

Hi David, I've made the corrections you need though you have repeated a lot of things which are already existing in the APIs

Date: July 16, 2025

Subject: Alignment of Frontend Design Requirements with Current Backend API Functionality

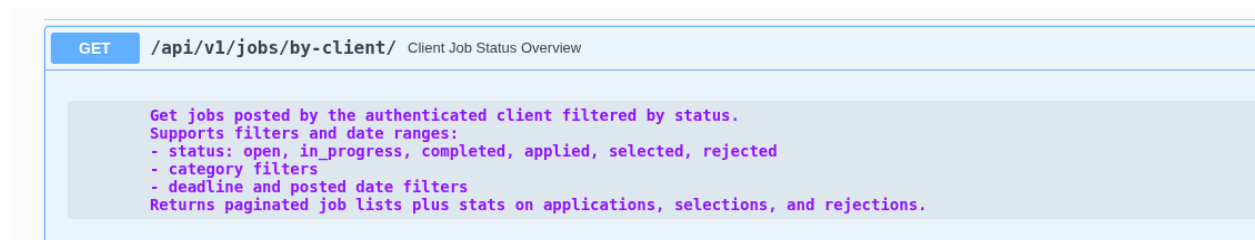
This document outlines critical discrepancies, missing functionalities, and new requests identified during the frontend development phase.

I. Previously Requested Features - Unimplemented

These features have been formally requested or discussed previously but are not yet implemented or reflected in the current API responses.

Job Application Statuses:

<http://127.0.0.1:8000/api/v1/jobs/by-client/>



The above endpoint is meant for client job listing and filtering.

1. Job Application Statuses:

- **Issue:** The API endpoint `/api/v1/jobs/{slug}/applications/` for client job applications listing does not return job application statuses (e.g., pending, accepted, rejected), preventing the display of application states on the 'Client Job Applications Listing' page.
- **Context:** This feature has been requested previously.

2. Job Application Filtering by Status:

- **Issue:** Filtering job applications by status is currently blocked, as the necessary status fields are not present in the API response (refer to point 1).
- **Context:** Directly dependent on the implementation of job application statuses.

This other endpoint <http://127.0.0.1:8000/api/v1/jobs/{slug}/applications/> shows the applications on a single job added status, but you can't use it to filter. Each user will give you the appropriate status.

Payment Flow

3. Payment Flow Documentation & Implementation Clarification:

- **Issue:** Comprehensive documentation for the payment flow is required. The frontend design incorporates UI elements for Credit Card, MPESA, and PayPal. Recent discussions have indicated that only MPESA and Pesapal integrations are currently active, with a promise of written documentation that has yet to be provided. This impedes frontend integration.
- **Context:** Repeatedly requested.

Repeatedly You asked the wrong person. I referred you to Wendy or Charles. This is how the current implementation works. Everything is handled on the backed on your side you just need to initiate the payment through a pay button and handle the redirects. As per information i have, the client has to pay the whole amount not instalments, the freelancer is handled manually.

Pass the slug of the job the client is paying for to

<http://127.0.0.1:8000/api/v1/payment/initiate/{slug}/> then initiate it:

```
$ curl -X 'GET' \
'http://127.0.0.1:8000/api/v1/payment/initiate/proofread-and-edit-marketing-emails/' \
-H 'accept: */*' \
-H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2t1bl90eXBldjoiYWNjaZmVlbnNpdWkiLCJpYXQiOjE3NDAwMjEsImp0aSI6ImhjNTc4NmBLOWZiMDQ4Zjk4ZjlhbmVudC5kZWwucyJzZjZlcnRlc3QwIiwidXNlcl9pZCI6N30.nZDNYPIcenQ6zU9H3zytF_J15be0BJJaTrsERg3Ns' | jq
```

% Total	% Received	% Xferd	Average Speed Dload Upload	Time Total	Time Spent	Time Left	Current Speed				
100	206	100	206	0	0	190	0	0:00:01	0:00:01	--:--:--	190

```
{  
  "authorization_url": "https://checkout.paystack.com/ovf971x1wmwat5n",  
  "reference": "-Dpiyr3tHUMLC6soc1EV5czEOpmIoTfwJBzV6einYPvwGOOL1dNa5w99ge9RG-0QHs",  
  "message": "Redirect failed? Click the Pay Now button."  
}
```

Then handle the redirection. Once payment is successful it redirects the client to the same job, he/she paid for. Payment is for paystack and payments if for PayPal. Same logic.

Email verification flag, Existing, added another flag on login

4. Email Verification Status Flag:

- **Issue:** The `/api/v1/auth/login/` endpoint response for user details should include a boolean flag (e.g., `is_email_verified`) indicating whether the user's email has been verified.
- **Context:** Requested previously.

Job Category Data Consistency:

The last time we talked about this we agreed you were to handle it in the frontend. This is what you have been referring to, but its no longer in use:

```
CATEGORY_CHOICES = [
    ('data_entry', 'Data Entry'),
    ('translation', 'Translation'),
    ('transcription', 'Transcription and Captioning'),
    ('graphics', 'Graphics & Design'),
    ('writing', 'Writing and Editing'),
    ('web_dev', 'Web and App Development'),
    ('project_mgmt', 'IT Project Management'),
    ('testing', 'Software Testing'),
    ('virtual_assist', 'Virtual Assistance'),
    ('social_media', 'Social Media Management'),
    ('ai_training', 'AI Model Training'),
    ('seo', 'SEO and SEM'),
    ('content_creation', 'Content Creation'),
    ('video_editing', 'Video Editing'),
    ('photo_editing', 'Photo Editing'),
    ('customer_support', 'Customer Support'),
    ('sales', 'Sales and Lead Generation'),
    ('marketing', 'Digital Marketing'),
    ('cybersecurity', 'Cybersecurity'),
    ('qa', 'Quality Assurance'),
    ('cloud_services', 'Cloud Services'),
    ('blockchain', 'Blockchain Development'),
    ('devops', 'DevOps'),
    ('database_admin', 'Database Administration'),
```

But the last update, I mentioned that we updated it to be dynamic and not rely on fixed values. Job category listing gets you the name and the ID:

```

$ curl -X 'GET' \
  'http://127.0.0.1:8000/api/v1/jobs/categories/' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBldjoiYWNjZXRzIiwiaXNjaXNzUyNzQzMjUxLCJpYXQiOiE3NTI3Mzk2NTEsImp0aSI6Ijk3NjJjZDQ4MDZhNDQ1ZWJhMDc2YjI5Mjg2OWVkb2IiwidXNlcml9pZCI6Nn0.tVdU8gANGp56crLVLi55IaESAfSmq6cV2ZopudFcBwA' | jq
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total      Spent      Left   Speed
100    264    100    264     0     0    11584      0 --:--:-- --:--:-- --:--:-- 12000
{
  "count": 7,
  "next": null,
  "previous": null,
  "results": [
    {
      "id": 1,
      "name": "data entry"
    },
    {
      "id": 2,
      "name": "data_entry"
    },
    {
      "id": 7,
      "name": "social_media"
    },
  ],
}

```

Job list view just shows the name under category_display like job detail view:

Request URL

http://127.0.0.1:8000/api/v1/jobs/respond-to-customer-emails-part-time/

Server response

Code Details

200

Response body

```

{
  "id": 7,
  "client": {
    "id": 7,
    "first_name": "john",
    "last_name": "walker",
    "username": "johnwalker",
    "location": "",
    "profile_pic": "/media/profile_pic/pexels-naimbic-2610815.jpg",
    "email_verified": true
  },
  "status": "completed",
  "title": "Respond to Customer Emails (Part-Time)",
  "category_display": "virtual assistant",
  "description": "Help respond to basic customer inquiries and forward complex ones to the team.",
  "price": "1363845.00",
  "posted_date": "2025-07-04T15:23:44.748042+03:00",
  "deadline_date": "2025-07-04T10:07:27+03:00",
  "max_freelancers": 2,
  "required_freelancers": 1,
  "skills_required_display": [
    {
      "id": 15,

```

So unless you're asking something else

Freelancer Single Job View - Client Details Enrichment:

6. Freelancer Single Job View - Client Details Enrichment:

- **Issue:** When fetching a single job for a freelancer, the client object within the response requires additional properties: location, rating, name, profile_pic (renamed from image for consistency), total_jobs_posted, total_money_spent, total_hires, and date_of_joining.
- **Affected Endpoint:** A Specific single job endpoint for freelancers.
- **Context:** A detailed JSON response format for this was previously provided to the backend developer, yet the changes are not implemented.

This is the current url for job details <http://127.0.0.1:8000/api/v1/jobs/{slug}/>

Most of the properties you're asking for are there, I've added what is missing: money_spent,, just make sure you're using the right endpoint

```
{
  "portfolio": "nilltechsolutions.com/api/v1/freelance/portfolio/copypaste/",
  "profile_pic": "/media/profile_pic/pexels-pixabay-69932.jpg",
  "email_verified": true,
  "date_joined": "2025-06-08T12:10:01Z"
},
{
  "submitted_at": "2025-07-17T09:06:29.836565+03:00",
  "extra_data": null,
  "cv": "http://127.0.0.1:8000/media/responses/cvs/samplecv_RueDJye.pdf",
  "cover letter": null,
  "portfolio": null
}
],
"client rating": 0,
"client review count": 0,
"client recent reviews": [],
"bookmarked": false,
"has applied": false,
"has applied and bookmarked": false,
"client total jobs": 3,
"application count": 2
}
```

Content type
application/json

Copy Expand all Collapse all

```
{
  "id": 0,
  "client": "string",
  "status": "open",
  "title": "string",
  "category_display": "string",
  "description": "string",
  "price": "^-?\\(?:\\.\\.\\.\\)?$",
  "posted_date": "2019-08-24T14:15:22Z",
  "deadline_date": "2019-08-24T14:15:22Z",
  "max_freelancers": -9223372036854776000,
  "required_freelancers": 0,
  - "skills_required_display": [
    - {
      "id": 0,
      "name": "python"
    }
  ],
  "preferred_freelancer_level": "entry",
  "slug": "^-$",
  "selected_freelancer": "string",
  "payment_verified": true,
  - "responses": [
    - {
      "id": 0,
      "user": "string",
      "submitted_at": "2019-08-24T14:15:22Z",
      "extra_data": null
    }
  ],
  "client_rating": "string",
  "client_review_count": "string",
  "client_recent_reviews": "string"
}
```

II. Previously Requested Features - Partially Implemented

These features have seen partial implementation, but the current API response does not fully meet the requested specifications or contains extraneous data.

1. **Client Dashboard Summary Endpoint (/api/v1/jobs/dashboard/summary/):**
 - **Issue:** The current summary response provides some of the requested fields among unrequested fields while missing the requested field, wallet_balance.
 - **Required Fields:** open_jobs, completed_jobs, wallet_balance.
2. **Freelancer Dashboard Summary Endpoint (/api/v1/jobs/dashboard/summary/):**
 - **Issue:** The current summary includes extraneous fields (e.g., total_jobs_picked) and omits the crucial new_messages_count.
 - **Required Fields:** completed_jobs, active_jobs, pending_applications, total_earnings, new_messages_count.

These properties are existing <http://127.0.0.1:8000/api/v1/jobs/dashboard/summary/>

```
$ curl -X 'GET' \
  'http://127.0.0.1:8000/api/v1/jobs/dashboard/summary/' \
  -H 'accept: */*' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2t1b190eXB1Ijo1YWNjZXNzIiwiaXNzIjoxNzYyYXNTIzLCJpYXQ0IjE3NTI3NTc5MjMsImp0aSI6ImUwMzUyNjI4YzhjMDRlYmNiM2YzMjkwYTRjOTYwZjQzIiwidXNlc19pZCI6N30uY3KUTw7e5Y01hxxSSsfaYRfLyc4bNXlczX9j6dlcJ2g' | jq

% Total    % Received % Xferd  Average Speed   Time    Time     Time
           Dload  Upload   Total     Spent    Left     Speed
100    328    100    328     0     0    7960      0 --:--:-- --:--:-- --:--:--   8000
{
  "detail": "Dashboard summary retrieved successfully.",
  "summary": {
    "job_stats": {
      "total_jobs": 3,
      "open_jobs": 1,
      "in_progress_jobs": 1,
      "completed_jobs": 1,
      "total_responses": 5,
      "total_selected": 1,
      "total_rejections": 4,
      "total_pending_responses": 3
    },
    "wallet_stats": {
      "user_type": "client",
      "total_jobs_posted": 3,
      "total_amount_spent": 2727690.0
    }
  }
}
```

Message notifications are on <http://127.0.0.1:8000/api/v1/jobs/notifications/summary/>

