

# Mi-12 Assignment Requirement

assignment12\_category\_0008

🚩: 0 [ If we have any update we will mention it here ]. Do check frequently to see if any updates have been made.

## Objective

As we embark on this development journey, our goal is to create a robust and user-friendly platform using the MERN stack (MongoDB, Express.js, React, and Node.js). EduManage will revolutionize the way educational institutions, tutors, and students interact, making skill learning and class management more efficient and accessible than ever before.

## Key Rules

- Include a minimum of 20 notable GitHub commits on the client side
- Include a minimum of 12 notable GitHub commits on the server side
- Add a meaningful readme.md file with the name of your website, Admin username, password and live site URL. Include a minimum of 10 bullet points to feature your website.
- Make it responsive for all devices. You need to make it responsive for mobile, tablet and desktop views. Make the dashboard responsive as well.
- 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 in your website.
- Show sweet alert/toast/notification for all the crud operations, successful authentication login, and sign-up. Don't use the default browser alert.

- Implement tanstack query in all the data fetching functionality (For GET method only)

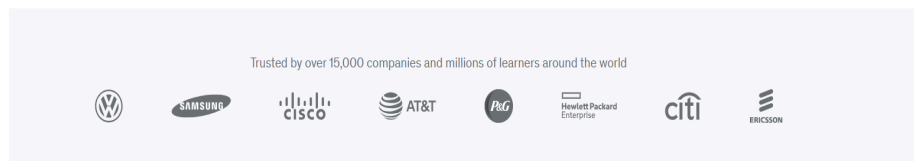
## Main Tasks

1. **Navbar:** Navbar has a **logo+website name**, **Home**, **All Classes**, **Teach on [your website name]**, and **Sign In** (when not logged in) button. If the user is logged in, his/her profile picture should appear on the navbar.

If the user clicks on the **profile picture**, a drop-down will appear with the following items: **User name** (not clickable), **Dashboard**, and **Logout** button.

2. **Homepage:**

- The banner section will have a banner images /carousel (images should be relevant to the website)
- Create a section highlighting partners or collaborators. Include their logos and a brief description of the partnership.



- Create a section to Highlight a few classes or courses that are currently popular or recommended. Based on highest enrollment or highest review. (use slider highly recommended)
- Creating a feedback section in the student dashboard is an excellent way to build trust and credibility. To enhance the user experience and provide a clear understanding of the feedback, you can utilize a carousel to display each feedback [see 8](#).
  - Feedback text
  - Name (who post the feedback)

- Image (who post the feedback)
- title(class title)
- Create a section to show the total users use this website, total classes, Total student enrollment in your website. On the left side show this information using a card and on the right side show an image relevant to your website.

**Note:**

- **Total User Card:** Displaying the overall user count in your system.
- **Total classes:** Showing the total number of classes added from the teacher dashboard.
- **Total enrollment:** Displaying the total enrollment count in your system.
- Create a section for teachers to join your website as a teacher. In this section the left side will have an image and the right side will have some text . Under text create a button start teaching today. When clicking the start teaching button the user redirects to teach on [your website name] page. For example



## **Become an instructor**

Instructors from around the world teach millions of learners on Udemy. We provide the tools and skills to teach what you love.

**Start teaching today**

- Added at least two extra sections on the home page relevant to the website.

### 3. All classes:

All classes' pages display classes added from the teacher dashboard if the admin approves the class. Each class is presented in a card format. On this card, you will have to show the following information.

- Title
- Name (who posted the class)
- Image
- price
- Short description
- Total enrolment
- Enroll

Note: For the enroll button task following number 4.

### 4. Class Details Page (Private Route):

After clicking the Enroll button he/she is redirected to the class details page(class/:id) containing information (all info about the class example teacher, price, etc, and the **Pay button**).

Note: After clicking the pay button he/she is redirected to the payment page and payment for class. After successfully payment redirects to /dashboard/my-enroll classes page. Where he/she sees his/her enrollment class. And data will be saved to another collection(payment and class information)

### 5. Teach on [your website name] (private route):

This page allows users to apply for a teaching position by providing the required form.

- Name
- Images(who logged in)
- Email (read only)
- Create an experience option field that includes 3 options (beginner, experienced, and mid-level)
- Title
- category (web development, digital marketing, etc added at least 5 categories) using select.

- Submit for review button

**Note:** Once they click the "submit for review" button, the data is saved in the database, and a teacher request page is shown on the admin dashboard. If the admin rejects the request, change the submit for review to request to another button. If the admin approves the teacher request the user will be a teacher. When the user's role is teacher this page(teacher request page) will show relevant messages and disappear the form.

## Student Dashboard (Private Route):

**Note:** This must be a dashboard layout

6. When a Student clicks on the Dashboard, he/she will be redirected to a page where there will be the following routes:

- A. My enroll class
- B. Profile

### 7. My enroll class:

My enroll class page displays classes he/she enrolled in from the details page. Each class is presented in a card format. on this card, you will have to show the following information

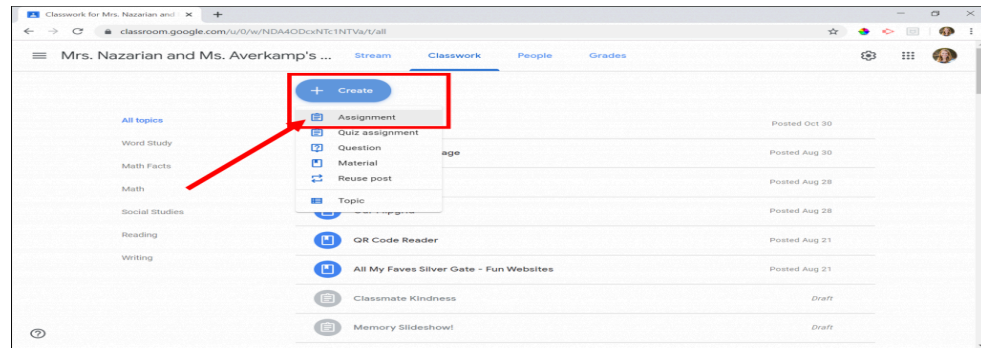
- Title
- Name (who posted the class)
- Image
- Continue button

8. My enroll class details: After clicking the Continue button he/she redirects to the My enroll class details.page(dashboard/myenroll-class/:id). On this page students see assignments added from the teacher dashboard. for these classes in tabular form with required data shown this data under the **TER(button)**

- Title
- Description
- Deadline
- Submit (beside every available assignment of that class)

**Note:** when you click the **submit button** assignment will be submitted and increment the assignment submission of the teacher class detail page per day assignment card by 1. Suppose per day assignment is 1 when you submit the assignment card will be 2.

- On this page Create a **Teaching Evaluation Report (TER)** button. when you click the button show a modal with the required input field. Make the button under the navbar.



- Description
- ratings use([react raring components](#))
- Send button

**Note:** When users click the **send button**, the data will be saved in another collection in the database. This feedback will be displayed in the feedback section on the homepage. Additionally, the admin can view The details (dashboard/class/:id) page.

## 9. My Profile:

See point no 14

## Admin Dashboard (Private Route):

**Note:** This must be a dashboard layout

**10.** When an admin clicks on the Dashboard, he/she will be redirected to a page where there will be the following routes:

- A. Teacher Request
- B. Users
- C. All classes
- D. Profile

**11. Teacher Request:**

On this page, the admin sees all teacher requests added from the Teach on [your website name] page. Each class shown in tabular form will have this information.

- Name
- Images(who apply for teacher)
- experience
- Title
- category
- status(pending)
- Approved button
- Reject button

**Note:**

- When the admin **approves** the teacher request this status is changed to **accepted** and the student can access the teacher dashboard(that means the student role is updated to teacher). If the admin **rejects** the class status is changed to **rejected** in the teacher request page and student dashboard my request page.
- After changing the status, the **approved button** and **reject button** will disabled.

**12. Users:**

Show all the users in a tabular form where each row will have:

- User name
- User email
- Make admin
- User image

The admin can make a user admin by clicking on the **Make Admin** button. When the user is admin, disable the admin button. Implement a server-side search functionality to find a specific user (via username/email).

### **13. All classes:**

Show all the classes in a tabular form where each row will have:

- Title
- Image
- Email (who post the class)
- Short description
- Approved button
- Reject Button
- See progress

#### **Note:**

- This page presents information from the teacher dashboard(add class page)
- When the admin **approved** the class this class showed in the navbar all classes page.
- If, unfortunately, the class doesn't meet the criteria and is rejected by the admin, its status changes to 'Rejected,' and it won't be listed on the 'Add Classes' page.
- On the Teacher dashboard, my class page data status is initially **pending**. When the admin approves the class this status is changed to **accepted**. If the admin rejects the class the status is changed to **rejected**
- The **see progress** button will initially disabled after the admin approves the classes see progress button will be enabled. When clicking the see progress button redirects to another page (dashboard/class/:id).In the details pages, the admin can view feedback for these classes.

### **14. My Profile:**



Show users information with the required data

- user's name
- role
- Image
- email
- phone

## Teacher Dashboard (Private Route):

**Note:** This must be a dashboard layout

**15.** When a Teacher clicks on the Dashboard, he/she will be redirected to a page where there will be the following routes:

- Add class
- My class
- Profile

### **16. Add class:**

Create a class page that enables teachers to add classes through a form.

The form should include the following fields

- Title
- name(not editable)
- email(not editable)
- Price
- Description
- Image
- Add class button

**Note:** When the 'Add class' button is clicked, data will be stored in a MongoDB database. after that user will be redirected to my classes Page and also added to all class pages (admin dashboard)

### **17. My class:**

On this page, the teacher sees his/her classes added from the add class page. Each class shown on the card will have this information.

- Title
- name(not editable)
- email(not editable)
- Price
- Description
- Image
- status(initially pending)
- Update button
- Delete button
- See details

Note:

- you can show the update form in a modal or another route. When the **'update button'** is clicked, data will be updated
- Delete action- After clicking the delete button, the class will be removed. Before the delete, ask for a delete confirmation.
- **See details** button will initially disabled after the admin approves the classes See Details button will be enabled. When clicking **see details button** redirects to another page (dashboard/my-class/:id).

**See details:**

this page will have two sections Class progress section and the add assignment section

- **Class progress section:** in this section teacher sees three card
  - Total enrollment card. In this card, the teacher sees the total enrollment in this class  
**Hints:** when students enroll the classes count the enrolment And update the count for specific classes.
  - Total assignment card In this card teacher sees he/her added total assignment
  - Per day assignment submitted.

**Hints:** when a user submits the assignment count the number updates the count for a specific assignment and sort the data based on the current date.

- **Class assignment:** under the class progress section show a button name create . For example



When teacher click the **create button** show a modal with required input field

- Assignment Title
- Assignment Deadline
- Assignment Description
- Add assignment

Note: when clicking the **create button** data will be saved on database and Show this assignment user enrolled specific classes [see 8](#) and in this page Total assignment card increment by 1. Suppose total assignment card is 1 When you create the assignment, the total assignment card will be 2.

### **18. My Profile:**

See point no 14

**19.Login & Registration systems:** On the Registration and Login pages, display relevant error messages when necessary

**Login Page:** When a user clicks on the login button, they will be redirected to the login page having the following:

- Email/Password
- Google Sign-in

- A link that will redirect to the registration page

Registration Page: The Registration page will have the Email/Password form having the following fields:

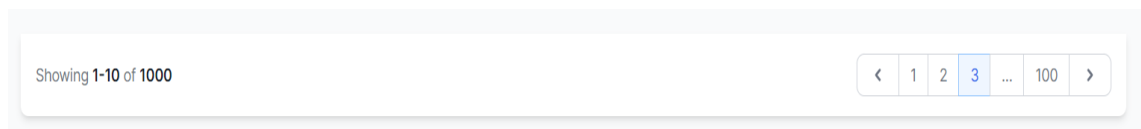
- Name
- Email
- Password
- PhotoURL

Note: Do not enforce the email verification method and forget & reset password method, as it will inconvenience the examiner. If you want, you can add it after receiving the assignment result

**\*\* when the user create default role is student**

## Challenges Part

1. Add tanstack mutation for every post request.
2. Implement [react-hook-form](#) in all form pages.
3. Implement JWT on login (Email/Password and social) and store the token in local storage.
4. Implement pagination at the footer of all the tables and cards you have implemented (show 10 users at a time). For example,



## Optional Tasks

- 1) On the student dashboard create a my order page where students can see him/her order in tabular form. This tabular form shows order information like class title, price, transaction id, email(who posted the email), teacher email(who posted the job), and invoice button.

Note: when clicking the invoice button download the invoice as a PDF using [jsPdf](#)

- 2) Added animation on your website using ([framer motion](#), or [aos](#))
- 3) Implement Axios interceptor.
- 4) Added a search bar in the navbar on all class page.
- 5) Use the infinite scroll on all class pages (we recommend you do this) [video link](#)

## What to Submit

1. Assignment Category/variant:
2. Submit admin email
3. Submit Admin password
4. Front-end Live Site Link:
5. Client Side Github Repository Link:
6. Server Side Github Repository Link: