

AlgoBulls Frontend Developer

Coding Assignment

Nov 2023

As a part of this assignment, the candidate is expected to implement a small project with the below mentioned details.

Project Requirements:

1. Create a basic Social Media Web App using React JS, TypeScript and Ant Design libraries. (We need just the Frontend. Backend is not required.)
 1. Sign Up, Log In and Log Out
 2. Create a new post (only textual content)
 3. View all posts
 4. View posts created by the user
 5. Like and Bookmark any post
 6. Comment on posts
 7. Edit/Delete self-post
 8. Edit/Delete self-comment
 9. Update profile details
2. Write unit, integration and E2E tests with 100% coverage.
3. Use GitHub Actions for basic CI/CD operations.
4. Generate an Android App from existing codebase using Capacitor JS.







Detailed requirements are given below.

App Structure:

1. The application should have Sign Up and Sign In pages.
2. The application should use [Ant Design Pro Layout](#) and setup a sidebar containing 5 menu items. (Sidebar should not be visible on Sign up or Sign in page.)
 - Home - shows posts from all users and gives option to create new posts
 - My Likes - shows all liked posts
 - My Bookmarks - shows all bookmarked posts
 - My Posts - shows all posts created by the user
 - My Profile - shows the user profile
3. On the 'Home' page:
 - The user should see all the posts created by them and others
 - The user should be able to create a new post.
 - i. Use [Ant Design Pro Card](#) to create a component for the post.
 - ii. Add CSS to the component you create
 - Each post should have the following details.
 - i. Name and Photo of the post owner
 - ii. The date on which the post was created or last edited

- iii. Post content (Max length: 200 characters)
 - iv. Like and Bookmark buttons as toggles
 - v. Show count of Likes and Bookmarks next to the Like and Bookmark buttons
 - vi. If the post is created by the user itself, then show 2 additional options:
 - 1. Option to Update the post
 - 2. Option to Delete the post
 - vii. Ability to add comments. Any user can add comments to any post.
 - viii. Show the comment count next to 'Add Comment' button.
4. On the 'My Likes' page, user should be able to see the posts they have liked. Use the same [Ant Design Pro Card](#) based design for the posts on this page (created for point 3).
 5. On the 'My Bookmarks' page, user should be able to see the posts they have bookmarked. Use the same [Ant Design Pro Card](#) based design for the posts on this page (created for point 3).
 6. On the 'My Posts' page, user should be able to see all the posts created by the user. Here we use a different design - all posts should be shown in a tabular format. We use [Ant Design Pro Table](#) to show the posts. Each post would be a row in the table. The row would have an expandable option to show the comments for that post, if any. Use any design you prefer to show the comments. All features (like editing, deleting, etc) should be available in this design.

Here is a sample screenshot of table with expandable option (you need to give proper titles to the table, etc)

Application Name	state	Number of containers 	creator																
 AppName	completed	8	Qu Lili																
 AppName	undone	0	Fu Xiaoxiao																
 AppName	undone	10	and so-and-so																
<table> <tr> <th>Date</th><th>Name</th><th>Upgrade Status</th><th>Action</th></tr> <tr> <td>2014-12-24 23:12:00</td><td>This is production name</td><td>Upgraded: 56</td><td>Pause Stop</td></tr> <tr> <td>2014-12-24 23:12:00</td><td>This is production name</td><td>Upgraded: 56</td><td>Pause Stop</td></tr> <tr> <td>2014-12-24 23:12:00</td><td>This is production name</td><td>Upgraded: 56</td><td>Pause Stop</td></tr> </table>				Date	Name	Upgrade Status	Action	2014-12-24 23:12:00	This is production name	Upgraded: 56	Pause Stop	2014-12-24 23:12:00	This is production name	Upgraded: 56	Pause Stop	2014-12-24 23:12:00	This is production name	Upgraded: 56	Pause Stop
Date	Name	Upgrade Status	Action																
2014-12-24 23:12:00	This is production name	Upgraded: 56	Pause Stop																
2014-12-24 23:12:00	This is production name	Upgraded: 56	Pause Stop																
2014-12-24 23:12:00	This is production name	Upgraded: 56	Pause Stop																
 AppName	in progress	18	Chen Shuaishuai																
 AppName	in progress	18	and so-and-so																

Reference Links to implement this design (Use 'Right Click' - 'Translate to English' to translate the documentation language to English.):

[Expandable Pro Table](#)

[Editable Pro Table](#)

7. On the 'My Profile' page, user should be able to view and edit certain basic details about the user.

App Features:

1. The user should be able to sign up. If the user has already signed up, then the user should be able to log in to the app.
2. The user should be able to edit only their own post. Upon editing, the date should get updated automatically.
3. The user should be able to delete their own post(s) only.
4. Implement infinite scroll while scrolling through posts.
(You can implement this feature on your own or use this 3rd party package to do the same - [react-infinite-scroll-component](#))
 - a. The user should be able to see a maximum of 10 posts in the beginning.
 - b. When the user reaches the end of the page, the next 10 posts should be fetched and so on.
5. The users should be able to comment on their own as well as other users' posts
6. The user should be able to edit/delete their comments on the post
7. User should be able to view/edit their profile via the 'My Profile' page. The profile can have the following fields:
 - a. Email ID (read-only)
 - b. Name (editable)
 - c. Profile Photo (editable)
 - d. [Any other fields you like to show in the profile]
8. Once logged in, the user session should persist until the user manually logs out.

CI/CD:

Implement the below mentioned GitHub actions for your GitHub repo:

1. Do JS Linting using **eslint**
2. Do CSS Linting using **stylelint**
3. Run tests automatically:
 - a. Unit (jest)
 - b. Integration (jest)
 - c. E2E (cypress)

Note:

1. Ensure that all the actions are triggered on every commit push.
2. Ensure that all the GitHub actions are passing before you submit your assignment for review.

Technology Stack to be used for creating this application:

Frontend:

1. [React v17+](#)
2. [TypeScript](#)
3. [Ant Design v5+](#)
4. [Ant Design Pro v5+](#)
5. [Ant Pro Table](#)
6. [Capacitor.js](#)

Database:

1. If you like to use an external database, you can use any database/service like Firebase, etc but ensure you are interacting with the database using REST APIs only (GET, POST, PUT, PATCH, DELETE) and no other way.
2. In case you don't like to use an external database, you can mimic backend functionality without implementing a backend by using REST APIs with mock data. You can follow this link to know more <https://v3.umijs.org/docs/mock>.

We recommend method #2 - use mock data

Note:

1. You can assume things that are not defined with a reasonable value. For example, we have not defined the pagination size for the table in 'My Posts' page. You can assume it as 15 or 20 or any other number which is generally acceptable.
2. Implement necessary design principles wherever possible. *For example, when a user tries to DELETE an existing post or comment, it is good to ask first for a confirmation from the user before firing the DELETE api. The design should look complete.*
3. Explore Ant Design sufficiently so you can use the available features from the existing widgets smartly. This will be preferred over creating completely new components. Feel free to take design inspiration from <https://app.algobulls.com>.
4. Share details on the REST APIs that you have created. Share the documentation using Google Doc or Postman or Swagger or any other similar tool.

Deliverables:

1. Your Social Media Web App (Frontend): Please come up with a sleek front end that can accommodate all the above mentioned requirements.
2. Implement Testing
 - a. Unit ([jest](#))
 - b. Integration ([jest](#))
 - c. E2E ([cypress](#))

3. Coverage using jest should be 100% (Exclude files/code which cannot be covered)
4. Coverage using cypress should be 100% (Exclude files/code which cannot be covered)
5. You should create an Android App from the existing codebase using [CapacitorJS](#).
6. Implement the mentioned GitHub actions and ensure they are all passing before submitting the assignment for review.

Duration:

The ideal time for completion is 5 days. Upto 7 days is acceptable. If you need extra time, please request the same with appropriate reasoning.

How to submit:

1. Deploy your application on any cloud service provider like Vercel, AWS, GCP, Azure, etc. and send us a link to your working application.
2. Please upload your code on a private GitHub repo and share it with the following GitHub user-ids - **algobulls-dev**, **adinath-uparkar-algobulls**

Ensure your GitHub repo is Private.

- a. Make sure your code is cleaned up as per standards, before you do the final submission. It is ok to keep pushing any number of intermediate code commits.
 - b. Add sufficient comments for complex logic, before you do the final submission.
 - c. Include a README in the repo that includes basic documentation on how to run the code.
 - d. Include complete coverage reports created using jest and cypress in the repo. Add screenshot of the coverage summary in the README section.
 - e. Check in your android app in the repo.
3. Once done, please ping on internshala chatbox with the link and other details. Also, please send a mail to both developers@algobulls.com & adinath.uparkar@algobulls.com, mentioning your name and GitHub ID, requesting a review. Attach this coding assignment pdf to the mail as well.

Asking for clarification/hints:

You can ask on the internshala chatbox for any queries. The response may be delayed as we may have a lot of applications and it may take time to respond to all of them.