**PROJECT TITLE: JOBTRACKER**

**PROJECT MEMBERS**

**Rahul Trivedi**

**Vedanshu Joshi**

**Surbhi Sharma**

**FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS**

**Functional Requirements:**

1. Browser Extension, which specifies which jobs are clicked and applied. The database regarding jobs applied for will store and manage the information as well.

* The browser extension should have a feature that tracks job listings clicked on by the user, recording the job title, company, and date/time of the click.
* Users should be able to mark job listings as "Applied" within the extension, which will also be recorded in the database.
* The extension may provide an indicator (e.g., a visual icon) next to job listings to show whether the user has already clicked on or applied for a specific job.
* The extension should maintain a database to store information about job applications made by the user.
* This database should include details such as the job title, company, date of application, application status (in progress, under review, applied, rejected, etc.), and any notes or comments added by the user.
* Users should be able to easily access and manage their job application history through the extension's interface.

1. The System should allow for User Authentication.

* To ensure data privacy and security, the system should implement user authentication, requiring users to log in or create an account.
* User authentication can be implemented using secure methods like Google Single-Sign-On, allowing users to sign in using their existing social media or email accounts.
* Authenticated users should have access to their personalized job application data and settings.

1. The Browser Extension should be cross-browser compatible.

* The browser extension should be designed to work seamlessly across various popular web browsers, such as Chrome, Firefox, Edge, and Safari.
* Compatibility testing should be conducted to ensure that the extension functions consistently and without issues on different browsers.
* Updates and bug fixes should be applied promptly to maintain compatibility with new browser versions.

1. The job application should be able to be tracked from LinkedIn, whether the current job application is in progress, under review, already applied for, or rejected.

* The extension should integrate with LinkedIn to extract and display the status of job applications made through LinkedIn.
* Users should be able to link their LinkedIn profile to the extension, allowing it to access and track their job application activities on the LinkedIn platform.
* The extension should provide real-time updates on the status of LinkedIn job applications, including whether they are in progress, under review, applied for, or rejected.

1. Content Detection and Adaptation:

* The extension should be able to detect and adapt to different types of content and web page structures across various websites like Handshake, Indeed, LinkedIn, etc.
* It must provide consistent functionality, regardless of the website being visited.

1. Bookmark for Applied Jobs:

* The Browser Extension, upon completion of a job application, should bookmark the website to a particular folder in the browser. It enables the user to easily look up all the jobs they have applied for.

1. Trigger Reminder Notification:

* The Browser extension should track ‘pending’ (unfinished) applications and send a trigger as a reminder to the user to complete the application. The trigger should be sent once every 24 hours until the application is completed.

1. Display Important Job Information:

* The Browser Extension should collect important information based on keywords relevant to the specific job.
* It should display important job descriptions, salary, skills required, etc.

1. The user can add or delete the extension from the store or the browser.

* Users can easily install the extension from their browser's extension store, such as the Chrome Web Store or Firefox Add-ons.
* Uninstallation is straightforward and can be done directly from the browser's extension management page.
* Updates to the extension can be automatically pushed to users, ensuring they have the latest features and bug fixes

1. The Extension should have a static website that has getting started steps on the first page and users can access their data using different links from the navbar.

* The static website is bundled within the extension, so users can access it by clicking the extension's icon in the browser toolbar.
* It provides a user-friendly landing page with clear "Getting Started" instructions to guide users on how to use the extension.
* The navbar includes links to different sections of the website, making it easy for users to navigate and access the desired information.

1. Rewrite the html job tile element with application status tag if it’s already applied.

* The job listing section on the website displays a list of available job opportunities with relevant details such as job title, company, and location.
* Each job listing includes a status tag (e.g., "Applied") for jobs that the user has already applied for, helping users keep track of their applications.
* Users can click on a job listing to view more details about the job and access application options.

1. By clicking on hide applied job functionality, the applied job should not display in the list of available jobs.

* Users have the option to enable or disable the "Hide Applied Jobs" functionality through a toggle or button on the extension.
* When "Hide Applied Jobs" is enabled, the extension dynamically filters out and hides job listings with the "Applied" status, providing a cleaner view of available jobs.
* Disabling the feature restores the full job list, including applied jobs, for users who may want to review their application history.

1. The user can delete the job data if any data is entered incorrectly.

* Users can easily delete a job entry from their saved data by clicking the "Delete" or "Remove" button associated with each job listing.
* Confirmations or prompts may be provided to ensure users don't accidentally delete data and to confirm their intention to remove a job.
* Deleted data should be permanently removed from the user's data, ensuring that it doesn't appear in subsequent sessions.

**Non-Functional Requirements:**

**Scalability:**

1. User Base Scalability:

* The extension should be designed to handle a large user base.
* It should be able to efficiently serve a growing number of users without significant degradation in performance.

1. Resource Utilization:

* The extension should use system resources (CPU, memory) efficiently.
* It should not cause excessive strain on the user's device, even when handling a high load.

1. Backend Scalability (if applicable):

* If the extension relies on external servers or APIs, those components should be designed to scale horizontally to accommodate increased traffic.

**Performance**:

1. Response Time:

* The extension should respond quickly to user interactions.
* Actions like clicking buttons or opening the extension popup should have minimal latency.

1. Page Load Time:

* If the extension interacts with web pages, it should not significantly impact the page load times of the websites the user visits.

1. Resource Management:

* The extension should release resources (memory, network connections) when they are no longer needed to avoid memory leaks or network congestion.

1. Efficient Algorithms and Data Structures:

* The extension should use efficient algorithms and data structures to perform tasks.
* For example, if the extension processes a large dataset, it should use algorithms with acceptable time complexity.

**Security**:

* 1. Data Encryption and Protection:
  + All sensitive user data, including job application details and personal information, must be encrypted both in transit and at rest.
  + Implement strong encryption protocols (e.g., TLS) for data transmission between the extension and any remote servers or APIs.
* 2. Authentication and Authorization:
  + Implement secure user authentication mechanisms to prevent unauthorized access to job application data.
  + Utilize role-based access control (RBAC) to ensure that only authorized users can perform specific actions within the extension.
* 3. Secure Storage of Credentials:
  + If the extension stores login credentials for job search websites or email accounts, these credentials must be securely encrypted and stored using industry best practices.
  + Implement secure key management for credential storage.

4. Secure Storage of Local Data:

* Data stored locally on the user's device (e.g., job application history) should be encrypted and protected from unauthorized access.
* Implement proper access controls to limit data exposure.

5. Session Management:

* Manage user sessions securely, including secure session storage, session timeouts, and the ability to log users out after periods of inactivity.

**Reliability:**

1. Uptime:

* The browser extension should aim for a minimum uptime of 99% during standard operating hours, with allowances for scheduled maintenance.
* In case of planned maintenance or upgrades that may temporarily affect uptime, users should be notified in advance through the extension's notification system.

1. Availability:

* The availability of the extension should be measured as the ratio of successful user interactions (e.g., job application tracking) to the total number of interactions over a defined period.
* To maintain high availability, the system should employ load balancing to distribute user requests evenly across servers and minimize the risk of overload.

1. Response Time:

* The extension should aim to maintain a response time of less than 500 milliseconds for common user actions, such as marking a job as "Applied."
* Performance monitoring tools should be in place to track response times and alert administrators if response times exceed predefined thresholds.

1. Error Rate:

* The error rate should be kept below 1% for user interactions, ensuring that users encounter minimal errors while using the extension.
* Regular error analysis and debugging procedures should be conducted to identify and address recurring issues.

1. Fault Tolerance:

* The extension should be designed to gracefully handle and recover from minor errors or disruptions without causing service downtime.
* Automated health checks should be performed to detect faults and initiate recovery processes when necessary.