

Frontend API Requirements - Freelancer Platform

This document outlines the desired API endpoints, data structures, and functionalities required by the frontend application, as derived from the design specifications. The aim is to ensure a seamless integration and enable the development of all intended UI features.

1. Authentication & User Management

1.1. Registration & Login

- **POST /auth/register**
 - **Purpose:** Register a new user account.
 - **Request Body:**
 - email: String (User's email address)
 - full_name: String (User's full name)
 - password: String (User's chosen password)
 - confirm_password: String (Confirmation of the password)
 - user_type: Enum (e.g., "client", "freelancer")
 - **Expected Response:** Success status, possibly a message. The user is then expected to verify their email.
- **POST /auth/login**
 - **Purpose:** Authenticate a user and receive session/auth tokens.
 - **Request Body:**
 - email: String (User's email address)
 - password: String (User's password)
 - **Expected Response:** Authentication token(s) (e.g., JWT access/refresh tokens), and basic user details (ID, email, user_type).
- **GET /auth/verify-email**
 - **Purpose:** Verify the user's email address via a link (contains tokens/IDs in URL parameters).
 - **Request Parameters:** uidb64, token (from verification link)
 - **Expected Response:** Success status/message for verification.

1.2. Profile Management

- **GET /profile/me**
 - **Purpose:** Retrieve the currently authenticated user's full profile details.
 - **Expected Response:**
 - id: String (User ID)
 - email: String
 - full_name: String
 - profile_photo_url: String (Optional)
 - user_type: Enum ("client" or "freelancer")
 - is_active: Boolean (Account status)
 - date_joined: DateTime
 - **(If Client):**
 - company_name: String
 - company_website: String
 - company_size: Enum (e.g., "1-10", "11-50", "51-200")
 - industry: Enum (e.g., "technology", "finance", "education", "UI/IX")
 - budget: Number (e.g., total project budget for clients)
 - preferred_languages: Array of Language Objects (id, name)
 - freelancer_skill_level: Array of Enums (e.g., "entry level", "intermediate level", "expert level")
 - **(If Freelancer):**
 - job_title: String (Primary professional title)
 - bio: String (Short biography)
 - location: String
 - years_of_experience: Number
 - availability: Enum ("full time", "part time", "remote")
 - skills: Array of Skill Objects (id, name)
 - languages: Array of Language Objects (id, name, proficiency)
 - portfolio_link: String (Primary portfolio link)
 - is_account_visible: Boolean (Toggle for public visibility)
- **PATCH /profile/me**
 - **Purpose:** Update parts of the current user's profile.
 - **Request Body:** (Partial update with fields to modify, e.g., full_name, bio, etc.)
 - **Expected Response:** Updated user profile object.
- **POST /auth/password-change**
 - **Purpose:** Change the user's password.
 - **Request Body:** old_password, new_password, confirm_new_password
 - **Expected Response:** Success status.
- **DELETE /profile/me (or /users/{id}/delete for owner)**
 - **Purpose:** Delete the user's account.
 - **Expected Response:** Success status.

2. Client-Specific APIs

2.1. Onboarding

- **POST /client/onboarding**
 - **Purpose:** Submit initial onboarding data for a new client.
 - **Request Body:**
 - company_size: Enum
 - company_name: String
 - website: String
 - industry: Enum (e.g., "technology", "finance", "education", "UI/IX")
 - budget: Number (Overall project budget)
 - language: Array of Language Objects (id, name)
 - freelancer_skill_level: Array of Enums (e.g., "entry level", "intermediate level", "expert level")
 - **Expected Response:** Success status.

2.2. Dashboard

- **GET /client/dashboard/stats**
 - **Purpose:** Retrieve key performance indicators for the client dashboard.
 - **Expected Response:**
 - open_jobs_count: Number
 - completed_jobs_count: Number
 - wallet_balance: Number

2.3. Jobs

- **POST /jobs**
 - **Purpose:** Create a new job posting.
 - **Request Body:**
 - title: String
 - category_id: String (ID of job category)
 - description: String
 - responsibilities: String (Text detailing responsibilities)
 - skills: Array of Skill Objects (id, name)
 - expiration_date: Date
 - budget: Number
 - payment_method: String (e.g., "fixed-price", "hourly")
 - **Expected Response:** Created Job Object (with id, slug, status, etc.)
- **GET /jobs**
 - **Purpose:** List job postings with filters.
 - **Request Parameters (Filters):**
 - status: Enum (e.g., "open", "in_progress", "completed", "cancelled")

- payment_method: Enum ("all", "hourly", "fixed", "per milestone")
 - location: String
 - experience_level: Enum (e.g., "entry", "intermediate", "expert")
 - category_id: String
 - search: String
- **Expected Response:** Paginated list of Job Objects. Each Job Object should include:
 - id: String
 - title: String
 - category: Category Object (id, name)
 - total_bids: Number
 - expiration_date: Date
 - skills: Array of Skill Objects (id, name)
 - description: String
 - responsibilities: String
 - is_flagged: Boolean (for admin moderation)
 - status: Enum (e.g., "open", "in_progress", "completed")
 - deadline_date: Date (Expected end date)
 - freelancer: (Optional) Freelancer Profile Object (if assigned/hired)
 - applicants: Array of Applicant Objects (for Client's jobs)
 - id: String (Application ID)
 - freelancer: Freelancer Profile Object
 - id: String
 - name: String (Full name)
 - profile_photo_url: String
 - email: String
 - bio: String
 - years_of_experience: Number
 - education_details: String (Text description of education)
 - application_date: DateTime
 - status: Enum (e.g., "pending review", "hired")
- **GET /jobs/{jobId}**
 - **Purpose:** Retrieve details for a single job posting.
 - **Expected Response:** Single Job Object (as detailed above for list, including is_flagged, status, deadline_date).

2.4. Training Materials

- **GET /jobs/{jobId}/training-materials**
 - **Purpose:** List training materials associated with a specific job.
 - **Expected Response:** Array of Training Material Objects:
 - id: String
 - title: String
 - description: String
 - attachments: Array of Attachment Objects (url, file_name, file_type)
 - video_url: String

2.5. Reviews

- **POST /reviews**
 - **Purpose:** Allow the client to create a review for a freelancer after job completion.
 - **Request Body:**
 - job_id: String
 - freelancer_id: String
 - rating: Number (1-5)
 - review_message: String

2.6. Messages

- **GET /messages/chats**
 - **Purpose:** List active chat conversations for the current user.
 - **Expected Response:** Paginated list of Chat Objects:
 - chat_id: String
 - participant1: User Profile Object (id, name, profile_photo_url, user_type)
 - participant2: User Profile Object (id, name, profile_photo_url, user_type)
 - latest_message: Message Object (preview)
 - unread_count: Number
- **GET /messages/chats/{chatId}/messages**
 - **Purpose:** Retrieve all messages within a specific chat conversation.
 - **Request Parameters:** chatId, page, page_size (optional for pagination)
 - **Expected Response:** Paginated list of Message Objects:
 - message_id: String
 - sender: User Profile Object (id, name, profile_photo_url)
 - receiver: User Profile Object (id, name, profile_photo_url)
 - content: String (Message text)
 - attachments: Array of Attachment Objects (url, file_name, file_type)
 - timestamp: DateTime
- **POST /messages/chats/{chatId}/messages**
 - **Purpose:** Send a new message within a chat.
 - **Request Body:**

- content: String
- attachments: Array of file uploads or URLs (if pre-uploaded)

2.7. Wallet

- **GET /wallet/balance**
 - **Purpose:** Retrieve the client's current wallet balance.
 - **Expected Response:**
 - balance: Number
- **GET /wallet/transactions**
 - **Purpose:** List all wallet transactions for the client.
 - **Expected Response:** Paginated list of Transaction Objects:
 - id: String (Transaction reference/ID)
 - type: Enum (e.g., "Hiring", "Deposit", "Received")
 - amount: Number
 - date: Date
 - time: Time
 - job_id: String (if related to a job)
 - from: User Profile Object (id, name, profile_photo_url, email) (if money received from user)
- **POST /wallet/deposit**
 - **Purpose:** Initiate a fund deposit into the wallet.
 - **Request Body:**
 - amount: Number
 - method: Enum ("card", "paypal", "mpesa")
 - **(If Card):** card_name_on_card, card_number, expiry_date, cvv
 - **(If PayPal):** paypal_email
 - **(If Mpesa):** phone_number
 - **Expected Response:** Success status, transaction ID.

3. Freelancer-Specific APIs

3.1. Dashboard

- **GET /freelancer/dashboard/stats**
 - **Purpose:** Retrieve key performance indicators for the freelancer dashboard.
 - **Expected Response:**
 - completed_jobs_count: Number
 - active_jobs_count: Number
 - pending_applications_count: Number
 - total_earnings: Number
 - new_messages_count: Number

3.2. Jobs

- **GET /jobs (Freelancer view)**
 - **Purpose:** List jobs with freelancer-specific filters.
 - **Request Parameters (Filters):**
 - sort_by: Enum ("best_matches", "most_recent")
 - bookmarked_by_me: Boolean (Filter for jobs bookmarked by the current freelancer)
 - category_id: String
 - subcategory_id: String
 - search: String
 - **Expected Response:** Paginated list of Job Objects. Each should include:
 - id: String
 - created_date: Date
 - total_bids: Number
 - title: String
 - budget: Number
 - duration: String
 - freelancer_skill_level: Enum
 - description: String
 - is_bookmarked_by_freelancer: Boolean (Indicates if the current freelancer bookmarked it)

- **GET /jobs/{jobId}**
 - **Purpose:** Retrieve details for a single job from a freelancer's perspective.
 - **Expected Response:** Single Job Object (as detailed above for the list), plus:
 - deadline_date: Date (Expected end date)
 - status: Enum (e.g., "open", "in progress", "completed")
 - training_materials: Array of Training Material Objects (as defined above)
 - skills: Array of Skill Objects (id, name)
 - client_details: Client Profile Object
 - profile_photo_url: String
 - name: String
 - rating: Number
 - location: String
 - total_jobs_posted: Number
 - total_money_spent: Number
 - total_hires: Number
 - date_of_joining: Date
- **POST /jobs/{jobId}/bookmark**
 - **Purpose:** Allow a freelancer to bookmark a job.
 - **Request Body:** (Implicitly uses authenticated freelancer ID and jobId from URL)
 - **Expected Response:** Success status.
- **DELETE /jobs/{jobId}/unbookmark**
 - **Purpose:** Allow a freelancer to unbookmark a job.
 - **Request Body:** (Implicitly uses authenticated freelancer ID and jobId from URL)
 - **Expected Response:** Success status.
- **GET /freelancer/bookmarked-jobs**
 - **Purpose:** List all jobs bookmarked by the current freelancer.
 - **Expected Response:** Paginated list of Job Objects (as detailed in GET /jobs freelancer view).
- **POST /jobs/{jobId}/apply**
 - **Purpose:** Submit a job application.
 - **Request Body:**
 - freelancer_skills: Array of Skill Objects (id, name) (skills relevant to this application)
 - portfolio_link: String
 - linkedin_url: String
 - years_of_experience: Number
 - cover_letter: String
 - payment_details: String (How freelancer wishes to be paid, e.g., "per hour", "fixed")
 - price: Number (Proposed bid/rate)
 - **Expected Response:** Success status, Application ID.

3.3. Wallet

- **GET /wallet/stats**
 - **Purpose:** Retrieve freelancer wallet statistics.
 - **Expected Response:**
 - pending_payout: Number
 - total_earnings: Number
- **GET /wallet/transactions**
 - **Purpose:** List all wallet transactions for the freelancer.
 - **Request Parameters (Filters):** duration_start_date, duration_end_date
 - **Expected Response:** Paginated list of Transaction Objects:
 - id: String (Reference)
 - transaction_date: DateTime
 - from: User Profile Object (id, name, profile_photo_url, email)
 - amount: Number
 - status: Enum ("pending", "paid")
- **GET /wallet/transactions/export/csv**
 - **Purpose:** Export transaction history as a CSV file.
 - **Expected Response:** CSV file download.
- **GET /wallet/invoices/{invoiceId}/download**
 - **Purpose:** Download a specific invoice as a file (e.g., PDF).
 - **Expected Response:** File download (e.g., PDF).
- **POST /wallet/setup-mpesa**
 - **Purpose:** Configure MPESA payment details for payouts.
 - **Request Body:**
 - phone_number: String
 - password: String (for confirmation)
 - **Expected Response:** Success status.

3.4. Portfolio

- **GET /portfolio**
 - **Purpose:** List a freelancer's portfolio items.
 - **Expected Response:** Array of Portfolio Item Objects:
 - id: String
 - image_url: String
 - description: String
 - role_title: String (Role/Job title on the project)
 - link_url: String (Link to live project/more details)
- **POST /portfolio**
 - **Purpose:** Create a new portfolio item.
 - **Request Body:** image_upload (File), description, role_title, link_url
 - **Expected Response:** Created Portfolio Item Object.

- **PUT /portfolio/{itemId}**
 - **Purpose:** Edit an existing portfolio item.
 - **Request Body:** image_upload (Optional File), description, role_title, link_url
 - **Expected Response:** Updated Portfolio Item Object.
- **DELETE /portfolio/{itemId}**
 - **Purpose:** Delete a portfolio item.
 - **Expected Response:** Success status.

3.5. Work Experience

- **GET /work-experience**
 - **Purpose:** List the freelancer's work experience entries.
 - **Expected Response:** Array of Work Experience Objects:
 - id: String
 - start_date: Date
 - end_date: Date (Can be null if current)
 - company_name: String
 - role_title: String
 - link_url: String (Link to company website/project)
 - description: String
- **POST /work-experience**
 - **Purpose:** Create a new work experience entry.
 - **Request Body:** start_date, end_date, company_name, role_title, link_url, description
 - **Expected Response:** Created Work Experience Object.
- **PUT /work-experience/{experienceId}**
 - **Purpose:** Edit an existing work experience entry.
 - **Request Body:** start_date, end_date, company_name, role_title, link_url, description
 - **Expected Response:** Updated Work Experience Object.
- **DELETE /work-experience/{experienceId}**
 - **Purpose:** Delete a work experience entry.
 - **Expected Response:** Success status.

3.6. Reviews (as a Reviewer)

- **POST /reviews/give**
 - **Purpose:** Allow a freelancer to give a review to a client after job completion.
 - **Request Body:**
 - job_id: String
 - client_id: String
 - rating: Number (1-5)
 - review_message: String
- **POST /reviews/request**
 - **Purpose:** Allow a freelancer to request a review from a client.
 - **Request Body:**

- job_id: String
- client_id: String
- note: String (Optional message to client)

3.7. Reviews (as a Recipient)

- **GET /reviews/my**
 - **Purpose:** List reviews received by the current freelancer.
 - **Expected Response:** Array of Review Objects:
 - id: String
 - client_details: Client Profile Object (id, name, profile_photo_url)
 - rating_given: Number
 - review_message: String
 - date: Date
- **PATCH /reviews/{reviewId}/visibility**
 - **Purpose:** Toggle the visibility of a received review.
 - **Request Body:**
 - is_visible: Boolean
 - **Expected Response:** Success status.

4. Admin-Specific APIs

4.1. Dashboard

- **GET /admin/dashboard/stats**
 - **Purpose:** Retrieve overall platform statistics for the admin dashboard.
 - **Expected Response:**
 - total_jobs_completed: Number
 - total_active_freelancers: Number
 - total_active_clients: Number
 - pending_payment_amount: Number
 - total_income: Number
 - total_earning: Number
 - total_transfer: Number
 - job_activity: Array of Job Activity Objects (recent jobs)
 - job_id: String
 - client_name: String
 - freelancer_name: String
 - start_date: Date
 - end_date: Date
 - budget: Number
 - payment_status: Enum ("pending", "completed", "overdue")
 - job_status: Enum ("pending", "in progress", "completed")

4.2. Jobs Management

- **GET /admin/jobs**
 - **Purpose:** List all jobs for admin overview.
 - **Expected Response:** Paginated list of Job Objects, including:
 - job_id: String
 - client_name: String
 - freelancer_name: String (if assigned)
 - start_date: Date
 - end_date: Date
 - budget: Number
 - payment_status: Enum
 - job_status: Enum
 - is_flagged: Boolean
- **POST /admin/jobs/{jobId}/suspend**
 - **Purpose:** Suspend a job posting due to a violation/issue.
 - **Request Body:**
 - reason: String
 - **Expected Response:** Success status.

4.3. Payments Management

- **GET /admin/payments/stats**
 - **Purpose:** Retrieve payment overview statistics for the admin.
 - **Expected Response:**
 - total_earnings: Number
 - pending_payment_amount: Number
 - pending_payment_number: Number
 - processed_payments_number: Number
 - pending_payment_percentage_vs_prev_week: Number
- **GET /admin/payments/jobs**
 - **Purpose:** List jobs with payment-specific filters.
 - **Request Parameters (Filters):**
 - payment_status: Enum ("all", "pending_payment", "completed_payment", "overdue_payment")
 - page, page_size
 - **Expected Response:** Paginated list of Job Payment Objects:
 - job_id: String
 - client_name: String
 - freelancer_name: String
 - budget: Number
 - job_status: Enum
 - payment_status: Enum
 - reference: String (Payment transaction reference)
- **POST /admin/payments/rollout**
 - **Purpose:** Initiate payment disbursement to freelancers/clients.
 - **Request Body:**
 - type: Enum ("single_client", "single_freelancer", "multiple_jobs")
 - client_id: String (Conditional)
 - freelancer_id: String (Conditional)
 - job_ids: Array of Strings (Conditional)
 - amount: Number (Conditional)
 - **Expected Response:** Success status, list of initiated transactions.
- **GET /admin/payments/history**
 - **Purpose:** List full payment transaction history for the admin.
 - **Expected Response:** Paginated list of Transaction History Objects:
 - date: Date
 - job_id: String
 - client_name: String
 - freelancer_name: String
 - amount: Number
 - job_status: Enum
 - payment_status: Enum

- reference: String

4.4. Single Job (Admin View)

- **GET /admin/jobs/{jobId}**
 - **Purpose:** Retrieve detailed information for a single job for the admin.
 - **Expected Response:** Job Object (as detailed in GET /jobs for client/freelancer) plus:
 - client_details: Client Profile Object (with profile_photo_url, name, rating, location, total_jobs_posted, total_money_spent, total_hires, date_of_joining)
 - freelancer_details: Freelancer Profile Object (if assigned, with profile_photo_url, client_rating, location, title, total_jobs_completed, date_of_joining)

4.5. Chat Management (Admin)

- **GET /admin/chats**
 - **Purpose:** List all chat conversations for admin monitoring.
 - **Expected Response:** Paginated list of Chat Objects (similar to user-level, but including):
 - client_details: Client Profile Object (id, name, profile_photo_url, is_active)
 - freelancer_details: Freelancer Profile Object (id, name, profile_photo_url, is_active)
 - latest_message: Message Object
 - messages: Array of Message Objects (for full chat history, can be paginated)

4.6. Users Management (Admin)

- **GET /admin/users/stats**
 - **Purpose:** Retrieve user statistics for the admin.
 - **Request Parameters (Filters):** date_start, date_end
 - **Expected Response:**
 - total_users: Number
 - active_clients: Number
 - active_freelancers: Number
 - suspended_accounts: Number
- **GET /admin/users**
 - **Purpose:** List all users with search and filters.
 - **Request Parameters (Filters):**
 - search_query: String (Search by name, email, etc.)
 - user_type: Enum ("freelancer", "client", "admin")
 - status: Enum ("active", "suspended")
 - page, page_size
 - **Expected Response:** Paginated list of User Objects:
 - id: String
 - name: String (Full name)
 - email: String
 - user_type: Enum

- status: Enum ("active", "suspended")
- unread_messages_count: Number
- last_message_date: DateTime
- is_suspended: Boolean
- **POST /admin/users/{userId}/suspend**
 - **Purpose:** Suspend a user account.
 - **Expected Response:** Success status.
- **POST /admin/users/{userId}/reactivate**
 - **Purpose:** Reactivate a suspended user account.
 - **Expected Response:** Success status.
- **GET /admin/users/export/csv**
 - **Purpose:** Export user list as CSV.
 - **Expected Response:** CSV file download.

4.7. Reports

- **GET /admin/reports/stats**
 - **Purpose:** Retrieve overall platform report statistics.
 - **Expected Response:**
 - total_clients: Number
 - total_freelancers: Number
- **GET /admin/reports/users**
 - **Purpose:** Get detailed user report data.
 - **Expected Response:** Paginated list of User Report Objects:
 - name: String
 - email: String
 - date_of_joining: Date
 - total_jobs: Number (Jobs associated with this user)

4.8. Admin Profile

- **GET /admin/profile/me**
 - **Purpose:** Retrieve the admin's profile.
 - **Expected Response:**
 - profile_photo_url: String
 - first_name: String
 - last_name: String
 - phone_number: String
 - email_address: String
- **PATCH /admin/profile/me**
 - **Purpose:** Update the admin's profile.
 - **Request Body:** (Partial update with fields to modify)
 - **Expected Response:** Updated Admin Profile Object.

4.9. Admin Client View

- **GET /admin/clients/{clientId}**
 - **Purpose:** View a specific client's detailed profile as an admin.
 - **Expected Response:** Client Profile Object:
 - profile_photo_url: String
 - name: String
 - is_account_verified: Boolean
 - job_history: Array of Job History Objects:
 - job_id: String
 - title: String
 - rating_received: Number
 - rating_given: Number
 - date_completed: Date
 - reviews_received: Array of Review Objects:
 - freelancer_profile_photo_url: String
 - freelancer_name: String
 - date: Date
 - verification_status: String
 - rating_given: Number
 - review_message: String

5. General Data / Lookup APIs

- **GET /languages**
 - **Purpose:** List all available languages on the platform.
 - **Expected Response:** Array of Language Objects (id, name).
- **GET /skills**
 - **Purpose:** List all available skills on the platform.
 - **Expected Response:** Array of Skill Objects (id, name).
- **GET /job-categories**
 - **Purpose:** List all job categories and their subcategories.
 - **Expected Response:** Array of Category Objects:
 - id: String
 - name: String
 - subcategories: Array of Subcategory Objects (id, name)
- **GET /notifications**
 - **Purpose:** List the user's notifications.
 - **Expected Response:** Paginated list of Notification Objects:
 - id: String
 - type: Enum (e.g., "job_application", "new_message", "payment_received")
 - message: String
 - read: Boolean
 - timestamp: DateTime
 - link_url: String (URL to navigate to when clicked)
- **POST /notifications/{notificationId}/mark-as-read**
 - **Purpose:** Mark a specific notification as read.
 - **Expected Response:** Success status.

This document represents the frontend's functional and data requirements based on the designs. Please review it, and then we can schedule a meeting to reconcile these expectations with the current backend capabilities and prioritize any necessary adjustments.