

**OBJECT ORIENTED SOFTWARE
ENGINEERING LAB**

(Practical file)

IT-359

**BACHELOR OF TECHNOLOGY
INFORMATION TECHNOLOGY**

**University School of Information,
Communication and Technology**



**GURU GOBIND SINGH INDRAPRASTHA
UNIVERSITY, DWARKA**

Submitted to:

Dr. S.S.Sodhi

Submitted by:

Apoorv Aron

01316401520

B.Tech, IT 5TH SEM

Problem Statement: (Socialize)

Nowadays, there are many social media platforms but the platform for interaction between college students has not been brought into the market yet.

The software will be a perfect platform for college students to contact and seek guidance from their seniors and alumni group from their college in a professional manner.

The goal is to create a platform that allows users to connect and share content with each other. The platform should allow users to create profiles, follow each other, post and interact with text, images, and receive notifications about activity related to their account.

Additionally, the platform should have moderation tools to allow administrators to enforce community guidelines and protect against inappropriate or spammy content.

The web app "Socialize" is a platform where students can connect and follow each other. The users can post, view and share their resources with each other. The users can follow other users and view their posts.

The unique selling point for our product is that a social media platform where alumni can guide juniors and post opportunities for students to grab.

The privacy of users will be maintained through proper authentication i.e, they can login & logout. This tool of ours intend to build social relation by making it easier for users to reach out to other people to seek help

Software Requirements Specification

for

Socialize

Version 1.0

Prepared by

Apoorv Aron (01316401520)

DATE : 24th November, 22

Table of Contents

1. Introduction	5
1.1 Purpose and Intended Audience	5
1.2 Project Scope.....	5
1.3 Terms, Definitions, and Acronyms	Error! Bookmark not defined.
1.4 References	6
2. Overall Description.....	7
2.1 Product Perspective	7
2.2 Product Features.....	7
2.3 User Classes and Characteristics	7
2.4 Operating Environment.....	7
2.5 Design and Implementation Constraints	7
2.6 Assumptions and Dependencies	8
3. System Features.....	8
4. Non-Functional Requirements.....	8
5. External Interface Requirements	9
5.1 User Interfaces.....	9
5.2 Hardware Interfaces.....	9
5.3 Software Interfaces	9
5.4 Communication Interfaces.....	9
6. Detailed Use Cases.....	10
7. Appendix	Error! Bookmark not defined.

Revision History

Name	Date	Reason For Changes	Version
Apoorv	20-10-22	Additions in project scope	1.1

1. Introduction

Nowadays, there are many social media platforms but the platform for interaction between college students has not been brought into the market yet. The software will be a perfect platform for college students to contact and seek guidance from their seniors and alumni group from their college in a professional manner.

The web app "Socialize" is a platform where students can connect and follow each other. The users can post, view and share their resources with each other. The users can follow other users and view their posts.

1.1 Purpose and Intended Audience

There are many social media platforms but the perfect platform for college students has not been brought into the market yet. The software will be a app for college students to contact and seek guidance from their seniors and alumni group from their college in a professional manner.

To create a platform where professionals can connect with one another and build networks within their industry. To facilitate the sharing of industry-specific knowledge and information among professionals. To help professionals discover new job opportunities and recruit new talent. To create a community of professionals who can support and encourage one another in their careers. To offer a way for professionals to stay up-to-date with the latest trends and developments in their industry.

Audience of this SRS are other project developer, users like college students , College alumni network and College staff. This SRS contains detail description about the product, its functioning, different external interfaces required, system features, Nonfunctional requirements, and some additional requirements.

1.2 Project Scope

The product will hold information stored by the user or shared by the user on the platform. It will not contain complete details about the background and current profession of the alumni members. The software will contain the basic details like name, batch, course and year of graduation. The software will not contain the topics of interests for every user, but the user may write these in the about section of their profile.

The website should allow users to create profiles and log in to the site. Each user should have a profile page that displays information about them and their activities on the site. Users should be able to post text, photos, videos, and other types of content to the site, and share content from other sources.

1.3 References

<https://twitter.com/?lang=en-in>

<https://www.wikipedia.org/>

<https://stackoverflow.com/>

<https://www.mongodb.com/>

2. Overall Description

2.1 Product Perspective

Social networking sites are places where young people can maintain and nurse their existing (offline) friendships and create new (online) friendships. Social networking sites are places where young people learn the crucial importance of being able to network which they can benefit from in their future professional life. Users will interact with the website to login , create account, edit profile, post and view resources and content shared on the platform.

2.2 Product Features

User can register to the social networking site and login whenever he wishes to do so. User can create his profile add edit information set visibility to various profile sections) so his own profile and upload profile picture to his profile. With the application users can connect with their peer groups through a common platform. user can search for his friends and will be shown appropriate results if matching profiles are found. 'n+ user can post status which will be visible to his connections. Maintenance of detail of various users. Online update the information. Status of all users and community details easily available.

2.3 User Classes and Characteristics

In our system have mainly two, first is the who uses a system, second is the administrator.

- **User:** user who can use application to post and view resources.
- **Admin:** These users have an authority to delete a post or ban a profile.

2.4 Operating Environment

Our project is based on cloud and user application runs on all platforms. So, we need any device like mobile, tablet, desktop to run the application.

2.5 Design and Implementation Constraints

There are two major components for our system are client App and cloud storage

2.6 Assumptions and Dependencies

Only assumptions in the system are that user has knowledge of smartphone and internet.

3. System Features

User will create an account if there is no existing account associated with user email-id

User must be registered to login. input will be the email id and password for login

Part of building user profile. User should be able to add the educational details. These fields can remain empty

Part of building user profile. User should be able to add the personal details like language gender about user and his interests.

Registered user can share posts with his friends. The post can only be plain English, with maximum 140 characters.

4. Non-Functional Requirements

4.3.0 Performance Requirements

The system should take immediate action and post show posts on feed as fast as possible.

4.3.1 Safety Requirements

The system/application is currently in developing phase so, don't use for professional market purpose.

4.3.2 Security Requirements

Here the data of users is stores in the cloud storage so we need to maintain the security and prevent any data leak or to maintain privacy of the users

4.3.3 Software Quality Attributes

• Reliability

The reliability that user can easily use applications to connect with people without any hassle.

- **Availability**

System deployed on firebase so there is 99.9% availability.

- **Maintainability**

Our system usually not required maintenance until some modification to be done.

- **Portability**

The application is works on all platforms.

5. External Interface Requirements

5.1 User Interfaces

The user of the system has a device to run the website along with a stable internet connection.

5.2 Hardware Interfaces

No hardware interfaces needed except the device.

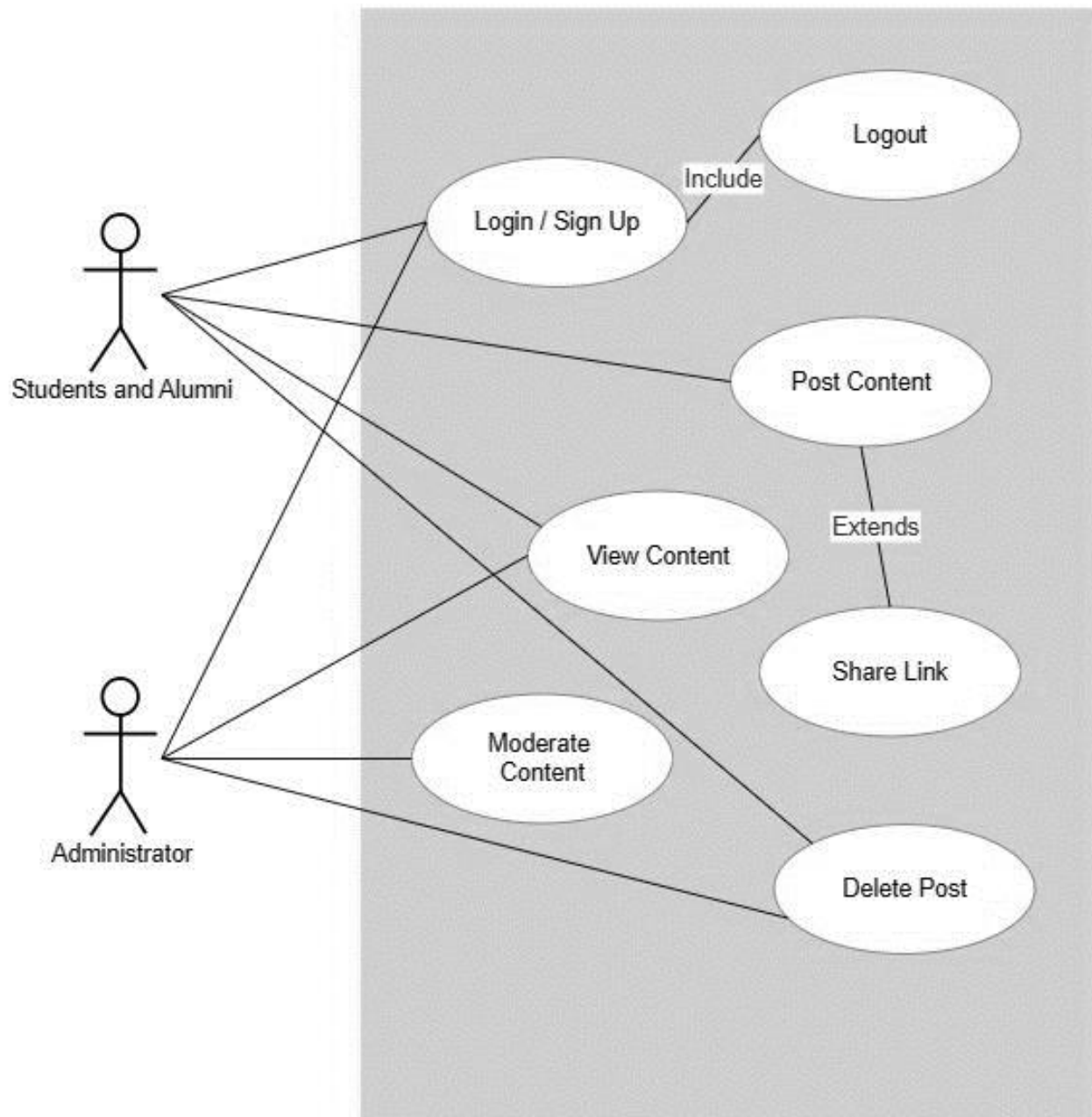
5.3 Software Interfaces

No software interfaces needed for frontend as we are using the same platform for authentication but to store the data we are using a third party platform i.e. Mongo DB Atlas.

5.4 Communication Interfaces

The system can be worked in both online and offline mode hence, communication interfaces are compulsory.

6. Detailed Use Case Diagram:

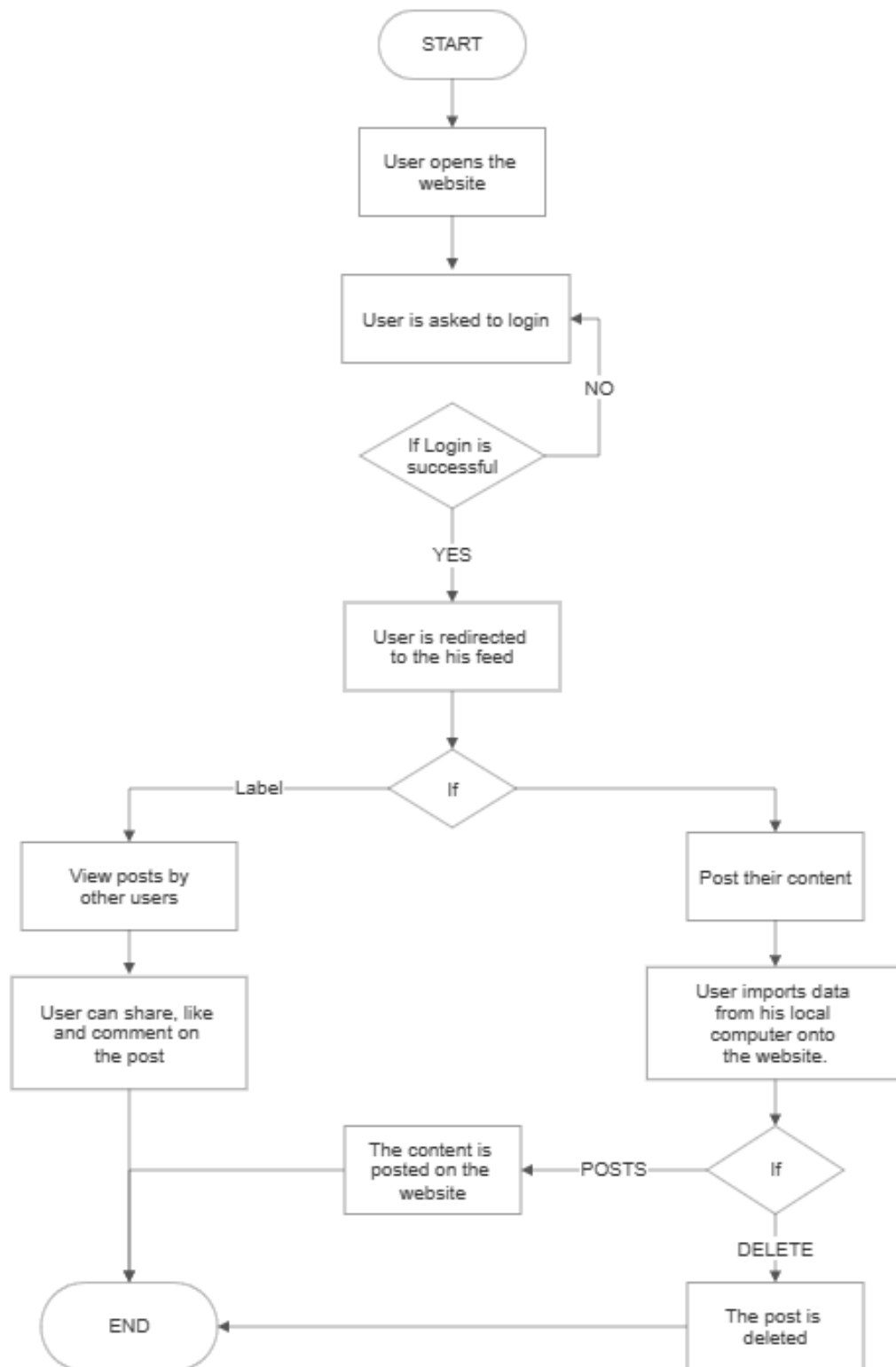


Here we have taken the use cases of login / signup , post content, View content, moderate content and delete post.

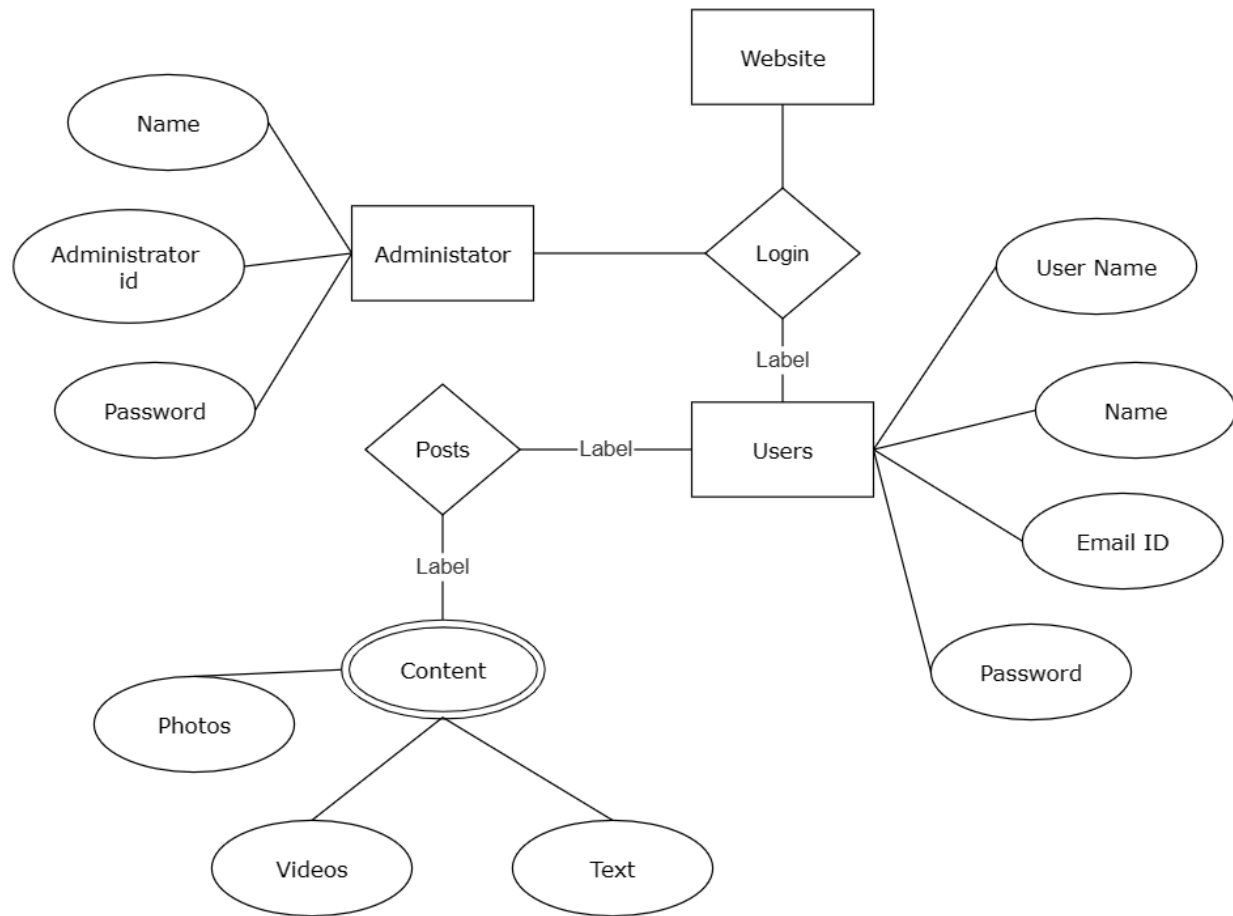
Use Case Description:

Use case ID	Scz_UC_05
Use Case Name	Post Content
Created By	Apoorv Aron
Date Created	05-01-2023
Description	This use case allows user to import / write content in the post section on the main page and user clicks on the post button to share the content on the feed.
Primary Actor	User
Secondary Actor	Administrator
Pre-conditions	User needs to login first
Post-Conditions	The content is posted on the feed

Flowchart:

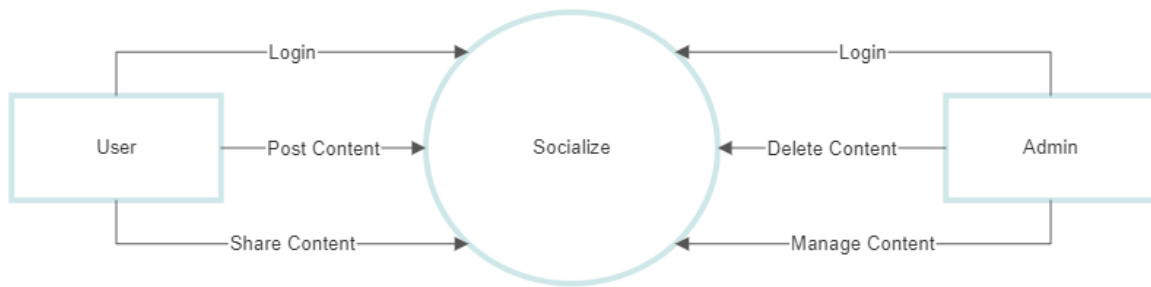


ER Diagram:

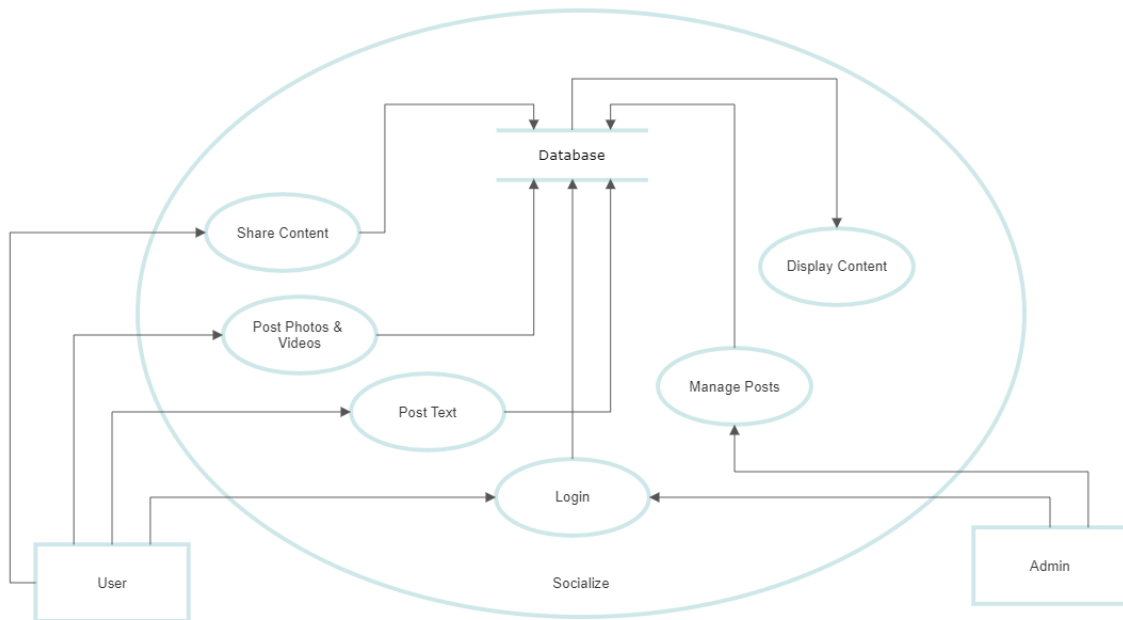


Data Flow Diagram:

Level 0 DFD:



Level 1 DFD:



Screenshots of the project:



Sign Up

Name


Email


Password

SUBMIT



Edit Profile



UPLOAD 

Name

apoorv

About

this is demo about section

Email


apoorv@gmail.com

Password

SUBMIT




Profile


 apoorv
apoorv@gmail.com

this is demo about section
Joined: Thu Jan 12 2023

POSTS FOLLOWING FOLLOWERS

 apoorv
Thu Jan 12 2023

post 3



Newsfeed



apoorv

Share your thoughts ...



POST



Aryan garg
Thu Jan 12 2023

Post 1



Who to follow



Aryan



FOLLOW

Newsfeed



apoorv

Hi this demo post



backiee-89586-landscape.jpg

POST

Who to follow



Aryan



FOLLOW

Sign Up

Name

Apoorv

Email

apoorv@ipu.ac.in

Password

! Password must be at least 6 characters.

SUBMIT

Sign Up

Name

Apoorv

Email

apoorv@ipu.ac.in

Password

❗ Password must be at most 12 characters.

SUBMIT

Profile



Aryan garg
aryan.01416401520@ipu.ac.in

UNFOLLOW

Joined: Thu Jan 12 2023

POSTS

FOLLOWING

FOLLOWERS



Aryan garg
Thu Jan 12 2023

Post 1



Snippets of Important Code:

```
1 import React, { useState, useEffect } from "react";
2 import { makeStyles } from "@material-ui/core/styles";
3 import Card from "@material-ui/core/Card";
4 import CardMedia from "@material-ui/core/CardMedia";
5 import CardContent from "@material-ui/core/CardContent";
6 import Typography from "@material-ui/core/Typography";
7 import unicornbikeling from "../assets/images/unicornbike.jpg";
8 import Grid from "@material-ui/core/Grid";
9 import auth from "../auth/auth-helper";
10 import findPeople from "../user/findPeople";
11 import Newsfeed from "../post/newsfeed";
12
13 const useStyles = makeStyles((theme) => ({
14   root: {
15     flexGrow: 1,
16     margin: 30,
17   },
18   card: {
19     maxWidth: 600,
20     margin: "auto",
21     marginTop: theme.spacing(5),
22     marginBottom: theme.spacing(5),
23   },
24   title: {
25     padding: `${theme.spacing(3)}px ${theme.spacing(2.5)}px ${theme.spacing(
26       2
27     )}px`,
28     color: theme.palette.text.secondary,
29   },
30   media: {
31     minHeight: 400,
32   },
33   credit: {
34     padding: 10,
35     textAlign: "right",
36     backgroundColor: "#d3d3d3",
37     borderBottom: "1px solid #000000",
38     "& a": {
39       color: "#344771",
40     },
41   },
42 }));
```

```
1 const create = async (user) => {
2   try {
3     let response = await fetch('/api/users/', {
4       method: 'POST',
5       headers: {
6         'Accept': 'application/json',
7         'Content-Type': 'application/json'
8       },
9       body: JSON.stringify(user)
10     });
11     return await response.json();
12   } catch(err) {
13     console.log(err);
14   }
15 }
16
17 const list = async (signal) => {
18   try {
19     let response = await fetch('/api/users/', {
20       method: 'GET',
21       signal: signal
22     });
23     return await response.json();
24   } catch(err) {
25     console.log(err);
26   }
27 }
28
29 const read = async (params, credentials, signal) => {
30   try {
31     let response = await fetch('/api/users/' + params.userId, {
32       method: 'GET',
33       headers: {
34         'Accept': 'application/json',
35         'Content-Type': 'application/json',
36         'Authorization': 'Bearer ' + credentials.token
37       },
38     });
39     return await response.json();
40   } catch(err) {
41     console.log(err);
42   }
43 }
```

```

client > JS App.js > ...
1  import React from 'react'
2  import MainRouter from './MainRouter'
3  import {BrowserRouter} from 'react-router-dom'
4  import { ThemeProvider } from '@material-ui/styles'
5  import theme from './theme'
6  import { hot } from 'react-hot-loader'
7
8  const App = () => {
9    React.useEffect(() => {
10      const jssStyles = document.querySelector('#jss-server-side')
11      if (jssStyles) {
12        jssStyles.parentNode.removeChild(jssStyles)
13      }
14    }, [])
15    return (
16      <BrowserRouter>
17        <ThemeProvider theme={theme}>
18          <MainRouter/>
19        </ThemeProvider>
20      </BrowserRouter>
21    )
22  }
23  export default hot(module)(App)
24

```

```

JS template.js M  JS user.model.js M  JS NewPost.js M  JS Home.js M  JS Profile.js X
client > user > JS Profile.js > ...
114  if (values.redirectToSignin) {
115    return <Redirect to="/signin"/>
116  }
117  return (
118    <Paper className={classes.root} elevation={4}>
119      <Typography variant="h6" className={classes.title}>
120        Profile
121      </Typography>
122      <List dense>
123        <ListItem>
124          <ListItemAvatar>
125            <Avatar src={photoUrl} className={classes.bigAvatar}/>
126          </ListItemAvatar>
127          <ListItemText primary={values.user.name} secondary={values.user.email}/> {
128            auth.isAuthenticated().user && auth.isAuthenticated().user._id == values.user._id
129          ? <ListItemSecondaryAction>
130            <Link to={"/user/edit/" + values.user._id}>
131              <IconButton aria-label="Edit" color="primary">
132                <Edit/>
133              </IconButton>
134            </Link>
135            <DeleteUser userId={values.user._id}/>
136          </ListItemSecondaryAction>
137          : <FollowProfileButton following={values.following} onClick={clickFollowButton}/>
138        </ListItem>
139      </List>
140      <Divider/>
141      <ListItem>
142        <ListItemText primary={values.user.about} secondary={"Joined: " + (
143          new Date(values.user.created).toLocaleDateString())}/>
144      </ListItem>
145    </List>
146    <ProfileTabs user={values.user} posts={posts} removePostUpdate={removePost}/>
147  </Paper>
148  )
149  }
150
151
152

```

```
JS config.js M X
config > JS config.js > ...
1  const config = {
2    env: process.env.NODE_ENV || "development",
3    port: process.env.PORT || 3000,
4    jwtSecret: process.env.JWT_SECRET || "YOUR_secret_key",
5    mongoUri:
6      process.env.MONGODB_URI ||
7      "mongodb+srv://root:cqVJgC0TCjZwG67n@socialize.i0jhyzy.mongodb.net/socialize?retryWrites=true&w=majority",
8  };
9
10 export default config;
11
```

Database Screenshot:

socialize.users

STORAGE SIZE: 1.13MB LOGICAL DATA SIZE: 561.94KB TOTAL DOCUMENTS: 4 INDEXES TOTAL SIZE: 36KB

Find Indexes Schema Anti-Patterns 0 Aggregation Search Indexes ●

INSERT DOCUMENT

FILTER { field: 'value' } ▶ OPTIONS Apply Reset

▼

```
_id: ObjectId('63bfe6da6c02f53df0c75fde')
following: Array
  0: ObjectId('63bfe6846c02f53df0c75fd8')
  1: ObjectId('63bfe6c66c02f53df0c75fdd')
followers: Array
  0: ObjectId('63bfe6846c02f53df0c75fd8')
name: "apoorv"
email: "apoorv@gmail.com"
salt: "1065540060613"
hashed_password: "fc9d1eb008a0fbc3c364cc832b566a3ebb469b14"
created: 2023-01-12T10:54:18.366+00:00
__v: 0
photo: Object
  contentType: "image/jpeg"
  data: BinData(0, '/9j/2wBDAAoHBwgHBgoICAgLCgoLDhgQDg0NDh0VFhEYIx8LJCiFiEmKzcVJik0KSEiMEEeXNDk7Pj4+JS5ESUM8SDc9Pjv,
updated: 2023-01-12T10:57:50.867+00:00
about: "this is demo about section"
```

Testing:

Boundary Value Analysis:

In the length of the password while Sign-up, length of the password expected is between 6 and 12.

Thus, the following test cases will be considered:

Test Case #	I/P	Expected O/P	Actual O/P	Similar (Y/N)
101	5	Password must be at least 6 chars	Password must be at least 6 chars	Yes, they are similar
102	6	New account successfully created.	New account successfully created.	Yes, they are similar
103	7	New account successfully created.	New account successfully created.	Yes, they are similar
104	9	New account successfully created.	New account successfully created.	Yes, they are similar
105	11	New account successfully created.	New account successfully created.	Yes, they are similar
106	12	New account successfully created.	New account successfully created.	Yes, they are similar
107	13	Password must be at most 12 chars	Password must be at most 12 chars	Yes, they are similar

Equivalence Testing:

The password length for a good length password is expected between 8 and 16.

Thus, the following equivalence classes will be considered:

- Invalid 1: Length less than 6(eg: 1, 5, 2)
- Valid: Length from 6 to 12 (eg: 8, 12, 10)
- Invalid 2: Length greater than 12 (eg: 17, 20, 50)
- Zero: No Password is entered.

Test Case #	Class	I/P	Expected O/P	Actual O/P	Similar (Y/N)
201	Invalid 1	2	Password alert should show Password must be at least 6 chars	Password alert shows Password must be at least 6 chars	Yes, they are similar
202	Valid	9	Password alert should show new account successfully created.	Password alert should show new account successfully created.	Yes, they are similar
203	Invalid 2	17	Password alert should show Password must be at most 12 chars	Password alert should show Password must be at most 12 chars	Yes, they are similar
204	Zero	0	Password alert should show Password is required	Password alert should show Password is required	Yes, they are similar