**Chapter 4: System Analysis and Design**

**4.1 Introduction**

This chapter provides a comprehensive explanation of the various components of the system, identifying the requirements as well as the overall structure of the proposed system. Additionally, this chapter will discuss the development methodologies throughout the project, explaining the rationale behind their selection. The methodologies will be illustrated using a combination of system analysis diagrams and system design diagrams. System analysis involves a detailed examination of the system’s components, interactions to identify functional and non-functional requirements. The diagrams drawn under the system analysis include a use case diagram, an activity diagram and a sequence diagram. System design is the process of conceptualising and specifying the overall architecture of the system to meet identified requirements and resolve any issues uncovered during analysis.

**4.2 System Requirements**

The functionalities required in the system to meet user specifications include:

**4.2.1 Functional Requirements**

1. Authentication module

The authentication module contains pages such as login, registration and password reset. The informal workers and employers will be required to register and authenticate to gain access to the system. Registration requires the users to key in their valid email address, password and full name in the input field provided. The system sends an email with a verification link to the user, which the user can click to be granted access to the system, after which they can log in to the system. In case the user forgets their password, they can reset it using their email.

1. Worker Profile Management Functionality

After signing up, the informal workers are prompted to fill out their profile details to access full platform features. This entails keying in their full name and contact information, selecting pertinent service categories, adding a list of skills they can provide, uploading certifications, specifying their location, prior job history and their availability. As part of the identity verification process, they must upload a valid government-issued national ID or other appropriate KYC verification documents. The informal worker earns a Verified Trust badge, which is displayed on their profile as a testament to their authenticity and reliability, permitting them to apply for jobs. The informal workers can update their information as needed.

1. Employer Profile and Job Posting

Upon registration, employers are prompted to complete their profiles by providing their full name, contact information and uploading a valid government-issued ID for KYC verification. Once verified, the employer receives a trust badge allowing them to post job openings on the platform, which include the job title, job description, location, required skills, duration and terms of payment.

1. Job matching

To facilitate efficient and focused job discovery, the job postings are displayed to workers who meet the predetermined criteria, such as relevant skills and geographical proximity. To guarantee dependable connections, it also considers the workers' ratings, trust status, and availability. Workers are also given tailored job recommendations based on their preferences and skill set, which makes it simpler for them to locate opportunities that suit them.

1. Administration module

Platform administrators are in complete control of user account management, content supervision, and dispute or report handling. They manage the skills advancement programs by keeping tabs on course offerings and user progress, moderate reviews and job postings to ensure quality, and review and approve KYC documents to guarantee user verification. Admins can also monitor platform performance and user activity by accessing comprehensive analytics and insights.

1. Review and Rating functionality

After a job is completed, employers can rate and review the workers they hired, and workers can also rate their employers. Each user's overall Trust Score, which reflects their dependability and professionalism, is compiled from these ratings and reviews. Trust Badges are given to verified users who have high Trust Scores. These badges are prominently displayed on their profiles to help build trust and credibility in the community.

1. Skills advancement module

Workers can sign up for short courses and training programs to improve their skills. These courses teach a wide range of things, from technical skills like plumbing and tailoring to important soft skills like how to deal with customers. After finishing a course, workers get digital certificates and skill badges that they can show off on their profiles. The platform also keeps track of how much each worker has learned and suggests courses that will help them keep growing in their careers.

**4.2.2 Non-Functional Requirements**

1. System Security

The platform must keep user data safe by only letting authorised users and administrators access it. Basic login information should keep user accounts safe, and only verified users should be able to do things like post jobs or edit their profiles. Admin access is role-based, ensuring only approved staff can manage users or review KYC documents. To help with accountability and resolving disputes, all important actions on the platform should be logged.

1. Usability

The platform must offer a user-friendly and intuitive interface to accommodate users with diverse levels of digital literacy. Additionally, the platform should be learnable and memorable, so returning users can easily pick up where they left off without confusion. Common tasks like setting up profiles, posting jobs or job applications should be straightforward, requiring little to no training. Users should be guided through each step by clear labels, prompts, visual feedback, and confirmation messages, which make the process seamless and free of frustration.

1. Performance

The system should respond quickly to any user action. It should be capable of supporting many concurrent users, processes and requests without experiencing any performance degradation.

1. Scalability

As adoption spreads throughout various regions, the platform should be able to accommodate horizontal scaling to manage increased traffic, job postings, and user registrations.

1. Maintainability

The code and configuration of the system should be clear, well-documented, and simple to update and maintain. Without the need for technical assistance, administrators ought to be able to oversee user reports, courses, and platform content. Debugging and system health monitoring should be supported by logging and error tracking.

1. Reliability

Under both typical and extreme circumstances, the platform should function reliably. As long as usage stays below the real-time database's current specified limit, the system should function flawlessly. Job posting, KYC processing, and authentication are examples of critical services that require high fault tolerance, with backup and recovery plans in place in case of failure.

1. Privacy

User information, including personal and identity documents, must be stored securely and only accessible to authorised admin personnel. The platform should comply with the relevant data protection laws. Users should be able to manage their personal information and understand how it is used.

## **4.3 System Analysis**

This section presents the system analysis using OOAD-based diagrams to understand the system functionality and object interactions. Each diagram helps in visualising system operations before development begins.

### **4.3.1 Use Case Diagram**

The use case diagram provides a high-level overview of how different users interact with the platform. It identifies three primary actors: job posters, informal workers, and system administrators. Job posters can create job listings, view applications, select workers, and rate completed tasks. On the other hand, informal workers can browse available jobs, apply for them, message clients, and receive feedback based on their performance. Administrators manage the system by overseeing user accounts, moderating content, and ensuring the smooth operation of all functionalities. It ensures that the needs of each actor are adequately captured and addressed in the system’s functionality.

A diagram of a workflow

AI-generated content may be incorrect.

Figure 4. 1Use Case Diagram

### **4.3.2 Activity Diagram**

The activity diagram focuses on modelling the workflows that occur within the system, particularly those initiated by the users. It provides a visual representation of the sequence of actions taken to complete various tasks, such as the process followed by a client to post a job and subsequently hire a worker. This includes steps like logging in, creating a job description, selecting a category, reviewing applications, and confirming hiring. This diagram is essential for understanding how users interact with the system in real-world scenarios and helps identify process bottlenecks or redundancies early in the design process.

A diagram of a diagram

AI-generated content may be incorrect.

Figure 4. 2 Activity Diagram

### **4.3.3 Sequence Diagram**

The sequence diagram captures the interaction between objects in the system in a time-ordered fashion. It emphasizes the communication flow during specific processes, such as job matching or real-time messaging. For instance, during job matching, the diagram illustrates the steps from when a worker applies to when the system notifies the job poster and the final decision is made. It includes the interactions between the User Interface and Controller. By visualizing these sequences, this diagram aids in understanding the logical flow of operations, identifying the roles of various system components, and ensuring synchronous communication between them.

A grid with many lines and text

AI-generated content may be incorrect.

Figure 4. 3 Sequence Diagram

### **4.3.4 Class Diagram**

It outlines the primary classes involved in the system, including User, Job, Review, and job poster. Each class is defined with its respective attributes for example, the User class includes attributes like userID, name, email, and role and relevant methods that define behaviours such as login(), and updateStatus(). Relationships among classes are also shown to demonstrate how objects collaborate within the system. For instance, a job poster can create multiple Job instances, and a Job can have multiple Reviews associated with it. This diagram serves as a blueprint for the actual coding process, ensuring consistency in the structure and behaviour of system components.

A diagram of a software application

AI-generated content may be incorrect.

Figure 4. 4 Class Diagram

### **4.3.5 System Sequence Diagram**

The system sequence diagram provides a more detailed look into specific interactions initiated by users and the system’s corresponding responses. It breaks down user actions like posting a job or accepting a task into a series of operations, each aligned in temporal order. For example, when a user submits a new job posting, the diagram outlines how the system captures the input and validates the information. It ensures that every event or action has a clearly defined response from the system. The diagram also supports the identification of necessary system functions and services that must be implemented to support user workflows.

A diagram of a job platform

AI-generated content may be incorrect.

Figure 4. 5 System Sequence Diagram

### **4.3.6 Entity Relationship Diagram (ERD)**

It defines key data entities such as Users, Jobs, and Reviews, along with their attributes and the relationships between them. For example, the ERD may show that each job is posted by one User and may be reviewed by another User after completion. The relationships, such as one-to-many or many-to-many, guide the creation of foreign keys and associative tables in the database. By visually modelling these relationships, the ERD helps us understand the data architecture, enforce referential integrity, and optimise queries for performance. It is an indispensable reference during the implementation of the backend database.

A diagram of a graph

AI-generated content may be incorrect.

Figure 4. 6 Entity Relationship Diagram

## **4.4 System Design**

This subsection describes the design of the system based on the OOAD methodology, using structural diagrams to define how the system components will be organised and interact.

**4.4.1 Database Schema**

The database schema represents a structured design for a job platform that connects informal workers with job opportunities. At the core is the User entity, which stores general user details such as name, email, phone, and profile information. Users can have specialized roles, including Informal Worker, Job Poster, and Admin, each extending the User entity with additional fields relevant to their functions. The Job entity represents tasks posted on the platform and contains information such as title, description, location, period, pricing, status, and associated categories. Jobs are linked to the User who created them and are also associated with reviews through the Review entity, which captures feedback including rating and comments. The Booking entity manages job reservations, linking jobs to clients (informal workers) and booking date, total price, and status. The Category entity classifies jobs for easier discovery. Admins manage users and set platform terms, while job posters maintain a list of their jobs. The schema provides a comprehensive and normalized structure to support user management, job posting, booking, reviewing, and categorization within the platform.

**A graph with text and a graph

AI-generated content may be incorrect.**

**4.4.3 System Architecture**

This section describes the way the system will appear to the users as they access the features relating to the application. It shows how each page in the web application will appear from the landing page to the job posting page. It clearly outlines each page's functions as well as the design for each function that is carried out on the page. It helps the users understand the main functions of the web application to increase accessibility of the web application to as many informal workers as possible.

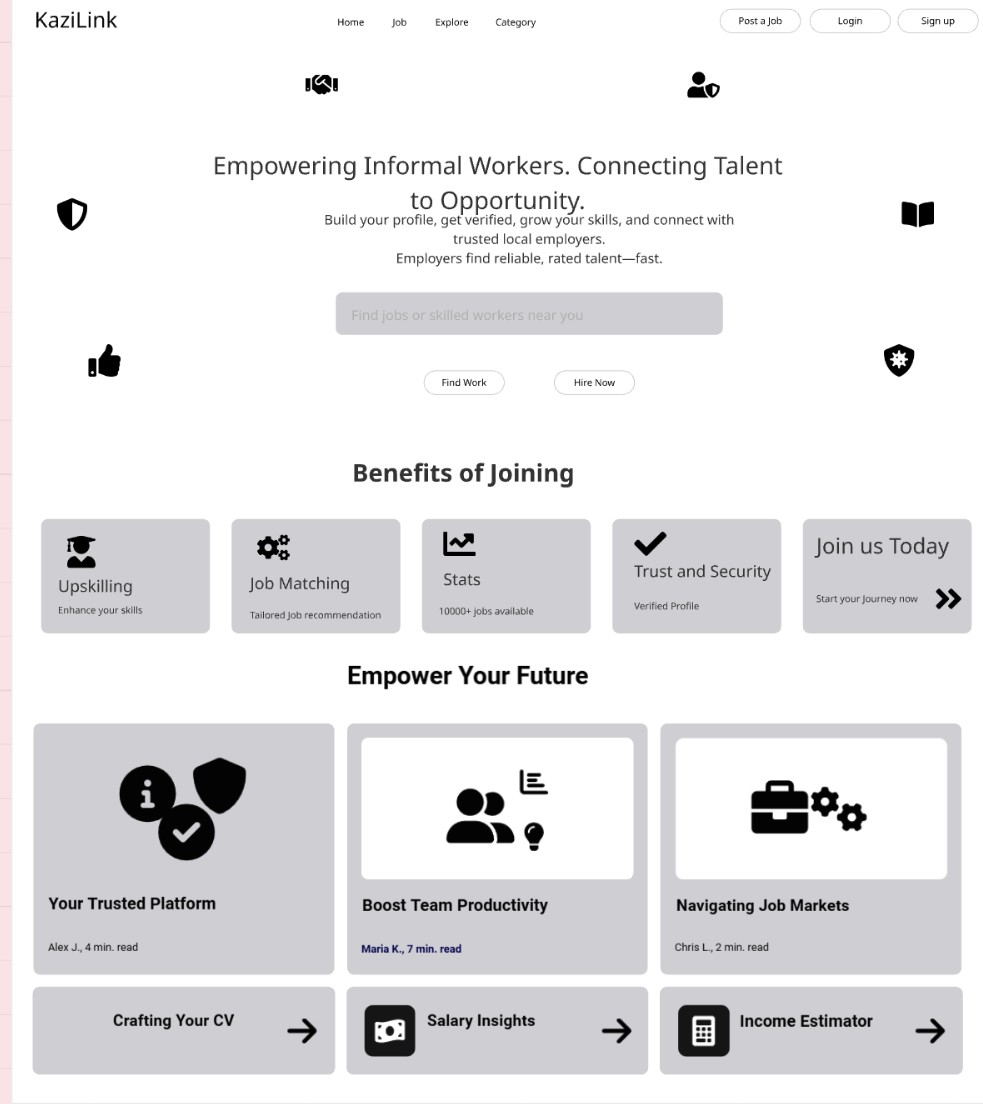
A diagram with many squares

AI-generated content may be incorrect.

**4.4.2 Wireframes**

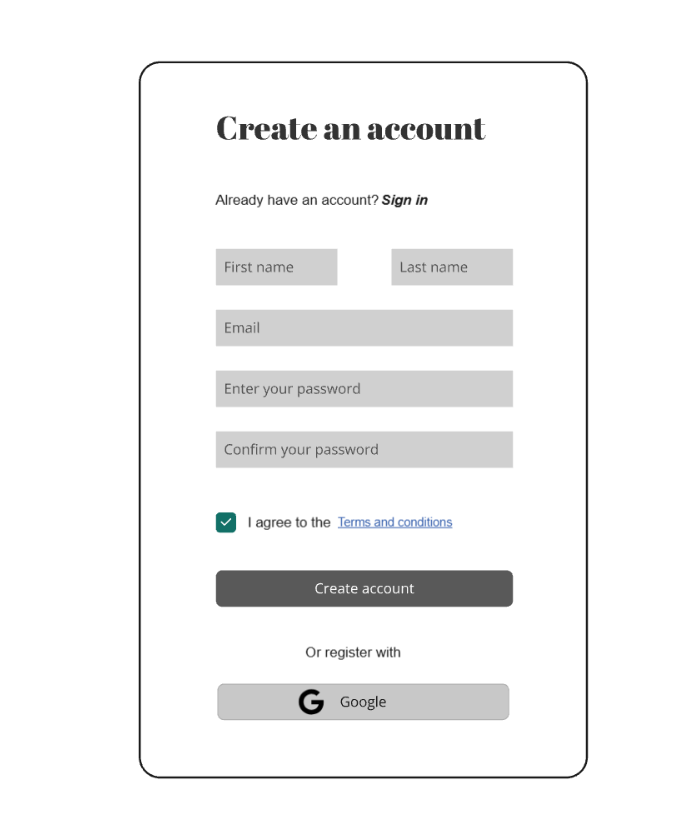
1. Landing page

The landing page serves as the welcoming interface of the platform. It includes a hero section with a compelling tagline and a call-to action button that the directs the user to the join the platform as an employer or a worker. Additionally, there is a “Empower your future section” that outlines the how the platform supports informal workers in skills advancement as well as a section highlighting the benefit of joining the platform.

****

1. Login and sign-up page

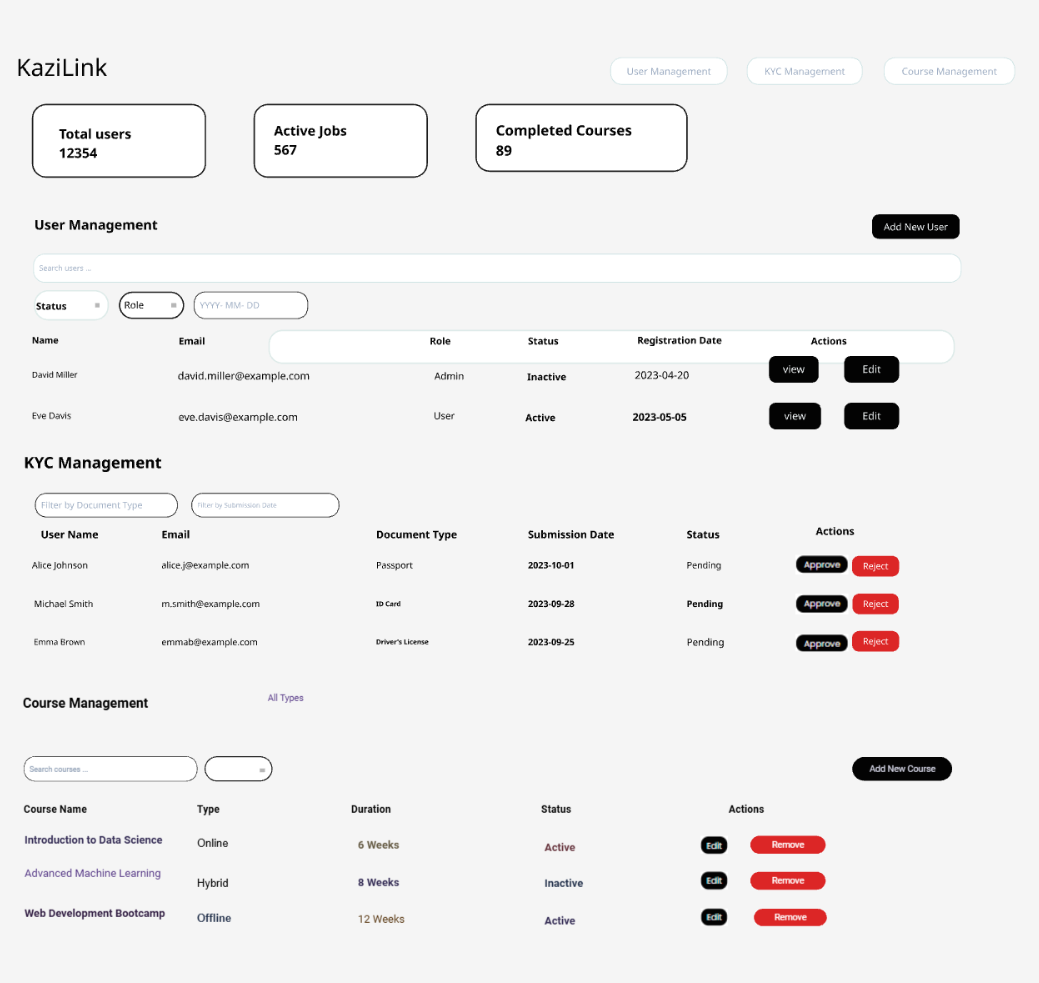
Workers and employers share a uniform login and sign-up interface to ensure a seamless user experience. Name, email and password input fields are included in the sign-up form, and there is also a quick registration option that uses Google authentication. Users receive an email with a verification link after completing the form. Clicking this link takes them to the login page and verifies their account. Users can then use Google Sign-in or their email address and password to log in. A user is automatically redirected to the account creation page to register if they attempt to log in without having an existing account.

**A screen shot of a login form

AI-generated content may be incorrect.**

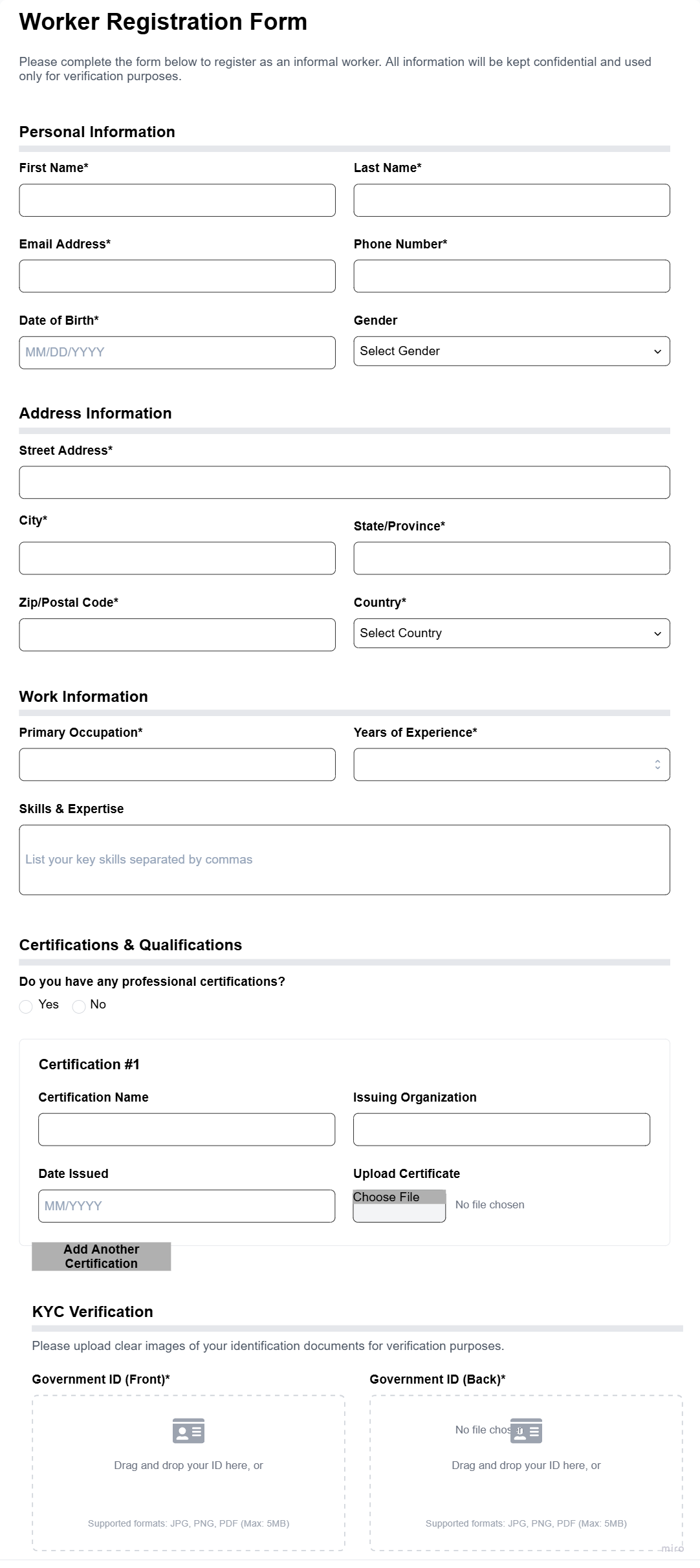
1. Admin Dashboard

The admin dashboard is a functional interface designed for managing the platform. It has panel for user management, course management, KYC management as well as the statistics for the number of users, active jobs and completed courses. Each of the sections provide detailed views and moderation actions like confirming the user profiles and monitoring user behaviour. The dashboard also facilitates complete update and delete operations, providing administrators with the tools they need to effectively manage records and uphold platform standards.

****

1. Profile setup forms

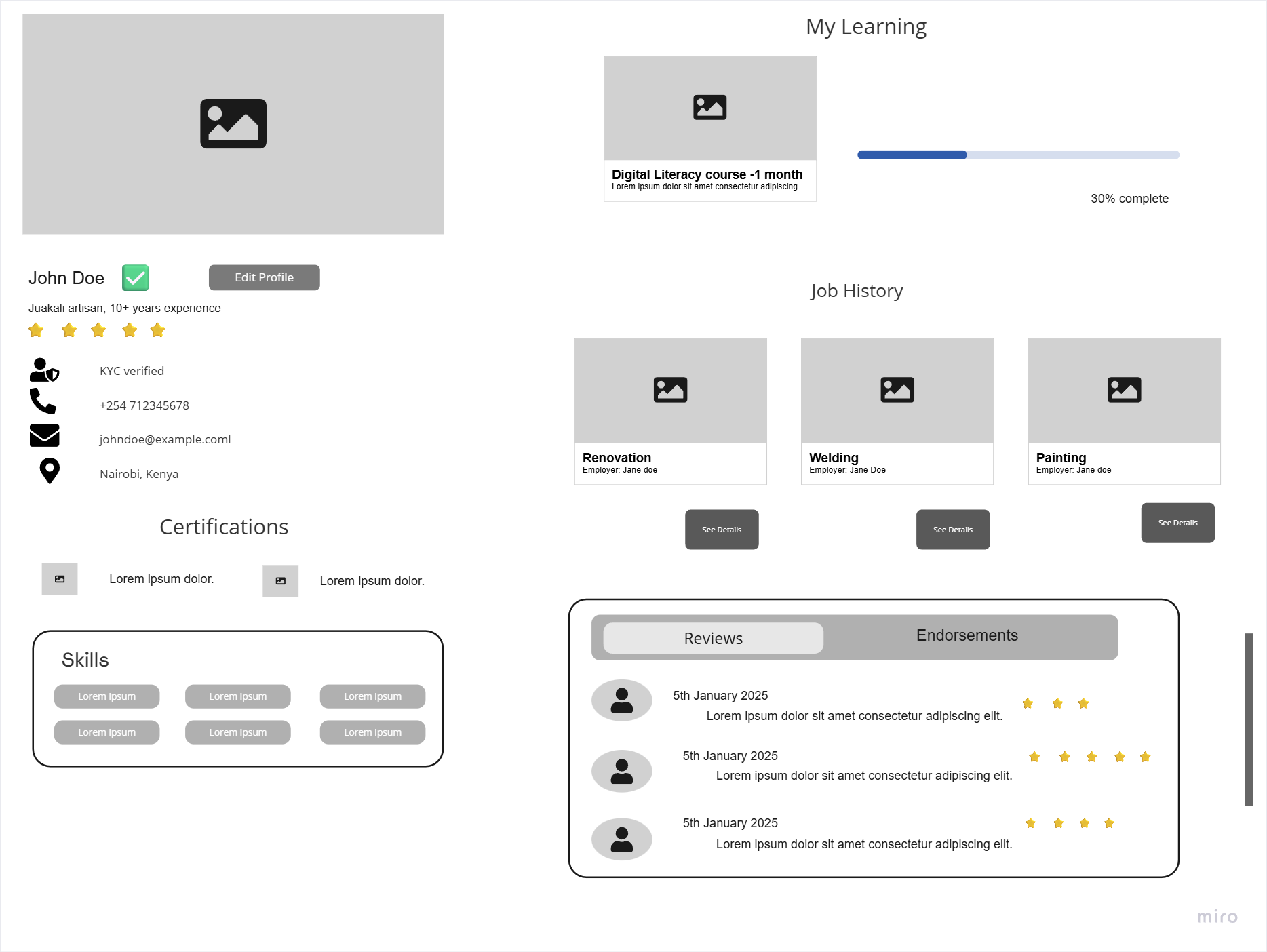
Once the user is logged in, they have the option to set up their profile as an employer or as a worker. The worker set up form collects basic information such as name, age, contact information, experience, skills, location, availability, certification, and ID verification with upload. The employer registration form is designed to onboard individuals or businesses looking to hire informal workers. Input fields for basic information like full name, phone number, email address, and ID verification with upload are included.

A screenshot of a profile

AI-generated content may be incorrect.

1. Worker profile Page

Once the Worker adds their details in the worker profile set up form, they are presented with a professional profile dashboard, displaying important data like the user's name, profile picture, contact information, location, experience level, and verification status. It includes section showcasing their certifications, skills, employment history with previous employers, ongoing learning progress and a panel of reviews and endorsements. The layout is intended to provide prospective employers with a clear and well-structured overview of the worker’s credentials, reputation, and work history.



1. Job posting page

After login, employers are permitted to post a new job. The job posting page includes details such as title, location, type, experience level, pay range, remote work options, job description, cost and particular requirements. Employers can also specify how applications should be received either through email or an external URL in the "Application Settings" section. They can also choose to include screening questions and request.

