**Project Name** : Task Manager

**Tech Stack Used** : MERN (Mongodb, Express, React, NodeJS)

**Project Link :**  <https://task-manager-dnd.netlify.app>

**Github Links:**

* Frontend: <https://github.com/pratikbusiness/task-manager-frontend>
* Backend: <https://github.com/pratikbusiness/task-manager-backend>

**Introduction** :

Firstly I would like to thank you for this mini project opportunity and for considering me as a viable candidate for the applied post. Moving to the project it is a simple task management app, with three status for a particular task ToDo, Doing and Done. It has a drag and drop feature to change status of the card and a soothing UI, Altough there is a lot of scope in improving UX, but given the limited time for implementation, this is all what I could harvest in the given timeframe. The following project has very limited screens.

* Login
* Signup
* Page Not Found
* Task Management Screen

**Personal Considerations :**

* Multiple users with a proper backend ( Alternate thing was to cache the state but db was my persoal choice )
* Small screen drag and drop should be using edit button and not with drag and drop.
* Tasks can be rearranged / reordered.
* Task are free to move from one to another column without any condition or restriction for example something like a user can direclty move from todo to done stage. We can also implement such that the user is told to first move the task to doing and then done If we wish so in the near future.
* Mobile and Tablet layout should be bit different from UX standpoint. (Currenlty couldn’t finish it, still left).

**Decisions Made** :

* The tech stack was clear earlier also as I am a MERN developer mostly invested in frontend.
* The tricky was to structure the application properly and make it flexible enough to accommodate any futher feature addition or any other changes
* Drag and Drop implementation was a very curcial decision. Comparing the avialable libraries or options that were in hand like jquery, plain html css js implementaiton, react dnd, react beautiful dnd.
* React beautiful dnd is something I had already worked on so it was much easiier for me comparatively.
* Other libraries like bootstrap, aixos, react router, etc were used which are very popular and very stable.

**Known Isssues**

* UX
* Dark Mode missing
* Drag and drop if dropped outside columns it gets stuck ( Solution for now is we need to refresh or start dragginf some other element)
* Validation can be improved a bit.
* Responsiveness can also be improved a bit.
* Website when idle for sometime the first request takes a bit longer to get fulfilled. That’s a limitation from the hosting provider (Heroku) where the backend is hosted.

**Deployment:**

The deployment for the following project is done by using various free services available.

* React JS (Frontend) – Netlify
* Express JS (Node JS framework)(Backend) – Heroku
* Mongodb (Database) – Mongodb atlas\

Currently No Backup strategy is implemented for now in mongodb and on the other hand heroku and netlify provide their own set of features for rollback and other such features.

**Code Explaination (ReactJS)**

* Environment.js : Works like a mini config file.
* App.js : Place where we wrap our website with context for declaring base urls and other so that deplyoment seems much easier.
* Context folder is the folder which has two important app wrappers auth and prefix . Prefix context is used to add prefix to fetch urls so when we move to deployment we just need to change it in one place and everything is taken care of for all the fetch and associated urls. Another file is auth context which soley takes care of the authentication part fort the websie.
* Routes/ index.js : Place where we define our routes with the help of react-router library. A basic 404 page is also implemented in the following project.
* Layouts is the folder where I have my two layouts MainLayout and ProtectedLayout, protected layout has useEffect to check if the logged in session is still valid on page navigation.
* In the same layouts folder there is another folder named templates which has actual header and footer components which are imported in layouts.
* Pages folder has pages in there which are login signup 404error page and last but most important taskmanager page.
* Task managerpage is futher divided into smaller components for better code readability and maintainance.
* Add task, edit task and show tasks are the components built which are imported in index.js located in the same folder.

**Future Scope of Improvement :**

1. Currently just for the sake of project we have not implemented any verification step for the user registration process.
2. More features like collabration between users and role based system implementation.
3. Drag and Drop feature needs refinement.
4. UX can overall be improved.
5. Change Password and Settings and basic options like such should be added.
6. Dark mode is something that can be implemented going further.

**Conclusion :**

To sum it all, it was a very good exp working on some mini project with a close deadline.

Thank you for your time and consideration.