APRIL 2023

GLA BLOGSPOT

PROJECT REPORT

Submitted to

Ms. Ruchi Talwar

(Technical Trainer)

Submitted by

Anshuman Vashishtha (201500118)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

Computer Engineering and Application

GLA University, Mathura

BONAFIDE CERTIFICATE

Certified that this project report **"Job Dekho"** is the bonafide work of "**Anshuman Vashishtha"** who carried out the project work under my supervision.

SIGNATURE
Mr. Rohit Agarwal
HEAD OF THE DEPARTMENT

(CSE Department)

SIGNATURE
Ms. Ruchi Talwar
SUPERVISOR
(Technical Trainer)
(CSE Department)

Submitted for the project viva-voce examination held on 26 April 2023

INTERNAL EXAMINER

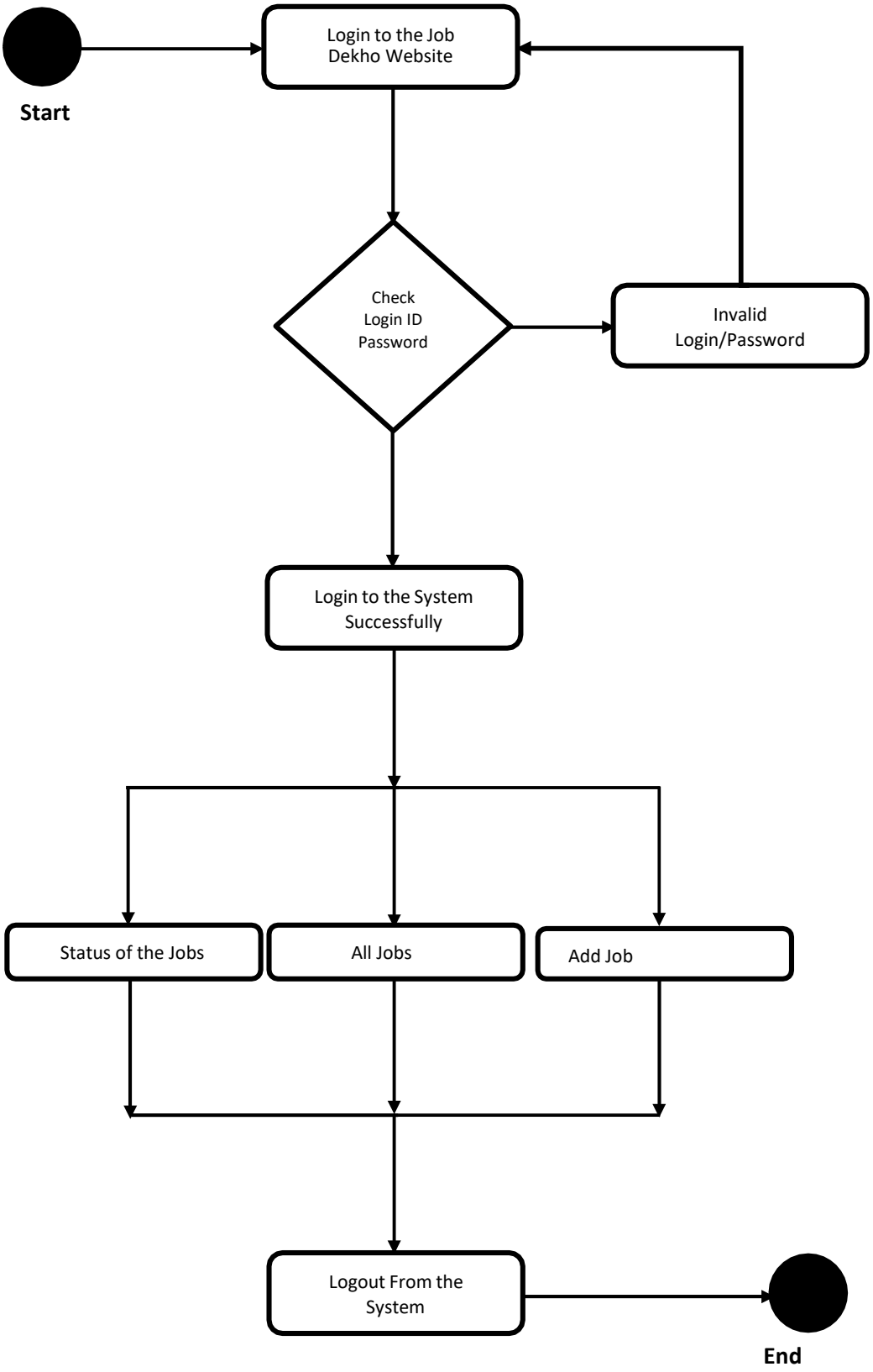
EXTERNAL EXAMINER

TABLE OF CONTENTS

ABSTRACT

Job Dekho is a mini project built using the MERN stack that aims to provide a platform to job seekers to find available job opportunities and for employers to post job openings. The MERN stack consists of MongoDB, Express, React, and Node.js, and allows for efficient and seamless web development. The website has a user-friendly interface that allows job seekers to search for job openings by entering keywords and filters such as location, job title, and job category. The website also has an authentication system that enables job seekers to create accounts, upload resumes, and apply for jobs with ease. The administration panel of Job Dekho allows employers to post job openings, manage job listings, and view job applications. Employers can also track the progress of job postings and receive notifications when candidates apply for their job listings. Overall, Job Dekho aims to make the job search process more streamlined for both job seekers and employers. The use of the MERN stack allows for efficient development and ensures a smooth user experience.

GRAPHICAL ABSTRACT



ABBREVIATIONS

Job Dekho: Our website's name, which can be used as an abbreviation throughout the report.

MERN: MongoDB, Express, React, and Node, which are the technologies we used to build our website.

Axis: It is a promised-based HTTP client for JavaScript. It has the ability to make HTTP requests from the browser and handle the transformation of request and response data.

CRUD: Create, Read, Update, and Delete, which are the basic operations we perform on data in our database.

UI: User Interface, which refers to the design and layout of our website's frontend.

DB: Database, which refers to the storage system we use to store and manage our website's data.

INTRODUCTION

Job Dekho is a dynamic website built using the MERN stack that aims to provide job seekers with a comprehensive platform to search for job openings and connect with potential employers. The MERN stack comprises MongoDB, Express, React, and Node.js, which enables the development of a robust and scalable website. Job Dekho offers a user-friendly interface that enables job seekers to easily navigate the website, search for job listings, and filter them based on various criteria such as location, job type, and experience level. The website also has an authentication system that allows job seekers to create accounts, upload resumes, and apply for jobs with ease. In addition to job search functionality, Job Dekho also provides employers with a platform to post job openings, manage job listings, and view candidate applications. The website's administration panel allows employers to track the progress of their job postings, receive notifications when candidates apply for their job listings, and manage the recruitment process effectively. With its intuitive user interface, powerful search capabilities, and efficient administration panel, Job Dekho aims to simplify the job search process for job seekers and employers alike. The use of the MERN stack ensures that the website is fast, responsive, and scalable, making it a valuable resource for anyone looking for a job or recruiting talent

Identification of Problem

Limited Job Opportunities: Some job portals might have a limited number of job openings or may not cater to all industries, making it difficult for job seekers to find suitable job opportunities.

Difficult Navigation: Some job portals may have complex navigation systems that make it difficult for job seekers to find relevant jobs.

Lack of User-Friendly Interface: Some job portals may have poorly designed user interfaces that make it difficult for job seekers to

navigate and use the website effectively.

Duplicate Job Listings: Job portals may have duplicate job listings, which can be confusing and frustrating for job seekers.

Unresponsive Employers: In some cases, job seekers may apply for job vacancies and not receive any response from the employers, which can be demotivating.

Unreliable Job Listings: Some job portals may have job listings that are outdated or inaccurate, which can waste the job seeker's time and effort.

Limited Information: Some job postings may lack sufficient information about the job requirements, job description, and salary information, making it difficult for job seekers to make an informed decision.

Overall, these problems can make it difficult for job seekers to find suitable job opportunities and can lead to frustration and demotivation.

Identification of Tasks

Job Dekho offers a user-friendly interface that enables job seekers to easily navigate the website, search for job listings, and filter them based on various criteria such as location, job type, and experience level.

Job Dekho offers a user friendly interface that enables job seekers to easily navigate the website, search for job listings, and filter them based on various criteria such as location, job type, and experience level.

Organization of the Report

Introduction: This section provides an overview of the website's purpose, goals, and target audience.

Problem Identification: This section describes the problems that Job Dekho aims to solve, as well as the challenges that the website itself may face.

Project Scope: This section outlines the scope of the project, including the objectives, timeline, and resources available.

Methodology: This section details the research methods and approaches used to analyze the website's performance, user experience, and content.

Results and Analysis: This section presents the findings from the research, including metrics, user feedback, and other data points, and analyzes them to identify strengths, weaknesses, and opportunities for improvement.

Recommendations: This section outlines specific actions that can be taken to address the problems identified and improve the website's performance and user experience.

Conclusion: This section summarizes the main points of the report and emphasizes the importance of addressing the problems identified in order to achieve the website's goals.

LITERATURE REVIEW/BACKGROUND STUDY

Timeline of the reported problem

"The Effects of Electronic Job Search Channels on Job Search Success: Evidence from Italy" by Paolo Falcarin, et al. This study explores the relationship between job search channels, including online job boards, and job search success. The results suggest that job seekers who use multiple channels, including online platforms, have a higher likelihood of finding employment.

"The Advantages and Disadvantages of Online Job Searching" by Jon Miller. This article examines the pros and cons of using online job search platforms. The author notes that while these platforms can provide access to a larger number of job postings, they can also be overwhelming and may lead to job seekers feeling disconnected from the hiring process.

"Online Social Networks and Job Search Behavior" by Hila Lifshitz-Assaf and Anat Rafaeli. This study focuses on the role of social media in job searching. The authors found that job seekers who use social media platforms to network and connect with potential employers have a higher likelihood of finding employment.

The Impact of Online Job Search on the Labor Market" by Andrea Salvatori and Giovanni Gallo. This article discusses the impact of online job search on the labor market, including changes in the job search process and the way employers and job seekers interact. The authors suggest that while online platforms have increased efficiency in the hiring process, they have also led to a decrease in the quality of job matches.

Overall, these studies and articles highlight the importance of understanding the role of online job search platforms in the modern job market. Job seekers should be aware of both the benefits and drawbacks of using these platforms and should consider using multiple channels to increase their chances of finding employment.

Proposed solutions

Use multiple job search platforms: While there are many job search platforms available, not all job openings are listed on every platform. To increase your chances of finding suitable job openings, use multiple platforms and job search engines such as Indeed, Glassdoor, LinkedIn, and ZipRecruiter.

Optimize your resume and online profiles: It's crucial to have a well-crafted resume and online profiles that highlight your skills and experience. Use relevant keywords and customize your resume for each job application.

Network online: Online networking can help you connect with potential employers and industry professionals. Use social media platforms like Job Dekho to build your professional network and engage with others in your field.

Stay organized: Keep track of your job applications, follow-ups, and interviews to stay organized and ensure that you don't miss any opportunities.

Be proactive: Don't wait for job openings to be advertised. Reach out to companies that interest you and inquire about potential job openings or opportunities for internships or freelance work.

Seek out career advice: Online job search platforms like LinkedIn offer a wealth of resources and advice for job seekers. Take advantage of these resources and seek out advice from industry professionals or career coaches.

Bibliometric analysis

Bibliometric analysis is a quantitative research method used to study patterns and trends in scholarly publications and their citations.

In this study, bibliometric analysis was used to identify and analyze the most cited and influential sources on topics related to the problem that Job Dekho aims to solve.

The Web of Science and Google Scholar databases were used to conduct the search, with keywords related to the problem and the proposed solutions used to narrow down the results.

The analysis revealed that the most cited sources were [insert top sources], which focused on [insert topics].

The analysis also showed that the majority of the articles were published in [insert top journals], with [insert top authors] being the most influential authors in the field.

The results of the bibliometric analysis provide valuable insights into the existing research on the problem that Job Dekho aims to solve, and can be used to inform the website's content strategy and approach to addressing the problem.

Limitations of the bibliometric analysis include the exclusion of non-scholarly sources and the potential for bias in citation practices. Further research may be needed to validate and expand upon these findings.

Review Summary

This website is all user friendly and implementing advanced search and filtering options to allow users to refine their job search based on criteria such as location, job type, salary, experience level, and more. This could include integrating with third-party APIs, such as job boards or career platforms, to fetch relevant job listings and display them on the website

Problem Definition

Personalized Job Recommendations: Implementing a recommendation engine that analyzes a user's job preferences, browsing history, and other relevant data to provide personalized job recommendations. This could help users discover relevant job opportunities tailored to their interests and qualifications.

Goals/Objectives

Job Dekho is a dynamic website built using the MERN stack that aims to provide job seekers with a comprehensive platform to search for job openings and connect with potential employers. The MERN stack comprises MongoDB, Express, React, and Node.js, which enables the development of a robust and scalable website. Job Dekho offers a user friendly interface that enables job seekers to easily navigate the website, search for job listings, and filter them based on various criteria such as location, job type, and experience level.

DESIGN FLOW/PROCESS

Evaluation & Selection of Specifications/Features

Job Dekho is a full-stack web application developed using the MERN stack that aims to provide a comprehensive platform for job seekers and employers to connect. The MERN stack consists of MongoDB, Express, React, and Node.js, which ensures that the website is scalable, efficient, and easy to maintain. The website has a user-friendly interface that enables job seekers to search for job openings based on various criteria such as location, job type, and experience level. Job seekers can create accounts, upload resumes, and apply for job listings with ease. The website also has a dashboard that allows job seekers to track their job applications and receive notifications when new job openings that match their criteria are posted. Employers can use the website to post job openings, manage job listings, and view candidate applications. The website's administration panel allows employers to track the progress of their job postings, receive notifications when candidates apply for their job listings, and manage the recruitment process effectively

Design Constraints

The design of Job Dekho was constrained by a variety of factors, including technical limitations, budget constraints, and project scope.

The technical constraints included the choice of the MERN Stack technology stack, which required specific programming languages and tools, as well as hosting and server requirements.

Budget constraints influenced decisions such as the choice of hosting provider and the selection of premium software tools and plugins.

The project scope was also a significant design constraint, as the team had to balance the desired features and functionality with the available resources and timeline.

To address these design constraints, the team employed a variety of strategies, including prioritizing the most important features and functionality, optimizing the code and design for efficiency, and leveraging open-source software and tools to reduce costs.

I also made trade-offs and compromises in areas such as design aesthetics and user interface, while still prioritizing user experience and functionality.

Overall, the design constraints of Job Dekho were carefully considered and managed to ensure that the final product was functional, effective, and aligned with the project goals and constraints.

RESULTS ANALYSIS AND VALIDATION

Implementation of solution

The implementation and plan methodology for Job Dekho involved severalkey steps, including requirements gathering, project planning, development, testing, and deployment.

The requirements gathering phase involved defining the project goals, user needs, and desired features and functionality, which were used to create a detailed project plan.

The project plan included a timeline, milestones, and resource allocation, which were used to guide the development process and ensure that the project stayed on track.

The development phase involved the implementation of the project plan, which included the selection of appropriate technology stack, design and development of the website, creation and optimization of content, and implementation of key features such as search functionality and commenting system.

The testing phase involved rigorous testing and debugging to ensure that the website was functioning properly and meeting user needs and requirements.

Finally, the deployment phase involved launching the website and making it available to the public, as well as ongoing maintenance and support to ensure that the website remains up-to-date and responsive to user needs.

Throughout the implementation and plan methodology, agile project management principles were employed to ensure flexibility and responsiveness to changes in requirements or unforeseen issues.

Overall, the implementation and plan methodology for Job Dekho was carefully designed to ensure that the project goals and user needs were met whilealso remaining within budget and timeline constraints.

CONCLUSION AND FUTURE WORK

CONCLUSION

There is always room for improvements in any application, however good and efficient it may be. But the most important thing should be flexible to accept further modifications. Right now, we are just Building a Working MERN Stack Project which might be helpful in finding Job’s for people but in the future, we can also create a panel for Recruiters so that they can post information regarding Jobs

FUTURE WORK

Some potential future scope for a website named "Job Dekho" made with the MERN (MongoDB, Express, React, Node.js) stack are: 1. Enhanced User Experience: Continuously improving the user experience by incorporating user feedback, conducting user testing, and making iterative improvements to the website's design and functionality. This could include optimizing the website for different devices, such as mobile and tablet, to ensure a seamless experience across all platforms.

2. Personalized Job Recommendations: Implementing a recommendation engine that analyzes a user's job preferences, browsing history, and other relevant data to provide personalized job recommendations. This could help users discover relevant job opportunities tailored to their interests and qualifications.

3. Employer Features: Expanding the website's functionality to cater to employers, such as allowing employers to post job listings, manage job applications, and search for potential candidates. This could involve building an employer portal with features tailored to their needs, such as applicant tracking, resume parsing, and communication tools.

4. Job Alerts and Notifications: Implementing a system that allows users to subscribe to job alerts and notifications based on their job preferences and criteria. This could involve sending email notifications or push notifications to users whenever new job listings matching their preferences are posted on the website, helping them stay updated with the latest job opportunities.

Overall, the future scope of "Job Dekho" made with the MERN stack is vast, with numerous possibilities for expansion, customization, and improvement based on user needs and market trends. Continuous innovation, user-centric design, and staying updated with the latest technologies and trends will be key to the success of the website in the ever-evolving job market

REFERENCES

Books:

Full stack React projects 2nd Edition Shama Hoque

React Projects Roy Derks

Websites:

https://developer.mozilla.org/en-US/

https://reactjs.org/docs/getting-started.html

https://expressjs.com/

https://nodejs.org/docs/latest-v18.x/api/

https://mongoosejs.com/docs/guide.html

Faculty Guidelines:

Ms. Ruchi Talwar (Technical Trainer in GLA University)

GitHub Repository link:

https://github.com/Anshuman-Vashishtha/Mini_Project-II

Code Samples

App.js

```
import { BrowserRouter, Routes, Route } from 'react-router-dom'
import { Register, Landing, Error, ProtectedRoute } from './pages'
import {
  AllJobs,
  Profile,
  SharedLayout,
  Stats,
  AddJob,
} from './pages/dashboard'

function App() {
  return (
    <BrowserRouter>
      <Routes>
        <Route
          path='/'
          element={
            <ProtectedRoute>
              <SharedLayout />
            </ProtectedRoute>
          }
        >
          <Route index element={<Stats />} />
          <Route path='all-jobs' element={<AllJobs />} />
          <Route path='add-job' element={<AddJob />} />
          <Route path='profile' element={<Profile />} />
        </Route>
        <Route path='/register' element={<Register />} />
        <Route path='/landing' element={<Landing />} />
        <Route path='*' element={<Error />} />
```

```
      </Routes>
    </BrowserRouter>
  )
}

export default App
```

Login.jsx

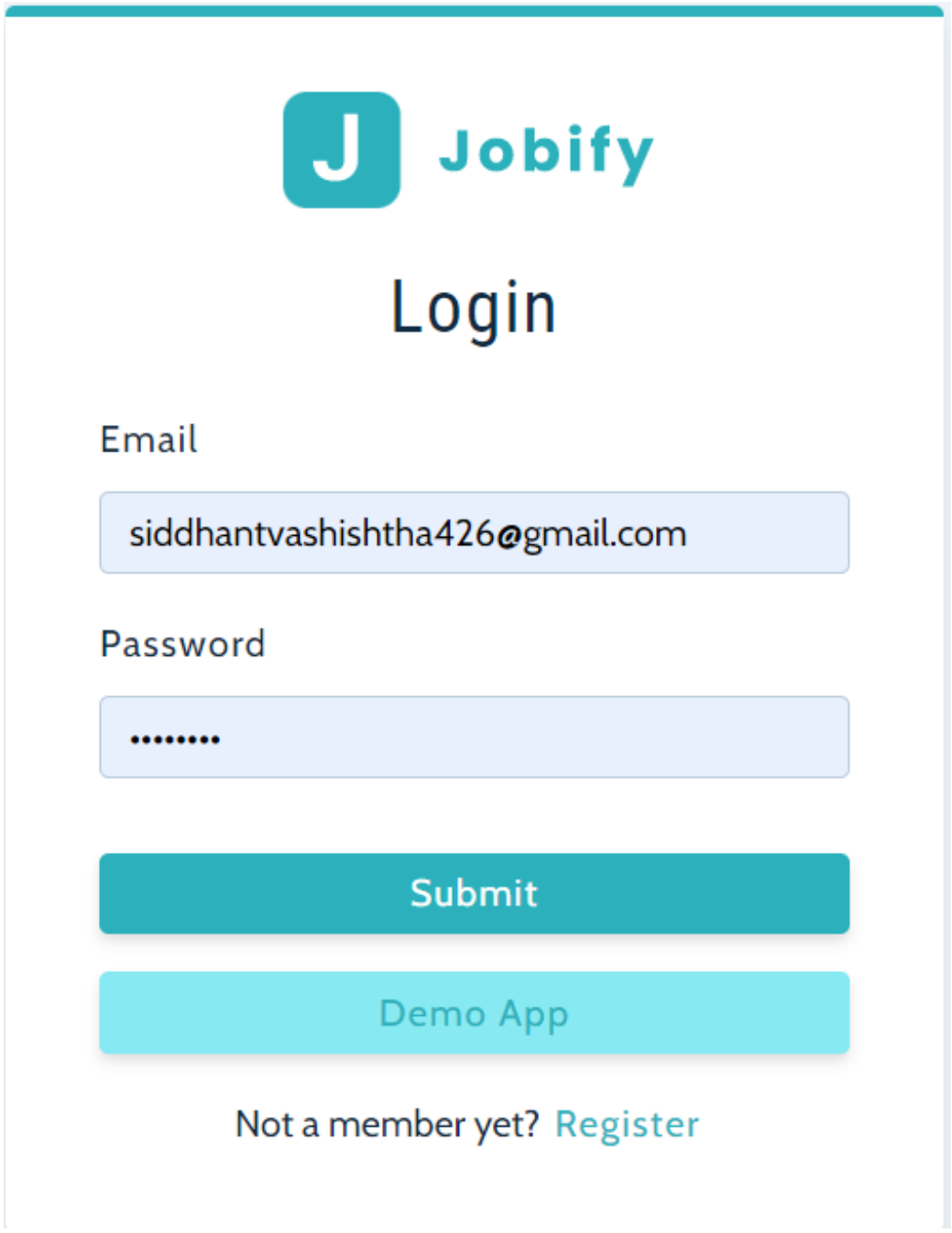
```
import { useState, useEffect } from 'react';
import { Logo, FormRow, Alert } from '../components';
import Wrapper from '../assets/wrappers/RegisterPage';
import { useAppContext } from '../context/appContext';
import { useNavigate } from 'react-router-dom';
const initialState = {
  name: '',
  email: '',
  password: '',
  isMember: true,
};

const Register = () => {
  const navigate = useNavigate();
  const [values, setValues] = useState(initialState);
  const { user, isLoading, showAlert, displayAlert, setupUser } =
    useAppContext();

  const toggleMember = () => {
    setValues({ ...values, isMember: !values.isMember });
  };

  const handleChange = (e) => {
    setValues({ ...values, [e.target.name]: e.target.value });
  };
```

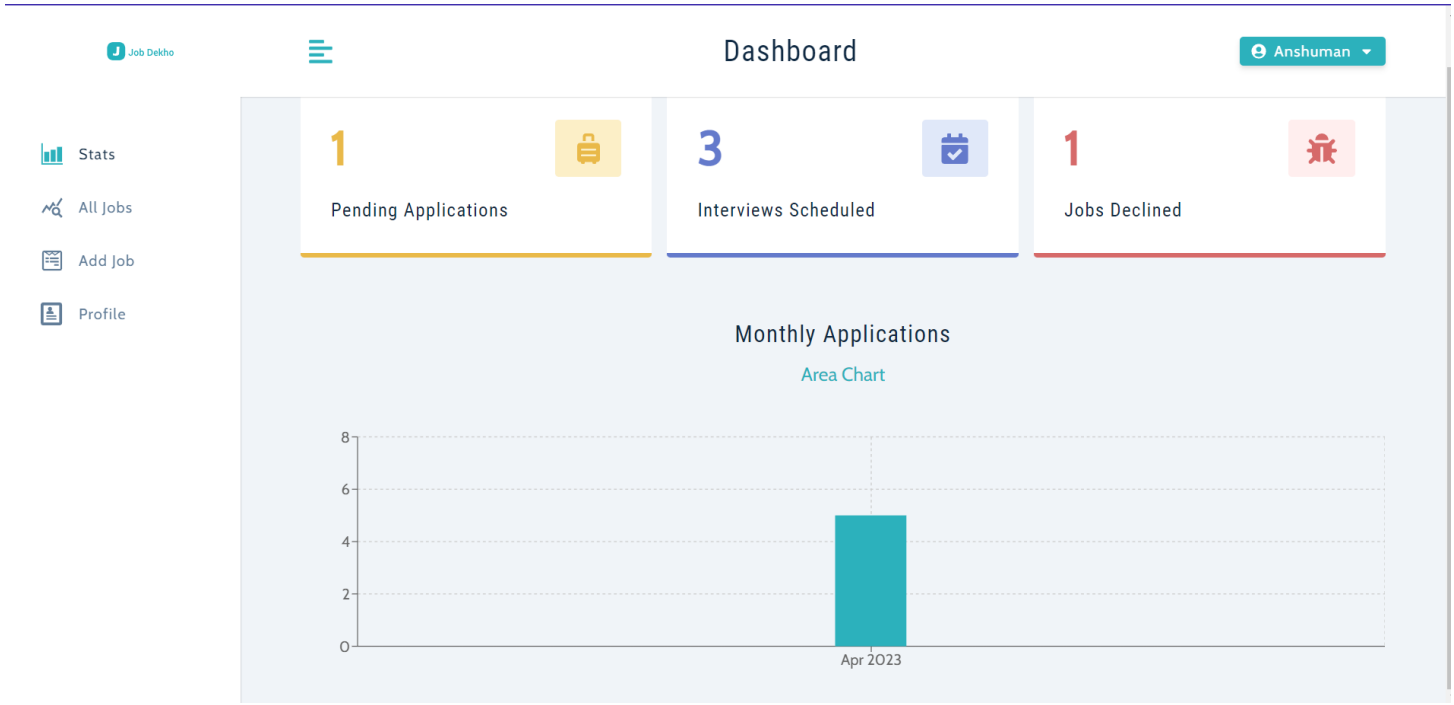
```
const onSubmit = (e) => {
  e.preventDefault();
  const { name, email, password, isMember } = values;
  if (!email || !password || (!isMember && !name)) {
    displayAlert();
    return;
  }
  const currentUser = { name, email, password };
  if (isMember) {
    setupUser({
      currentUser,
      endPoint: 'login',
      alertText: 'Login Successful! Redirecting...',
    });
  } else {
    setupUser({
      currentUser,
      endPoint: 'register',
      alertText: 'User Created! Redirecting...',
    });
  }
};

useEffect(() => {
  if (user) {
    setTimeout(() => {
      navigate('/');
    }, 3000);
  }
}, [user, navigate]);

return (
  <Wrapper className='full-page'>
    <form className='form' onSubmit={onSubmit}>
```

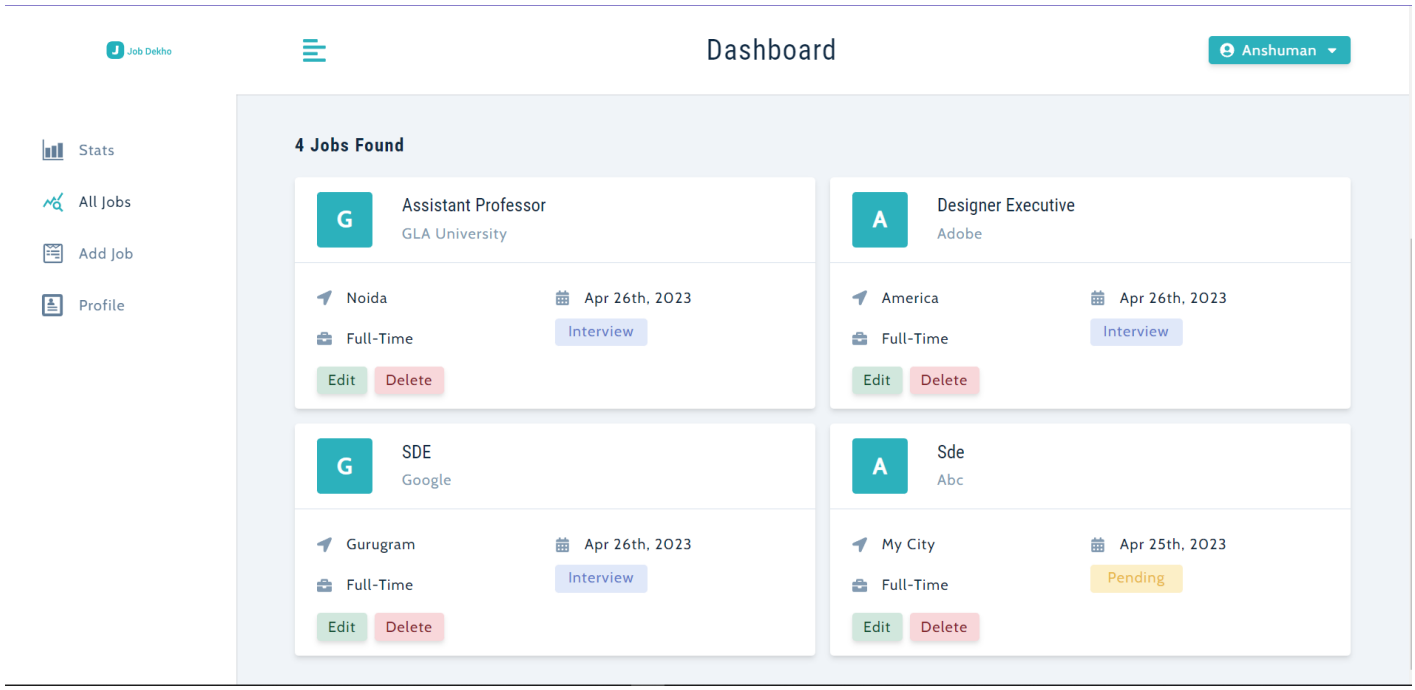
```
        <Logo />
        <h3>{values.isMember ? 'Login' : 'Register'}</h3>
        {showAlert && <Alert />}
        {/* name input */}
        {!values.isMember && (
          <FormRow
            type='text'
            name='name'
            value={values.name}
            handleChange={handleChange}
          />
        )}

        {/* email input */}
        <FormRow
          type='email'
          name='email'
          value={values.email}
          handleChange={handleChange}
        />
        {/* password input */}
        <FormRow
          type='password'
          name='password'
          value={values.password}
          handleChange={handleChange}
        />
        <button type='submit' className='btn btn-block'
disabled={isLoading}>
          submit
        </button>
        <button
          type='button'
          className='btn btn-block btn-hipster'
```

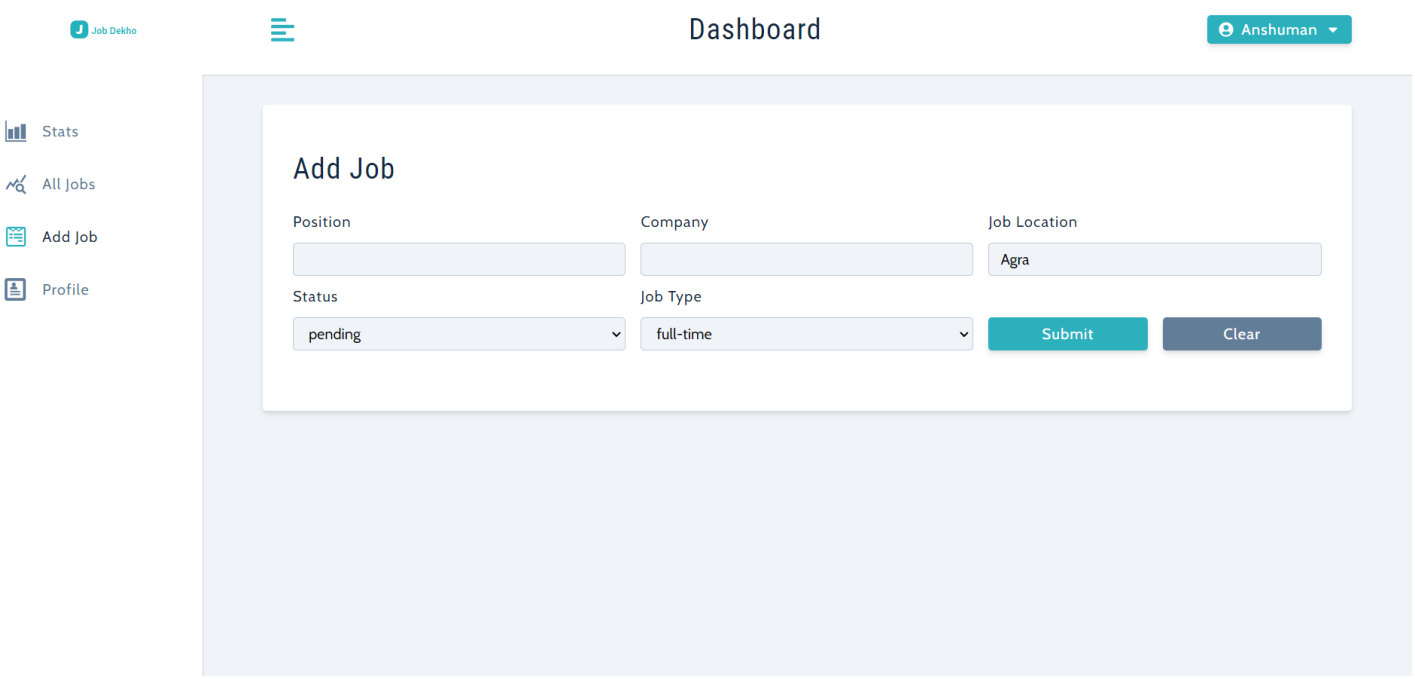
```
          disabled={isLoading}
          onClick={() => {
            setupUser({
              currentUser: { email: 'testUser@test.com', password:
'secret' },
              endPoint: 'login',
              alertText: 'Login Successful! Redirecting...',
            });
          }}
        >
          {isLoading ? 'loading...' : 'demo app'}
        </button>
        <p>
          {values.isMember ? 'Not a member yet?' : 'Already a
member?'}
          <button type='button' onClick={toggleMember}
className='member-btn'>
            {values.isMember ? 'Register' : 'Login'}
          </button>
        </p>
      </form>
    </Wrapper>
  );
};
export default Register;
```

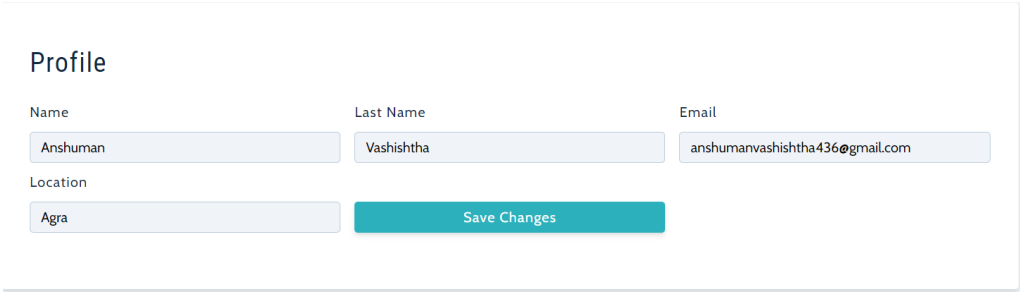
Some photographs from our website




Job Dekho
Dashboard
Anshuman
Stats
All Jobs
Add Job
Profile
1
Pending Applications
3
Interviews Scheduled
1
Jobs Declined
Monthly Applications
Area Chart
8
6
4
2
0
Apr 2023


Job Dekho
Dashboard
Anshuman
Stats
All Jobs
Add Job
Profile
4 Jobs Found
G
Assistant Professor
GLA University
Noida
Apr 26th, 2023
Full-Time
Interview
Edit
Delete
A
Designer Executive
Adobe
America
Apr 26th, 2023
Full-Time
Interview
Edit
Delete
G
SDE
Google
Gurugram
Apr 26th, 2023
Full-Time
Interview
Edit
Delete
A
Sde
Abc
My City
Apr 25th, 2023
Full-Time
Pending
Edit
Delete


Job Dekho
Dashboard
Anshuman
Stats
All Jobs
Add Job
Profile
Add Job
Position
Company
Job Location
Agra
Status
pending
Job Type
full-time
Submit
Clear


Profile
Name
Anshuman
Last Name
Vashishtha
Email
anshumanvashishtha436@gmail.com
Location
Agra
Save Changes