This document outlines the functional and non-functional requirements for a Freelancer Job Advertisement Portal, a platform that connects freelancers with clients seeking freelance services. The portal will allow clients to post projects, freelancers to bid on projects, and both parties to manage their work and payments.

1. Introduction

Purpose: The Freelancer Job Advertisement Portal is a web-based platform designed to facilitate freelance work by connecting clients with freelancers. It enables clients to post projects, freelancers to bid on them, and both parties to collaborate and manage payments.

Scope: The portal will support project posting, bidding, collaboration, and payment processing for freelancers and clients.

Target Audience: Freelancers, Clients, and Administrators.

2. User Roles

Clients:

- Create and manage profiles.
- Post projects with detailed descriptions, budgets, and deadlines.
- Review freelancer bids and select freelancers.
- Manage ongoing projects and payments.

Freelancers:

- Create and manage profiles (including portfolios and skills).
- Search and bid on projects.
- Submit work and receive feedback.
- Manage earnings and invoices.

Administrators:

- Manage user accounts (clients and freelancers).
- Moderate projects and bids.
- Generate system reports and analytics.

3. Functional Requirements

User Authentication and Authorization:

- Users can register and log in (clients, freelancers, and administrators).
- Role-based access control (RBAC) to restrict access to specific features.

Client Features:

- Post projects with details (title, description, budget, deadline, etc.).
- Review freelancer profiles and bids.
- Select freelancers and manage contracts.
- Track project progress and release payments.

Freelancer Features:

- Create and update profiles (including skills, portfolios, and rates).
- Search and bid on projects.
- Submit work and communicate with clients.
- Manage invoices and track earnings.

Administrator Features:

- Approve or reject projects and bids.
- Manage user accounts (suspend, delete, or update).
- Generate system reports (e.g., number of projects, bids, earnings, etc.).

Search and Filtering:

- Advanced search functionality for freelancers (skills, location, rates, etc.).
- Clients can search for freelancers based on skills, ratings, and portfolios.

Notifications:

- Email and in-app notifications for new projects, bids, messages, and payment updates.

Payment Processing:

- Integration with payment gateways (e.g., PayPal) for secure transactions.
- Escrow system for holding payments until project completion.

Reporting and Analytics:

- Clients can view project statistics.
- Freelancers can view earnings and performance metrics.
- Administrators can generate system-wide reports.

---

4. Non-Functional Requirements

Performance:

- The system should handle up to 10,000 concurrent users.
- Project search results should load within 2 seconds.

Security:

- Secure user authentication (two-factor authentication).
- Data encryption for sensitive information (e.g., payment details, personal data).
- Regular security audits and vulnerability assessments.

Scalability:

- The system should scale to support increasing numbers of users and projects.

Usability:

- Intuitive and user-friendly interface.
- Mobile-responsive design.

Availability:

- 99.9% uptime with minimal maintenance downtime.

Compliance:

- Adherence to data protection regulations.

---

5. System Architecture

Frontend: Responsive web application (HTML, CSS, JavaScript,).

Backend: RESTful API (Python, or Java).

Database: Relational database (e.g., MySQL, SQL) for storing user data, projects, bids, and payments.

Hosting: Cloud-based hosting .

Third-Party Integrations

- Payment gateway (e.g., PayPal).
- Email service for notifications.
- Social media login (e.g., Google, LinkedIn).

6. Use Cases

Client Use Case

1. Client logs in.
2. Client posts a new project.
3. Client reviews bids and selects a freelancer.
4. Client manages the project and releases payment upon completion.

Freelancer Use Case

1. Freelancer registers and logs in.
2. Freelancer searches for projects and submits bids.
3. Freelancer communicates with the client and submits work.
4. Freelancer receives payment and feedback.

Administrator Use Case

1. Administrator logs in.
2. Administrator moderates projects and user accounts.
3. Administrator generates reports.

7. Assumptions and Constraints

- Assumptions:
  - Clients and freelancers have access to the internet.
  - Users will provide accurate information during registration.
- Constraints:
  - Limited budget for development and maintenance.
  - Timeline for initial launch is 3 weeks.

8. Future Enhancements

- Integration with LinkedIn for profile importing.
- AI-based project matching for freelancers.
- Mobile application for iOS and Android.
- Multilingual support.
- Dispute resolution system for conflicts between clients and freelancers.

9. Glossary

- Project: A task or job posted by a client.
- Bid: A proposal submitted by a freelancer for a project.
- Escrow: A payment system where funds are held until project completion.