

Contest Creation Platform / Contest Platform

Assignment12_Category_0006

Read the 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 Gobind 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 / Headlessui & have taken a card component from Daisy/Headlessu, 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.

DESCRIPTION :

Welcome to ContestHub, the ultimate platform where innovation meets recognition! ContestHub is a dynamic and user-friendly Project Contest Creation Platform designed to foster creativity, engage communities, and celebrate talent across various domains. Whether you're organizing a design competition, coding challenge, or any other creative contest, ContestHub is your go-to destination for seamless contest creation and efficient winner selection and also must **use font family in your web application.**

Main Requirements:

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)

HOME PAGE:

1. The navbar has a logo & website name, Home, and Login, All contests and Extra Routes/Section(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 have a search bar. Your search word should be based on contests tags. The search implementation functionality should be done in the backend. Also, you make this section unique and look beautiful section
3. This is a popular contest section show (minimum 5 contests) (popular filter by user participation count). For every popular contest show the following properties:
 - a. Contest Name
 - b. Image but design thinking unique
 - c. Attempted count/ participation count (participants count and is it will be the summation of participants)

- d. Details button (when clicking the details redirects to the details page)
- e. Contest short Description (use slice or ...)

Note: When a user is not logged in and if he/ she clicks on the View Details button, redirect him/ her to the login page. Without a login, you can not visit the single contests details page.

The section will have a Show All button that will redirect you to the All Contests page (/all-contests Route).

- 4. This section advertises the competition and shows the contest winner and some relevant text in this section, also inspires the user to participate in the contest. This section must look beautiful and have a unique design. You use some images, some text, etc.(this value is dynamic, contest winner, contest participation count, and Total contests winner count) and you can use static text if necessary.
- 5. This section highlights the best contest creator (minimum 3 creators)with properties: Name, image, contest name, and some contest description (this section must be dynamic (use contests participate count value) and do not use a simple slider, think of a creative design or any unique slider package).
- 6. Create a meaningful footer. (including website logo && name, copyright information , social media links (facebook and Linked-In)
- 7. create a 404 Error-Page also. It will show for invalid routes or errors with a button. By clicking the button it will take the user to the homepage.

⇒**All Contest Page:** Create an all-contest route and show all contests added by any content creators and In this section can an user will show all the contests or those contests which an admin has confirmed through the dashboard. Display the below instruction

- Contest Name
- Image but design thinking unique
- Attempted count / participants count .
- Details button (when clicking the details redirects to the details page)
- Contest short Description (use slice or ...)
Implement a tab view here according to the contest type/tags.(see add contest feature for more info.)

⇒**Contest Details Page (Private)**: When the user clicks the contest details button redirects to the login page and after login redirects to the details page (when the user logs login). This page shows the contest details,

- a. Contest Name
- b. image/banner
- c. Attempted count/ participation count (Dynamic: By default 0, when any user completes contests payment, increase the count)
- d. Contest Description and Details.(Task or any relevant info)
- e. Contest Prize (Money or others)
- f. Contest Winner Name and image. (if the deadline is over and the winner is decided. The contest winner is decided Only Contest Creator)
- g. Show Deadline Counter. This should show a dynamic timer and show not-available (Note: use conditionally, when the contest date end users do not participate that contest shows the message “not available”).
- h. Contest Registration Button. When the user clicks on the Contest Registration button, redirect him/her to the payment page. If the payment is completed send the details to the database, and register him/her for the contest.

ROLE MANAGEMENT:

Implement **three** roles:

1. **Admin**: The admin can approve the contest, change the user role, confirm contests, and Comment for pending Contests.

2. **Contest Creator:** The contest creator can add a contest, update the contest, and Declared the winner of the contests.
3. **User:** The user Only Participated Contests, updated user information and saw her profile/percentage(Example: He participated in 5 contests and won 3 contests so he won 60% contests, show her profile), and won participation details. Details User Dashboard page.

A. User Dashboard (Private): When a user clicks on the Dashboard, he/she will be redirected to the dashboard route a page where there will be the following routes:

1. My Participated Contest.
2. My Winning Contest Page:
3. My Profile:

1. My Participated Contest: Show all the contests to the user for which he/she has **completed payment** (show the payment status) . Here the user can participate in each contest till the deadline. Implement a sorting feature to show My upcoming Contests to the user.

2. My Winning Contest Page: Show his/her success here and also details won Contests. (Show your creativity here).

3. My Profile: Implement chart for the following:

- Show The Win percentage (by attempted and complete)
- Let a user update his display name and profile picture from this route and also add one information user: User add any property(Exam: Address, location, etc)

B. Creator Dashboard (Private): When a user(creator) clicks on the Dashboard, he/she will be redirected to a page where there will be the following routes:

- A. Add Contest,
- B. My Created Contest,
- C. Contest Submitted Page

A. Add Contest Page: This page will have a form with the following fields:

- a. Contest Name
- b. image
- c. Contest Description
- d. Contest Price
- e. Prize money or others.
- f. Task Submission text instruction (**Details of what tasks a user will submit, or what tasks will be submitted.**)| .
- g. Contest Type/Tags [Tags should be Image Design Contests, Article Writing, Marketing Strategy, Digital advertisement Contests, Gaming Review, Book Review, Business Idea Concerts, and Movie Review Don't Add any other tags.] Use a dropdown to select the tag name. We recommend the Headless UI dropdown.

h. Contest Deadline. (use react-datepicker)

B. My Created Contest: If a creator visits this page, he/she can see all the contests he/she posted. Show them in tabular form. Each row will have edit and delete buttons for the contest, By default the contest status will be pending. If an admin approves the contest the status changes to accepted. The contest will then appear on the all-contest page. Also if an admin accepts the contest the creator can not edit and delete the contest. Only an admin can delete the contest after this. Also, keep a see submission button for every contest. Clicking on this redirects him/her to the contest-submitted page.

C. Contest Submitted Page: If a creator visits this page, he/she can see all the submissions and multiple contests of creator posted. Show The all particular Contents Title and Prize, When clicking the contest title then shows all user-submitted contests. Show the contest name and submission details (participant name, email, and submitted task(use drive or any link). Beside every task keep a button named Declare Win. Clicking on this the creator can declare the winner for that

particular task and also remove all Declared Win buttons for every task and show Un-success.

Note: In a contest, there must be only one Winner.

Admin Dashboard(Private) :

When a user (admin) clicks on the Dashboard, he/she will be redirected to a page where there will be the following routes:

- a. Manage User,
- b. Manage Contests,

A. Manage User Page: If an Admin visits this page, he/she can see all the users and Show them in tabular form. Each row will have. user email and name Here admin can toggle the role of any user(admin \Leftrightarrow Contest creator \Leftrightarrow Normal User) and Delete Users. Also, the Admin Can block users/creators(when users. creator block he does not can add or participate in contests or any operations. Only view access). Admin can unblock users' creators (when the user blocks \Rightarrow unblocks Admin change rolls normal users).

B. Manage Contest Page: If an admin visits this page, he/she can see all the contests. Show them in tabular form with Three buttons(Delete, Confirm, and Comment). Show some information on all the contests in each row. Whenever an admin deletes a contest, the contest gets removed from the database. When the admin confirms the contests and then publishes the contests, Otherwise contests do not show the website. Also Admin clicks the comment button to open a modal and show the input box to write comments and submit also these comments Show the creator dashboard ~ My Created Contest Page ~ (**Besides she/he contests Right side**)

Login and 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

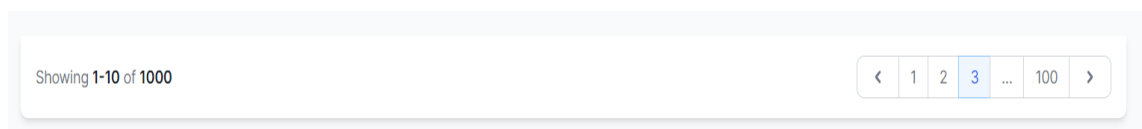
Registration Page: The Registration page will have the Email/Password form having the following fields:

- Name
- Email
- Password
- Photo URL

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 a /leaderboard route. Here This should be dynamic (Contests Win Count) . Implementation of this feature is up to you.
2. **Implement react-hook-form in all form pages.**
3. Implement JWT on login (Email/Password and social) and store the token local storage/cookie and also API secure minimum 3 (use verifyToken function and user verify) .
4. Implement pagination at the footer of all the tables and cards you have implemented (show 10 data at a time). For example,



5. Try to create two extra routes(The extra section you created mentions and add your repository Redmifile), sections relevant to contest platforms, for example The Contests Timeline Visualization section and Participations Progress Tracking.
6. Theme Customization Add a Theme Toggling Button for changing themes from light to dark / dark to light (use local storage WHEN after reload not affected by select theme). By changing the theme, it will make the full system dark / light based on user interaction.
7. Use Captcha for Registration

Note: Not added contact page, review, footer, slider Q/A, etc. Thinking and adding creative design.

- [No hints will be provided bonus tasks. Don't ask for hints. Make your own decision.]

Optional Tasks

You have to implement any two tasks from the following:

- 1) Added animation on your website using ([framer motion](#), or [aos](#))
- 2) Implement Axios interceptor.
- 3) Implement a package system for creating a contest. That is, a contest creator can purchase the amount of contests he can post.

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) [Firebase Hosting Setup Complete Issue](#)
4. Host your server-side application on Vercel. If needed, you can host somewhere else as well. [How to deploy a Node/Express server using Vercel CLI](#) [Some Common Vercel Errors](#)

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.

What to Submit

1. Assignment Category:
2. Admin email:
3. Admin password:
4. Front-end Live Site Link:
5. Client Side Github Repository Link:
6. Server Side Github Repository Link:
7. If possible submit creator's email and password.