**Deliverable 4 Job Hunt/Hub - Connect**

**Development Phase 2**

**Enhanced Functionality**

**Requirements:**

**Front End Enhancements (Critical Features):**

* **Enhanced Job Search Functionality:**

Improve the existing job search functionality by adding basic search filters, such as job title and location.

Display search results in a clear and organized manner, allowing users to easily browse through job listings.

* **Job Application Management:**

Allow users to view a list of job applications they have submitted.

Display basic details of each application, such as job title, company, and application status.

Enable users to click on an application to view details.

* **Collect Additional Details:**

Develop a user-friendly form component where applicants can input their details for job applications.

* **Resume/CV Management:**

Implement a feature for users to upload their resumes/CVs during registration or profile creation.

Display the uploaded resume/CV in the user's profile for reference.

**Backend Improvements (Critical Features):**

* **User Data Storage:**

Expand the backend to store additional user details, such as resume/CV files and application history.

Utilize a simple database or file storage system to store user data securely.

* **Application Submission Handling:**

Develop backend endpoints to handle job application submissions from the frontend.

Store application data in a structured format for future reference and retrieval.

**Server Enhancements (Critical Aspects):**

* **Reliable Hosting:**

Deploy the application on a reliable hosting platform to ensure availability during project development and demonstrations.

**Requirements for Phase 2:**

1. **Advanced Search Filters:**

* Improve the existing job search functionality by adding basic search filters, such as job title and location.
* Display search results in a clear and organized manner, allowing users to easily browse through job listings.

1. **Apply for Jobs:**

* Allow registered users to apply for jobs directly through the platform, streamlining the application process.
* Provide a user-friendly interface for submitting job applications, ensuring ease of use and accessibility for applicants.
* Improve the existing job search functionality by adding basic search filters, such as job title and location.
* Display search results in a clear and organized manner, allowing users to easily browse through job listings.

1. **Upload Resume/CV:**

* Enable users to upload their resumes or CVs directly to their profiles, enhancing their visibility to employers and recruiters.
* Implement secure storage mechanisms to store uploaded documents, ensuring the privacy and integrity of user data.
* Provide options for users to update or replace their resumes as needed, keeping their profiles up-to-date with the latest information.

1. **Job Application Submission:**

* Facilitate the submission of job applications through the platform, allowing users to apply for multiple positions easily.
* Include features such as tracking the status of submitted applications and receiving notifications for any updates or responses from employers.
* Ensure that the application submission process is seamless and intuitive, guiding users through each step with clear instructions and feedback.

1. **Collect Additional Details:**

* Gather additional details from users during the job application process to provide employers with more comprehensive information about candidates.
* Include fields for users to input relevant details such as work experience, education background, skills, and certifications.
* Design the data collection process to be user-friendly and intuitive, minimizing user effort while maximizing the value of collected information.

1. **User Storage and Data Management:**

* Enhance user data storage capabilities to accommodate additional details like resume/CV files and application history securely.
* Implement backend endpoints to handle job application submissions seamlessly, storing data in a structured format for future retrieval.
* Utilize a simple yet robust database or file storage system to ensure efficient and secure storage of user information and application data.
* Develop functionalities for users to access and manage their stored data, including updating profiles and reviewing application history.
* Continuously monitor and optimize backend performance to maintain data integrity and provide a seamless user experience.

1. **Responsive Design:**

* Continue to prioritize responsive design principles to ensure that the platform remains accessible and functional across a wide range of devices and screen sizes.
* Optimize the user interface for mobile devices, considering factors such as touch interactions, screen resolutions, and viewport sizes.
* Conduct regular testing and optimization to ensure consistent performance and usability across different platforms and devices.

By incorporating these Phase 2 requirements into our Job Hub project, we aim to further enhance the user experience and functionality of the platform, making it a valuable tool for job seekers.

The Phase 2 plan has been revised to prioritize core functionalities due to project constraints and priorities. Initially, it included features like **Application Management for Job Seekers**, **Employer Account Management**, and **Personalized Job Alerts**. However, to ensure a streamlined development process and timely delivery, the focus will be on the essential functionalities mentioned above.

The decision to cut certain features was made due to limitations in resources and time, as well as the need to focus on essential functions that provide the most value to users. By concentrating on tasks like **Advanced Search Filters**, **Apply for Jobs**, **Upload Resume/CV**, **Collect Additional Details**, **User Storage and Data Management,** and making the platform responsive, we aim to complete Phase 2 effectively and set the stage for future improvements.

**Removed Features and Justification:**

* **Application Management for Job Seekers:**

**Justification:** Although Application Management would enhance user experience by enabling job seekers to track their applications, implementing this feature requires substantial backend development and user interface design. Given our project's resource constraints and time limitations, prioritizing this feature could prolong the project timeline beyond acceptable bounds.

* **Employer Account Management:**

**Justification:** Employer Account Management is crucial for allowing employers to post jobs and manage their listings. However, its implementation involves creating specialized user authentication and authorization systems tailored for employers, as well as additional frontend and backend components. Given the current project scope, focusing on core functionalities like user registration and basic job search capabilities is more pragmatic.

* **Personalized Job Alerts:**

**Justification:** Personalized Job Alerts offer tailored job recommendations based on user preferences, necessitating the development of advanced algorithms and data processing mechanisms. Integrating this feature would significantly extend the project timeline and require substantial backend and frontend development efforts. Prioritizing core functionalities like job listing aggregation and basic search capabilities better aligns with our project's constraints and objectives.

**Cross-Platform Compatibility:**

* **Responsive Design:**

Implementing responsive design techniques to ensure that user interfaces adapt seamlessly to different screen sizes and devices, providing a consistent user experience across platforms.

* **Compatibility Testing:**

Conducting thorough compatibility testing across various browsers, operating systems, and devices to identify and address any compatibility issues that may arise.

* **Adaptive Layouts:**

Designing adaptive layouts that adjust content placement and sizing dynamically based on screen dimensions, optimizing usability and readability on different devices.

* **Touch-Friendly Interaction:**

Incorporating touch-friendly navigation and interaction elements to accommodate mobile and tablet users, enhancing accessibility and user engagement.

**Database Management:**

* **Schema Design:**

Designing efficient database schemas that accurately represent the relationships between data entities, ensuring optimal data organization and retrieval.

* **Data Modeling:**

Employing data modeling techniques to define the structure, constraints, and integrity rules of the database, maintaining data consistency and integrity.

* **Indexing Strategies:**

Implementing indexing strategies to improve query performance and optimize data retrieval speed, enhancing overall system responsiveness.

* **Backup and Recovery:**

Establishing backup and recovery procedures to safeguard against data loss and ensure business continuity in the event of system failures or disasters.

**UML Diagram:**

**Class Diagram:**

* This diagram illustrates the static structure of the system, depicting classes, their attributes, methods, and relationships.
* Essential classes for Phase 1 may include Job, JobSearchSystem, User, Resume, Application.

**Job:**

* **Attributes:** Job Id, Job Title, Job Description, responsibilities, Company Details.

**JobSearchSystem:**

* **Methods:** searchJobs(items), displayJobResults().

**User:**

* **Attributes:** username, email, expected CTC, No.of Experience.

**Resume:**

* **Attributes:** resume id, user id, resume file, upload date.

**Application:**

* **Attributes:** jobid, userid, appid, status, dateApplied.

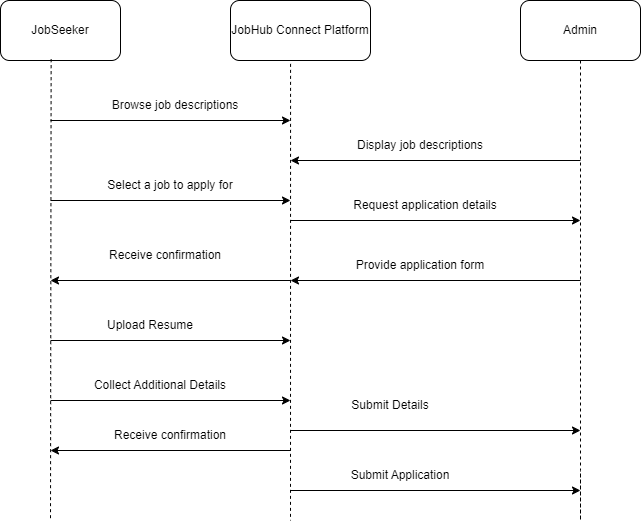
In the relationships:

* Job has a composition relationship with Company, indicating that each job description includes details about the company offering the job.
* **JobSearchSystem:** Aggregates Job instances: The JobSearchSystem aggregates instances of Job class to manage the list of job listings.
* **JobApplication:** Belongs to User: Each JobApplication instance is associated with a User instance representing the applicant who submitted the application.
* **Resume:** Belongs to User: Each Resume instance belongs to a user instance, representing the user to whom the resume belongs.
* **User:** User aggregates the information of the username, official email, number of years’ experience and expected CTC.

**A diagram of a software company

Description automatically generated with medium confidence**

**Sequence Diagram:**

****

JobHub

User Interface

This sequence diagram depicts the interactions between a JobSeeker, a User Interface, and a JobHub system in the context of an online job application process. Here's a description of the sequence:

1. The JobSeeker starts by browsing job descriptions displayed by the User Interface.

2. The JobSeeker selects a job they want to apply for.

3. The User Interface requests application details from the JobHub system.

4. The JobHub system provides an application form to the User Interface.

5. The User Interface displays the application form to the JobSeeker.

6. The JobSeeker receives a confirmation from the User Interface.

7. The JobSeeker uploads their resume through the User Interface.

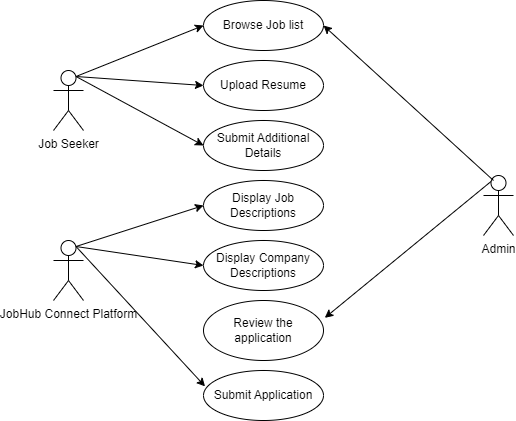
8. The JobSeeker collects and provides additional details required for the application.

9. The User Interface submits the details to the JobHub system.

10. The JobSeeker receives another confirmation from the User Interface.

11. The User Interface submits the complete application to the JobHub system.

**Use case Diagram (Normal handling):**

****

JobHub

User Interface

**Normal Case:**

1. **Browse Job List:**

* **Description:** The job seeker browses the list of available job opportunities.
* **Actor:** Job Seeker
* **Precondition:** None
* **Main Success Scenario:**
  + Job Seeker interacts with the User Interface to browse the job list.
  + User Interface displays the available job descriptions from the JobHub.
* **Postcondition:** Job Seeker can view the available job listings.

2. **Upload Resume:**

* **Description:** The job seeker uploads their resume for job applications.
* **Actor:** Job Seeker
* **Precondition:** Job Seeker has selected a job to apply for.
* **Main Success Scenario:**
  + Job Seeker selects the option to upload their resume through the User Interface.
  + Job Seeker provides their resume file.
  + User Interface facilitates the upload of the resume.
* **Postcondition:** Job Seeker's resume is uploaded and associated with their job application.

3. **Submit Additional Details:**

* **Description:** The job seeker submits additional details required for the job application.
* **Actor:** Job Seeker
* **Precondition:** Job Seeker has selected a job to apply for and uploaded their resume.
* **Main Success Scenario:**
  + Job Seeker fills in the required additional details (e.g., personal information, education, work experience) through the User Interface.
  + Job Seeker submits the additional details.
  + User Interface facilitates the submission of the additional details to the JobHub.
* **Postcondition:** Job Seeker's additional details are associated with their job application.

4. **Display Job Descriptions:**

* **Description:** The User Interface displays detailed descriptions of the available job opportunities.
* **Actor:** User Interface
* **Precondition:** Job Seeker has requested to view job descriptions.
* **Main Success Scenario:**
  + User Interface retrieves job descriptions from the JobHub.
  + User Interface presents the job descriptions to the Job Seeker.
* **Postcondition:** Job Seeker can view detailed information about the available job opportunities.

5. **Display Company Descriptions:**

* **Description:** The User Interface displays information about the companies offering the job opportunities.
* **Actor:** User Interface
* **Precondition:** Job Seeker has requested to view company information.
* **Main Success Scenario:**
  + User Interface retrieves company descriptions from the JobHub.
  + User Interface presents the company descriptions to the Job Seeker.
* **Postcondition:** Job Seeker can view information about the companies offering the job opportunities.

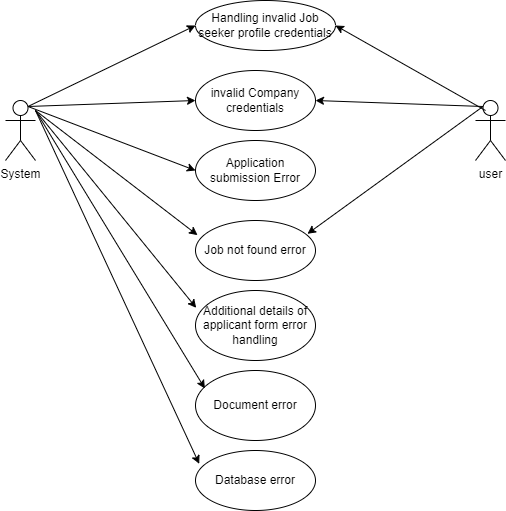
6. **Review the Application (Backend):**

* **Description:** The job seeker reviews their job application before submission.
* **Actor:** Job Seeker
* **Precondition:** Job Seeker has completed the application process (uploaded resume, submitted additional details).
* **Main Success Scenario:**
  + Job Seeker reviews their application through the User Interface.
  + Job Seeker confirms the accuracy of the application details.
* **Postcondition:** Job Seeker has reviewed and confirmed their job application.

7. **Submit Application:**

* **Description:** The job seeker submits their job application.
* **Actor:** Job Seeker
* **Precondition:** Job Seeker has reviewed and confirmed their job application.
* **Main Success Scenario:**
  + Job Seeker selects the option to submit their application through the User Interface.
  + User Interface facilitates the submission of the application to the JobHub.
* **Postcondition:** Job Seeker's application is successfully submitted to the JobHub for further processing.

**Use Case Diagram (Error case handling):**

****

In Phase 2 there are many errors handling, here we are showing two error case.

Invalid Credentials are already handled in Phase 1.

**Error Case:**

1. **Application Submission Error:**

* Description: The system encounters an error during the submission of a job application by a user.
* Actor: System
* Precondition: User attempts to submit a job application.
* Alternative Flow:
  + User completes the job application process and initiates the submission.
  + System encounters an error during the submission process (e.g., network issue, database problem).
  + System displays an error message indicating that the application submission was unsuccessful.
  + User is prompted to retry the submission or contact support.
* Postcondition: Job application remains in an incomplete or unsubmitted state until the issue is resolved.

2. **Job Not Found Error:**

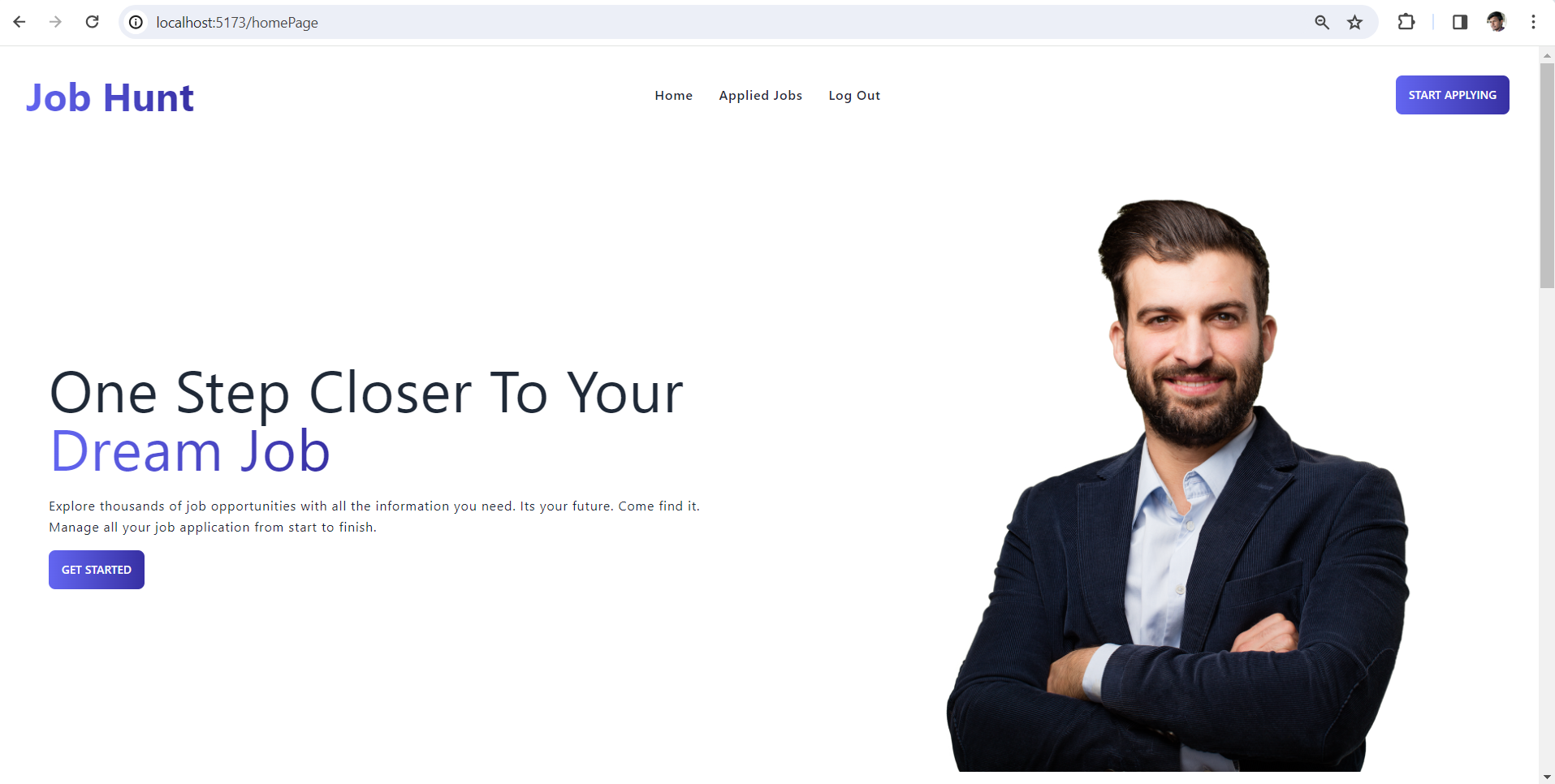
* Description: The system encounters an error when a user attempts to view or apply for a job that does not exist or has been removed.
* Actor: System
* Precondition: User attempts to access or apply for a specific job listing.
* Alternative Flow:
  + User selects a job listing from the search results or enters a specific job ID.
  + System fails to find the requested job listing (e.g., job has been removed, invalid job ID).
  + System displays an error message indicating that the requested job was not found.
  + User is prompted to search for other job listings or contact support.
* Postcondition: User cannot proceed with viewing or applying for the requested job listing.

This Use Case diagram depicts the interactions between users and the system for Phase 2 of the JobHub Connect project, including both normal and error case handling scenarios.

**Test Cases:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Functionality** | **Input** | **Output** | **Expected** |
| Job Listing | Empty | Successfully routing | Successfully Routing |
| Job Advanced Filtering | Company Name, Location, Salary, Job Role | Company Name, Location, Salary, Job Role jobs should be listed | Company Name, Location, Salary, Job Role jobs should be listed |
| Job Description | Empty | Job Details listed Successfully | Job Details will be listed Successfully |
| Resume Upload | File (.pdf, .docx) | Resume Upload Successfully | Resume Upload Successfully |
| Collect Additional Details | Empty | Error (Enter all details) | Error (Enter all details) |
| Collect Additional Details | Name, email, expected CTC, YOE | Job applied Successfully | Job applied Successfully |
| Job Applied to same applied previously | Resume, Additional Details | Error | Application Submission error |

**Test Case 1: Job Listing when user successfully login.**

****

After successful login, User able to access the portal.

Click on ‘Start Applying’ to view all jobs.

A screenshot of a computer

Description automatically generated

**Test Case 2: Job Advanced Filtering**

**Job Listing based on Company name:**

**A screenshot of a computer

Description automatically generated**

Job listing based on ‘Google’ Company Name.

**Job Listing based on Role:**

**A screenshot of a computer

Description automatically generated**

Job Listing based on Role as ‘Engineer’. JobHub Portal will list all jobs which have Engineer as a role.

**Job Listing based on Location:**

**A screenshot of a computer

Description automatically generated**

Job Listing based on Location as ‘Anywhere. JobHub Portal will list all jobs which have Anywhere as Location.

**Job Listing based on Salary:**

**A screenshot of a computer

Description automatically generated**

Job Listing based on Salary as ’95,000’. JobHub Portal will list all jobs which have salary in between 95,000$.

**Test Case 3: Job Description**

**A screenshot of a computer

Description automatically generated**

After Click on view Details, User able to all job details and description.

**Test Case 4: Resume Upload.**

**A screenshot of a computer

Description automatically generated**

After Successful upload resume, it will redirect to additional details collect.

Upload resume is an Option for user to upload it or not.

**Test Case 5: Collect Additional Details submitting with empty details.**

**A screenshot of a computer

Description automatically generated**

If a user tries to submit the application without data, it will trough is required error.

**Test Case 6: Collect Additional Details submitting with details.**

**A screenshot of a computer

Description automatically generated**

After entering all details, User will able to submit the application.

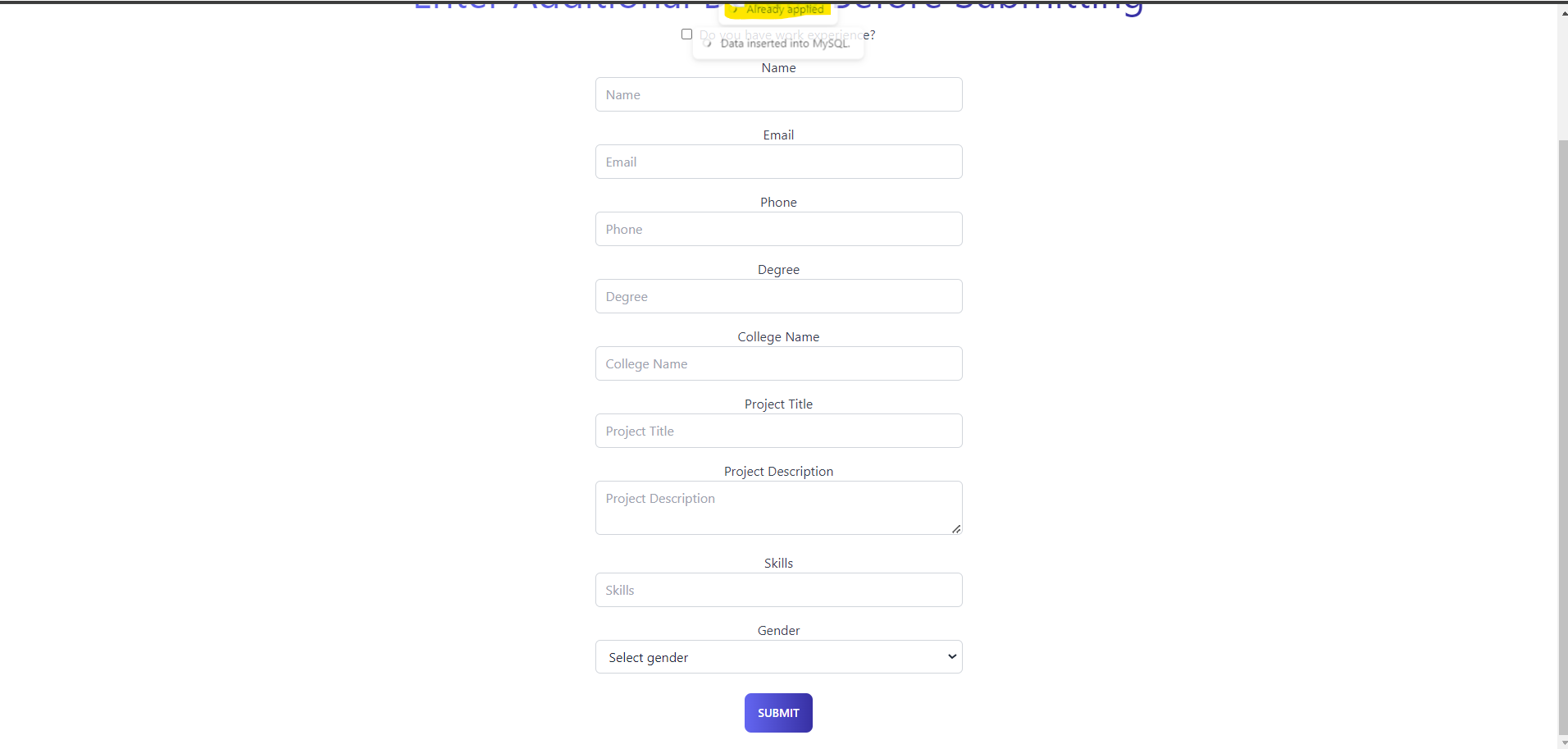
**Test Case 7: Job Applied to same applied previously.**

Previously applied to meta if user try to apply to same role in same company, it will throw an error called “Already Applied”.

**A screenshot of a computer

Description automatically generated**

Tries to Submit same job.

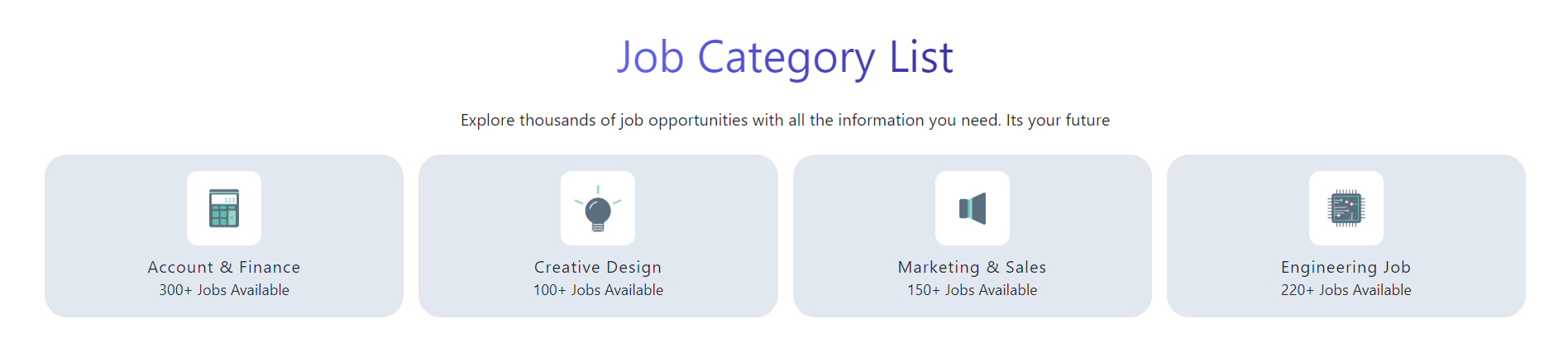
****

‘Already Applied’ message toasted to user.

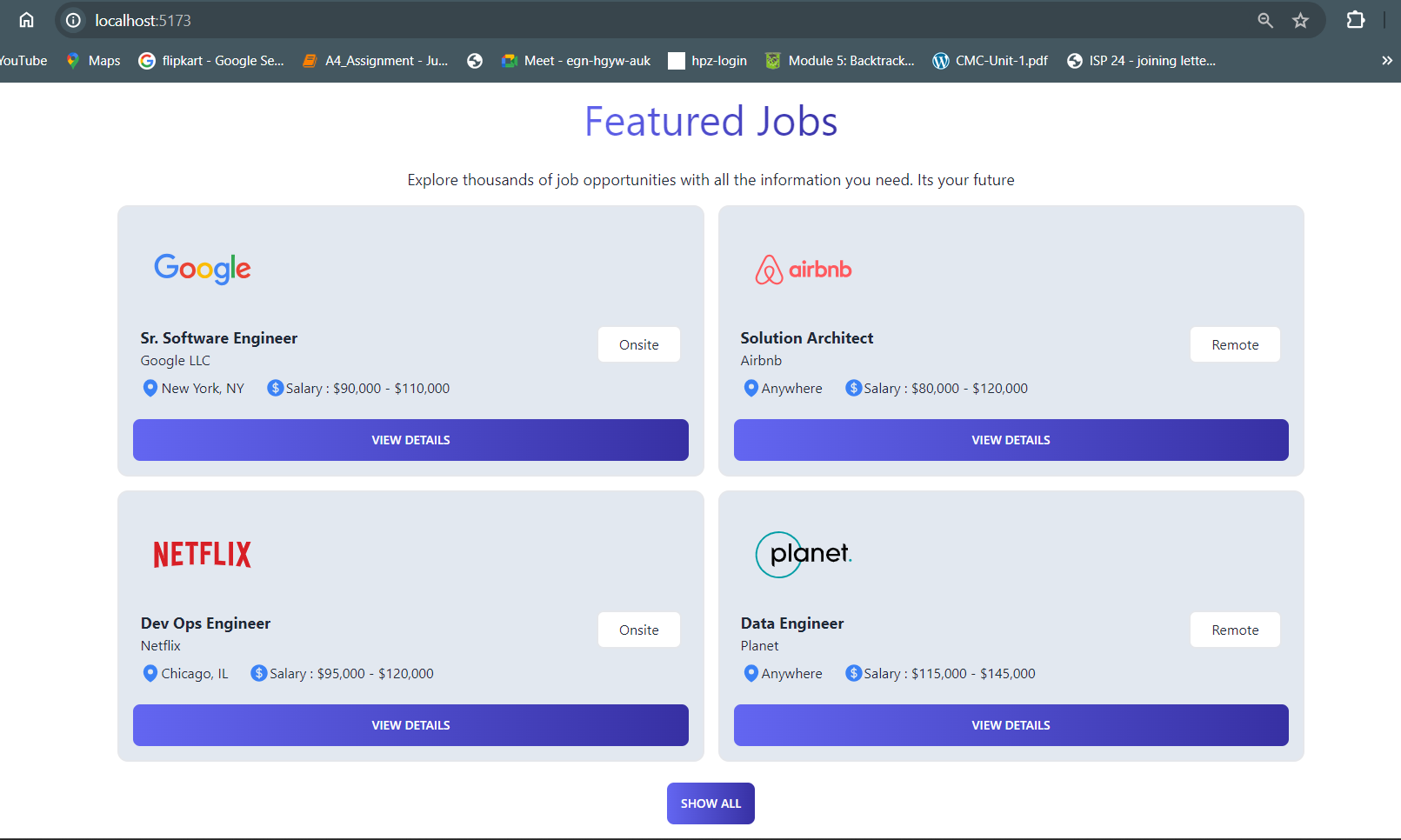
**User Manual Guide (Walk Through):**Please go to <http://localhost:5173/registerPage>

**A screenshot of a video game

Description automatically generated**After the Registration, You need to go to<http://localhost:5173/loginPage> **A computer screen with a pen

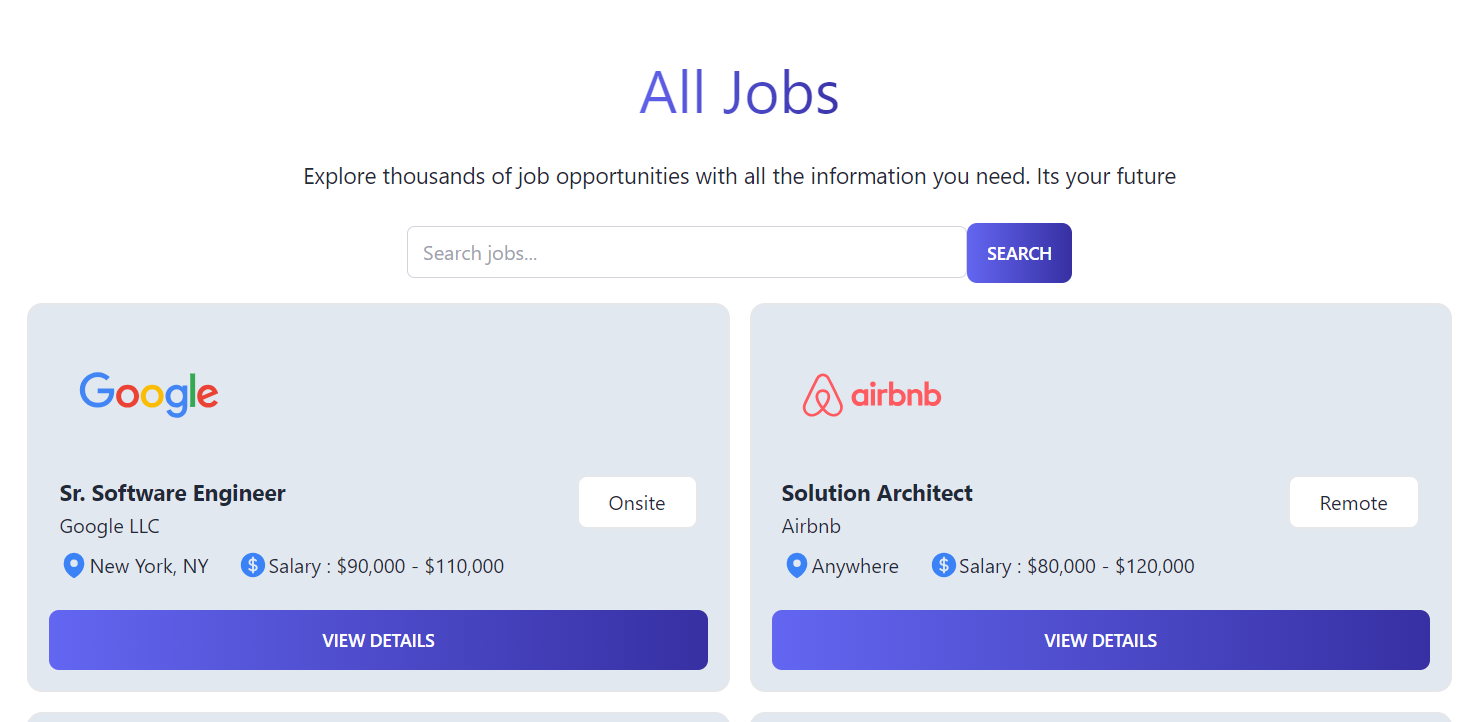
Description automatically generated  
A) Job Categories (Home Page)  
**

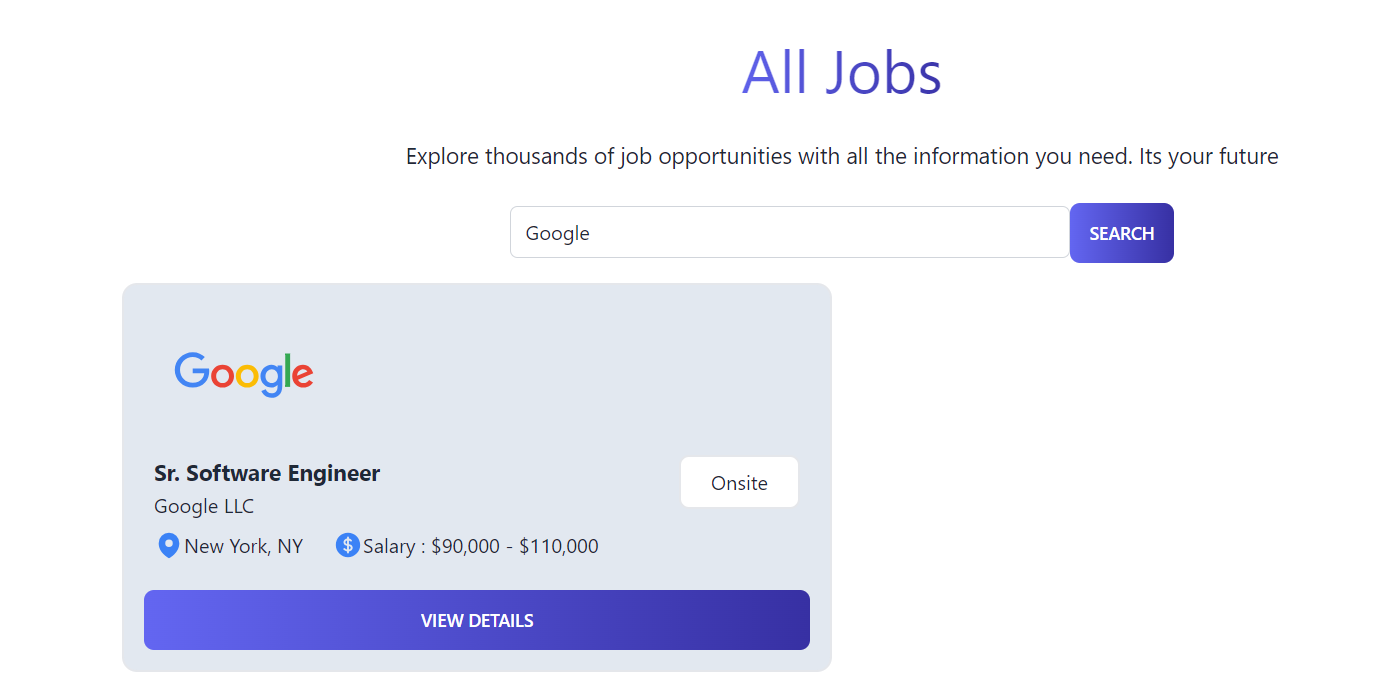
The Job Category List on JobHub Connect acts as a comprehensive directory, encompassing a myriad of industries, sectors, and specializations to cater to the diverse career interests and aspirations of users. From traditional fields such as IT, healthcare, finance, and education to emerging sectors like renewable energy, digital marketing, artificial intelligence, and blockchain, the platform offers an extensive array of job categories. Each category is meticulously curated to ensure relevance and depth, empowering users to explore opportunities that align with their skill sets, passions, and career objectives. With a user-friendly interface and intuitive navigation, the Job Category List serves as a gateway for users to embark on their professional journey with confidence and clarity.

**B) Featured Jobs (Home Page)  
**

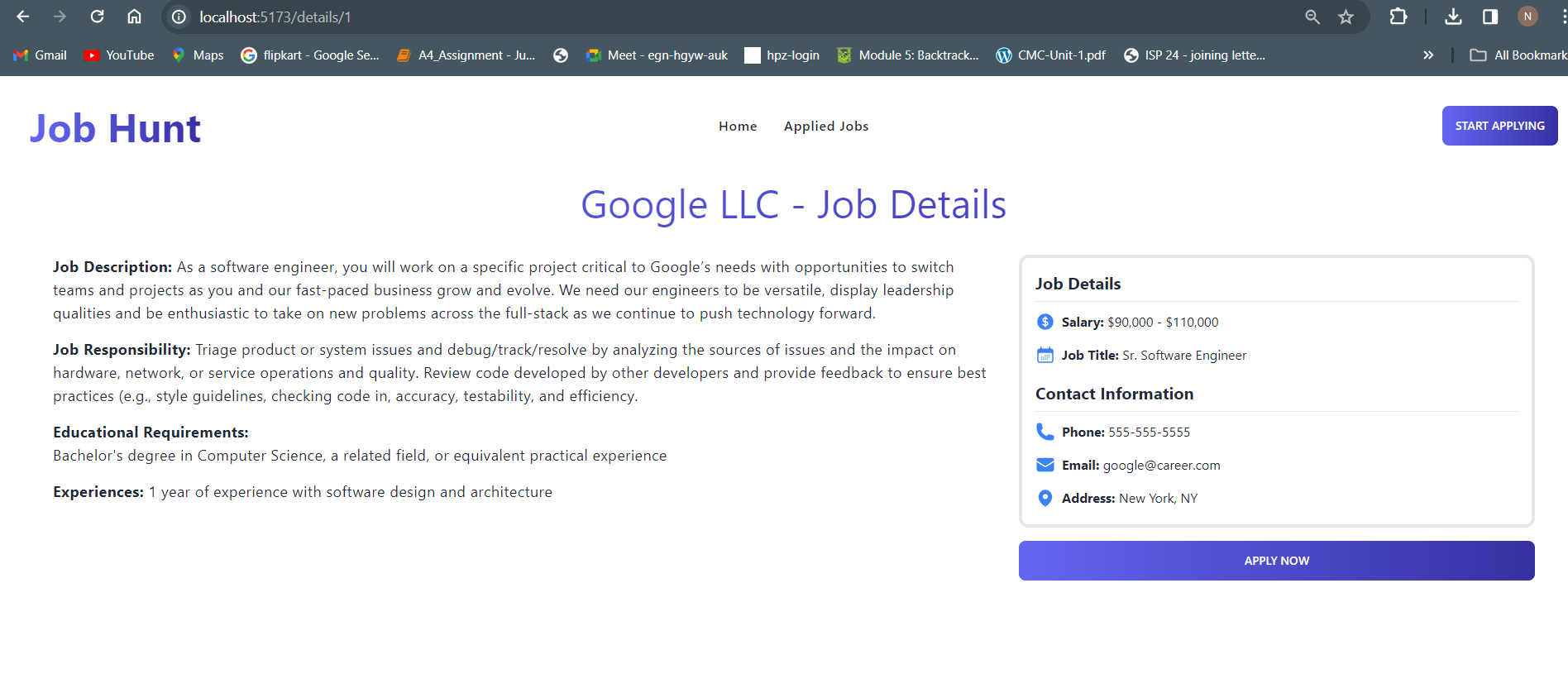
Featured Jobs on JobHub Connect spotlight premium job listings that represent exceptional career opportunities with leading employers, prominent organizations, and innovative startups. These select positions are strategically highlighted on the platform's homepage or dedicated sections, capturing the attention of users seeking top-tier roles and competitive advantages. Featured Jobs may include executive positions with industry leaders, specialized roles with renowned companies, or exclusive opportunities with attractive compensation packages and perks. By showcasing these standout opportunities prominently, JobHub Connect enables users to discover and pursue coveted career paths that elevate their professional trajectory and unlock new possibilities for growth and success.

**C) All Jobs Along with Search Functionality (** <http://localhost:5173/allJobs> **)**

****The All Jobs section on JobHub Connect serves as a comprehensive repository of job listings sourced from a diverse range of sources, including company websites, job boards, and recruitment agencies. Equipped with robust search functionality, users can effortlessly navigate through the extensive collection of job opportunities by leveraging advanced filters and search parameters. Whether users are searching for specific job titles, locations, industries, or salary ranges, the platform's intuitive search features empower them to refine their search results and pinpoint relevant opportunities that match their preferences and qualifications. With real-time updates and personalized recommendations, JobHub Connect ensures that users have access to the latest job listings and tailored recommendations tailored to their unique needs and preferences.

**D)Searching Functionality with Company Name, Location, Salary, Job Role. (** <http://localhost:5173/allJobs> **)  
**

JobHub Connect's advanced search functionality extends to enabling users to search for job listings by company name, providing enhanced granularity and specificity in their job search process. By entering the name of a specific company into the search bar, users can uncover job opportunities offered by their preferred employers or organizations. This feature enables users to explore career opportunities aligned with their values, culture, and aspirations, facilitating targeted searches tailored to their preferences and career objectives. Whether users are interested in joining Fortune 500 companies, innovative startups, or mission-driven nonprofits, JobHub Connect empowers them to discover and pursue opportunities that resonate with their professional goals and aspirations.

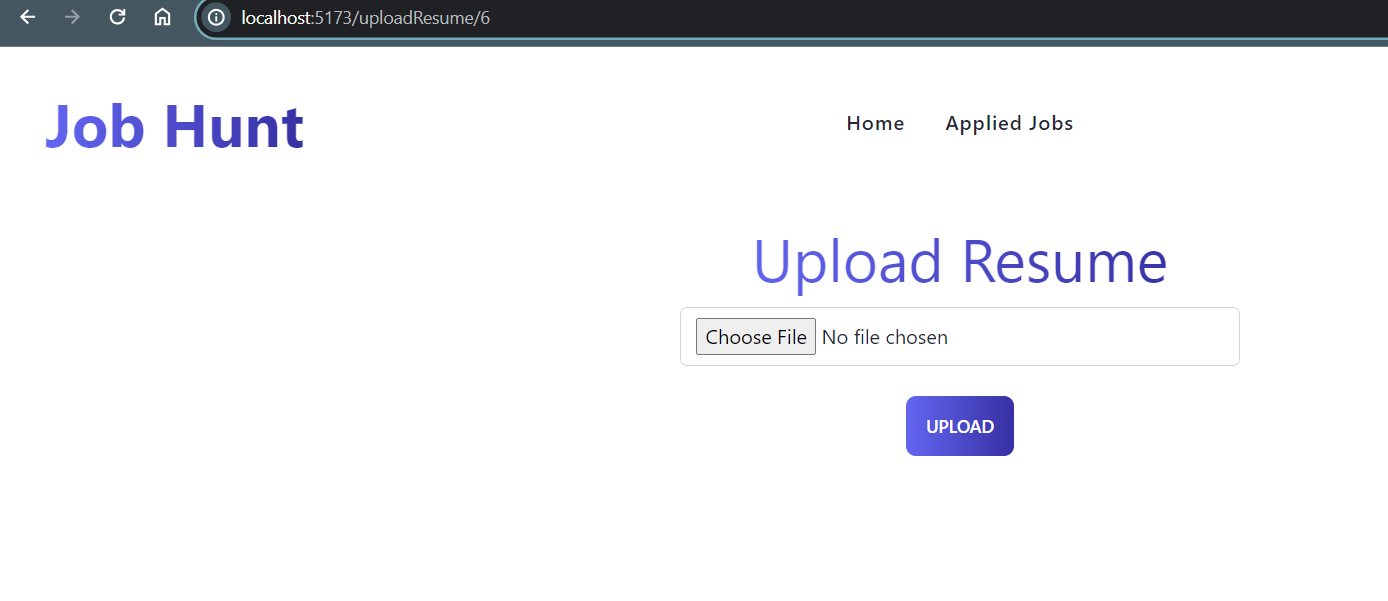
**E)After View Details Button – Opening Job description (**<http://localhost:5173/details/1>**)  
**

Upon clicking the "View Details" button for a specific job listing, users are seamlessly directed to the detailed job description page, where they can access comprehensive information about the position, company, and application process. The job description page offers a wealth of relevant details, including job responsibilities, qualifications, salary range, benefits, and application instructions. Users can delve deeper into the specifics of the job opportunity, gaining insights into the company's culture, values, and growth prospects. With user-friendly navigation and interactive features, JobHub Connect ensures that users have a seamless and informative browsing experience, empowering them to make informed decisions about their career advancement.

**F)After clicking apply You Will Be redirected to upload resume screen (Reference link is too long).**

****After successfully submitting their job application by clicking the "Apply Now" button, users receive immediate confirmation through a popup message indicating that their application has been successfully submitted. The "Successfully Applied" popup serves as a reassuring acknowledgment of the user's action, providing instant validation and peace of mind. This prompt confirmation enhances the user experience by fostering trust, transparency, and engagement, ensuring that users feel valued and informed throughout the application process. By delivering a seamless and responsive application experience, JobHub Connect reinforces its commitment to user satisfaction and success, establishing itself as a trusted ally in the journey towards meaningful career opportunities.

**G) After clicking the Upload button you will be redirected to “Additional details screen”.**

****

**H)After clicking Submit button you will be redirected to “Home” Page  
Along with Successfully Applied Screen (Reference link is too long).**

**A screenshot of a computer

Description automatically generated**

If user have any Work Experience, then user will be able to click in check box to choose that User have an experience.

**Experienced user Screen:**

A screenshot of a computer

Description automatically generated

**Report Compilation Instructions:**

1. Clone the React Project:

- If the project is hosted on a Git repository and you want to clone it, follow these steps:

- Open a terminal or command prompt.

- Navigate to the directory where you want to clone the project.

- Use the `git clone` command to clone the project repository. Replace `https://github.com/username/project.git` with the actual URL of the repository you want to clone:

```

https://github.com/KudidalaPavan/ALPHA.git

```

- If you already have the project files or if you're not using version control, you can skip this step and proceed to the next one.

2. Navigate to the Project Directory:

- Open a terminal or command prompt.

- Use the `cd` command to navigate to the directory where your project is located. For example:

```

cd ALPHA/Project Source Code/Phase 2/

```

3. Install Project Dependencies:

- Once you're in the project directory, run the following command to install the necessary dependencies specified in the `package.json` file:

```

npm install

```

- This command will download and install all the required packages for the project.

4. Start the React Development Server:

- After the dependencies are installed, you can start the React development server by running the following command:

```

npm run dev

```

Looks something like this  
A black screen with white text

Description automatically generated

- This command will start the development server and open your default web browser to display the React application.

- You should see the React project running locally at `http://localhost:5173/` by default.

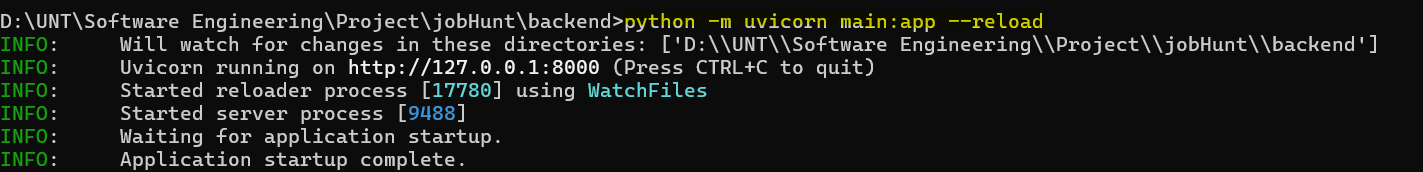
5. Set Up IDE (Optional):

- If you prefer to use an Integrated Development Environment (IDE), open your preferred IDE (e.g., Visual Studio Code, Atom, Sublime Text).

- Navigate to the directory where your project is located using the IDE's file explorer.

- Open the project folder in the IDE.

6. Backend Command To start the BE for job hunt:



These steps should cover the process of cloning a React project from a Git repository, installing project dependencies, starting the development server, and setting up your IDE for development.

**Report Code Inspection Feedback:**

1. **Clarity and Conciseness:**

* The Phase 2 requirements are clearly outlined, making it easy to understand the goals and objectives of this development phase.
* Each requirement is presented succinctly, without unnecessary elaboration, ensuring that the key points are conveyed effectively.

2. **Prioritization of Features:**

* The decision to prioritize core functionalities such as Advanced Search Filters, Apply for Jobs, and Upload Resume/CV is justified considering the project's constraints and priorities.
* By focusing on essential features, we can deliver value to users while efficiently managing resources and meeting project deadlines.

3. **Justification for Removed Features:**

* The rationale for removing features like Application Management for Job Seekers, Employer Account Management, and Personalized Job Alerts is well-founded.
* The explanation provided for each removed feature highlights the considerations taken into account, such as development complexity and resource availability.

4. **Recommendations for Enhancement:**

* While the report effectively communicates the planned functionalities, incorporating visual aids such as diagrams or wireframes could enhance understanding, particularly for complex features like data management and application submission.
* Including a section on potential challenges and mitigation strategies would add depth to the plan, demonstrating foresight and risk management capabilities.
* Providing estimates for development time and resource allocation for each feature could help stakeholders better understand the project's timeline and resource requirements.

5. **Overall Assessment:**

* The Phase 2 plan demonstrates a strategic approach to development, focusing on core functionalities that align with user needs and project objectives.
* The decision-making process for feature prioritization and removal is transparent and logical, reflecting a thorough understanding of project constraints and goals.
* With minor enhancements in visualization and risk assessment, the Phase 2 plan will effectively guide the development team towards successful implementation.

In conclusion, the Phase 2 plan sets a solid foundation for the continued development of our Job Hub platform. By addressing the recommendations provided, we can further refine the plan and ensure its effectiveness in guiding the project towards successful completion.

**Report Reflection:** **Report Reflection:**

This reflection aims to highlight key insights gained and outline areas for improvement to ensure the success of JobHub project.

**Strengths:**

1. **Comprehensive Planning:** The Phase 2 plan demonstrates thorough planning and consideration of essential functionalities. By prioritizing core features like Advanced Search Filters and Job Application Submission, we ensure that our project remains focused on delivering value to users.

2. **Transparent Communication:** The report effectively communicates project constraints, such as resource limitations and time constraints. This transparency fosters understanding among team members and stakeholders, aligning expectations with project realities.

3. **Strategic Decision-making:** The decision to remove certain features, such as Personalized Job Alerts, reflects strategic prioritization based on project constraints and objectives. This demonstrates a pragmatic approach to project management and resource allocation.

**Areas for Improvement:**

1. **Detailed Risk Assessment:** While the plan acknowledges potential challenges, a more detailed risk assessment with identified risks and mitigation strategies would enhance project readiness. This proactive approach ensures that we are better prepared to address unforeseen obstacles.

2. **User Feedback Integration:** Incorporating user feedback from Phase 1 could provide valuable insights into user preferences and priorities. This iterative approach ensures that our project aligns closely with user needs and expectations.

3. **Continuous Evaluation:** Implementing mechanisms for continuous evaluation and feedback throughout the development process enables us to adapt and refine our strategies as needed. Regular retrospectives and stakeholder meetings foster collaboration and alignment towards project goals.

**Action Plan:**

1. **Enhanced Risk Management:** We will conduct a more detailed risk assessment, identifying potential risks and developing mitigation strategies to minimize their impact on the project.

2**. User Feedback Integration:** We will gather and incorporate user feedback from Phase 1 into our development process, ensuring that our project evolves based on real user experiences and preferences.

3. **Continuous Improvement:** We will establish regular checkpoints for project evaluation and reflection, allowing us to assess progress, identify areas for improvement, and make necessary adjustments to our development approach.

By addressing these areas for improvement, we can enhance our project's effectiveness and increase our chances of delivering a successful outcome that meets the needs of our users and stakeholders.

**Team Member Contribution:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Team member** | **Individual Contributions** | **Overall Contribution (%)** | **Note**  **(if applicable)** |
| Uday Kalyan Nuthalapati | All Jobs Listing | 13% |  |
| Pavan | Collect Additional Details | 11% |  |
| Nithin | Creating Testcases | 10% |  |
| Abhishek | Creating functionalities document | 11% |  |
| Keerthi | Created Database to store details | 13% |  |
| Shravani | Working on User Manual | 10% |  |
| Vishal | Worked on UML Diagrams | 10% |  |
| Venkata | Database to store all details. | 11% |  |