

Developer's Hub (Job Recommendation System)

A Project Report

Submitted by:

Krunal Savaj (AU1940271)

in partial fulfillment for the award of the degree

of

**BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING
At**



**Ahmedabad
University**

School of Engineering and Applied Sciences (SEAS)

Ahmedabad, Gujarat

May 2023

DECLARATION

I hereby declare that the project entitled “Developer’s Hub (Job Recommendation System)” submitted for the B. Tech. (Computer Science and Engineering) degree is my original work and the project has not formed the basis for the award of any other degree, diploma, fellowship or any other similar titles.

A small rectangular box containing a handwritten signature in blue ink.

Signature of the Student

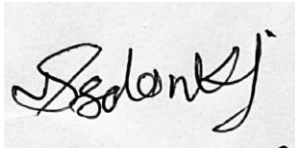
Place: Ahmedabad

Date: 1st May 2023

CERTIFICATE

This is to certify that the project titled “Developer’s Hub (Job Recommendation System)” is the bona fide work carried out by Krunal Savaj, a student of BTech (Computer Science and Engineering) of School of Engineering and Applied Sciences at Ahmedabad University during the academic year 2022-2023, in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (Computer Science and Engineering) and that the project has not formed the basis for the award previously of any other degree, diploma, fellowship or any other similar title.

This project was at **Asite Solutions Pvt. Ltd** under the supervision of the industry mentor **Snehal Solanki**



Signature of the Industry Mentor
Mentor

Place:

Date:

Signature of the Faculty

Place:

Date:

ACKNOWLEDGEMENT

I greatly appreciate Asite Solutions Private Ltd. for giving me the chance to work on such a significant project and for introducing me to so many new tools and technology. I would like to express my sincere gratitude to Ahmedabad University's Department of Computer Science and Engineering for providing me with the skills and values necessary to conduct myself professionally at work.

I sincerely appreciate Mr. Parth Shinojiya and Ms Snehal Solanki's assistance with this effort. Together with a few other colleagues, I too had the pleasure of working with Mr. Dixit Desai.

I would want to express my gratitude to Professor Shefali Naik for her ongoing assistance, inspiration, and advice in making my project successful.

I would like to express my gratitude to my family for their unwavering support throughout this time. Also, I am grateful for the friends and Colleagues who inspired me during my BTech Project and graduation. I also want to thank Stack Overflow and the technology Community for making such a useful service and all the people that worked on the technology.

Krunal Savaj
AU1940271

ABSTRACT

Developer's Hub (Job Recommendation System) platform connects organizations, individuals, Students and freelancers from around the world so that they can complete tasks quickly and affordably. Our platform offers a marketplace of experienced freelancers that are prepared to take on your project, whether you require a graphic designer, a software developer, a content writer, or a virtual assistant. The criteria for a project can be posted by a client, and interested freelancers can submit bids. Before choosing, they can look over the evaluations, work history, and profiles of the freelancers. Our platform provides tools to organize projects, communicate with freelancers, and monitor progress. With our user-friendly interface and vast pool of talented freelancers, we make it easy for clients to get the job done quickly and affordably.

The main components of this internet application tool included ReactJS for JavaScript, NodeJS, ExpressJS, MongoDB, and a few more. With a well-planned installation, these technologies are available to everyone. This Web-App is accessible to everyone who has one of the supported browsers and hasn't already installed any further extensions. We anticipate that the suggested stakeholders will be able to make appropriate use of this platform.

TABLE OF CONTENTS

DECLARATION.....	i
CERTIFICATE.....	ii
ACKNOWLEDGEMENT.....	iii
ABSTRACT.....	iv
TABLE OF CONTENTS	v
LIST OF FIGURES.....	vii
LIST OF TABLES	vii
GANTT CHART.....	viii
CHAPTER 1: INTRODUCTION.....	1
1.1 Problem Definition	1
1.2 Project Overview	1
1.3 Hardware Requirements	3
1.4 Software Requirements	3
CHAPTER 2: LITERATURE SURVEY.....	4
2.1 Background Research.....	4
2.1.1 User Stories	4
2.1.2 Technology Stack For Development.....	5
2.1.3 Existing Systems	6
2.1.4 Unique and Core Features	6
2.1.5 Salient Features	7
2.2 Feasibility Study	7
2.2.1 Technical Feasibility	7
2.2.2 Risk Feasibility	8
CHAPTER 3: METHODOLOGY	9

3.1	Requirements Specification.....	9
3.1.1	Purpose.....	9
3.1.2	Essential Requirements	9
3.1.3	Non-Functional Requirements.....	10
3.2	Diagrams.....	11
3.2.1	Use-case Diagrams	11
3.2.2	Class Diagram	12
3.3	System Design.....	12
3.3.1	System Architecture.....	12
3.4	Development Phase	13
3.4.1	Software Development Model	13
3.4.2	Frontend Implementation	13
3.4.3	Backend Implementation	16
	CHAPTER 4: RESULTS	22
4.1	Screenshots of Outcomes.....	22
4.2	My Contribution to the Project	29
4.3	Learning Outcomes.....	29
4.4	Real World Application.....	30
4.5	Future Works	30
	CHAPTER 5: CONCLUSION.....	32
	CHAPTER 6: REFERENCES.....	33

LIST OF FIGURES

Figure 1 MERN Tech Stack.....	5
Figure 2 Use-Case Diagram.....	11
Figure 3 Class Diagram	12
Figure 4 System Architecture	12
Figure 5 Landing Page	22
Figure 6 Registration Page (Organization)	23
Figure 7 Registration Page (Freelancer)	23
Figure 8 Login Page (Freelancer)	23
Figure 9 Login Page (Organization)	24
Figure 10 Dashboard Freelancer	24
Figure 11 Create and Edit Profile of Freelancer	24
Figure 12 Add Education Details Page.....	25
Figure 13 View All the Freelancer.....	25
Figure 14 View all Jobs	25
Figure 15 Individual Freelancer's Profile.....	26
Figure 16 Education List.....	26
Figure 17 Create and edit Organization's Profile	26
Figure 18 Dashboard For Organization	27
Figure 19 Posting Job.....	27
Figure 20 View All Jobs by individual ORG.....	27
Figure 21 All Details Regarding Posted Job.....	28

LIST OF TABLES

Table 1 Hrdware Requirements	3
Table 2 Software Requirements.....	3
Table 3Authentication URL end points	18
Table 4 Freelancers Profile End Points.....	20
Table 5Organization Profiles End Points.....	21
Table 6 Jobs End Points	21

GANTT CHART



CHAPTER 1: INTRODUCTION

1.1 Problem Definition

Existing platforms struggle to maintain consistent service quality due to their limited-service categories, communication barriers, payment and dispute problems, trust and safety issues, lack of customization, unfavorable pricing structures, restricted geographic reach, and unfavorable pricing structures. The project aims to create a comprehensive freelance platform that addresses these limitations and offers seamless experiences for clients and freelancers, as well as a variety of services, efficient communication, secure payments, trust and safety measures, customization options, reasonable pricing, increased freelancer visibility, and global accessibility.

1.2 Project Overview

This Platform is designed to address the challenges of the freelance economy. The rise of the gig economy has created a need for platforms that can connect freelancers with clients looking for their services. The main problem that seeks to solve is the issue of matching clients with the right freelancers for their projects.

The search for capable and trustworthy freelancers is one of the difficulties clients encounter. Finding the proper freelancer for a job might be difficult because there are so many freelancing markets. Customers might not have the time or resources to sort through numerous applications and profiles, and they might not be aware of how to evaluate the quality of a freelancer's work. Yet, freelancers could have difficulties finding clients that are a good fit for their qualifications and experience. Freelancers frequently face competition from others who might provide comparable services for less money. It could be challenging to stand out and draw customers as a result. The problem of credibility and trust is another topic that is addressed. Since freelancers and clients frequently collaborate online, this might raise questions about both parties' dependability and the calibre of their work. The platform offers trust-building tools like verified credentials, escrow payment systems, and user ratings and reviews. In general, the goal of this platform is to make it simpler for

clients and freelancers to connect, collaborate, and establish enduring connections. The process of identifying and recruiting freelancers is streamlined by offering a platform that links independent contractors with clients and provides tools for communication and collaboration. The platform also offers a certain amount of credibility and trust that can aid in creating long-lasting connections between clients and freelancers.

1.3 Hardware Requirements

The application has minimal hardware requirements because it is web-based. To make sure the program functions well, it is advised to adhere to following guidelines.

Category	Specification,
RAM	8 GB or above
Processor	Intel i5
Storage	128GB or above
I/O Devices	Keyboard, Mouse

Table 1 Hrdware Requirements

1.4 Software Requirements

Tools Used	Purpose	FE/BE/Both
VS Code	Coding	Both
Vercel / Netlify	Hosting	Both
Github	Code Version Control	Deployment, Production
Postman API	Testing	Backend
Figma	Designing	Frontend
MongoDB	Database	Backend
Google Docs	Documentation	Both
<ul style="list-style-type: none"> ● Chrome v64+ ● Firefox v59+ ● Safari v11+ ● IE v11+ ● Edge v50+ 	Browser Requirement	Frontend

Table 2 Software Requirements

CHAPTER 2: LITERATURE SURVEY

2.1 Background Research

2.1.1 User Stories

Freelancers

- Freelancers sign up for the application; they can be self-employed people or students.
- After signed up, they build a profile by supplying the relevant information, such as their skill set, contact information, portfolio, social media profiles, etc.
- Businesses that have registered with the app and generated profiles can view them after they have been created.
- When necessary, freelancers can update the information on their profiles. This will enable them to add new skills as they pick them up.
- The app allows freelancers to examine the profiles of other freelancers.
- Freelancers can contact with businesses.
- Freelancers can browse the jobs that businesses have advertised and show their interest if they would rather work on the job.

Business

- The application allows for business registration.
- After registering, they will have the ability to establish a profile and provide more details about their company.
- Upon the completion of profile creation, the company will be able to submit jobs that meet their requirements and browse eligible freelancers.
- Freelancers will be able to examine and express interest in these positions.
- Companies can view the list of freelancers who have expressed interest in each of their unique jobs.
- Check out the recommended freelancers for each project. The software creates a list of recommended independent contractors whose skill sets exactly fit a job's requirements.
- They can email freelancers if they're interested in working with them.

Who are freelancers in our application?

A self-employed individual with the skills required to put their expertise on current projects. In

order to give students a place to work on projects in the present, our application also includes students as independent contractors.

Who are businesses in our application?

Companies who are trying to outsource their real-time projects to Freelancers.

2.1.2 Technology Stack For Development

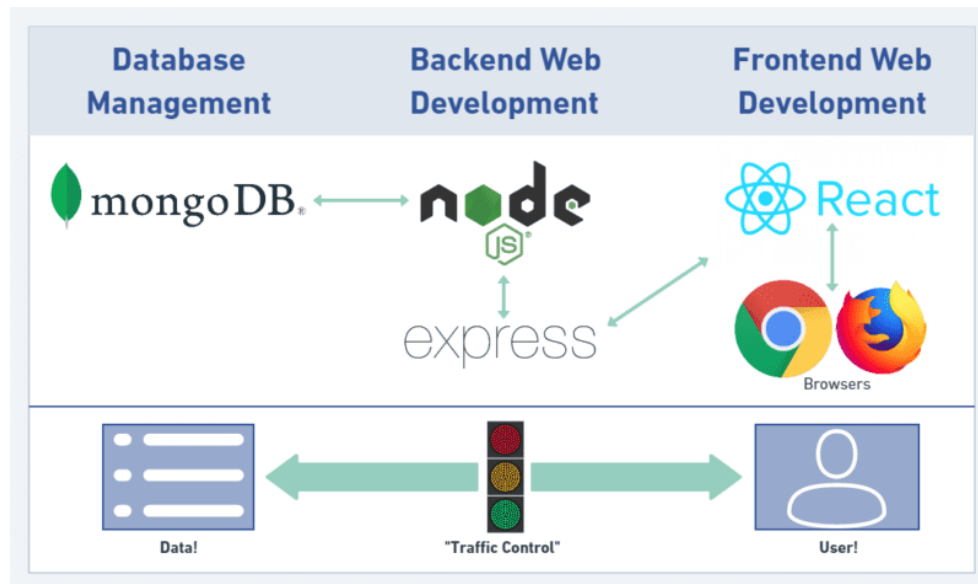


Figure 1: MERN Tech Stack

MERN (MongoDB, Express.js, React.js, Node.js) stack usually contains the following components while developing the application/Project:

Frontend: The user interface is created, views are handled, and components are managed using React.js.

Backend: Express.js and Node.js are used to build server-side functionality, manage API routes, and communicate with databases.

Database: For storing and retrieving data, MongoDB is used as a NoSQL database.

Redux: Redux is a state management tool that enables centralized, predictable data flow throughout the application.

Additional Libraries: Axios for managing HTTP requests, React Router for client-side routing, and Redux Thunk for managing asynchronous Redux actions are just a few examples of the additional libraries and tools that can be used.

This stack offers solution that makes use of Redux for state management while facilitating effective development between the frontend and backend components.

2.1.3 Existing Systems

There are several products available on the market that enable contract employees to find and offer their services online. Examples of such application are UpWork, Fiverr, and Freelancers. Each application has been reviewed and a summary of its capabilities for assisting freelancers has been provided. The bid process that each freelancer must go through in order to win the job is a worry with this application, and these apps encourage team registrations.

Unique Features in the application

- No teams may register, and no freelancers may submit bids for projects.
- The project's budget, duration, and other details are decided by business.
- Giving students the chance to work as freelancers and get experience on a genuine project.
- Creating a recommendation system to assist organizations in finding independent contractors whose skill sets are a perfect match.

2.1.4 Unique and Core Features

Allow people to sign up for accounts and register as either clients or independent contractors.

User profiles: Give users the option to establish and customize their profiles, which might include personal data, skills, employment history, and a portfolio.

Allow customers to post job specifications, including project specifics, a budget, and a deadline. Allow clients and freelancers to leave reviews and ratings for one another based on interactions

Gig Listings: Give freelancers the ability to establish and list their services (gigs), complete with descriptions, prices, and deadlines.

Implement search functionality to assist users in locating particular gigs or jobs based on parameters like location, budget, skillsets, and keywords.

Allow freelancers to submit bids and proposals for jobs they are interested in, along with details about their proposed pricing, delivery schedule, and relevant experience.

Messaging and communication: Set up a messaging platform where clients and freelancers may exchange messages, work out terms, and discuss project specifications.

Allow clients and freelancers to leave reviews and ratings for one another based on their interactions.

Messaging and communication: Set up a messaging platform where clients and freelancers may exchange messages, discuss project specifications

2.1.5 Salient Features

Implement an advanced search and recommendation engine that makes relevant suggestions for freelancers or gigs based on user preferences, previous interactions, and project specifications.

Allow freelancers to display their prior work, portfolios, and case studies to highlight their expertise and draw in potential clients.

Integrate social networking platforms so that users may share their accomplishments and recommendations with their networks as well as advertise their profiles, gigs, and job ads.

Implement safety measures, such as identity verification, user reviews and ratings, dispute resolution procedures, and secure payment methods, to ensure user confidence and safety on the site.

Bulk job posting: Give customers the choice to post numerous work requirements at once, saving time and effort for users with numerous projects.

Offer niche-specific marketplaces within the platform to cater to specialized industries or professions, enabling clients and freelancers to connect with their specific target audience.

2.2 Feasibility Study

2.2.1 Technical Feasibility

The main technologies used in the project are:

- IntelliJ
- VS Code
- Git and Github
- Postman
- Jenkins
- JavaScript - Angular
- JavaScript - NodeJS

- MS SQL Server
- MongoDB

2.2.2 Risk Feasibility

2.2.2.1 Technical Risks:

All the open-source tools and technologies and AWS services used for the project are stable and well established. Also, this application is responsive to Safari, Chrome, IOS, Firefox, Edge, Opera and Android.

2.2.2.2 Development and Deployment Risk:

The project is divided into two categories: development and production. Any new feature or urgently corrected problem is first fixed and pushed into the development environment before being manually tested and then posted to the production bucket. By doing this, we can guarantee that the production branch's active application is operating without error.

2.2.2.3 Process Risks:

If any issue comes up or further changes are to be implemented then they can be immediately implemented.

CHAPTER 3: METHODOLOGY

3.1 Requirements Specification

3.1.1 Purpose

Develop a Web application named “Developer’s Hub (Job Recommendation System)”. The purpose of the application is to develop a platform that helps businesses to find freelancers. For example, this application can help small businesses who are trying to develop a website for their customers, it can help businesses who are trying to hire developers based on their technology requirements, and it can help students to work on real-time projects.

3.1.2 Essential Requirements

Below are the fundamental conditions that must be met for webapp to function. The users are used to categorize the requirements.

Freelancers

User can choose to signUp as a freelancer or student. After login into the application user can create his/her profile. They can see the posted job. If they find that job meets their profile they can show interest by upvoting the option. They can communicate with the organizations Via email.

Business Providers

They will be registered as the organizations. They can post the job. They will be notified when user will show interest into their job. They can browse the recommended Freelancers.

They can communicate with the freelancer via email.

Admin

Admin can manipulate the jobs if its not credibal.

They can also delete user if there are some not professional activity

3.1.3 Non-Functional Requirements

- **Availability**
 - The application should be available to all the users 24*7
 - In any case of any failure, the application should be available in 1 or 2 hours.
- **Security**
 - The application must provide configurable role-based authentication.
- **Performance**
 - The application should be able to handle multiple users at the same time and the application should run smoothly without being crashed and hang.
- **Data Integrity**
 - Data should be accurate across all database tables. Sync Should be there.
- **Usability**
 - The website should be responsive and should be able to load successfully inChrome, Firefox and Safari.
 - The application should be responsive with tablets.

3.2 Diagrams

3.2.1 Use-case Diagrams

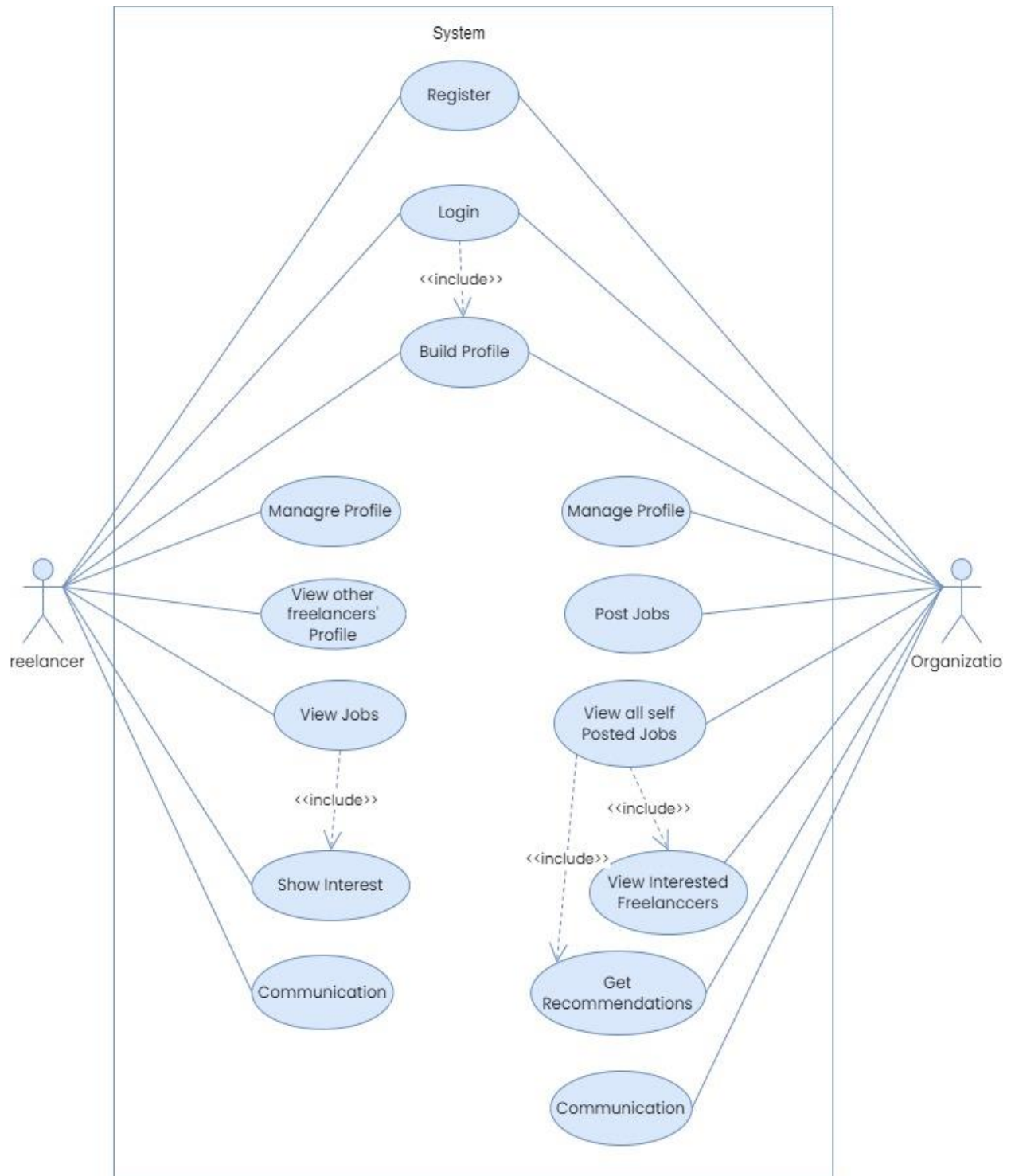


Figure 2 Use-Case Diagram

3.2.2 Class Diagram

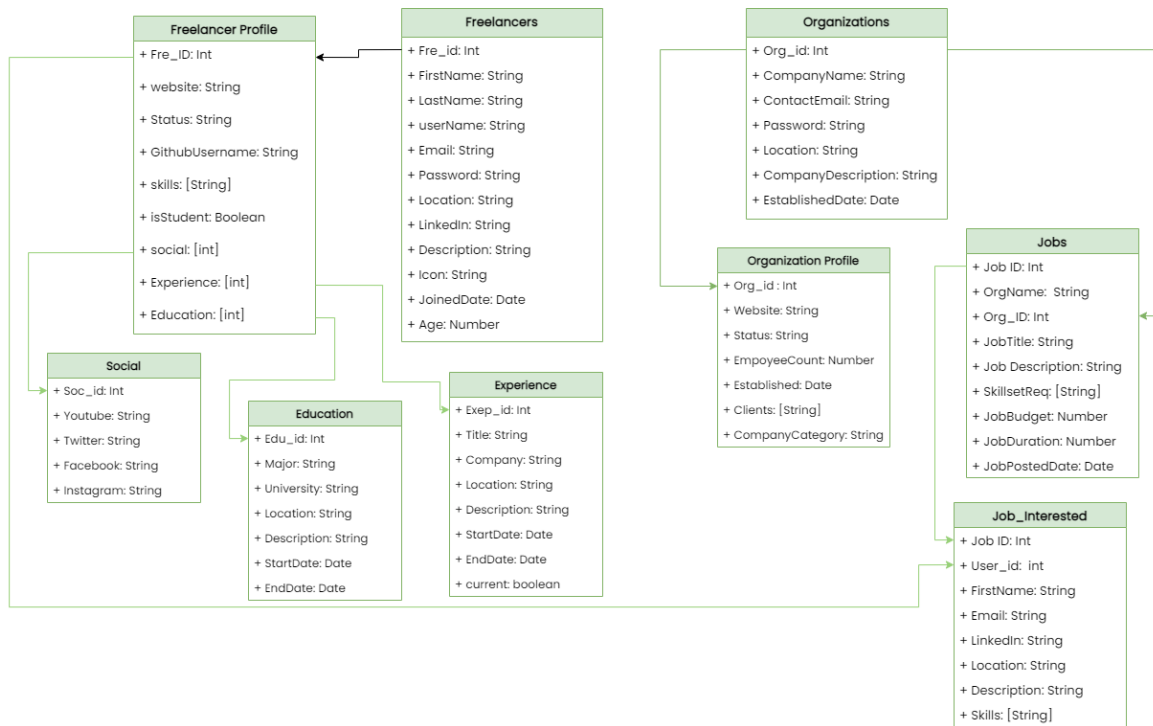


Figure 3 Class Diagram

3.3 System Design

3.3.1 System Architecture

This project follows to monolithic architecture. An application is formed as a single, autonomous unit under a monolithic architectural. A monolithic architecture would mean that frontend and backend of the program are created and implemented as a single entity.

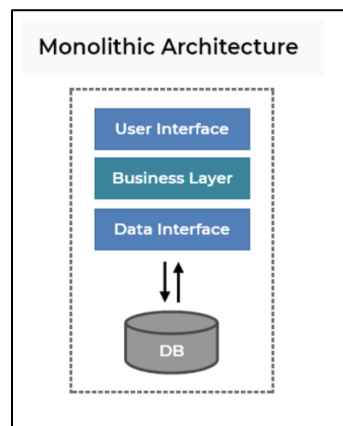


Figure 4 System Architecture

- Client-Side Architecture:
- Server-Side Architecture:
- Database and Data Layer:
- External Services Integration:
- Infrastructure and Deployment:

3.4 Development Phase

3.4.1 Software Development Model

Depending on the project's size and requirements, one can choose from a number of software development approaches. Software development approaches include some of the following.

- Waterfall
- Agile
- Scrum
- Kanban
- Extreme XP

Scrum is known as an alternative approach to agile methodology. In agile the progress is updated by making frequent client interactions. In Scrum, frequent interactions take place but within the team members. After understanding how each methodology works, It is found out that the scrum development approach is best suited for the development of applications with intermediate complexity.

3.4.2 Frontend Implementation

The front-end part of the webapp has been made by using React.js.

Core-Dependencies in the Application:

- “moment”
- “React-redux”
- “React-bootstrap”

Component level State

The data found in the registration and login components are kept in a component

level state.

Application-Level State

In the application we can put component-level states as an example sign Up and Registration, but for the freelancers' profiles and jobs finding, the state should be application level because it needs to be required by a number of other components.

How Actions works?

All of the application's actions send the axios request to the API's endpoints. The axios request changes depending on the different HTTP method, such as GET, POST, PUT, and DELETE. For instance, when a user attempts to get all the organization profiles with the application, axios.get() sends a request to the API endpoint along with the necessary input as a parameter(if applicable).

- **Axios**

In the application axios is used, a component of the JavaScript library, to send HTTP queries to Node.js-made API endpoints. With the help of axios, any CRUD activities for HTTP requests can be carried out.

The necessary input fields are built as a function, freelancersignup(), and they are supplied to it as an argument. The "content-type" is supplied as part of the header when performing an axios request like a POST, and the "content-body" contains all the fields that the user must fill out. The action SIGNUP_SUCCESS is sent to the reducer once the response is received, and the reducer then passes the state down to the components.

Action files:

FreelanceroAuth.js

Actions present in the file will be called when freelancer will try to login or signup in the portal.

Actions present in the file are as below.

- Loadfreelancer()

- signupFreelaner()
- Freelancerlogin()
- Logout()

OrganizationoAuth.js

Actions present in the file are a s below.

- LoadOrganization()
- signupOrganization()
- Organizationlogin()
- Logout()

Alerts.js

When an alert needs to be displayed in the application's front end, the function Alert(), which can be called, is generated.

freelancerProf.js

There are some action made in this particular file which call organization profile related URL end points

- getcurfreelancerProf()
- createfreelancerProf()
- addexp() and addedu()
- delexp() and deledu()
- getprofs()

OrganizationProf.js

The actions created in this file call the organization profile-related API endpoints.

Jobs.js

Actions created in this file hit the API endpoints related to Jobs.

Addjobs()

Addinterests()

Getjobsdetails()

Getalljobs()

Working of Reducer

The reducer receives the stated actions and dispatches them to it. The reducer then updates the state and sends it on to the components.

The initial state object is produced in each reducer file, and reducers maintain the application's state. The state object is modified based on the actions when the actions are sent to the reducer.

The payload that the state object needs is provided by the axios call answer.

Reducers files:

FreelanceroAuth.js

OrganizationoAuth.js

Alerts.js

OrganizationProf.js

freelancerProf.js

jobs.js

3.4.3 Backend Implementation

In the backend also known as server. It is mainly divided into 4 sub-parts APIs (Routes), Configurations, Middleware and app.js file. There is a file named *package.json* file which contains all kind of dependencies that will be required for the application from the backend side.

Encryption of password

We must deliver a JSON web token once the user has signed up for the application.

These tokens can be used to log in as the user and open system-protected routes. In application is utilizing the user's id as the payload in order to produce the JWT token. The payload can be anything connected to the user.

JWT was installed with the command "npm install jsonwebtoken." After it is set up, we

can use `jwt.sign()` to create the web token and `jwt.verify()` to check that it is valid when a user tries to access a protected route.

The API route is used to carry out user authentication. The user id is delivered as a payload and a JWT secret—which is kept in the configuration file of our application as well as a JWT expiration time are needed. The user might need to log back into the application if the time runs out.

The token will be sent as a response as soon as it is produced. Once the user has the token, we can send it along with the API request as part of the header, allowing the author to confirm the user's identity before granting access to restricted routes.

Mongoose Schemas

The next step was to create mongoose Schema when the server was set up and the database was linked. There are mainly 5 models are used.

- Freelancers
- Freelancer Profiles
- Businesses
- Business Profiles
- Job

Routes and APIs

In the webapp there are multiple routes are present In which some of them are private and some of them are Public.

Route Files:

- `oAuth.js`
- `Organizations.js`
- `Freelancers.js`
- `OrganizationProfiles.js`
- `FreelancerProfiles.js`
- `Jobs.js`

In the following tables, Each includes information on each route's HTTP method, its path, and the middleware that is used to verify the user's identity before accessing that route. The input fields for this route are defined with a description and, if the HTTP method is POST, a request for additional information from the user.

A) oAuth.js

HTTP Method	URL Endpoints	Middleware	Description OF the API	Input Param
GET	api/oauth/freelancers	User Authentication	This route verifies freelancers' identities and returns the information they gave after registration.	None
POST	api/oauth/freelancers	None	This method authorises freelancers to access the application. It creates the JWT token required for freelancer authentication.	Email and Password
GET	api/oauth/organization	Organization Authentication	This will fetch the Organization details which has been given at the registration time.	None
POST	api/oauth/organization	None	this route logs organizations into the application. It creates the JWT token needed to verify the identity of the company.	Email and Password

Table 3 Authentication URL end points

B) FreelancerProfiles.js

HTTP Method	URL Endpoints	Middleware	Description OF the API	Input Param
GET	api/freelancerprofiles/m	User Authentication	This route will fetch all the details of the profile of user to the user.	None
GET	api/freelancerprofiles	None	This will get all the freelancers' profiles which are present in the DB.	None
GET	Api/freelancerprofiles/:user_id	None	This route will fetch the data of the particular freelancer whose ID has been passed	None
POST	api/freelancerprofiles	User Authentication	By gathering the essential information, this POST route assists in the creation of profiles for freelancers.	Studey details, portfolio-link, status, username of github ,social Media Links
PUT	api/freelancerprofiles/experience	User Authentication	After creating the profile user will be able to add experience and can update profile constantly.	Title, Company, Location, From-to, Description
PUT	api/freelancerprofiles/education	User Authentication	If is user is a student, then he/she can add education or specific course Details in the profile.	Course of Study, University, Location, Description
DELETE	api/freelancerprofiles	Admin Authentication	This route will delete the user profile for given ID	None

DELETE	api/freelancerprofiles/ experience/: experience_id	User Authentication	After adding multiple experience if user wants to delete the any experience details, this route will fetch the ID and delete it.	None
DELETE	api/freelancerprofiles/ education/: education_id	User Authentication	If user(Student) wants to delete any education details Then this this route takes an ID and delete it.	None

Table 4 Freelancers Profile End Points

C) OrganizationProfiles.js

HTTP Method	URL Endpoints	Middleware	Description OF the API	Input Param
GET	Api/organizationprofiles/ me	Organization Authentication	Route will fetch all the Profile information of the Particular Organization.	None
GET	Api/organizationprofiles	None	This will fetch all the organization profiles present in the DB	None
GET	api/organizationprofiles/: business_id	None	Unique business_id will be passed as param and route will fetch that particular profile.	None
POST	api/organizationprofiles	Organization Authentication	This route will help in creating profile for organization once it register in the portal	Web Link, Status, Established date, clients, organization category, employee count

DELETE	api/organizationprofiles	Admin Authentication	This route will delete the profile for given ID	None
---------------	--------------------------	----------------------	---	------

Table 5 Organization Profiles End Points

D) Job.js

HTTP Method	URL Endpoints	Middleware	Description OF the API	Input Param
GET	api/jobs	User and Organization Authentication	This Route will fetch all the Posted Jobs by all the organization	None
GET	api/jobs/:job_id	Organization Authentication	The details of job whose ID is passed in the param will be shown by using the given route.	None
POST	api/jobs	Organization authentication	After having required details organization will be able to post a job which will be stored in the DB	Job title, Description, Skill sets, Budget, Job duration
DELETE	api/jobs/:job_id	Organization Authentication	Organizations can delete posted jobs, for which relevant job id will be passed in the param	None
PUT	api/jobs/interested/:user_id	User Authentication	By this route user can show interest in the job, after that in the job in “interested” property user_id will be stored	None
PUT	api/jobs/uninterest/:job_id	User Authentication	By this route user can remove already shown interest on the job	None

Table 6 Jobs End Points

The express validator is used to verify the input fields before sending requests to the server.

CHAPTER 4: RESULTS

4.1 Screenshots of Outcomes

1. Landing Page

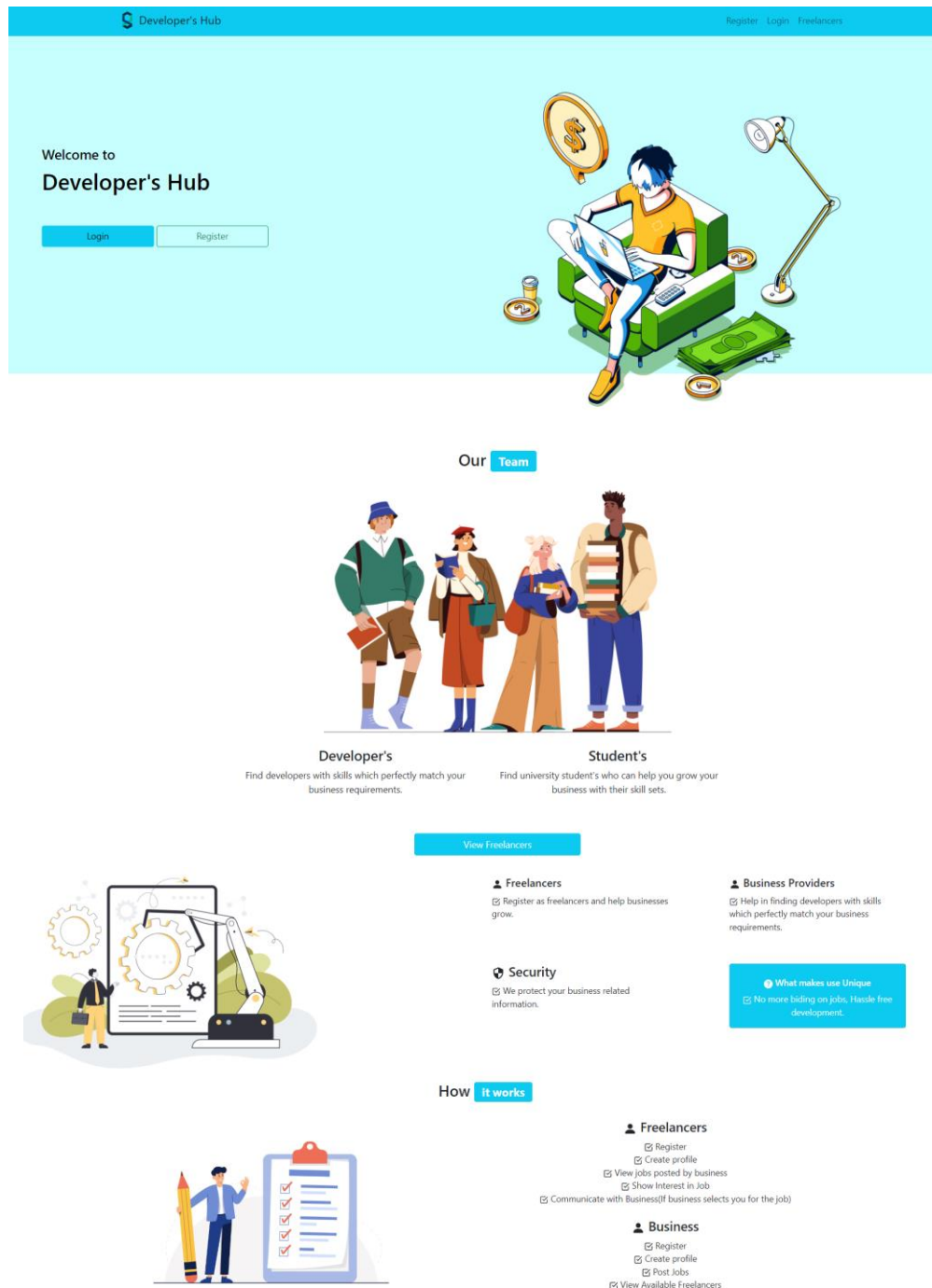
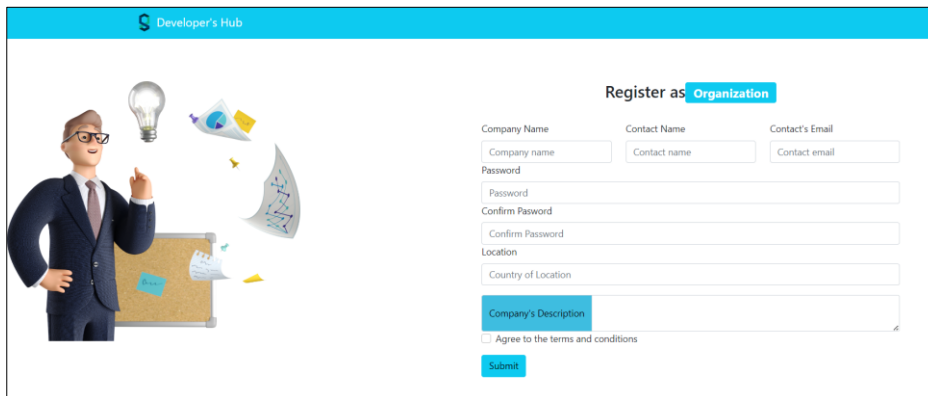


Figure 5 Landing Page

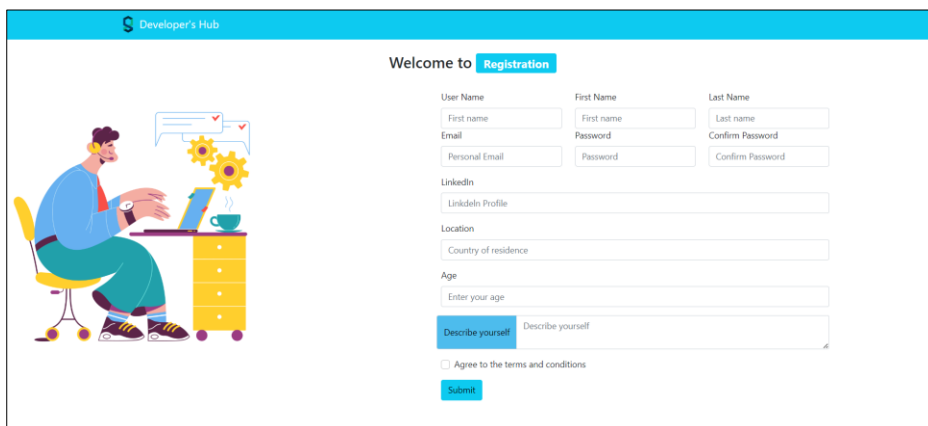
2. Registration page for Organization



The registration page for an organization features a blue header with the 'Developer's Hub' logo. On the left, an illustration shows a man in a suit pointing at a lightbulb, with a corkboard and papers nearby. The main heading is 'Register as Organization'. The form includes fields for 'Company Name', 'Contact Name', and 'Contact's Email'. Below these are 'Password' and 'Confirm Password' fields. The 'Location' section has a 'Country of Location' dropdown and a 'Company's Description' text area. At the bottom, there is a checkbox for 'Agree to the terms and conditions' and a 'Submit' button.

Figure 6 Registration Page (Organization)

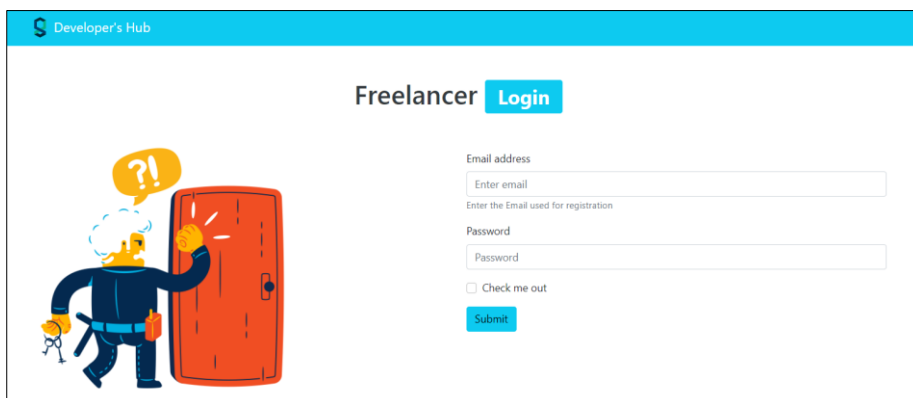
3. Registration of Freelancer



The registration page for a freelancer features a blue header with the 'Developer's Hub' logo. On the left, an illustration shows a person sitting at a desk with a laptop, a coffee cup, and gears. The main heading is 'Welcome to Registration'. The form includes fields for 'User Name' (First name, Last name), 'Email' (Personal Email, Password, Confirm Password), 'LinkedIn' (LinkedIn Profile), 'Location' (Country of residence), and 'Age' (Enter your age). There is a 'Describe yourself' text area and a checkbox for 'Agree to the terms and conditions'. A 'Submit' button is at the bottom.

Figure 7 Registration Page (Freelancer)

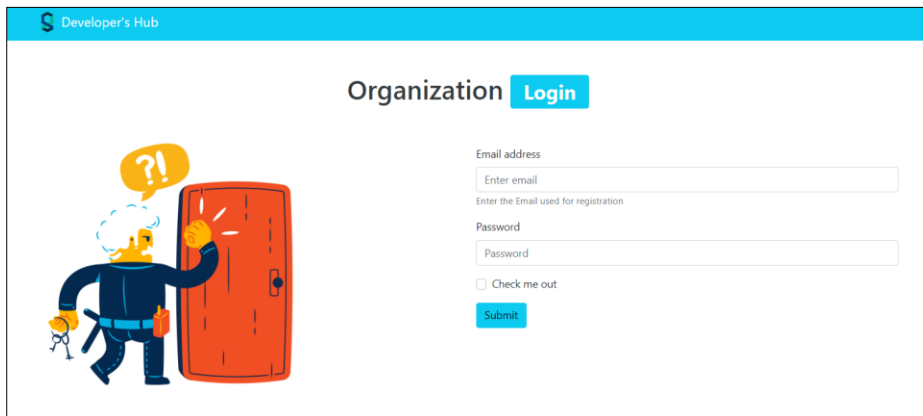
4. Login of Freelancer



The login page for a freelancer features a blue header with the 'Developer's Hub' logo. On the left, an illustration shows a person in a blue uniform with a question mark above their head, standing next to a red door. The main heading is 'Freelancer Login'. The form includes fields for 'Email address' (Enter email, Enter the Email used for registration) and 'Password' (Password). There is a checkbox for 'Check me out' and a 'Submit' button.

Figure 8 Login Page (Freelancer)

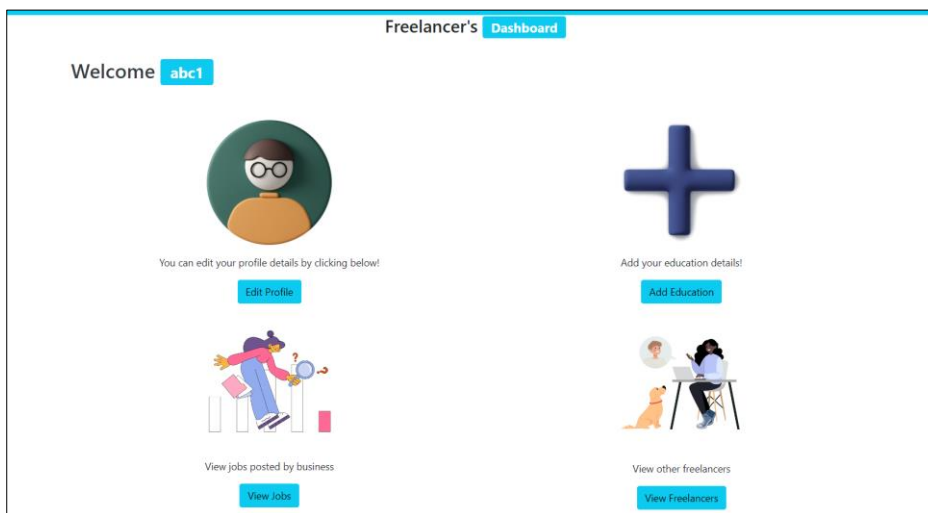
5. Login of Organization



The screenshot shows the 'Organization Login' page. At the top left is the 'Developer's Hub' logo. The main heading is 'Organization Login'. On the left, there is an illustration of a person in a blue suit with a yellow speech bubble containing a question mark and exclamation mark, standing next to a red door. On the right, there is a login form with the following fields: 'Email address' (with a placeholder 'Enter email' and a note 'Enter the Email used for registration'), 'Password' (with a placeholder 'Password'), and a checkbox labeled 'Check me out'. A blue 'Submit' button is at the bottom right of the form.

Figure 9 Login Page (Organization)

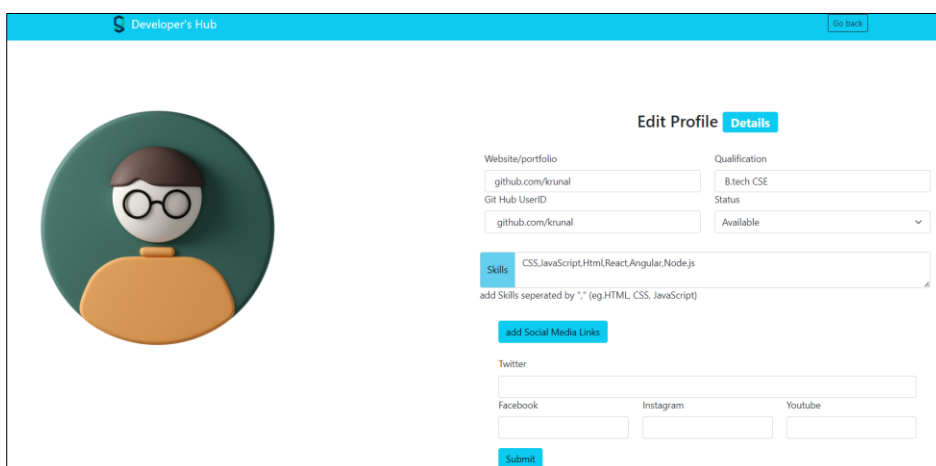
6. Freelancer Dashboard



The screenshot shows the 'Freelancer's Dashboard'. At the top left is the 'Developer's Hub' logo. The main heading is 'Freelancer's Dashboard'. Below the heading, it says 'Welcome abc1'. There are four main action cards: 1. 'Edit Profile' with a person icon and the text 'You can edit your profile details by clicking below!'. 2. 'Add Education' with a large blue plus icon and the text 'Add your education details!'. 3. 'View Jobs' with an icon of a person at a desk and the text 'View jobs posted by business'. 4. 'View Freelancers' with an icon of two people at a desk and the text 'View other freelancers'. Each card has a blue button at the bottom.

Figure 10 Dashboard Freelancer

7. Create and Edit Profile of freelancer



The screenshot shows the 'Edit Profile' page. At the top left is the 'Developer's Hub' logo. The main heading is 'Edit Profile Details'. On the left, there is a large circular profile picture placeholder. On the right, there is a form with the following fields: 'Website/portfolio' (placeholder 'github.com/krunal'), 'Qualification' (placeholder 'B.tech CSE'), 'Git Hub UserID' (placeholder 'github.com/krunal'), 'Status' (dropdown menu with 'Available' selected), 'Skills' (text area with 'CSS, JavaScript, HTML, React, Angular, Node.js' and a note 'add Skills separated by "." (eg. HTML, CSS, JavaScript)'), 'add Social Media Links' (blue button), 'Twitter' (text input), 'Facebook' (text input), 'Instagram' (text input), and 'Youtube' (text input). A blue 'Submit' button is at the bottom right.

Figure 11 Create and Edit Profile of Freelancer

8. Add Education

Developer's Hub [Go back](#)

Add Education

Course Of Study

University

Location

From(YYYY-MM-DD)

yyyy-MM-dd

To(YYYY-MM-DD)

yyyy-MM-dd

Course Description

Submit

Figure 12 Add Education Details Page

9. View Freelancers

Developer's Hub [Go back](#)

Our Freelancers

Browse and Connect with Freelancers

@UserName: Emma1

@Status: Available

Name: Emma

Skills:

- HTML
- CSS
- JavaScript
- NodeJS

About: I'm a full stack web developer with expertise in HTML, CSS, JavaScript, ReactJS, Angular and NodeJS

[View Profile](#)

Location: United States of America

@UserName: Audrey1

@Status: Available

Figure 13 View All the Freelancer

10. View all jobs

Developer's Hub [Go back](#)

View All Jobs

Company: asile

Job: dba

[View Business details](#)

Company Name: asile

Contact Name: aray

Email: abc@gmail.com

Location: Ahmedabad

Skills Required:

- sql
- aws
- cloud
- db

Salary: 2000000

Views: 34

[Interested](#)

Location: Ahmedabad

Posted On: 05/05/2023

Figure 14 View all Jobs

11. See one particular Freelancer's Profile

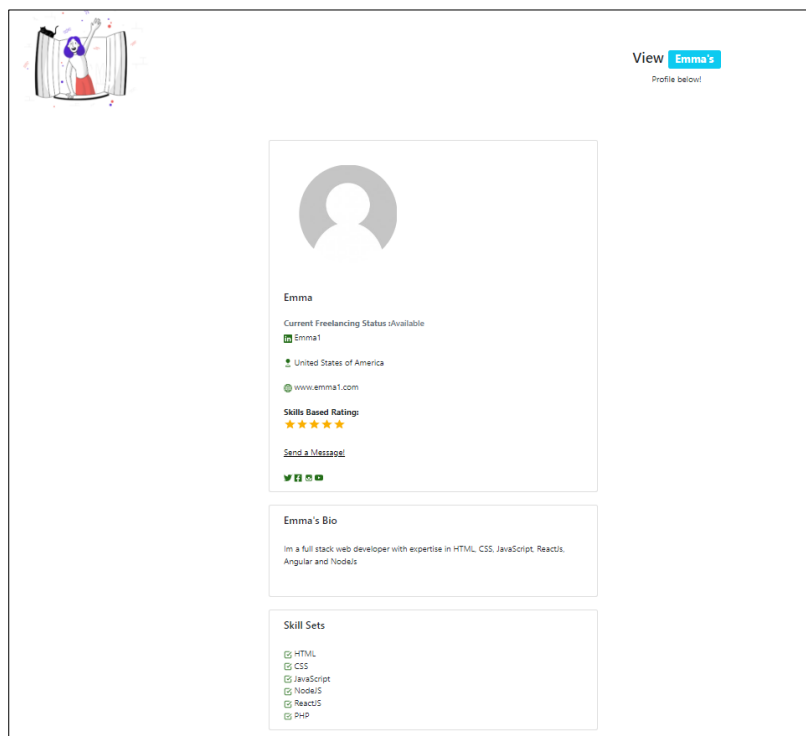


Figure 15 Individual Freelancer's Profile

12. Education List for Student

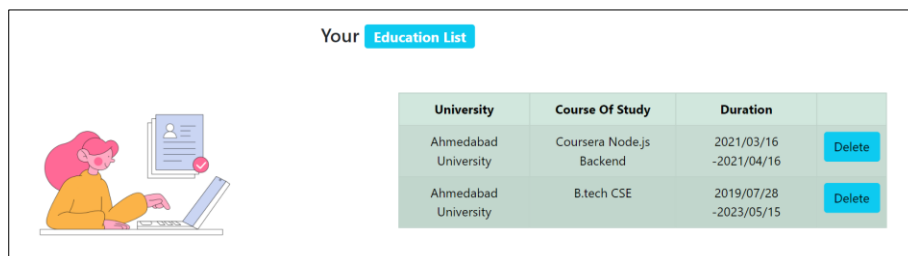


Figure 16 Education List

13. Create and Edit organization's Profile

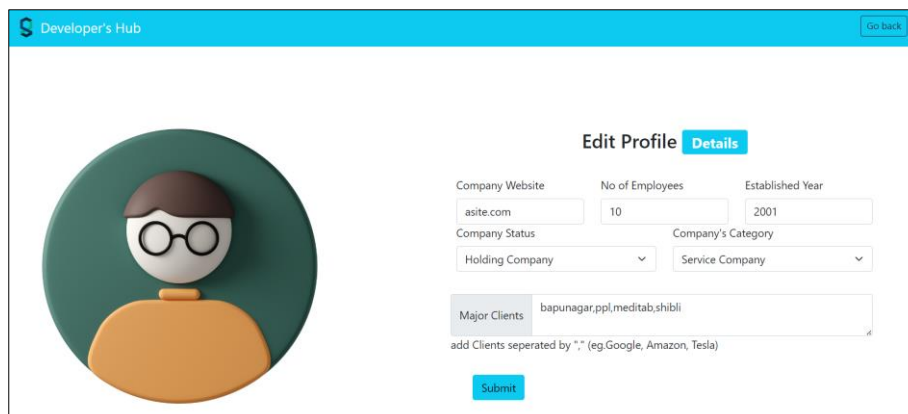


Figure 17 Create and edit Organization's Profile

14. Organization's Dashboard

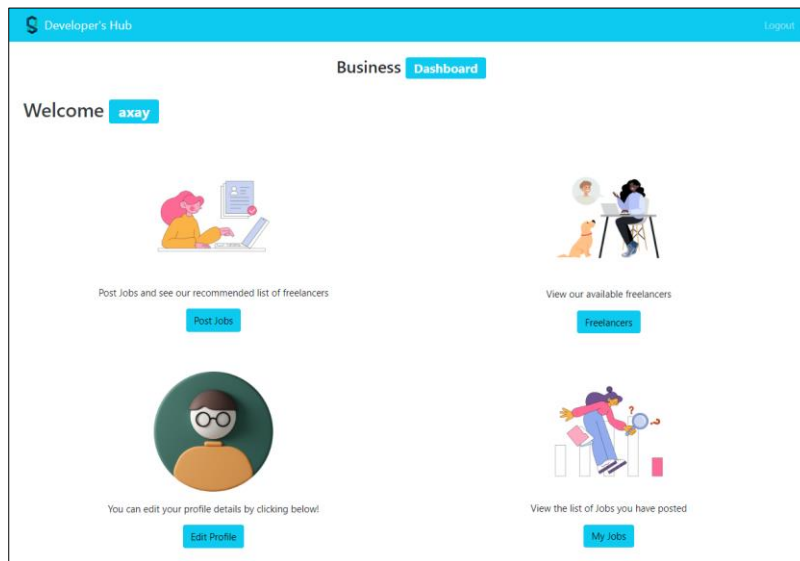


Figure 18 Dashboard For Organization

15. Posting job by organization

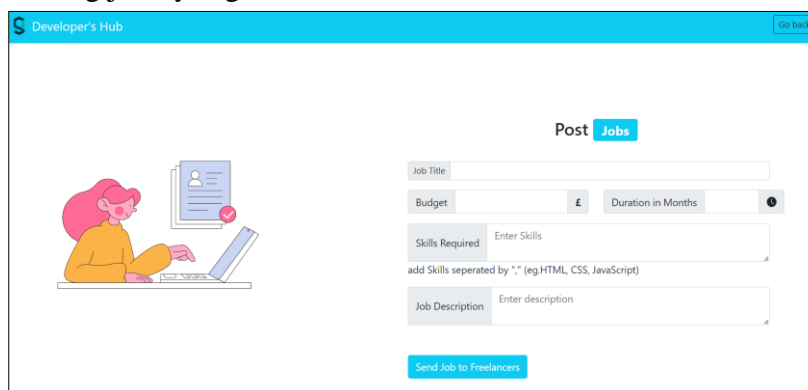


Figure 19 Posting Job

16. See all the jobs posted by one particular Organization

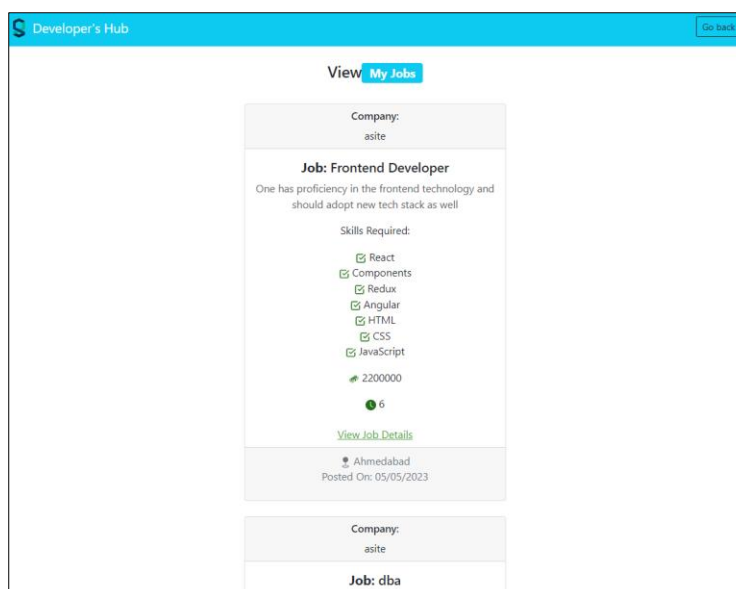


Figure 20 View All Jobs by individual ORG

17. See all the details of one particular Job

Developer's Hub

Go back

Job Name: dba

Job Details

Interested Freelancer's

No of Freelancer's interested - 1

dba

Budget

2000000

Duration

34 Months

Essential Skills

sql

aws

cloud

db

Recommendations

Below Freelancers match you requirements!

No Freelancer's that match your required skill set! View interested freelancer's instead

Developer's Hub

Go back

Job Name: dba

Job Details

Interested Freelancer's

abc1

abc@gmail.com

Ahmedabad

https://www.linkedin.com

Send a Message to abc1

Recommendations

Below Freelancers match you requirements!

No Freelancer's that match your required skill set! View interested freelancer's instead

Figure 21 All Details Regarding Posted Job

4.2 My Contribution to the Project

The project's objectives, features, and target audience have all been defined through brainstorming and conceptualization. I made a thorough project plan that outlined the developmental phases and checkpoints. I'm in charge of creating the system architecture, which includes the frontend components, backend APIs, and database structure. This includes figuring out the application's overall structure and the relationships between its many modules and components. I used tools like React.js to create the front end. In order to do this, user interfaces must be developed, responsive designs must be used, and interactive elements like user profiles, message systems, and job postings must be integrated. I have worked with server-side development using tools like Node.js and Express.js. This would entail creating APIs, putting authentication and permission systems in place, integrating with databases, and making sure that data processing and storage are done effectively. In order to ensure effective communication between the frontend and backend of the system, I am in charge of integrating various system components. Additionally, you would carry out exhaustive testing to find any issues and resolve them, ensuring the dependability and stability of the application. I have overseen the project's development, liaising with the team, delegating duties, and assuring timely delivery.

4.3 Learning Outcomes

Work on the frontend and backend of an application requires both of these areas. I've developed a solid understanding of creating user interfaces, managing client-server communications, and creating backend systems. Scalability, performance, and maintainability are things to keep in mind when designing and implementing an application's architecture. I've gained knowledge on how to construct a system that can support a big number of users, organize a codebase, and separate concerns. An application must have effective user management. I pick up some implementation skills for managing user profiles, logging in, and user registration. In order to ensure secure access to various application components, I also develop knowledge of authentication and authorization procedures. Real-time user interaction is frequently required by platforms. By creating functionalities like messaging systems, developers will gain practical expertise with tools like WebSockets that allow for real-time, two-way communication. Effective project management abilities are needed while

developing a complicated platform. I gained knowledge of how to organize, prioritize, and manage projects, work with a team, and fulfill project deadlines. These abilities are helpful in a variety of professional contexts. I faced obstacles, bugs, and technological problems during the development process. In addition to improving my ability to solve problems, troubleshooting and solution-finding will also teach me how to debug and optimize my code.

A platform can spark an entrepreneurial mindset. Understanding business models, monetization techniques, and client acquisition strategies will help you better appreciate the opportunities and challenges of operating an online marketplace. Overall, creating a project offers a complete learning experience that combines technical abilities with project management, problem-solving, and business objectives.

4.4 Real World Application

A project has numerous practical applications in the real world. It could completely transform how people and companies communicate and work together. For instance, in the creative sector, it might provide a platform for independent designers, authors, and painters to present their portfolios and solicit work from clients all over the world. It can make remote software development easier in the tech industry, enabling businesses to access a global talent pool for certain projects. It can also be used in the consulting sector, allowing experts to offer their expertise and opinions on a freelance basis. Additionally, the platform can be expanded to include sectors like virtual assistance, marketing, translation, and many more, giving people the chance to make money from their abilities and giving businesses access to specialized services on demand.

4.5 Future Works

In the future, a project will concentrate on a number of areas to improve its usability and functionality. First and foremost, upgrading the database to MS SQL Server would be essential for enhancing scalability. By providing strong scalability features like horizontal partitioning and distributed database support, MS SQL Server enables the platform to handle growing user loads and data volumes. This guarantees that the platform may

develop without interruption as the user base grows. Second, creating a responsive and seamless user experience depends on performance optimization. Utilizing MS SQL Server's query optimization, in-memory caching, and other performance optimization capabilities can greatly enhance system speed and query response times. As a result, pages load more quickly, search results appear more quickly, and transaction processing is improved. Furthermore, increasing functionality is essential to remaining competitive and satisfying customer demands. Features like real-time collaboration tools, sophisticated search capabilities, interaction with well-known third-party services or APIs, increased payment possibilities, and improved analytics for users and administrators can all be added by the platform. These upgrades raise the platform's total value, bringing in additional users and boosting engagement. Finally, UI adjustments can be quite important in delivering a user experience that is both intuitive and visually appealing. Modern design ideas, responsive layouts, and improved navigation can be included into the user interface to improve usability and make the platform more aesthetically pleasing. To guarantee a seamless and pleasurable user experience across various platforms, factors like user feedback, accessibility, and mobile responsiveness should be taken into account. A project can develop into a comprehensive and feature-rich platform, providing a superior user experience and remaining ahead in the cutthroat market by concentrating on certain future work areas.

CHAPTER 5: CONCLUSION

In conclusion, a project has the potential to significantly alter how people communicate and work together in the digital space. Such a project responds to the changing needs of organizations and people looking for specialized services and flexible work schedules by developing a platform that supports freelancing, remote employment, and service exchanges. The project can offer a reliable and scalable solution thanks to a comprehensive technological stack that includes the MERN stack and incorporates extra tools like Redux for state management. In areas like full-stack development, application architecture, user management, payment integration, real-time communication, project management, problem-solving, and user experience design, the development process of such a project generates beneficial learning results. The project may also continue to develop and give an amazing user experience by taking into account future work such as database migration, performance optimization, increased functionality, and UI changes. This will strengthen its position in the rapidly expanding freelance and gig economy industry.

CHAPTER 6: REFERENCES

1. *React-Bootstrap*. react. (n.d.). Retrieved May 5, 2023, from <https://react-bootstrap.github.io/>
2. React. (n.d.). Retrieved May 5, 2023, from <https://react.dev/>
3. *Redux - a predictable state container for JavaScript apps.: Redux*. A predictable state container for JavaScript apps. (n.d.). Retrieved May 5, 2023, from <https://redux.js.org/>
4. Node.js. (n.d.). Retrieved May 5, 2023, from <https://nodejs.org/en>
5. *NPM*. npm. (n.d.). Retrieved May 5, 2023, from <https://www.npmjs.com/>
6. Dey, V. (2022, May 30). *Collaborative filtering vs content-based filtering for Recommender Systems*. Analytics India Magazine. Retrieved May 5, 2023, from <https://analyticsindiamag.com/collaborative-filtering-vs-content-based-filtering-for-recommender-systems/>