Assignment12_category_017

📽️ Requirement Explaintion Video 📽️

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

Objective

You must develop a **Medical Camp Management System (MCMS)** using the MERN stack. This system helps the **Organizer,** and **Participants** easily manage and coordinate medical camps.

Key Rules

- Include at least 20 notable GitHub commits on the client side.
- Include at least 12 notable GitHub commits on the server side.
- Add a meaningful README.md with the website name, **Organizers** username, password, live site URL, and at least 10 bullet points featuring the website.
- Make the site responsive for mobile, tablet, and desktop, including the dashboard.
- Ensure users are not redirected to the login page after reloading a private route.
- Use environment variables to hide Firebase config keys and MongoDB credentials.
- Avoid using Lorem ipsum text on the website.

- Show sweet alerts/toasts/notifications for all CRUD operations and successful authentication, avoiding default browser alerts.
- Implement TanStack Query for all data fetching (GET method only).
- Implement 404 Page.
- Add a Footer to provide essential links, contact information, or quick access to other parts of the web app.

Main Tasks

HomePage:

1. Navbar has a **logo+website name, Home, Available Camps, Join US (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. The banner section will feature a slider showcasing success stories and impactful moments from past medical camps, highlighting achievements and positive outcomes.

3. **Create a section showcasing popular medical camps(highest participant count) with details for a maximum of six camps, including Camp Name, Image, Camp Fees, Date and Time, Location, Healthcare Professional, and** participant count.
   Note:
   A. Participant Counts begin at zero and increase with each user joining a camp (refer to camp details).
   B. After implementing the above features, include a "See All Camps" button to direct users to an "Available Camps" page.

4. For each popular camp, add a button linking to detailed information following the existing route (/camp-details/:campId). The details will have the following:
    a. Camp Name
    b. Image
    c. Camp Fees
    d. Date and Time
    e. Location
    f. Healthcare Professional
    g. participant count
    h. Description
    i. Join Camp Button

5. In the details page, Implement a modal for participant registration. Click "the Join Camp Button" to open. Collect Info
    A. Camp Name (read-only)
    B. Camp Fees(read-only)
    C. Location(read-only)
    D. Healthcare Professional Name(read-only)
    E. Participant Name (from loggedIn info)
    F. Participant Email(from loggedIn info)
    G. Age
    H. Phone Number
    I. Gender
    J. Emergency Contact

    After getting the info Save data to a dedicated database collection.

    Note : The participant count will increase by one every time a user joins a camp.

6. Improve system clarity by adding two new sections.
    A. Introduce "**Feedback and Ratings**" to display participant feedback and ratings, derived from the participant dashboard after payment. This provides detailed insights into their camp experiences.
       (see **Registered Camps route** **Feedback and Ratings section** **for details**)

B. Develop a dedicated section for this project as you choose. Ensure it's related to this project.

Available Camp Page:

7. In the route, display all the camp data added by organizers from add a camp route (see **Add A Camp** for details). Provide essential details for each camp card:

   A. Camp Name,
   B. Image,
   C. Date & Time,
   D. Location,
   E. Healthcare Professional Name
   F. Participant Count
   G.Description.

8. Include a prominent "Details" button next to each camp listing. This button links to a details page where participants can express interest and register using a "Join Camp" button.

   Note: Link each camp on the "Available Camps" page to its Details Page.

Additional Feature:

A. Introduce a search bar allowing users to search for specific camps based on keywords, dates, or other relevant criteria.
B. Allow users to sort available camps by various criteria, including Most Registered, Camp Fees, and Alphabetical Order (Camp Name).
C. Design a layout button. Initially, cards will be displayed in three columns. After clicking the button, they will switch to a two-column layout.

Organizer Dashboard (Private Route):

Note: This must be a dashboard layout

9. When an admin clicks on the Dashboard, he/she will be redirected to a page where there will be the following routes: (You need to set an organizer manually)
    A. Organizer Profile.
    B. Add A Camp
    C. Manage Camps
    D. Manage Registered Camps

10. **Organizer Profile**
    Allow organizers to manage their profile information, including name, image, and contact details. They can click the "Update" button to open a form with their current information for editing.

11. **Add A Camp**
    On this page, the organizer will input the following Form fields:

    A. Camp Name
    B. Image
    C. Camp Fees
    D. Date & Time
    E. Location
    F. Healthcare Professional Name
    G. participant count(starts at 0)
    H. Description

    Utilizing Formik or React Hook Form, validate and save the entered data in the database, ensuring accuracy and completeness. The camp details entered by the organizer will be shown on the available camps page for everyone to see.

12. **Manage Camps**

Show a table of camps created by the organizer with key details like name, Date & Time, Location, Healthcare Professional, and Include buttons for editing and deleting each camp.

Update: Give organizers the ability to change camp details using the "Update" button. They can access this feature through the simple **endpoint: /update-camp/:campId**

Delete: Allow organizers to swiftly erase camps from the system with the mighty "Delete" button. This action can be executed through the straightforward route endpoint: /delete-camp/:campId.

13. Manage Registered Camps

On this route, display all the camps data of users who have already registered. Ensure that the camp details (Camp name, Camp Fees, Participant Name) are organized and presented in a clear table format. The camp information should be retrieved from the database. Additionally, include fields such as payment status, payment confirmation status, and a cancel button.

Payment Status:
Clearly indicate whether participants have paid or unpaid, ensuring transparency in financial transactions.

Confirmation Status:
Initially set to "Pending." Upon clicking the "Pending" button, it updates to "Confirmed" once the payment is successfully processed by the participant.

Cancellation:
Organizers can easily cancel registrations with a friendly confirmation dialog. If payment is "Paid" and confirmation is "Confirmed," the cancel

button will be disabled. After cancellation, data will be promptly removed to keep records accurate.

Note : Here is a example of table data

Participant Name	Camp Name	Camp Fees	Payment Status	Confirmation Status	Cancel
John Doe	Summer Camp	$200	Paid	Confirmed	❌
Jane Smith	Adventure Camp	$250	Unpaid	Pending	✅
Alice Johnson	Sports Camp	$180	Paid	Confirmed	❌

Participant Dashboard (Private Route):

Note: **<u>This must be a dashboard layout</u>**

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

    A. Analytics.
    B. Participant Profile
    C. Registered Camps
    D. Payment History

15. **Analytics (Based on logged-in participant's data)**
    In this route, you're tasked with implementing a chart feature. The chart will display data related to participants' lifetime registered camps, such as camp names, fees, or any other relevant information. You have the freedom to design and customize the chart's appearance and functionality according to your preferences. However, ensure that the chart accurately reflects the data from camps registered by a participant.

    Note: Use [Recharts]() or a similar library to display analytics in this route. Visualize data from a participant's registered camps accurately.

16. **Participant Profile**

Allow participants to manage their profile information, including name, image, and contact details. They can click the "Update" button to open a form with their current information for editing.

17. **Registered Camps**

On this route, display all the data, showcasing only camps registered by each participant. Ensure clear organization and presentation of camp details (Camp name, Camp Fees, Participant Name) in a table format. Include additional fields such as payment status, payment confirmation status, a feedback button, and a cancel button.

Payment Status:

Show if each camp is unpaid or paid. For unpaid camps, provide a "Pay" button to redirect participants to payment. After successful payment via Stripe, display a notification with the transaction ID and update the "Pay" button to "Paid," disabling it.

Note: Upon successful payment, document transaction details in a dedicated collection for the participant's payment history

Confirmation Status:

Set initially to "Pending" and automatically updated to "Confirmed" upon organizer approval.

Cancellation:

Each camp row features a "Cancel" button for participants to revoke their registration. The button is active before payment but becomes inactive after payment. Upon cancellation, data is promptly removed from the database.

Feedback and Ratings:

Once payment is successfully completed and confirmed by the organizer, the feedback button becomes visible. Participants can provide feedback and ratings by clicking this button, and their responses are collected and stored in a feedback collection. Additionally, all participant feedback is displayed on the home page section.

Note : Here is a example of table data

Camp Name	Camp Fees	Participant Name	Payment Status	Confirmation Status	Cancel Button	Feedback Button
Camp A	$200	John Doe	Paid	Confirmed	[Cancel]	[Feedback]
Camp B	$150	Jane Smith	Pay	Pending	[Cancel]	N/A
Camp C	$300	Alice Johnson	Paid	Confirmed	[Cancel]	[Feedback]

18. **Payment History**

    Let participants easily see their camp payment history. They can check past and current transactions for each camp. Show details like Camp Name, Fees, Payment Status, and Confirmation Status in a simple table.

Authentication and Authorization:

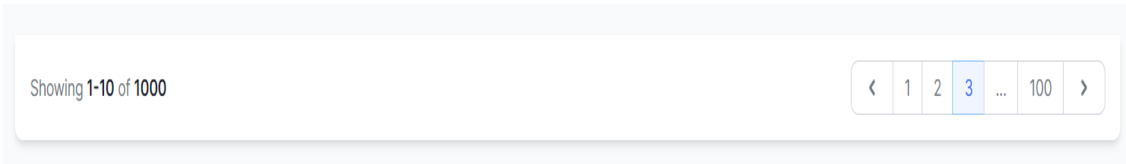
Join US Page: This is the page to implement authentication. Users will see a login form and add a link that will redirect the user to the Register page. Add at least one social login in both the Join Us and Register pages.

Implement react-hook-form in the registration & login page.

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.

Challenges Tasks

1. Implement pagination at the footer of all the tables you have implemented (show 10 rows at a time). For example,



2. Introduce a search at the top of all the tables you have implemented. allowing users to search for specific camps based on keywords(Camp name, Date, Healthcare Professional Name)or other relevant criteria. You can use reusable component for this.

3. Implement JSON Web Token (JWT) authentication for routes containing sensitive data. Utilize JWT libraries for both server-side and client-side to effectively manage authentication. **Use Local Storage to implement JWT.**

Optional Tasks

1) Added animation on your website using (framer motion, or AOS - Animate on scroll library
2) Implement Axios interceptor.
3) Avoid incorporating Daisy UI; instead, opt for alternative UI component libraries like Material Tailwind, Ant Design, or shadcn/ui.
4) Implement the react-awesome-button and React-select package.
5) Add one extra feature of your own in the dashboard.

Example:

a) Health Records Management (Participant)
b) Volunteer Management (Organizer)
c) Appointment Scheduling(Participant)

d) Community Forums(Participant)
e) Participant Management(Organizer)

What to Submit

1. **Organizer email:**
2. **Organizer password:**
3. **Front-end Live Site Link:**
4. **Client Side Github Repository Link:**
5. **Server Side Github Repository Link:**