**Project Title: GigConnect: An AI-Driven Freelancing Platform for India**

**Team Members:**

* **Alwin Dharsan A M - Team Captain**
* **Gurruprasaath M K**
* **Hirtheesh V J**
* **Abhitha M**

**1. System Architecture and Design**

**1.1 Overview**

GigConnect is designed as a web application that provides a platform for freelancers and employers in India to connect. The architecture comprises multiple components hosted on Vultr’s cloud infrastructure, ensuring scalability, security, and reliability. The primary components include:

* **Frontend:** Developed with HTML, CSS, JavaScript, and ReactJS for an interactive user experience.
* **Backend:** Powered by Python’s Flask framework, handling the API endpoints and application logic.
* **Database:** PostgreSQL for secure, efficient data storage, hosting user profiles, job postings, and transactional data.
* **AI Engine:** Machine-learning models for job recommendations and freelancer matching, hosted on Vultr's infrastructure.
* **Escrow System:** A secure transaction management module integrated with payment gateways for reliable payment handling.
* **Laravel based UI:** GigConnect’ s Laravel UI offers a responsive interface with Blade templates, ensuring efficient rendering, secure interactions, and a seamless user experience for freelancers and employers.

**1.2 Detailed Project Plan**

**1.2.1Project Goals**

* Establish an All-Inclusive Freelance Marketplace for India: By offering a specific platform that takes into account regional freelancing trends and demands, GigConnect seeks to link employers with independent contractors in India.
* Use AI to Improve User Experience: Employ AI-driven insights and recommendations to assist businesses and independent contractors in locating qualified applicants and pertinent job postings.
* Assure Safe Transactions: Establish secure payment gateways and an escrow system to foster confidence and enable seamless financial transactions.
* Use Vultr's Cloud Infrastructure: To meet the technical requirements of the platform, make use of Vultr's services for scalable and secure hosting, storage, load balancing, and security.

**1.2.2Project Timeline with Milestones**

**Phase 1: Start of the Project**

* Specify the needs and parameters of the project.
* Assign team members to roles and responsibilities.
* Configure the first development environments and cloud infrastructure for Vultr.

**Phase 2: Development of Core**

* Create a module for user registration and authentication.
* Manage employer and freelancer profiles, including uploading portfolios.
* Integrate the job market so that companies can advertise openings and independent contractors can apply or receive invitations.

**Phase 3: Integration of AI and Machine Learning**

* Create and incorporate recommendation algorithms powered by AI to match candidates with jobs.
* Use Vultr's cloud computing to train early AI models for tailored suggestions.

**Phase 4: Escrow and Payment System**

* Install the escrow feature and set up a safe payment gateway.
* To guarantee secure and dependable transactions between employers and freelancers, do thorough testing.

**Phase 5: Quality Assurance and Testing**

* To find and fix usability problems, do user testing with a focus group.
* To make sure the platform remains stable when there is a heavy user load, run performance testing.

**Phase 6: Deployment and Launch**

* Complete the platform configuration on the cloud infrastructure of Vultr.
* Initiate a modest rollout after launching the platform to collect preliminary user input.

**Continued: Post-Launch Assistance and Enhancements**

* Keep an eye on platform performance and take care of any problems that crop up.
* Gather input and keep refining the platform's features and AI suggestions.

**1.2 High-Level Design**

The system architecture comprises three main layers:

**1. Presentation Layer:** User interface for freelancers and employers to interact with the platform.

**2. Application Layer:** Business logic, job matching, and API services.

**3. Data Layer:** Storage for user data, job data, transaction logs, and escrow details.

**2. Key Components and Modules**

**2.1 Freelance Job Marketplace**

* Allows employers to post job listings and browse freelancer profiles.
* Freelancers can search for job listings and apply, or receive job invites.

**2.2 Freelancer Profile and Portfolio Management**

* Profile builder for freelancers to showcase their skills, experience, and portfolio items.
* Includes rating and review system for credibility and performance tracking.

**2.3 AI-Powered Search & Analytics**

* Uses machine learning models to match freelancers with job opportunities based on skill relevance and past projects.
* Provides personalized recommendations for employers, helping them find freelancers suited to their specific requirements.

**2.4 Escrow Account System**

* Ensures secure financial transactions by holding funds until project completion.
* Integrated with payment gateways for deposits, payments, and withdrawals.

**2.5 Laravel-Based UI**

GigConnect includes a Laravel-based UI layer that complements the primary frontend, enhancing user interactions and supporting dynamic server-rendered views. Laravel serves as the backbone for building responsive, interactive components that improve the user experience across both freelancer and employer profiles. Key features include:

* **Blade Frontend Templating:** Laravel’s Blade templating engine makes it possible to create reusable and effective user interface elements, guaranteeing a unified appearance and feel throughout the platform. This supports faster page loads and seamless navigation, especially on pages with frequently updated data.
* **Integrated Authentication and Security**: An additional degree of protection is added to user data on the platform by Laravel’s integrated security capabilities, which safeguard user accounts with secure login and session management.
* **Effective Data Handling:** The Laravel UI dynamically handles data-driven pages, including real-time job notifications and user profile updates, using server-rendered views. This keeps the user experience fast and reduces load on the client-side.
* **Backend API Integration:** By rendering data straight from the server and enabling the effective handling of sophisticated UI logic, Laravel collaborates well with the Flask backend APIs. As a result, users may easily access real-time job posts, updates, and communications through a seamless, interactive interface.

**3. API Documentation**

**3.1 User Registration and Login**

**Endpoint:** /api/register

**Method:** POST

**Parameters:**

* username (string)
* email (string)
* password (string)

**Response:** JSON object with user details or error message.

**3.2 Job Posting**

**Endpoint:** /api/postJob

**Method:** POST

**Parameters:**

* title (string)
* description (string)
* skills\_required (array)
* budget (number)

**Response:** Confirmation of job posting or error message.

**3.3 Job Recommendations**

* **Endpoint:** /api/recommendJobs
* **Method:** GET
* **Parameters:** userId (integer)
* **Response:** List of recommended job postings based on user profile.

**4. Setup and Usage Instructions**

**4.1 Prerequisites**

* Python 3.8+
* PostgreSQL
* Node.js (for frontend dependencies)
* Access to Vultr cloud services

**4.2 Installation Steps**

**1. Clone Repository**

bash

git clone https://github.com/your-repo/gigconnect.git

cd gigconnect

**2. Install Backend Dependencies**

bash

pip install -r requirements.txt

**3. Database Setup**

Configure PostgreSQL database.

Run initial migrations using Flask-Migrate.

**4. Frontend Setup**

bash

cd frontend

npm install

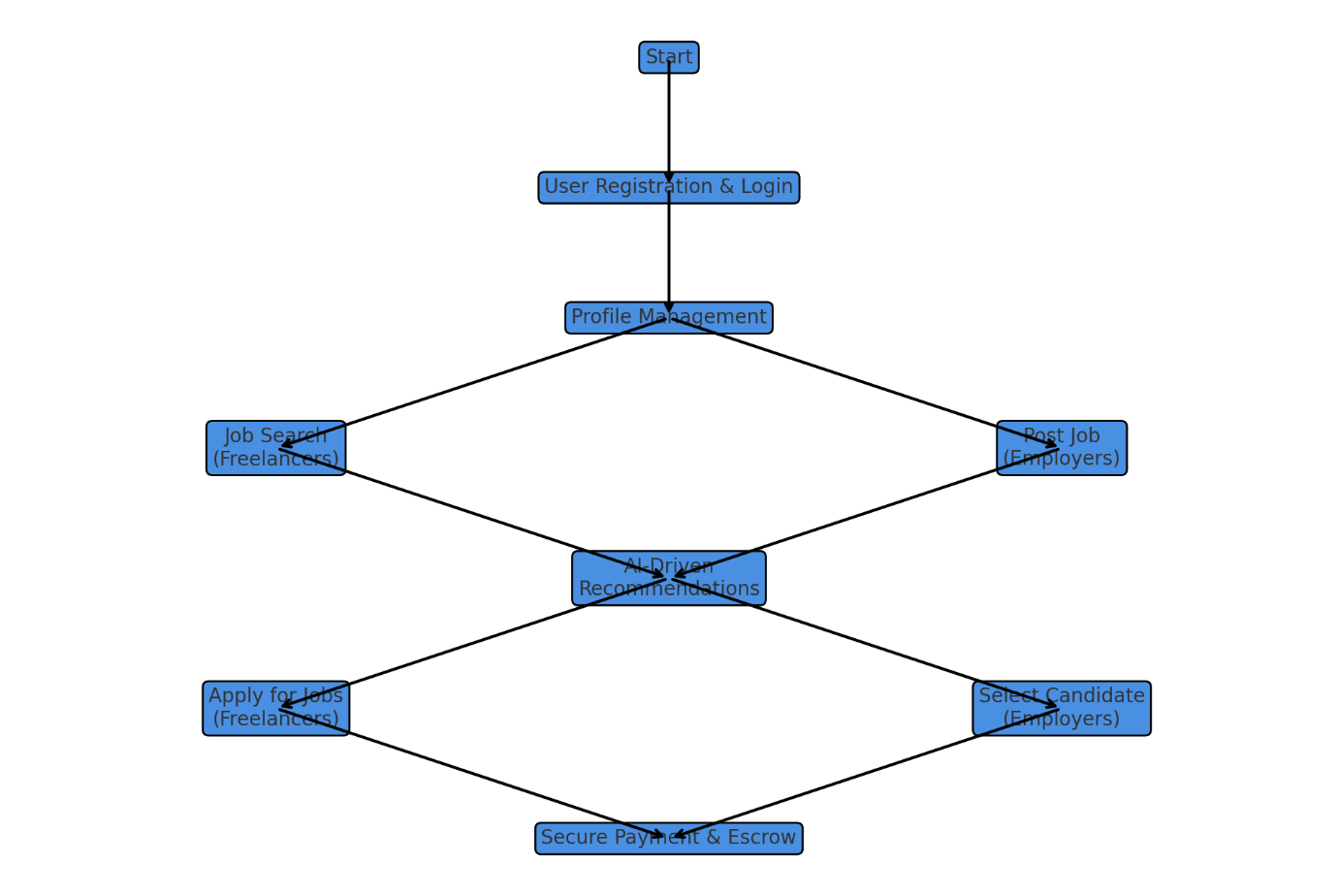
npm start

**5. Running the Application**

* Start the backend server: python freeback.py
* Access the platform via http://127.0.0.1:5000

**4.3 Usage**

* **Employer:** Post jobs, view applications, and select freelancers.
* **Freelancer:** Apply for jobs, manage profile, and receive feedback.



**1. Home Page**

* **Purpose**: The main landing page for all users, showcasing the platform’s services, available job listings, and popular freelancers.
* **Key Sections**:
  + **Job Listings**: Highlights the latest and featured jobs available on the platform.
  + **Featured Freelancers**: Displays profiles of top-rated freelancers based on skills, job success, and ratings.
  + **Search Bar**: Allows users to search for jobs or freelancers by skills, industry, or keywords.
  + **Navigation Links**: Quick links to popular categories like “Graphic Design,” “Web Development,” “Content Writing,” etc.
* **User Actions**:
  + Browse jobs or freelancer profiles.
  + Navigate to login, registration, or other key pages.

**2. Freelancer Profile Page**

* **Purpose**: A dedicated page for freelancers to showcase their skills, portfolio, and experience.
* **Key Sections**:
  + **Profile Information**: Includes freelancer name, profile picture, location, bio, and hourly rate.
  + **Skills**: Lists specific skills with proficiency levels or endorsements from clients.
  + **Portfolio**: Showcases completed projects with descriptions and visuals.
  + **Ratings and Reviews**: Displays feedback from previous clients to help new employers evaluate.
  + **Availability Status**: Allows freelancers to set their current availability (e.g., Open for work, Busy).
* **User Actions**:
  + Edit or update profile details.
  + Upload new portfolio items.
  + Update skills and availability status.

**3. Job Details Page**

* **Purpose**: Provides a detailed view of individual job listings posted by employers.
* **Key Sections**:
  + **Job Description**: Detailed information about the job requirements, responsibilities, and desired qualifications.
  + **Budget and Payment Terms**: Specifies the payment range, terms (hourly or fixed), and other financial details.
  + **Employer Profile**: Brief information about the employer, including rating, past job postings, and reviews from freelancers.
  + **Application Options**: Button for freelancers to apply or submit proposals.
  + **Related Jobs**: Suggestions for similar job listings that may interest the freelancer.
* **User Actions**:
  + Apply for the job by submitting a proposal.
  + Save the job listing for later viewing.

**4. Employer Dashboard**

* **Purpose**: A centralized hub for employers to manage their job postings, applications, and active projects.
* **Key Sections**:
* **Posted Jobs**: Lists all jobs the employer has posted with current status (e.g., Open, In Progress, Completed).
* **Applications Management**: View and respond to applications from freelancers, shortlist candidates, or reject proposals.
* **Project Tracking**: Monitor the status of ongoing projects with assigned freelancers, deadlines, and deliverables.
* **Financial Overview**: Track budgets, payments made to freelancers, and pending invoices.
* **Notifications**: Receive alerts for new applications, project updates, and messages from freelancers.
* **User Actions**:
  + Post a new job or update existing job listings.
  + Review and manage freelancer applications.
  + Track progress on ongoing projects and make payments upon completion.

**5. Freelancer Dashboard**

* **Purpose**: A workspace for freelancers to manage job applications, ongoing projects, and completed work.
* **Key Sections**:
  + **Applications**: Track all submitted applications and their status (e.g., Awaiting Response, Accepted, and Rejected).
  + **Active Contracts**: View details of active jobs, deadlines, and payment schedules.
  + **Project Milestones**: Track progress and submit completed milestones for employer approval.
  + **Earnings Summary**: Display of total earnings, pending payments, and withdrawn amounts.
  + **Notifications**: Alerts for new messages, job status changes, and milestone approvals.
* **User Actions**:
  + Apply to jobs and monitor application status.
  + Manage active projects, communicate with clients, and submit deliverables.

**6. Messaging Center**

* **Purpose**: Enables direct communication between freelancers and employers for job discussions and project updates.
* **Key Sections**:
  + **Conversation List**: A list of all active conversations, organized by most recent messages.
  + **Chat Window**: Real-time messaging interface to communicate and share files.
  + **Project-Specific Conversations**: Separate threads for each project to keep discussions organized.
* **User Actions**:
  + Send and receive messages in real time.
  + Share files, documents, or links related to the project.

**7. AI-Powered Search & Recommendations**

* **Purpose**: Helps users find relevant jobs, freelancers, or resources based on their profiles and activity.
* **Key Features**:
  + **Job Recommendations**: Personalized job suggestions for freelancers based on skills, previous applications, and market trends.
  + **Freelancer Recommendations**: Suggested freelancer profiles for employers, filtered by skills, rates, and success rates.
  + **Skill-Based Filtering**: Allows detailed search for specific skillsets or job categories.
* **User Actions**:
  + Use search filters to find specific job listings or freelancer profiles.
  + Review AI-generated recommendations and apply for suggested roles or projects.

**8. Payments & Escrow Service**

* **Purpose**: Ensures secure and transparent payment transactions between employers and freelancers.
* **Key Sections**:
  + **Escrow Setup**: Employers deposit funds into escrow for each project to secure payments.
  + **Milestone Payments**: Enables partial payments upon completion of set milestones.
  + **Withdrawal Options**: Freelancers can withdraw their earnings after project completion and employer approval.
  + **Payment History**: Record of all transactions for both freelancers and employers.
* **User Actions**:
  + Employers fund escrow for projects, and freelancers receive payments upon milestone completion.
  + Freelancers withdraw earnings using preferred payment methods.

**9. Admin Dashboard**

* **Purpose**: Manages the overall platform, monitors user activity, and handles disputes and support requests.
* **Key Sections**:
  + **User Management**: View and manage freelancer and employer accounts, including profile verifications.
  + **Job Management**: Oversee posted jobs and ensure compliance with platform guidelines.
  + **Dispute Resolution**: Address conflicts between freelancers and employers, including payment disputes.
  + **Analytics and Reporting**: Generate insights on platform usage, popular categories, and financial statistics.
* **Admin Actions**:
  + Suspend or restrict accounts in case of violations.
  + Resolve disputes and process refunds if necessary.
  + Monitor platform health and user engagement metrics.

**GitHub Repository:**