

Challenges & Solutions:

1->Search design for web App->

To find user interactive design is very difficult task.so first challenge I faced to find user interactive design for our Web-based Task Management Dashboard.

Solutions ->

For solutions this problem first i saw many video on YouTube for design reference but almost all video had same pattern design. So I go to google and find design in which i got images design. I did mobile responsive by myself.

2->Data Inject->

Handling asynchronous operations, such as making API requests to fetch and update task data, can be challenging. It's crucial to ensure that the user interface remains same and same as according to requirements during these operations.

Solution->

Utilize asynchronous programming techniques, such as JavaScript Promises or async/await, to manage API calls efficiently. And also I used TypeScript for avoid errors. And also I break whole app in component for better customization.

3->Validation

Implementing client-side form validation to ensure that task details are entered correctly can be difficult. Because we do not want to push wrong format data in database. Validation rules may include checking for empty fields and validating due dates.

Solution->

For overcome this problem, first I make an object of all state then before calling Api. First, I check every validation which was given in the task then call Api and inject data into database. And also, I show appropriate error via toast message if any task does not follow validation rule. If we do not validate our data, our database can go in inconsistent state.

4->Responsive Design

Designing a dashboard that looks and functions well on both desktop and mobile devices can be challenging due to differences in screen sizes and resolutions.

Solution->

Implement responsive design techniques using CSS media queries to adapt the layout and styling to various screen sizes. Test the dashboard on multiple devices and use frameworks like Material-UI, tailwind css.

5-> Data Synchronization

Keeping data synchronized between the every component of frontend can be complex, especially in multi-functionality web app.

Solution->

For overcome Data Synchronization problem I used many useState in our web app like after delete and edit our task listing page is also update according to action.

6->Backend Challenge->

So in backend , first challenge is handle CORS error so for this problem I used npm cors library. Second problem is handle proper code architecture so for this problem i used MVC architecture. In backend error handle is very important so for this i used custom error class and proper middleware.

Security & Optimization

1-> Ensuring the security of user data, especially in a task management application, is critical. Challenges include protecting against, cross-site scripting (XSS), and ensuring data privacy. Although this term is not use in our web app but for complex task management web app these are very important term. For this problem sanitize user inputs to prevent XSS attacks. Implement user authentication and authorization to control access to sensitive data. Store passwords securely using hashing algorithms.

2-> Define and enforce role-based access control to restrict what users can do within the application. For example, regular users should not have access to administrative functions. Authorization ensures that users only see and modify their own tasks or tasks they are allowed to access.

3-> Implement detailed error handling to avoid exposing sensitive information to potential attackers. Use generic error messages for end-users and log detailed error information for developers.

4-> Use code splitting to break your application's code into smaller, more manageable chunks. This allows you to load only the necessary code for the current route or component, reducing initial load times.

5-> Consider implementing SSR with frameworks like Next.js. SSR can improve initial page load times and SEO performance by rendering pages on the server before sending them to the client.