**Deliverable 3 Job Hunt/Hub - Connect**

**Development Phase 1**

**Minimum Viable Product (MVP)**

**Requirements:**

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

* **User registration and login:**

Allow users to register for an account and log in securely.

Implement user authentication mechanisms to ensure secure access to user-specific features.

* **Job listing aggregation:**

Job listing aggregation in the JobHub Connect platform involves gathering job listings from various sources and displaying them in a user-friendly manner. The system presents static content of job listings, ensuring ease of navigation and accessibility for users. By aggregating job postings, explore diverse employment opportunities within a unified interface, streamlining their job search process.

* **Basic search functionality:**

Provide users with basic search capabilities to find relevant job listings based on keywords, location, category, etc.

Ensure the search functionality is intuitive and responsive.

**Back End Requirements (Critical Features):**

* **User authentication and authorization:**

Implement robust authentication mechanisms to verify user identities during login and registration processes.

Enforce authorization rules to control user access to different parts of the platform based on their roles and permissions.

**Server Requirements (Critical Aspects):**

* **Reliable hosting infrastructure to ensure uptime:**

Deploy the application on a reliable hosting infrastructure to ensure high availability and minimal downtime.

Choose a hosting provider with adequate resources and reliability guarantees to support the initial launch and usage of the platform.

**Requirements for Phase 1:**

1. **User Registration and Login:**

* Users should be able to register for an account by providing a username, email address, and password.
* The registration form should include validation checks to ensure that all required fields are filled out correctly.
* Upon successful registration, users should be able to login into portal by using username/email address and password.

2. **Job Listing Aggregation:**

* **Gathering job listings from diverse sources:** JobHub Connect platform collects job postings from various sources, including different job portals and websites.
* **Displaying in a user-friendly manner:** The system presents static content of job listings in a user-friendly format, ensuring easy navigation and accessibility for users.
* **Streamlining the job search process:** By aggregating job postings from different sources (Using Static Content), JobHub Connect provides users with a unified interface to explore diverse employment opportunities, simplifying and enhancing their job search experience.

3. **Basic Search Functionality:**

* Users should be able to search for job listings based on keywords and category.
* The search functionality should be intuitive and responsive, providing relevant results to users based on their search criteria.
* Search results should be displayed in a user-friendly format, allowing users to easily navigate through the listings.

4. **User Authentication and Authorization:**

* The platform should implement robust user authentication mechanisms to verify user identities during login and registration processes.
* User authorization rules should be enforced to control user access to different parts of the platform based on their roles and permissions.

5. **Responsive Design:**

* The platform should be designed with responsiveness in mind, ensuring that it is accessible and functional across various devices and screen sizes.
* User interface elements should adapt smoothly to different screen resolutions and orientations to provide a consistent user experience.

These requirements are essential for Phase 1 of the project and form the foundation of the JobHub Connect platform. They aim to deliver key functionalities to users while laying the groundwork for future development phases.

The plan for all phases has been modified to better align with project constraints and priorities. Initially, the scope included additional features such as tracking application status, accessing support services, and an employer portal. However, after careful consideration, it was determined that focusing on the core functionalities mentioned above would allow for a more streamlined development process and timely delivery of a Minimum Viable Product (MVP).

The decision to remove certain features was influenced by resource limitations, time constraints, and the need to prioritize essential functionalities that provide the most value to users. We can successfully complete Phase 1 and build the foundation for future platform revisions and additions by focusing our efforts on user registration, job listing aggregation, search capabilities, authentication, and responsive design.

**Removed Features and Justification:**

We had suggested several extra features to improve the Job Hub Connect platform's usefulness and functionality in our original project plan. But after doing more research and taking project priorities and financial limitations into account, we have decided to take these features out of project. The features that will not be included in Phase 1 and other phases are listed below, along with explanations for their exclusion:

**Job Listing Aggregation from Multiple Sources:**

In order to give customers, access to an extensive database of job openings, our initial aim was to compile job listings from various sources. We have chosen to emphasize other essential features in our project instead of integrating several APIs, nevertheless, because of resource limitations and the technological difficulties involved.   
  
**Justification**: Users would have access to a wider variety of job possibilities and the platform's content would be more diverse if job listing aggregation from various sources were integrated. However, in order to manage numerous APIs, guarantee data consistency, and preserve system stability, this functionality necessitates a large development effort. We have chosen to postpone this functionality until later stages due to time restrictions and the requirement to prioritize essential functions like user authentication and job application submission.

**Track Application Status and Access Support Services:**

Our original goal was to include tools that would let users monitor the progress of their job applications and make use of support services like career counselling and resume help. However, we have chosen to postpone the development of these features to later stages due to the difficulty of implementing them and the requirement to prioritize essential functions.

**Justification:** Implementing these features requires integration with employer systems, data synchronization, and user notification mechanisms. Given the complexity and time required for implementation, we have chosen to prioritize core functionalities essential for the platform's initial launch and usability.

**Employer Portal:**

We initially planned to develop an employer portal that would allow employers to post job openings, review job applications, schedule interviews, conduct interviews, select candidates, and communicate with applicants. However, due to the complexity and scope of these features, we have decided to prioritize user-centric functionalities for our project.

**Justification**: Employer engagement and recruitment procedures must be facilitated by implementing an employer portal where job openings can be posted, applications can be reviewed, and applicant communications can be managed. But developing a thorough employer portal calls for complex features like application tracking, user role management, and communication tools. We have eliminated this functionality from the initial release to give priority to the completion of JobHub and guarantee a flawless user experience for job seekers.

**Key Functionalities:**

The following crucial elements will receive special attention in order to be in line with the project's goals and guarantee effective completion within the allotted timeframe:

1. **User Authentication and Management:**

* Implement robust authentication mechanisms to ensure secure access to the platform.
* Allow users to register for accounts and manage their profiles securely.

1. **Job Listings Aggregation:**

* Display a curated list of job listings to users, facilitating easy navigation and exploration.
* Ensure the static job listings are comprehensive and relevant to users' interests and qualifications.

1. **Search for Jobs by Keywords:**

* Enable users to search for specific job listings based on keywords, enhancing the efficiency of their job search process.
* Provide intuitive search functionality with accurate and relevant results.

1. **Apply for Jobs:**

* Allow users to apply for jobs directly through the platform, simplifying the application process.
* Provide a seamless experience for users to submit applications and track their progress.

1. **Upload Resume/CV:**

* Enable users to upload their resumes or CVs to their profiles, enhancing their visibility to employers.
* Ensure smooth integration and secure storage of uploaded documents.

1. **Job Application Submission:**

* Facilitate the submission of job applications through the platform, streamlining the process for both applicants and employers.
* Provide features for users to attach documents, such as cover letters or portfolios, to their applications.

1. **Job Hub FAQs:**

* Create a comprehensive FAQ section to address common questions and concerns related to platform usage.
* Offer users easy access to helpful resources and information, improving their overall experience.

1. **Collect Additional Details:**

* Gather additional details from users during the application process, enriching their profiles and providing more comprehensive information to employers.
* Ensure data collection processes are user-friendly and comply with privacy regulations.

1. **Review Page:**

* Implement a review page feature where users can provide feedback and ratings on their application experiences.
* Foster transparency and accountability in the recruitment process, enabling users to make informed decisions.

By concentrating on these essential elements, we hope to provide a useful and intuitive platform that satisfies the requirements of employers and job searchers alike. These capabilities will ensure the JobHub Connect platform's continuous success and relevance in the employment market by laying the groundwork for future improvements and iterations.

**UML Diagrams:**

**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 User, JobListing, SearchCriteria, UserInterface.
* Relationships between classes, such as associations, aggregations, and inheritances, should be clearly defined to represent the system's structure accurately.

**User**:

* Attributes: id, username, email, password
* Methods: register(), login()

**JobListing:**

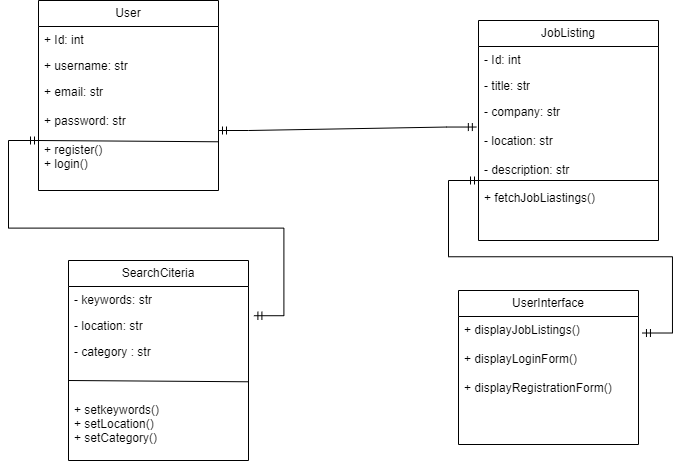
* Attributes: id, title, company, location, description
* Methods: fetchJobListings()

**SearchCriteria:**

* Attributes: keywords, location, category
* Methods: setKeywords(), setLocation(), setCategory()

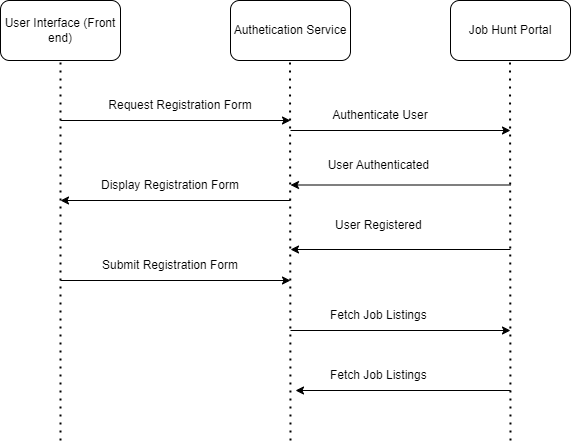
**UserInterface:**

* Methods: displayJobListings(), displayLoginForm(), displayRegistrationForm()



**Sequence diagram:**

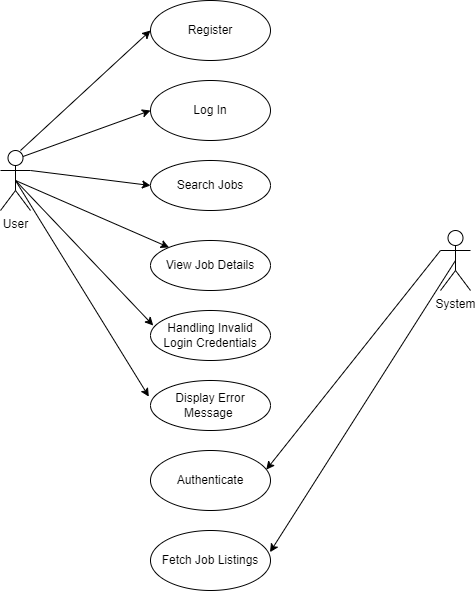
* The "User Interface" initiates the interaction by requesting the registration form.
* The "Authentication Service" authenticates the user upon receiving the registration form submission.
* Once the user is authenticated, the "User Interface" displays the registration form.
* Upon submission of the registration form, the "Authentication Service" registers the user.
* Simultaneously, the "JobHub Portal" fetches job listings.
* Once job listings are fetched, the "JobHub Portal" sends the listings to the "User Interface," which then displays them.



This sequence diagram demonstrates the flow of interactions between different components involved in Phase 1 of the JobHub Connect project, including the frontend user interface, authentication service, and external job portal.

**Use Case Diagram:**

The Use Case Diagram for Phase 1 of the JobHub Connect project should illustrate the various interactions between actors (users) and the system.

****

**Normal Case:**

1. **Register:**

* + Description: The user registers for an account on the platform.
  + Actor: User
  + Precondition: User is not registered.
  + Main Success Scenario:
* User provides registration details.
* System validates and registers the user.
* User receives confirmation of successful registration.
  + Postcondition: User is registered and can now log in.

2. **Log in:**

* + Description: The user logs in to their account.
  + Actor: User
  + Precondition: User is registered.
  + Main Success Scenario:
    - User provides login credentials.
    - System authenticates the user.
    - User gains access to their account.
  + Postcondition: User is logged in and can access platform features.

3. **Search Jobs:**

* Description: The user searches for job listings.
* Actor: User
* Precondition: User is logged in.
* Main Success Scenario:
* User enters search criteria (e.g., keywords, location).
* System retrieves relevant job listings.
* User views search results.
* Postcondition: User finds relevant job listings.

4. **View Job Details:**

* Description: The user views details of a specific job listing.
* Actor: User
* Precondition: User has searched for job listings.
* Main Success Scenario:
  + User selects a job listing from search results.
  + System displays details of the selected job.
* Postcondition: User views detailed information about the job.

**Error Case:**

1. **Invalid Login Credentials:**

* Description: The user enters invalid login credentials.
* Actor: User
* Precondition: User is not logged in.
* Alternative Flow:
* User enters an incorrect email or password.
* System fails to authenticate the user.
* User receives an error message indicating invalid credentials.
* Postcondition: User remains logged out and must retry login with correct credentials.

This Use Case diagram illustrates the interactions between users and the system for Phase 1 of the JobHub Connect project, including both normal and error case handling scenarios.

**Test Cases:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Functionality** | **Input** | **Output** | **Expected** |
| Register | Empty | Error | Unable to sign up |
| Register | Unique Credentials | Sign Up Successfully | Signup Successfully |
| Login | Empty | Error | Unable to Login |
| Login | Unique Credentials | Login Successfully | Login Successfully |
| Browse Listing | Empty | Successfully routing | Successfully Routing |
| Job Filter | Company Name | Company name jobs should be listed | Company name jobs should be listed |

1. **Register Page checking for empty fields**

**A screenshot of a computer

Description automatically generated**

**b) Register Page successful for correct details in fields  
A screenshot of a computer

Description automatically generated**

After Successful register it will redirect to login page to access the JobHunt Portal.

1. **Login page with Empty fields**A screenshot of a computer

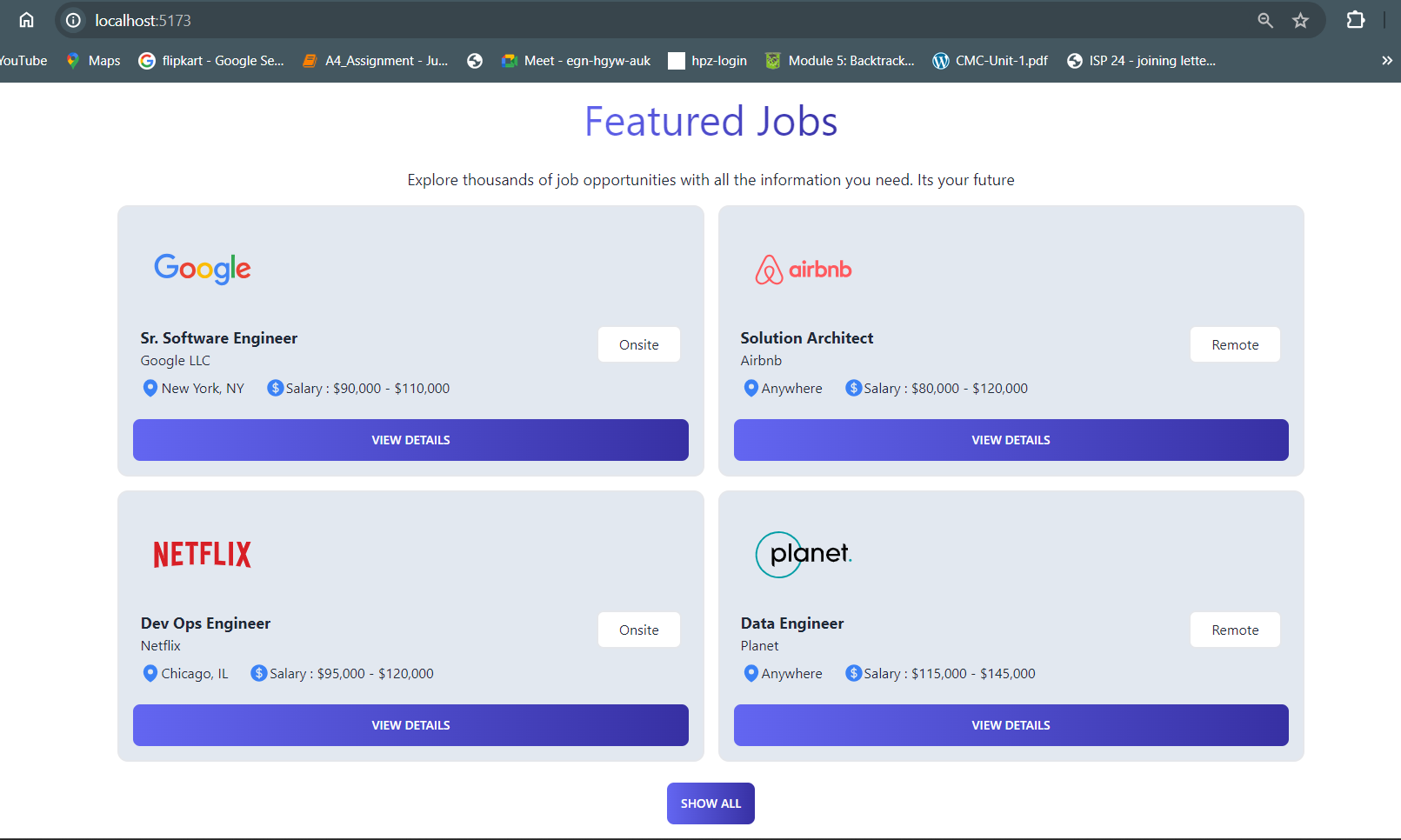
   Description automatically generated

Login page failed due to login with empty fields.

**d) Login with register details for successful login   
A screenshot of a computer

Description automatically generated**

After entering the correct credentials user able to access the JobHunt content.

**e) After Successful login user able to access the content of Portal   
**

User able to access the featured jobs and able to apply the jobs.

**f) Job Filter   
A screenshot of a computer

Description automatically generated**

After typing the company click on search to get the results based on it.

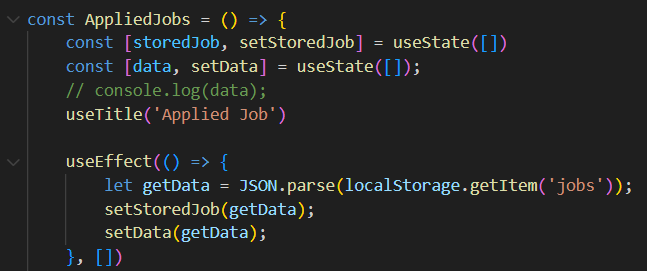
**A screenshot of a computer

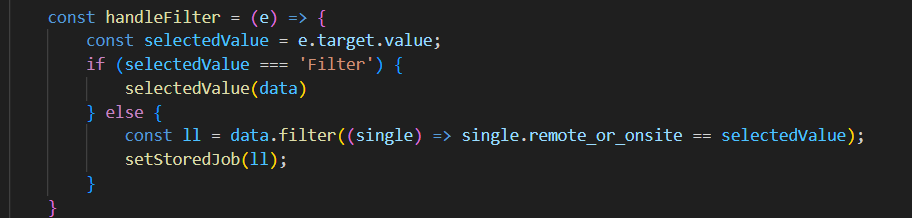
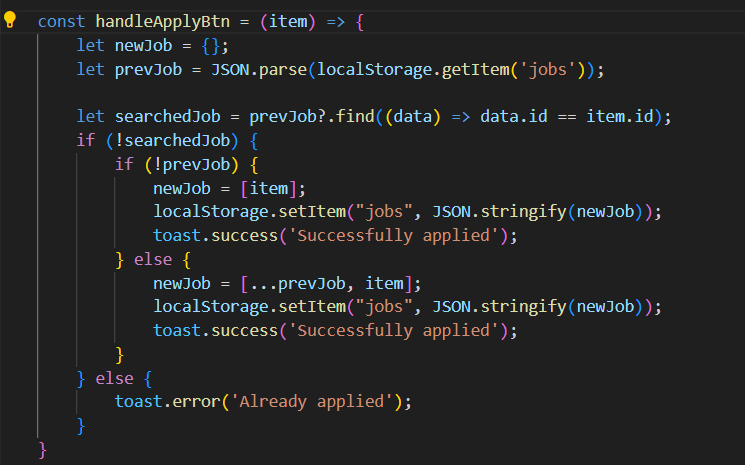
Description automatically generated**

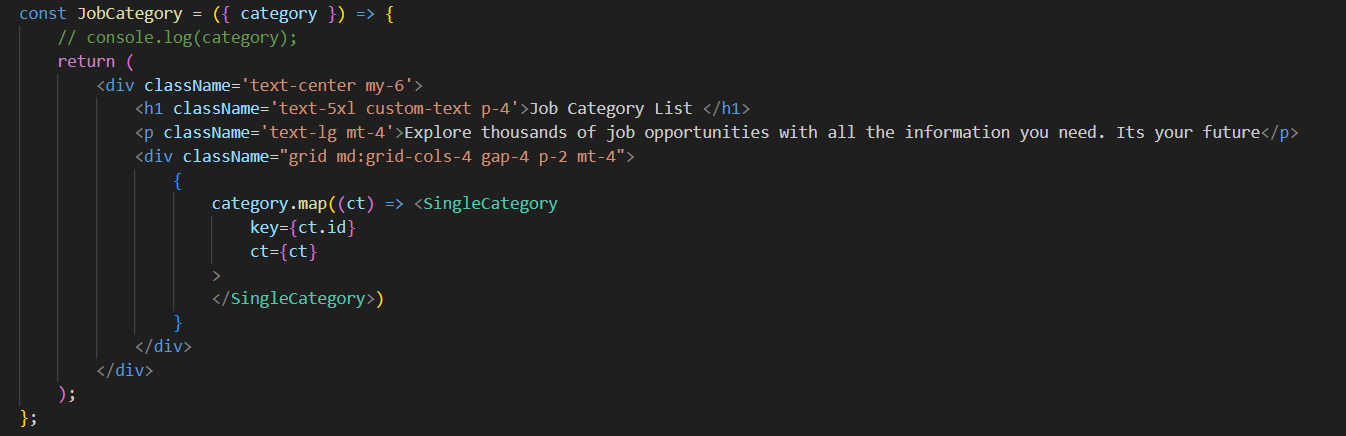
Based on company name jobs are filtered.

**b) Database Check:**

1. **Applied Jobs will be storing.**

****

**2)Filter Functionality  
  
  
  
3)Apply for Job  
**

**4)List All the Job Categories  
**

**User Manual:**

Please go to <http://localhost:5173/registerPage>

A screenshot of a computer

Description automatically generated

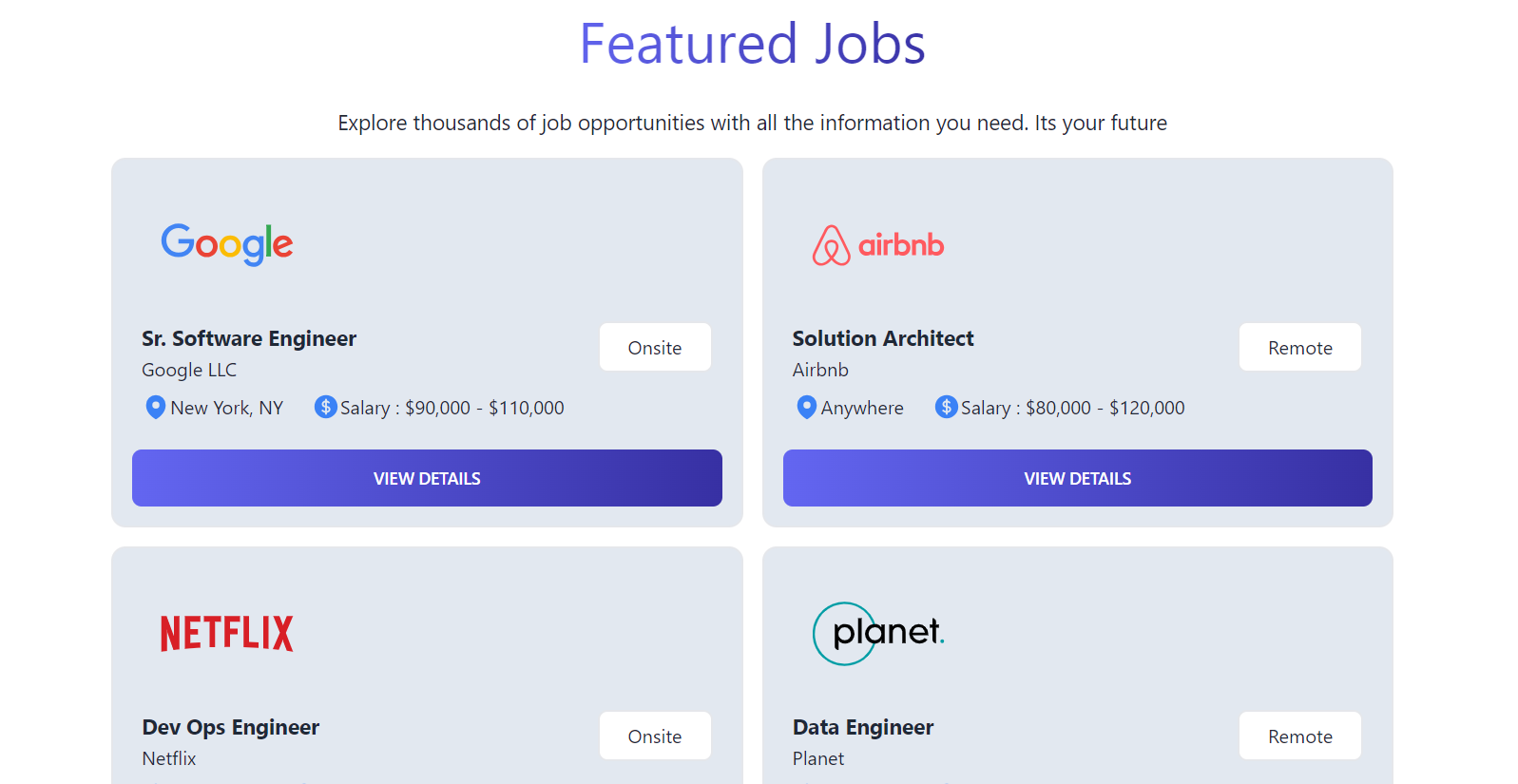
After Successful signup, application will redirect you to <http://localhost:5173/>

There please enter your details

A screenshot of a computer

Description automatically generated

If the entered details are accurate, you will be redirecting homepage <http://localhost:5173/homePage>   
A person with his arms crossed

Description automatically generated  
  
  
You can scroll down the home page, you find one section called Featured Jobs, where u can find featured jobs   
  


If you click on “Start Applying” it will redirect to all jobs page, where we can see all the listed jobs. <http://localhost:5173/allJobs>

A screenshot of a computer

Description automatically generated

Report Compilation Instructions:

**Steps for Installation:**

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/KudidalaPavan/ALPHA.git ` with the actual URL of the repository you want to clone:

```

git 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/Phase-1/

```

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.

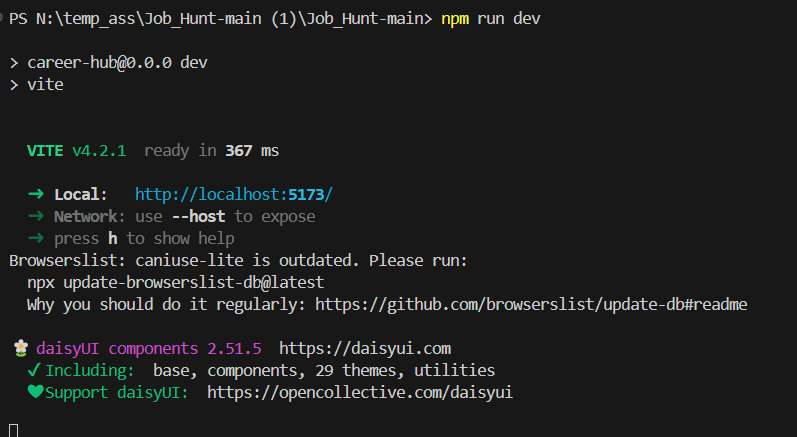
4. **Start the React Development Server:**

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

```

npm run dev

```

Looks something like this  


- 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:3000/` 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.

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.

|  |  |  |  |
| --- | --- | --- | --- |
| **Team member** | **Individual Contributions** | **Overall Contribution (%)** | **Note**  **(if applicable)** |
| Uday Kalyan Nuthalapati | User Authentication and Management | 13% |  |
| Pavan | Browse Job Listings | 11% |  |
| Nithin | Creating Testcases | 10% |  |
| Abhishek | Creating functionalities document | 11% |  |
| Keerthi | Functional requirements Update | 13% |  |
| Shravani | Working on User Manual | 10% |  |
| Vishal | Worked on UML Diagrams | 10% |  |
| Venkata | Worked on Sequence Diagram | 11% |  |

**Team Member Contribution:**