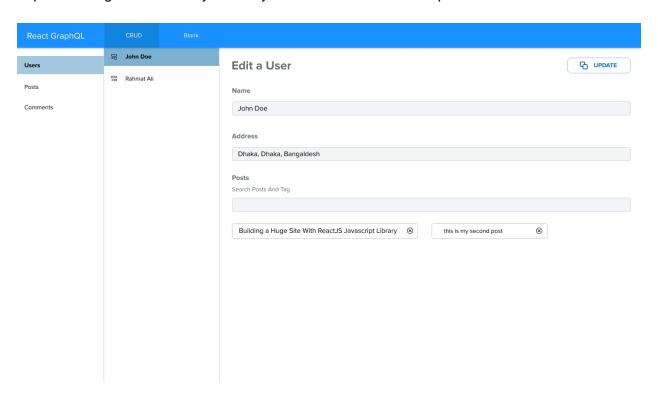**SHIKHO TECHNOLOGIES BANGLADESH LTD**

RANGS PARAMOUNT SQUARE; FLOOR 11, HOUSE 11, ROAD 17, BANANI C/A, DHAKA-1213, BANGLADESH

The following program is designed to test the basic skills for ReactJS along with the react core components, project basic architecture & GraphQL through a simple CRUD app. Let's see how much you think in React.

Design File:

https://www.figma.com/file/Uyuh5lLZcyucx96ifKO2GP/React-GraphQL-Test?node-id=0%3A1

Story:

Imagine you are building a blog site with GraphQL and you need a Panel to insert your data. You have **Users** Data, **Post** Data & **Comment** Data. While writing **Post** a user is able to tag **Comments** in the post. While Editing **Users** data you can tag **Posts** and vice versa. Explore the design in Figma to get a better idea.

Application Summary:

It's a Single Page Application. It has two pages connected to the UI Route (CRUD, Blank).

In <u>CRUD</u> Page,
- First Column has the Name of Models (**Users**, **Posts**, **Comments**).
- Second Column is depended on the first column on click, and It fetches all the Data connected to the first column. Ex, If you click **Users** it second column will have the list of all users.
- Third column is a Form to edit the Single Entity of the Data.
  - Each editing form has fields related to the model itself along with a typeahead Input field. For example, While editing **Posts** typeahead field will be able to search for existing **Comments**. While editing the **Users** the typeahead input field will be able to search for existing **Posts** and vice versa. (See Figma Design for better Idea)
  - **Comments** has Backward Connection to Posts which means while editing a comment a user can tag a Single Post to It.

In <u>Blank</u> Page,
- This page is empty !

Tools :
Use any of your favourite Graphql Explorer to explore all the queries & mutations. **No data inserted in the projects so, Please insert some demo data of your own via mutation.** Use only the query & mutation & parameter that is needed or required.

GraphAPI : https://api.apito.io/secured/graphql
Authorization : Bearer
6rZvWY1K4jgvV6WNVR93c7RXxXeI6CwetOZv23K6sGbu3VhRIAEA2Zb3wlsDCqIaMAjkxyKA9ABNYI
KKwl2HgxvQYSxtQqkGVlW0VuR4ugGHWx0LFQvCotao0jfaFIg8T0oBcyCvylRs5NRcAY5L3ho5wQObO
Dx

Conditions:
- Use any UI lib you want. We prefer *Ant Design*
- Use any React framework you want or Use Default Create React App. We prefer *Next.js*
- Use only *Apollo Client* for Store & State

- Deploy static files with *Vercel* to show us the final result. Submit the code via public github link. (Avoid hardcoding the token in the app. Load it from ENV variable)

Timeline: 5 days

Bonus: Create/Update is Missing from the Design. So, try adding a Create Data Modal for each model.

Questions & Submissions: Through Email

GraphQL Query Samples: Use any Graphql Playground or Explorer (https://github.com/graphql/graphql-playground) you like to explore the variables

Query & Mutation Names:

```
mutation MyMutation
  ▶ createComment
  ▶ createPost
  ▶ createUser
  ▶ deleteComment
  ▶ deletePost
  ▶ deleteUser
  ▶ updateComment
  ▶ updatePost
  ▶ updateUser
  ▶ userLogin

query MyQuery
  ▶ comment
  ▶ comments
  ▶ commentsConnection
  ▶ post
  ▶ posts
  ▶ postsConnection
  ▶ user
  ▶ users
  ▶ usersConnection
```

Sample Queries (More queries are available in the graphql endpoint. Use any graphql explorer to explore)

```graphql
query SampleQueries {
  users {
    id
    data {
      email
      first_name
      phone
    }
  }
  posts {
    id
    data {
      title
      body {
        text
      }
    }
  }
  user(_id: "") {
    id
    data {
      email
      first_name
      phone
      avatar {
        url
      }
    }
  }
  post(_id: "") {
    id
    data {
      title
      body {
```

```
        text
      }
    }
  }
  comments {
    id
    data {
      body {
        text
      }
    }
  }
  comment(_id: "") {
    id
    data {
      body {
        text
      }
    }
  }
}
```

Sample Mutations (More mutations are available in the graphql endpoint. Use any graphql explorer to explore)

```
mutation MyMutation {
  createPost(payload: {title: "", body: {html: ""}}) {
    id
  }
  createComment(payload: {body: {html: ""}}) {
    id
  }
  updatePost(_id: "") {
    data {
      title
      body {
        text
      }
```

```
    }
    id
  }
  userRegister(phone: "", secret: "") {
    id
    id_token
    refresh_token
  }
}
```

Submission Deadline: 11pm, 28 January 2022

Submission Email: fahim.shoumik@shikho.tech