

A Field Project Report on  
**INTERVIEW GUIDE**

Submitted to the

**Department of Information Technology**

in partial fulfillment of the requirements for the completion of course  
FIELD PROJECT (22SD5IT202)

**BACHELOR OF TECHNOLOGY**

**IN**

**INFORMATION TECHNOLOGY**

Submitted by

M.NIKHIL	22071A12G4
N.RITHWIK	22071A12G6
N.TEJASHWINI	22071A12G9
N.SAI MAHESH	22071A12H0

Under the guidance of

**Mr. Ravi Tene & Mr. S Murali Mohan**  
**(Course Instructors)**

Assistant Professor, Department of IT, VNRVJIT



**DEPARTMENT OF INFORMATION TECHNOLOGY**

**VALLURUPALLI NAGESWARA RAO VIGNANA**  
**JYOTHI INSTITUTE OF ENGINEERING &**  
**TECHNOLOGY**

An Autonomous Institute, NAAC Accredited with 'A++' Grade, NBA

Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS,  
India

**JUNE 2024**

**VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI  
INSTITUTE OF ENGINEERING AND TECHNOLOGY**

An Autonomous Institute, NAAC Accredited with 'A++' Grade, NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B. Tech Courses, Approved by AICTE, New Delhi, Affiliated to JNTUH, Recognized as "College with Potential for Excellence" by UGC, ISO 9001:2015 Certified, QS I GUAGE Diamond Rated Vignana Jyothi Nagar, Pragathi Nagar, Nizampet(SO), Hyderabad-500090, TS, India

**DEPARTMENT OF INFORMATION TECHNOLOGY**



**CERTIFICATE**

This is to certify that the project report entitled "**Interview Guide**" is a bonafide work done under our supervision and is being submitted by **M.NIKHIL (22071A12G4), N.RITHWIK(22071A12G6), N.TEJASHWINI(22071A12G9), N.SAI MAHESH (22071A12H0)** in partial fulfilment for the award of the degree of **Bachelor of Technology** in Information Technology, of the VNRVJIET, Hyderabad during the academic year 2023-2024.

Course Instructor Name

**Ravi Tene & S. Murali Mohan**

Assistant Professor, IT

**Dr D Srinvasa Rao**

Associate Professor & HOD, IT

**Field Projects Reviewer**

# VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institute, NAAC Accredited with 'A++' Grade,  
Vignana Jyothi Nagar, Pragathi Nagar, Nizampet(SO), Hyderabad-500090, TS, India

## DEPARTMENT OF INFORMATION TECHNOLOGY



## DECLARATION

We declare that the Field project work entitled “**Interview Guide**” submitted in the Department of Information Technology, Vallurupalli Nageswara Rao Vignana Jyothi Institute of Engineering and Technology, Hyderabad, in partial fulfilment of the requirement for the award of the degree of **Bachelor of Technology in Information Technology** is a bonafide record of our own work carried out under the supervision of **Ravi Tene & S. Murali Mohan , Assistant Professor, Department of IT, VNRVJiet**. Also, we declare that the matter embodied in this thesis has not been submitted by us in full or in any part thereof for the award of any degree/diploma of any other institution or university previously.

Place: Hyderabad.

**M.NIKHIL**

22071A12G4

**N.RITHWIK**

22071A12G6

**N.TEJASHWINI**

22071A12G9

**N.SAI MAHESH**

22071A12H0

## ACKNOWLEDGEMENT

We express our deep sense of gratitude to our beloved President, **Sri. D. Suresh Babu**, VNR Vignana Jyothi Institute of Engineering & Technology for the valuable guidance and for permitting us to carry out this project.

With immense pleasure, we record our deep sense of gratitude to our beloved Principal, **Dr. C.D Naidu**, for permitting us to carry out this project.

We express our deep sense of gratitude to our beloved HOD, **Dr. SRINIVASA RAO DAMMAVALAM**, Associate Professor, Department of Information Technology, VNR Vignana Jyothi Institute of Engineering & Technology, Hyderabad-500090 for the valuable guidance and suggestions, keen interest and through encouragement extended throughout the period of project work.

We take immense pleasure to express our deep sense of gratitude to our beloved Guide, **Ravi Tene and S Murali Mohan**, Assistant Professor in Information Technology, VNR Vignana Jyothi Institute of Engineering & Technology, Hyderabad, for his/her valuable suggestions and rare insights, for constant source of encouragement and inspiration throughout my project work.

We express our thanks to all those who contributed for the successful completion of our project work.

M.NIKHIL

22071A12G4

N.RITHWIK

22071A12G6

N.TEJASHWINI

22071A12G9

N.SAI MAHESH

22071A12H0

## **ABSTRACT**

The "Interview Guide" project is a web-based platform designed to assist job candidates in preparing effectively for interviews. It offers a comprehensive suite of tools and resources, featuring an intuitive user interface that includes modules for login, viewing, and answering interview questions. The authentication service ensures secure user access, while the question management module provides a diverse set of interview questions and tips from various companies and industries.

Candidates can browse and practice answering these questions, gaining familiarity with the types of queries they might encounter in real interviews. Additionally, the platform includes a progress tracking service that monitors users' improvements over time, offering detailed reports and insights to help them identify strengths and areas for further practice. All user data, questions, and progress records are securely stored in a robust database.

By combining a vast repository of interview questions with detailed feedback and progress tracking, the Interview Guide creates a supportive environment for job seekers. It aims to enhance candidates' confidence and readiness, ensuring they are well-prepared for actual interviews and improving their chances of success in securing job offers.

## TABLE OF CONTENTS

S No	Contents	Page No
1	Abstract	1
2	Table of Contents	2
3	Introduction	3
4	Source code	4-23
5	Test Cases/Output	24-26
6	Conclusion	27
7	References	28

# **1. INTRODUCTION**

## **1.1 PROBLEM DEFINITION**

In the current job market, candidates often struggle to find reliable and comprehensive resources for interview preparation. The “Career Insight” project seeks to resolve this by developing a dynamic platform that aggregates a wide range of interview questions, insightful tips, and strategic guidance across various industries. This platform will empower job seekers to prepare effectively, gain confidence, and improve their chances of success in interviews

## **1.2 OBJECTIVE**

The purpose of the “Career Insight” project is to create a comprehensive platform that assists users in preparing for interviews by providing relevant resources, practice tools, and personalized feedback to enhance their performance and confidence.

## 2. SOURCE CODE

### App.js

```
import React from
'react'; import
'bootstrap/dist/css/boot
strap.min.css'
import { BrowserRouter, Routes, Route }
from 'react-router-dom';
import Navbar from
'./components/navbar/Navbar';
import Home from './pages/Home';
import Companies from './pages/Companies';
import About from './pages/About';
// import Loginsignup from
'./pages/LoginSignup'; import
Latestjobs from
'./pages/Latestjobs'; import
Questions from
'./pages/Questions'; import
JobDetails from
'./pages/JobDetails'; import
Login from './pages/Login';
import Signup from
'./pages/Signup';

function App() {
  return (
    <BrowserRouter>
      <Navbar />
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/companies" element={<Companies />} />
```



```

    <Route path="/about" element={<About />} />
    <Route path="/login" element={<Login/>} />
    <Route path="/signup" element={<Signup/>} />
    <Route path="/latestjobs" element={<Latestjobs />} />
    <Route path="/job-details/:jobId" element={<JobDetails />} />
    <Route path="/questions/:companyId"
    element={<Questions />} /> </Routes>
  </BrowserRouter>
);}
export default App;

```

## Index.html

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta name="description"
    content="Web site created
    using create-react-app"
    />
    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
    <!--manifest.json provides metadata used when your web app is installed on
    a user's mobile device or desktop. See
    https://developers.google.com/web/fundamentals/web-app-
    manifest/
    -->
    <link rel="manifest"
    href="%PUBLIC_URL%/manifest.json" /> <!--
  -

```

Notice the use of %PUBLIC\_URL% in the tags above.

It will be replaced with the URL of the `public` folder during the build. Only files inside the `public` folder can be referenced from the HTML.

Unlike `"/favicon.ico"` or `"favicon.ico"`, `"%PUBLIC_URL%/favicon.ico"` will work correctly both with client-side routing and a non-root public URL.

Learn how to configure a non-root public URL by running ``npm run build``.

-->

`<title>React App</title>`

`</head>`

`<body>`

`<noscript>You need to enable JavaScript to run this app.</noscript>`

`<div id="root"></div>`

`<!--`

This HTML file is a template.

If you open it directly in the browser, you will see an empty page.

You can add webfonts, meta tags, or analytics to this file.

The build step will place the bundled scripts into the `<body>` tag.

To begin the development, run ``npm start`` or ``yarn start``.

To create a production bundle, use ``npm run`

`build`` or ``yarn build``. -->

`</body>`

`</html>`

## Server.js

```
const express = require('express'); const mongoose = require('mongoose'); const cors
= require('cors'); const questionsRouter = require('./routes/questions'); const
usersRouter = require('./routes/users'); const path=require('path')
```

```

const
app =
express
s();
const
PORT
=
5000;

app.use(express.static(path.join(__dirname, '../client/build')))

mongoose.connect('mongodb://localhost:27017/interview-guide', {
  useNewUrlParser: true,
  useUnifiedTopology:
    true,
});

const db = mongoose.connection;
db.on('error', (error) =>
console.error(error)); db.once('open', ()
=> console.log('Connected to
MongoDB'));

app.use(cors()); app.use(express.json());

app.use('/api', questionsRouter); app.use('/api/users',
usersRouter); app.listen(PORT, () =>
console.log(` Server running on port ${PORT} `));

```

### **About.js**

```

import React, { useState } from 'react';
import '../pages/css/About.css';
import RateUsModal from '../pages/RateUsModal'
const About = () => {

```

```

const [isModalOpen, setIsModalOpen] =
useState(false);
const openModal = () => {
  setIsModalOpen(true);
};
const closeModal = () => {
  setIsModalOpen(false);
};
return (
  <div className="about-page">
    <header className="about-header">
      <h1>About Interview Guide</h1>
      <p>Your ultimate resource for interview
preparation and job opportunities.</p>
    </header>

    <section className="about-section">
      <h2>Our Mission</h2>
      <p> At Interview Guide, our mission is to
empower job seekers with the knowledge and tools
they need to succeed in their interviews. We strive
to provide the most comprehensive and up-to-date
collection of technical and non-technical interview
questions from top companies, along with detailed
job descriptions and opportunities.
      </p>
    </section>
    <section className="about-section">

      <h2>What We Offer</h2>

      <ul>

```

**Technical and Non-Technical Questions:** A curated collection of interview questions from various companies, helping you prepare for any type of interview.

**Latest Job Opportunities:** Detailed information about the latest job openings, including job descriptions, salaries, titles, roles, and places of work.

**Our Values**

- Integrity
- Innovation
- Customer Satisfaction
- Community Engagement

**Contact Us**

Email: [info@interviewguide.com](mailto:info@interviewguide.com)

Phone: +123-456-7890

Address: Bachupally, Hyderabad

**Follow Us**

Stay connected with us through social media.

[\*fab fa-facebook-f\*](https://facebook.com)

```

      <a href="https://twitter.com" target="_blank"
rel="noopener noreferrer" className="social-
icon"><i className="fab fa-twitter"></i></a>
      <a href="https://instagram.com"
target="_blank" rel="noopener noreferrer"
className="social-icon"><i className="fab fa-
instagram"></i></a>
    </div>
  </section>
  <div>
    <button className='btn btn-primary text-
danger' onClick={openModal}>Rate Us</button>
  </div>
  <RateUsModal isOpen={isModalOpen}
onClose={closeModal} />
</div>
);
};
export default About;

```

## Companies.js

```

import React from 'react';
import { useNavigate } from 'react-router-dom';
import AllCompanies from
'../components/assets/AllCompanies';
import './css/Companies.css';

const Companies = () => {
  const navigate = useNavigate();

  const handleButtonClick = (companyId) => {
    navigate(`/questions/${companyId}`);
  };
};

```

```

return (
  <div className='company'>
    <hr/>
    <p>Know Previously Asked Questions
Here</p>
    <hr/>
    <div className='card-container'>
      {AllCompanies.map((item) => (
        <div className='card' key={item.id}>
          <img src={item.image} alt={item.name}
className='card-image' />
          <div className='card-content'>
            <h2 className='card-
title'>{item.name}</h2>
            <p className='card-text'><b>Role</b>:
{item.role}</p>
          </div>
          <div className='card-footer'>
            <p className='card-salary'>Salary:
₹{item.salary}</p>
            <button onClick={() =>
handleButtonClick(item.id)}><b>Click
Here</b></button>
          </div>
        </div>
      )
    )}
    </div>
  </div>
);
};

export default Companies;

```

## JobDetails.js

```
import React from 'react';
import { useParams } from 'react-router-dom';
import AllJobs from '../components/assets/AllJobs';
// Adjust this path if necessary
import '../pages/css/JobDetails.css'; // Import the
CSS file

const JobDetails = () => {
  const { jobId } = useParams();
  const job = AllJobs.find(job => job.id ===
parseInt(jobId));

  if (!job) {
    return <div className="job-details">Job not
found</div>;
  }

  return (
    <div className='job-details'>
      <h2 className='job-title'>{job.jobTitle}</h2>
      <p><span className='side-
heading'>Company:</span>
{job.companyName}</p>
      <p><span className='side-heading'>Job
Description:</span> {job.jobDescription}</p>
      <p><span className='side-
heading'>Salary:</span> {job.jobSalary}</p>
      <p><span className='side-
heading'>Requirements:</span>
{job.requirements}</p>
```



```

    <p><span className='side-
heading'>Qualifications:</span>
{job.qualifications}</p>
    <p><span className='side-heading'>Place of
Work:</span> {job.placeOfWork}</p>
  </div>
);
};

```

```
export default JobDetails;
```

### LatestJobs.js

```

import React from 'react';
import { Link } from 'react-router-dom';
import AllCompanies from
'../components/assets/AllCompanies';
import '../pages/css/Companies.css'; // Import the
CSS file

```

```

const Latestjobs = () => {
  return (
    <div className='company'>
      <hr />
      <p>Know Previously Asked Questions
Here</p>
      <hr />
      <div className='card-container'>
        {AllCompanies.map((item) => (
          <div className='card' key={item.id}>
            <img src={item.image} alt={item.name}
className='card-image' />
            <div className='card-content'>

```

```

        <h2 className='card-
title'>{item.name}</h2>
        <p className='card-text'><b>Role</b>:
{item.role}</p>
    </div>
    <div className='card-footer'>
        <p className='card-salary'>Salary:
₹ {item.salary}</p>
        <Link to={` /job-
details/${item.id} `}><button><b>Know
More</b></button></Link>
    </div>
</div>
    )}
</div>
</div>
);
};

```

```
export default Latestjobs;
```

## Login.js

```

import React, { useState } from 'react';
import { Link, useNavigate } from 'react-router-
dom';
import axios from 'axios';
import './css/Login.css'; // Update the path to your
CSS file

```

```

function Login() {
    const [email, setEmail] = useState("");
    const [password, setPassword] = useState("");
    const [message, setMessage] = useState("");

```

```

const navigate = useNavigate();

const handleSubmit = async (e) => {
  e.preventDefault();
  try {
    const response = await
axios.post('http://localhost:5000/api/users/login', {
email, password });
    setMessage(response.data.message);
    if (response.data.redirectTo) {
      navigate('/'); // Redirect to home page
    }
  } catch (error) {
    console.error('Error during login:', error);
    setMessage('Error during login: ' +
(error.response?.data?.message || error.message));
  }
};

return (
  <div className="login-container">
    <div className="login-card">
      <h2>Login</h2>
      <form onSubmit={handleSubmit}>
        <input
          type="email"
          placeholder="Email"
          id="email"
          value={email}
          onChange={(e) =>
setEmail(e.target.value)}
          required
        />
        <input

```

```

        type="password"
        placeholder="Password"
        id="password"
        value={password}
        onChange={(e) =>
setPassword(e.target.value)}
        required
      />
      <button type="submit">Login</button>
    </form>
    {message && <p
className="error">{message}</p>}
    <p>Don't have an account? <Link
to="/signup">Sign Up</Link></p>
  </div>
</div>
);
}

```

```
export default Login;
```

### Questions.js

```

import React, { useEffect, useState } from 'react';
import { useParams } from 'react-router-dom';
import axios from 'axios';
import '../pages/css/Companies.css';
import '../pages/css/Questions.css';
// import Item from '../components/item/Item'
// import AllCompanies from
'../components/assets/AllCompanies'

const Questions = () => {
  const { companyId } = useParams();

```

```

const [questions, setQuestions] = useState([]);

useEffect(() => {
  const fetchQuestions = async () => {
    try {
      const response = await
axios.get(`http://localhost:5000/api/questions/${companyId}`);
      setQuestions(response.data);
    } catch (error) {
      console.error('Error fetching questions:', error);
    }
  };

  fetchQuestions();
}, [companyId]);

return (
  <div className="questions-container">

    <h1>Check the latest questions</h1>
    <ul className="question-list">
      {questions.map((question, index) => (
        <li key={index}>
          {question.question}
          {question.link && (
            <a href={question.link} target="_blank"
rel="noopener noreferrer">
              View Question
            </a>
          )}
        </li>
      ))}
    </ul>
  )}

```

```

    </div>
  );
};

export default Questions;

```

### **RateUsModel.js**

```

import React, { useState } from 'react';
import '../pages/css/RateUsModal.css';

const RateUsModal = ({ isOpen, onClose }) => {
  const [rating, setRating] = useState(0);
  const [showFeedbackForm,
setShowFeedbackForm] = useState(false);
  const [feedback, setFeedback] = useState("");

  const handleStarClick = (index) => {
    setRating(index + 1);
  };

  const handleRatingSubmit = () => {
    if (rating > 0) {
      setShowFeedbackForm(true);
    }
  };

  const handleFeedbackChange = (event) => {
    setFeedback(event.target.value);
  };

  const handleFeedbackSubmit = () => {
    // Handle the feedback submission logic here
    console.log('Rating:', rating);
  };

```

```

    console.log('Feedback:', feedback);

    // Optionally, you can clear the feedback form or
close the modal after submission
    setShowFeedbackForm(false);
    onClose();
  };

  if (!isOpen) return null;

  return (
    <div className="modal-overlay">
      <div className="modal-content">
        <h2>Rate Us</h2>
        <div className="stars">
          {[...Array(5)].map((_, index) => (
            <i
              key={index}
              className={`fas fa-star ${index < rating ?
'active' : ''}`}
              onClick={() => handleStarClick(index)}
            ></i>
          ))}
        </div>
        {!showFeedbackForm ? (
          <button onClick={handleRatingSubmit}
className="btn btn-primary">
            Submit Rating
          </button>
        ) : (
          <div className="feedback-form">
            <textarea
              value={feedback}
              onChange={handleFeedbackChange}

```

```

        placeholder="Please leave your feedback
here..."
    ></textarea>
    <button onClick={handleFeedbackSubmit}
className="btn btn-primary mb-5">
        Submit Feedback
    </button>
</div>
)}
<button onClick={onClose} className="btn
btn-secondary mt-3">
    Close
</button>
</div>
</div>
);
};

```

```
export default RateUsModal;
```

### Signup.js

```

import React, { useState } from 'react';
import { Link, useNavigate } from 'react-router-
dom';
import axios from 'axios';
import './css/Signup.css';

function Signup() {
    const [name, setName] = useState("");
    const [email, setEmail] = useState("");
    const [password, setPassword] = useState("");
    const [message, setMessage] = useState("");
    const navigate = useNavigate();

```



```

const handleSubmit = async (e) => {
  e.preventDefault();
  try {
    const response = await
axios.post('http://localhost:5000/api/users/signup', {
name, email, password });
    setMessage(response.data.message);
    navigate('/login'); // Redirect to login page
after successful signup
  } catch (error) {
    console.error('Error during registration:',
error);
    setMessage('Error during registration: ' +
(error.response?.data?.message || error.message));
  }
};

return (
  <div className="signup-container">
    <div className="signup-card">
      <h2>Sign Up</h2>
      <form onSubmit={handleSubmit}>
        <input
          type="text"
          placeholder="Name"
          id="name"
          value={name}
          onChange={(e) =>
setName(e.target.value)}
          required
        />
        <input
          type="email"

```

```

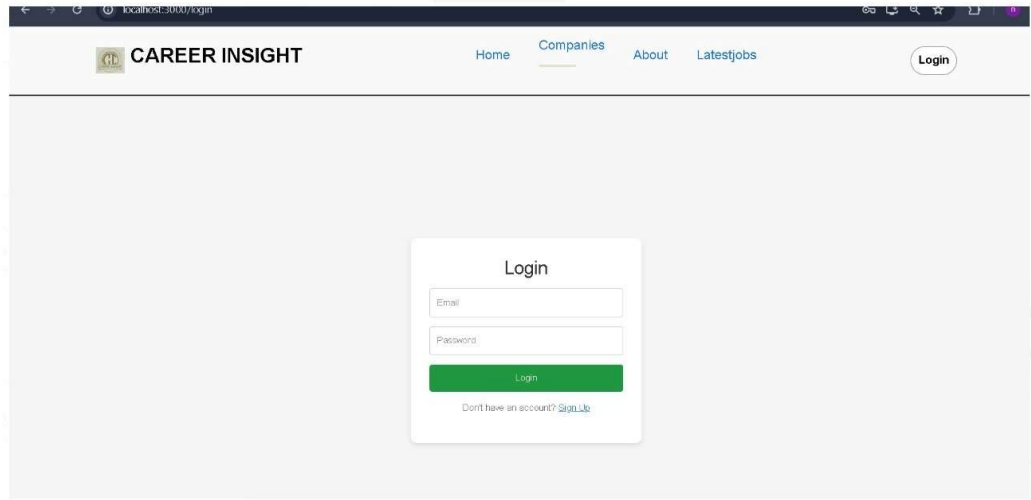
        placeholder="Email"
        id="email"
        value={email}
        onChange={(e) =>
setEmail(e.target.value)}
        required
    />
    <input
        type="password"
        placeholder="Password"
        id="password"
        value={password}
        onChange={(e) =>
setPassword(e.target.value)}
        required
    />
    <button type="submit">Sign
Up</button>
</form>
    {message && <p
className="error">{message}</p>}
    <p>Already have an account? <Link
to="/login">Login</Link></p>
    </div>
    </div>
    );
}

export default Signup;

```

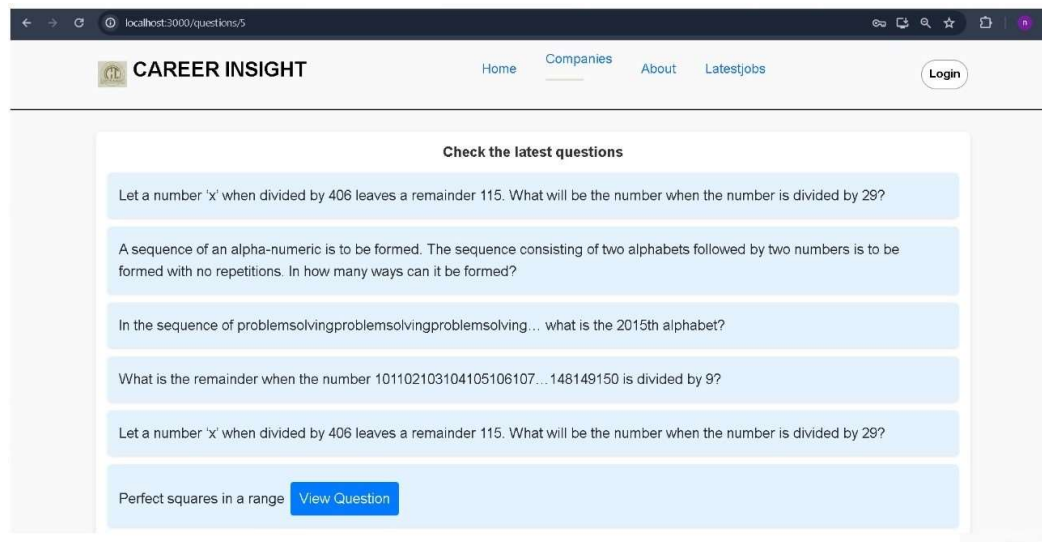
## 3. TEST CASES/ OUTPUT

### 3.1 Test case 1:



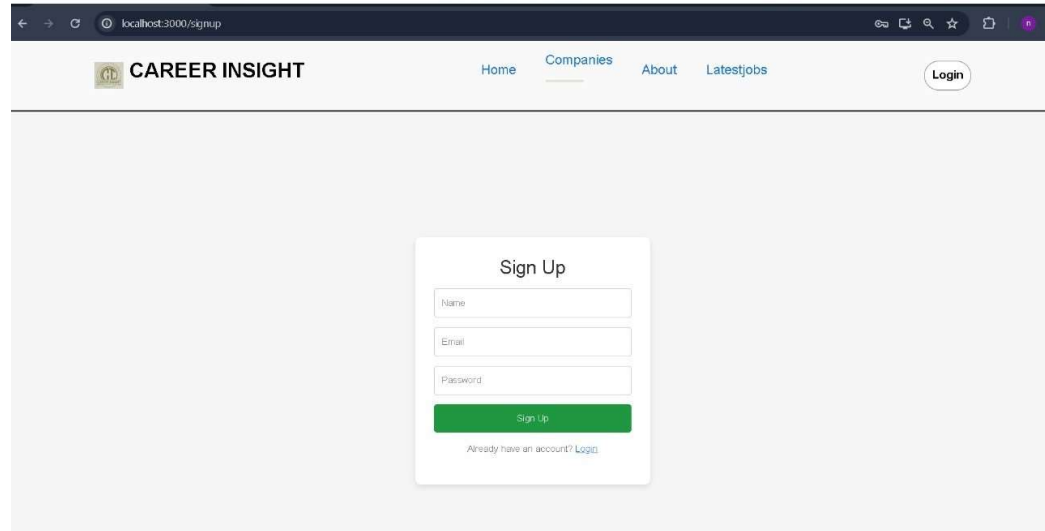
The screenshot shows a web browser at the URL `localhost:3000/login`. The page features a header with the "CAREER INSIGHT" logo and navigation links for "Home", "Companies", "About", and "Latestjobs". A "Login" button is in the top right. The main content area contains a centered "Login" form with input fields for "Email" and "Password", a green "Login" button, and a link for "Don't have an account? Sign Up".

### 3.2 Test case 2:



The screenshot shows a web browser at the URL `localhost:3000/questions/5`. The page features the same header as the login page. The main content area is titled "Check the latest questions" and displays a list of six questions in light blue boxes. The first five questions are identical: "Let a number 'x' when divided by 406 leaves a remainder 115. What will be the number when the number is divided by 29?", "A sequence of an alpha-numeric is to be formed. The sequence consisting of two alphabets followed by two numbers is to be formed with no repetitions. In how many ways can it be formed?", "In the sequence of problemsolvingproblemsolvingproblemsolving... what is the 2015th alphabet?", "What is the remainder when the number 101102103104105106107...148149150 is divided by 9?", and "Let a number 'x' when divided by 406 leaves a remainder 115. What will be the number when the number is divided by 29?". The sixth question is "Perfect squares in a range" followed by a blue "View Question" button.

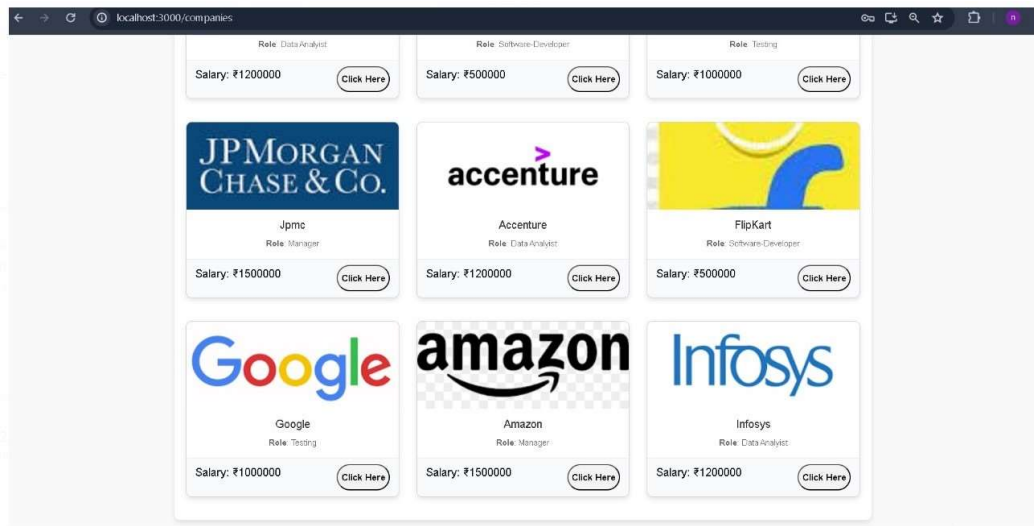
### 3.3 Test case 3:



A screenshot of a web browser showing the 'Sign Up' page of 'CAREER INSIGHT'. The browser's address bar shows 'localhost:3000/signup'. The page has a dark blue header with the site logo and navigation links: 'Home', 'Companies', 'About', 'Latestjobs', and a 'Login' button. The main content area is light gray and features a white 'Sign Up' form in the center. The form includes input fields for 'Name', 'Email', and 'Password', a green 'Sign Up' button, and a link for users who 'Already have an account? Login'.

Field	Type
Name	Text Input
Email	Text Input
Password	Text Input
Sign Up	Submit Button
Already have an account? Login	Link

### 3.4 Test case 4:



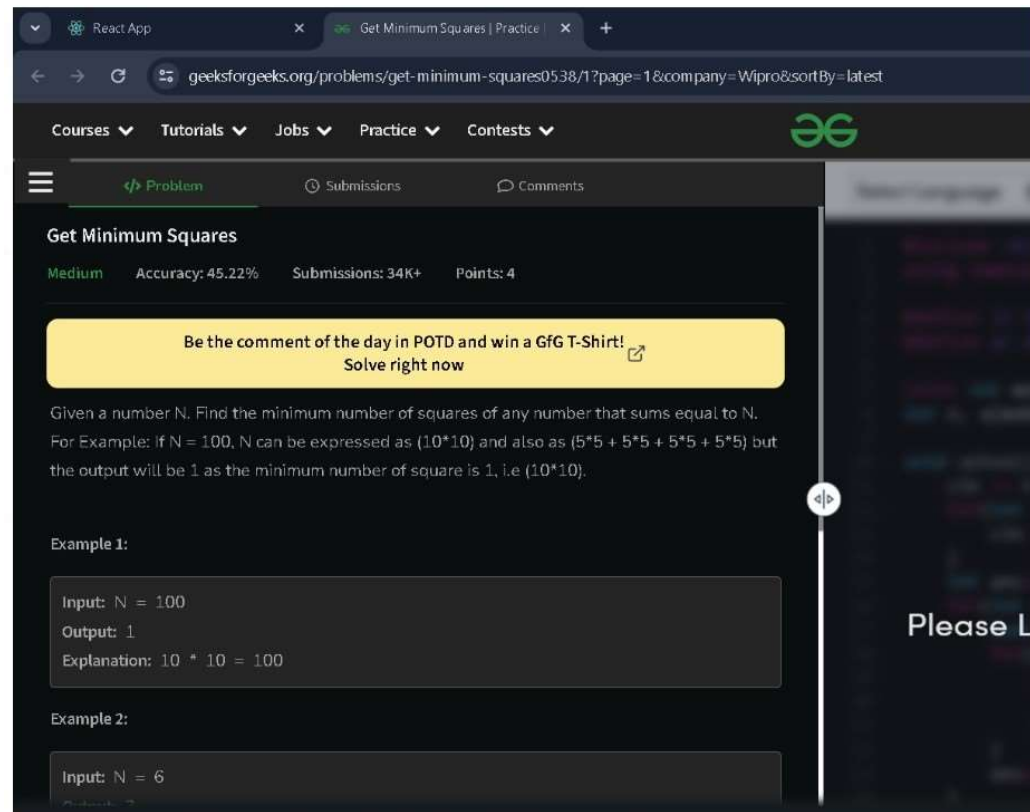
A screenshot of a web browser showing the 'Companies' page of 'CAREER INSIGHT'. The browser's address bar shows 'localhost:3000/companies'. The page displays a grid of nine company cards, each with a logo, company name, role, salary, and a 'Click Here' button. The companies are arranged in a 3x3 grid.

Company	Role	Salary	Action
JPMC	Manager	₹1500000	Click Here
Accenture	Data Analyst	₹1200000	Click Here
FlipKart	Software Developer	₹500000	Click Here
Google	Testing	₹1000000	Click Here
Amazon	Manager	₹1500000	Click Here
Infosys	Data Analyst	₹1200000	Click Here

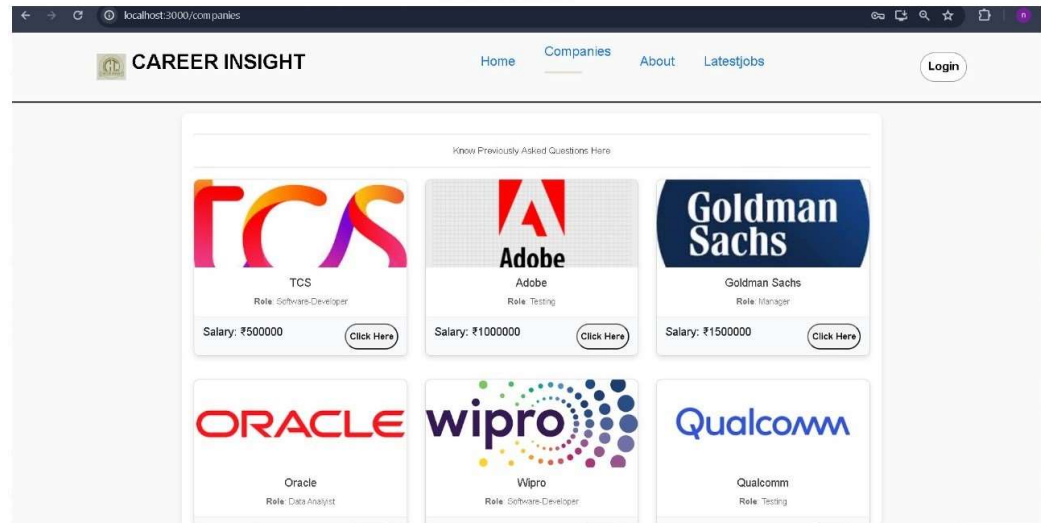
### 3.5 Test case 5:



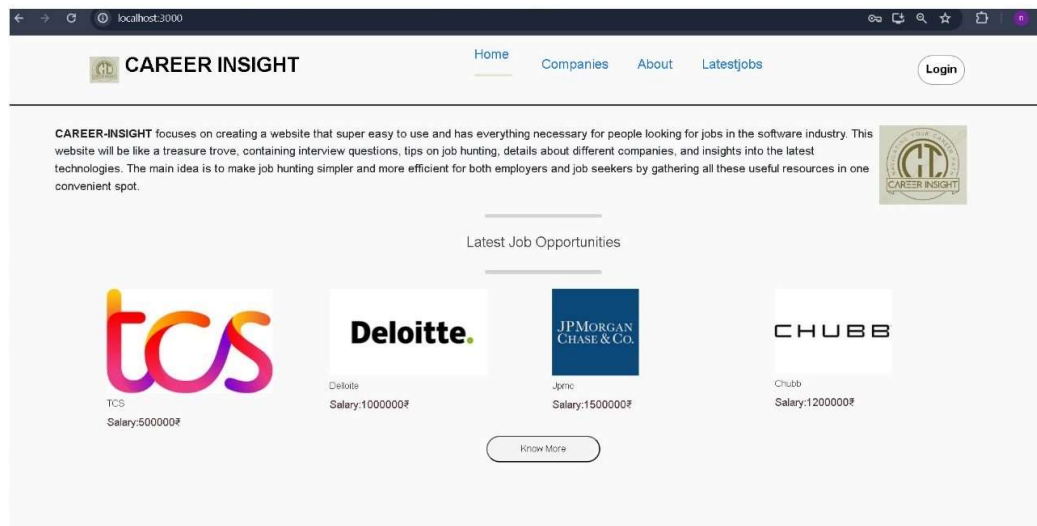
### 3.6 Test case 6:



### 3.7 Test case 7:



### 3.8 Test case 7:



# CONCLUSION

Career Insight/Interview-Guide serves as a comprehensive resource designed to empower job seekers by providing a centralized repository of interview questions and insightful answers. This project bridges the gap between aspirants and their dream jobs by eliminating the need for personal consultations and expensive coaching. Users can access a vast array of commonly asked interview questions, tailored to various industries and roles, all curated to reflect the latest trends and employer expectations. This guide not only covers technical and role-specific queries but also delves into behavioral and situational questions, ensuring

# REFERENCES

- CSS Custom Properties (Variables): MDN Web Docs. Retrieved from

[https://developer.mozilla.org/enUS/docs/Web/CSS/Using\\_CSS\\_custom\\_properties](https://developer.mozilla.org/enUS/docs/Web/CSS/Using_CSS_custom_properties)

- Flexbox Guide: CSS-Tricks. Retrieved from:

<https://csstricks.com/snippets/css/aguide-to-flexbox/>

- Responsive Web Design Basics: Google Developers. Retrieved from:

<https://developers.google.com/web/fundamentals/design-and-ux/responsive/>

- HTML and CSS Best Practices: W3C. Retrieved from:

<https://www.w3.org/standards/webdesign/html>