Proposed use cases:

1. Create a new task with title, description, time, and location
2. View the information of an existing task
3. Edit an existing task
4. Delete an existing task
5. Add a category to an existing task
6. Delete an existing category of an existing task
7. Search for tasks based on the task name
8. Search for tasks based on the task category
9. View the complete task list

Execution plan:

Plan for the developer:

1. Complete the basic backend and frontend implementation of the proposed functionality by 28 December.
2. Polish the frontend using the material UI library of React by 5 January.
3. Deploy the application onto heroku by 10 January.

Plan and user guide of the user:

1. Enter the task list and see the list of existing tasks
2. Add a new task by clicking the new task button or edit or delete an existing task by clicking the respective button of the corresponding task
3. Search for an existing task by entering the keyword or selecting the category and clikcing the search button

Suggestion:

Maybe there can more guidance on the heroku deployment, especially regarding the measures to take to make sure that the application behaves consistently when it is run on the local host and the heroku platform.

Current problem:

It seems that the front end component is rendered differently when the application is deployed on heroku. To be more specific, while the page is supposed to be divided into the left panel and the right panel, as is the case on local host, the page is divided into the top part and the bottom part instead when the application is deployed on heroku.