

PRODUCT FRONTEND TECHNICAL ASSESSMENT

Things to keep in mind

- You're expected to submit your answer as soon as possible and **must still be within 48 hours** to complete all challenges; This time will be count from the time this email was be opened.
- You can use Google while taking the test, however, make sure you are the person who actually took the test;
- In case of any change in your progression and have to extend the deadline, it's must be updated as soon as possible via email and be accepted by our HR Team.

Test result note

- All of the source code, screenshots of running results, relevant information must be attached in your result
- Please upload all of them into 1 folder and submit via form <https://forms.gle/mQucFQ3XMJLgtJTZ8>
- Screenshots of running results must be attached in your result.

Suggestions

- You can use libraries/frameworks like React, Angular, Vue, Material UI, Ant Design, etc.
The template in the assignment uses Refine (React Query + Ant Design)
- A text file describing your approach (which libraries/tools were used and why)
- Source code (excluding node_modules , vendor , .git , built, and minified files)
compressed in a zip file
- A README.md file describing how to set up & start the project

CHALLENGE

Implement a web app similar to the template found here: <https://gi-pf.vercel.app/>

Specific requirements:

- Integrate with the API <https://jsonplaceholder.typicode.com/> to fetch data
- Use the API <https://ui-avatars.com/> to generate avatars for users
- Include 2 modules:
 - Album:
 - Album list: displayed as a table
 - ◇ Contains all albums (100 albums)
 - ◇ Includes columns for ID, Title, User (displaying name & avatar, based on userId), Actions (button to view album detail)
 - ◇ The table has pagination, and the pagination state is synced with the URL so that if the user reloads the page, the pagination state is preserved
 - ◇ Allows clicking on User information to navigate to the User detail screen
 - Album detail: displays the following information
 - ◇ User information: avatar, name, email. Allows clicking on the name to navigate to the User detail screen
 - ◇ Album title
 - ◇ List of photos in the album: use thumbnailUrl to view the list, users can click to use url to view the photo
 - User:
 - User list: displayed as a table
 - ◇ Contains all users (10 users)
 - ◇ Includes columns for ID, Avatar, Name, Email, Phone, Website, Actions (button to view user detail)
 - User detail: displays the following information
 - ◇ User information: avatar, name, email
 - ◇ List of user's albums (10 albums), with a button to view album detail

- Layout displays well on screens 1280px wide and up
- UI should be tidy with reasonable colors
- Display loading state when calling the API
- Clickable elements should display a pointer cursor
- Emails and phone numbers should be clickable to perform the corresponding actions
- Clicking on an external link (website URL) should open a new tab
- All img tags must have an alt attribute
- Source code should be organized with a clear structure, use sensible variable names, and be free of spelling errors

---Thank you---