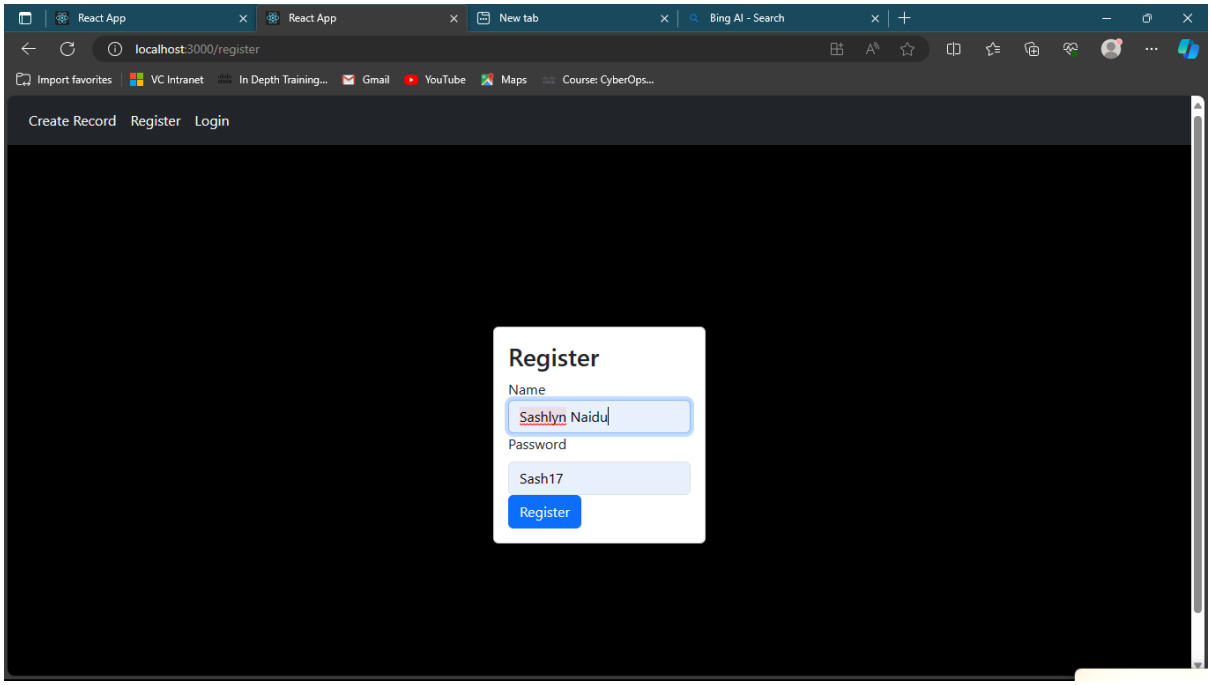
APDS7311

POE

Sashlyn
ST10084497@vcconnect.edu.za

Contents

Register

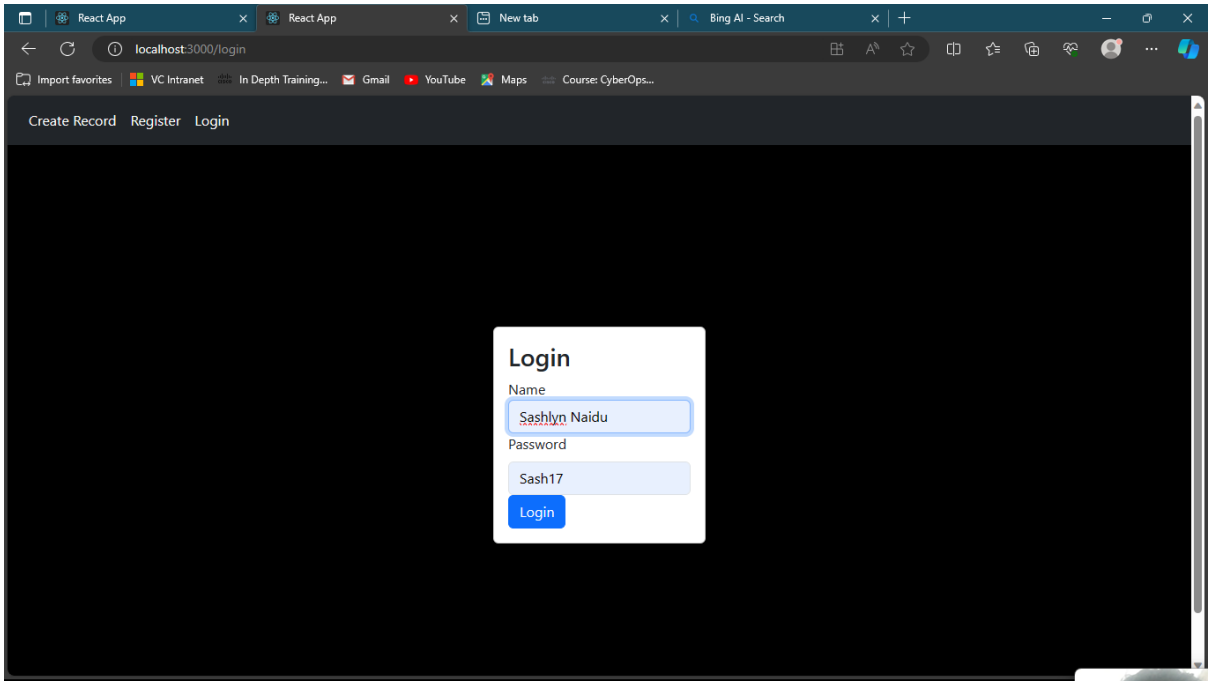


1. **Accessing the Registration Form:**
    - **Users initiate the process by accessing the dedicated registration form page on the application or website. This is typically achieved through navigation to a specified URL or a designated registration section.**
2. **Completion of Form Fields:**
    - **Upon arrival at the registration form page, users are presented with the form, encompassing the subsequent fields:**
        - **Name: Users input their name into this field.**
        - **Password: Users elect and input a password into this field.**
3. **Form Completion:**
    - **Users engage with the form by populating their particulars. To proceed, they must supply their name and a password.**
4. **Form Validation:**
    - **The form may incorporate validation mechanisms to ensure the provision of mandatory information. If any of the fields are left vacant, users might encounter validation errors or receive notifications prompting them to fulfill these obligatory requirements.**
5. **Submission of Registration Form:**
    - **Following the successful completion of the form, users execute the submission by clicking the "Register" or a corresponding submit button positioned at the form's base.**
6. **Registration Procedure:**

- **Subsequent to submission, the provided code undertakes the registration procedure. It conveys the user's registration information, inclusive of their name and password, to the application's server or backend for registration.**

7. **Error Handling:**
    - **In the event of any hitches during the registration process, such as server issues, network anomalies, or unforeseen errors, users may be confronted with an error message or an alert on the display, apprising them of the inability to conclude the registration process successfully.**
8. **Confirmation of Success:**
    - **Should registration be successful, users might receive a confirmation message or be redirected to another section within the application or website.**
9. **Navigation:**
    - **Depending on the application's blueprint and routing, users might find themselves redirected to a disparate part of the application or the primary screen.**
10. **Supplementary Features:**
    - **The registration form may be an integral component of a more intricate application equipped with supplementary functionalities. Following registration, users may be presented with diverse actions and navigation alternatives, such as accessing user profiles, logging in, or exploring the application's services.**

Login

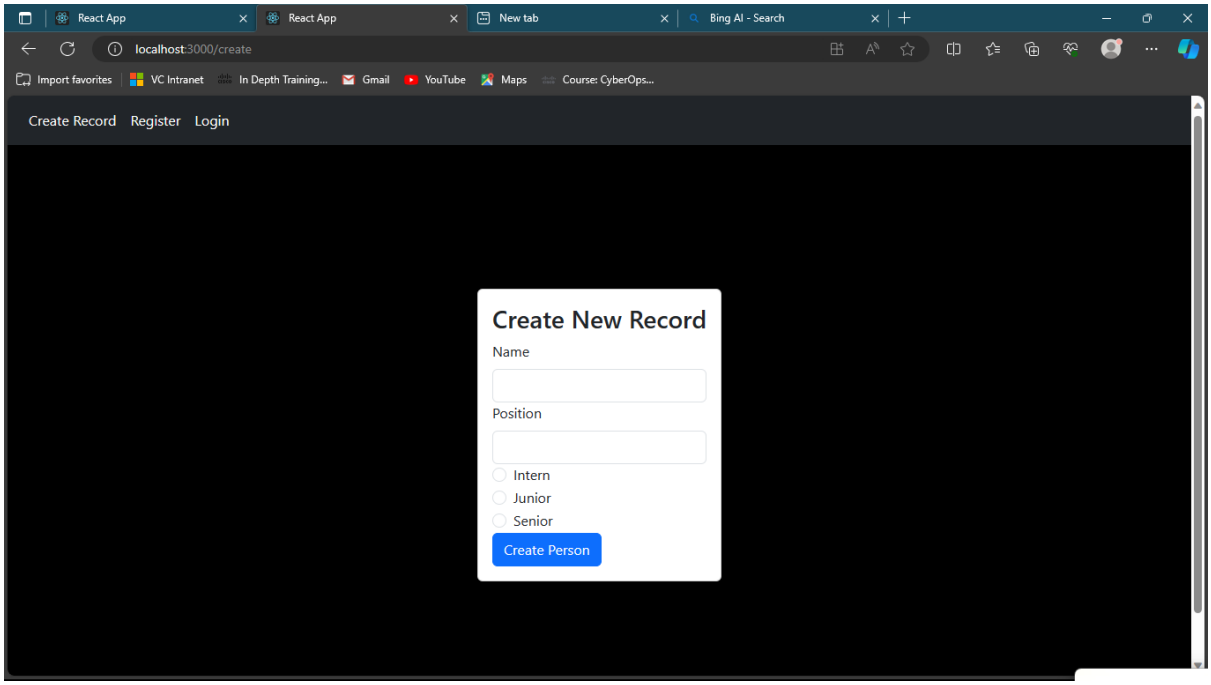


1. **Accessing the Login Form:**
    - Users commence by reaching the login form, usually through a designated URL or login section on the application or website.
2. **Form Fields:**
    - Upon arrival at the login form page, users are presented with the form, including the following fields:
        - **Name (Username):** Users enter their username.
        - **Password:** Users input their password.
3. **Completing the Form:**
    - Users interact with the form by providing their username and password, satisfying the prerequisites for logging in.
4. **Form Validation:**
    - The form may incorporate validation checks to guarantee mandatory details are supplied. Leaving fields empty could trigger validation errors or alerts, urging users to complete the necessary fields.
5. **Submitting the Login Form:**
    - Upon correctly entering their username and password, users click the "Login" button or a corresponding submit button located at the form's lower section.
6. **Login Procedure:**

- Subsequent to submission, the code processes the login data. It dispatches the user's login information, including their username and password, to the application's server for authentication.

7. **Handling Server Responses:**
    - The code manages the server's response following the login request. It anticipates a JSON response, logs the data to the console, including user data and a token. The token is stored in the user's browser's local storage, commonly for authentication and authorization purposes.
8. **Error Handling:**
    - In the event of issues during the login process, like server glitches or network difficulties, users may encounter an error notification, indicating that the login process couldn't be successfully completed.
9. **Confirmation of Success:**
    - In the case of a successful login, users may be directed to another section of the application or website. The precise destination may differ, contingent on the application's structure and routing.
10. **Added Functionality:**
    - The login form is often an element of a more comprehensive application with supplementary features. Following a successful login, users may access various actions and navigation options, such as personalized content, user profiles, or account settings.

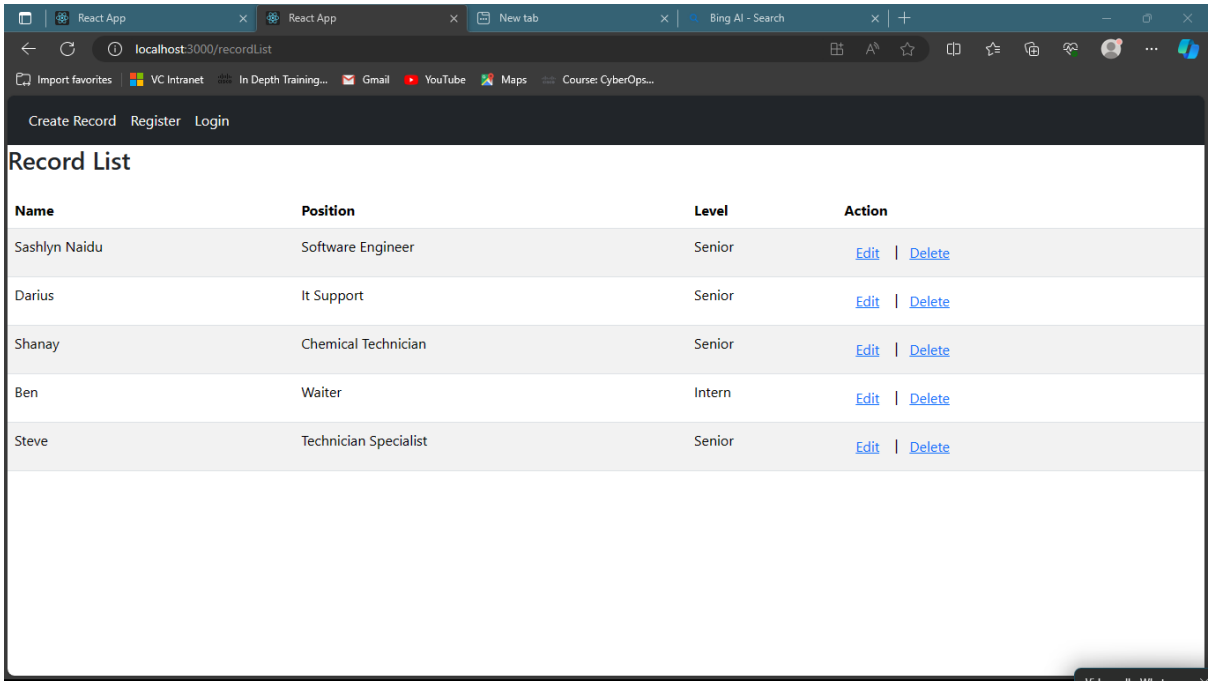
Creating New Record



1. **Accessing the "Create New Record" Form:**
    - Users start by accessing the "Create New Record" form, typically available through a specific URL or section dedicated to record creation.
2. **Authentication Token:**
    - The code checks if the user has an authentication token stored in their local storage. This token is essential for securing the record creation process.
3. **Form Fields:**
    - Upon reaching the "Create New Record" form, users will encounter the following input fields:
        - **Name:** Users enter the name of the person for the new record.
        - **Position:** Users specify the person's position.
        - **Position Level:** Users choose the position level from options like "Intern," "Junior," or "Senior."
4. **Completing the Form:**
    - Users interact with the form by providing the required information. They enter the name, position, and select the appropriate position level.
5. **Form Validation:**
    - The form may include validation checks to ensure users provide the necessary information. Errors or alerts may appear if essential fields are left empty.

6. **Submitting the Form:**
    - After successfully completing the form, users initiate the submission process by clicking the "Create Person" button or a corresponding submit button located at the bottom of the form.
7. **Record Creation Process:**
    - Following submission, the code processes the data provided by the user. It sends the new person's information, including the name, position, and position level, to the server for record creation.
8. **Handling Server Responses:**
    - The code manages the server's response after the record creation request. It expects a JSON response and may log this data, including details about the newly created record.
9. **Error Handling:**
    - If any issues arise during the record creation process, such as server errors or network disruptions, users may see an error alert, indicating that the creation process was unsuccessful.
10. **Confirmation of Success:**
    - In the case of a successful record creation, users may be directed to another section within the application, such as a list of records or relevant information.
11. **Additional Features:**
    - The record creation form is typically part of a more extensive application with supplementary features. After creating a new record, users may access various actions and navigation options, such as viewing a list of records, editing records, or returning to the main application interface.

Record List



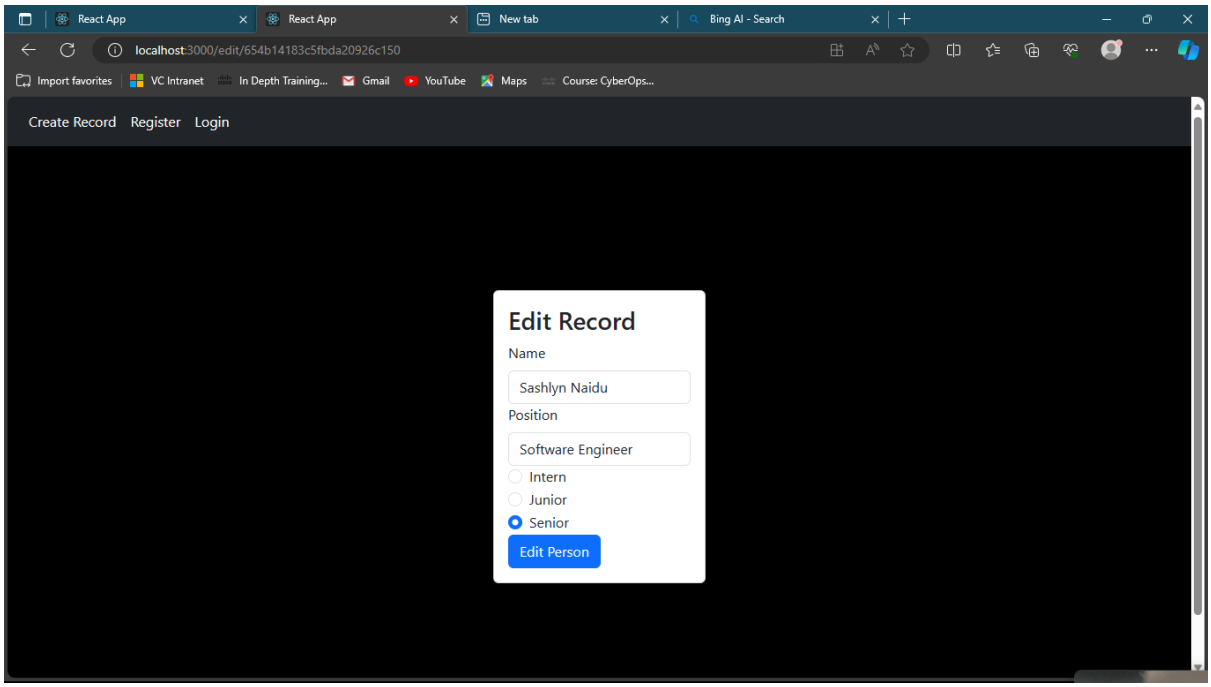
1. **Accessing the Record List:**
    - Users initiate their interaction with the Record List page, which serves as a repository of records. This page is typically accessible through a specific URL or a dedicated menu option within the application.
2. **Browsing the Record List:**
    - Upon reaching the Record List, users are presented with a table that showcases the records, featuring columns for "Name," "Position," "Level," and an "Action" column for executing edit and delete actions.
3. **Inspecting a Record:**
    - Users have the ability to inspect the particulars of each record within the displayed table, including details like the name, position, and level. Each record's data is arranged in a row.
4. **Modifying a Record:**
    - Users can initiate changes to a specific record by selecting the "Edit" button linked to that record. Clicking this button redirects users to a distinct page where they can make alterations to the record's specifics.
5. **Removing a Record:**
    - Users possess the capability to delete a record by activating the "Delete" button adjacent to the record they intend to eliminate. This action triggers a request to the server, instructing it to delete the record.
6. **Confirmation of Deletion:**

- Following the deletion of a record, users receive confirmation of the action, and the record vanishes from the Record List.

7. **Navigation Choices:**
    - Users can navigate through the list of records to explore additional records, make modifications, or perform deletions. The design may incorporate features such as pagination or scrolling, depending on the specific implementation.
8. **Error Handling:**
    - In case of any irregularities, such as difficulties in fetching records or issues during record deletion, users may encounter error notifications or messages, keeping them informed about the encountered issues.

.

Edit Record



1. **Accessing the Edit Record Page:**
    - Users begin their interaction by accessing the "Edit Record" page, typically available through a specific URL or an option in the application's menu.
2. **Authentication Token:**
    - The code checks for the presence of an authentication token in the user's local storage to ensure a secure record editing process.
3. **Retrieving Existing Record Data:**
    - Upon entering the "Edit Record" page, the code retrieves the current record data associated with the record being edited. This data includes details like the name, position, and position level.
4. **Viewing and Modifying Record Details:**
    - Users are presented with a form containing fields for editing record details. They can view the existing record information and make alterations as required.
5. **Form Validation:**
    - The form may include validation checks to ensure that users provide the necessary information. Validation errors or alerts may be displayed if mandatory fields are left empty or if data format is incorrect.
6. **Making Revisions:**
    - Users interact with the form to make changes to the record's details. This includes editing the name, position, and selecting a new position level.
7. **Submitting the Edit:**

- After making adjustments, users initiate the submission process by clicking the "Edit Person" button or an equivalent submit button located at the form's bottom.

8. **Updating Record Data:**
    - The code sends a PATCH request to the server, transmitting the updated record data. This request triggers the update of the record information within the database.
9. **Confirmation of Success:**
    - Users receive feedback confirming the successful update of the record. This may involve a confirmation message or redirection to another page.
10. **Error Handling:**
    - In the event of issues, such as problems with the update request or network interruptions, users may encounter error notifications or messages to inform them of the encountered difficulties.

References

https://www.mongodb.com/

https://chat.openai.com/