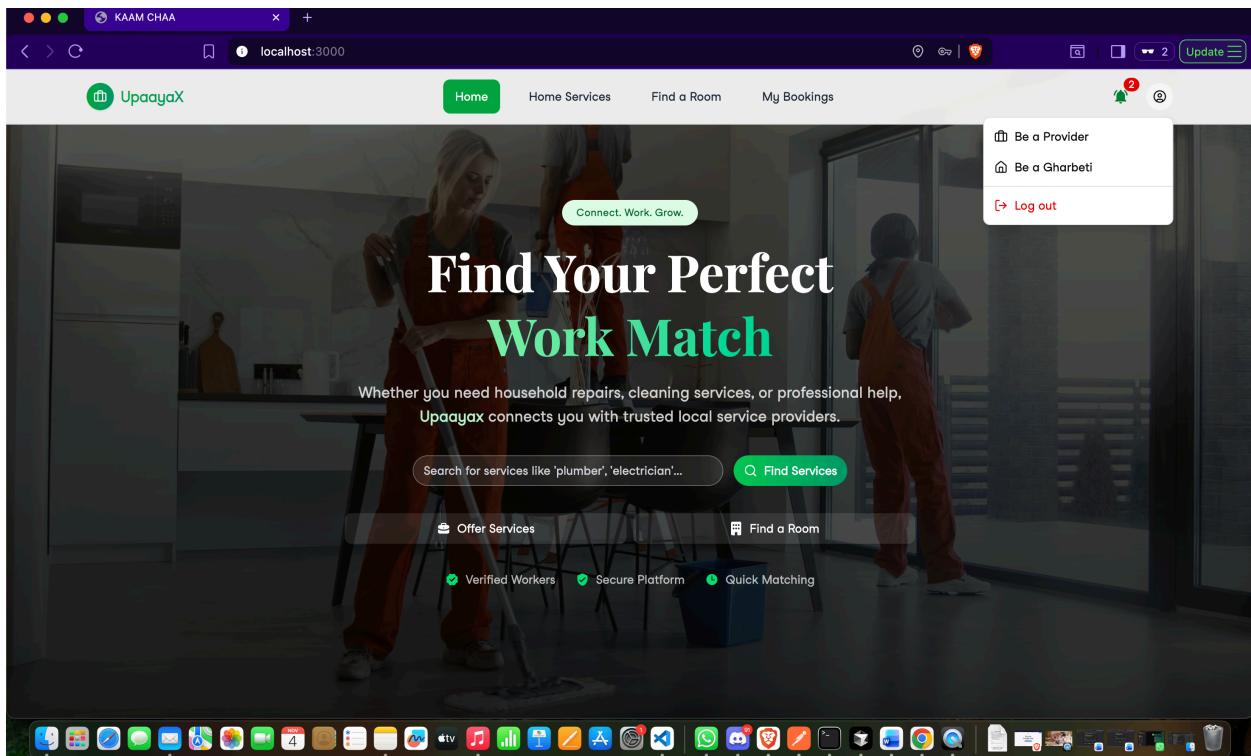
The screenshot shows the Postman application interface. At the top, there's a header bar with various tabs and a dropdown for environments. Below that is a toolbar with 'Save' and 'Share' buttons. The main area has a URL input field set to 'http://localhost:5000/api/booking/provider?status=completed'. The method dropdown is set to 'GET'. Underneath, there are sections for 'Params', 'Authorization', 'Headers (9)', 'Body', 'Scripts', and 'Settings'. The 'Body' tab is selected, showing options for 'none', 'form-data', 'x-www-form-urlencoded', 'raw', 'binary', and 'GraphQL'. A note says 'This request does not have a body'. On the right side, there are sections for 'Cookies' and 'Headers'. At the bottom, the response is displayed. It shows a '200 OK' status with a timestamp of '14 ms' and a size of '1.16 KB'. The 'Body' tab is selected, showing a JSON response with 19 lines of code. The response content is as follows:

```
1 {  
2     "total": 3,  
3     "limit": 10,  
4     "offset": 0,  
5     "results": [  
6         {  
7             "id": 1,  
8             "status": "completed",  
9             "clientCompleted": true,  
10            "providerCompleted": false,  
11            "lat": null,  
12            "lng": null,  
13            "createdAt": "2025-11-04T06:43:00.000Z",  
14            "deletedAt": null,  
15            "userId": 4,  
16            "serviceScheduleId": 34,  
17            "serviceProviderServiceId": 1,  
18            "packageId": null,  
19            "user": {}  
}
```

Below the response, there are buttons for 'Runner', 'Start Proxy', 'Cookies', 'Vault', 'Trash', and a help icon.

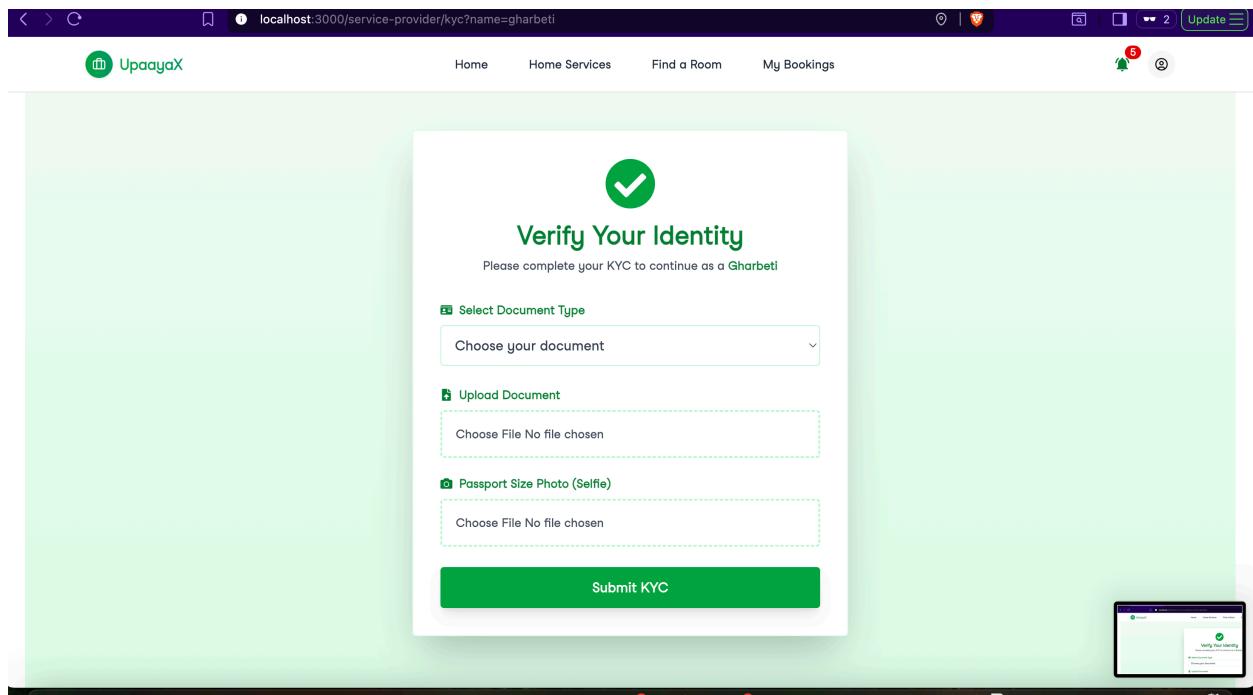
## 10 For gharbeti Role

Click on be a gharbeti from the user icon



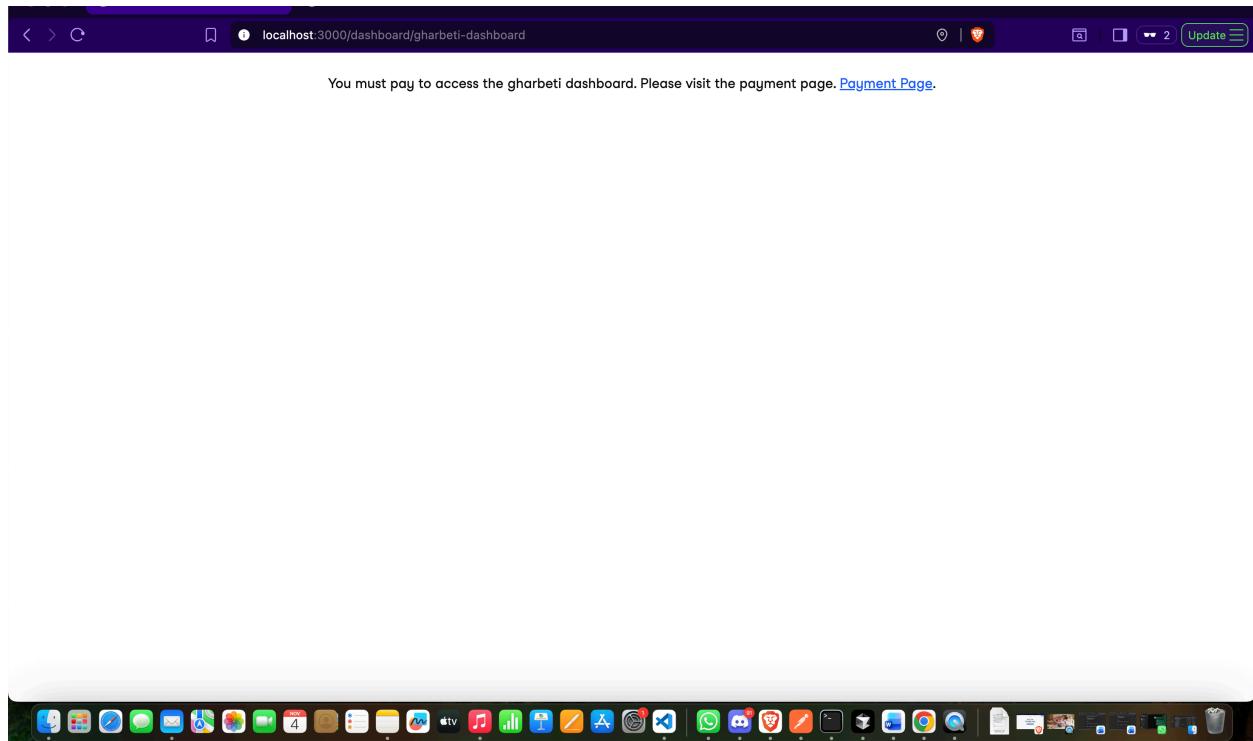
To get the access for gharbeti dashboard

Click on be a gharbeti button.

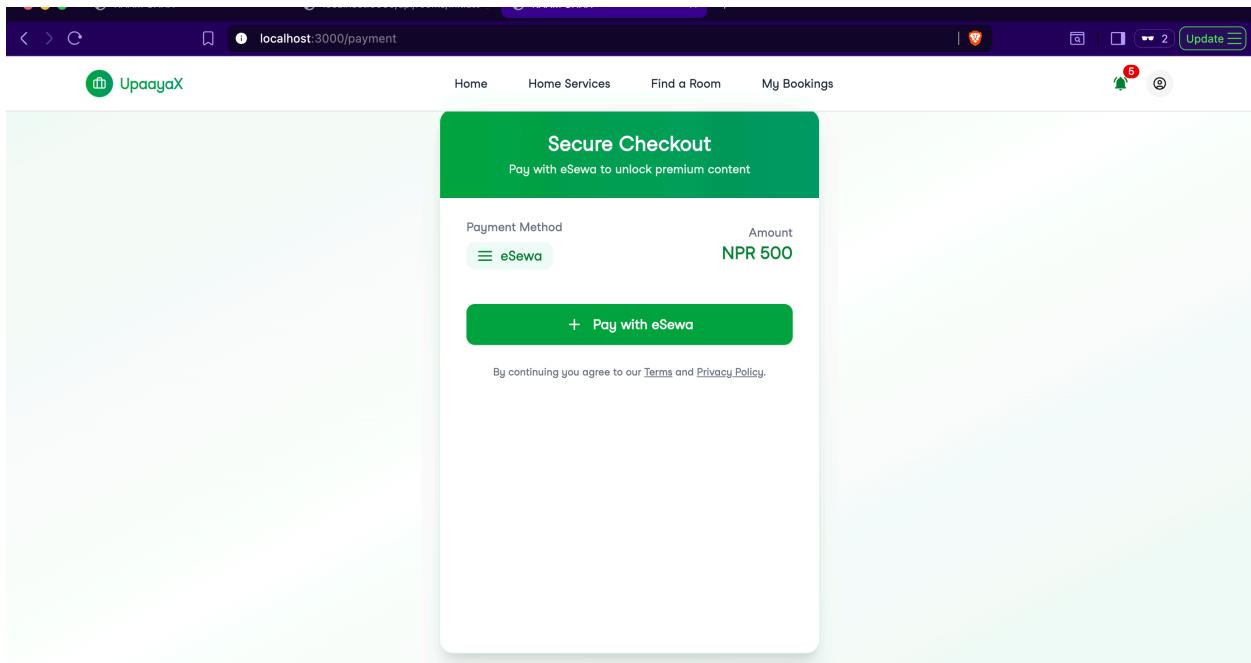


On Click on gharbeti it asks for again the kyc verification for gharbeti

After the kyc submission you must pay the 500 to get the gharbeti dashboard

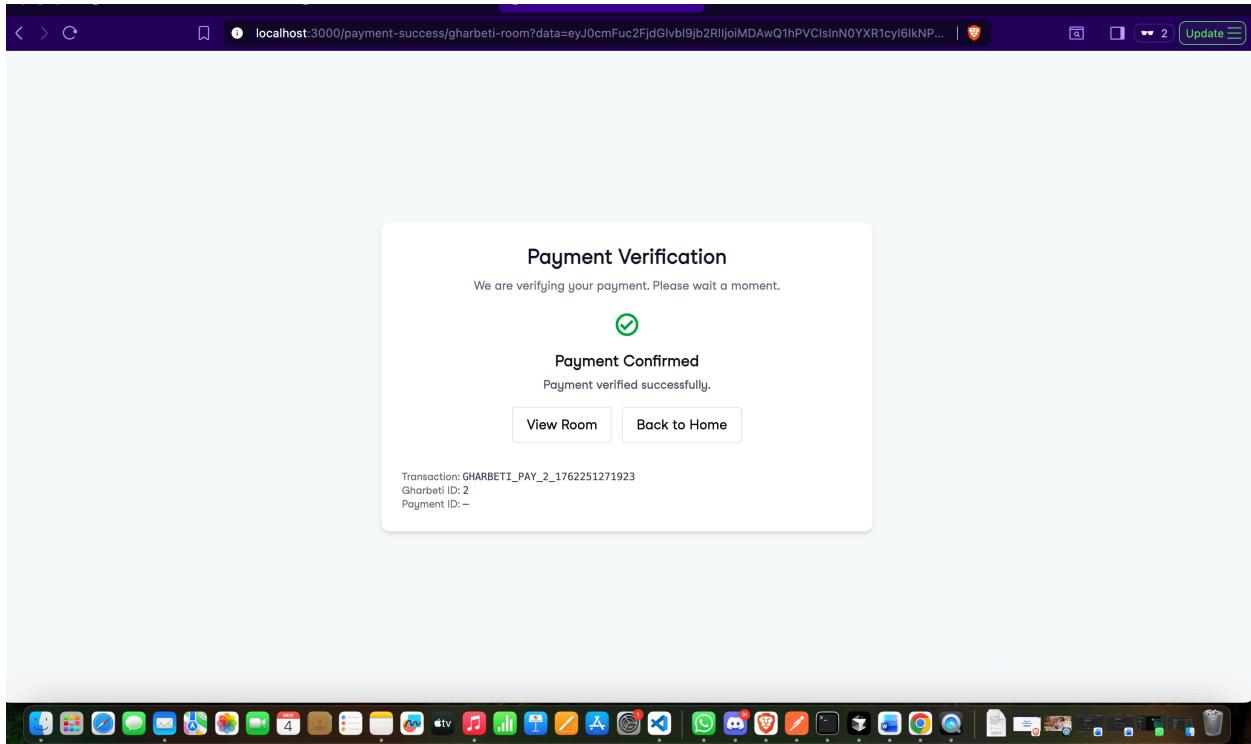


On click of that payment page



You gets redirected to this page click on pay with esewa

Once the payment is confirmed



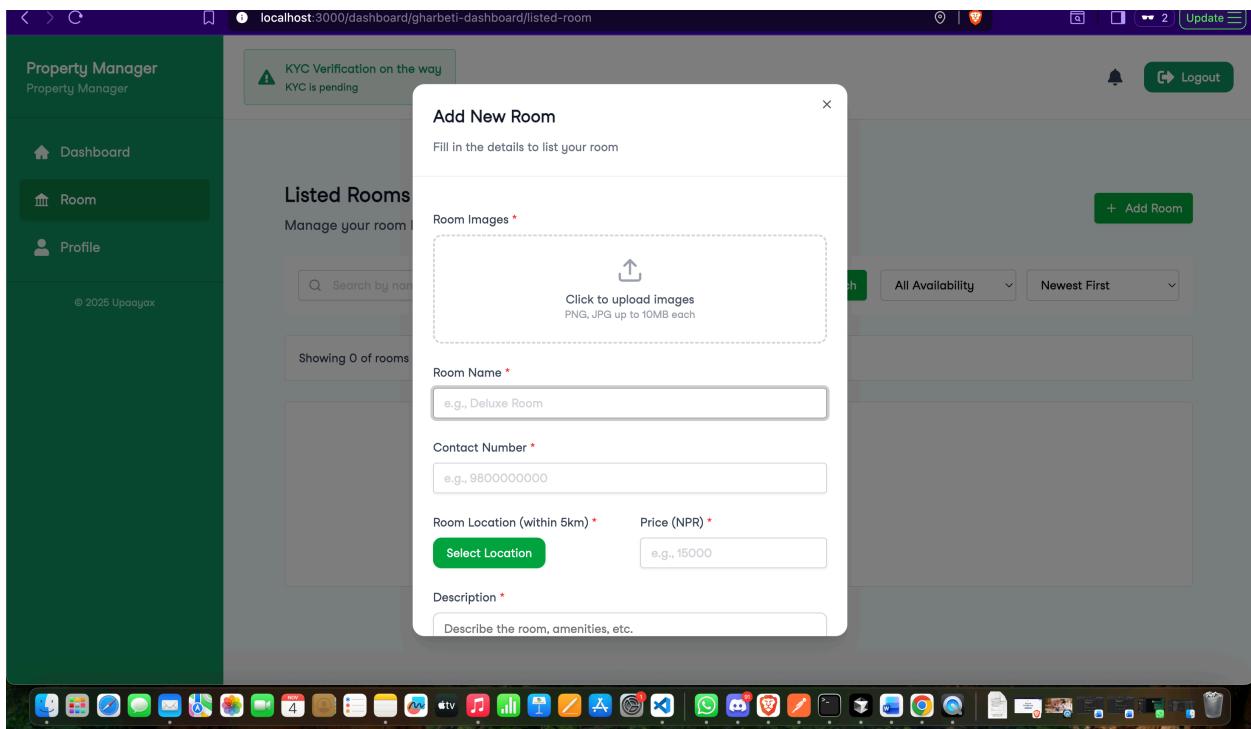
The gharbeti account is created and can now access the gharbeti dashboard

Now again on click of that be a gharbeti you wil be redirected to gharbeti dashboard

The screenshot shows a web browser window titled 'localhost:3000/dashboard/gharbeti-dashboard'. The page has a dark-themed header with the title 'Gharbeti Dashboard' and a sub-header 'Overview of your listed rooms'. On the left, there's a sidebar with a green header 'Property Manager' and a 'Property Manager' section. Below it are navigation links for 'Dashboard', 'Room', and 'Profile', along with a copyright notice '© 2025 Upayak'. A green banner at the top right says 'KYC Verification on the way KYC is pending'. The main content area includes four cards: 'Total Rooms' (0, Total listings), 'Verified' (0, Approved listings), 'Unverified' (0, Pending review), and 'Blocked' (0, Blocked listings). Below these are two sections: 'Room status distribution' with three bars (Verified, Unverified, Blocked) and 'Quick breakdown' with four items (Total, Verified, Unverified, Blocked, all 0). The bottom of the screen shows the Mac OS X dock with various application icons.

On room navigation bar and click on add room

## 10.1 Add room



You get this page and add the photos

Add all the detail, for the location part

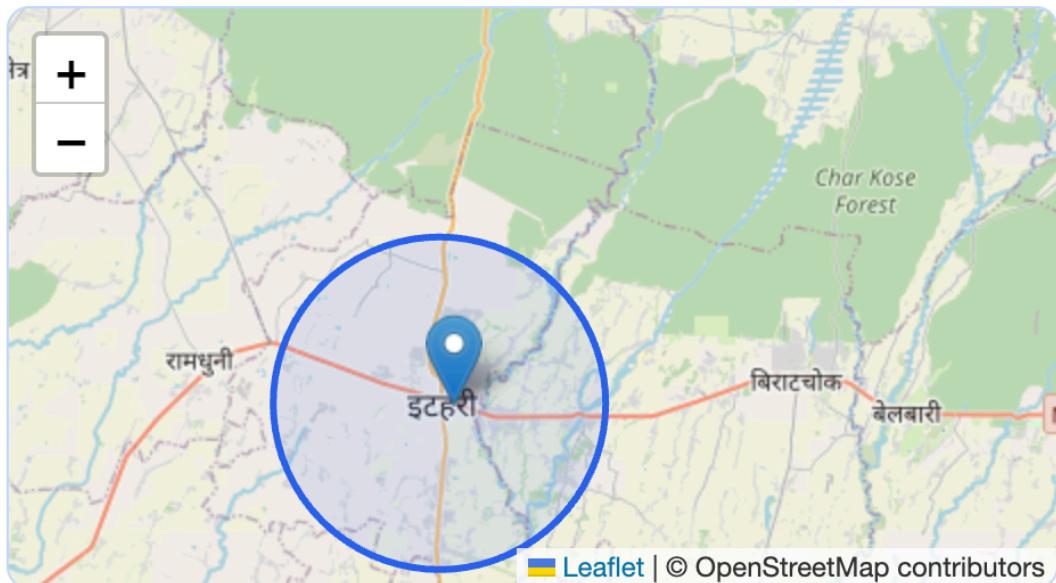
Only from current location to radius of 5km we can select the location

Fill in the details to list your room

Room Images \*

## Select Room Location

Pick a location within 5km of your current position



Cancel

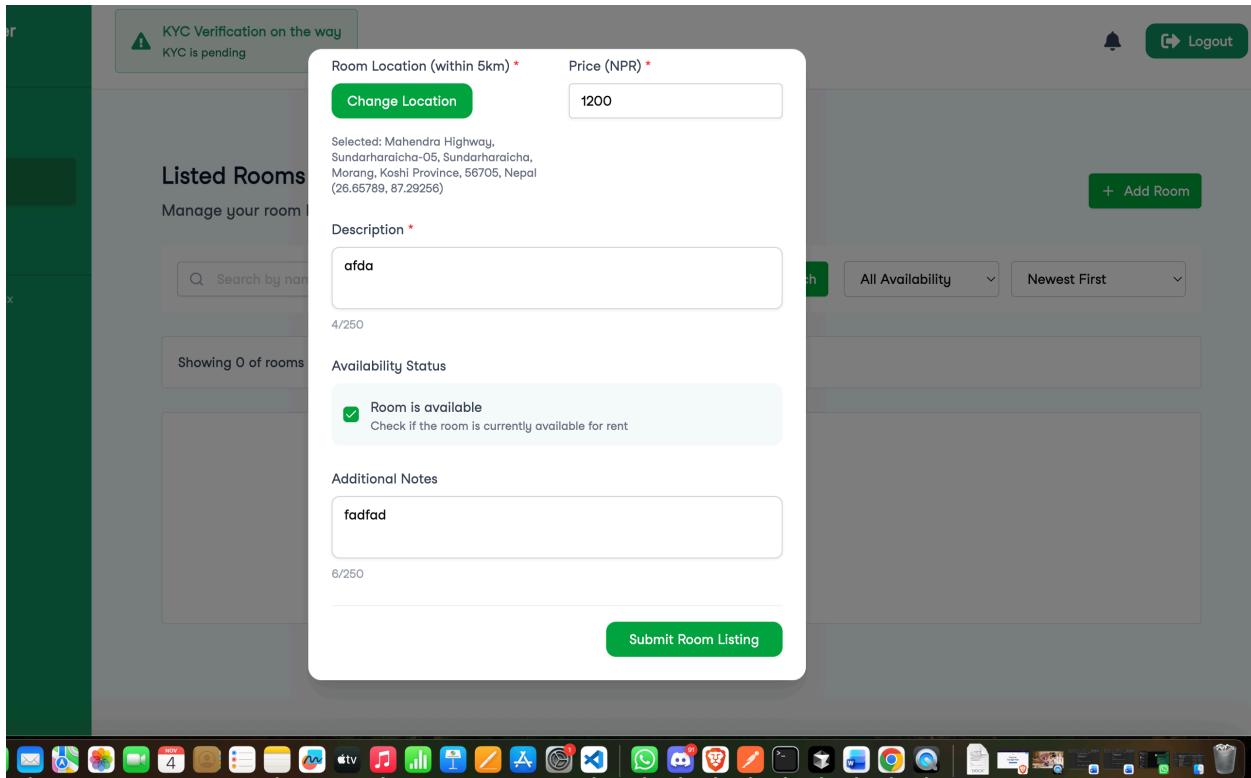
Confirm Location

Select Location

e.g., 10000

Description \*

When the details are added click on submit room listing



The fields it expects are:

Room images: (images)

Contact number: (10 digits Nepali number starts with 98 or 97)

Room location : (within 5km from current location)

Price: Positive number

Description: string

Availability status: 1 or 0

Additional Notes: String

The api for this is

<http://localhost:5000/api/rooms/create>

and the fields it expects are the given below

HTTP upaayax / gharbetiroomadd

POST http://localhost:5000/api/rooms/create

Send

Params Authorization Headers (11) Body Scripts Settings Cookies

None form-data x-www-form-urlencoded raw binary GraphQL

Key	Value	Description	Bulk Edit
images	File ▾ Black and White Simple Minimalist Job...		
name	Text ▾ new fadfalkajfa		
location	Text ▾ Mahendra Highway, Sunderharaicha-05, Sunderhar...		
lat	Text ▾ 26.657891331896508		
long	Text ▾ 87.29255676269533		
availability_status	Text ▾ 1		
note	Text ▾ adfa		
description	Text ▾ fasjdfkaf		
contact	Text ▾ 9876547883		
price	Text ▾ 3000		
Key	Value	Description	

Body Cookies (1) Headers (10) Test Results | ⏱

201 Created 26 ms 813 B ⓘ Save Response ⚙️

{ } JSON ▾ Preview Visualize ▾

```
1 {  
2   "room": [  
3     {"id": 2,  
4       "name": "new fadfalkajfa",  
5       "lat": 26.657891331896508  
6     }  
7   ]  
8 }
```

Runner Start Proxy Cookies Vault Trash ⓘ

The gharbeti can toggle the available and unavailable status

The screenshot shows a Postman interface with the following details:

- Method:** PUT
- URL:** <http://localhost:5000/api/rooms/toggle-availability/1>
- Body:** Raw JSON
- Request Body:**

```
1 {
2     "availability_status": false
3 }
```
- Response Status:** 200 OK
- Response Time:** 12 ms
- Response Size:** 479 B
- Response Content:**

```
1 {
2     "status": "success",
3     "code": 200,
4     "message": "Room availability status toggled successfully",
5     "data": {
6         "availability_status": false
7     }
8 }
```

## 10.2 For gharbeti profile

Use this api

<http://localhost:5000/api/users/about/gharbeti-detail>

GET http://localhost:5000/api/users/about/gharbeti-detail

Send

Params Authorization Headers (9) Body Scripts Settings Cookies

Headers 8 hidden

Key	Value	Description	Bulk Edit	Presets
Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...			
Key	Value	Description		

Body Cookies (1) Headers (10) Test Results

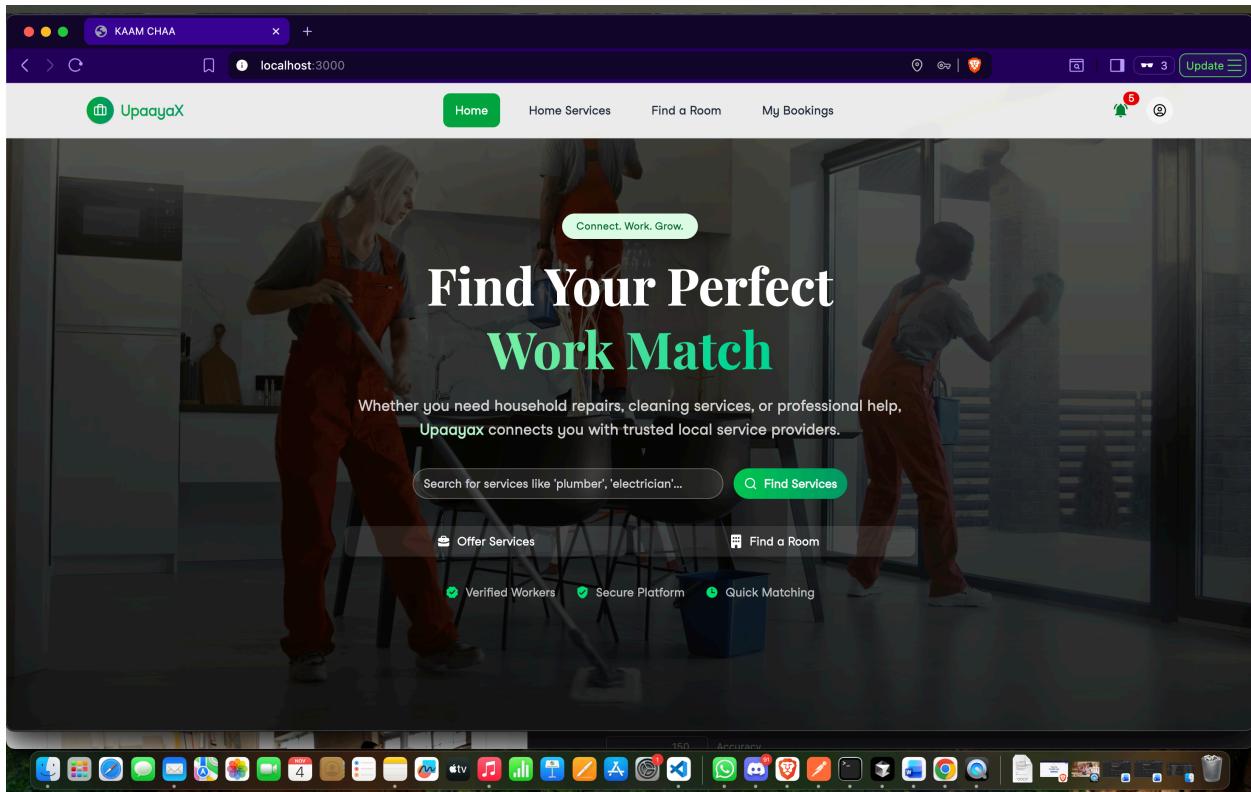
200 OK 25 ms 1.79 KB Save Response

{ } JSON Preview Visualize

```
1 {  
2     "status": "success",  
3     "code": 200,  
4     "message": "Gharbeti profile fetched successfully.",  
5     "data": {  
6         "gharbeti": {  
7             "status": "success",  
8             "code": 200,  
9             "message": "Gharbeti profile fetched successfully.",  
10            "data": {  
11                "gharbeti": {  
12                    "is_paid": true,  
13                    "is_verified": false,  
14                    "user": {  
15                        "id": 4,  
16                        "username": "admin43",  
17                        "refresh_token": null,  
18                        "email": "admin@gmail.com",  
19                        "phone_number": null,  
20                        "profile_picture": "/images/default-profile.jpg",  
21                        "is_active": true,  
22                    },  
23                },  
24            },  
25        },  
26    },  
27}
```

Runner Start Proxy Cookies Vault Trash ?

## 11 For booking part from user



Logged in from an account and click on Home Services

The screenshot shows the UpayaX website interface for home services. At the top, there's a navigation bar with links for Home, Home Services (which is the active tab), Find a Room, and My Bookings. On the right side of the header, there are icons for notifications (with 5 notifications) and account settings.

On the left, there are three filter boxes: "Category" (set to All Categories), "Price Range" (with input fields for Min Price and Max Price), and "Your Location" (showing Latitude: 26.6619, Longitude: 87.2741 with a "Show Map" button).

The main content area is titled "Available Services" and features a large image of a living room with a brown sofa set and a glass-top coffee table. Below the image, the service name "Appliance Repair" and its price "Rs. 42.00" are displayed.

A green button labeled "1" is located at the bottom center of the page.

Click on that service

This screenshot shows the UpayaX service detail page for the "Appliance Repair" service. The top navigation bar and filters are identical to the previous page.

The main visual is a larger image of the same brown sofa set and coffee table. Below it is a smaller inset image showing a close-up of the sofa.

The service name "Appliance Repair" is prominently displayed at the bottom of the service detail section.

To the right, there's a "Book this Service" form. It includes fields for "Select Day" (with options for Mon, Tue, Wed, Thu, Fri, Sat, Sun), "Select Time Slot", "Contact Number" (with placeholder "e.g. 98XXXXXXX"), "Price (Rs.)" (set to 42.00), "Bid Your Price (Optional)" (with placeholder "Enter your bid (max 6 digits)"), and a large green "Confirm Booking" button.

A Mac OS X dock is visible at the very bottom of the screen.

KAAM CHAA

localhost:3000/services-detail/1



**Appliance Repair**

Non natus aliquid no

**Rs. 42.00** (estimated price)

Notes:

Distance to provider: 0.3 km

Select Day

Mon	Tue	Wed
Thu	Fri	Sat
Sun		

Select Time Slot

11:00 AM - 12:00 PM	12:00 PM - 1:00 PM
1:00 PM - 2:00 PM	2:00 PM - 3:00 PM

Contact Number

Price (Rs.)

Bid Your Price (Optional)

Book this Service ✓ Booking placed. Price: Rs.1200

No packages available.

Confirm Booking

150 Accuracy

11:00 AM - 12:00 PM 12:00 PM - 1:00 PM  
1:00 PM - 2:00 PM 2:00 PM - 3:00 PM

Now for negotiation use my bookings section

## 12 My booking

Click on my bookings section and click on open negotiation

The screenshot shows a web browser window with the URL `localhost:3000/user/bookings`. The page is titled "My Bookings" and displays two service bookings:

- Appliance Repair** (Pending):
  - Provider
  - mon • 11:53 - 12:53
  - Location not provided

Open Negotiation
- Appliance Repair** (Completed):
  - Provider
  - sun • 14:53 - 15:53
  - Location not provided

Open Negotiation

At the bottom of the page, it says "Showing 1 - 2 of 2 bookings". The browser's address bar shows the URL `localhost:3000/user/bookings`. The top right corner of the browser window has a notification icon with the number 6.

localhost:3000/user/bookings/2

UpayaX Home Services Find a Room My Bookings 6

### Booking Details

Booking ID: #2

Negotiation Room  
Chat with your service provider

Rs. 1200  
11/4/2025, 4:23:34 PM

Enter your bid amount...  Send Bid

Service Actions  
Manage your booking actions

Payment Not Available

Cancel Booking

Mark As Complete

The screenshot shows a web application interface for managing bookings. At the top, there's a navigation bar with links for Home, Home Services, Find a Room, and My Bookings. A notification bell icon indicates 6 unread notifications. The main content area is titled "Booking Details" and shows a "Booking ID: #2". Below this, there's a "Negotiation Room" section with a "Chat with your service provider" link. A message card displays "Rs. 1200" and the date "11/4/2025, 4:23:34 PM". To the right, there's a "Service Actions" section with a "Manage your booking actions" link. It contains three buttons: "Payment Not Available" (disabled), "Cancel Booking" (disabled), and "Mark As Complete". At the bottom, there's a standard Mac OS X dock with various application icons.

Now the provider whose service was this can see this and do a bidding

This is the service provider page whose service was booked

The screenshot shows a service provider's dashboard with a green sidebar on the left and a white main content area on the right.

**Sidebar (Left):**

- Provider User
- Service Provider
- [Dashboard](#)
- [Services](#)
- [Profile](#)
- [Payment](#)
- [My Bookings](#) (highlighted in green)

**Main Content Area (Right):**

Logout

### My Bookings

Filter: Pending

Appliance	Status	Payment	Actions
Appliance Repair	Pending	Payment: No	Payment Info
ID: #2			
admin			
mon, 11:53 AM - 12:53 PM			
Bid: Rs.1200			
Booked on: Nov 4, 2025			

© 2025 Kaam-Chaa. All rights reserved.

The dashboard includes a standard Mac OS X dock at the bottom with various application icons.

The screenshot shows a web-based provider dashboard titled "Provider User Service Provider". The left sidebar includes links for Dashboard, Services, Profile, Payment, and My Bookings. The main area is titled "My Bookings" and shows a "Booking #2 — Appliance Repair" window. This window displays a bid history: "Rs. 1200" (by "U" at 11/4/2025, 4:23:34 PM) with "Accept" and "Reject" buttons; "Rs. 200" (by "R" at 11/4/2025, 4:27:10 PM); and a text input field "Enter your bid amount..." with a "Send Bid" button. The background shows other bookings and a footer with copyright information.

The screenshot shows a mobile-style provider interface titled "UpayaX". The top navigation bar includes Home, Home Services, Find a Room, and My Bookings. The "My Bookings" tab is active. The main content area is titled "Booking Details" and shows "Booking ID: #2". It features a "Negotiation Room" section with a "Chat with your service provider" link and a bidding history: "Rs. 1200" (by "U" at 11/4/2025, 4:23:34 PM), "Rs. 200" (by "R" at 11/4/2025, 4:27:10 PM), and "Rs. 500" (by "R" at 11/4/2025, 4:27:26 PM). Below this is a "Send Bid" button. To the right, a "Service Actions" panel lists "Payment Not Available", a red "Cancel Booking" button, and "Mark As Complete". The bottom of the screen shows a standard iOS-style toolbar with various app icons.

Now the user can accept or decline the bid

The screenshot shows a web browser window for 'UpayaX' at 'localhost:3000/user/bookings/2'. The main content is titled 'Booking Details' with 'Booking ID: #2'. On the left, under 'Negotiation Room', there's a chat interface with four messages:

- Rs. 1200 Rejected (11/4/2025, 4:23:34 PM)
- Rs. 200 Rejected (11/4/2025, 4:27:10 PM)
- Rs. 500 Accepted (11/4/2025, 4:27:26 PM) - This message is highlighted with a blue background.
- Rs. 800 Rejected (11/4/2025, 4:27:31 PM)

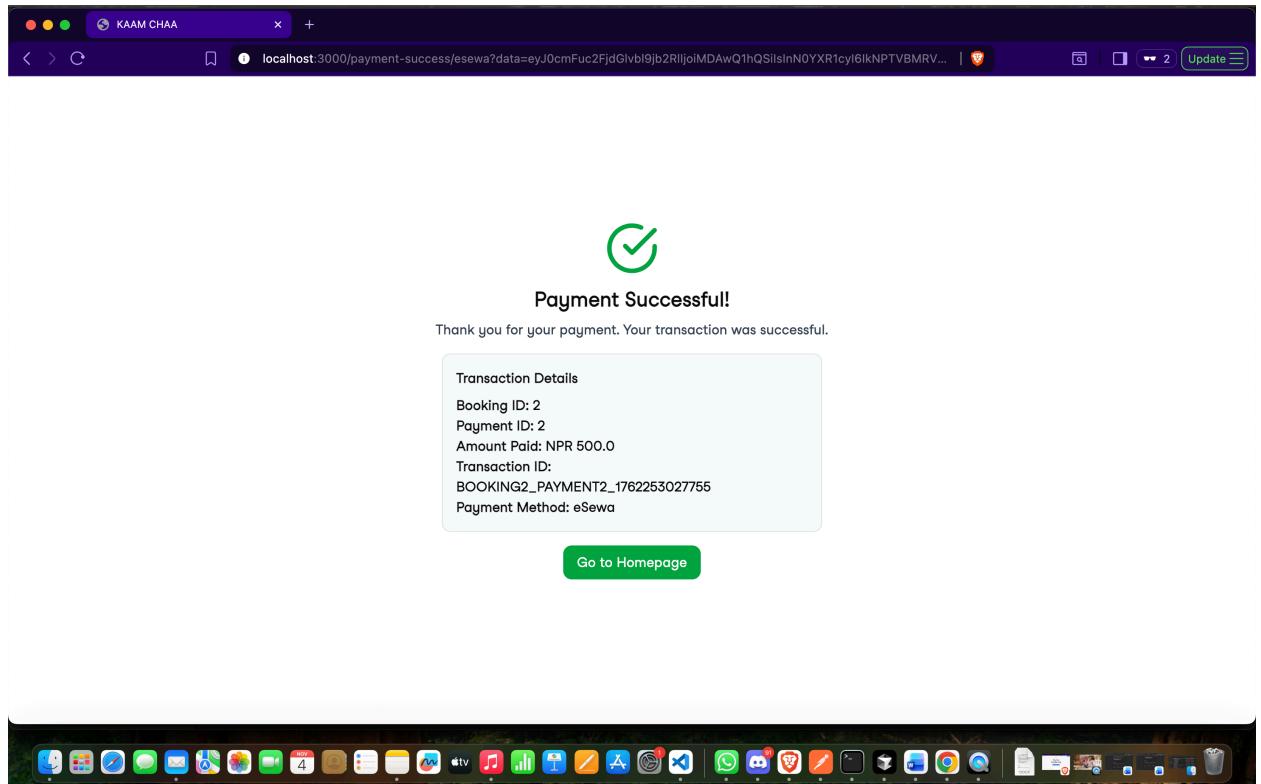
A green box at the bottom of this section says 'Bid Accepted! A bid has been accepted. No further bidding is allowed.'

On the right, under 'Service Actions', there are three buttons:

- Pay Rs. 500 (green button)
- Cancel Booking (disabled, grey button)
- Mark As Complete (disabled, grey button)

A green notification bubble in the top right corner says 'Bid Rs.500 accepted!' with a checkmark icon.

When the user accepts the bid then he cant cancel the book and mark as complete unless he had paid



It has like this now the user only has the option for mark as complete

Now on click on mark as complete rating and comments are entered

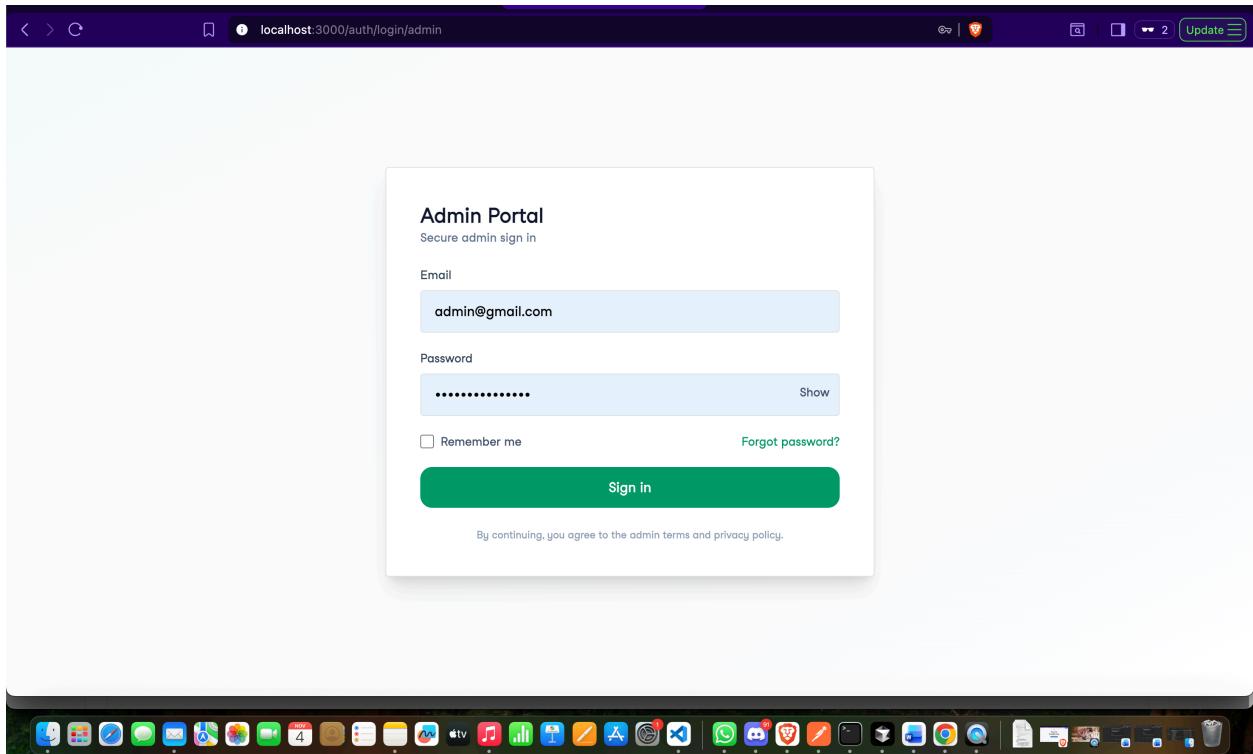
The screenshot shows a web browser window for 'UpayaX' at 'localhost:3000/user/bookings/2'. The main content area is titled 'Booking Details' and shows 'Booking ID: #2'. On the left, there's a 'Negotiation Room' section with a heading 'Chat with your service provider'. It displays a history of bids:

- A bid from 'U' for 'Rs. 200' on '11/4/2025, 4:27:10 PM'
- A bid from 'U' for 'Rs. 500' with the status 'Accepted' on '11/4/2025, 4:27:26 PM'
- A final bid from 'U' for 'Rs. 800' on '11/4/2025, 4:27:31 PM'

To the right, under 'Service Actions', it says 'Manage your booking actions' and shows a message: 'You have rated this booking average' with a star rating of 3.5 stars and the comment 'good one' on '11/4/2025, 4:30:25 PM'.

Now this can be seen from the admin site as well

## 13 Admin Site



When the admin is logged in, it gets redirected to admin dashboard

localhost:3000/dashboard/admin-dashboard

admin  
admin@gmail.com

## Admin Dashboard

Overview of key metrics and performance

Today: 11/4/2025

Total Users: 2

Providers: 1

Services: 11

Bookings: 2

Revenue: Rs. undefined

Gharbeti: 2

Revenue (last 7 days)

Trend and income

Recent Bookings (2 recent)

- Appliance Repair - Nov 4, 2025 Provider: raghavdahal Rs. 500 Completed
- Appliance Repair - Nov 4, 2025 Provider: raghavdahal Rs. 200 Completed

© 2025 Upayax. All rights reserved.

## On bookings nav item

This is the booking page, now the booking that was confirmed in 500 is seen and can be view

The screenshot shows the Admin — Bookings page. On the left, there is a sidebar with the following navigation items:

- Dashboard
- Bookings** (selected)
- Manage Users
- Manage KYC
- Services
- Rooms
- Manage Categories
- Settings

At the bottom of the sidebar, it says "© 2025 Upayak. All rights reserved."

The main content area has a title "Admin — Bookings". Below the title are six buttons for filtering bookings: All (2), Pending (0), Confirmed (0), Completed (2), Cancelled (0), and Rejected (0). There is also a search bar and a dropdown menu set to "All".

The main content area displays two completed bookings for "Appliance Repair".

Booking ID	Description	Status	Time	Amount	Action
#2	Appliance Repair	completed	mon 11:55 AM - 12:55 PM	Rs. 500	<a href="#">View</a>
#1	Appliance Repair	completed	sun 2:53 PM - 3:53 PM	Rs. 200	<a href="#">View</a>

At the bottom of the main content area, it says "Showing 1-2 of 2 booking(s)" and "Page 1 / 1".

The bottom of the screen shows a Mac OS X dock with various application icons.

On viewing that booking, the admin can see the booking details

The screenshot shows a web browser window titled "Admin — Bookings" on the "localhost:3000/dashboard/admin-dashboard/bookings" page. The left sidebar has a green header "admin" and "admin@gmail.com" with options: Dashboard, Bookings (selected), Manage Users, Manage KYC, Services, Rooms, Manage Categories, Settings, and a footer "© 2020 Upstage. All rights reserved.". The main content area shows a modal for "Booking #2" for "Appliance Repair". The booking is marked as "Completed" by "raghavdahal" (raghavdahal@gmail.com). The modal details are:

User	Provider
Contact: 9876543212	raghavdahal
Schedule	raghavdahal@gmail.com
mon 11:53 AM - 12:53 PM	—
Confirmed bid	Package
Rs. 500	—
Location	Other info
<a href="#">See on Google Maps</a>	—

Below the modal, there are two completed bookings listed:

- Booking ID: #2, Status: completed, Rs. 500, View
- Booking ID: #1, Status: completed, Rs. 200, View

A "Close" button is at the bottom right of the modal. The Mac OS X dock is visible at the bottom with various application icons.

## 13.1 manage user part

The screenshot shows the 'Manage Users' page of a web application. The left sidebar is dark green and contains the user information 'admin' and 'admin@gmail.com'. Below this, there are several menu items: 'Dashboard', 'Bookings', 'Manage Users' (which is highlighted in blue), 'Manage KYC', 'Services', 'Rooms', 'Manage Categories', and 'Settings'. At the bottom of the sidebar, it says '© 2025 Upaayax. All rights reserved.'.

The main content area has a title 'Manage Users' with the sub-instruction 'Search, filter, and manage user access.' Below this are four summary boxes: 'Total Users 2' (blue icon), 'Active Users 2' (green icon), 'Blocked Users 0' (red icon), and 'Service Providers 1' (blue icon). There is a search bar with placeholder text 'Search users by name or email...' and three filter buttons: 'All' (green), 'Active' (green), and 'Blocked' (red).

Two user entries are listed in a table-like format:

User Icon	User Name	Email	Status	Action Buttons
A	admin	admin@gmail.com	Active	Details (green), Block (red)
R	raghavdahal	raghavdahal@886@gmail.com	Blocked	Active (green), Details (green), Block (red)

The bottom of the screen features a dark dock with numerous application icons, typical of a Mac OS X desktop environment.

On click of detail it shows the user details

The screenshot shows a web application running on localhost:3000 at the URL [localhost:3000/dashboard/admin-dashboard/users](http://localhost:3000/dashboard/admin-dashboard/users). The interface has a dark purple header with navigation icons and a title bar. On the left, a green sidebar menu lists various administrative functions: Dashboard, Bookings, Manage Users, Manage KYC, Services, Rooms, Manage Categories, and Settings. The 'Manage Users' option is selected. The main content area is titled 'Manage Users' with a subtitle 'Search, filter, and manage user accounts.' It displays a user profile for 'admin' (admin@gmail.com), showing roles 'Admin' and 'Gharbeti'. Below this, there are tabs for 'Overview', 'Past Bookings', and 'Services'. The 'Overview' tab is active, showing details like Email (admin@gmail.com), Status (Active), Joined (11/4/2025), and Roles (Admin, Gharbeti). To the right, there is a modal window titled 'Service Providers' with tabs for 'All', 'Active', and 'Blocked'. The 'Active' tab is selected, showing three service providers. At the bottom of the main content area are 'Close' and 'Block' buttons. The bottom of the screen shows a Mac OS X-style dock with various application icons.

Along with all past-bookings, services and all

## On past\_booking section

The screenshot shows a web application interface for managing users. On the left, there is a sidebar with the following menu items:

- Dashboard
- Bookings
- Manage Users
- Manage KYC
- Services
- Rooms
- Manage Categories
- Settings

The main area is titled "Manage Users" and has a sub-header "Search, filter, and manage user details." A modal window is open, showing the "Past Bookings" tab for the user "admin". The modal displays two completed bookings:

- Appliance Repair** (1/4/2025) - Accepted: NPR 200
- Appliance Repair** (1/4/2025) - Accepted: NPR 500

At the bottom of the modal, there are "Close" and "Block" buttons.

On services section this is the services section

The screenshot shows a web-based administration interface for managing users. The URL in the browser bar is `localhost:3000/dashboard/admin-dashboard/users`. The left sidebar, under the heading 'admin admin@gmail.com', includes links for Dashboard, Bookings, Manage Users (selected), Manage KYC, Services (selected), Rooms, Manage Categories, and Settings. The main content area is titled 'Manage Users' with the subtitle 'Search, filter, and manage user access.' It displays four summary boxes: Total Users (2), Active Users (2), Blocked Users (0), and Service Providers (1). A central modal window is open for the user 'admin' (admin@gmail.com), showing roles Admin and Gharbeti. The modal has tabs for Overview, Past Bookings, and Services (which is selected). Below the tabs, it says 'No services available.' and features 'Close' and 'Block' buttons. To the right of the modal, there are three smaller cards for 'Active', 'Details', and 'Blocked' users.

On manage kyc part, this is the kyc page that has the option for pending, approved, rejected,

The screenshot shows a web-based application interface titled "KYC Verification Management". The left sidebar is a dark green navigation bar with the following items:

- admin (username)
- admin@gmail.com
- Dashboard
- Bookings
- Manage Users
- Manage KYC
- Services
- Rooms
- Manage Categories
- Settings

Below the sidebar, a copyright notice reads: "© 2025 Upayax. All rights reserved."

The main content area is titled "KYC Verification Management" and includes a sub-header "Review and approve provider applications and documentation". It features a summary section with four boxes:

- Total KYC Submissions: 3 (blue icon)
- Pending KYC: 2 (orange icon)
- Approved KYC: 1 (green icon)
- Rejected KYC: 0 (red icon)

Below this is a search bar with placeholder text "Search applicants by name or email..." and filters for "All Entity Types" and "All Document Types". There are also "Reset" and "Refresh" buttons.

The main list displays two pending applications:

- admin43**: Applied nov 4 2025 + gharbeti. Status: pending. Actions: Front, Back, Passport Photo, Reject, Approve.
- raghavdahal686841**: Applied nov 4 2025 + gharbeti. Status: pending. Actions: File, Passport Photo, Reject, Approve.

The bottom of the screen shows a Mac OS X dock with various application icons.

On view detail it shows like this

Screenshot of a web browser displaying a user profile page. The URL in the address bar is `localhost:3000/dashboard/admin-dashboard/pending-kycs`. The page shows a user named "admin" with the email "admin@gmail.com". The user is marked as "Active".

**Basic Information:**  
Username: admin43  
Email: admin@gmail.com  
Phone: —  
User ID: 4

**Account Details:**  
Status: active  
Joined: 11/4/2025  
Last Updated: 11/4/2025

**KYC Information:**  
Document Type: Citizenship\_card  
KYC Status: pending  
Verified at: N/A

Three thumbnail images showing a workshop environment.

**Send Email**

The browser window is part of a Mac OS X desktop, as indicated by the Dock at the bottom which includes icons for various Apple applications like Mail, Safari, and Finder.

## 13.2 Manage the kyc

The screenshot shows the Upayax Admin Dashboard interface. On the left, a green sidebar menu lists various administrative functions: Dashboard, Bookings, Manage Users, Manage KYC, Services, Rooms, Manage Categories, and Settings. The main content area displays a summary of KYC submissions: Total KYC Submissions (2), Pending KYC (2), Approved KYC (0), and Rejected KYC (0). Below this, a search bar allows filtering by applicant name or email, and dropdown menus for All Entity Types and All Document Types. A 'Refresh' button is also present. The main list shows two pending KYC applications from 'raghavdahal686819' (raghavdahal686@gmail.com), both applied on November 6, 2025. Each application has sections for Submitted Documents, View Details, Contact, Reject, Approving..., and Approve buttons. Navigation buttons for Previous, Next, and page 1 are at the bottom.

Clicking of approve button will trigger the api

<http://localhost:5000/api/admin/kyc/verify>

that expects the payload

HTTP upayaxFINAL APIs / private api / admin / kyc-approve

POST http://localhost:5000/api/admin/kyc/verify

Overview Params Authorization Headers (11) Body Scripts Settings Cookies Schema Beautify

Body raw JSON

```
1 {
2     "action": "approve",
3     "kycId": 7
4 }
5 // for reject action: "reject" and add rejectionReason
```

Body Cookies (1) Headers (10) Test Results

200 OK 2.92 s 713 B Save Response

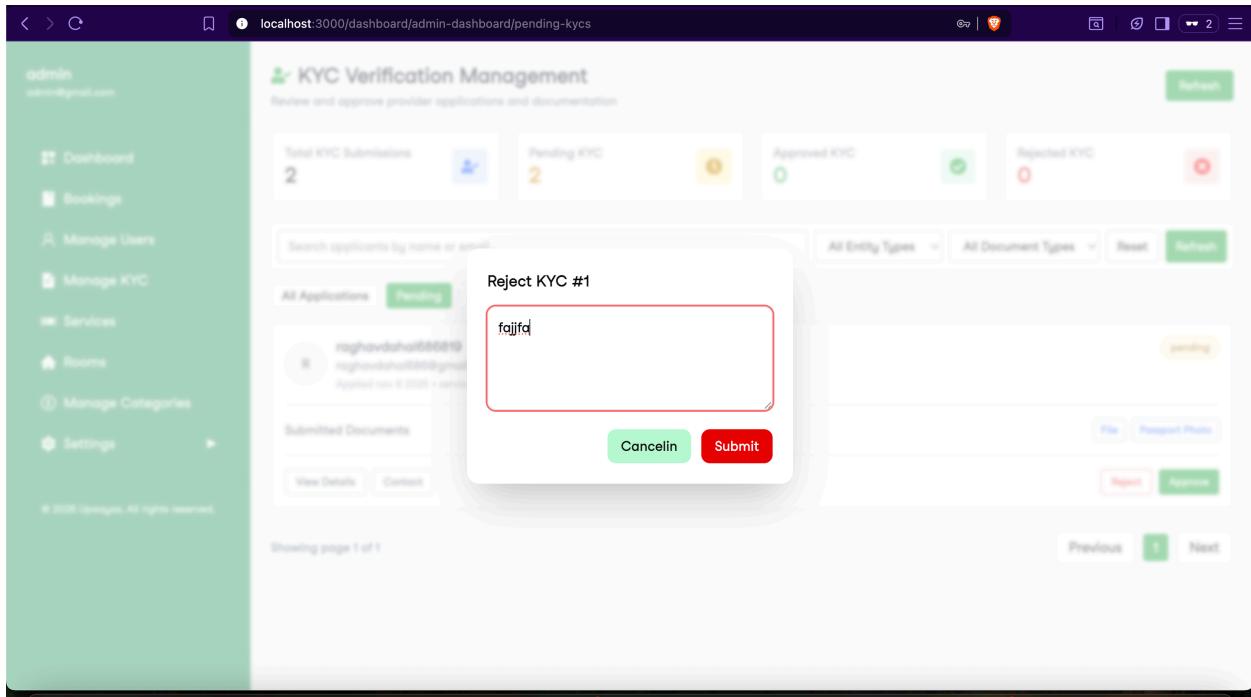
{ } JSON Preview Visualize

```
1 {
2     "status": "success",
3     "code": 200,
4     "message": "KYC verification successful.",
5     "data": {
6         "kyc": {
7             "id": 7,
8             "document_type": "passport",
9             "status": "approved",
10            "verified_at": "2025-11-06T13:21:13.272Z",
11            "rejection_reason": null,
12            "entityType": "service_provider",
13            "entityId": 4,
14            "createdAt": "2025-11-06T12:44:30.000Z",
15            "updatedAt": "2025-11-06T13:21:13.272Z",
16            "deletedAt": null,
17            "userId": 8
18        }
19    }
20 }
```

Runner Start Proxy Cookies Vault Trash

For Rejection the action: “reject”

And specify the rejectionReason



Here I had rejected this kyc if the kyc is rejected then the user must again submit the kyc from beginning

Note: all the messages are sent in email as well

## 13.3 Services

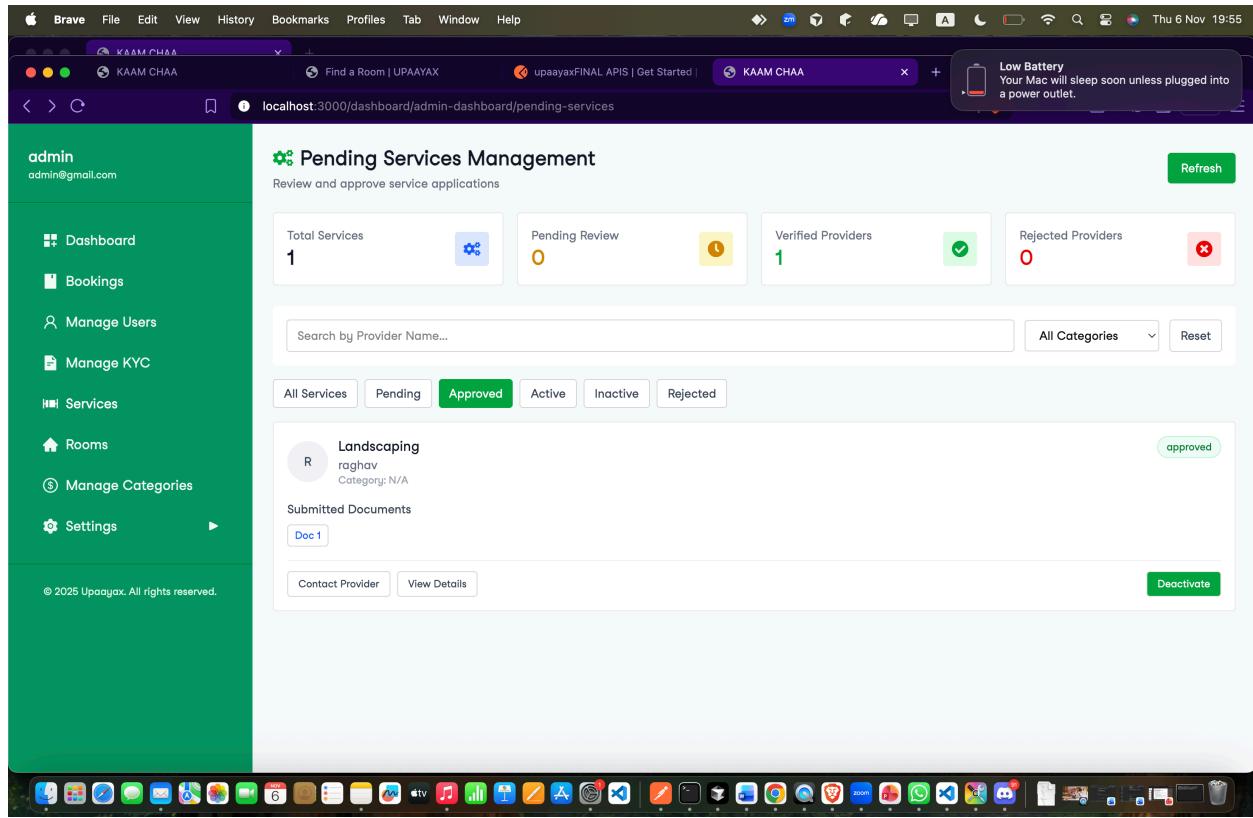
The screenshot shows a web application titled "Pending Services Management" running on a Mac OS X system. The left sidebar is a dark green navigation menu with the following items: Dashboard, Bookings, Manage Users, Manage KYC, Services (selected), Rooms, Manage Categories, and Settings. The main content area has a white background. At the top, there are four status boxes: "Total Services 1" (blue icon), "Pending Review 1" (yellow icon), "Verified Providers 0" (green icon), and "Rejected Providers 0" (red icon). Below these are search and filter controls: a search bar ("Search by Provider Name..."), a dropdown for "All Categories", and buttons for "All Services", "Pending" (highlighted in green), "Approved", "Active", "Inactive", and "Rejected". A detailed service entry is shown: "Landscaping" by "raghav" (Category: N/A), with a "pending" status indicator. Under "Submitted Documents", there is a link "Doc 1". At the bottom of the main area are "Contact Provider" and "View Details" buttons, followed by "Reject" and "Approve" buttons. The Mac OS X dock at the bottom of the screen shows various application icons.

This is the page for manage services, only the approved services are seen in the home page

On view detail the api it triggers is

```
http://localhost:5000/api/services/service-detail/1?
```

Approving the services



Similarly on rejection the message is required and the rejection message is sent to mail

### 13.4 Deactivating the services

The admin can deactivate the services by pressing the deactivate button that triggers to this api

<http://localhost:5000/api/admin/service/toogle-service-status/1>

the payload it expects are : {

is\_active:false/true}

based on deactivate and activate status

Import POST bloc POST unbl GET mana GET get us POST kyc- PUT deact GET rooml POST add- PUT edit-s ➔ a power outlet.

HTTP upayaxFINAL APIs / private api / admin / deactivate service

PUT http://localhost:5000/api/admin/service/toogle-service-status/4 Send

Overview Params Authorization Headers (11) Body Scripts Settings Cookies (11) </> Schema Beautify

Body none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "is_active"
3 :
4   false
5 }
6
7 // change true for activate same API
```

Body Cookies (1) Headers (10) Test Results 200 OK 3.31 s 818 B Save Response

{ } JSON Preview Visualize

```
1 {
2   "status": "success",
3   "code": 200,
4   "message": "Service provider service active status updated successfully",
5   "data": {
6     "serviceProviderService": [
7       {
8         "id": 4,
9         "rate": "23.00",
10        "description": "dsadf",
11        "status": "approved",
12        "rejected_reason": null,
13        "isReapplied": false,
14        "reappliedCount": 0,
15        "includes": [
16          "fa"
17        ],
18        "note": "faf",
19        "reappliedAt": null
20      }
21    ]
22  }
23}
```

Runner Start Proxy Cookies Vault Trash ?

## 13.5 Manage rooms

Manage rooms will trigger

The screenshot shows a web-based administrative interface titled "Manage Rooms". On the left, a sidebar menu includes "Dashboard", "Bookings", "Manage Users", "Manage KYC", "Services", "Rooms", "Manage Categories", and "Settings". The main content area displays room statistics: "Total Rooms 5", "Pending 5", "Approved 0", and "Rejected 0". Below these are four room listings:

- new updated**  
raghav  
Price: NPR 2200  
Status: pending  
Actions: View, Approve, Reject
- dghjk**  
raghav  
Price: NPR 4567  
Status: pending  
Actions: View, Approve, Reject
- fasdfa**  
raghav  
Price: NPR 1200  
Status: pending  
Actions: View, Approve, Reject
- fdaf**  
raghav  
Price: NPR 1221  
Status: pending  
Actions: View, Approve, Reject

The browser address bar shows "localhost:3000/dashboard/admin-dashboard/pending-rooms". The bottom of the screen shows a Mac OS X dock with various application icons.

**Note: for room now no need of approve and reject (it will only be listed once the payment is done) before that the client had told for the manual approve and rejection**

## 13.6 Manage Categories

The screenshot shows a web-based admin dashboard titled "Services". The left sidebar, titled "admin" and "admin@gmail.com", includes links for Dashboard, Bookings, Manage Users, Manage KYC, Services, Rooms, Manage Categories, and Settings. The main content area is titled "Services" and describes "Manage services offered on the platform". A search bar and a green "Add Service" button are at the top right. Below is a table with columns: ID, Name, Package, and Actions. The table lists 10 services, all currently set to "Disabled". Each service row has edit and delete icons in the "Actions" column.

ID	Name	Package	Actions
1	Plumbing	Disabled	
2	Electrical	Disabled	
3	Carpentry	Disabled	
4	Cleaning	Disabled	
5	Painting	Disabled	
6	Appliance Repair	Disabled	
7	Landscaping	Disabled	
8	Pest Control	Disabled	
9	Moving Services	Disabled	
10	Roofing	Disabled	

This is the services page the admin can enable and disable the packages for the services. And add the services as well

### For add services

The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:5000/api/admin/service/add`
- Method:** POST
- Body:** JSON (selected)  
Content:

```
1  {
2     "name"
3 :
4     "newokay"
5 }
```
- Response Headers:** 201 Created, 11 ms, 596 B
- Response Body:** JSON (selected)  
Content:

```
1  {
2      "status": "success",
3      "code": 201,
4      "message": "Service added successfully",
5      "data": {
6          "newService": [
7              {
8                  "status": "admin",
9                  "id": 14,
10                 "name": "newokay",
11                 "package_enabled": false,
12                 "updatedAt": "2025-11-06T13:22:19.722Z",
13                 "createdAt": "2025-11-06T13:22:19.722Z"
14             }
15 }
```

## For editing the services

The screenshot shows a POSTMAN API client interface. At the top, there is a header bar with the title "Edit Services". Below it, a search bar contains the URL "http://localhost:5000/api/admin/service/edit/12". To the right of the URL is a large blue "Send" button.

The main area is divided into tabs: "Overview", "Params", "Authorization", "Headers (11)", "Body", "Scripts", and "Settings". The "Body" tab is currently selected and has a green dot next to it. Below the tabs, there are several input options: "none", "form-data", "x-www-form-urlencoded", "raw" (which is selected and highlighted in blue), "binary", and "GraphQL". To the right of these options are "Schema" and "Beautify" buttons.

The "Body" section contains the following JSON payload:

```
1 {  
2   "name": "hehe",  
3   "package_enabled": false  
4 }
```

Below the body, there is a preview section titled "Body" which shows the response in JSON format. The response code is "200 OK" and the status message is "success". The response body is as follows:

```
1 {  
2   "status": "success",  
3   "code": 200,  
4   "message": "Service updated successfully",  
5   "data": {  
6     "updatedService": {  
7       "id": 12,  
8       "name": "hehe",  
9       "package_enabled": false,  
10      "status": "admin",  
11      "createdAt": "2025-11-05T13:20:43.000Z",  
12      "updatedAt": "2025-11-05T13:22:21.000Z",  
13      "deletedAt": null  
14    }  
15  }  
16 }
```

At the bottom of the interface, there are several navigation and utility buttons: "Runner", "Start Proxy", "Cookies", "Vault", "Trash", and a help icon.

localhost:3000/dashboard/admin-dashboard/services

Services

Manage services offered on the platform.

Search services... + Add Service

ID	Name	Package	Actions
1	Plumbing	Disabled	
2	Electrical	Disabled	
3	Cleaning	Disabled	
4	Cleaning	Disabled	
5	Painting	Disabled	
6	Appliance Repair	Disabled	
7	Landscaping	Disabled	
8	Pest Control	Disabled	
9	Moving Services	Disabled	
10	Roofing	Disabled	

**Edit Service**

Update allowed fields for this service

Service Name: Plumbing

Package Enabled:

Cancel Save

The screenshot shows a Mac OS X desktop environment. At the top is the Dock with various application icons. Below the Dock is the application window for 'localhost:3000/dashboard/admin-dashboard/services'. The window title is 'Services' and it says 'Manage services offered on the platform.' There is a search bar and a green 'Add Service' button. A modal dialog titled 'Edit Service' is open over the main table, prompting to 'Update allowed fields for this service'. It contains a 'Service Name' input field with 'Plumbing' typed in, a 'Package Enabled' toggle switch which is turned on (indicated by a blue circle), and two buttons at the bottom: 'Cancel' and 'Save' (highlighted in green). The main table lists ten services: Plumbing, Electrical, Cleaning, Cleaning, Painting, Appliance Repair, Landscaping, Pest Control, Moving Services, and Roofing, each with a status of 'Disabled' and edit/delete icons. The browser's address bar shows the URL 'localhost:3000/dashboard/admin-dashboard/services'. The top right of the screen shows the system menu with icons for battery, signal, and other system status.

## 13.7 Payment section

For release payment on booking part

localhost:3000/dashboard/admin-dashboard/settings/payment-release

Payment Release Management

Manage and release payments for completed bookings

1 Record

BOOKING ID	SERVICE & PARTIES	SCHEDULE	AMOUNT	STATUS	ACTIONS
#1	Moving Services Provider: raghav raghavdahal686@gmail.com	21:05:00-22:05:00	Rs. 12	completed	<a href="#">View</a> <a href="#">\$ Releasing...</a>

On clicking of release handle the api

The screenshot shows the Postman application interface. At the top, there's a navigation bar with various API endpoints like 'initial', 'GET list-ro', 'PUT chang...', 'GET view...', 'GET gharb', 'GET gharb', 'GET getdo', 'GET room', 'admin', and 'POST rele...'. Below the navigation is a search bar with the URL 'http://localhost:5000/api/payments/release-payment/1'. The main area has tabs for 'Overview', 'Params', 'Authorization' (which is selected), 'Headers (10)', 'Body', 'Scripts', and 'Settings'. Under 'Authorization', it says 'Auth Type: Bearer Token' and shows a redacted token value. A note below explains that the token will be automatically generated. The 'Body' tab is selected, showing a JSON response with status: "success", code: 200, message: { amountReleased: 10.8, serviceProviderId: 1 }, and data: "Payment released successfully". The status bar at the bottom indicates a 200 OK response with 52 ms duration and 479 B size.

```
1
2   "status": "success",
3   "code": 200,
4   "message": {
5     "amountReleased": 10.8,
6     "serviceProviderId": 1
7   },
8   "data": "Payment released successfully"
9
```

For room payment fetch

**Room Payments Settings**

ID	User	Email	Amount	Status	Type	Payment Date	Created At
6	raghav	raghavdahal686@gmail.com	NPR 100	completed	viewer	11/6/2025, 7:40:11 PM	11/6/2025, 7:36:09 PM
5	raghav	raghavdahal686@gmail.com	NPR 500	completed	gharbeti	11/6/2025, 7:24:19 PM	11/6/2025, 7:23:16 PM
4	raghav	raghavdahal686@gmail.com	NPR 500	pending	gharbeti	—	11/6/2025, 7:21:14 PM
3	raghav	raghavdahal686@gmail.com	NPR 500	pending	gharbeti	—	11/6/2025, 7:19:56 PM
2	raghav	raghavdahal686@gmail.com	NPR 500	pending	gharbeti	—	11/6/2025, 7:19:52 PM
1	raghav	raghavdahal686@gmail.com	NPR 500	pending	gharbeti	—	11/6/2025, 7:19:11 PM

Showing page 1 of 1 — 6 items

Prev | Next

This is the page above

HTTP upayaayxFINAL APIs / private api / admin / room payment

GET http://localhost:5000/api/admin/room-verification/room-payments?page=1&limit=10

room payment

```

1 {
  "status": "success",
  "code": 200,
  "message": "Request processed successfully",
  "data": [
    {
      "data": [
        {
          "id": 6,
          "userId": 8,
          "userName": "raghav",
          "userEmail": "raghavdahal686@gmail.com",
          "amount": 100,
          "status": "pending",
          "type": "viewer",
          "roomTitle": "sdifghj",
          "contact": "9808776657",
          "paymentDate": null,
          "createdAt": "2025-11-06T13:15:51.000Z"
        }
      ]
    }
  ]
}

```

This is the api that it hits above