



**75**  
YEARS

# COMP [39]900 Term 3 2024

# Project Report

**Project Name:** FlexiLabour - A customer-tradie-worker three sides labouring platform

**Team Name:** CodingMaster

**Team Members:**

Jiaming Guo (z5486270, [z5486270@ad.unsw.edu.au](mailto:z5486270@ad.unsw.edu.au)) - Scrum Master

Zhixun Ji (z5217267, [z5217267@ad.unsw.edu.au](mailto:z5217267@ad.unsw.edu.au)) - Product Owner

Jiangshan Xu (z5461268, [z5461268@ad.unsw.edu.au](mailto:z5461268@ad.unsw.edu.au)) - Backend developer

Hardique Dasore (z5456086, [z5456086@ad.unsw.edu.au](mailto:z5456086@ad.unsw.edu.au)) - Front-end developer

Zhenhua Liu (z5433153, [z5433153@ad.unsw.edu.au](mailto:z5433153@ad.unsw.edu.au)) - Backend developer

Liqi Yi (z5508132, [z5508132@ad.unsw.edu.au](mailto:z5508132@ad.unsw.edu.au)) - Front-end developer

**Project Report Submission Date:** November 18, 2024

## Table of Contents

1. Introduction .....	1
1.1 Overview of FlexiLabour .....	1
1.2 Challenges in the Construction Industry .....	1
1.3 Addressing Challenges with FlexiLabour .....	1
1.4 User Roles and Platform Features .....	1
1.5 Real-Time Communication and Payment Integration .....	2
1.6 Relevance in the Post-COVID-19 Era .....	2
2. Installation Manual .....	2
2.1 Prerequisites .....	2
2.2 Environment Variables .....	3
2.3 Running the Application .....	3
2.4 Accessing the Application .....	3
2.5 Stopping the Application .....	3
2.6 Troubleshooting and Common Issues .....	3
2.7 Environment Variables and Secrets .....	4
2.8 Testing and Verification .....	4
3. System Architecture Diagram .....	4
3.1 Front-End .....	4
3.2 Back-End .....	5
3.3 Persistent Storage .....	5
3.4 Real-Time Features .....	6
3.5 Mapbox Integration .....	6
3.6 Data Flow .....	6
4. Design Justifications .....	6
4.1 Focused Industry Approach .....	7
4.2 Evolution of Messaging System .....	7
4.3 Authentication and Authorization .....	7
4.4 Location Services .....	7
4.5 Payment System .....	7
4.6 Adaptation to Client Requirements .....	8
5. User-Driven Evaluation of Solution .....	8
5.1 Evaluation Approach .....	8
5.2 User Testing and Feedback .....	8
5.3 Customers .....	9
5.4 Contractors .....	14
5.5 Workers .....	16
5.6 Real-Time Interaction and Communication .....	18
5.7 Key Performance Indicators (KPIs) and Survey Results .....	19
5.8 User Suggestion-Based Improvements .....	20
6. Limitations and Future Work .....	20
6.1 Communication Restrictions .....	20
6.2 Review and Rating System .....	20
6.3 Mobile Application .....	20
6.4 Spam and Content Moderation .....	21
6.5 Multilingual Support .....	21
6.6 Future Improvements .....	21
7. Conclusion .....	22
8. Acknowledgements .....	22



## **1. Introduction**

### **1.1 Overview of FlexiLabour**

The FlexiLabour project is a three-sided online platform designed for customers, contractors (tradies), and workers (labourers). Our main issue was to simplify the process of posting jobs and recruiting in the sectors of construction and general labouring with different types of users: customers hiring workers, contractors managing job contracts, and labourers seeking job opportunities. This report covers the design, development, evaluation, and limitations of the FlexiLabour system.

### **1.2 Challenges in the Construction Industry**

The construction industry often faces the problem of how to effectively link the right kind of talent with the right kind of opportunity. Identification of skilled workers in the construction industry is a lengthy, cumbersome process when recruiting is not streamlined. Inefficient platforms are lacking in the industry to support organized connectivity among all stakeholders, resulting in delays, cost overruns, and dissatisfaction among customers, contractors, and workers. Word of mouth and advertisements are now very outdated methods of sourcing labour, which do not apply the modern technological capabilities.

### **1.3 Addressing Challenges with FlexiLabour**

FlexiLabour addresses these challenges with an intuitive digital platform that reduces complexity in hiring and job searching, increasing productivity, therefore improving satisfaction on all sides. It acts as a bridge between customers, contractors, and workers into one ecosystem where all benefit from efficiency provided by digital solutions. The customer is given an easy-to-use interface that helps them post jobs quickly, the contractor can find suitable jobs and recruit workers, and workers can find job opportunities that match their skills and availability.

### **1.4 User Roles and Platform Features**

The platform was designed with ease, accessibility, and efficiency in mind. The process of posting a job has been made to be even easier than it needs to be for customers; this means even those who have very little technical proficiency needn't struggle to create a listing of their requirements. Customers can indicate the type of work required, set their budget, and find a suitable tradie without having to sift through so many complexities.

For contractors, FlexiLabour facilitates a less cumbersome way of accessing various jobs on offer, finding suitable workers to recruit for different projects they may be handling. Quite often, contractors grapple with the problem of finding skilled workers for current and future projects. The platform allows them to review worker profiles, assess their suitability, and make informed recruitment decisions, thus saving time and reducing the risk associated with hiring unverified labour.

FlexiLabour gives laborers an excellent avenue to showcase their skills and mark their availability. Workers must put up with irregularity in finding jobs quite contrary to the nature of fragmented opportunities in the construction sector. The workers can be matched easily with contractors looking for their skillset by maintaining a worker profile and 'availability calendar' that is updated.

### **1.5 Real-Time Communication and Payment Integration**

Thus, the core value proposition that FlexiLabour brings forth to its customers is real-time communication and seamless job management of all parties concerned. Firebase Messaging allows real-time updates to task status; Mapbox provides more location-based details to reach a spot, whereas Stripe is used for wallet-based payment systems. Consecutively, deploying the usage of advanced



**UNSW**  
SYDNEY

1

technology only creates an environment wherein the management of the need for labor can be more reliable and effective in the construction industry.

## 1.6 Relevance in the Post-COVID-19 Era

The role of FlexiLabour is more articulated in recent years, especially with rapid digital transformation accelerated by the COVID-19 pandemic. The pandemic showed how much dependence on digital tools would play a major role in sustaining productivity across industries. In construction, where physical presence is necessary, having a digital platform that streamlines processes such as recruitment and job management can hugely enhance efficiency in operations. FlexiLabour tries to fill this gap by creating a platform that, while matching jobs for all users, makes sure the communication is transparent and the financial transitions secure.

The following sections give a detailed overview of the technical details, design considerations, user-driven evaluations, limitations, and future potential of the FlexiLabour platform. This report will seek to present efforts made in addressing industry-specific challenges while pointing out how FlexiLabour works as a transformative solution within the construction and general laboring sectors.

## 2. Installation Manual

The following describes the procedures necessary for the installation and configuration to deploy and run the FlexiLabour application. Where applicable, the software quality and usability standards will be followed in these instructions.

### 2.1 Prerequisites

- Docker and Docker Compose are installed on your system.
  - You can download Docker from [Docker's official website] (<https://www.docker.com/products/docker-desktop>).
  - Check Docker installation with the following commands `docker --version` and `docker-compose --version` inside terminal.
- The application has been full dockerized, meaning that all the application dependencies and services are packed inside Docker containers to make it easy to deploy.

### 2.2 Environment Variables

- The `\*.env` file provided in the zip folder includes all the environment variables. These include credentials for Firebase, MongoDB, and Stripe, respectively, for authentication, database connection, and payment processing.
- Setup of Environment Variables:
  - Unzip the zip folder containing all the project files.
  - Place the `\*.env` file in the very root of the project directory.
  - The `\*.env` file should not be altered unless there is a need for customization. This file contains sensitive information and is not to be committed to the public repository.

### 2.3 Running the Application

- Step 1: Open a terminal window and change into the root directory of the project where the `docker-compose.yml` file is located.
- Step 2: Run the following command to bring up the app:  
*docker compose up*
  - This will create and start all the services, which in this context, are frontend, backend, and database as described in `docker-compose.yml`.



- Expected output You should see Docker pull the necessary images and, afterward start the containers. It should end with all services, that is, frontend, backend and database up and running.

## 2.4 Accessing the Application

- With all services started and up, fire up your favorite browser and go to:
  - `http://localhost:3000` to access the FlexiLabour frontend application.
  - The backend server runs on PORT 5000 and handles all the API requests. It is located at `http://localhost:5000`, in case it needs to test the APIs.
- By default, using *PORT 3000* for the frontend and *PORT 5000* for the backend keeps the different services separated from each other.

## 2.5 Stopping the Application

- Step 1 Press `Ctrl+C` in the terminal, where you've executed the command `docker compose up` to stop the application and gracefully shut down all services currently running.
- Step 2 Run the one-liner for stopping and removing all containers:

*docker compose down*

- The command stops all services and removes the associated containers, networks, and volumes.

## 2.6 Troubleshooting and Common Issues

- Port Conflict Resolution: At times, port conflicts may arise since a port is in use currently; you can easily modify these ports via composition within the `docker-compose.yml` file.
  - Example: Change `ports: - "3000:3000"` to `ports: - "3001:3000"` to use a different external port.
- Environment Variables: Make sure the `.env` file has been placed in the root directory with valid keys for Firebase, MongoDB, and Stripe.
- Docker Issues: If Docker fails to start make sure Docker Desktop is opened, and also that system resources are assigned, such as memory and CPU.

## 2.7 Environment Variables and Secrets

- Location In the root of the project, according to the explanation, there should be a file named `\*.env`.
- Content Environment variables are:
 

**FIREBASE\_API\_KEY, MONGODB\_URI, STRIPE\_SECRET\_KEY, etc.**
- Security Note: Never push your `\*.env` file into a public repository like GitHub for security against exposure of private credentials.

## 2.8 Testing and Verification

- Frontend Verification: Once deployment is complete, verify the frontend by opening `http://localhost:3000` in your browser and see that the homepage comes up.
- Backend Verification: Verify your backend through sending example API requests via tools like Postman or cURL to `http://localhost:5000`.
- Database Connection: Check if there is a successful connection with the MongoDB instance from the logs of the Docker container or connect using any MongoDB client.



**UNSW**  
SYDNEY

Next, follow these detailed steps to deploy, run, and interact with the FlexiLabour application on your local machine. Instructions will be provided to make sure that the application will be set up correctly, all services running, without any potential issues in advance.

### 3. System Architecture Diagram

The system architecture diagram is shown in Figure 1 below.

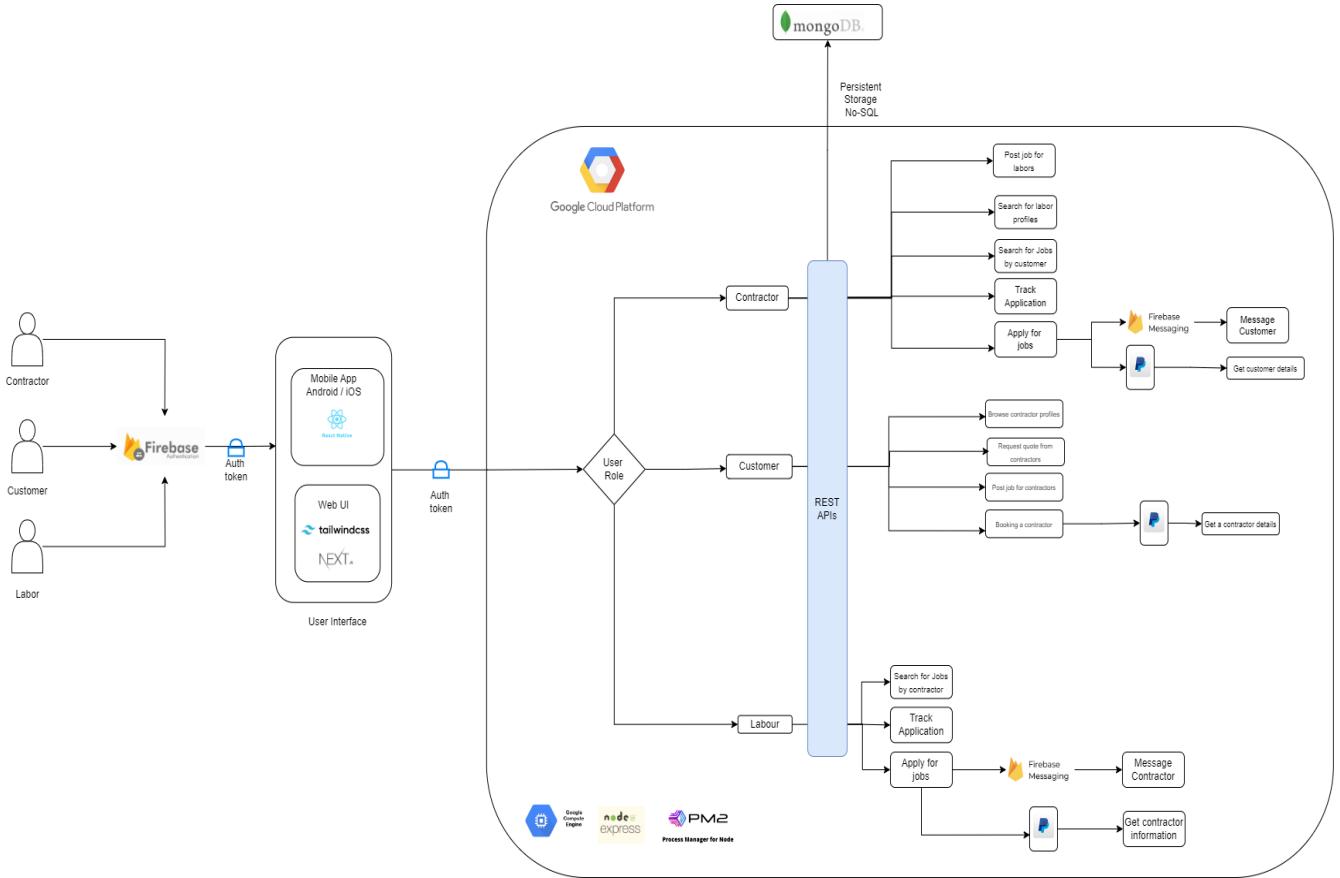


Figure 1. System architecture diagram.

#### 3.1 Front-End

The front-end was designed using Next.js, with styling by Tailwind CSS, hence very intuitive and responsive. By allowing Next.js to support server-side rendering out of the box, this app aims to improve performance, SEO, and user experience for a public-facing platform. Tailwind CSS, because of its utility-first approach, means faster development of consistent, modern UI elements, ensuring that the front-end looks not only good-looking but also easy to navigate.

The front-end is the main interaction point for all users, such as customers and contractors/ workers. Here, the user will log in, post jobs, view postings, apply, manage his profile, and wallet balance. The interface has been optimized for users of low technical knowledge in performing actions with minimal guidance.

#### 3.2 Back-End

Node.js along with Express has been used to create the back-end, pretty apt for the RESTful API requests powering this application. The GCP has been used to host the back-end services of said scalability and reliability for the platform. The backend will take care of authentication, job posting management, transaction processing, and data exchange between the front and persistence.



Firebase Authentication is used for secure user authentication and authorization, supporting social login systems such as Google Sign-In. All role-based actions are performed on the backend through REST APIs providing different operations, whether one is a customer, contractor, or worker. This ensures that data integrity and security across the platform are appropriately maintained.

Stripe integration by the backend handles the payment processing. Wallet funding and unlocking of messaging features is considered a transaction. Every financial transaction is tracked for complete transparency and absolute audit trail to the users and administrators.

### 3.3 Persistent Storage

MongoDB acts as the primary database and store very important information regarding user profiles, job postings, transactions, or communication records. MongoDB has been chosen because of its flexibility, which goes well with the diverse range of requirements in the data of the platform. Data would be categorized into three collections, namely, users, jobs, and transactions, which are modelled by an Object Data Modeling (ODM) library known as Mongoose, providing a schema-based solution to application data.

The users' collection stores information such as roles, personal information, and wallet balances. The jobs' collection contains all details about the jobs, including descriptions, budgets, locations, among others, while the transactions' collection records all monetary interactions, ensuring a full history of user activities and financial interactions.

Figure 2 below is a database diagram overview of the data structure:

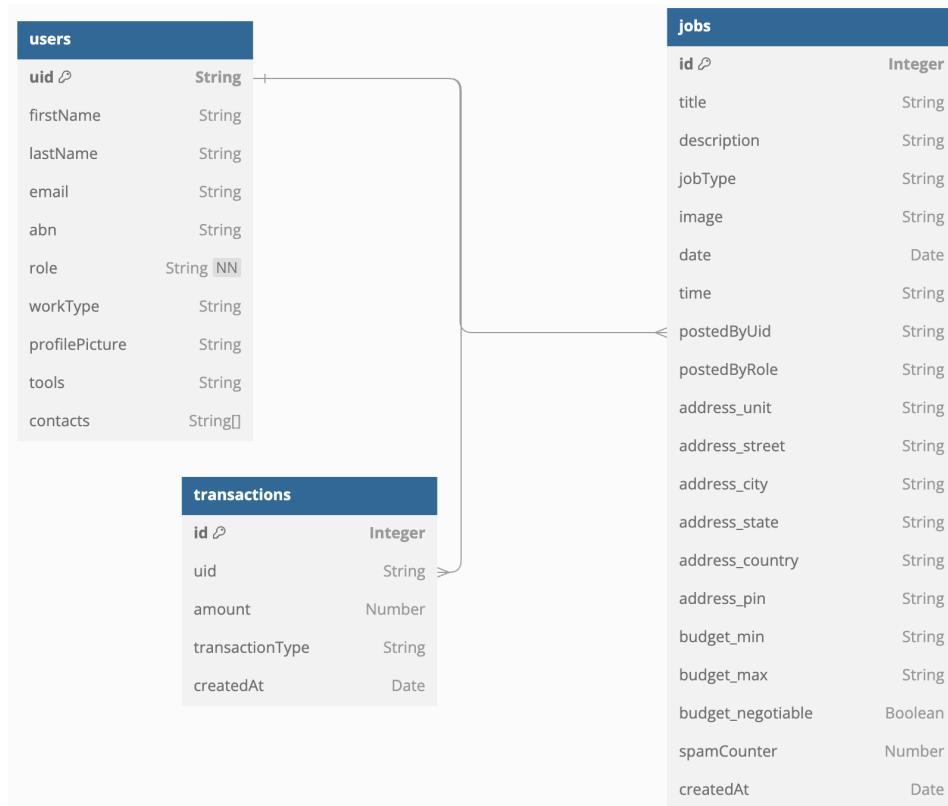


Figure 2. Database diagram of the data structure.

### 3.4 Real-Time Features

Real-time communication is one of the main features of FlexiLabour, which allows smooth interaction between the contractors and workers. Firebase Messaging will be implemented for real-time



notifications where users instantly notify others about their job postings, applications, or messages. This will help a lot in making the hiring process more efficient since there won't be much delay in communication.

### 3.5 Mapbox Integration

It uses Mapbox integration for location-based services, enabling users to set locations for jobs with precision. This feature is important in the construction industry, where clarity of job sites should be defined for effective planning and logistics. Users will have an easy time selecting locations through an intuitive map interface, hence making it easier for both contractors and workers to access the jobs for feasibility based on proximity to each other.

### 3.6 Data Flow

Exchange of data in FlexiLabour happens both on the front-end, the back-end, and persistent storage. The front-end does an action for sending up a request via the REST API to the back-end through its interaction with a user, such as posting a job or applying for one. The back-end then acts upon this request by doing validation of input, interacting with MongoDB, and sending the proper response back to the front-end.

In every sensitive data user action, it checks the identity of the user with Firebase Authentication tokens. This will ensure that only authorized users can view, let alone modify, certain data. When a customer posts a job, it stores the details of the job in MongoDB's jobs collection and then sends notifications via Firebase Messaging to potential contractors. On the other hand, the transaction between the hiring of contractor-workers would also be stored under the same collection of transactions while sending notification emails to relevant parties.

After discussions with the client of the project multiple times, it was decided not to develop a mobile application version due to time and development cost constraints, so that all focus would be centered on a web application. This decision allowed the project team to develop a fully functional and optimized web experience that would serve all users effectively.

The system architecture tries to grant scalability, security, and ease of use to all stakeholders. With modern frameworks and cloud services, FlexiLabour can scale up into a higher number of users by maintaining high performance and reliability. Firebase, Mapbox, and Stripe add great value to the user experience: real-time interaction, location management, and secure payment processing.

## 4. Design Justifications

The project, initially designed to present a job marketplace to all kinds of tradespeople. At that point in time, however, our client said to focus exclusively on the construction and general labouring industries. Given that decision, big changes in the design were welcomed, enabling us to simplify the features of the platform to be more specific for construction-related positions.

### 4.1 Focused Industry Approach

In narrowing our application scope to include only construction and general labouring, we were really focusing on features most relevant and beneficial for these particular industries. Rather than trying to support a wide array of trades with a broad set of functionalities, this system is optimized for location-based job posting, rapid worker recruitment, and effective project management. This helped weed out unnecessary features into a much more cohesive and focused user experience.

### 4.2 Evolution of Messaging System



Every engagement through the message, initially, had to be paid for separately. About this, usability was questioned. The approach was not user-friendly as it broke the workflow and introduced friction in the communication. We, therefore, changed this to a wallet-based payment system in which the user can preload their credits. This reduced transactional overhead since the user could pay upfront in bulk and, therefore, simplify subsequent interactions easily. The wallet system also improved the user experience because of better communication and provided predictable cash flow to the platform.

#### 4.3 Authentication and Authorization

Firebase Authentication has been selected for development to manage user authentication and authorization. It was chosen based on grounds of scalability, ease of integration, and capability to support various authentication methods such as email/password and Google Sign-In. Firebase allowed us to treat user credentials with care using a very secure mechanism that was important to the integrity of the platform, especially sensitive operations around job posting and payment. The integration of Firebase has also enabled us to implement role-based access efficiently, making sure that customers, contractors, and workers have the right levels of access to features and data.

#### 4.4 Location Services

We utilized Mapbox to enhance the user experience when specifying the location of a job. Full-featured APIs from Mapbox support both geocoding and visualizing a map to provide a strong location selection. The user can type in an address or choose from a map to improve the accuracy and also help contractors or workers assess the feasibility of a job site. Mapbox's choice was dictated by ease of integration and flexibility. It needed the capability to provide an intuitive interface for location-based job management-a very critical function in the construction and general labouring industries.

#### 4.5 Payment System

Stripe payment processing was used because of the developer friendliness of its APIs, robust security, and pervasiveness. Stripe implemented, which gave us wallet-based transactions-secure-in nature, whereby credits can be loaded inside for spending on various services such as posting jobs and accessing messaging features. The decision to use a wallet system instead of making payments each time helped reduce friction and improved the overall usability of the platform. This approach also meant that the load on the payment system was reduced in that several smaller transactions were combined into a single process, which made it easier on the user.

#### 4.6 Adaptation to Client Requirements

The development process required us to adapt to the changing requirements of the client continuously. While initially the scope was broad, narrowing it down to the construction and general laboring sectors allowed us to develop a more specific solution. Every sprint included reassessment of priorities through feedback; therefore, the design was improved iteratively. For instance, discarding the plan to develop a mobile app allowed us to assign extra resources to the web platform, ensuring that it was stable and robust. This flexibility was very important in order to produce a final result that met not only the expectations from the client but also the users' needs. The system was initially proposed as a job marketplace for all tradespeople, from carpenters and electricians to hairdressers. Later on, our client insisted on the construction and general labouring industries, leading to some high-impact design changes. This narrowed the focus down to a more specific target market, which cut out unnecessary features in the platform targeted at the construction-related roles. The changing requirements brought along several changes to the messaging system. First, each interaction of a message was to incur a separate payment; that rapidly iterated into a wallet-based system when usability concerns were raised to reduce friction. This was done by allowing users to load their wallet and make upfront payments,



hence streamlining the subsequent transactions in improving user experience. This change also reduced the transactional load on the payment system and provided a more predictable cash flow.

**Authentication and Authorization:** Firebase Authentication was chosen due to scalability issues, ease of integration, and support for multiple methods, such as email/password, Google sign-in, among others. Considering the access that needed to be securely provided to user data and also payment functionality, Firebase Authentication appeared robust, wherein the credentials of users were managed with safety.

**Location Services:** We introduced Mapbox for a seamless location selection experience in job postings. Locations can be specified either by typing addresses or by picking from an interactive map. The reason for implementing Mapbox is its comprehensive API, ease of integration, and support for both geocoding and map visualization.

Stripe was leveraged for payment processing due to the fact that it's an API-friendly service which boasts strong security and prevalence. Stripe integration enables users to load wallets securely and make payments for various services such as posting of jobs and messaging. The wallet system was made with a key focus on reducing friction in end-user interaction, and furthermore enhancing the general usability of the platform.

## 5. User-Driven Evaluation of Solution

### 5.1 Evaluation Approach

This is evaluated from the perspective of satisfying the primary needs for each identified user role: customer, contractor, and worker. Multiple user testing sessions, qualitative feedback through surveys, and key performance indicators related to ease of posting a job, applying for work, and hiring workers were used to complete the evaluation of the FlexiLabour platform. Representative users included potential customers, contractors, and workers who participated in the evaluation, thus providing critical insights for iterative improvement of the platform.

### 5.2 User Testing and Feedback

In user testing, candidates used the website while we observed them and recorded any points of friction. We segmented users into three main profiles: Customers, Contractors, and Workers, and evaluated the extent the website served the needs of each user type. The following are comprehensive reviews of the functionality of the website and the feedback provided by users.

### 5.3 Customers

Customers can sign up with the website, create and post jobs, and contact contractors. The enhanced location auto-complete powered by Mapbox made the process of posting a job easy. On the customer side, the onboarding was simple, and it was easy for them to specify details of their projects. They also found this intuitive and seamless as they switched between creating job postings, managing projects, and communicating with contractors. Below is the interface for customer registration, focused on simplicity and clarity of instructions.



**UNSW**  
SYDNEY

## Connecting You with Trusted Hands for Every Task

Easily connect with skilled professionals for your home and business needs. From electricians to gardeners, we've got you covered.

[Browse Jobs for Contractor](#)[Browse Jobs for Workers](#)

### Get a helping hand

Whether you need help with repairs, maintenance, or special projects, our platform makes hiring the right professional simple and stress-free. Post your job, set your budget, and let the experts come to you. We connect customers with a network of trusted contractors, while contractors can hire workers, too, for projects that need an extra hand.



#### Reliable Workers

We provide you with experienced workers you can trust.



#### Affordable Rates

Our rates are competitive and tailored to your needs.



#### Quality Service

Every worker is vetted to ensure they deliver the best service.

## We pride ourselves on connecting you with trusted professionals



#### Find a Worker

Easily find skilled workers for your job needs.

#### Hire a Contractor

Hire certified contractors to handle your projects.

#### Look for a Job

As a worker or contractor, find the perfect job that matches your skills.

#### FlexiLabour

Find Trusted Helping Hands for Any Job, Big or Small



#### Quick Links

[Search a Job](#)  
[Categories](#)  
[How it works](#)

#### Featured Categories

[Painting](#)  
[Gardening](#)  
[Construction](#)

#### Contact Us

[john@doe.com](mailto:john@doe.com)  
0123456789  
213, Ryde, Sydney, NSW

© 2024 FlexiLabour

Figure 3. Main interface 1.



**UNSW**  
SYDNEY

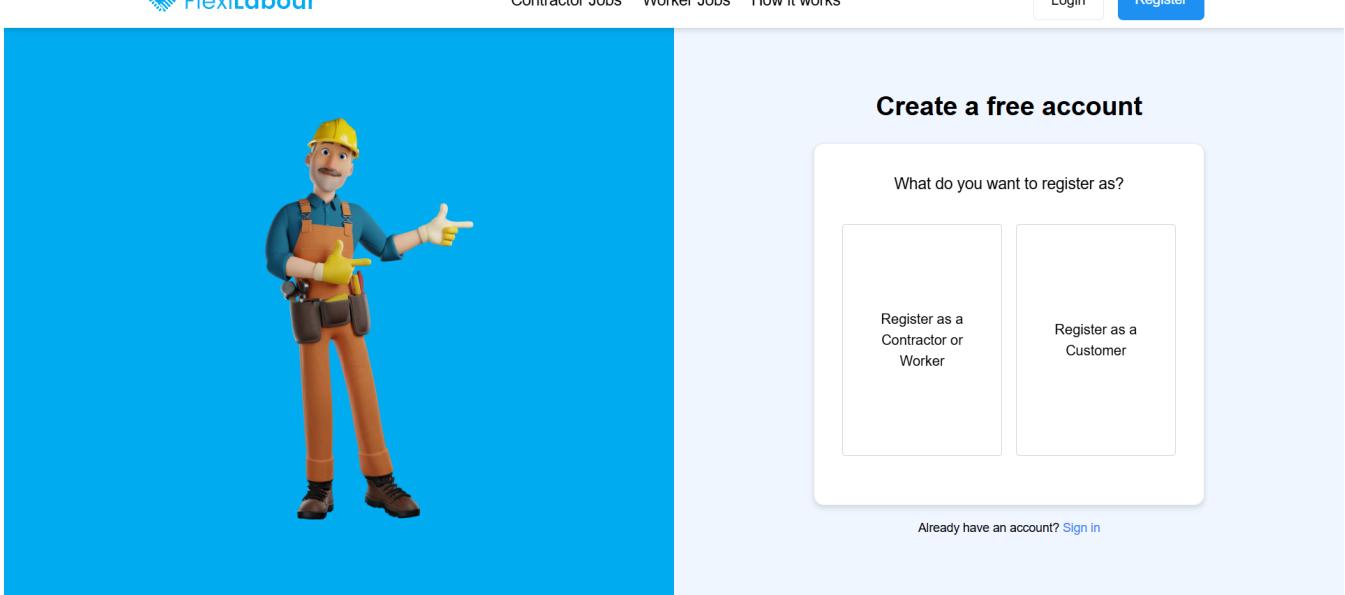


Figure 4. Create a free account.

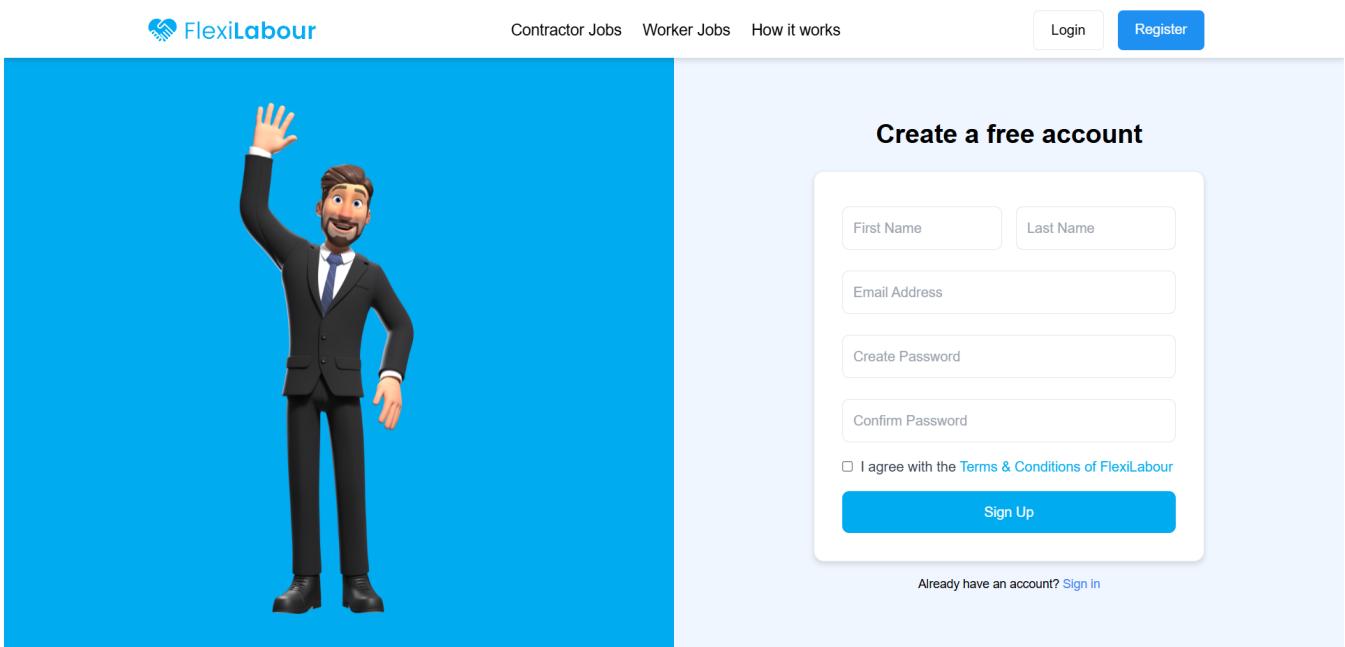


Figure 5. Customer creates a free account.

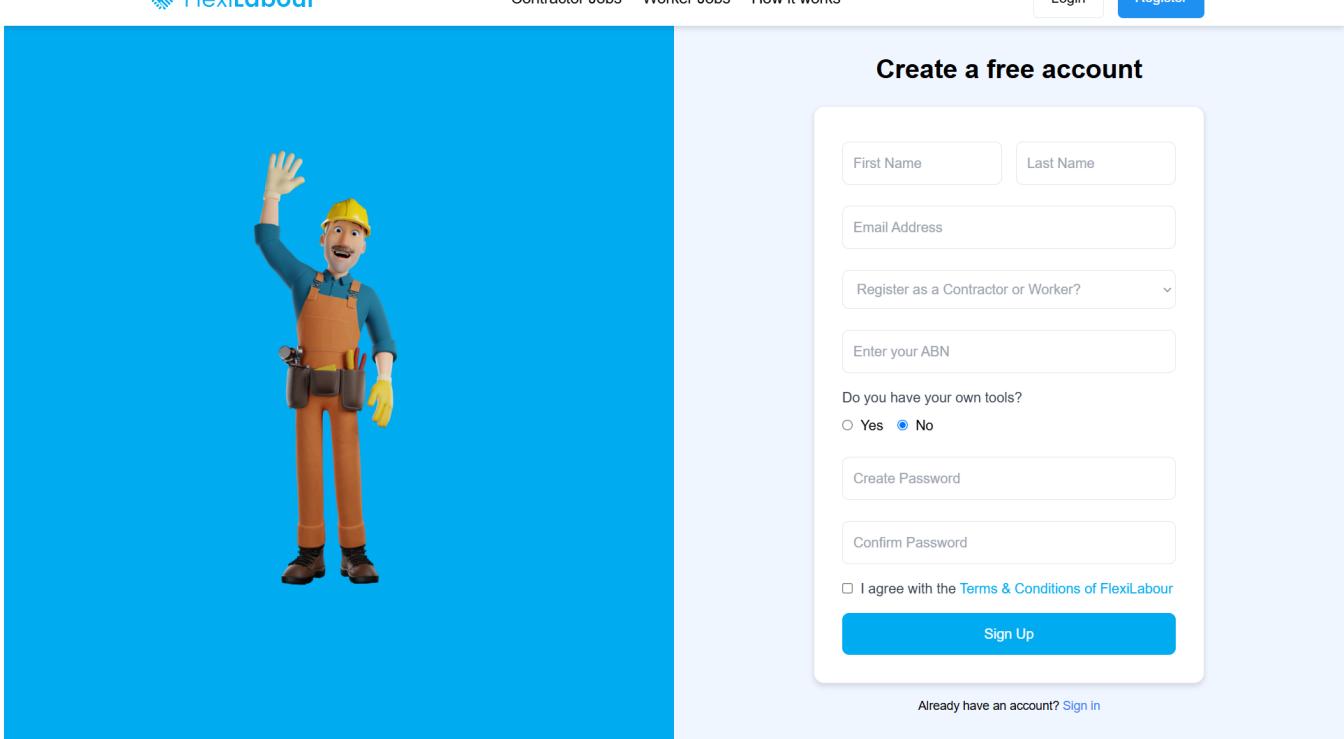


Figure 6. Contractor/worker creates a free account.

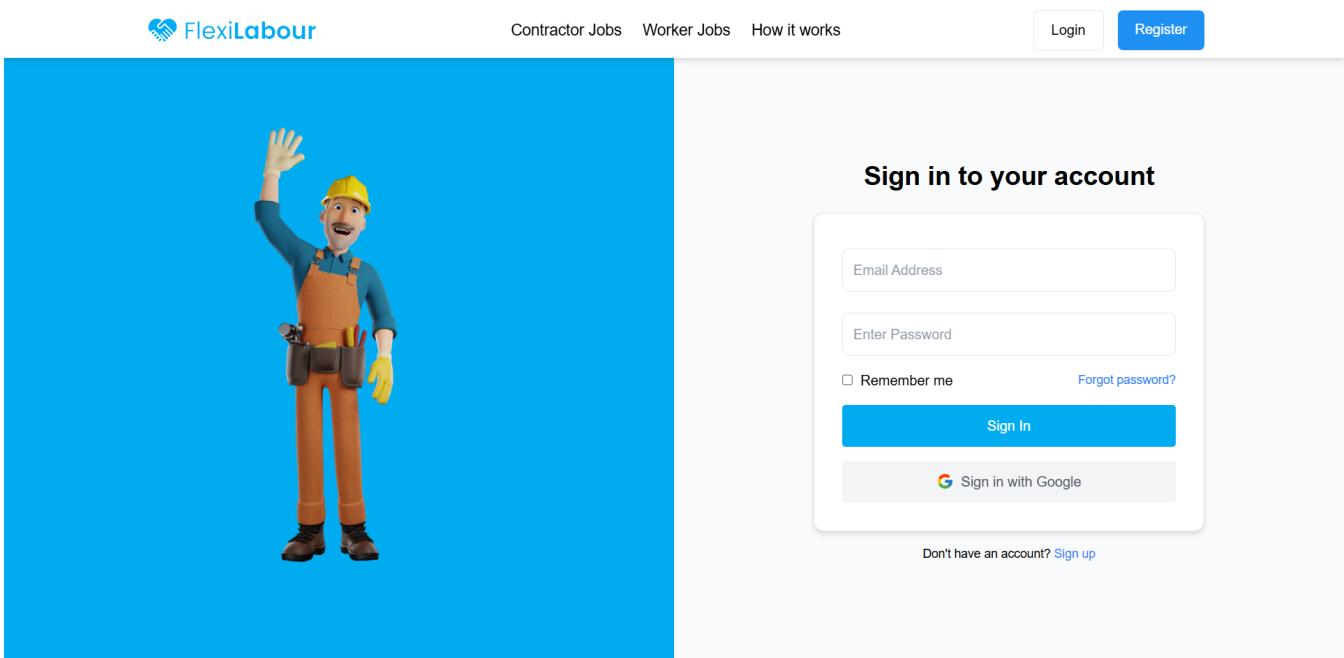


Figure 7. Sign in.



### Profile



First Name  
Hardique

Last Name  
Dasore

Email  
test4@demo.com

ABN  
123456789

[Edit Profile](#)

Figure 8. User profile interface.



Dashboard

Transaction Logs

Posted Jobs

### Dashboard

Welcome to FlexiLabour,  
**Hardique Dasore**

Active Jobs  
**1**

Jobs Posted  
**6**

Post a Job

Wallet Balance  
**\$620**

Update Balance

Figure 9. Dashboard interface.

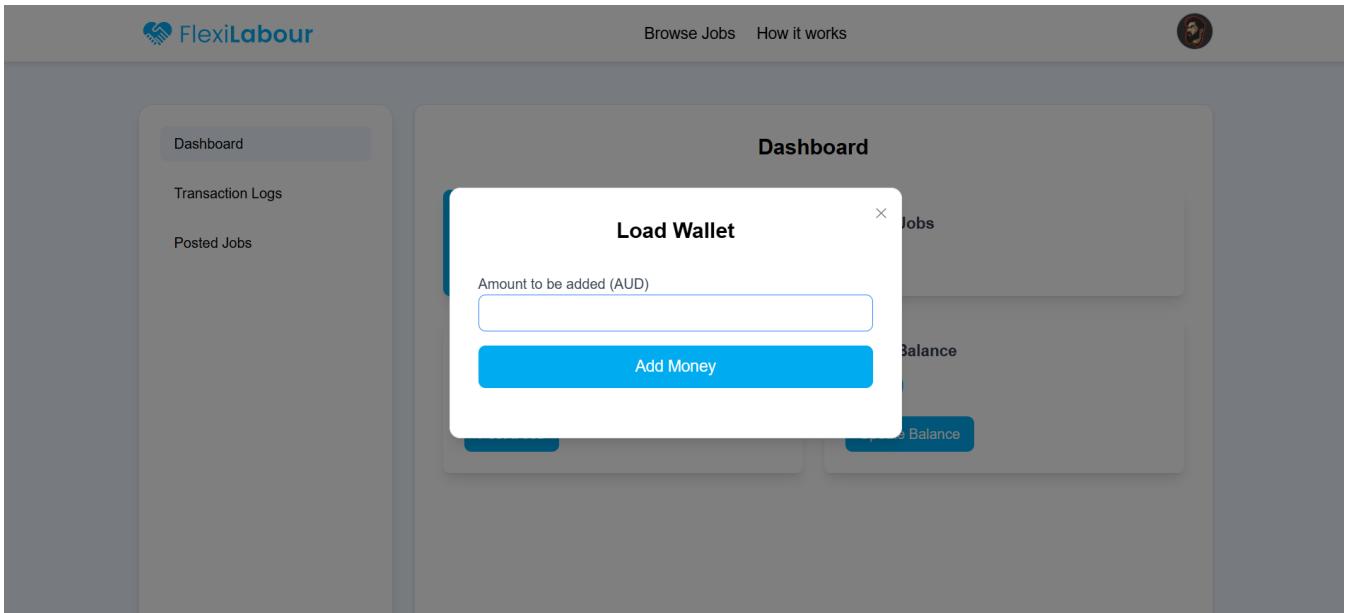


Figure 10. Load wallet - Add account interface.

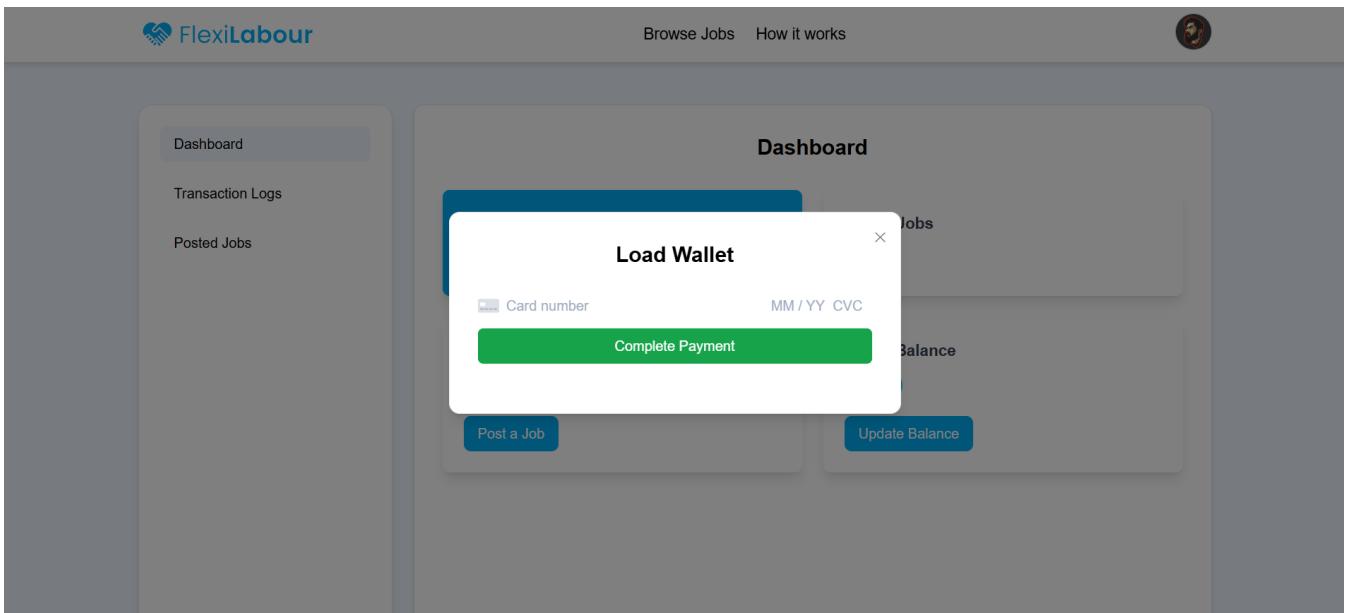


Figure 11. Load wallet - Add payment interface.



- [Dashboard](#)
- [Transaction Logs](#)
- [Posted Jobs](#)

### Transaction Logs

Date	Transaction Amount	Transaction Type
18-11-2024	100	CREDIT
17-11-2024	25	CREDIT
14-11-2024	5	DEBIT
13-11-2024	100	CREDIT
13-11-2024	15	CREDIT
12-11-2024	5	DEBIT
12-11-2024	100	CREDIT
12-11-2024	5	DEBIT
12-11-2024	5	CREDIT
12-11-2024	100	CREDIT
12-11-2024	50	CREDIT
12-11-2024	25	CREDIT
12-11-2024	25	CREDIT
12-11-2024	50	CREDIT

Figure 12. Transaction interface.

- [Dashboard](#)
- [Transaction Logs](#)
- [Posted Jobs](#)

### Jobs Posted

Title	Location	Rate	Job Start Date	Action
Test	Sydney	100	2024-11-15	
Home Baker Required	Sydney	500	2024-11-11	
Painting a house	Sydney	1000	2024-11-14	
Chef Required	Sydney	1000	2024-11-13	
Barista Required	Sydney	1000	2024-11-21	
Excellent Barista for a cafe	Sydney	1000	2024-11-16	

Figure 13. Job posted record list interface.

#### 5.4 Contractors

Contractors can view job postings, deposit credits via Stripe, and spend those credits on applying to jobs or messaging access. They can also hire workers by viewing profiles of workers, which is a big part of contractors' needs in finding and building a team for a project. The feedback contributed by contractors about preloading credits into a wallet and using it on various tasks such as unlocking of job details,



messaging, etc., helped a lot. There's no more repetitive work in payment processes because they would now be free to develop the management of projects instead of managing payments all the time.

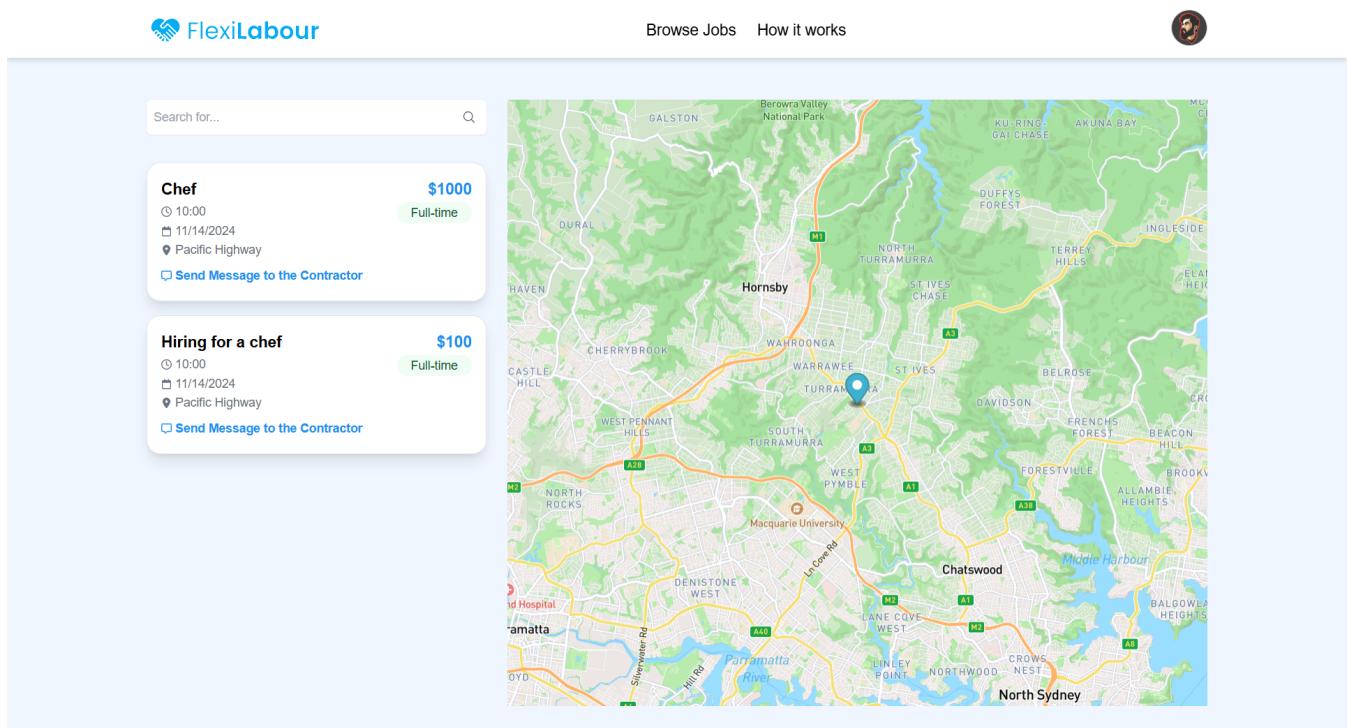


Figure 14. Mapbox interface.

The screenshot shows a job list interface with a search bar at the top left. Below the search bar are two job listing cards, identical to those in Figure 14. To the right of the cards is a detailed view of a 'Chef' job listing. The listing includes a profile picture of a chef, the title 'Chef', the posting date 'Posted on: 11/14/2024', the posting time '10:00', the location '294-296, Pacific Highway, Sydney, New South Wales, Australia - 2065', and a description 'Chef'. Below the description is a large blue button labeled '\$1000'. To the right of the button are two smaller buttons: 'Message' and 'Report as spam'.

Figure 15. Job list interface.

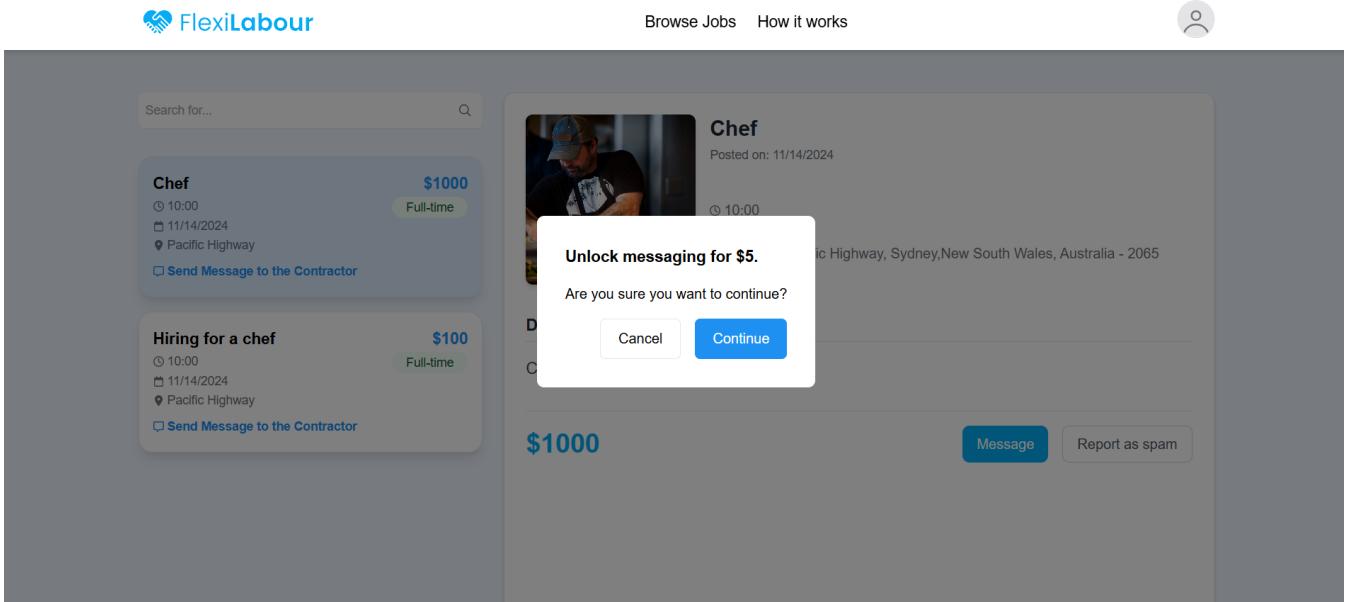


Figure 16. Unlock the contact interface.

## 5.5 Workers

The workers will be able to apply for the jobs posted by contractors, keep up their personal profiles that include their skills, availability, and contact information.

Workers especially liked the availability calendar feature, which allowed them to plan their time and never get job offers at times when they were not available. This feature reduces potential conflicts and increases working opportunities. The real-time messaging feature of this platform lets workers communicate with contractors directly, explain and understand the requirements of a job, and set expectations.

Figure 17. Job post - Title and description interface.



**Post a job**

- ① Title and Description
- ② **Date and Time**
- ③ Location
- ④ Budget

### Date and Time

Select Date
Select Time

X
↻

November 2024						
MON	TUE	WED	THU	FRI	SAT	SUN
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
<b>18</b>	<b>19</b>	<b>20</b>	<b>21</b>	<b>22</b>	<b>23</b>	<b>24</b>
25	26	27	28	29	<b>30</b>	1

Next

Figure 18. Job post - Data and time interface.

**Post a job**

- ① Title and Description
- ② Date and Time
- ③ **Location**
- ④ Budget

### Location

Unit / Apartment Number

Street

Unit 2, 294-296, Pacific Highway, Greenwich

294-296 Pacific Highway, Greenwich New South Wales 2065, Australia

294-296 Pacific Highway, Crows Nest New South Wales 2065, Australia

294-296 Pacific Highway, St Leonards New

City

State

Country

PIN Code

Next

Figure 19. Job post - Location interface.



Browse Jobs How it works

**Post a job**

- ① Title and Description
- ② Date and Time
- ③ Location
- ④ Budget

**Budget**

Minimum Budget

Maximum Budget

Negotiable Budget

**Submit**

Figure 20. Job post - Budget interface.



Browse Jobs How it works

**Post a job costs \$5.**

Are you sure you want to continue?

**Cancel** **Continue**

FlexiLabour

Find Trusted Helping Hands for Any Job, Big or Small

Search a Job Categories How it works

Painting Gardening Construction

Contact Us

john@doe.com 0123456789 213, Ryde, Sydney, NSW

© 2024 FlexiLabour

Figure 21. Job post - Payment interface.

## 5.6 Real-Time Interaction and Communication

Real-time interaction across the different user roles was assessed using Firebase Messaging, which drives instant notifications. Customers, contractors, and workers alike love to get timely updates on posting their job, its applications, and communications. However, one big point of improvement related to it: cost. Many users mentioned, and most of them workers and small contractors, that the initial payment required to unlock messaging made them prefer to not commit fully to the platform without first realizing the value of its proposition.



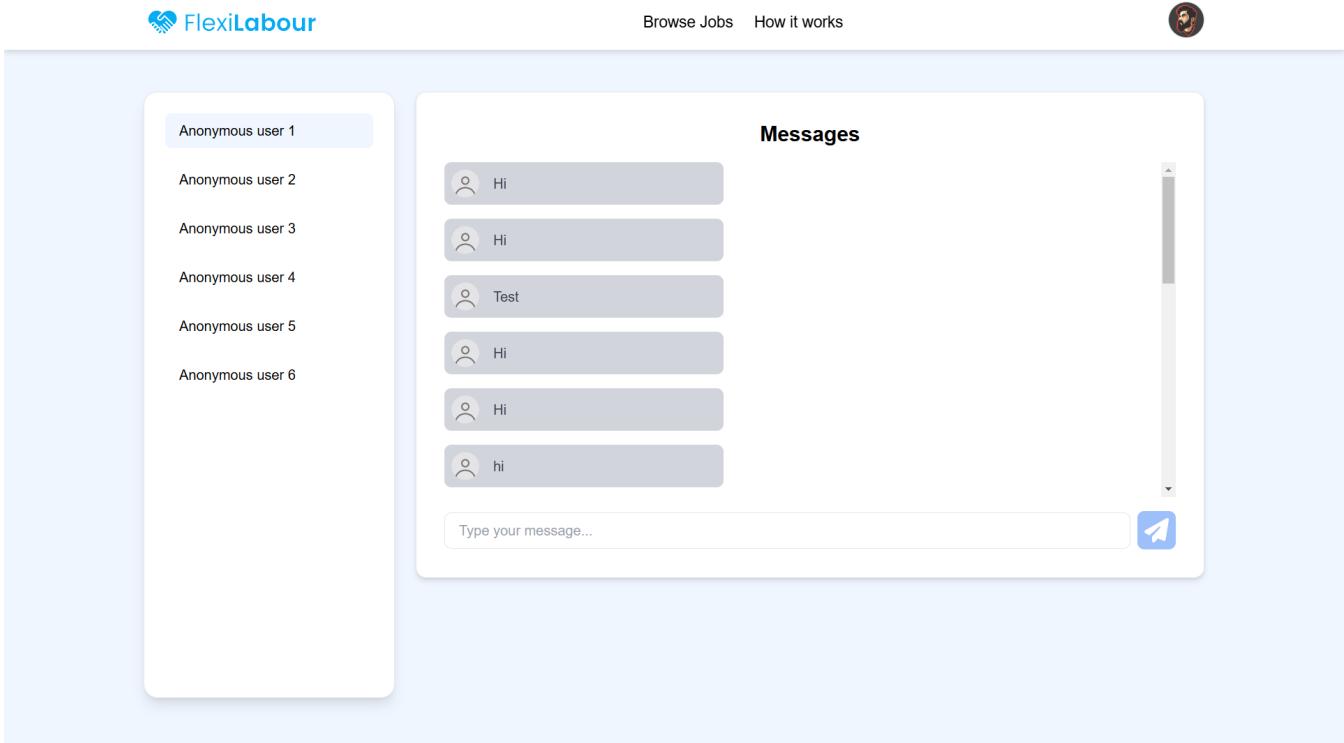


Figure 22. Real-Time Interaction and Communication interface.

## 5.7 Key Performance Indicators (KPIs) and Survey Results

To evaluate the performance of the FlexiLabour platform, we measured several key performance indicators:

- **Onboarding Time:** The average time taken by users to register and set up their profiles was around 5 minutes, which met our usability expectations.
- **Job Posting Completion Rate:** More than 80% of customers successfully completed their job postings without seeking external help, indicating that the interface was user-friendly.
- **Response Time to Job Applications:** Onboard Firebase Messaging had significantly improved the response time, averaging 15 minutes for contractors to respond to job postings.
- **User Satisfaction:** In post-session surveys in the user testing session, 85% felt satisfied with the ease of use and functionality of the platform, whereas 15% showed room for improvement, mainly in messaging costs and no review feature.

## 5.8 User Suggestion-based Improvements

Several improvements were made considering users' suggestions to better the user experience:

- **Lower Messaging Barriers:** Users will have a certain number of free messages so that the barrier to messaging is lower for new users who might want to get familiar with the site.
- **Profile Verification:** Users asked for verification to inspire more trust in the profiles of workers and contractors. Identity verification features in the plans shall help users overcome this worry.
- **Incorporating Reviews and Ratings:** One such feature incorporated was the review and rating system. They wanted to go through the reviews about workers and contractors before making a decision. This is in the pipeline for future updates, and design-related work has already begun.

The assessment pointed out not only strengths but also areas for improvement. The users told that this platform is path-breaking for construction and general laboring sectors, but taking care of their concerns will further enhance the usabilities and reliability of this platform.



## **6. Limitations and Future Work**

Limitations of the platform now are that the options to communicate are restricted, there is no review system available for workers, and mobile support is limited. Partially, this was because of time constraints whereby the development needed to focus on core functionality.

### **6.1 Communication Restrictions**

Now, messaging between users is restricted to those who have paid for access. This discourages new users from being fully involved in the platform and being those who might seek assurance yet not want to pay without understanding the benefits of the platform. We're working on putting into place a tiered messaging system whereby users will get a set number of messages free, after which they'll be invited to upgrade their service to a premium messaging service. This will allow new users to have a glimpse of the platform's features of communication before making any financial commitments.

### **6.2 Review and Rating System**

The review and rating system lacks the ability of users to rate different contractors or workers based on previous activities. This reduces the potential of users in making proper decisions and reduces overall transparency of the platform. Soon, we are going to integrate a review and rating feature to help users comprehensively rate experiences. This feature will form the basis for creating trust in and credibility of the overall application and as such will increase user satisfaction and engagement.

### **6.3 Mobile Application**

There is no corresponding mobile application on the platform for the application, making it relatively inaccessible to users who would rather have access through their mobile devices. Mobile support is greatly needed due to the nature of the labor marketplace, where users are mostly on the move. Making a mobile app will be one of the key focuses of the subsequent updates in order to make access wider and the user experience even smoother. The mobile version will also include all the key features optimized for ease of use on smaller screens to make workers and contractors manage their tasks without problems on-site.

### **6.4 Spam and Content Moderation**

Currently, job posts and messages go through manual moderation, which is time-consuming and prone to many errors. This will be vastly improved with the use of NLP models that automatically detect spam and inappropriate content on the platform, hence making it reliable. An AI-based content moderation system will enable communications and job postings to be kept professional and at community standards for all users to enjoy the platform.

### **6.5 Multilingual Support**

Currently, it only supports English; this might be a limitation as far as regions that do not speak English are concerned. In the future, multilingual support will be added to increase the usage of the platform and ensure more inclusivity. This update will include translation of the user interface and multilingual customer support for a smooth experience across different linguistic cultures. By expanding language support, FlexiLabour will increase its ability to reach a wider global audience and support the needs of diverse regions.

### **6.6 Future Improvements**

Tiered Messaging System: This shall be achieved through the provision of new users with a limited number of free messages in terms of the tiered messaging system. It shall minimize the initial hesitation for people to start using the tool.



- Full Review System: This includes rating and reviewing, adding trust and therefore transparency to all user roles' decision-making process.
- Mobile Application Development: A fully functional mobile application will be developed targeting the growing number of mobile users, including basic functionality that will be easy to handle in a mobile environment.
- AI-Based Content Moderation: Integration with NLP-based spam and content detection will reduce the need for manual moderation and ensure professionalism in user interactions consistently.
- Multilingual Interface: Having multilingual support will extend the application and make it more usable in non-English-speaking countries for better adoption.

With these enhancements, FlexiLabour will surely be able to provide a more efficient and user-friendly avenue through which one can manage construction and general labour needs. These shall aid in the expansion of the user base, raise engagement, and establish FlexiLabour as a reliable and versatile solution in the industry. Currently, limitations to the platform include restricted communication options, a review system lacking for workers, and limited mobile support. These were due partly to developmental time constraints and a need for prioritizing core functionalities. Messaging Restrictions: Presently, messaging amongst users is currently available to paying customers only. The paywall may prevent new users from fully interacting with the service. We intend to implement a tiered messaging system whereby a limited number of free messages will be made available, after which there would be a premium messaging service.

- Review System: There is no review system for rating different contractors or workers based on performance. The review system builds trust within a platform. In future versions, there is a rating and review feature where the user can share their experience. The feature will contribute to better transparency and more informed decision-making by the users.
- Mobile Application: The site does not have a dedicated mobile application yet, reducing accessibility for users who would have preferred access through their mobile devices. Mobile support is crucial for a labour marketplace since users are usually on the go. In due course of time, a mobile application will be developed to help extend access and enhance user experience.
- Spam and Content Moderation: Manual job posting and message moderation are not only extremely time-consuming but also error-prone. Automation of the moderating task by including NLP models for spam or inappropriate content detection will improve the quality of-and trust in-the platform. Professional and appropriate content for all users will be guaranteed once an AI-based moderation system is in place.
- Multilingual Support: The platform currently supports only English, which may limit its adoption in non-English-speaking regions. Adding multilingual support will help broaden the user base, making the platform accessible to a more diverse audience. This feature will involve translating the user interface and providing multilingual customer support, ensuring a seamless experience for users from different linguistic backgrounds.

## 7. Conclusion

The FlexiLabour platform was well realized in the construction industry by addressing customer, contractor, and employee needs. In this process, the project has undergone many iterations and continuous user feedback, which refined both its feature set but also in general how it feels for users. There are certain limitations, such as restricted messaging and a lack of mobile support, but even these minor points show quite a promising base for further extensions and improvements that could even further extend its capabilities and user satisfaction.



FlexiLabour will be bright, changing how labor is sourced and managed in the construction industry. Those additions, along with further expansion to mobile platforms and support for multiple languages, will make it the go-to solution in many regions for labor needs. Moreover, AI-driven moderation will introduce much in safe and smooth user experience development, as will the enhanced features in the field of payment.

The entire development journey of FlexiLabour underlined how vital adaptability can be in project management, with respect to client feedback and user needs. These experiences will be useful in other projects, especially those with multiple stakeholder platforms where different user roles must interact smoothly. In this light, FlexiLabour is sure to innovate, hence being conducive to demands that are in continuous flux within both the construction and labour sectors.

## Acknowledgements

*Thanks, all professors and the tutor of COMP9900 course Umar Farooq for his guidance and support, and our client Dai Bang for his trust.*

*Thanks, all members of CodingMaster for their cooperation and hard work in this project.*

*Wish you all a bright future!*

*Greatness needs no words!*



**UNSW**  
SYDNEY

22