Assignment_ID: assignment_category_0001

Welcome to Job assessment by BJET Inc.

We're currently in search of a talented junior MERN stack developer to join our team in Japan. As a junior member, you'll have the opportunity to work closely with our experienced developers on exciting projects utilizing MongoDB, Express.js, React, and Node.js.

This role offers an excellent chance for growth, where you'll be able to enhance your skills while contributing to the development of innovative web applications. We're seeking someone with a solid foundation in web development and a passion for learning new technologies. If you're enthusiastic about building dynamic and user-friendly applications and eager to kick-start your career in the MERN stack environment, we'd love to hear from you. We Invited you to complete this job assessment to prove your skill

Read This Following details carefully

Focus on making the website visually appealing. Ensure that

- Color contrast is pleasing to the eye
- The website does not have a gobindo design
- The website has proper alignment and space

If needed, customize the design of any component you are taking from any component library. (For example, you are using daisy ui & have taken a card component from Daisy, if needed, customize the styling of the card to make it reasonable rather than just copy & paste it.)

Your website can not be related to your previous assignments' layout/ design or any practice project shown in the course modules or our conceptual sessions.

Ex: You can't copy any design or similar functionality/ layout of

- Dragon news
- Espresso Emporium
- Genius Car
- Career Hub Website,
- Any conceptual session projects
- Projects shown in our course.
- Any of your previous assignments

If any similarities are found, you will get zero(0) as a penalty

Website Theme⭐ 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.

Key Rules:

- Include a minimum of 18 notable GitHub commits on the client side
- Include a minimum of 8 notable GitHub commits on the server side
- Add a meaningful readme.md file with the name of your website and a live site URL. Include a minimum of five bullet points to feature your website. Write your selected category's name here.
- Make it responsive for all devices. You need to make it responsive for mobile, tablet and desktop views.
- After reloading the page of a private route, the user should not be redirected to the login page.
- Use the Environment variable to hide the Firebase config keys and Mongodb credentials.
- Don’t use any Lorem ipsum text; you can not use the default alert to show any error or success message.

Main requirements

All the pages mentioned below are must have to be created

1. **Home page(public)** -

   will have a navbar, footer, a banner section, a feature section, and a FAQ section.

   - **Navbar** - will have a logo, assignments, login, register link before logged in and will have a logo, assignments, create assignments, pending assignments and user avatar after logged in. While click on the user avatar it will open a dropdown and the dropdown will have my attempted assignments, logout button
   - **Banner section** - Design it based on the project theme
   - **Feature section** - will have few cards of feature
   - **Faq** - will have some frequently asked questions
   - **Footer** - will have copyright and relevant links

2. **Create assignment page**(**private page**)

   will have a create assignment page to create assignments for all users.

   Assignment creation Functionalities:

   - 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 MUST 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.

3. **Assignments page(public page)**

   will have all the assignments created by any user using "create assignment" form and Will have filter assignments functionality based on assignment difficulty level.

[Hint: While you are posting an assignment there is a difficulty level field. So make a similar dropdown/filter button here. While users will select any level (easy, medium, hard) just find assignments from the database based on the difficulty level]

Every individual assignment will have a thumbnail, title, marks, assignment difficulty level and a delete, update, and view assignment button. It's up to you how you will display the cards of the assignment

- ○ **Delete button functionality**
  - 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. Show a confirmation modal to confirm the delete action while pressing the delete button.
    *Hint: While a user will create an assignment you have to store the user email with the assignment data [followed **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.
- ○ **Update button functionality**
  - Any user can update any assignment.

  - While a user will try to update an assignment the input fields of the updated assignment form will be filled automatically with currently stored data of the assignment**.** Users can change the input field data and can update by clicking on the update assignment button.

  - While the assignment will be updated, show a success message and redirect the user to the assignments page. **[Redirecting is optional]**
- ○ **View assignment button functionality**

- By clicking on the view assignment button the user will be navigated to the assignment details page (this page will be a private dynamic page) 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/doc link submission and another text area for giving a quick note text.

- By default, every submitted assignment will be in **pending status**. Save the **user email** with the submitted assignment so that it can be determined who has submitted it.

4. **My Submitted/attempted 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.
   - You can see the assignment title, assignment status, assignment marks, your obtained marks, and feedback(if you got the marks)

5. **Pending assignments page**(**private page**)
   will have all the pending assignments submitted by any users
   - The pending assignments page will contain **all the pending assignments(which assignments are not marked yet) submitted by** the users. Exclude the assignments which are marked already.

- Every submitted assignment will have the assignment title, assignment marks, examinee name(who has submitted the assignment), and a “**give mark”** button.

- By clicking on the give mark button it will open **a modal or will navigate to a new page**. A user will be able to see the pdf/docs link, and notes which the examinee has submitted. There will be a marks input field, a feedback input field and a submit button for giving marks.

- The user can give mark by clicking on the submit button after filling feedback, and mark input fields.

- After marking an assignment the status of the assignment will be changed to **completed**.

6. Application will have a login and registration page(both are **public pages**). please read the Authentication section in the functionalities

Challenge Requirements:

1. **Validation** :
   - Add validation in the create assignment form.
   - Add validation in all authentication-related forms.

2. **JWT**:
   Upon login, you will create a JWT token and store it on the client side. 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.

3. **Show a preview:** Show a Preview of every pdf with iframe on the Submitted assignment page. You can use any packages for this. You can also use <iframe/>

4. **Theme Customization**
   Add a Theme Toggling Button for changing themes from light to dark / dark to light. By changing the theme, it will make the full system dark / light based on user interaction.

What to submit:

1. Your Assignment ID/variant
2. Your client-side code GitHub repository
3. Your server-side code GitHub repository
4. Your live website link

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.
5. Pagination: Implement pagination on the all assignments page.

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.

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.