

Data Foundation Systems

Discussion Forum
Integration

Meet our Team: Pops&Pills

Laksh Balani
2020102019

Ishanya
Sethi
2020102014

Knv
Karthikeya
2020102003

- Worked on UI, frontend. Made the rich text box and worked on the likes and dislikes part
- Worked on the API calls, backend and hierachial structure of comments and profanity filter. Also helped in UI part
- Worked on the database and schema building and storing data in MySQL.



Project Scope and Our Objective

Project Scope

- Integration of an open source discussion forum (e.g. NodeBB, Discourse, Talkyard) into an existing web application (Data Foundation).
- A platform for users to discuss data sets, models, and blog posts related to the Data Foundation.
- Create a simple and intuitive interface that enables users to engage in meaningful discussions about data-related topics within the project's scope.
- Implementation of a toxic comment classification and profanity filter to maintain a healthy discussion environment





Project Scope and Our Objective

Objectives

- Integrate a discussion forum software system into the Data Foundation web application.
- Develop a user-friendly interface that allows users to easily discuss data-related topics.
- Use ReactJS and NodeJS technologies to develop the software system.
- Ensure the software system is scalable, reliable, and secure.
- Test and deploy the software system to a live server.
- Implement a toxic comment classification and profanity filter to filter out inappropriate content





TechStack Used

Programming Languages

- ReactJS
- ExpressJS

Database

- MySQL

Frameworks Used

- NodeJS

Libraries Used

- **emotion/react**
- **emotion/styled**
- **@mui/core**
- **@mi/icons-material**
- **@mui/material**
- **@testing-library/jest-dom**
- **@testing-library/react**
- **@testing-library/user-event**
- **axios**
- **cors**
- **draft-convert**
- **draft-js**
- **draft.js**
- **draft.js-to-html**
- **html-react-parser**
- **leo-profanity**
- **mysql**
- **react**
- **react-dom**
- **react-draft-wysiwyg**
- **react-draft-wysiwg-a**
- **react-scripts**
- **react-usestateref**
- **web-vitals**



Functional Requirements

01

Threads and Replies

The system shall allow users to create new threads and reply to existing posts.

02

Rich Text Editor

Provide a rich text editor that allows users to format their posts with different fonts, colors, and styles, and add images or other media.

03

Edit and deletion

The system shall allow users to edit or delete their own threads and replies.

04

Upvotes and downvotes

The system shall allow users to upvote or downvote threads and replies

05

Hierarchical Structuring of Threads

Implement a hierarchical structure for threads and replies, where replies are nested under the thread they belong to.

06

Profanity Filter

The system shall implement a profanity filter to prevent users from posting inappropriate content.



Non-Functional Requirements

Performance

The system shall be able to handle a high volume of traffic and database queries with minimal response time.

Scalability

The system architecture shall be designed to scale horizontally or vertically to meet increasing traffic and database load.

Usability

The user interface shall be user-friendly, intuitive, and responsive to provide a good user experience.

Availability

The system shall be available 24/7 with minimal downtime for maintenance and upgrades.

Compatibility

The system shall be compatible with different web browsers, devices, and screen sizes.

Schemas Used



Field	Type	Description
id	VARCHAR(255)	The id of the forum
commentsjson	longtext	Hierarchical tree like structure for comments

← **Database Schema**

Field	Type	Description
id	NUMBER	The id of the comment
content	STRING	Contains the data given by the user in the comment section
items	OBJECT	Replies to a particular comment
author	STRING	token of the user who has written the comment
authorName	STRING	name of the user who has written the comment
authorEmail	STRING	email of the user who has written the comment
likes	NUMBER	Number of likes to a particular comment or reply
dislikes	NUMBER	Number of dislikes to a particular comment or reply
likers	OBJECT	The list of likers to a particular comment or reply
dislikers	OBJECT	The list of dislikers to a particular comment or reply
last_updated	OBJECT	The Date-Time when the comment was last updated

← **Comment Schema**

API Endpoint

1. Create a new thread

```
/api/init
```

The body of the request should be a JSON object with the following fields:

```
{
  "id": "forum_id"
}
```

2. Get a thread by id

```
/api/get
```

The body of the request should be a JSON object with the following fields:

```
{
  "id": "forum_id"
}
```

API Endpoint

3. Update a thread

```
/api/create
```

The body of the request should be a JSON object with the following fields:

```
{
  "id": "forum id",
  "commentsjson": "hierarchical tree like structure for comments"
}
```

All the above endpoints are POST requests.

Integrating the forum

1. Run npm install lakshbalani/Pops-Pills in the folder
2. Import the DFSNestedComments component
3. It requires 3 parameters
 - a. ForumID
 - b. user: token of user logged in (taken from local storage)
 - c. email: email of user logged in (taken from local storage)
4. Initialize the database using the /init api call when using the particular forumID for the first time.

Demo of the website

Type your comment here...

COMMENT

Laksh Balani
Last updated at 13:36 13 May, 2023

Hey

1 like 0 replies

▲ 1 Reply

Laksh Balani
Last updated at 13:36 13 May, 2023

This is a reply

0 likes 0 replies

▲ 1 Reply

Ishanya Sethi
Last updated at 13:42 13 May, 2023

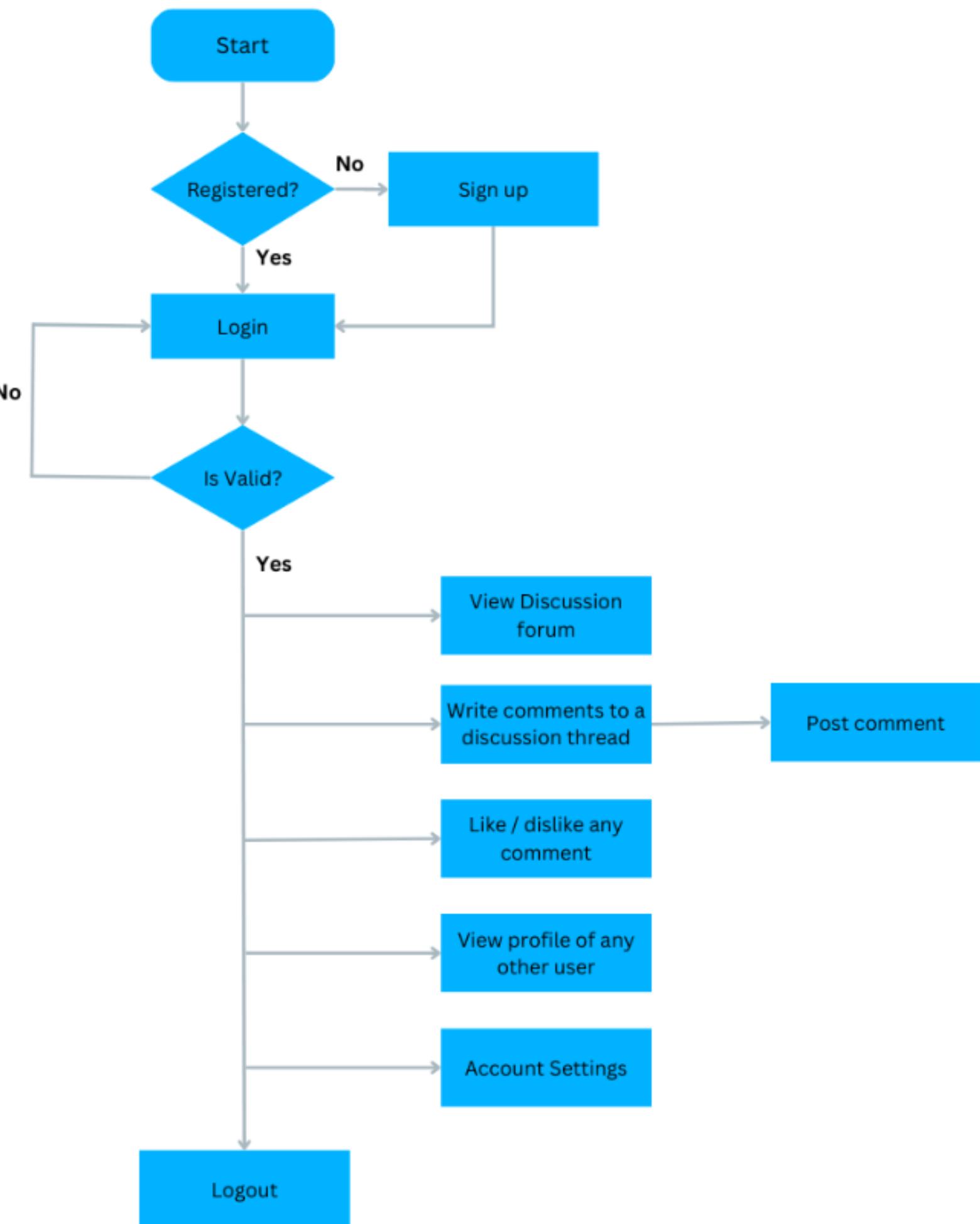
Reply from another account

1 like 0 replies

▲ 1 Reply

...

User workflow





Thank You