

Mi-12 Assignment Requirement

assignment12_category_0014

🚩 : 0 [Check frequently for updates]

Objective:

This project aims to develop a Hostel Management system for a university, enabling administrators to manage student meals and food reviews using the MERN stack (MongoDB, Express.js, React.js, Node.js). Key features include student login, meal management by admins, and the ability to edit or delete food reviews and can manage their students.

Now, you will develop a fully functional Hostel Management system that allows students to log in, view, and review meals, and enables administrators to manage meal data and reviews efficiently. This system ensures streamlined operations and enhances the student experience in the university hostel..

Main Tasks

Key Rules:

1. GitHub Commits:

- Include a minimum of 20 notable GitHub commits on the client side.
- Include a minimum of 12 notable GitHub commits on the server side.

2. Readme File:

- 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.

3. Responsiveness:

- Make the website responsive for mobile, tablet, and desktop views.
- Ensure the dashboard is responsive as well.

4. Persistent Login:

- After reloading the page of a private route, the user should not be redirected to the login page.

5. Environment Variables:

- Use environment variables to hide Firebase config keys and MongoDB credentials.
- Hide `.env` credentials from both client and server Git repositories.

6. Alerts/Notifications:

- Show sweet alerts/toasts/notifications for all CRUD operations, successful authentication login, and sign-up.
- Do not use the default browser alert.

7. Data Fetching:

- Implement TanStack Query in all the data-fetching functionality (For GET method only).

8. Content:

- Do not use any Lorem ipsum text on your website.

Pages and Features

1. Home Page:

- **Navbar:**
 - Logo + website name, Home, Meals, Upcoming Meals, Notification icon, and Join Us (if not logged in) button.
 - If logged in, show the profile picture. Clicking the profile picture should display a dropdown with Username (not clickable), Dashboard, and Logout button.
- **Banner Section:**
 - A slider/banner with a heading title, short description, and search input field with a button.
- **Meals by Category:**
 - Implement a tab system with Breakfast, Lunch, Dinner, and All Meals categories.
 - Each tab shows at least 3 meal cards with title, image, rating, price, and details button.

- Clicking the details button redirects the user to the meal details page (`.../meal/${_id}`).
- **Additional Sections:**
 - Add one or two extra sections relevant to the site.
- **Membership Section:**
 - Display three cards for upgrading to premium packages: Silver, Gold, and Platinum, each with different prices.
 - Clicking each package card redirects to the checkout page (`.../checkout/${package_name}`).
- **Footer Section:**
 - Create a relevant footer for the website.

2. Meal Detail Page:

- **Displays meal details** including meal image, admin/distributor name, description, ingredients, post time, rating, like button, meal request button, and reviews.
- **Like Button (requires login):**
 - Clicking the Like button increases the reaction count by +1.
- **Meal Request Button (requires login):**
 - Clicking this button allows the user to request the meal (requires a package subscription).
 - A post request saves the meal and user info with a pending status.
- **Review Section:**
 - Users can post reviews and see the review count.

3. Meals Page:

- Show all meals with search functionality based on meal data.
- Implement filter-by-category and filter-by-price-range options.
- Implement infinite scrolling to load more meal cards as the user scrolls.

You can use these packages:-

- [react-infinite-scroller](#)
- [react-infinite-scroll-component](#)

4. Upcoming Meals:

- Display all upcoming meals as cards.
- Only premium(Silver/Gold/Platinum) users can give likes to each meal (one like per meal).

5. Checkout Page (Private Route):

- Dynamic route for purchasing packages.
- Display package details and integrate Stripe for payment.
- Show a confirmation modal after successful payment and save payment data to DB..
- Assign a badge based on the purchased package.

6. Join Us Page (Login/Register):

- Login and registration forms with social login options.
- Implement [react-hook-form](#) for form handling.
- By default a new user will hold the Bronze Badge and save the user information to DB.

7. User Dashboard (Private Route):

- **My Profile:**
 - Displays user name, image, email, and badges (Bronze, Gold).
- **Requested Meals:**
 - Show requested meals in a table with meal title, likes, reviews, status, and cancel button.
- **My Reviews:**
 - Show user reviews in a table with meal title, likes, review, edit, delete, and view meal buttons.
- **Payment History:**
 - Show payment history of the logged in user. If no data is found, show a relevant message .

8. Admin Dashboard (Private Route & Admin Route):

- **Admin Profile:**
 - Displays admin name, image, email, and number of meals added.
- **Manage Users:**
 - Show all users in a table with username, email, make admin button, and subscription status.
 - Implement server-side search by username and email.
- **Add Meal:**
 - Form to add meal details including title, category, image, ingredients, description, price, rating, post time, likes, reviews, admin name, and email.
 - Implement [react-hook-form](#) for form handling.
- **All Meals:**
 - Show all meals in a table with title, likes, reviews, distributor name, update, delete, and view meal buttons.
 - Implement sort by likes and reviews count.
- **All Reviews:**
 - Show all reviews in a table with meal title, likes, reviews_count, delete, and view meal buttons.
- **Serve Meals:**
 - Show requested meals with title, user email, name, status, and serve button.
 - Change status to delivered on serving.
 - Implement server-side search by username and email.
- **Upcoming Meals:**
 - Show all upcoming meals on a table, sorted by likes count.
 - Each table row shows a Publish button. After clicking the button, the specific meal will be added to all meals or mealsCollection..
 - Make Add Upcoming Meal button, when the admin clicks the button, show a modal to add an upcoming meal like **Add_Meal**

Challenges and Optional Tasks:

1. Challenges:

- Implement search functionality based on all information in the mealCollection using MongoDB indexing.
- Publish an upcoming meal to the main collection after receiving a minimum 10 likes.
- Implement JWT on login (Email/Password and social) and store the token and also implement the Axios interceptor on client site application.
- Implement pagination at the footer of all tables, showing 10 users/meals at a time.

2. Optional Tasks:

- Implement the "About Me" section on the user's profile with an edit button.
- Integrate [react-awesome-button](#), [React-select](#), and [react-modal](#) packages.

Submission Requirements:

1. **Assignment variant/category:**
2. **Admin Email:** Provide the admin email.
3. **Admin Password:** Provide the admin password.
4. **Front-end Live Site Link:** Provide the live site URL.
5. **Client Side GitHub Repository Link:** Provide the client repository link.
6. **Server-Side GitHub Repository Link:** Provide the server repository link.

Ensure the assignment meets all requirements, is fully functional, and is user-friendly.