

INDUSTRIAL TRAINING REPORT

Name: Rajorshi Ghosh

College: Future Institute of Engineering
and Management

Training details: Node JS Training from
Internshala Trainings

Training Duration: 6 weeks

Project Name: MongoDB CRUD App

MongoDB CRUD App

This web application demonstrates the Create, Read, Update, Delete (CRUD) and search operations on MongoDB database using AJAX requests connected via a Node server running Express framework. This application can be used to digitally store singular data values with each a unique ID to track.

This application was developed as a part of industrial training provided by Internshala.

Aim:

To demonstrate CRUD operations on MongoDB using an express webapp.

Characteristics:

- Platform Independent
- Large Storage
- Cloud Database

Programming Languages used:

- HTML
- CSS
- JavaScript

Framework used:

- ExpressJS

Database:

- MongoDB


Server:

- NodeJS

Features:

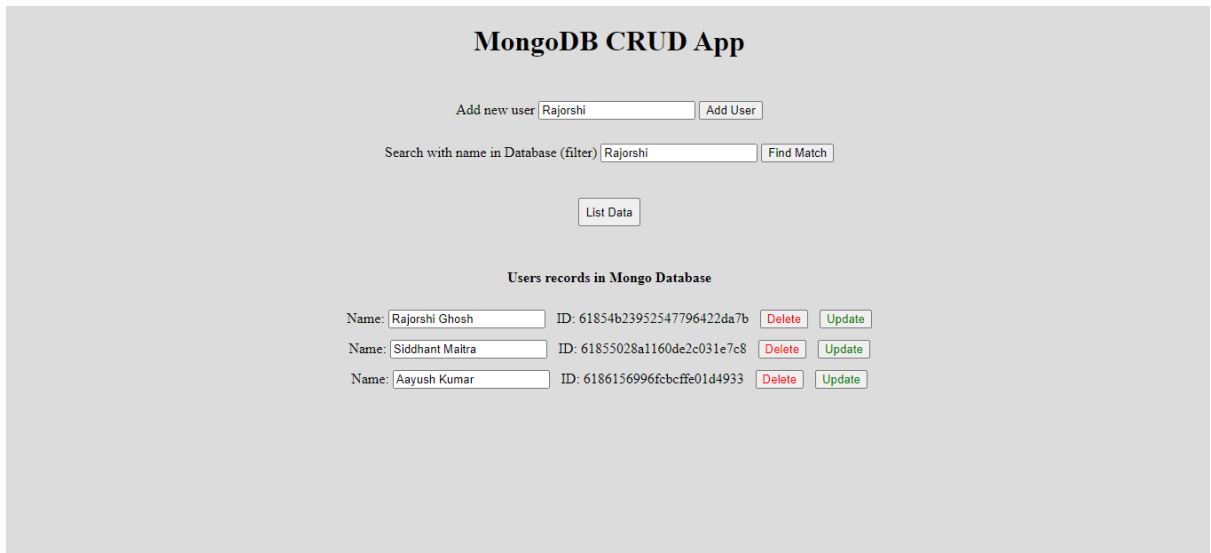
1. Read records:

On visiting the website the user will connect to the online **MongoDB** database and see this screen.



The screenshot shows the 'MongoDB CRUD App' interface. At the top, there's a title 'MongoDB CRUD App'. Below it, there are two input fields: 'Add new user' with the value 'Rajorshi' and an 'Add User' button. Below that, there's a search section with 'Search with name in Database (filter)' and a text input with 'Rajorshi', followed by a 'Find Match' button. At the bottom, there's a 'List Data' button.

If the user clicks the “List Data” button a **fetch request** will be sent to get the existing records from the Database and will be **dynamically rendered** below on the page.



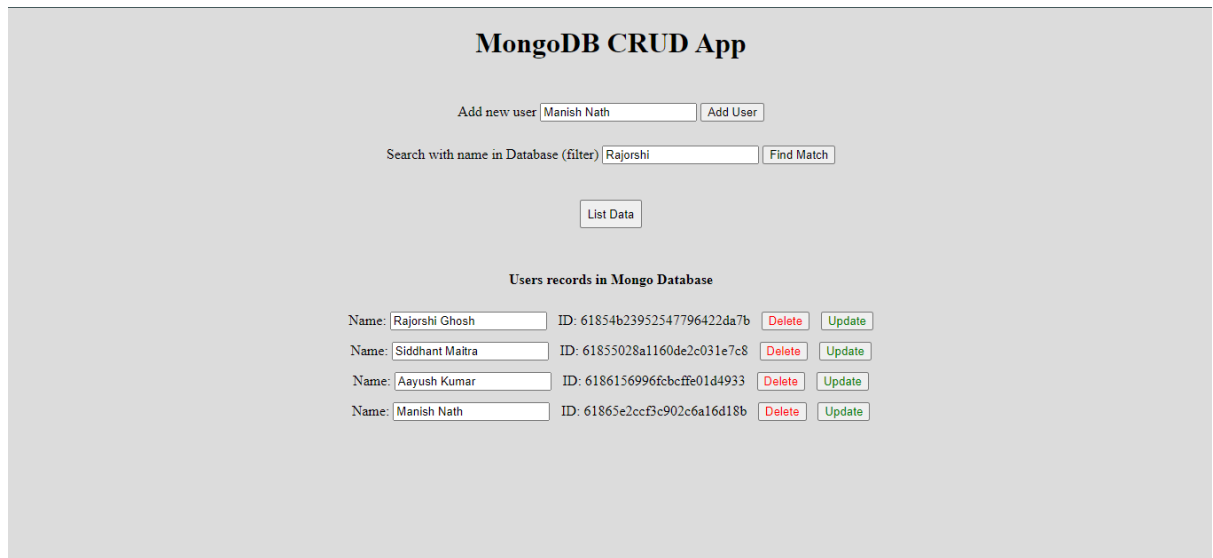
The screenshot shows the 'MongoDB CRUD App' interface after clicking the 'List Data' button. The top section remains the same. Below the search section, there's a 'List Data' button. Underneath, there's a heading 'Users records in Mongo Database'. Below this heading, there's a table with three rows of user data. Each row has a 'Name' field, an 'ID' field, and two buttons: 'Delete' (red) and 'Update' (green).

Name	ID	Delete	Update
Rajorshi Ghosh	61854b23952547796422da7b	Delete	Update
Siddhant Maltra	61855028a1160de2c031e7c8	Delete	Update
Aayush Kumar	6186156996fcbcf01d4933	Delete	Update

2. Create a record:

The app provides the user with an input field beside “Add new user” caption to enter the new record name and “Add User” button beside it to add it to the database.

After adding a record the user must press “List Data” button to observe the new record being added to the list.



MongoDB CRUD App

Add new user

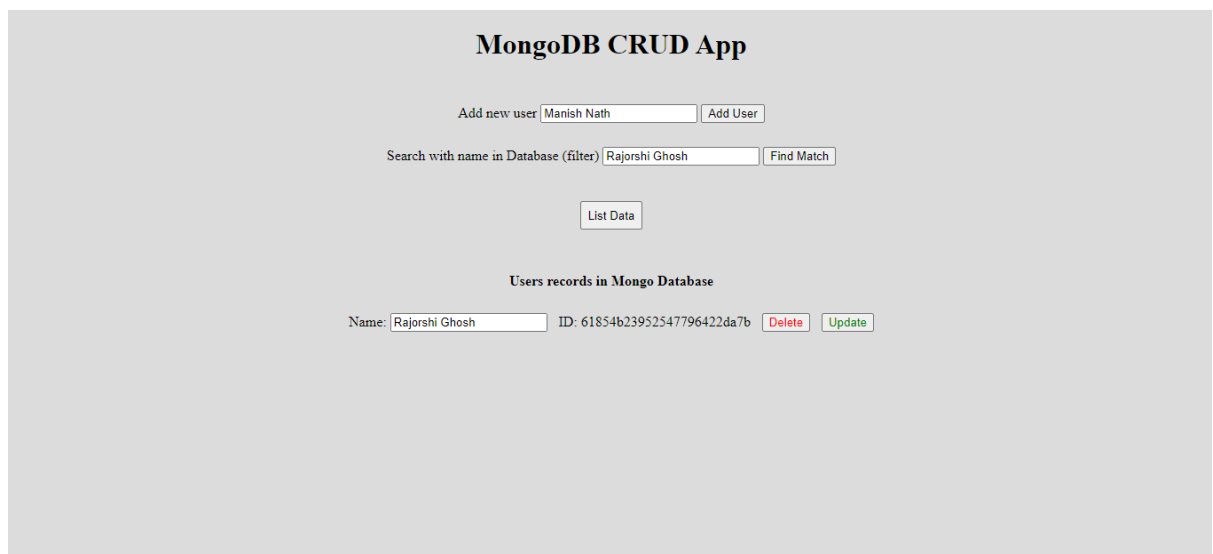
Search with name in Database (filter)

Users records in Mongo Database

Name: <input type="text" value="Rajorshi Ghosh"/>	ID: 61854b23952547796422da7b	<input type="button" value="Delete"/>	<input type="button" value="Update"/>
Name: <input type="text" value="Siddhant Maitra"/>	ID: 61855028a1160de2c031e7c8	<input type="button" value="Delete"/>	<input type="button" value="Update"/>
Name: <input type="text" value="Aayush Kumar"/>	ID: 6186156996fcbccfe01d4933	<input type="button" value="Delete"/>	<input type="button" value="Update"/>
Name: <input type="text" value="Manish Nath"/>	ID: 61865e2ccf3c902c6a16d18b	<input type="button" value="Delete"/>	<input type="button" value="Update"/>

3. Search for a record:

The user can search for any specific record by entering the name details in the input field beside “Search with name in Database” caption and pressing the “Find Match” button beside it. If the record/s is found in the database they will be displayed below.



MongoDB CRUD App

Add new user

Search with name in Database (filter)

Users records in Mongo Database

Name: <input type="text" value="Rajorshi Ghosh"/>	ID: 61854b23952547796422da7b	<input type="button" value="Delete"/>	<input type="button" value="Update"/>
---	------------------------------	---------------------------------------	---------------------------------------

4. Updating a record:

Any record can be updated by the user by editing the current record name shown in the input field of the record beside “Name:” caption.

To update with the new value the user must press the “Update” of the record of work. The user can list all data again to confirm their updating.

The screenshot displays the MongoDB CRUD App interface. At the top, there's a title "MongoDB CRUD App". Below it, there's a form to "Add new user" with a text input field containing "Manish Nath" and an "Add User" button. Below that, there's a search section with the text "Search with name in Database (filter)" and a text input field containing "Rajorshi Ghosh", followed by a "Find Match" button. Below the search section, there's a "List Data" button. The main section is titled "Users records in Mongo Database" and contains a table of user records. Each record has a "Name" input field, an "ID" field, and "Delete" and "Update" buttons. The records are:

Name	ID	Delete	Update
Rajorshi Bose	61854b23952547796422da7b	Delete	Update
Siddhant Maltra	61855028a1160de2c031e7c8	Delete	Update
Aayush Kumar	6186156996fcbcff01d4933	Delete	Update
Manish Nath	61865e2ccf3c902c6a16d18b	Delete	Update

5. Deletion of record:

Any record can be deleted from the database by the user by clicking the “Delete” button of the record. The user can list all data again to confirm their deletion.

This screenshot is identical to the one above, showing the MongoDB CRUD App interface. It includes the "Add new user" form, the search section, the "List Data" button, and the "Users records in Mongo Database" table with four records, each having "Delete" and "Update" buttons.

Future Scope:

Few improvements can be done on this web application to make it more feature rich in future.

Some of the points are mentioned below:

1. At present the web application is running on on-premise servers. In near future the web application can be deployed on cloud to make the application more scalable and faster.
2. For designing, Bootstrap/Tailwind CSS framework can be used to make it look more pleasing and user friendly.
3. One can optimize the client-side scripting code with the help of React/Angular framework.

Conclusion:

The Application successfully focuses on Back-End development and exhibits the basic functionalities of interacting with a database and has room for improvements and utilizing technologies learnt through out this training more full-fledged applications using similar tech-stacks can be developed.
