JOBS BRIDGE

M HUNAIN TARIQ
01-134221-034
SUMAYA ZAMAN
01-134221-101

Supervisor: Dr Sabina Akhtar

Bachelor of Science in Computer Science

Department of Computer Science
Bahria University, Islamabad

December 2025

Certificate

We accept the work contained in the report titled "Jobs Bridge", written by Mr. Muhammad Hunain Tariq and Ms. Sumaya Zaman as a confirmation to the required standard for the partial fulfillment of the degree of Bachelor of Science in Computer Science.

Approved by:

Supervisor: Dr. Sabina Akhtar

_____

Internal Examiner: Name of Internal Examiner (title)

_____

External Examiner: Name of External Examiner (title)

_____

Project Coordinator: Mr. Asim Ali

_____

Head of Department: Dr. Khuram Ehsan

_____

December, 2025

Abstract

Unemployment and inefficient access to skilled workers are major socio-economic problems, especially in developing regions. Traditional recruitment procedures for blue-collar and skilled workers, including electricians, plumbers, carpenters, masons, and technicians, are largely informal, time-consuming, and unreliable. Employers face difficulties in finding verified and available workers, while skilled individuals struggle to secure regular employment. This disconnect leads to project delays, income instability, and reduced economic productivity.

In order to address these challenges, the Jobs Bridge project focuses on developing a web-based platform that digitizes the process of connecting skilled workers with employers seeking reliable services. The system allows workers to create verified profiles highlighting their skills and experience, while employers can post job requirements, browse worker profiles, and directly hire suitable candidates. The platform supports role-based access for workers, employers, and administrators, ensuring secure authentication, worker verification, and efficient job management.

Jobs Bridge also incorporates intelligent features such as job history tracking, a review and rating system, service price estimation, and multilingual support, including Urdu, to enhance usability and trust. In addition, training resources such as instructional videos are provided to help workers understand system usage, improving accessibility for users with limited technical expertise.

By digitizing the skilled worker recruitment process, Jobs Bridge aims to improve hiring efficiency, promote fair employment opportunities, reduce reliance on informal networks, and support economic empowerment. The platform enhances communication transparency and establishes a reliable connection between skilled workers and employers, contributing to workforce development and sustainable employment growth.

Acknowledgments

We extend our heartfelt gratitude to Allah Almighty for blessing us with the wisdom, strength, and opportunity to undertake this meaningful project, Jobs Bridge. His divine guidance has been our anchor in overcoming challenges and refining our vision to empower Workers through technology.

We are very grateful to our family, who gave us the strength and courage and supported us financially and morally throughout our educational careers. We are humbly thankful to our supervisor, Dr. Sabina Akhtar, for her invaluable mentorship, technical expertise, and unwavering belief in our mission. Her patience and encouragement during moments of uncertainty have been pivotal to our progress, and we are honored to work under her guidance

This journey would not have been possible without the collective support of these pillars in our lives.

MUHAMMAD HUNAIN TARIQ AND SUMAYA ZAMAN
Bahria University Islamabad, Pakistan

December 2025

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

AI	Artificial Intelligence
API	Application Programming Interface
CSS	Cascading Style Sheets
HTML	HyperTextMarkupLanguage
JS	JavaScript
ML	Machine Learning
React	JavaScript library for building user interfaces
OTP	One-Time Password
UI	User Interface
UX	User Experience
DB	Database
FYP	Final Year Project
MERN	MongoDB, Express.js, React.js, Node.js
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
JWT	JSON Web Token
REST	Representational State Transfer
SQL	Structured Query Language
UI	User Interface
UX	User Experience
WC	Worker Category
WP	Worker Profile

Chapter 1

Introduction

Skilled labor is integral to thriving communities, but the recruitment of workers like electricians, plumbers, carpenters, painters, and technicians is still unorganized in developing nations such as Pakistan. Employers regularly go to informal markets or linger by the roads to recruit workerss, who rely on unpredictable daily opportunities. This conventional process is deemed as inefficient and time consumed ; in addition, it is non-transparent, which brings about unfair pricing, miscommunication and obstacles of obtaining credit from dependable job providers. As the world is moving towards digitalization, the modernization of this workflow is also necessary and should be realized through a secure, centralized and comparable system that effectively links both parties.

In order to tackle these issues, this project proposes Jobs Bridge, a digital recruitment platform that organizes skilled workers and employers in a harmonized manner. Employers can now submit their job specifications and browse the posts of authenticated workers, and workers can showcase their skills, information, and service to facilitate easier and more effective hiring decisions. The platform also includes a Flask-based machine-learning price prediction model for fair service cost prediction and OTP-based email authentication. Furthermore, through an API, Jobs Bridge has been enabled to be accessible in Urdu, allowing those with minimal knowledge of English to also make use of the system. An admin panel to monitor and manage platform activities to have a controlled and secure environment. By combining AI-driven insights, technology, Jobs Bridge aims to create a new worker recruitment process online that represents the core concepts of efficiency, transparency, and inclusivity.

1.1 Project Overview

Jobs Bridge is an online application that is aimed at connecting skills and jobs directly to workers and employers, simplifying the recruitment process. It provides a solution to the

long-time existing problem of workers waiting by the road and waiting to get employed, since they can now create detailed profiles (skills, rates, send requests, reviews), and the employers can read and respond to them via a basic request option. Key features include a machine learning-powered price predictor to calculate approximate project costs, a safe authentication system, and support of the Urdu language to simplify platform usage by local users and the administration panel. The platform, which is developed in React.js, Node.js, Express.js, and MongoDB, focuses on scalability and data security. It also offers a simple guide and language assistance to lessen the digital literacy gap, such that all people can use it without difficulty.

1.2 Problem Description

> *"Every day I sit on this footpath, hoping someone needs a plumber. Some days no one comes."*
> — Local laborer in Islamabad

The traditional labor hiring process has failed to address inefficiencies that cause unemployment, unstable income, and mismatches between skilled workers and employers. These challenges can be attributed to the following factors:

- **Inefficient Job Allocation:** Skilled workers (carpenters, masons, electricians, etc.) spend hours or even days waiting on roadsides or in informal markets for work opportunities, leading to waste of production, income instability, and underuse of their skills.
- **Employers' Sufferings:** Employers struggle to find verified skilled workers, often relying on word-of-mouth or unregulated agents, which leads to project delays and inflated costs.
- **Exclusion of marginalized workers:** Existing solutions cater primarily to English-speaking users, excluding Urdu-speaking laborers who struggle with language barriers.
- **Lack of Transparency:** Traditional hiring lacks adequate review and rating systems, creating trust issues between workers and employers. Employers cannot verify workers' past performance, and workers also struggle to prove their credibility.

1.3 Objective

To create a digital hiring platform that revolutionizes the hiring experience for workers and employers alike. With key tasks automated, we want to simplify the process of browsing profiles and posting jobs. Fair pricing estimations, all focused on bringing the hiring experience up to date and simplify it for every person on the hiring team.

- To build an efficient and user friendly digital platform for connecting the skilled worker and the employer.
- For posting jobs and browsing worker profiles, in a safe and open manner.
- Ensure fair price estimation using the ML-based cost prediction model.
- To support multilingual access (including Urdu) to accommodate greater community usage.
- Admin panel (Administration) to hold the platform's integrity and moderators.

1.4 Project Scope

The Jobs Bridge platform is designed as a comprehensive digital marketplace that connects skilled workers such as electricians, plumbers, carpenters, masons, painters, and technicians with employers seeking reliable and qualified service providers. Workers are enabled to create detailed profiles that describe their skills, experience etc and employers can list job requirements, explore worker posts, and directly recruit worker that meet their needs. For better communication, the service uses WhatsApp based messaging that allows workers and employers discussing job details, work shifts and other information to communicate in real time. A dedicated history tab let both the workers and employers to keep track on the past jobs, applications, finished works and previous contacts the tab brings more transparency and helps users keeps track of them as they grow. One of the system's biggest highlight points is its machine learning based price prediction tool that allows for precise and fair cost estimation for different tasks. It helps financial transparency, eliminates overpricing or underpricing, and gets trust from employers and service providers.

For the safety and reliability of the platform, Jobs Bridge provides safe authentication, including OTP verification. The review and rating system promotes further accountability, allowing employers to comment on jobs that were completed and assisting workers to enhance their credibility. To keep the platform user friendly and make it easy for all the users, especially those who are not used to using digital services, warm step training video will be provided for access through the system. In the video directions guide on

how worker can sign up, edit their profiles, apply to jobs and interact with employers, a seamless onboarding experience.

The platform administration have full access to the control panel to check worker profiles user activity, and more to make sure the whole system operates securely and transparently. With full English and Urdu support, Jobs Bridge is accessible to a broader local audience and provides wonderful readability for all levels of literacy and technical levels.

Overall, Jobs Bridge provides a safe, easy to use, and tech-forward platform that streamlines hiring processes, advocates skilled workforce, facilitates employer worker communication, and delivers digital convenience to traditional service markets.

Chapter 2

Literature Review

2.1 Analysis

The digital revolution has transformed talent acquisition and job searches across industries. Traditional methods, such as newspaper advertisements and direct inquiries, are increasingly ineffective in meeting modern demands [1]. Job seekers now expect algorithmic filtering and real-time access to vacancies, while employers require streamlined systems for posting, reviewing, and managing candidates. Jobs Bridge addresses these needs by providing a centralized, intelligent, and user-friendly platform.

Existing recruitment solutions show common shortcomings are lack of structured job postings, poor communication channels between employers and candidates, and absence of intelligent price prediction [2]. Jobs Bridge overcomes these gaps by implementing a structured posting workflow and a machine learning based price prediction module.

Advances in web development enable high-speed, scalable, and flexible recruitment systems. Jobs Bridge leverages the Flask framework for its predictive modules and integrates APIs for multilingual support, including an Urdu interface, enhancing accessibility and adoption [3].

The platform also incorporates automated decision support tools to help employers make efficient decisions, a core aspect of Jobs Bridge [4]. Its web based interface emphasizes clean design, intuitive navigation, and seamless functionality. The modular architecture allows integration of new AI-driven features over time. By prioritizing verified content and user-centric design, Jobs Bridge provides a reliable and adaptable solution within the contemporary recruitment landscape.

2.2 Existing Systems

The development of the digital labor marketplace has reached a new level due to the introduction of the most advanced web technologies, artificial intelligence, and automated verification tools. The existing platforms vary in their coverage, capabilities, intended audience, and geographical utility. This section has offered a thorough insight on the advantages and limitations of the available job-matching solutions through the analysis of the globally and locally used systems. The analysis will be the foundation of creating Jobs Bridge, a platform that will attract both seasoned professionals and daily-wage employees in Pakistan.

2.2.1 LinkedIn Jobs

LinkedIn is a global and one of the biggest professional recruiting networks that provide verified profile of a company, position advertisements, job application tracking, and automated candidate suggestions through AI. Its ranking system examines skills, recommendations, experience and activity to find applicant a potential employer. Despite the good verification and minimization of counterfeit posts, LinkedIn is aimed mostly at highly skilled workers. The site does not support low-income workers in day labor and there are no localized applications that cater to low literacy users.[5]

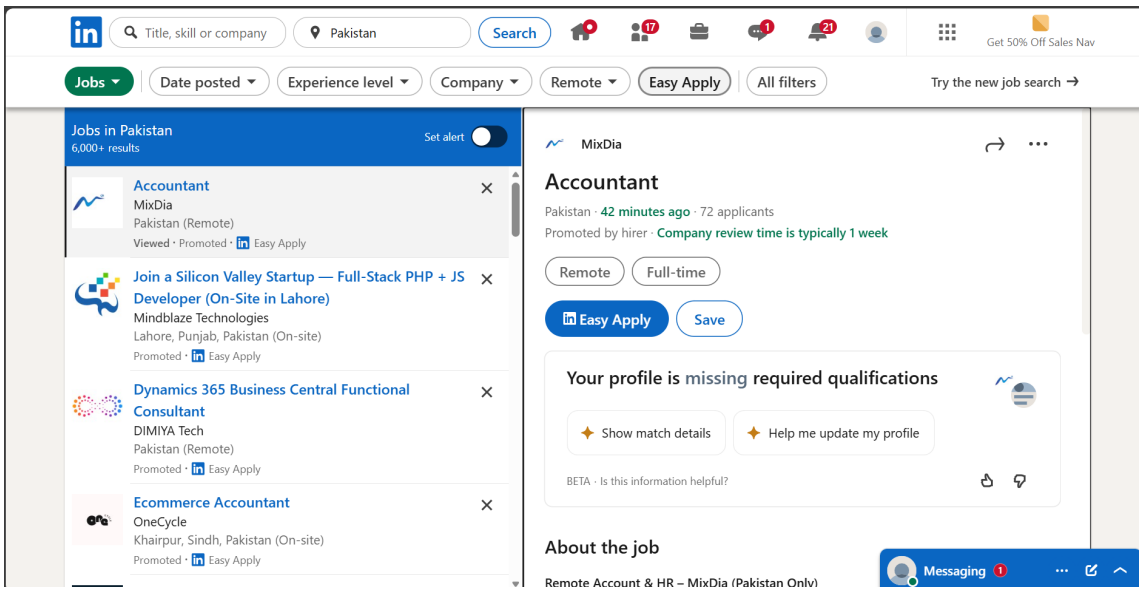

Figure 2.1: LinkedIn Job Details page showing job description, company information, skill match and recommended jobs.

2.2.2 Indeed

Indeed combines job postings in several places, and offers features like resume upload, advanced filtering and employer dashboards. It also has job trend analytics that have natural-language search. Although Indeed checks with certain employers, the site continues to experience the problems of duplicating or even fake advertisements, especially in developing countries. The interface is also not made to support localized languages thus restricting access to users in Pakistan.[6]

2.2.3 Rozee.pk

Rozee.pk is the biggest online recruitment platform in Pakistan that provides job postings, candidate search, and dashboard analytics to the employer. Verifying employers will minimize fraudulent job ads and will help increase user confidence. Nevertheless, in spite of its popularity, Rozee.pk lacks the support of AI-based pay forecasting,and convenient daily-wage employee hiring interfaces. The system is more oriented towards punctual professional jobs as opposed to the temporary work. [7]

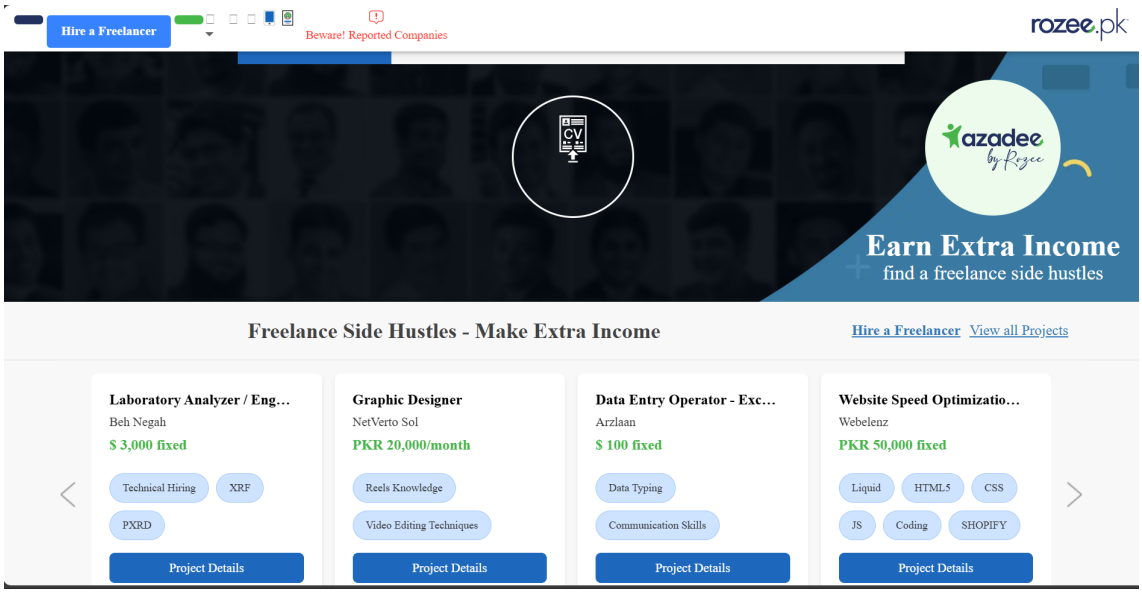

Figure 2.2: Rozee.pk Job listing page showing local job categories, filters, and job results .

2.2.4 Mustakbil.com

Mustakbil.com offers employer branding, resume search and verified job postings. The site is primarily oriented towards full-time professional employment and lacks any support of CNIC-based verification and predictive capabilities based on AI. Also, multilingual interfaces are not available, which discourages the use by larger groups of users. [8]

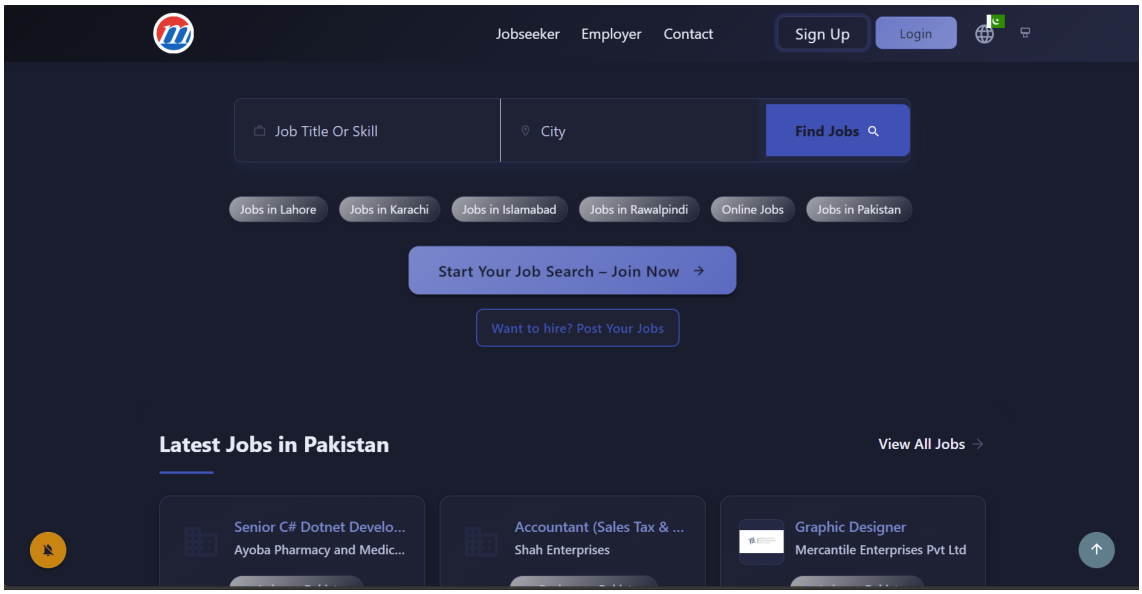

Figure 2.3: Mustakbil.com Job listings interface showing job categories, filters, and postings (Source: Mustakbil.com, 2022).

2.2.5 Local Labour Apps: Kamata and Mazdoor

Kamata and Mazdoor App are other local applications in Pakistan that are dedicated to match laborers with employers on small scale or on daily wage jobs. Although they are applicable to the local market, such apps have fewer AI-based capabilities, and are not equipped with sophisticated employer verification (such as character certificate), as well as tend to cover smaller sets of job categories. They are not optimized to the literacy challenged or of multilingual access by their interfaces.[**?**, 9]

2.2.6 Afolabi et al. (2018) Academic System

Afolabi et al. came up with construction-labor sourcing platform to assist the contractors in achieving digital worker sourcing. Although the system worked well in a particular industry, it could not support multilingual and price prediction. This study notes that additional scalable and AI-toppled labor-matching systems can be used across industries. [10]

2.3 Limitations of Existing Platforms

Despite the useful functions of the above platforms in terms of digital recruitment, there are still a number of major limitations:

- **No Wage Prediction Models:** No AI-based salary forecasting for Pakistani labor markets.
- **Fraud and Fake Job Posts:** Repetitive or unverified workers remain common.
- **High Costs:** Advanced features on global platforms require paid subscriptions.
- **Limited Daily-Wage Support:** Corporate jobs dominate while unskilled labor categories are ignored.

Jobs Bridge is expected to solve these issues by providing a single, convenient, highly localized, affordable platform with built-in admin verification, AI-driven wage prediction, and the ability to bring skilled workers types to the table.

Table 2.1: Comparative Analysis of Job Posting and Labor Matching Platforms

Aspect	**LinkedIn**	**Rozee.pk**	**Mustakbil**	**Kamata**	**Jobs Bridge**
Target Users	Corporate professionals	Pakistani workforce	Professionals	Daily-wage labor	All labor/job categories
Verification Level	Strong identity verification	Employer validation	Basic verification	Limited verification	Character certificate verification
Search & Matching	AI recommendations	Filters	Resume search	Basic filtering	Price prediction
Localization	English only	Partial Urdu	English	Basic Urdu	Full Urdu support
Cost	High	Medium	Low	Free	Free / Minimal
Daily-Wage Support	Very low	Low	Low	Medium	High

Chapter 3

Requirement Specifications

In this section, the Jobs Bridge platform technical specification and its functionality are described in detail. The document is intended to provide a general overview of the system, core features, assumptions, and the technology needed to develop. Also this chapter presents the use cases of the system and provides an overview of the operation of Jobs Bridge from the views of a worker, employer, and admin, which helps the reader to fully grasp what the platform is designed to perform

3.1 Existing Specification

Hiring daily wage and skilled worker in Pakistan still relies on manual, informal procedures. The current system comprises, among others, the following pivotal stages:

- Laborers congregate in person at roadside locations or informal labor markets where they wait to be hired.
- The hiring process also relies significantly on contractors or individuals face-to-face visits to these places.
- There is no proper screening, documentation as well as a filtering based on skills of the workers.
- It is hard for employers to find dependable, skilled workers quickly.
- Workers are subject to insecurity, income fluctuations, and opaqueness in work allocation.
- There is no real time tool to monitor job openings, worker profiles, or demand.

3.2 Limitations of the Current System

The traditional labor hiring system suffers from the following fatal flaws:

- **High Human Dependency:** The process is entirely human dependent and physically conducted at the road side labor sites, making the job distribution erratic and unstable.
- **No Verification:** Workers are not formally vetted, so the employers have enjoyed the trust of workers.
- **Hard to Source Talented Worker:** Whenthe bosses want, they also want to quickly find dependable and competent workers.
- **No Official channels for Communication:** There is no established, secure Information flow between employers and employees.
- **Unavailability of Logs:** The system does not keep digital records of the workers, their job history or employer reviews, so logs of any kind cannot be maintained or analyzed.
- **Minimal Online Support:** Current job sites (Rozee.pk for example) are mostly geared towards advertising white collar jobs and do not cater to the needs of blue-collar workers.
- **No Multi language Support:** Present systems cannot be accessed in local languages such as Urdu, leaving a large part of the workforce out.

The issue is that there are no processesissions enabled by digital technology that a verified worker and a native language facilitator for local outreach can help resolve, which leads to inefficiencies, risks to worker safety, and other challenges in ensuring fair and transparent engagement.

3.3 Proposed System

Jobs Bridge is an online employment platform where employers can find verified and experienced workers in services such as electricians, plumbers, carpenters and more. Make their recruitment process post for workers, view their posts and chat on the integrated WhatsApp app, all in one place. The system aims to aid employers locate dependable workers and gives workers increased visibility and easier access to job leads.

3.3.1 Major Aspects of the Proposed System:

Following are the major features of the Jobs Bridge Platform web application.

- The UI is localized and user friendly Urdu compatible with voice based tutorial.
- Wage prediction is based on machine learning trained with data, service types, and regional influences.
- Admin verifies worker profiles to protect their legitimacy and vouches for workers' competence.
- It has a properly structured feedback and rating system which makes it accountable and trustworthy.
- Users have a chance to report workers for misconduct and maintain safety and reliability of application and services provided.
- A scalable, secure backend architecture to support growth and future integrations.

3.4 Requirement Specification

The primary users of Jobs Bridge are the jobseekers, the employers, and the system administrator. It was decided that users needed a secure system that was easy for posting jobs, easy for finding jobs, and made the worker and employer relationship more transparent. Authorized users can log in securely while admins manage the users and the access. Employers can advertise temporary jobs, with information such as skills required, location, and approximate compensation, and workers can search for jobs and apply according to their verified skill profile. To open access to a wider workforce, the platform should need to be localized and multilingual (Urdu, among others). Admins can track platform usage and control user access via a secure dashboard. It should also enable employers and workers to communicate directly and have access to past job records. This means that Jobs Bridge offers a simple, yet comprehensive, framework for connecting skilled workers to prospective employers in need of their talent.

3.4.1 Functional Requirements

The following are the functional requirements that can be seen as expressing the user needs and system expectations.

- **FR 01: User Registration**
  Both workers and employers registering an account must enter certain required

information. This guarantees that the user is properly authenticated prior to accessing the platform.

- **FR 02: User Authentication**
  Users who have registered can securely log in with their verified email credentials. Only authorized individuals can access their respective dashboards.
- **FR 03: Profile Management**
  Workers are able to create or update skill sets, certifications, and contact details. Employers can edit the profile details for a job.
- **FR 04: Job Posting**
  Employers can post jobs with descriptions, requirements, and locations.
- **FR 05: Prediction model for Price** A machine learning based model to predict the price of jobs based on related jobs data. It allows employers to gain budget estimates.
- **FR 06: Built-in Communication** WhatsApp API provides messaging services between workers and employers. It encourages work discussion, scheduling, and collaborating.
- **FR 07: Ratings** Employers are allowed to rate their workers after the completion of the jobs, and then the workers are given a score based on their performance.

FR 08: Incident Reporting Users can report misconduct, disputes, or fraudulent activity. Incidents that have beenreported are sent to the admin to be reviewed and acted upon.

FR 09: Admin Control Panel The administrator can perform user management, profile verification, and platform monitoring. Analytics track system performance and usage.

FR 10: API for Urdu Language Support Through a built-in API, the system supports the Urdu language API for better accessibility. Users can interact with features or actions in Urdu to offer more convenience for local users.

3.4.2 Non-Functional Requirements

The non-functional requirements capture the quality attributes and the operational and developmental constraints the system will be subjected to. These are not critical for basic functionality but are really necessary to ensure System maintains certain levels of performance, reliability, and usability.

- **NFR1: Scalability:** The system must be designed to allow for future growth, such as having more functions, service categories, and users.

- **NFR2: Security:** The user sensitive password needs to be encrypted. Comprehensive authentication and authorization systems (e.g., JWT) should protect resources from unauthorized access.
- **NFR3: Usability:** The system should be easy to use, support multiple languages (Urdu for local accessibility), and cater to different levels of digital literacy. Video-guided tutorials should be available to assist first time users.
- **NFR4 :Compatibility):** The website should be compatible with all major browsers (Chrome, Firefox, Safari) and all major platforms(deskptop, ios, android, tablet).

3.5 Use Case Modeling

Use case diagram shows the core features and functions of Jobs Bridge and provides a high level description of how the users are expected to the system. User registration and Login, posting a job, requesting a job, saving history, etc., are the essential system functionalities and cover system and user interactions. [11].

3.5.1 Use Case: Register Account

Table 3.1 describes the creation of an account for new users of the Jobs Bridge platform to identify themselves to the system.

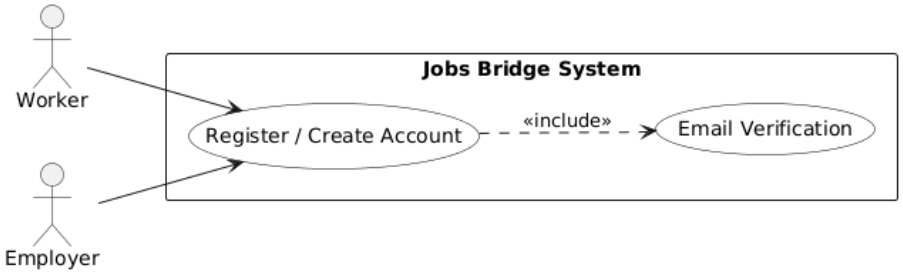

Figure 3.1: Registration

3.5.2 Tabular Description of Registration

Table 3.1: Register Account Use Case Description

Use Case ID	JB-UC-1001
Use Case Name	Register Account
Actors	Non-Registered Employer / Non-Registered workers
Description	A non-registered user (employer or workers) can create an account on the Jobs Bridge platform by providing their personal and contact details.
Preconditions	The user must not have an existing registered account in the system.
Postconditions	User is redirected to the login page after successful registration, and their details are saved in the system database.
Normal Flow	1. The user navigates to the signup page, which shows the registration form. 2. The system checks the input and sends a One-Time Password (OTP) on the user's email for verification. 3. The user fills and submit registration form 4. With a success message, the user is redirected to the login page by the system.
Exception	1. If email is already present, show an error message from System. 2. If any mandatory information is not filled or is invalid, the system will ask the user to re-enter it.

3.5.3 Use Case: Login

Table 3.2describes how registered users log in to the Jobs Bridge platform. This use case guarantees security of system features by user credentials verification.

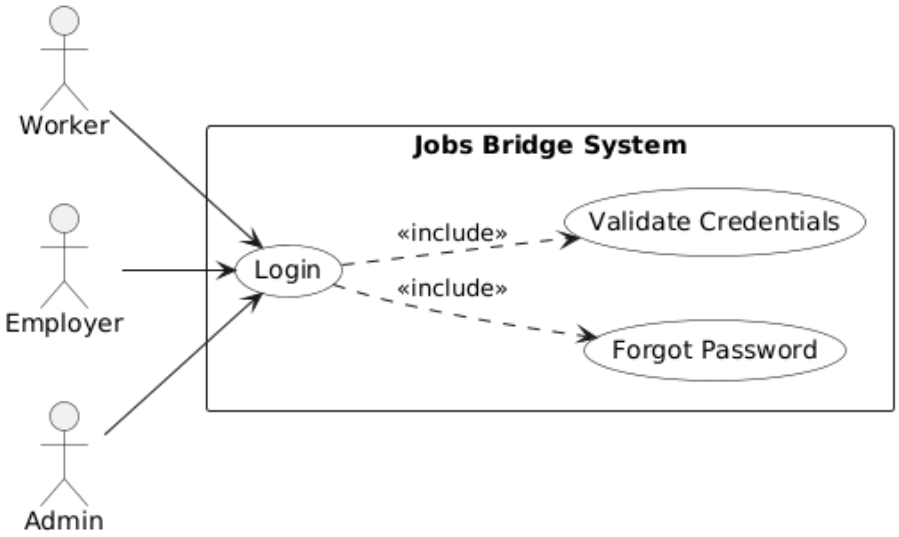

Figure 3.2: login

3.5.4 Tabular Description of Login

Table 3.2: Login Use Case Description

Use Case ID	JB-UC-1002
Use Case Name	Login
Actors	Registered Employer / Registered workers
Description	Authenticates registered users by verifying provided credentials against stored records, granting access to platform features upon successful validation.
Preconditions	• User must have completed registration • Account must be in active status
Postconditions	• User gains access to dashboard
Normal Flow	1. User clicks login button and fills in details.
Exception	Invalid credentials: System displays error message

3.5.5 Use Case: Job Posting

Table 3.3provides the job advertising functionality on the Jobs Bridge platform so that employers can post jobs and be connected with relevant skilled workers.

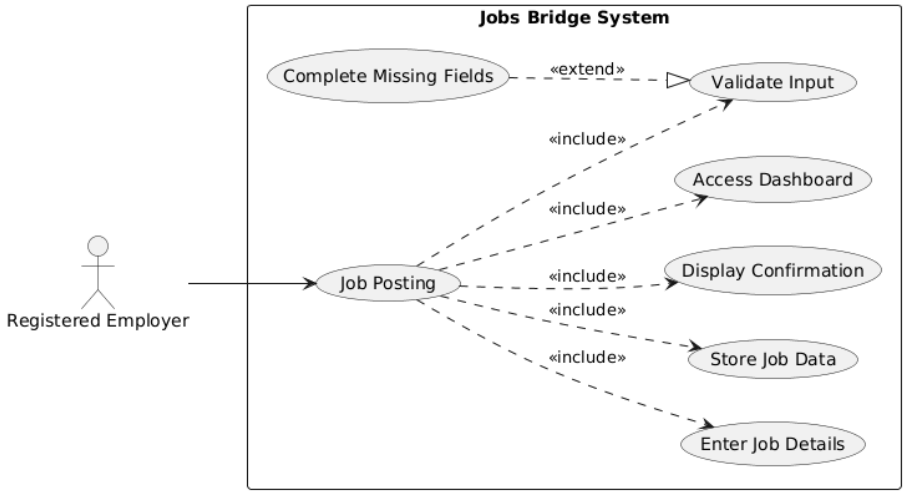

Figure 3.3: Job posting

3.5.6 Tabular Description of Job Posting

Table 3.3: Job Posting Use Case

Use Case ID	JB-UC-1004
Name	Job Posting
Actors	Registered Employers
Description	Authenticated employers can create and post job listings specifying work details, required skills, location.
Preconditions	• Employer must be authenticated • Employer profile must exist in the system database
Postconditions	Job post successfully created and made visible to all registered workers
Normal Flow	1. Employer logs in to the dashboard and click on Post Job button. 2. Employer fills in the job information 3. Employer submits the form via Submit Job button System validates the input 4. System stores the job data in the database 5. System displays confirmation: Job posted successfully
Exception	Missing required fields: prompt user to complete form

3.5.7 Use Case: Worker Profile Creation (with OCR API)

Table 3.4 details the process of creating a worker profile within the Jobs Bridge platform, integrating OCR technology for automated data extraction from uploaded documents.

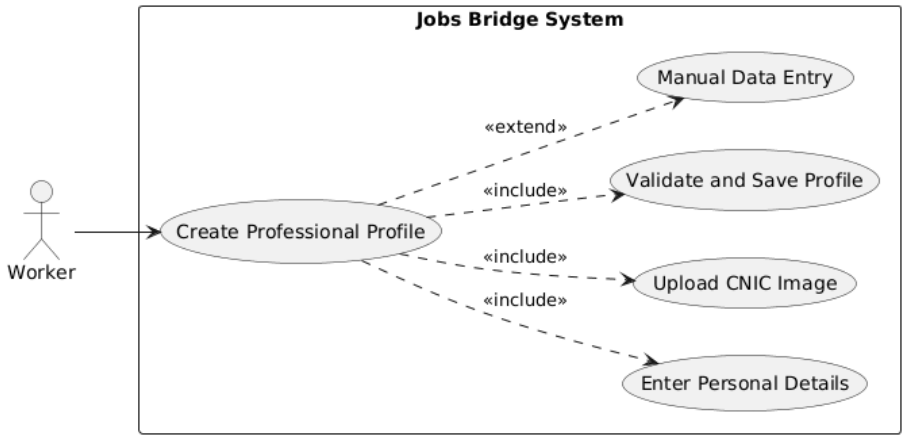

Figure 3.4: Worker Profile Creation (with OCR API)

3.5.8 Tabular Description of Worker Profile Creation (with OCR API)

Table 3.4: Worker Profile Creation (with OCR API) Use Case Description

Use Case ID	JB-UC-1005
Use Case Name	Worker Profile Creation (with OCR API)
Actors	Registered Worker
Description	Allows workers to create their professional profiles by entering personal details. The OCR API assists in automatically extracting data from uploaded CNIC to speed up the process.
Preconditions	• Worker must be logged in. • OCR API must be active and connected.
Postconditions	• Worker profile successfully stored in MongoDB. • Confirmation notification sent to worker.
Normal Flow	1. Worker clicks create profile. 2. The system displays the profile creation form, and the worker enters details and uploads CNIC image. 3. OCR API extracts data from uploaded image. 4. System validates and saves profile information. 5. Confirmation message displayed and profile becomes active.
Exception	OCR extraction fails: System prompts manual data entry.

3.5.9 Use Case: Job Application by Worker

Table 3.5 outlines the process for workers to apply for available jobs within the Jobs Bridge platform, allowing employers to receive and review applications efficiently.

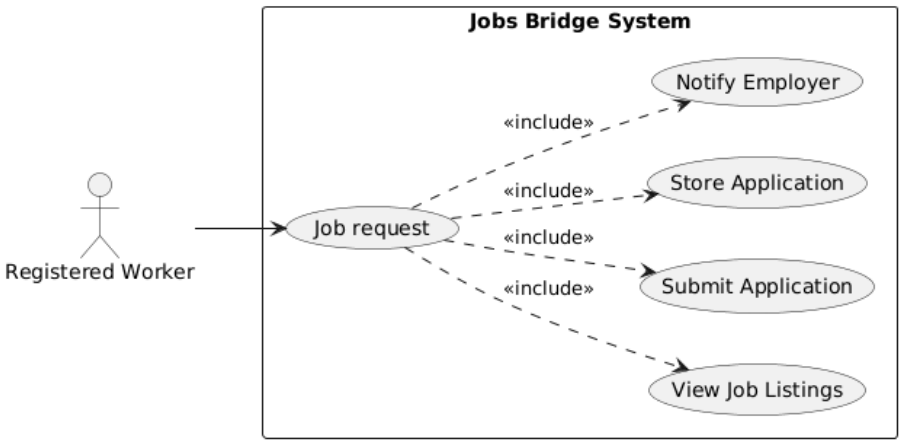

Figure 3.5: Job Application by Worker

3.5.10 Tabular Description Job Application by Worker

Table 3.5: Job Application by Worker Use Case Description

Use Case ID	JB-UC-1006
Use Case Name	Job Application by Worker
Actors	Registered Worker
Description	Workers submit job applications for listed opportunities.
Preconditions	• Worker account verified
Postconditions	• Application stored in system • Employer notified
Normal Flow	1. Worker views job listings 2. Selects desired job and Clicks request 3. System records application and sends alert
Exception	Nill

3.5.11 Use Case: Job Price Prediction (Flask Module)

Table 3.6 outlines the process for workers or employers to estimate job price using a Flask-based prediction module.

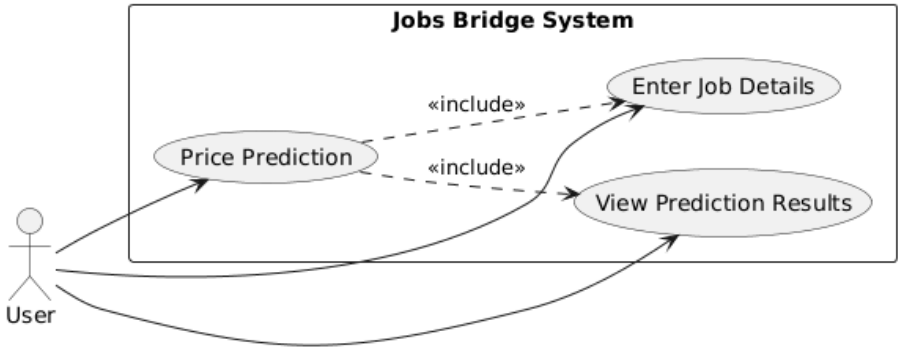

Figure 3.6: Job Price Prediction (Flask Module)

3.5.12 Tabular Description Job Price Prediction (Flask Module)

Table 3.6: Job Price Prediction Use Case Description

Use Case ID	JB-UC-1007
Use Case Name	Job Price Prediction (Flask Module)
Actors	User
Description	This use case allows employers or workers to estimate job prices using a Flask-based prediction module.
Preconditions	Navigate to the price prediction tab.
Postconditions	Predicted price is displayed on the screen.
Normal Flow	1. Employer opens the Price Prediction page and enters details. 2. Flask module processes input. 3. System displays the estimated job price.
Exception	If input data is missing, the system requests complete details.

3.5.13 Use Case: Worker Verification by Admin

Table 3.7 outlines the process for aadmin to verify a worker before activation.

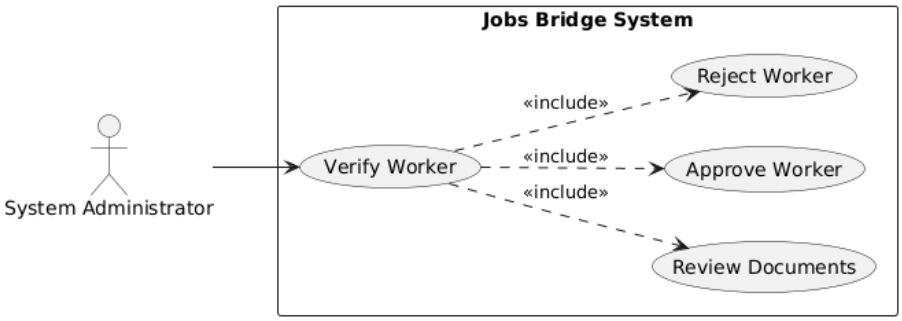

Figure 3.7: Worker Verification by Admin

3.5.14 Tabular Description Worker Verification by Admin

Table 3.7: Worker Verification Use Case Description

Use Case ID	JB-UC-1008
Use Case Name	Worker Verification by Admin
Actors	System Administrator
Description	This use case allows the admin to verify worker profiles and documents before activation.
Preconditions	Admin must be logged in.
Postconditions	Worker is marked as verified or rejected.
Normal Flow	1. Admin navigates to Pending Verifications. 2. Reviews worker documents and details. 3. Clicks Verify or Reject. 4. System updates verification status.
Exception	NILL

3.5.15 Use Case: Review and Rating Submission

Table 3.8 outlines the process for for employers to provide feedback and rate workers after job completion.

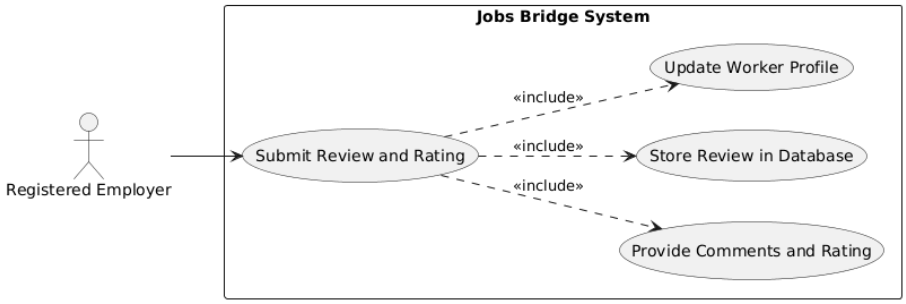

Figure 3.8: Review and Rating Submission

3.5.16 Tabular Description Review and Rating Submission

Table 3.8: Review and Rating Submission Use Case Description

Use Case ID	JB-UC-1009
Use Case Name **Actors**	Review and Rating Submission Registered Employer
Description	This use case allows employers to provide feedback and rate workers after job completion.
Preconditions	Job must be successfully completed.
Postconditions	Review and rating are saved in the database.
Normal Flow	1. Employer opens completed task and Clicks Submit Review. 2. Provides comments and rating. 3. Submits the review form. 4. System stores review and updates worker profile.
Exception	If review fields are empty, system prompts user to complete all required inputs.

Chapter 4

Design

In this chapter, the design of the Jobs Bridge system, a web-based interface that brings skilled workers and employers together in a virtual environment, is presented. The system design specifies the high-level system architecture and all major system components, the data flow, operational requirements, performance requirements, and development requirements. This chapter also describes, in brief, interaction among various modules like user registration, posting a job, worker profiles, searching for a job, service price prediction (Flask module), and the Urdu Language Support API in system architecture.

4.1 System Architecture

The Jobs Bridge facility provides a direct and real time communication channel for workers, employers, and the price prediction through a web based application. The system takes various details related to the job from the user, processes the information, and shows the predicted outputs on a simple web-based interface. The architecture partitions the system into several components, namely, the web application interface, backend processing unit, machine learning based price prediction module, data storage unit, and integration with external APIs for language support. It also results in ease of maintenance, better scalability, and further interfaces with additional enhancements in the future.

4.1.1 Core Components

4.1.1.1 Web Application Interface

- Workers and employers can enjoy the web-based interface.
- Workers and their employers can build out their profiles, they are able to add their skill-sets and check out details on jobs.
- Employers may post jobs, update job details and browse worker posts.

- Facilitates a specialized interface for the Price Prediction Module where users specify job particulars.
- It allows users to change the language to Urdu language with the help of an integrated Urdu API to make it usable for everyone.

4.1.1.2 Backend Processing Unit

- The platform's main backend is Node.js (Express) based, which holds the platform's core system logic, such as user authentication, session handling, worker profile management, job posting, and form submissions.
- The web interface (React/front-end) and the server communicate through RESTful APIs which are implemented in Express.
- There is a separate backend module written in Flask that specifically deals with the Price Prediction Module.
- Flask interacts with the machine-learning model to calculate a work price prediction given a set of wokr specifics that are input manually.

4.1.1.3 Machine Learning Price Prediction Module

- Includes the trained regression models such as Linear regression, Random Forest, and XGBoost Regressor.
- It accepts job inputs (job category, time duration, etc.) and the actual price is predicted.
- Predicts the cost of the job.
- Deployed using Flask so users can instantly view the estimated job cost.

4.1.1.4 Urdu Language Support API

- An external Urdu translation API is integrated into the system.
- The interface is made friendly for Urdu users by translating the English content of the website into urdu.
- This local module adds multilingual capabilities inside of the platform, enhancing UI experience.

4.1.1.5 Data Storage and Database System

- The system is designed with data storage in mind, using MongoDB, a NoSQL database, which is better suited for handling large amounts of data and provides more flexibility.
- MongoDB collections store:
  - Worker profiles and skills information
  - Employer accounts
  - Job posts and job information
- The NoSQL design enables flexible and schema-less dynamic data storage, which can accommodate more kinds of jobs and user inputs.
- Efficient retrieval system performance are guaranteed by indexing and optimized queries.

4.1.1.6 Frameworks, Languages, and Libraries

- **Frontend:** React.js, Tailwind CSS.
- **Backend:** Node.js with Express for core system logic; Python (Flask) only for the Price Prediction Module.
- **Machine Learning:** scikit-learn (Linear Regression, Random Forest, XGBoost) for Price Prediction.
- **Language Support:** Google Translate API service to translate the website text from English to Urdu.
- **Database:** MongoDB (NoSQL) for flexible, scalable data storage.
- **Data Processing:** NumPy and Pandas for preprocessing and performing MLcomputations.

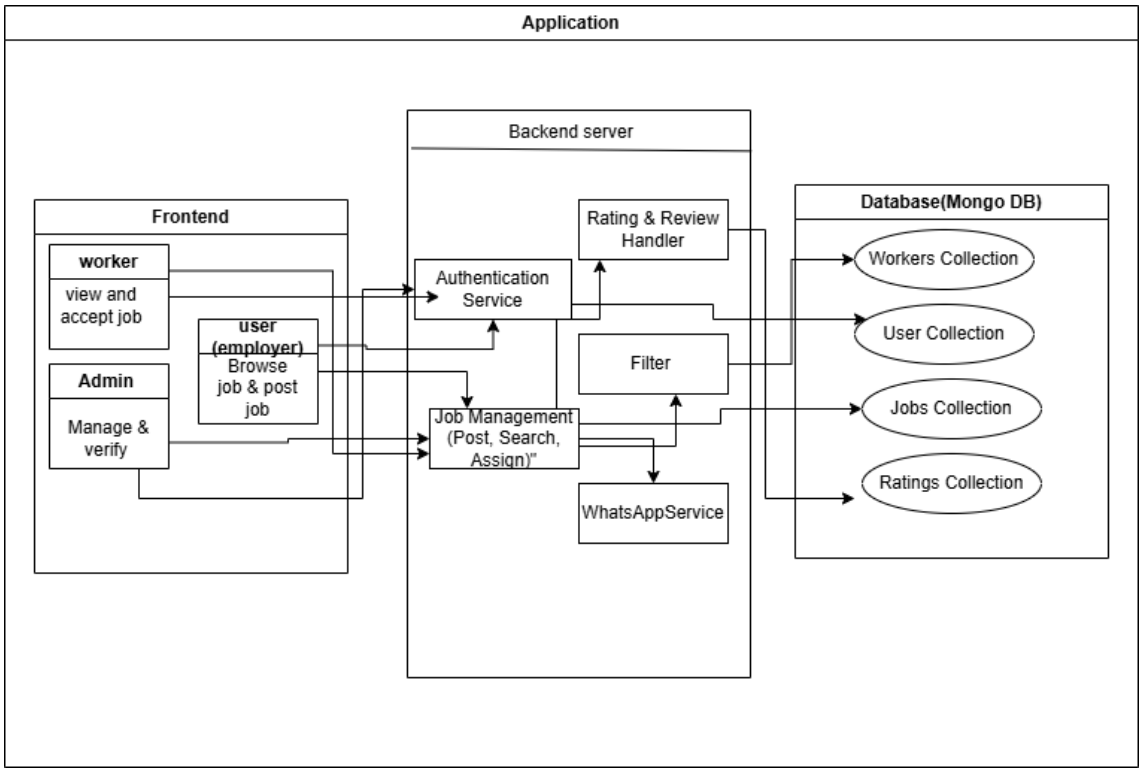

Figure 4.1: System Architecture

4.2 Design Constraints

4.2.1 Performance and Resource Usage

- This system is supposed to work in real-time on popular web browsers without any lag.
- Price Prediction Module computations are performed on the server (Flask backend) so as not to slow down the client interface.
- The React.js frontend should be responsive at all times when processing large quantities of job, or worker profiles.

4.2.2 Technology Stack Limitations

- The main system backend is written in Node.js with Express, and the frontend is in React.js, styled by Tailwind CSS.
- Price Prediction Module is a Python (Flask) module that uses scikit-learn models for performing regression based predictions.
- Urdu translation is available through the Google Translate API, for which you need to have a proper stable network connection.

4.2.3 Design Requirements

- **Code Style:** React code is written in a way that follows the standard React idioms so that it is easy to read and consistent across the code base. Code for the price prediction module is PEP8 compliant.
- **API Design:** REST APIs are well-documented and have consistent request and response payloads that make it easy to predictably communicate data between frontend and backend.
- **Database Schema:** MongoDB schemas are built with scalability and flexibility in mind to accommodate dynamic job and user information.
- **UI/UX Standards:** Tailwind CSS and Material Design guidelines adoption ensures a neat, simple, easy to interact with and interesting interface.

4.3 Design Methodology

While building Jobs Bridge, we will be embracing an Agile approach, which will allow us to develop in a flexible and iterative manner. Main processes include user registration, worker profile creation, administrator verification, job posting, request and assignment, review management, support for the Urdu language, and a cost prediction tool. The earliest development phase encompasses the most rudimentary features, like registration, dashboard access, and logging in. Personal details and skills along with the documents are provided, and then the worker profile is verified by the admin. Then job posting and search, application, and assignment features.The plumbers, electricians, and painters pricing prediction tool recommends how much to charge for each job to workers and clients according to job type, etc. At each stage the developers design and test the phase to ensure that the output is a correct and reliable software product. This is an iterative method that is a flexible, good quality solution that efficiently connects employers to competent workers. Also the design activities for Jobs Bridge employ simple UML style techniques, including use case diagrams, activity diagrams, and a general architecture diagram. These graphic representations are used to lead and enhance the design of the system, even though object-oriented modeling was never fully implemented.

4.4 High-Level Design

This is a multi architectural viewpoint description of the Jobs Bridge The architectural views will allow us to say what the infrastructure of the system looks like from inside and from outside. They describe how the system is logical, processed and coded. These perspectives combined yield a complete high level design model of the system.

4.4.1 Conceptual View (Logical Design)

The conceptual view describes the main functional components of the Jobs Bridge system and how they interact. It expresses the structure of the conceptual without considering any implementation technologies.

Key Components:

- **User Management Module:** The login, registration, authentication, and profile creation for both workers and employers.
- **Worker Profile & Verification Module:** Workers can upload documents, skills, and personal information. The profiles are verified and approved by Admin.
- **Job Posting & Search Module:** Employers post jobs, and workers filter their jobs and apply.
- **Request & Assignment Module:** Submit request to jobs, assign tasks, and update job status.
- **Review & Rating Module:** Employers can give reviews and ratings after the job is done.
- **Urdu Language Support Module:** Facilitates translation help utilizing API with support for dual language usage.
- **Price Prediction Module (Flask API):** Recommends price estimation for plumbing, electrical, and painting services.

4.4.2 Process View

The Process View describes the collaboration of the Jobs Bridge system core components at runtime. This process is concerned with the system's dynamic behavior and with the interactions among the modules to process events triggered by users, such as searching for a job, posting a job, reqesting for a job, and updating a user profile.

4.4.2.1 User Interaction Flow

Job Search Process

- The job seeker applies filters in the Job Interface.
- The request is forwarded to the system, which queries the database for relevant job posts.
- The user interface is again returned with the matching job posts.

Job Posting Process (Employer)

- The employer completes the job details in the Job Posting Interface.
- The system stores the new job post in the database.
- The updated list of job posts is shown to the employer.

Job request Process(worker)

- The job seeker hits “request” on a particular job.
- Upon receiving the application request on the backend, the application record is persisted.
- The system sends confirmation or status notification back to the user interface.

Profile Update Process

- Users modify the profile details
- The system takes the input and update the database.
- The user interface shows new profile information.

4.4.2.2 Data Workflow

Handling job data

- When a job seeker selects the filter, job data is obtained from the Job Database.
- Filters are processed in the frontend.
- Just the matched results will be returned to the report interface.

Applications Data Flow

- Once a job seeker has applied for a job, the application information is saved by the system in the Applications Database.
- Employers can access these applications via their dashboard.

Profile Data Flow

- A user makes changes to their profile; the system updates with the new information and saves the data in the Users Database.
- The user interface now displays the new information.

4.5 Low Level Design

LLD is a detailed description of how one particular module will be designed and implemented in the system. It focuses on the pairwise component interactions on a component flow level to realize a functionality one at a time.

4.5.1 Sequence Diagrams

Jobs Bridge sequence diagrams are used to represent Jobs Bridge with sequence diagrams. These diagrams show how the frontend, backend, and database interact, e.g., for the login and registration features.

4.5.1.1 Signup Sequence Diagram

Signup sequence diagram describes how the user interacts with the Jobs Bridge system while signing up. The user submits details, which are verified by the system. After validation successfully, the user data is stored in the database, and a confirmation response is sent. The user is subsequently taken to the login or dashboard page.

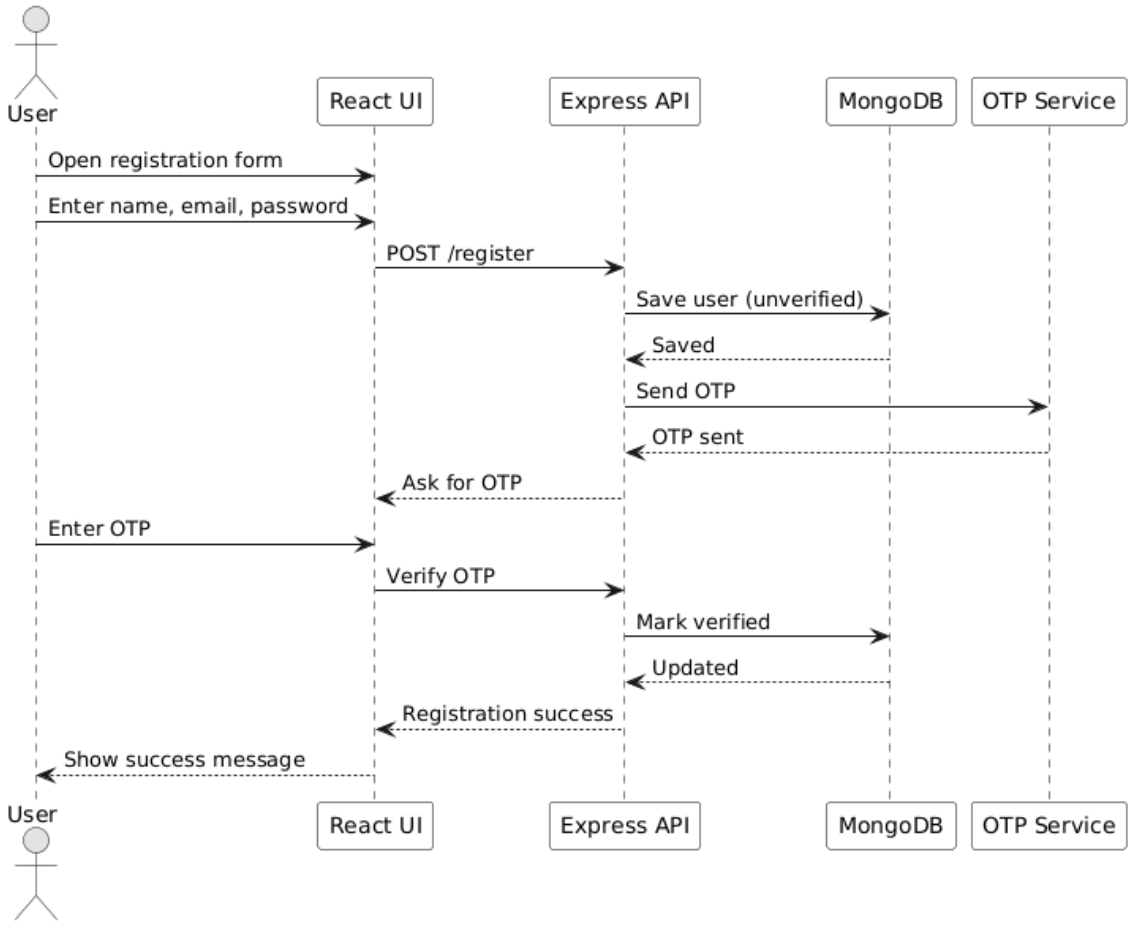

Figure 4.2: Signup Sequence Diagram

4.5.1.2 Login Sequence Diagram

The login sequence diagram is a detailed description of the user authentication flow within the system that starts with the user opening the login page and submitting the credentials. After the user enters the user name and password, the login page communicates with the database. When the user has submitted their username and password, the login page interacts with the database to check the credentials. Based on the verification, the user can be either allowed to log in to the application or can be shown to try again for secure and robust login experience.

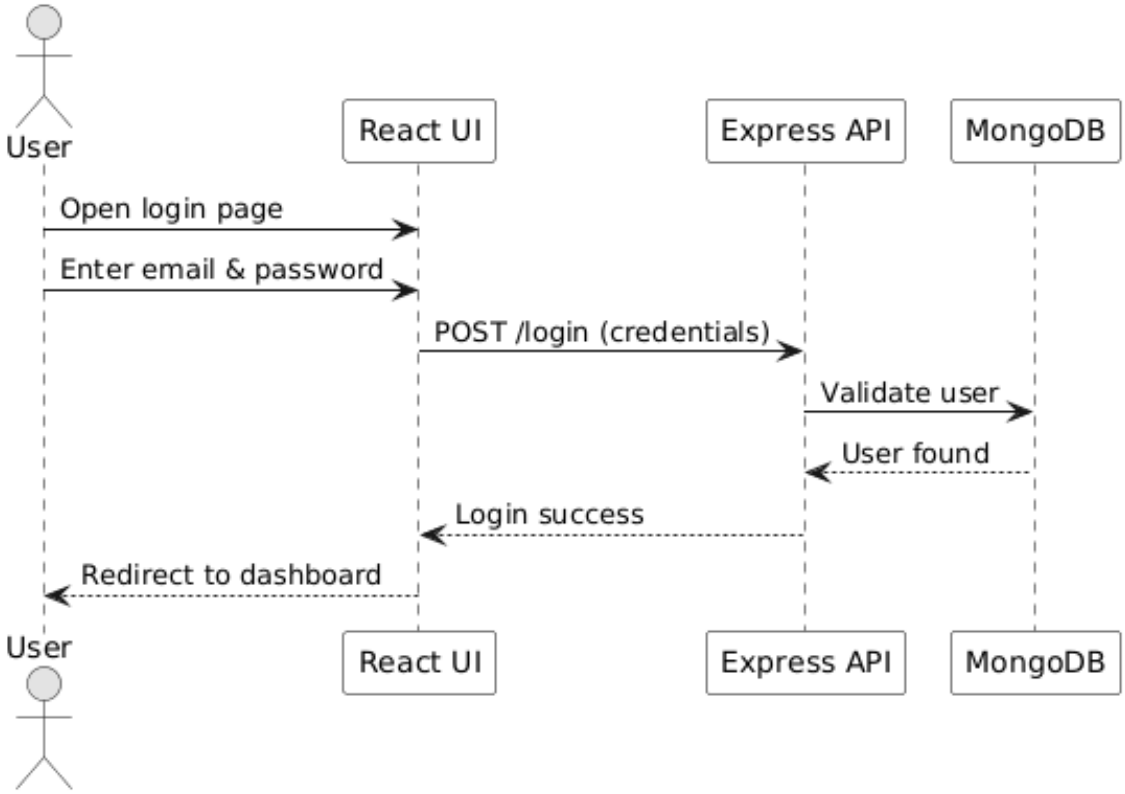

Figure 4.3: Login Sequence Diagram

4.5.1.3 Employer Job Posting Sequence Diagram

The employer job posting sequence diagram is a representation of the flow of the employer and system in posting a new job. It describes how the frontend interacts with the backend to validate the field, store the job details in a database, and successfully post the job.

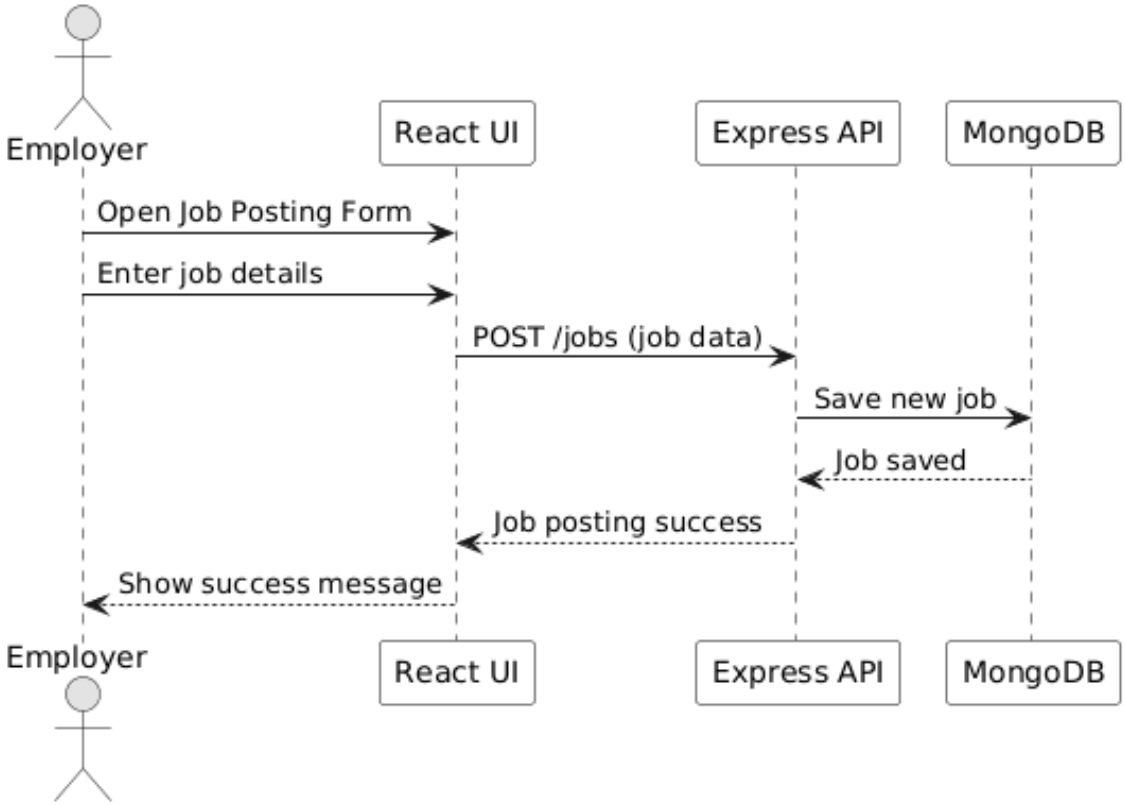

Figure 4.4: Employer Job Posting Sequence Diagram

4.5.1.4 Worker Applying for a Job Sequence Diagram

The worker applying for a job sequence diagram represents the sequence of a job application in the system between the worker and the system. It demonstrates the frontend request and backend response to get the job list, send the worker's application, save it in the database, and employer contact to worker via WhatsApp.

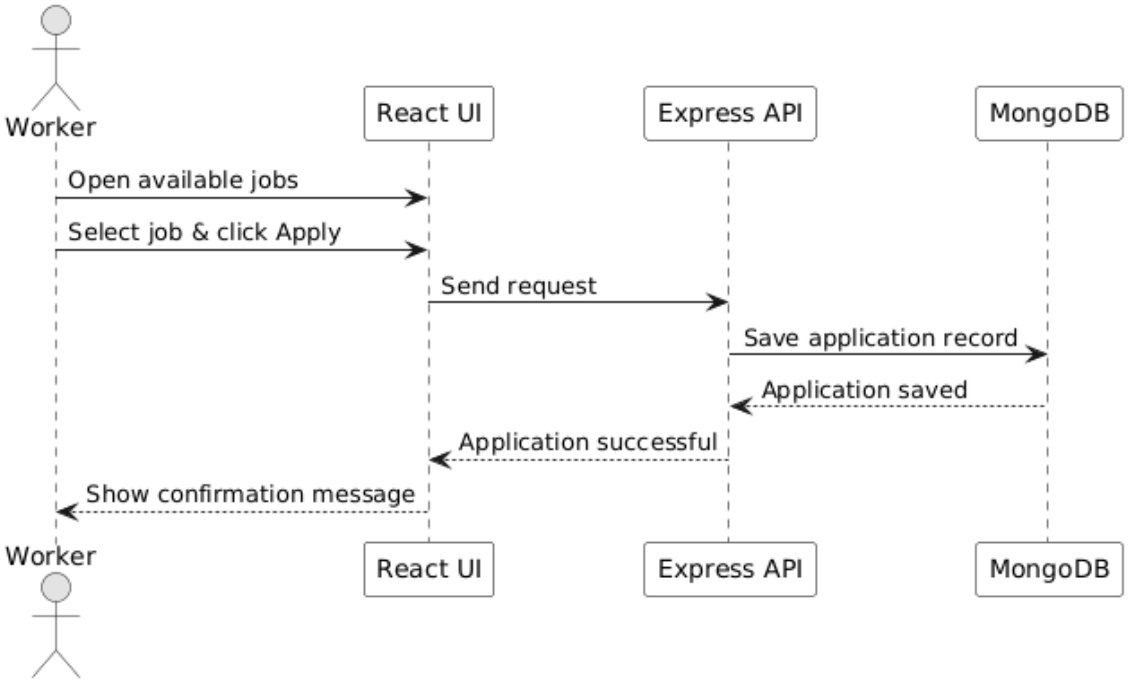

Figure 4.5: Worker Applying for a Job Sequence Diagram

4.5.1.5 Price Prediction Sequence Diagram

This sequence diagram shows how Estimate Price works in Price Prediction. sequence shows the user requesting a service price estimate by pressing the “Estimate Price” button and the system calculating and display the predicted result.

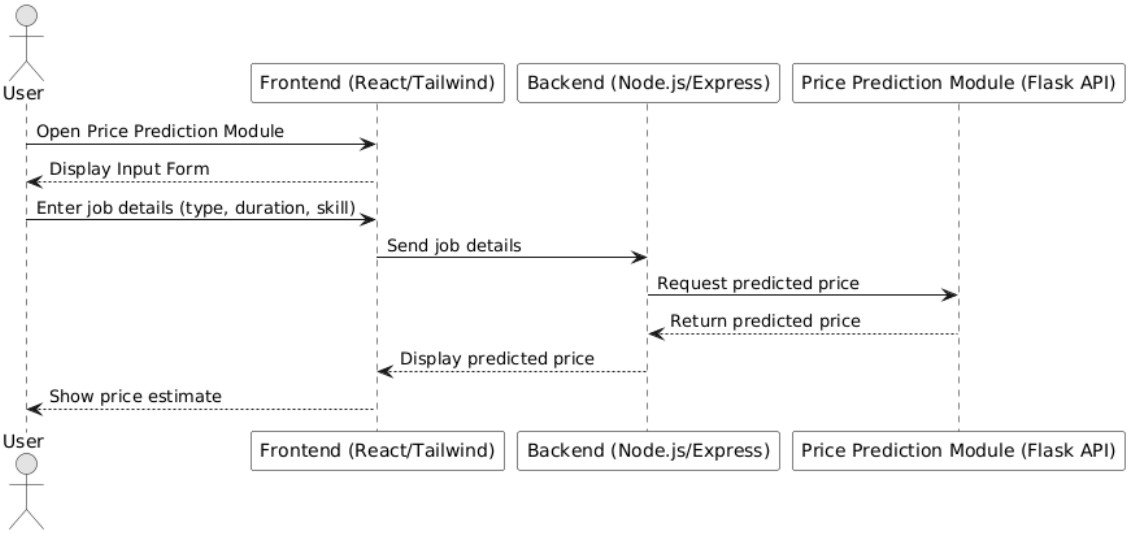

Figure 4.6: Price Prediction Sequence Diagram

4.5.1.6 Admin Verification Sequence Diagram

The admin approval sequence diagram showing the worker verification process is as follows. Applicant profile CNIC info When a worker uploads a profile or cnic information, the system record that information with pending verification status. The admin then logs into the site, access the pending profiles, reads through each one, and either approves or denies the verification application. The result is updated in DB and the worker is informed of the decision accordingly.

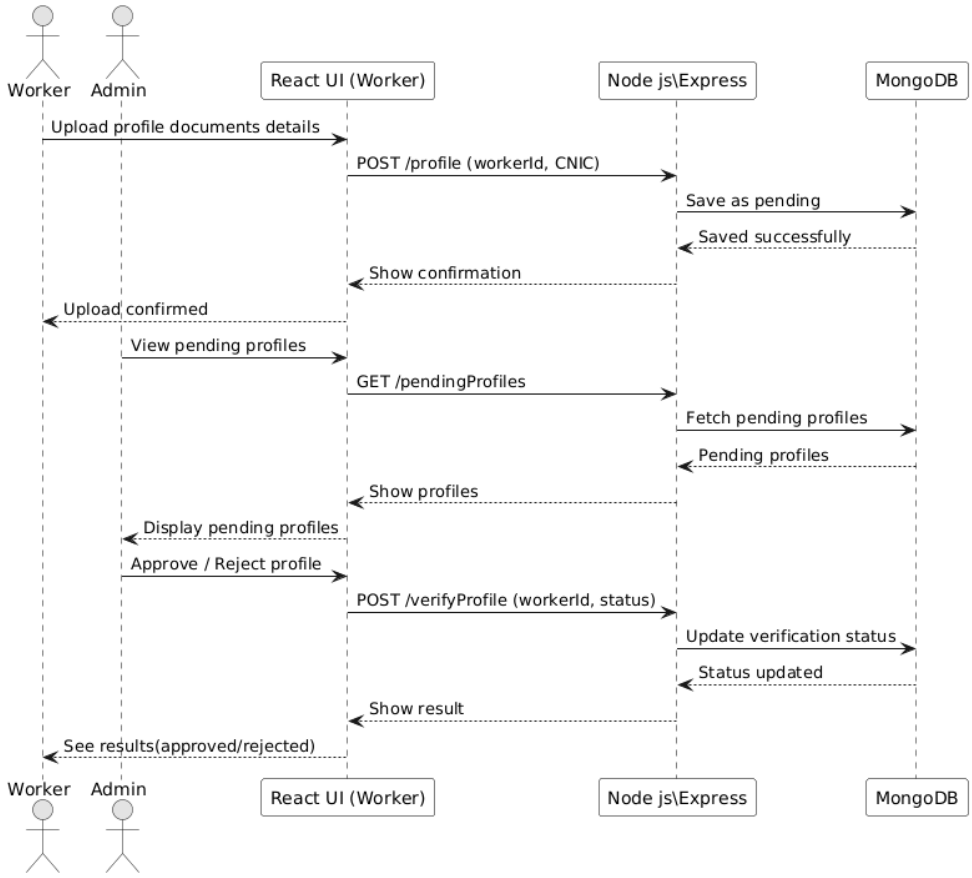

Figure 4.7: Admin Verification Sequence Diagram

4.5.1.7 Review and Rating Sequence Diagram

Review and rate sequence diagrams show how an employer will review the performance of the worker after completion of the job. The worker profile is then obtained by the employer, submits a review and rating, which is saved in the database. Then, the system recalculates the worker's overall rating and updates the worker's profile. This makes services more transparent and of higher quality, and also allows other users to trusted workers.

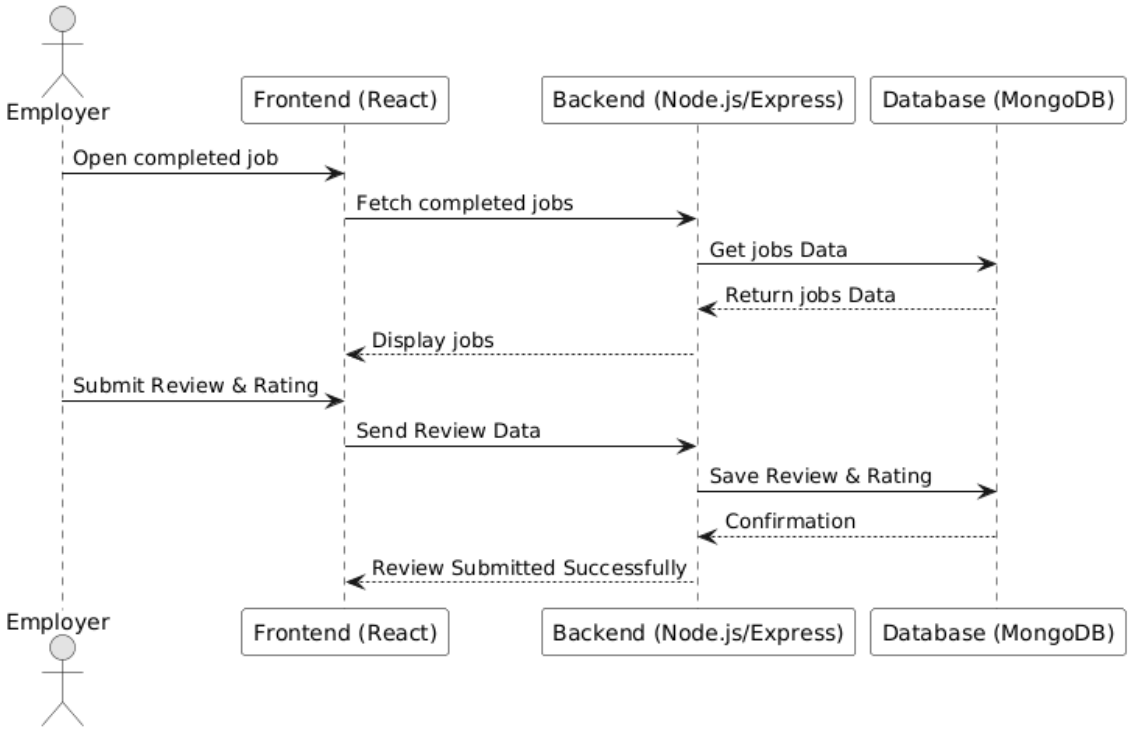

Figure 4.8: Review and Rating Sequence Diagram

4.5.2 Finite State Machine

The Jobs Bridge platform employs a Finite State Machine (FSM) to represent every stage of the workflow of both user interaction and system operation. The system is in a single state at one point in time, for example, searching for jobs or sending job requests. Transitions between these states are triggered by certain user behaviors or system-related events, thereby providing a controlled flow within the application. Figure 4.9 depicts the finite state machine (FSM) for the Jobs Bridge use case.

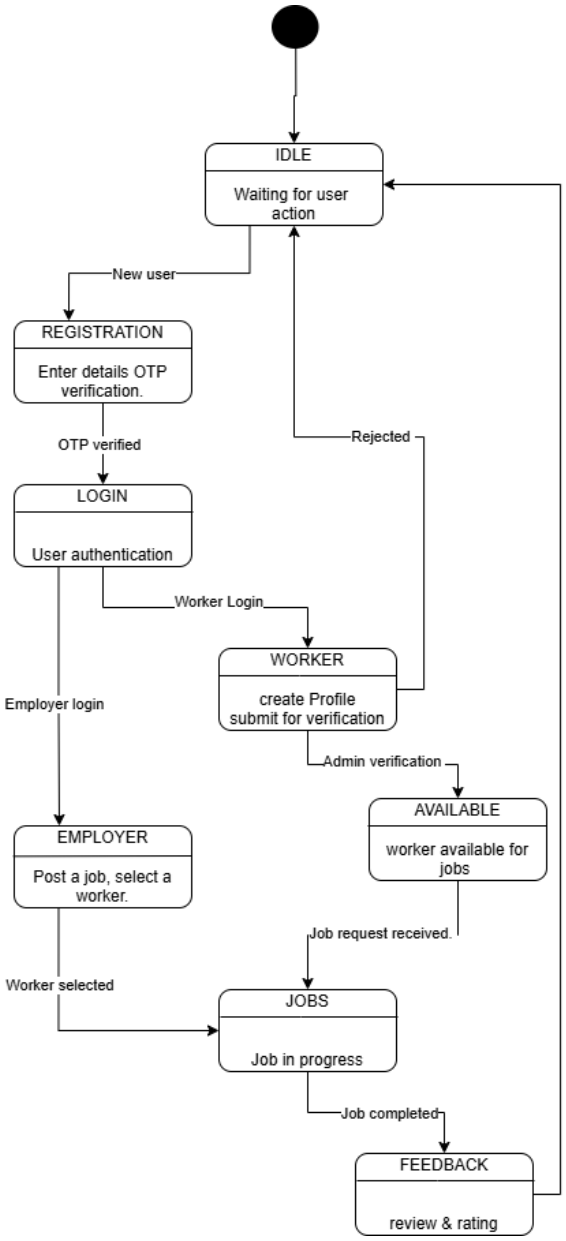

Figure 4.9: Finite State Machine of Jobs Bridge workflow

4.6 Activity Diagrams

Activity diagrams are used to model in a dynamic manner through the flow of activities. These diagrams provide insight into when actions are taken and when decisions are made, as well as the process flow of the system. In the Jobs Bridge system we use activity diagrams to describe all the important processes of the system, such as user registration, log-in, job posting, price prediction, job application, job evaluation, incident reporting, and management of jobs history.

4.6.1 User Registration Activity Diagram

The user registration activity diagram describes the process through which a new user creates an account on the Jobs Bridge system. The user enters the required personal information and submits the registration form. The system verifies the information and performs OTP-based verification. Upon successful verification, the user account is created and stored in the database, allowing the user to access the platform.

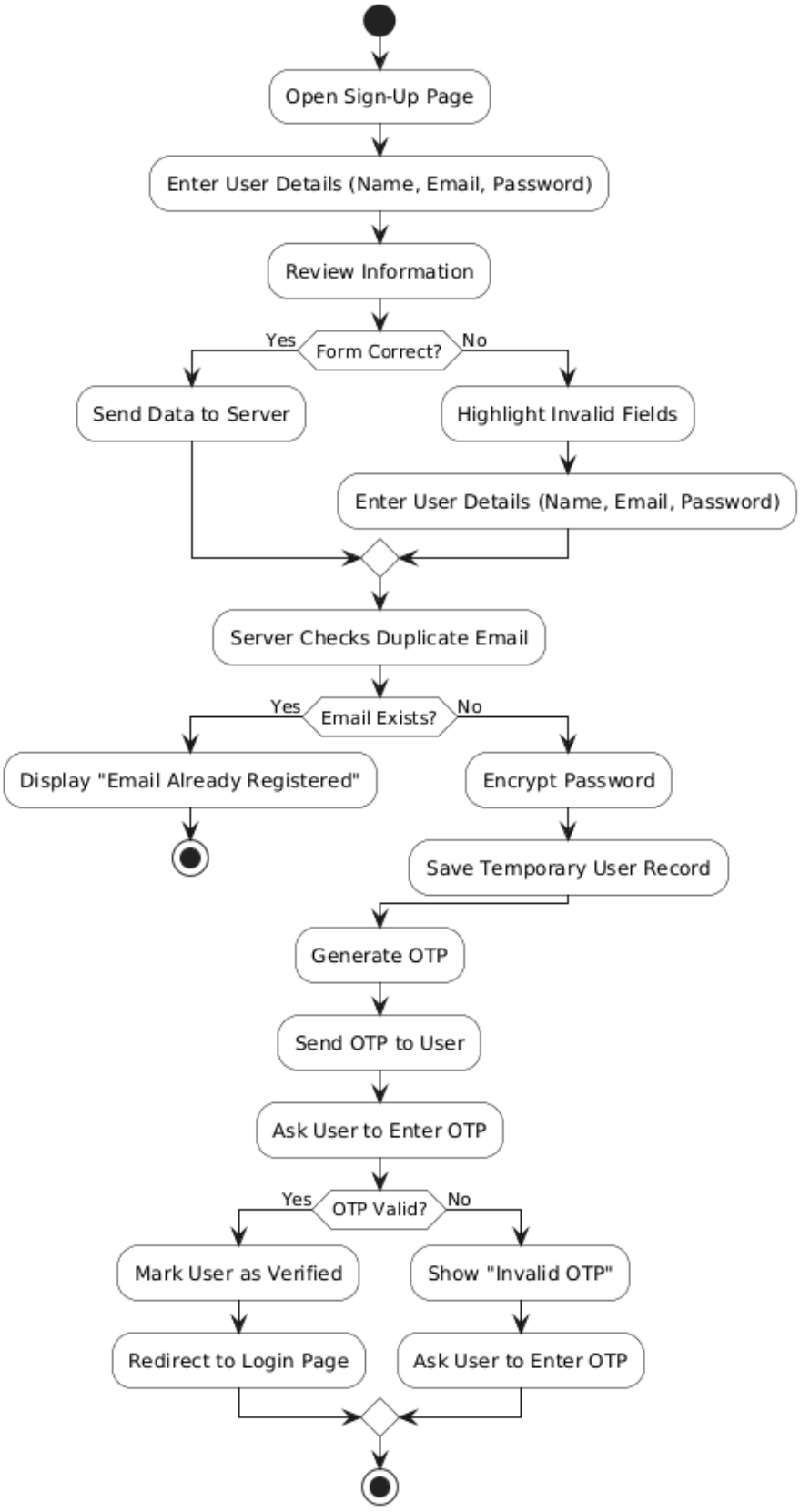

Figure 4.10: User Registration Activity Diagram

4.6.2 User Login Activity Diagram

The user registration activity diagram describes the flow of activity to create an account for a new user in Jobs Bridge system The user makes the necessary entries in the personal details needed and presses on the Register button. The system validates the data, and an OTP-based validation is executed. After completion of verification, the user account is created and saved in the database and then user can access the platform.

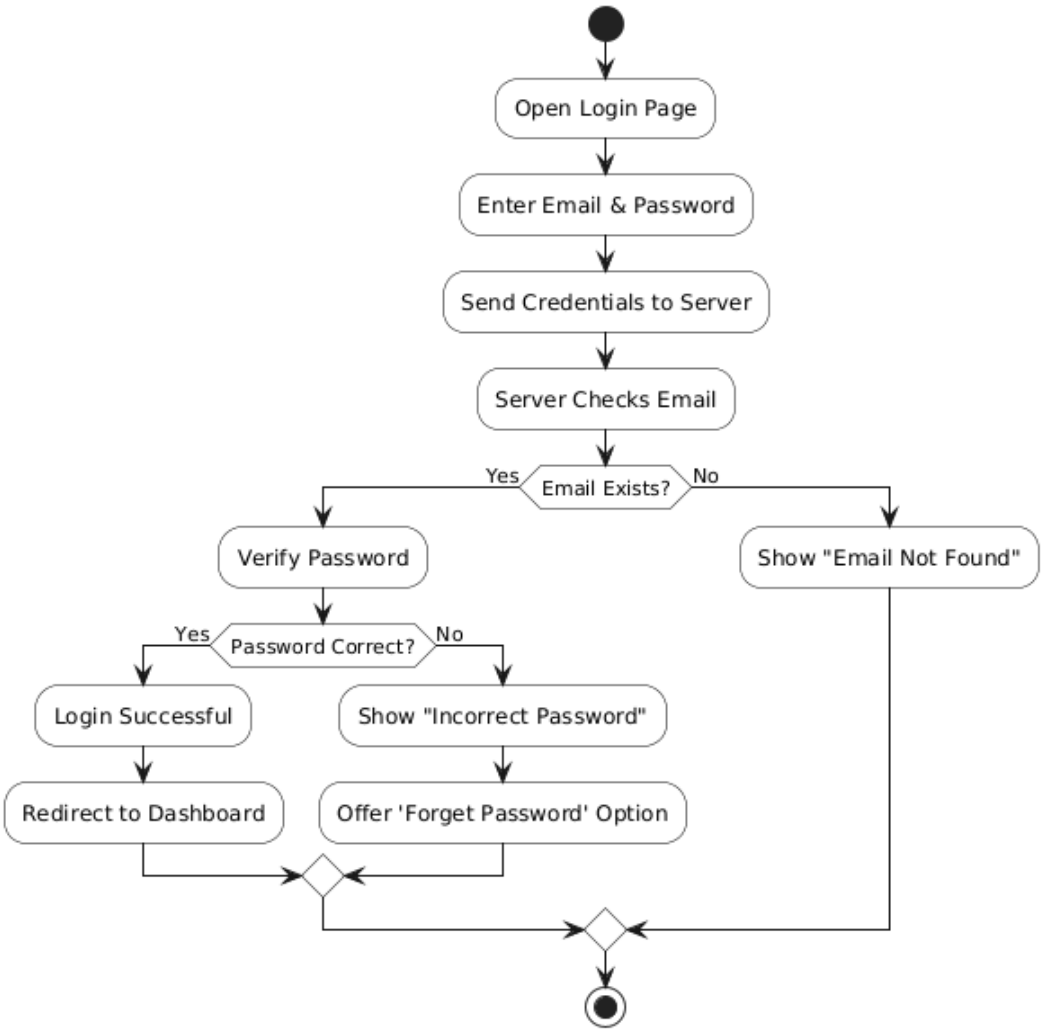

Figure 4.11: User Login Activity Diagram

4.6.3 Job Posting Activity Diagram

The activity diagram shows the posting jobs performed by the employers. The employer provides the information related to job type, description, and time frame. The job is saved by the system into the database, and it is visible to the workers. This allows the employer to post jobs on the site efficiently.

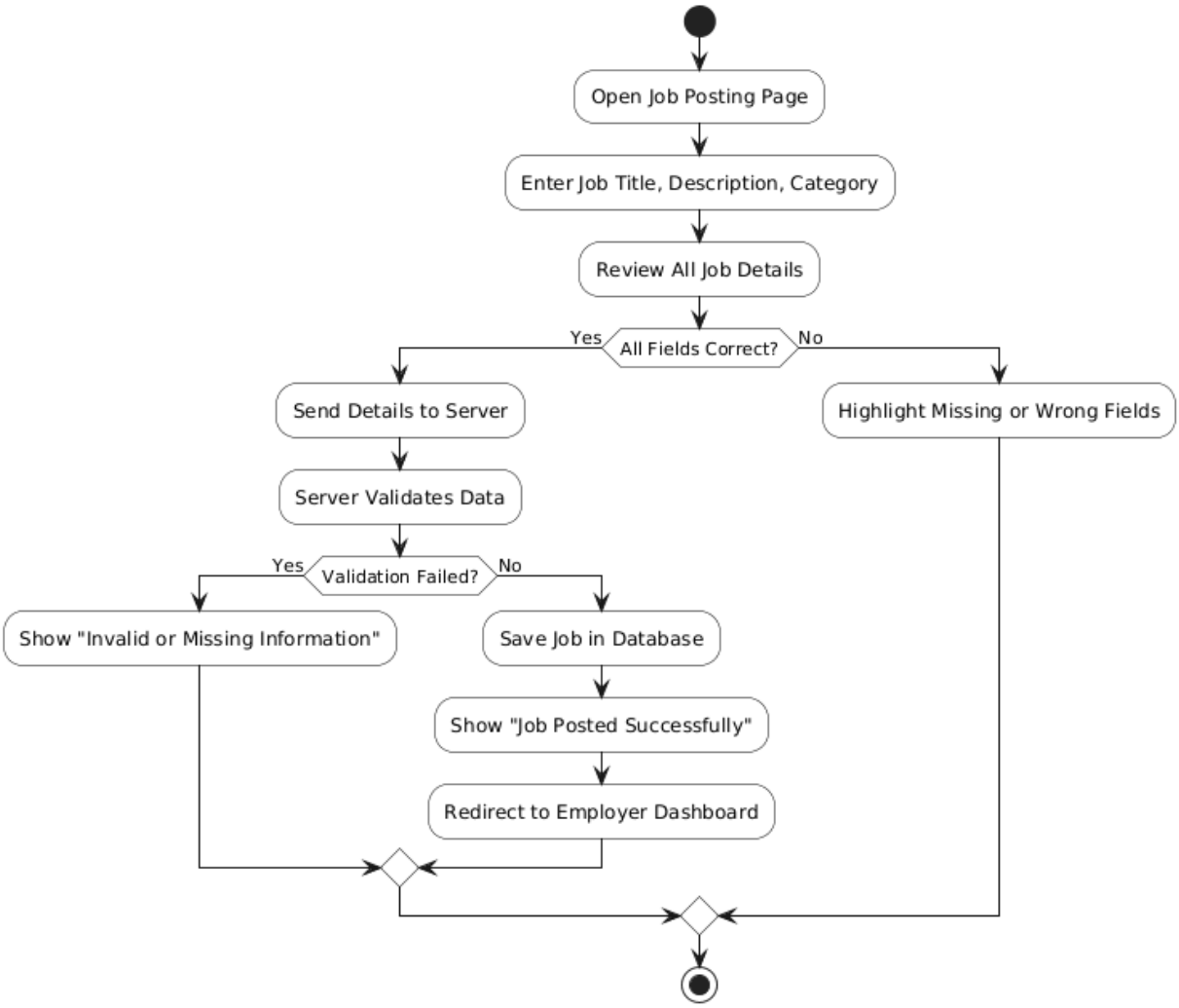

Figure 4.12: Job Posting Activity Diagram

4.6.4 Job Price Prediction Activity Diagram

The workflow of ML (Machine Learning) price prediction module is described by the activity diagram. The user chooses a job type and provides job information. The information is posted to a Flask API, which uses the trained ML model to predict an estimated job price. The estimated cost is then returned and presented to the user.

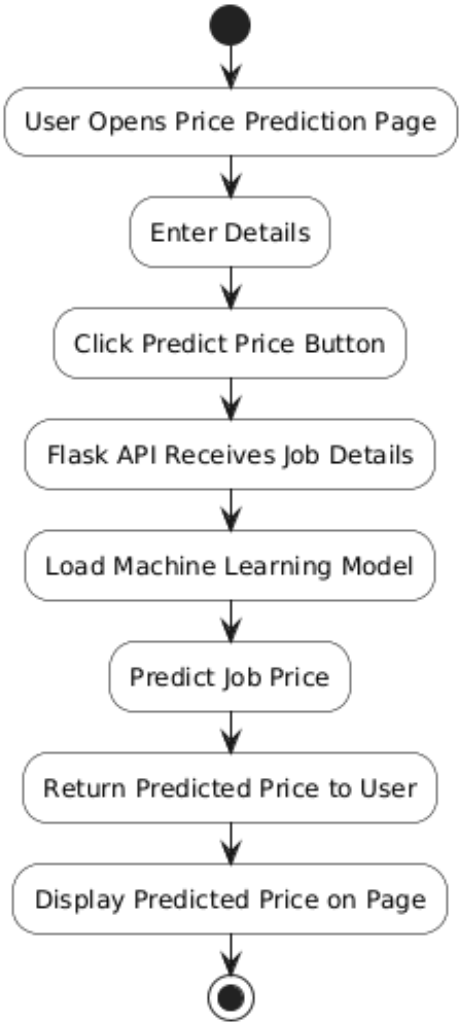

Figure 4.13: Job Price Prediction Activity Diagram

4.6.5 Admin Verification of Worker Account Activity Diagram

The administration verifies worker accounts in this activity diagram shown. Worker accounts are left in a pending state after profile setup. The admin that created the listing reviews the information and documents supplied by the worker. The account is either approved or rejected on the basis of verification results. This procedure guarantees validity and confidence in the system.

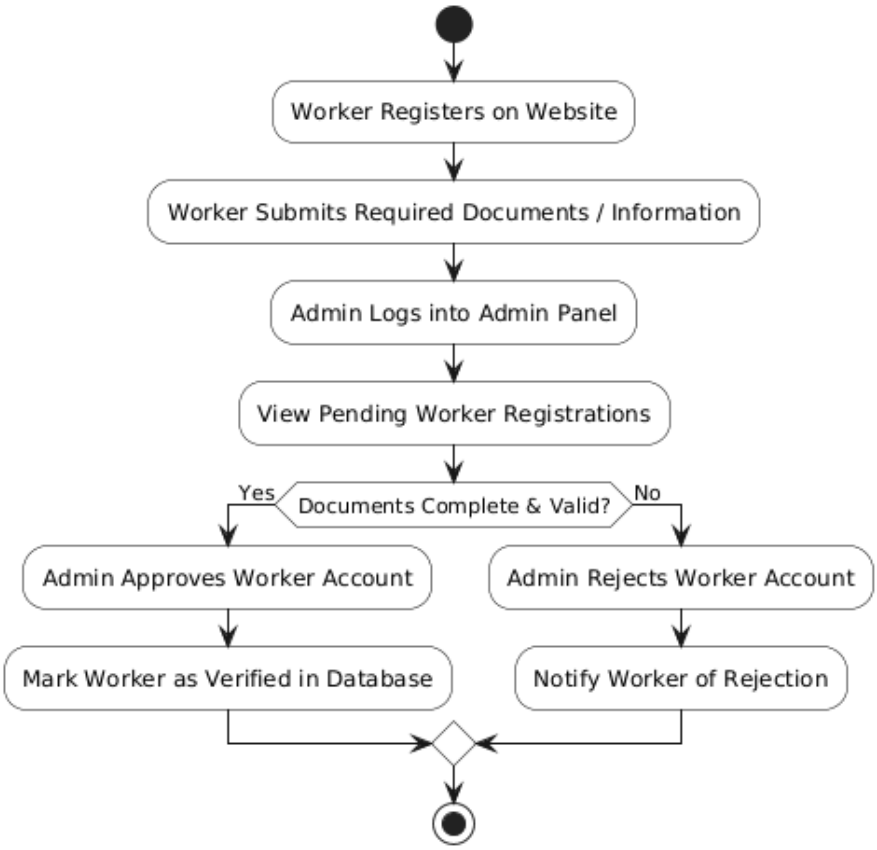

Figure 4.14: Admin Verification Activity Diagram

4.6.6 Rating and Feedback Activity Diagram

In the Rating and Feedback activity, the employer chooses a finished task and gives a rating with comments. The system saves this information in the database and modifies the corresponding user posts. This is a way to keep service quality and accountability..

Figure 4.15: Rating and Feedback Activity Diagram

4.6.7 Incident Reporting and Account Suspension Activity Diagram

This activity diagram describes incident reporting and handling in the Jobs Bridge system. An incident report is submitted by a worker or employer via the system. The admin reads the report and acts accordingly. In case the incident is deemed as valid, the admin can suspend the worker account and further restrict actions, in order to keep the system safe.

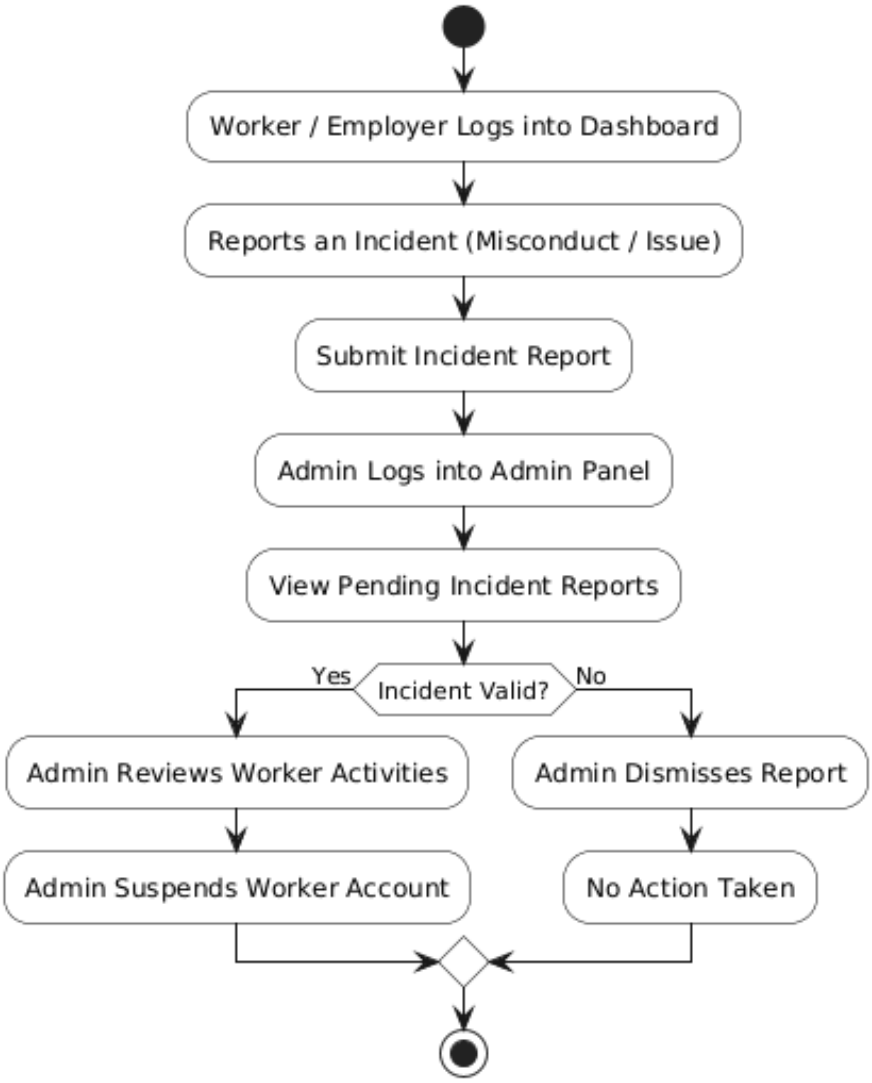

Figure 4.16: Incident Reporting Activity Diagram

4.6.8 Job History Activity Diagram

The job history activity diagram illustrates the flow of activities to maintain and show the information of the job record in the past. Employers can view all jobs they have assigned, while workers can view all jobs they have completed. The system accesses the database for the information and presents it in the history, which enables transparency and a record of performance.

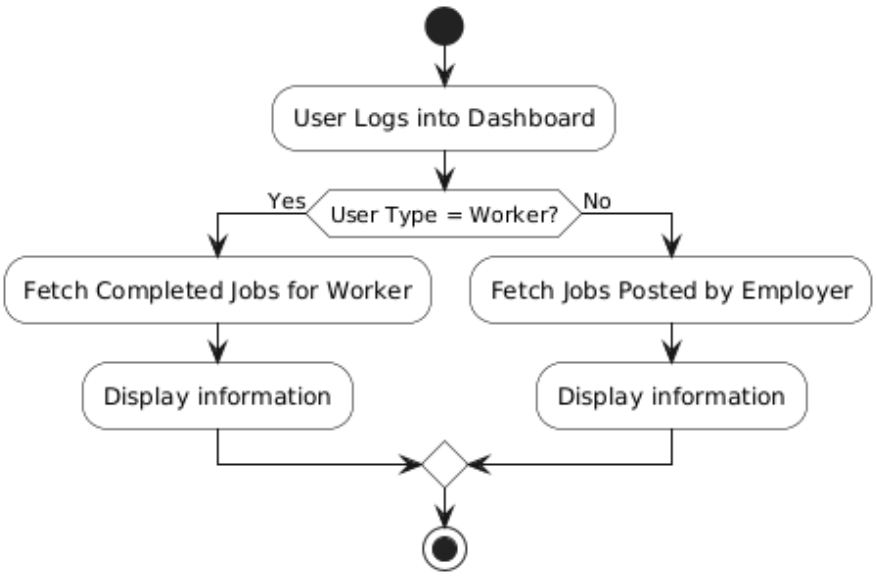

Figure 4.17: Job History Activity Diagram

Chapter 5

System Implementation

5.1 Introduction

In this chapter, we show the full implementation of the Jobs Bridge solution, discussing how the architecture formulated in Chapter 4 was transformed into an end-to-end web application. The majority of the platform was built with the MERN stack (MongoDB, Express.js, React.js, and Node.js), with Python being used in the price prediction module to generate machine learning based price predictions. The system combines user authentication, job posting and application procedures, employer and worker communication, rating and reviewing systems, and multilingual (Urdu) support via a third party API. Each of the subsystems is built to be standalone and, at the same time, interwork with one another to ensure maximum scalability and reliability, as the high-level design allowed for. This chapter also includes instructions on how to build the development environment, how to set up the system, how to use the backend and frontend modules, the database schema, the APIs, the Python prediction service, and how to run the resulting Jobs Bridge platform.

5.2 Development Environment

The Jobs Bridge application was built on a MERN-based stack for its robustness, scalability, and applicability for contemporary full-stack web app development. The entire application was implemented and tested on a Windows PC with Visual Studio Code (VS Code) as the main IDE. VS Code also gave a great environment to juggle the React front end, Node.js and Express backend, MongoDB database, and Python price prediction module.

The system's frontend and backend were built with React.js and Express.js, while MongoDB was used as database to save user profiles, job postings, applications, and ratings. The price prediction feature was developed with the help of a minimalistic Python environment, and the price prediction module was implemented as a Flask-based RESTful API and integrated with the MERN backend. Support for the Urdu language was enabled

using a Google Translate API that was set up during the development. Model training and evaluation of the price prediction module were done in a Jupyter Notebook before being containerized as a Python microservice. The following handling of the MongoDB cluster, authentication rules, and API access was done on the MongoDB dashboard. Integration of the whole system's components was tested in the VS Code workspace to verify every module got along with each other well. The key steps in setting up the development environment were

- **Install Visual Studio Code:** VS Code is installed and configured with extensions for JavaScript, React, Node.js, and Python.
- **Set Up Nodejs Environment:** Node.js was employed both at the frontend in React and at the backend in Express, for all around JavaScript development.
- **Setup Python Environment:** A distinct Python virtual environment was established for the implementation of the price prediction module.
- **Install Dependencies:** For the frontend and backend dependencies such as React, Tailwind CSS, Express, Mongoose, and Axios, npm was used, and for the Python libraries for data preprocessing and model prediction, pip was used.
- **Database Setup:** MongoDB is set up for data storage with authentication and access rules.
- **Project Integration and Testing:** All modules were combined in VS Code to work coherently, from frontend to backend to database to the Python prediction service.

5.3 Tools and Technologies

The development tools and technologies of Jobs Bridge are modern, practical, and popular in full-stack web development. These tools allow the frontend, backend, and database to be seamlessly integrated.

5.3.1 Visual Studio Code (IDE)

VS Code is the IDE selected for this project. It has a clean and simple UI packed with a terminal, live server, debugging features, etc. Developers can add extensions for JavaScript, Python, and many more technologies for great project development even if they are not connected to the internet all the time.

5.3.2 React.js Framework (Frontend)

The Jobs Bridge frontend is developed using React.js, a JavaScript framework that is fast and component based. We ensure that there is a responsive and dynamic page for exchanging information between users and workers interact efficiently. React is supported by:

- **Tailwind CSS** styling the UI in a fast, responsive, and utility based way.
- **Lucide React & React Icons** for a consistent and modern icon style.
- **React Toastify** for real time notification and status message.
- **Axios** to handle the API calls smoothly from front end to back end.
- **Chart.js and React Chart. js** forrending analytics and insights.

5.3.3 Node.js and Express (Backend)

The system's backend is written in Node.js using Express. It handles user authentication and job posting and acts as a communicator of information between the frontend and the MongoDB. Node.js provides an event driven architecture, which is efficient and suitable for real-time web applications.

5.3.4 Python and Flask API

Python is leveraged to construct and train the machine learning model used in the price prediction feature. The trained model is wrapped into a lightweight Flask API that takes user inputs and predicts estimated prices and sends the results to the frontend in real time.

5.3.5 Database MongoDB

MongoDB is the database that stores the users, the workers, the job posts, and the reviews. It offers flexible data storage through a NoSQL model and guarantees scalability for massive data sets.

5.3.6 Other Libraries and Tools

- **Nodemailer** for sending confirmation emails and notifications.
- **Cloudinary** to upload images and save them safely.
- **Free OCR API** to extract text data when needed.
- **Google Translate API** to add support for Urdu language translation.

5.4 Installation and configuration of the system

This section illustrates the entire process from setting up to configuring the Jobs Bridge system. The installation involves the backend, frontend, database, and the price prediction service based on Flask. All were installed, controlled, and tested under Visual Studio Code to guarantee the flow of developing was quite fluent.

5.4.1 The project organization setup

The system adopts a modular folder structure, which aims at separation of concerns and maintainability. It's split into:

- **client/**React.js,pages and styles for the frontend.
- **server/**Node.js and Express routes for the backend routes, controllers, and middleware.
- **models/**Mongoose schemas using MongoDB for users, jobs, and reviews.
- **python_service/** Price prediction API based on Flask and ML model.
- **config/** Environment variables and database configuration files.

It allows the application to be scalable and clear separation between the frontend, backend, and microservices.

5.4.2 Node.js/Express Backend Setup

The backend was initiated with Node.js and structured with Express. Steps in setting up the configuration.

- Starting the project with npm init.
- Required back-end dependencies: Express, Mongoose, Cors, dotenv, Bcrypt, and JSON Web Tokens.
- The following route handlers: authentication, job posting, worker applications, and sending in a review.
- Middleware for validation, CORS, and secure token-based authentication was also set-up.
- Secure Local Connection to MongoDB The following line connects us to our local MongoDB server through a secure connection string(saved in the `.env` file).

5.4.3 Frontend Configuration (React.js)

The frontend was built with React.js as a single page application. The configuration consisted of:

- Initialization via Create React App.
- Installing UI and utility libraries: Tailwind CSS, Axios, React Router, React Toastify and Chart.js.
- Setting up routing for login, dashboard, jobs, applications, and profile.
- Added API calls to the Express backend with Axios to get data in real time.
- Generic components for buttons, forms, tables, cards, notification alerts, and more.

5.4.4 Database Configuration (MongoDB)

MongoDB was adopted as the main database for the Jobs Bridge platform, as it allows a flexible, schema-less storing of the application data. A normal MongoDB environment was established in which collections were created for users, employers, workers' jobs, applications, and reviews. The MERN backend, built with Node.js and Express, communicates with MongoDB using the Mongoose ORM, which allows defining schemas, validation, and a structured approach to access the stored data. This MongoDB database design offers effective CRUD operations, elastic data storage.

5.4.5 Environment Variables and API Keys

The sensitive information, such as passwords, was saved securely with environment variables for the security of the application. These include:

- MongoDB connection string.
- JWT secret key for authentication.
- Cloudinary API credentials for image hosting.
- Google Translate API key for support for the Urdu language.

Environment variables are kept in `.env` files and are ignored by version control to prevent leaking of system information.

5.5 Module-Wise Implementation

5.5.1 User Interface Implementation

The user interface is built with React and consists of several pages with which users and employers can engage with the Jobs Bridge site. Secure sign-in and sign-up are allowed by the Authentication Page. The active jobs and application summaries are displayed in the Dashboard Page. Jobs are listed on the Job Listings Page. Users can view and edit profiles, write reviews, and see past history through the Profile and Review Pages.

5.5.2 Authentication Service Implementation

User authentication is based on JWT authentication on the Node.js server. The React interface sends the credentials to the service that checks the credentials against the MongoDB records and returns a signed JWT token. On the client side, tokens are securely stored and validated on each API request to allow protected route access.

5.5.3 MongoDB Database Service Implementation

All data operations in Jobs Bridge are handled by MongoDB. User, employer, job, request, and review are the collections. Mongoose models are used in the backend to define schema, get the data validated, and perform CRUD operations. Queries are tailored for efficient retrieval (dashboard, job search, analytics).

5.5.4 Job Posting and Application Pipeline

The job processing pipeline handles job posting, updating, and deleting job posts, and responses from workers. When a job is posted by an employer, it is saved in MongoDB. Workers can send requests to jobs, and these requests are tied to both the job and the applicant in the database. Business rules (e.g., a worker can not request the same job twice) are enforced by the backend.

5.5.5 Review and Rating Module

The Review and Rating module lets employers review their workers after a job is done. Employers have the option to rate the applicants and write reviews in the frontend that are submitted to the backend. The Node.js backend saves the review and rating in the MongoDB database, and the This module is to be designed to ensure the evaluation of matter to be transparent and quality work to be rewarded, and trust between employers and workers to be stronger.

5.6 Price Prediction Module

The Jobs Bridge site comes with a price prediction tool that allows the estimation of jobs. This module is executed with trained machine learning models in Python as a Flask-based RESTful API.

5.6.1 Processing logic

The following enumerates the normal input-processing output flow of work for the price estimation feature:

5.6.2 Processing Logic

The price prediction feature follows the system's standard input-processing-output workflow:

1. **Input:** The employers enter the suitable parameters of the job, such as the type of job, its description, required hours, and urgency level. This data is posted in the frontend, and it goes to the Flask API.
2. **Processing:** Input data is received by Flask and forwarded to the trained machine learning model associated with the chosen service module. The model predicts a notified price for the job using past data and the given parameters.
3. **Output:** The predicted job price is returned by the API to the MERN backend and is visible in real-time on the employer's interface. This allows employers to make informed decisions about posting job opportunities.
4. **Repeat:** Employers can modify parameters and submit more jobs, and the system will predict each new jobs.

5.6.3 Model Integration

The machine learning models are trained using historical job data and saved as .pkl files. The Flask API loads the appropriate model at runtime based on the selected service module, enabling the backend to compute predictions efficiently. Cross-origin settings are configured to ensure seamless communication between the Flask service and the Node.js backend.

5.6.4 Modules and Model Performance

The price prediction service consists of three dedicated modules: Painting, Electrician, and Plumber. Table 5.1 summarizes the models used and their performance metrics on the test dataset.

Table 5.1: Price Prediction Modules and Model Performance

Service Module	ML Model	R^2 Score	MAE (PKR)	RMSE (PKR)
Painting	Random Forest Regressor	0.97	12,008	19,780
Electrician	Random Forest Regressor	0.96	4,512	8,489
Plumber	XGBoost Regressor	0.97	10,046	20,072

5.6.5 Performance Overview

All three regression methods showed good predictability in the test data. Random Forest was the best performing algorithm for the Painting and Electrician modules, and XGBoost was better than other algorithms for the Plumbing module. Reporting MAE and RMSE along with R^2 leads to a more realistic evaluation of prediction accuracy and makes sure that the employers receive trustworthy price estimates for their jobs to post. The models were exported to .pkl files and connected to the Flask backend, allowing for serving real-time predictions on the Jobs Bridge website.

5.6.6 Integration of Subsystems

All components of the Jobs Bridge runs together, as one cohesive system. At the React frontend, the user communicates with the Node.js backend, while MongoDB stores all user, job, application, and review data. The price prediction Flask service is added as a module on the frontend that allows users to estimate the cost in real time. Each of the modules Authentication, Job Posting, Applications, Reviews, and Prediction all adhere to the same data schema and definition, and information is exchanged freely between these modules, allowing a silky, bug free, full experience workflow across the entire platform.

5.7 GUI Design

The Jobs Bridge System with GUI has been designed according to the user-centered approach that meets the user's convenience in all facets and is designed to be simple, straightforward, and user-friendly. The interface design is aligned with modern web design practices, including but not limited to responsive design, visual hierarchy, and repeating design patterns throughout all system modules. The GUI provides simple communication protocols between workers, employers, and administrators and is professional and structured enough to be used in the hiring process.

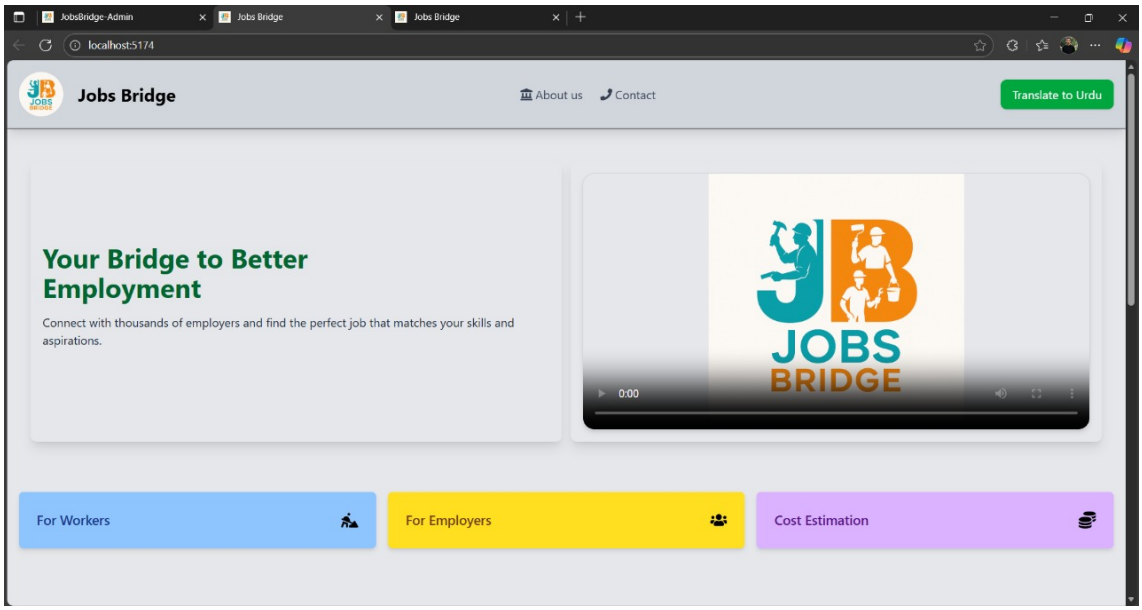

Figure 5.1: Landing Page Interface

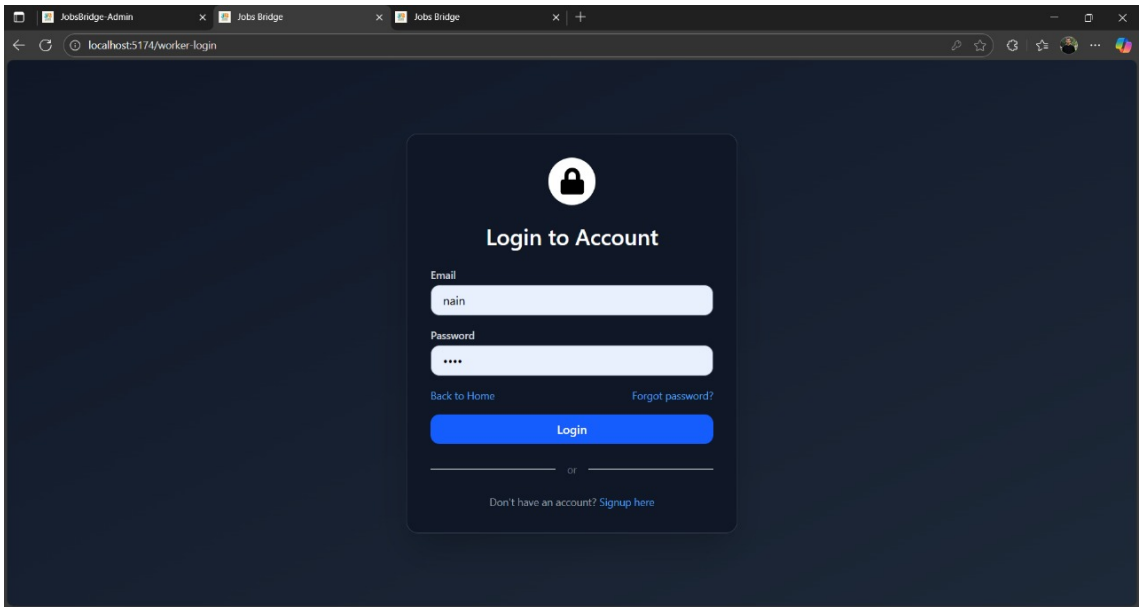

Figure 5.2: User login Interface

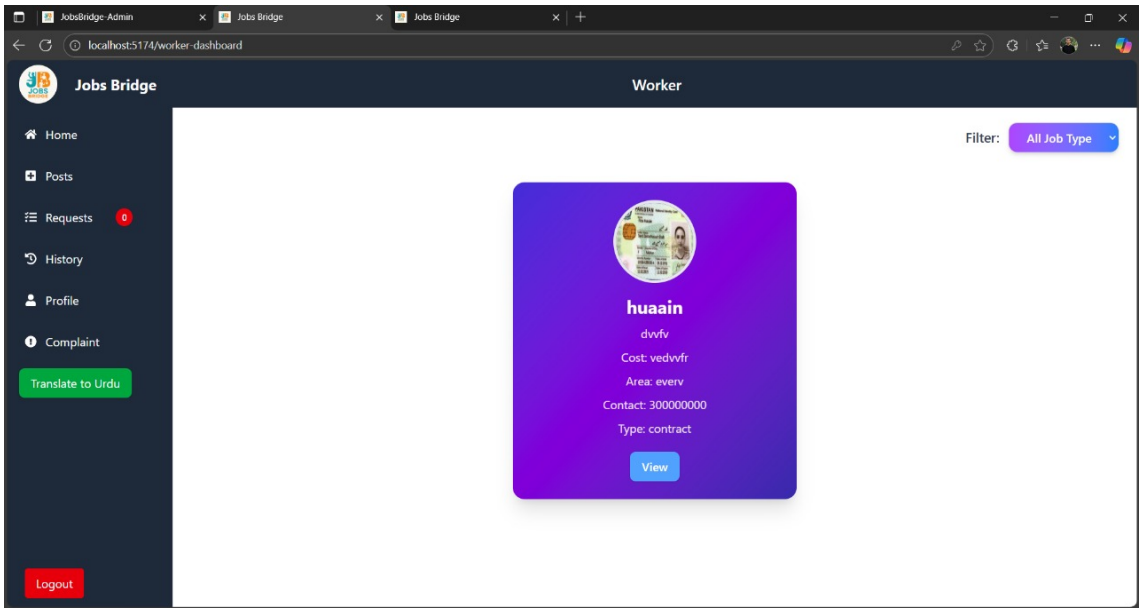

Figure 5.3: Worker's Dashboard Interface

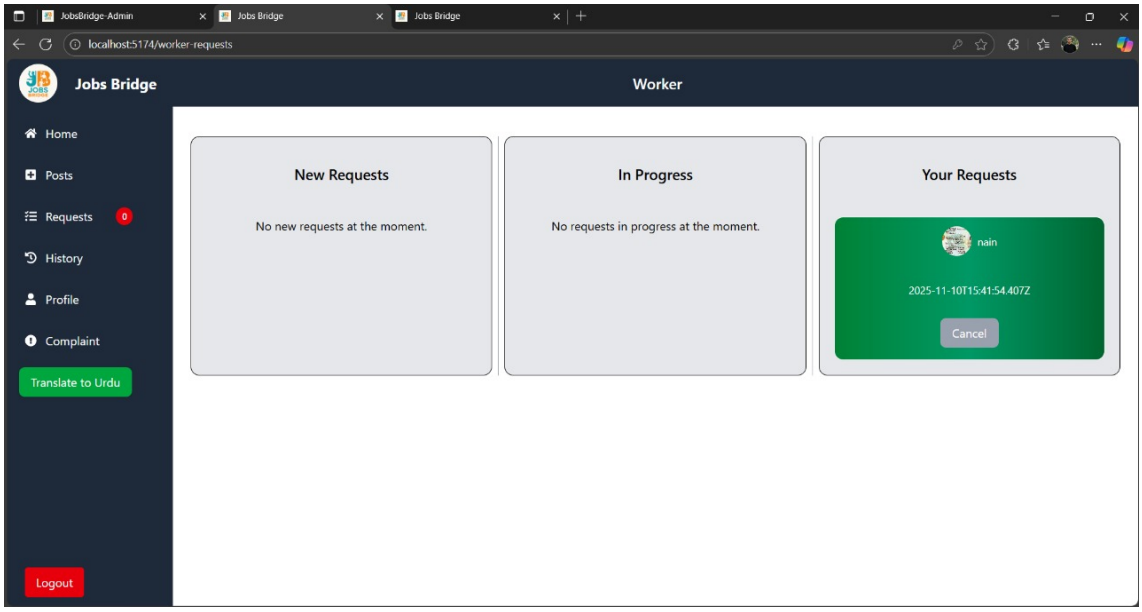

Figure 5.4: Available job Interface

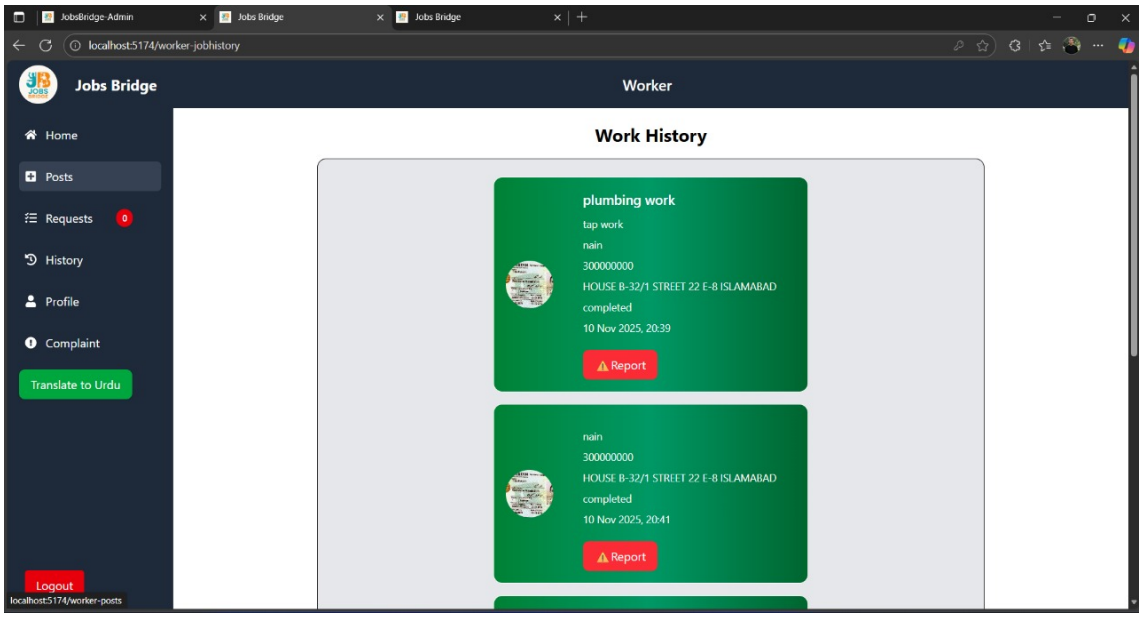

Figure 5.5: Worker's Work history Interface

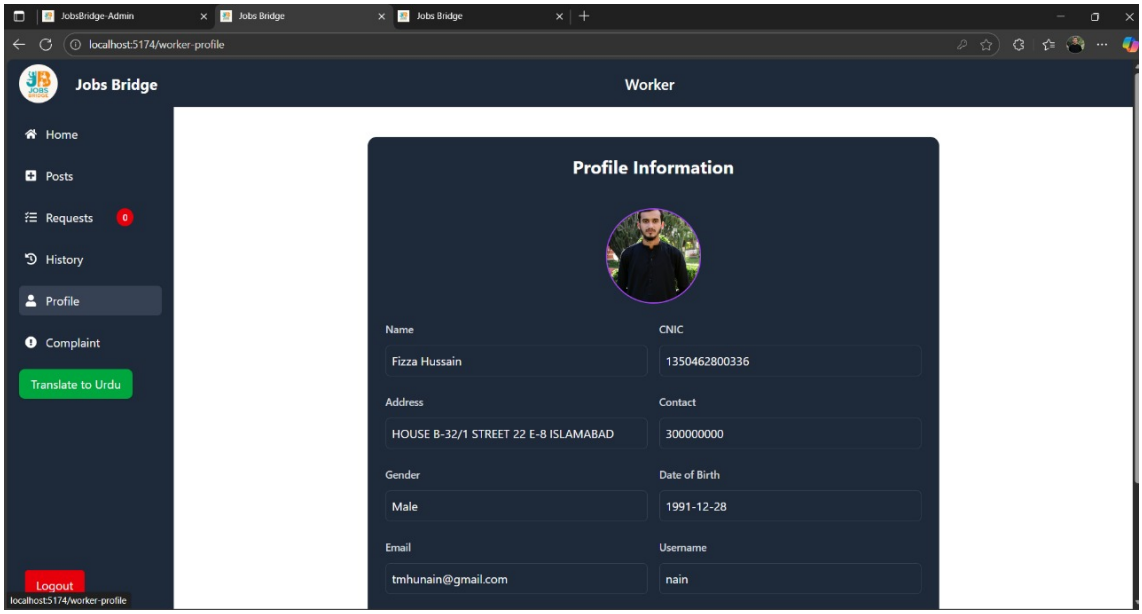

Figure 5.6: Worker's Profile Interface

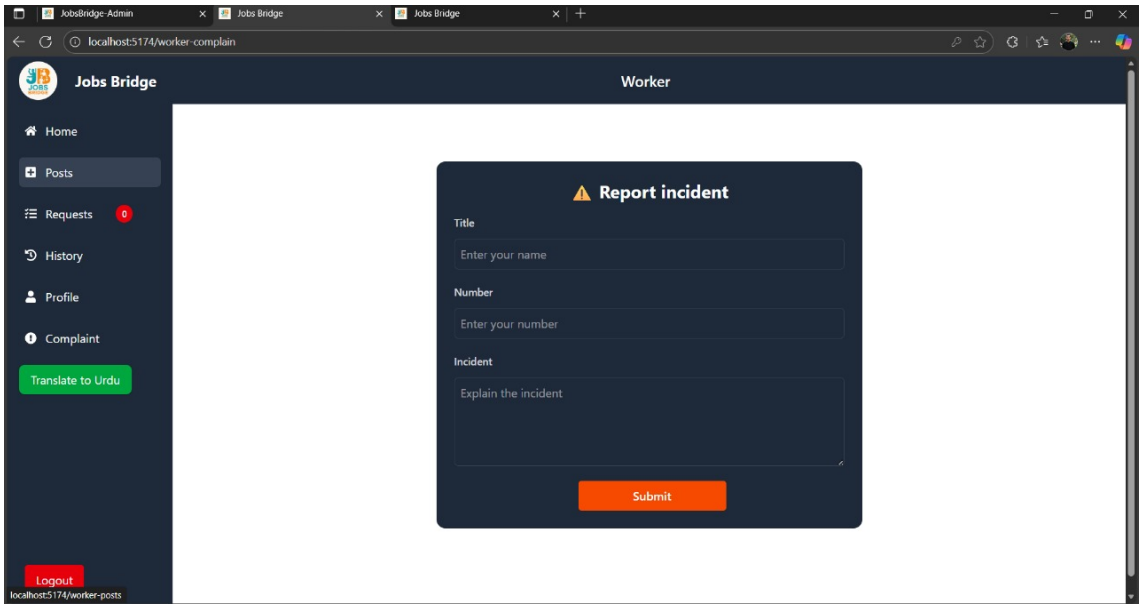

Figure 5.7: Workers complain

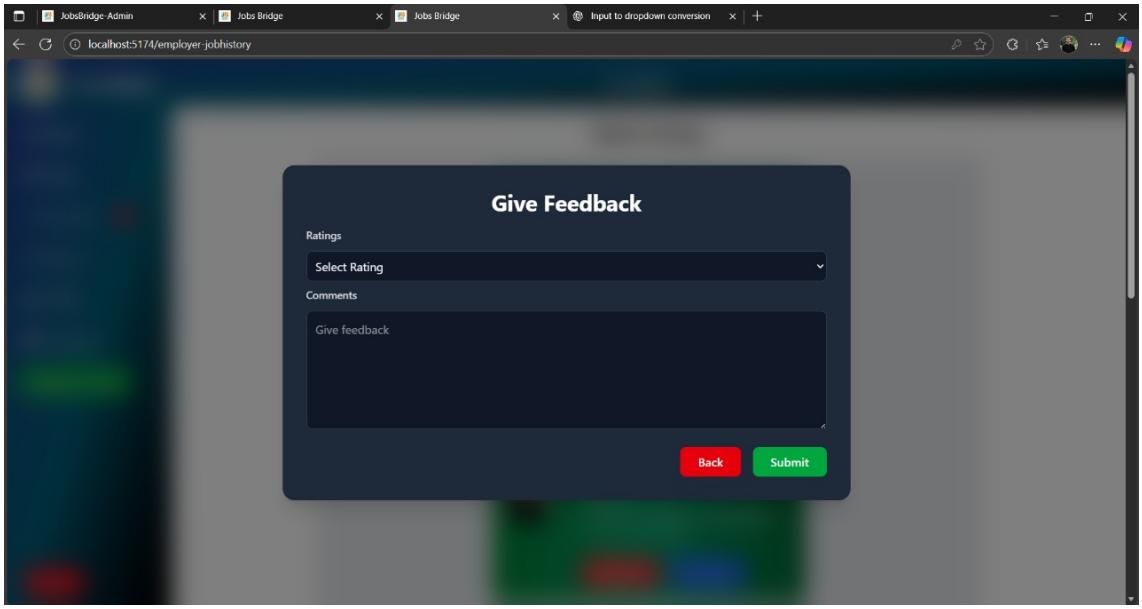

Figure 5.8: Employer rating and feedback

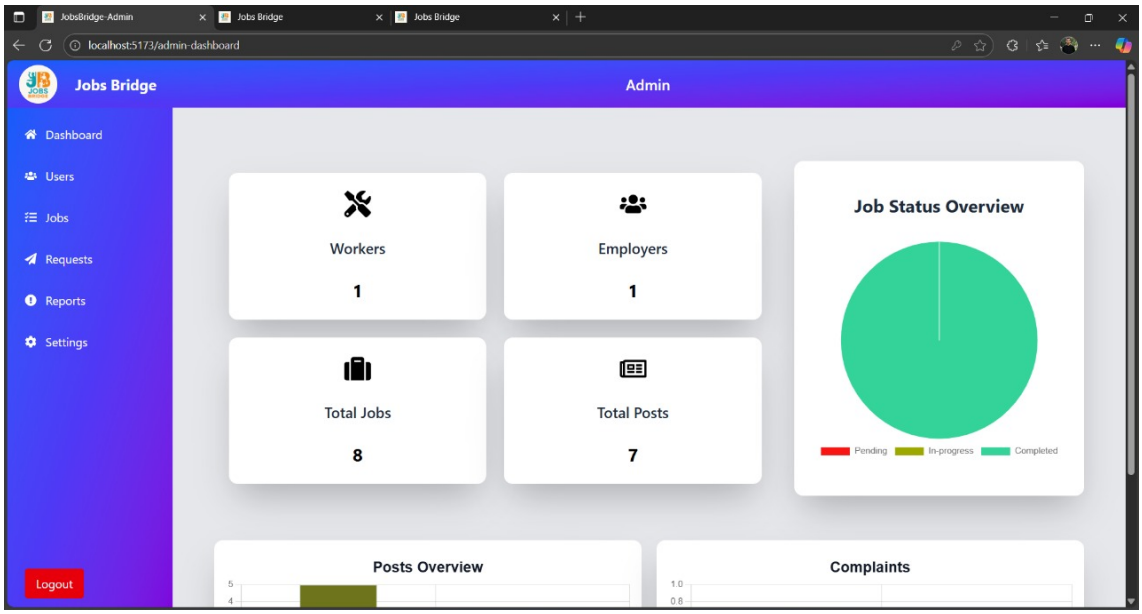

Figure 5.9: Admin Dashboard

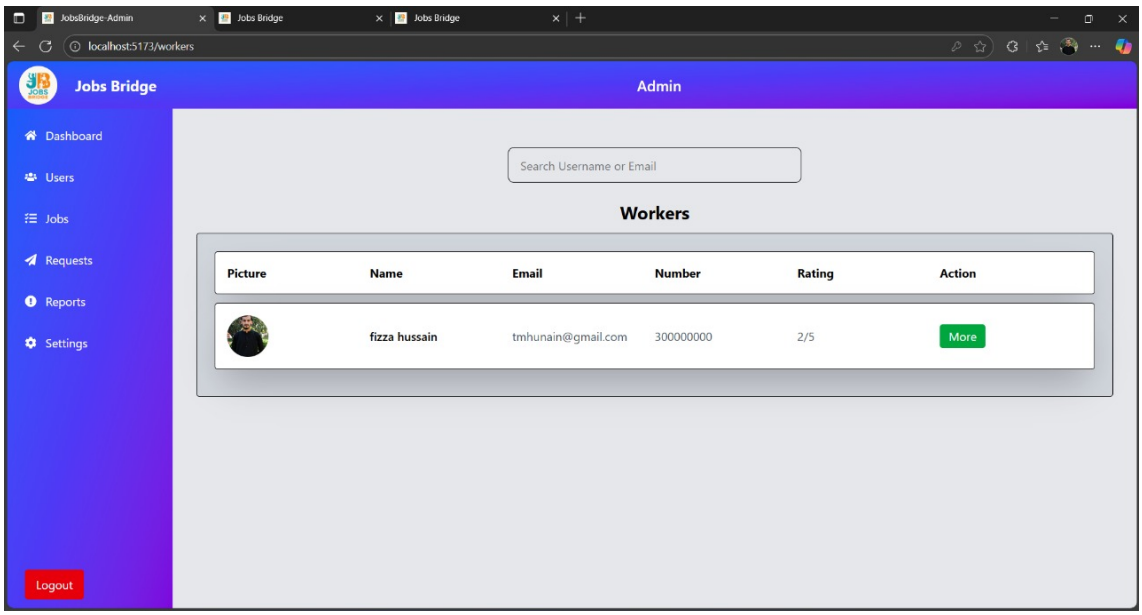

Figure 5.10: Workers Details in Admin Dashboard

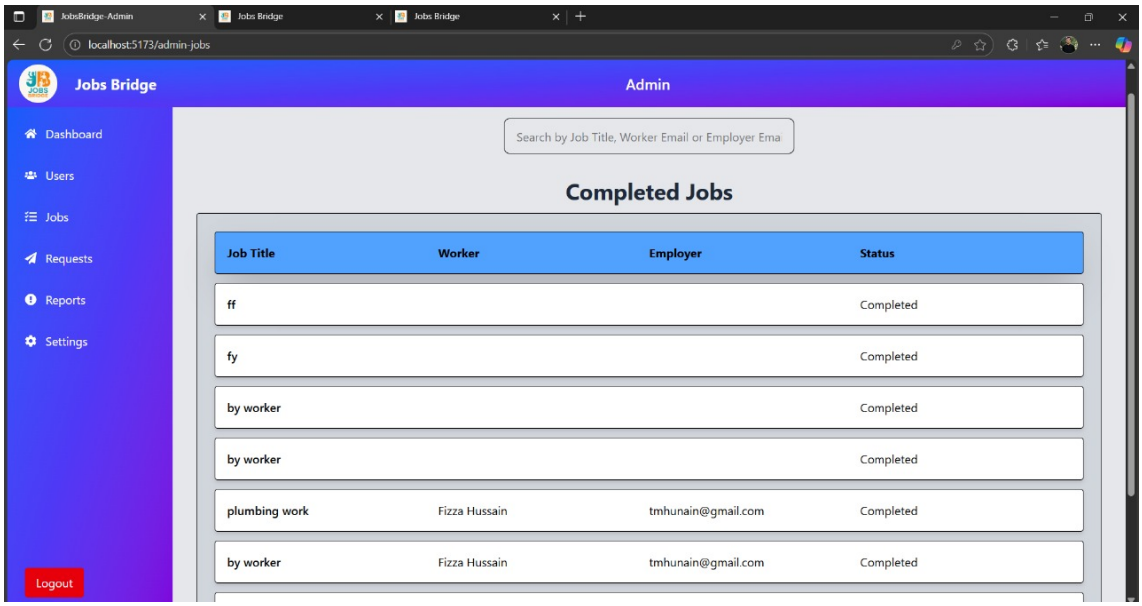

Figure 5.11: Jobs History in Admin Dashboard

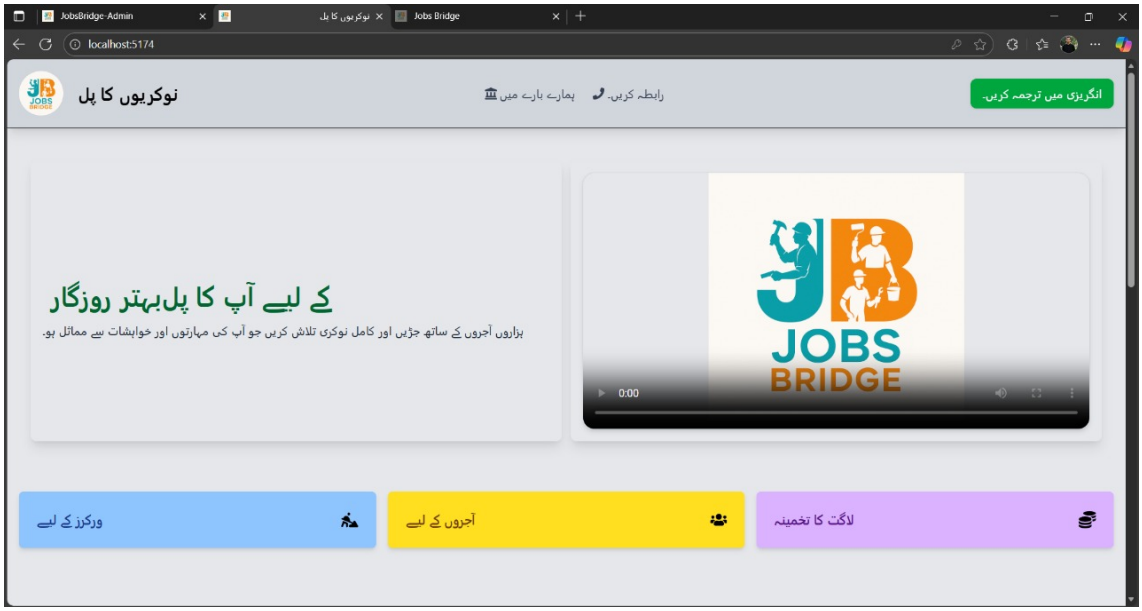

Figure 5.12: Urdu Interface

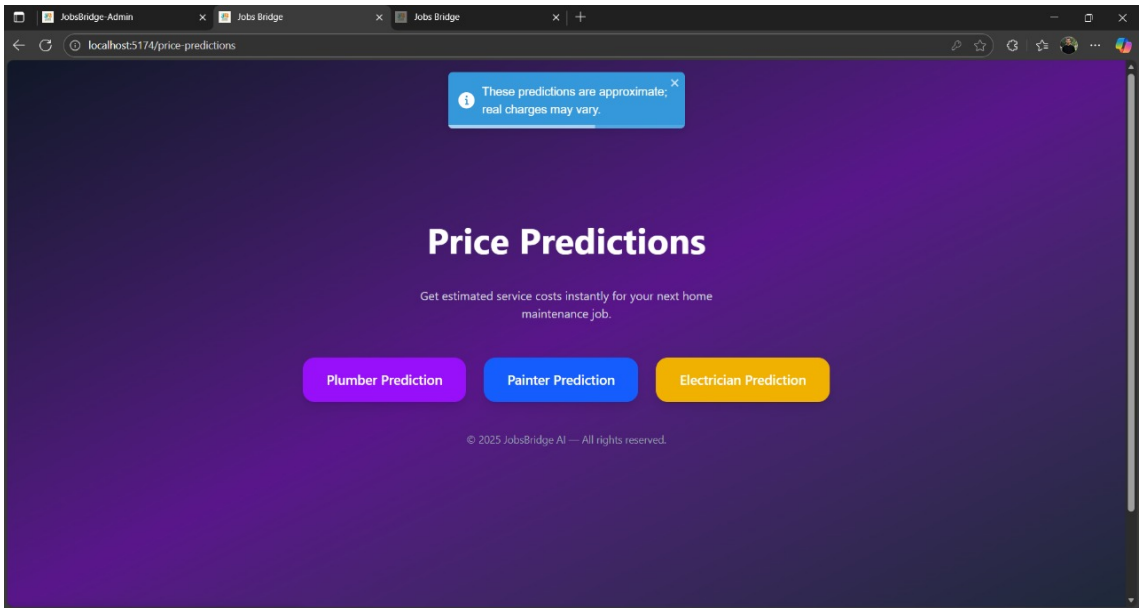

Figure 5.13: Price Prediction Interface

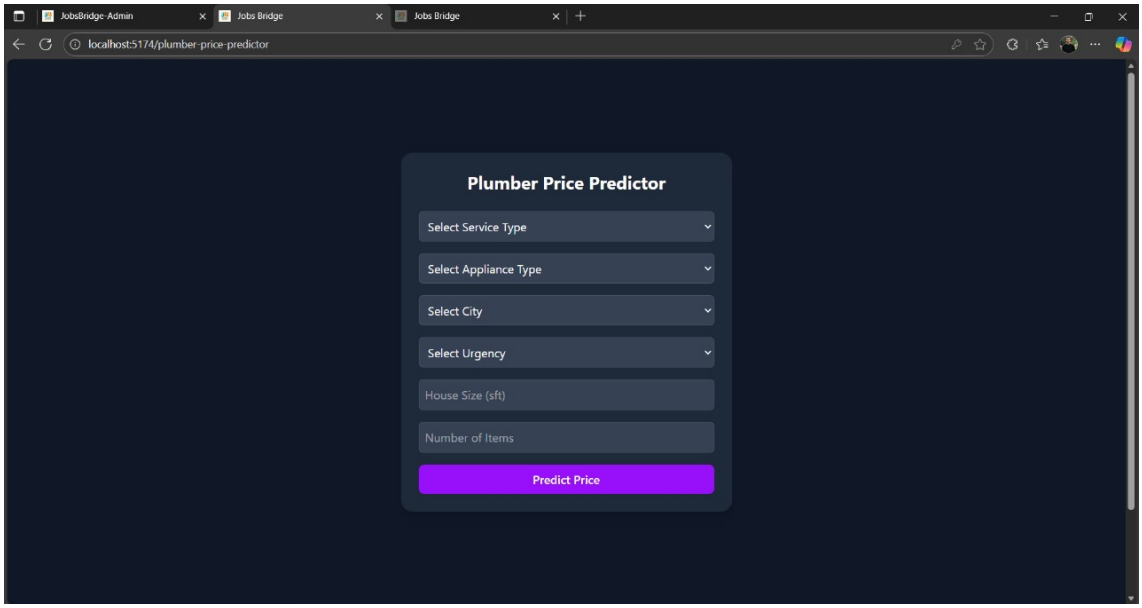

Figure 5.14: Plumbing Service Module Interface

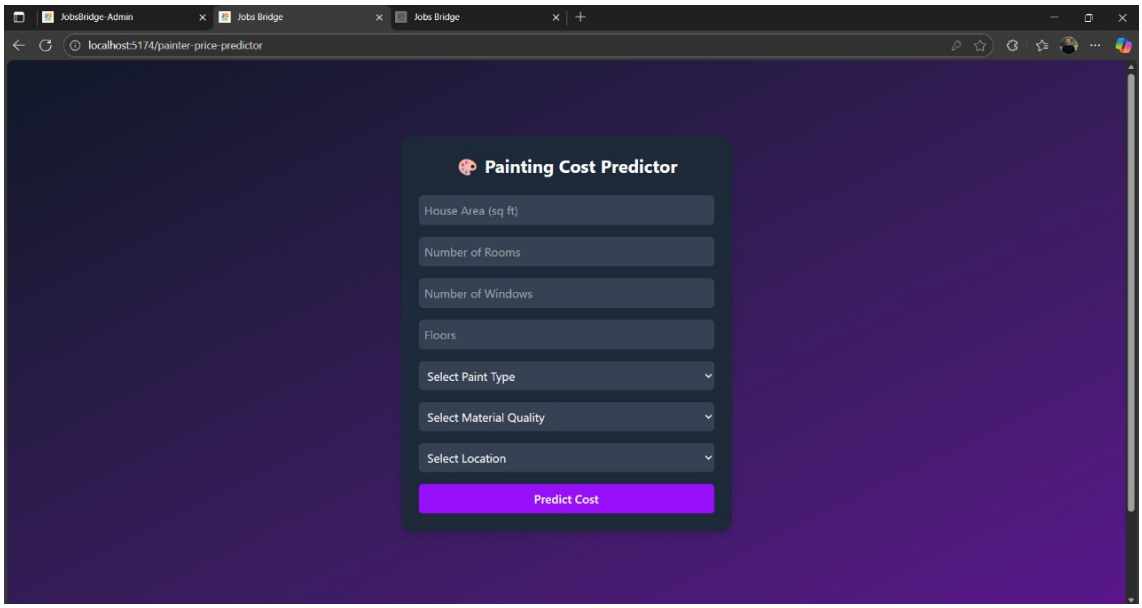

Figure 5.15: Painting Service Module Interface

Chapter 6

System Testing and Evaluation

System test is a high-level test to check whether the whole system works as per the expectations and meets the requirements. It contributes significantly to system and application development by detecting errors and defects and providing reliability to the performance. This integrated review is comprised of both qualitative testing, which is based on detailed user needs, and quantitative testing, which is an objective measurement of group dynamics. Since there is no such thing as a perfect system, the system testing phase serves to show the strengths and weaknesses of the system. Various testing techniques are applied during this phase to detect and resolve issues

6.1 Testing Strategy

6.1.1 Testing Objectives

The system testing goals for the Jobs Bridge job matching site are:

- To ensure that all the functional requirements are met and that each of the workflows (post a job, browse jobs, apply for a job, status update, and reviews) works as expected.
- To ensure that the frontend website, backend APIs, database operations, and optional price-prediction module together constitute an integrated system that does not disrupt any workflow.
- To assess the scalability, achievability, and resiliency of the Jobs Bridge platform for both worker and employer.

6.1.2 Test Levels and Types

Even though unit and integration tests have been run informally during development, we will focus on system tests in this chapter. The Jobs Bridge system has the following test cases:

6.1.3 Test Levels and Types

This chapter concentrates on testing at the system level. The following test types will be conducted at the system level for the Jobs Bridge Platform:

- Graphical User Interface (GUI) testing on all screens of the website.
- Functionality testing according to the identified FRs.
- Performance testing on API response time.
- Robustness and exception-handling tests for scenarios such as invalid login attempts, use of incomplete forms, network disconnection during application job posting, and backend API crash.
- Security testing of user authentication, role-based access control, session management, and security of job posts and applications from unauthorized access.

6.1.4 Unit Testing

Unit tests were performed informally during development. Unit tests were run on small, decoupled components of the Jobs Bridge system, such as the form validation logic, job posting handler, database write, and the price prediction API. These allow us to verify that each component is functioning properly on its own before they are combined into the final product. Testing was limited to critical parts, as the emphasis of the work is on the behavior of the fully integrated system.

6.1.5 System Testing

System testing makes sure that everything in the Jobs Bridge application, including visual elements and user interactions, behaves as expected. The one validating that all the modules, like worker registration, employer job posting, job application, price prediction, etc., respond as intended to user inputs. This includes verifying that data transfer between frontend and backend is real, that navigation is perfect, and when in doubt, each feature on the platform is tested if needed. It also confirms that transitions between different sections

of the system, from the job posting to the job request and from registration to verification and so on, are seamless with not too much delay or errors. System testing is also important to verify that the application design and flow result in a system that is easy to use, efficient, and free of errors. In short, by rigorously testing both the functionality and design of the system, we make sure that Jobs Bridge is a solution that works well and is a quality product that all users workers, employers, or the admin can rely on.

6.1.6 Test Case 1: User Registration with OTP Verification

Verify a new user is correctly registered in Jobs Bridge by entering all mandatory fields, e.g., name, email, and password. A further security layer with OTP (One-Time Password) verification is added to check the user's identity before creating an account. The goal is to verify that the system handles registration cleanly with appropriate confirmation messages and with secure storage of verified user information.

Table 6.1: Test Case for User Registration with OTP Verification

Test Case ID	TC- 01
Description	Registering a new user in the system with OTP verification.
Requirement	The user must input name, email, and password, and verify their identity using a valid OTP sent via email.
Steps to be Taken	1. Go to the registration page. 2. Enter valid details in the registration form. 3. Send the form and get an OTP. 4. Type in the OTP received in your e-mail to verify your identity. 5. After a successful OTP verification, complete the registration.
Expected Result	A new user account is created after successful OTP verification, and a confirmation message is displayed.
Actual Result	The user account was successfully created after OTP verification, and the confirmation message was displayed.
Status	Success
Remarks	OTP verification ensures secure and authentic user registration.

6.1.7 Test Case 2:User Login

The test case is related to login for existing users of Jobs Bridge. It checks if the system properly logs in with valid credentials; for example the email is registered and the correct password is entered. Once the user logs in successfully, they will be redirected to their dashboard (employer or worker), which means that the login are working fine.

Table 6.2: Test Case for User Login

Test Case ID	TC-02
Description	To verify that registered users can successfully log in using valid credentials.
Requirement	The user must be registered and the account should be active.
Steps to be Taken	1.Navigate to the login page. 2. Type in your registered email and password. 3. Press the "Login" button to continue. 4. The credentials are verified by the system with the help of stored information.
Expected Result	The system verifies credentials and redirects the user to their respective dashboard.
Actual Result	The user successfully logged in and was redirected to the correct dashboard.
Status	Success
Remarks	Login authentication worked as expected.

6.1.8 Test Case 3: Job Posting

This test case is designed to check whether registered employers can post jobs on the Jobs Bridge site. It verifies the ability of authenticated employers to post jobs, including title, description, wage and so on, and then submit the job listing to the system. The test is designed to ensure that job information is saved in the database and is shown in the public job listings.

Table 6.3: Test Case for Job Posting by Employer

Test Case ID	TC-03
Description	To verify that a registered employer can successfully post a new job with valid details.
Requirement	The employer must be logged in and should have access to the job posting dashboard.
Steps to be Taken	1. Navigate to the employer dashboard after login. 2. Click on the "Post a Job" button. 3. Fill in all required fields such as job title, description, location, category, and wage. 4. Submit the job posting form.
Expected Result	The system validates the entered information, saves the job post in the database, and displays a confirmation message.
Actual Result	The job was successfully posted, and a confirmation message was displayed.
Status	Success
Remarks	Data was correctly stored in the database, and the job became visible in the job listings.

6.1.9 Test Case 4: Worker Profile Creation (with OCR API)

This test case will check creating of a worker profile. It allows workers to create profiles by entering personal information and scanning their ID card and the system automatically extracts and verifies data through OCR module. The API allows the OCR module to read ID card data, eliminating the need for manual entry and improving the accuracy of the data. This is to ensure that captured data (name, CNIC, DOB etc.) is saved into the database and linked with the worker's account after process.

Table 6.4: Test Case for Worker Profile Creation with OCR API Integration

Test Case ID	TC-04
Description	Verify that workers can create their profiles and that ID data is extracted and validated automatically using the OCR module API.
Requirement	Worker must be registered and logged in before creating a profile. The OCR API must be functional.
Steps to be Taken	1. Registered worker login to the system 2. Upload a valid ID card image. 3. The OCR module extracts data using the integrated API. 4. Verify extracted details and submit the profile form.
Expected Result	The profile is created successfully with data automatically extracted and validated from the uploaded ID card.
Actual Result	The worker's ID data was successfully extracted and stored in the database during profile creation.
Status	Success
Remarks	The OCR module automates data extraction, reducing errors and improving the efficiency of profile creation.

6.1.10 Test Case 5: Job Application by Worker

The test case tests the functionality enabling registered and authenticated workers to submit their requests to apply to the existing job openings on the platform of the Jobs Bridge. It makes sure that the job applications are properly handled by the system and stored in the database. The test is also useful for ensuring that only confirmed workers (who have completed profiles) can apply for jobs. An effective submission will ensure that the job application module and the operation of the database for employer functions are as planned.

Table 6.5: Test Case for Job Application by Worker

Test Case ID	TC-05
Description	Verify that registered workers with completed profiles can successfully apply for available job listings.
Requirement	Worker must be logged in with a verified profile before applying for a job.
Steps to be Taken	1. Log in as a registered worker. 3. Select a job listing from the list. 4. Click the "Request" button.
Expected Result	The system records the application, displays a success message, and notifies the employer of the new applicant.
Actual Result	The worker's application was successfully recorded, and the employer received the corresponding notification.
Status	Success
Remarks	Only verified workers are allowed to apply for jobs.

6.1.11 Test Case 6: Job Price Prediction (Flask Module)

This test case tests the functionality of the price prediction module in the Jobs Bridge platform. This module is developed in Flask, and it forecasts the estimated wage or job price on the basis of the input parameters. The test makes sure that the model is fed the appropriate input by the frontend, runs it through the trained machine learning model, and gives the user the right outputs. It also confirms that the Flask API is efficient on the requests and presents the results of the predictions without delays and system failures.

Table 6.6: Test Case for Job Price Prediction using Flask Module

Test Case ID	TC-06
Description	Verify that the price prediction module accurately predicts job wages based on the provided input parameters.
Requirement	Flask API and trained machine learning model must be running and connected to the frontend interface.
Steps to be Taken	1. User open the website. 2. Navigate to the "Price Prediction" section. 3. Enter job details such as category, details, etc. 4. Submit the form to send data to the Flask API. 5. Observe and record the predicted price returned by the system.
Expected Result	The system returns an accurate job price prediction and displays it on the user interface.
Actual Result	The Flask module processed the input data successfully and displayed accurate wage predictions in real time.
Status	Success
Remarks	The Flask-based API effectively integrates with the platform, providing fast and reliable wage estimations.

6.1.12 Test Case 7: Urdu Language Support (API Integration)

The test case tests the functionality of the feature of Urdu language support of the platform. The system engages an external Urdu language API to translate. The test makes sure that when the users type or make a choice with the Urdu language, the API reads the content, translates the content, and gives the relevant response without compromising the performance of the platform. It will also ensure that the content that was translated is properly represented in the interface.

Table 6.7: Test Case for Urdu Language Support API Integration

Test Case ID	TC-07
Description	Verify that the Urdu Language API accurately translates and processes Urdu text within the platform.
Requirement	Urdu Language API must be active and properly connected to the system interface.
Steps to be Taken	1. Log in to the platform as a registered user. 2. Navigate to the "Urdu." 3. Observe the system's response and verify the translated output.
Expected Result	The system accurately translates and displays Urdu text through the integrated API without errors.
Actual Result	The Urdu Language API successfully processed user input and returned accurate Urdu translations in real time.
Status	Success
Remarks	The Urdu Language API enhances accessibility, allowing bilingual users to interact with the system smoothly.

6.1.13 Test Case 8: Worker Verification by Admin

In this test case, the functionality being tested will consist of testing the functionality to enable the review and approval of worker profiles on the Jobs Bridge platform by the admin. After registration and the creation of a profile, the worker is put in a pending verification state. The administrator examines the information about the worker and supporting documents, like the character certificate, and makes sure that it is authentic. Once approved, the employee will have complete access. The test also makes sure that the verification workflow is functioning properly and is keeping the trust and data integrity throughout the system.

Table 6.8: Test Case for Worker Verification by Admin

Test Case ID	TC-08
Description	To verify that the admin can review and approve worker profiles after successful registration and character certificate verification.
Requirement	The worker must have a completed profile. Admin must have access to the admin dashboard.
Steps to be Taken	1. Log in to the system as an admin. 2. Navigate to the "Pending Worker Verification" section. 3. Select a worker profile awaiting approval. 4. Review the worker's submitted information and ID verification status. 5. Reject or apporve the worker profile.
Expected Result	The system updates the worker's account status to "Verified" upon approval, granting full access to platform features.
Actual Result	The admin successfully verified the worker profile, and the status was updated to "Verified" in the database.
Status	Success
Remarks	This process ensures that only verified and authentic workers can participate in the platform's job.

6.1.14 Test Case 9: Review and Rating Submission

This test is based on allowing employers to post reviews and ratings after finishing a job on the Jobs Bridge site. This is to check whether the reviews are stored, assigned to the relevant job record, and then displayed on user profiles correctly. This can be done to maintain transparency and trust on the platform.

Table 6.9: Test Case for Review and Rating Submission

Test Case ID	TC-09
Description	To verify that registered users can submit reviews and ratings after completing a job and that feedback is stored and displayed correctly.
Requirement	The job must be marked as completed. Employer must be logged in and have valid accounts.
Steps to be Taken	1. Log in to the system as an employer. 2. Navigate to the "Completed Jobs" section. 3. Select the completed job for which feedback is to be submitted. 4. Enter a review comment and select a rating (e.g., 1–5 stars). 5. Submit the feedback form.
Expected Result	The review and rating are successfully stored in the database and displayed on the relevant worker's profile.
Actual Result	The system successfully saved the review and rating, and it appeared correctly on the worker's profile.
Status	Success
Remarks	The review and rating system strengthens accountability and helps users make informed decisions about future engagements.

6.2 Non Functional Testing

6.2.1 Usability Evaluation

Usability testing for Jobs Bridge was conducted by having a representative user (a student acting as a prospective worker or employer) perform tasks associated with major use cases, such as registration, login, profile creation, job posting, applying for jobs, and submitting reviews.Observations, time to complete the task, and subjective ratings regarding ease of use, clarity of interface was recorded. The following are the results of the usability test:

- The dashboard is neat and clean and good for navigation. Easily find jobs, job requests, profiles, and notifications in the separate different portion.
- It was also easy to find the buttons to take action and the input forms and to perform tasks without any confusion.
- Processes like job posting, CNIC upload, and price prediction were easy to find and use and were not cognitively demanding.
- Do confirmation messages, alerts, and error messages guide the user well through the workflows

6.2.2 Performance Evaluation

The Jobs Bridge system was qualitatively analyzed for its speed of execution. Actions like job searches, profile updates, job postings, and job requests were handled quickly, with job listings and alerts refreshed promptly on the dashboard. Translated price prediction functions are responsive. In summary, performance is consistent with the essential non-functional criteria.

6.2.3 Compatibility Testing

Compatibility testing is required to make the system efficiently run on other platforms such as Android, Windows, iOS, etc. As the application is web based, it was important to ensure that employers and workers would be able to apply for jobs, register as a worker, make a price prediction etc, without encountering any technical or display problems.

6.2.4 Security Considerations

Jobs Bridge Security Audit Security verification Authentication, Data Protection, and Control Access. Try wrong users, bypass the login form by accessing job postings, worker profiles, or employer data directly on the URL, and confirm unauthenticated users have no access to these. Password policies like minimum length and complexity are applied, and database passwords are saved on configuration files and not on source code. They ensure that user accounts are protected and platform data is safeguarded against unauthorized access.

6.3 Limitations

- **Limited Job Categories:** There are limited job types and industries on the platform, so users may feel restricted in their options when searching for jobs.
- **Scalability Limitations** Service quality may become poor when a large number of jobs or users participate in the system.
- **Restrictions on Language Support:** The platform is currently available in English and Urdu. Jobs in other languages may not be processed or matched properly.
- **Verification and Fraud Prevention:**Verification for workers and employers is limited to basic checks. There could be inaccurate or fraudulent profiles.

- **Reliance on Device Performance:** The system's smoothness and responsiveness are somewhat correlated with the user's device. Users with older smartphones may experience slow loading or diminished performance.
- **Internet Access Needed:** Real-time updates and notifications require a stable internet connection. Bad connectivity may lead to miss information or the information to be incomplete.

Chapter 7

Conclusions

Overall, the development of the Jobs Bridge platform is a major move to bridge the divide between the employers and the job seeker via an intelligent and user-friendly digital platform. The platform makes effective use of modern Web technologies to facilitate the process of job searching, recruiting, and checking without sacrificing the need for usability and accessibility. With a React-based frontend, an Express.js backend, and a Flask-based machine learning model to predict job prices, this architecture will ensure data fluidity, scalability, and performance. An API in the Urdu language is also added so users from different linguistic backgrounds can interact with the system without any difficulty. The platform's modular design also enables easy synchronization of the modules, along with authentication and data validation systems based on JWT to ensure secure user communication. The interface is clean, and the system replaces horse trading and makes the process transparent and trustworthy through a verification module and a review module. In addition, the stringent performance test and optimization demonstrate the readiness of serving multiple users at the same time and delivering immediate responses along with good-quality data processing of Jobs Bridge.

In all, Jobs Bridge is not just a job site for this; it is an innovative mechanism of using automation and data intelligence and multilingual communication to revolutionize job seeking. It also adds value for the users by making sure that they derive fair employability, convenient and effective recruitment, and smart decisions. This platform lays the groundwork for more innovation models in the digital employment ecosystem, merging convenience, inclusivity, and intelligence in one solution.

7.1 Future Work and Enhancements

- **AI-Powered Job and Candidate Recommendations:**Another possible extension of the Jobs Bridge could be the inclusion of more advanced machine learning techniques for recommending jobs to seekers and candidates to the employer, based on past jobs applied to or jobs posted by the employer, skill sets, and user behavior patterns.
- **Integration with Cloud Infrastructure:** The deployment of the system upon the scalable cloud platforms (e.g., AWS, Azure, or Google clouds) is expected to bring benefits such as enhanced performance and security, an increased number of users, and real time data analyses.
- **Dedicated Mobile Application:**The development of an Android and iOS mobile app will enable users to access the platform on the go, from anywhere, at any time, thus enhancing flexibility and convenience in the job searching and hiring process. Push notifications will inform the user of job matches or employer replies in real time.
- **Chatbot to Real-Time Support:** An AI chatbot can assist users with profile creation, application tracking and responding to questions in real time, improving the overall user experience and support.
- **Real-Time Job Market Analytics:** With the added data visualization dashboard for both the administrator and user, they can view the salary trend and job opening and skill evolution information to help the user and employer pick their best fit upon data.
- **External Job Portal / API Integration:** Jobs Bridge can be connected to well known job portals and APIs, through which Jobs Bridge would be able to grow its database and offer you one place to look for the most comprehensive job listings.
- **Multilingual Expansion:**Besides Urdu and English, catering to more regional and international languages would allow more users of different nations to use the platform.

Appendix A

User Manual

Welcome to Jobs Bridge. This user manual provides a step-by-step guide to help workers and employers effectively use the Jobs Bridge platform. The system is designed to digitally connect skilled workers with employers, enabling secure registration, job posting, worker selection, communication, and feedback. This guide explains account creation, system usage, and job management for all user roles.

A.1 System Requirements

- **Device:** Smartphone, tablet, or desktop computer
- **Operating System:** Android 9.0 or later / iOS / Windows
- **Network:** Stable internet connection (Wi-Fi or mobile data)
- **Browser:** Google Chrome, Mozilla Firefox, or Microsoft Edge

A.2 Getting Started

A.2.1 Create an Account

Open the Jobs Bridge platform and select the Register option. Users are required to enter basic information such as name, email address, role (worker or employer), and password. During registration, a one-time password (OTP) is sent to the registered email address for verification. After successful email-based OTP verification, the account is created.

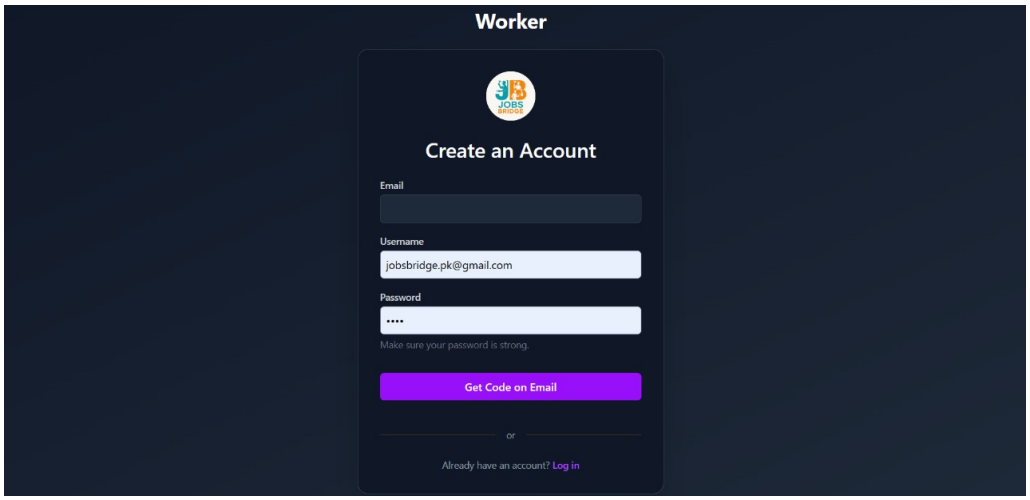

Figure A.1: Registration screen with email-based OTP verification

A.2.2 Log In

After registration, users can log in using their registered email address and password. If incorrect credentials are entered, the system displays an appropriate error message. Options for password recovery are also available.

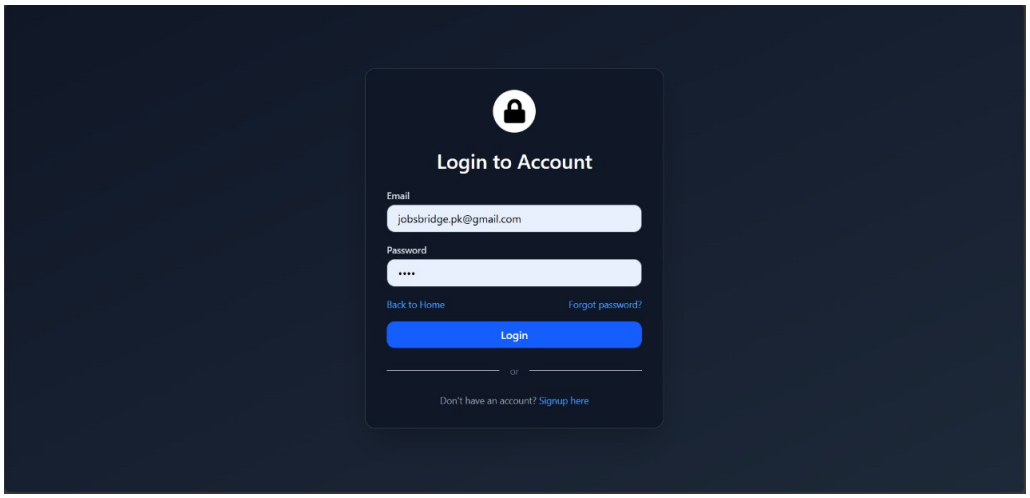

Figure A.2: Login screen for registered users

A.3 Using the Platform

A.3.1 Home / Dashboard

After successful login, users are redirected to the dashboard. The dashboard displays options according to the selected role. Workers can manage their profiles and view available jobs, while employers can post jobs and view available workers.

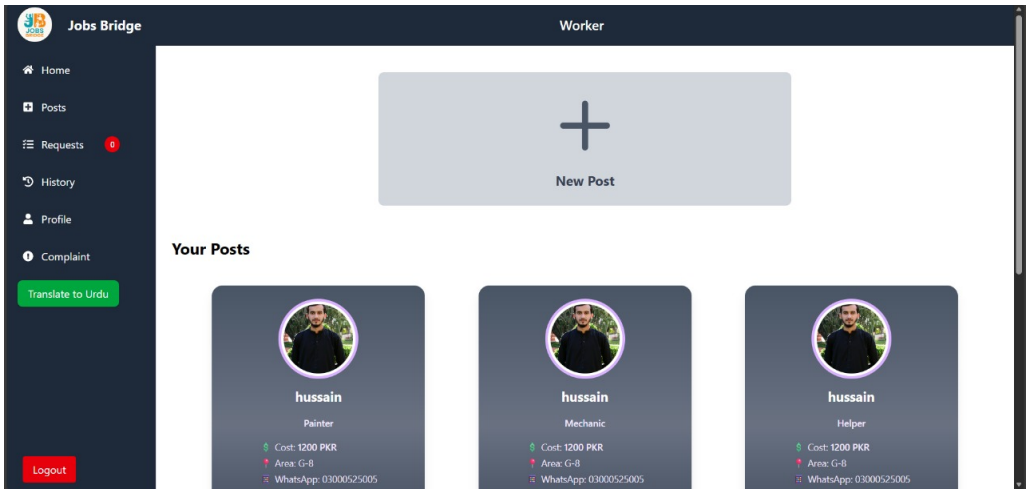

Figure A.3: Dashboard view based on user role

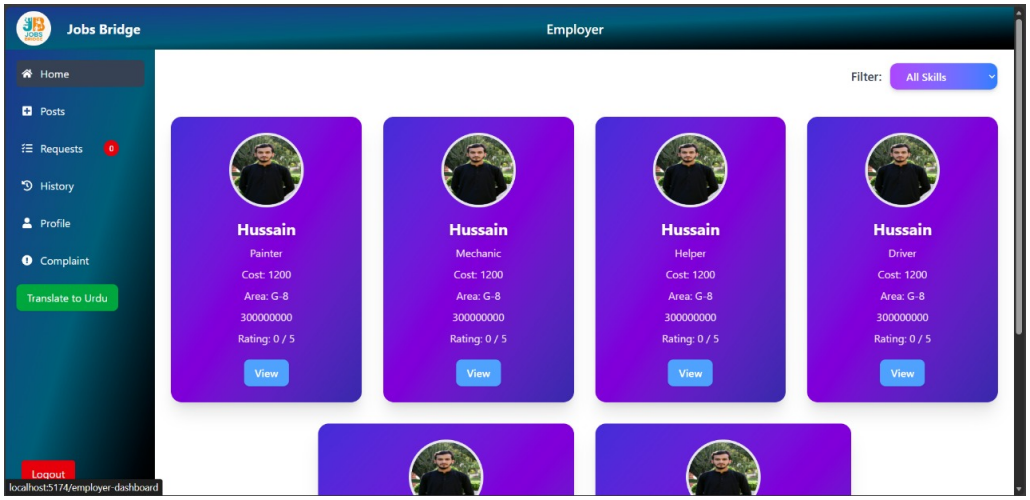

Figure A.4: Dashboard view based on user role

A.3.2 Price Estimation

Jobs Bridge provides a price estimation option on the landing page for both workers and employers. To estimate the service cost, the user selects a service module such as Painting, Plumbing, or Electrician. The user then enters the required service details, after which the system displays an estimated price. This feature helps users understand expected costs before proceeding further.

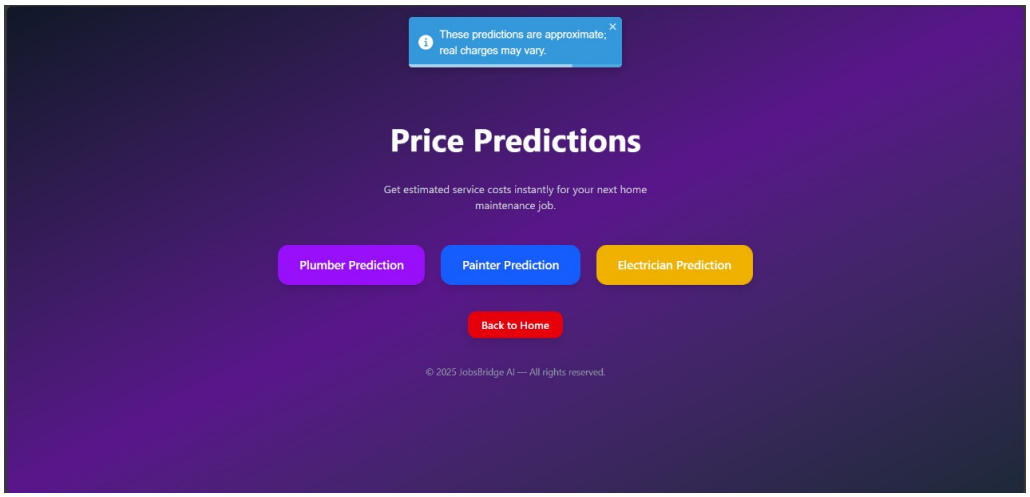

Figure A.5: Price estimation feature showing service selection and cost estimation

A.3.3 Worker Profile

Workers can create or update their profiles by adding skills, experience, and location details. Submitted profiles are verified by the admin.

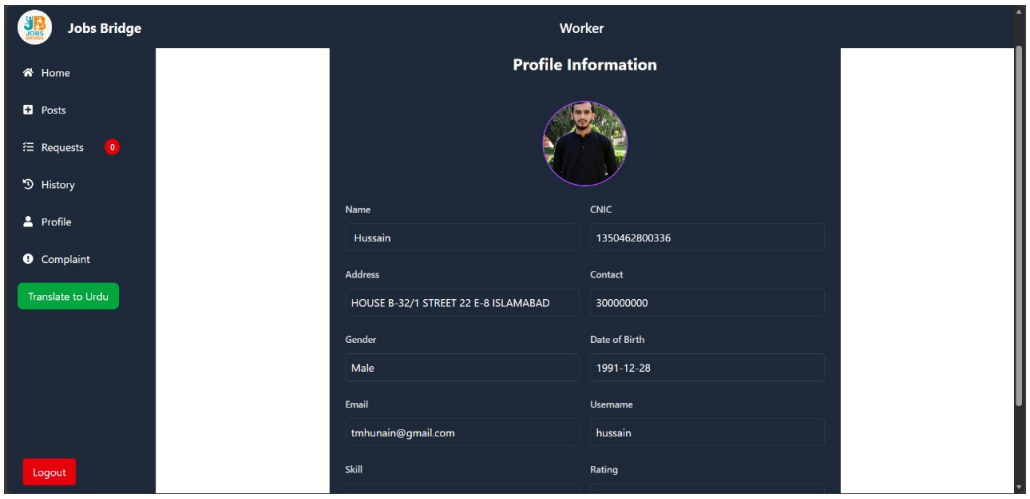

Figure A.6: Worker profile and availability management

A.3.4 Job Posting and Worker Requests

Employers can post job requirements by specifying the job category, location, and expected budget. Once a job is posted, it becomes visible to all verified and available workers on the platform. Interested workers can view job details and submit a job request to the employer through the system. The employer then reviews incoming requests and selects a suitable worker for the job.

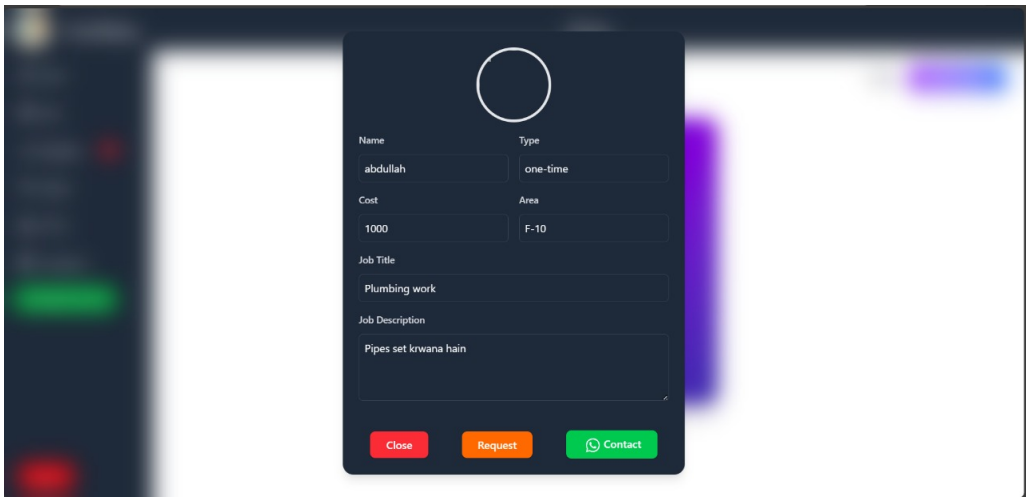

Figure A.7: Job posting selection screen

A.3.5 Job Execution and Communication

Once a job request is accepted, the employer and worker coordinate the execution of the task. Further communication, including discussion of job details, timing, and location, is carried out through WhatsApp using the contact information provided in the system.

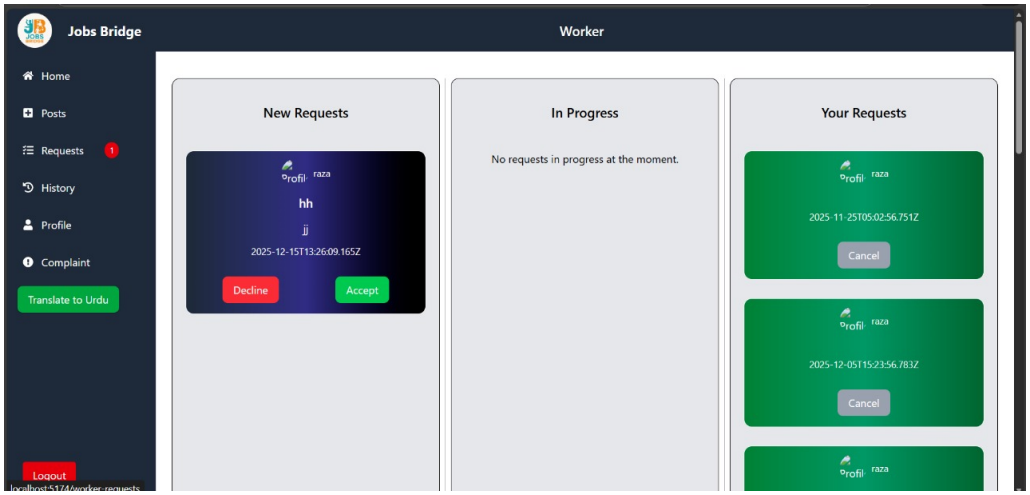

Figure A.8: job communication selection screen

A.3.6 Urdu Language Support

Jobs Bridge allows users to switch the application interface to Urdu from the settings menu. This feature helps users easily understand job details, navigation options, and system messages.

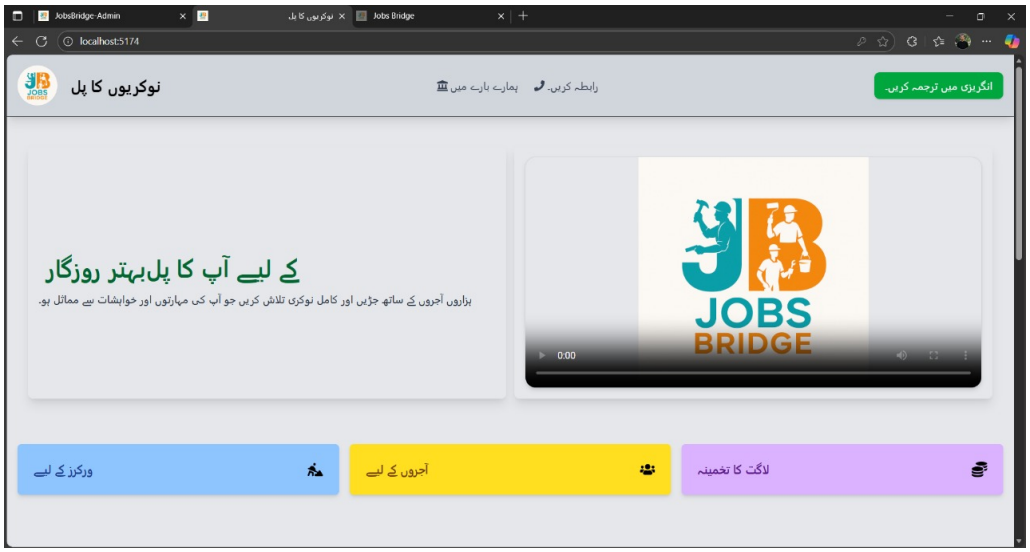

Figure A.9: Urdu language interface of the Jobs Bridge platform

A.3.7 Review, Rating, and Job History

After job completion, employers can submit a review and rating for the worker. The system updates the worker's job history and overall rating, which helps future employers in making informed hiring decisions.

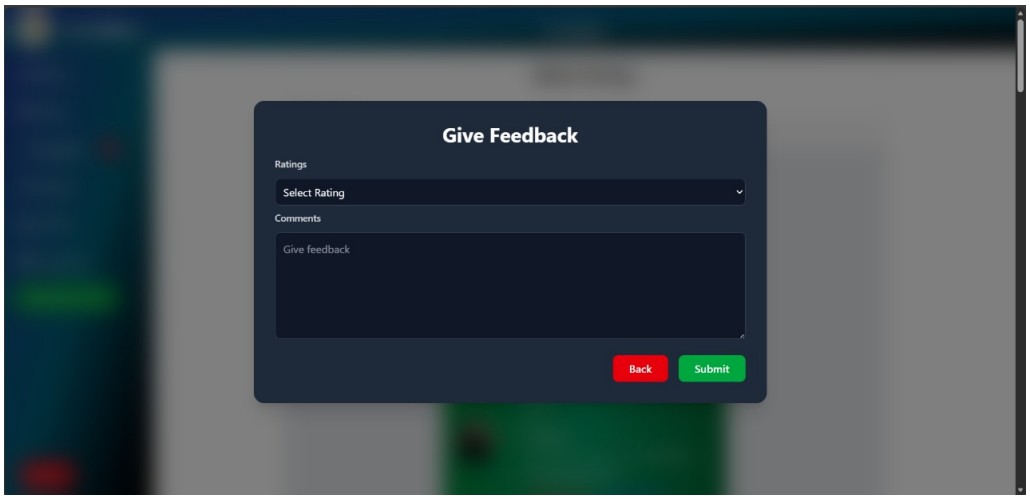

Figure A.10: Review, rating, and job history screen

A.4 Tips and Troubleshooting

- **OTP not received?** Check your email inbox and spam or junk folder.
- **Login issues?** Verify credentials or use the password recovery option.
- **Slow performance?** Use a stable internet connection.

A.5 Security

Jobs Bridge ensures user data privacy through secure authentication mechanisms, including email-based OTP verification and role-based access control.

References

[1] Peter Cappelli. *The Future of the Office: Work from Home, Remote Work, and the Hard Choices We All Face*. Wharton School Press, 2021. Cited on p. 5.

[2] R. Stewart et al. Online and afraid: User privacy concerns on job search platforms. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW2):1–24, 2020. Cited on p. 5.

[3] K. Unsworth and A. Forte. Designing for inclusion: Language and accessibility in global digital platforms. *Journal of the Association for Information Science and Technology*, 72(10):1287–1302, 2021. Cited on p. 5.

[4] A. Chalfin et al. Algorithmic hiring: Assessing the benefits and risks of automation in employment decisions. *Journal of Economic Perspectives*, 35(3):203–230, 2021. Cited on p. 5.

[5] Linkedin job platform, 2023. Cited on p. 6.

[6] Indeed job search, 2023. Cited on p. 7.

[7] Rozee.pk recruitment platform, 2022. Cited on p. 7.

[8] Mustakbil.com job portal, 2022. Cited on p. 7.

[9] Mazdoor app, 2023. Cited on p. 8.

[10] A. Afolabi et al. Web-based human resource sourcing. *Automation in Construction*, 92:1–12, 2018. Cited on p. 8.

[11] Y. Waykar. Role of use case diagram in software development. *International Journal of Management and Economics*, 2015. Cited on p. 14.