

**Assignment\_ID:** assignment\_category\_0001

Video LINK:

<https://drive.google.com/file/d/1cfdnAl4VtXwcYkl1fJfEaXdoKm6woHeV/view?usp=sharing>

### **Online Group-Study assignment**

In this assignment, you will build a web application for online group study with friends(Every registered user is a friend of others ).Users can create assignments, complete them, and grade their friends' assignments.

#### **Main requirements:**

##### **1. Authentication:**

- Users are able to register an account by providing name, photoURL, email, and password using firebase .
- Users are able to login using social login. Implement at least one social login(like google login, github sign up) .

##### **2. Assignment Management: [Here all the necessary routes will be private route ]**

###### **a. Assignment creation:**

- Any logged in user is able to create an assignment for all users.
- An assignment will have a title, description, marks, thumbnail Image URL, assignment difficulty level(easy, medium, hard) **[YOU MAY USE DROPDOWN SELECT INPUT FIELD]**, and due date [use this package for selecting date **"https://www.npmjs.com/package/react-datepicker"**] .
- A **success message** will be shown when the assignment will be created successfully. **[YOU MAY USE TOAST OF MODAL]**

###### **b. Assignment deleting:**

- Only the user is able to **delete an assignment who has created the assignment**. A user is not able to delete assignments which are created by other users .

Hint: While a user will create an assignment you have to store the user email with the assignment data [followed by **Assignment creation requirement**] . And then while a user will be trying to delete an assignment you will

compare the current user email with the assignment creator email. And if it matches then the assignment will be deleted.

- A successful message will be shown when the assignment will be deleted successfully .
- An error message will be shown when a user tries to delete an assignment which is not created by himself ...

**c. Assignment Updating:**

- Any user is able to update any assignment .  
**[optional : it's recommended to let the user update an assignment task who has created it. Follow hints of delete operation]**
- While a user will try to update an assignment the input fields of updated assignment form will be filled automatically with previous assignment data **[automatically filled is optional . But you should try it ]**. And users can change the input field data and can update with clicking on update assignment button.
- While the assignment is updated, show a success message and redirect the user to the all assignment page. **[Redirecting is optional]**

**3. Taking Assignment as an individual:**

- All users are able to visit the assignment page (without logged in) and are able to filter assignments based on assignment difficulty level .  
**[Hint: While you are posting an assignment there is a difficulty level field . So make a similar input field here . While users will select any level from (easy, medium, hard) just find data from the database based on the difficulty level]**
- Users will see a thumbnail, title, marks, assignment difficulty level and **“View Assignment”** and **“Update Assignment”** buttons for every assignment . It's up to you how you will display the cards of assignment .
- By clicking on the view assignment button the user will be navigated to the assignment details page (this page will be private) where the user will see the assignment details which are provided while the assignment was created. And the **“Take assignment”** button will be shown.
- Users are able to submit an assignment by clicking on the **“Take assignment” button** and it will open a modal [Or you may open a new page ,it's up to you] with the assignment submission form .

- The assignment submission form will have input fields for PDF link submission and another text area for giving a quick note text.
- By default every submitted assignment will be in **pending status** . And save the **user email** with the submitted assignment so that it can be determined who has submitted it.

#### 4. Marking assignment:

- The submitted assignment page will contain **all submitted assignments** by the users .
- Only pending assignments will be shown on the submitted assignments page.
- Every submitted assignment will have the assignment title, assignment marks, examinee name, and a “**give mark**” button.
- By clicking on the give mark button it will open **a modal or will navigate to a new page**. And a user will be able to see the google drive link(pdf file) , note submitted by examinee. There will be a marks input field and a feedback input field and a submit button for giving marks.
- User is able to give a mark after clicking on the submit button .
- After marking an assignment the status of the assignment will be changed to **completed** .

**Pages:** \*\*\*All the pages mentioned below are must be created\*\*\* (If you find any page are missing. Then you can add.)

1. Home page(**public page**) - will have a navbar, footer, a banner section, a feature section, and a faq section.
  - a. Navbar - will have a logo, assignments, login, register link before logged in and will have logo, create assignments, assignments, my assignments, submitted assignments links, log out button with user image(while hover on the user image it will show the user name)
  - b. Banner section - Design it based on the project theme
  - c. Feature section - will have few cards of feature
  - d. Faq - will have some frequently asked questions
  - e. Footer - will have copyright, and relevant links
2. Create assignment page(**private page**) - will have functionality mentioned above .
3. All Assignments page(**public page**) - will have all the assignments created by any user and it will have functionality mentioned above.
4. My assignment page(**private page**) - will have all assignments which are submitted by the specific user. For example, if you logged in you will only see your submitted assignment on the my assignment page

- a. You can see assignment title, assignment status, assignment marks, your obtain marks, feedback(if you got the marks)
5. Submitted assignment page(**private page**) - will have all the pending assignments submitted by any user and have the functionality mentioned above.
6. Application will have a few more dynamic pages(**private pages**) like **update assignment page, viewing a single assignment page**.
7. Application will have login, and registration page(both are **public pages**)

### **Bonus Requirements:**

1. Commits & readme:
  - Minimum 18 meaningful git commits on the client-side repository
  - Minimum 8 meaningful commits on the server-side repository
  - Create a readme for client-side and write about your project (at least 5 features with bullet points). **\*\*Remember to add your client-side live link of your website here\*\***
2. Previewing docs or pdf:
  - In the submitted assignment page show preview for every submitted assignment resource preview(which pdf link was submitted while a user submitted an assignment) in the modal.

Hints: Search in google "<https://www.npmjs.com/package/react-pdf>". You will find some resources. Study and implement it.

3. Pagination: Implement pagination on the all assignments page
4. Validation:
  - Add validation in the create assignment form.
  - Add validation in the registration form.
5. Make the website fully responsive for mobile, tablet & desktop view.
6. Upon login, you will create a jwt token and store it on the client-side and you will send the token with the call and verify the user. Implementing 401 and 403 is optional. Ensure you have implemented jwt token and create a token and store it on the client-side for both email/password-based authentication and social login. You must implement JWT on your private routes.

### **Optional (But Highly Recommended):**

Implement any two tasks from the following optional list:

1. Add a spinner when the data is in a loading state. You can add a gif/jpg, use any package or customize it using CSS.
2. Explore and implement any of the animations from the Framer Motion
3. Explore and implement Tanstack query mutations in your api.
4. Add one extra feature of your own. This will help you in the future to differentiate your project from others.

**Additional information:**

1. You can host images anywhere.
2. You can use vanilla CSS or any library.
3. Try to host your site on Firebase (Netlify hosting will need some extra configurations)
4. Host your server-side application on Vercel. If needed, you can host somewhere else as well.
5. Make Sure you deploy server-side and client-side on the first day. If you have any issues with hosting or GitHub push, please join the "Github and deploy" related support session.

**What to submit:**

1. Your client-side code GitHub repository
2. Your server-side code GitHub repository
3. Your live website link

**Some Guidelines:**

1. Do not waste much time on the website idea. Just spend 15-20 minutes deciding, find a sample website, and start working on it.
2. Do not waste much time finding the right image. You can always start with a simple idea. Make the website and then add different images.
3. Don't look at the overall task list. Just take one task at a time and do it. Once it's done, pick the next task. If you get stuck on a particular task, move on to the next task.
4. Stay calm, think before coding, and work sequentially. You will make it.
5. Be strategic about the electricity issue.
6. use chat gpt to generate JSON data. You can use chatGPT for Other purposes as well.