Execution Details:

- Main Page:
  1. *Appearance:* Will show the current month, week and day at the top of the page. The center of the page will show the todo list with incomplete tasks.
  2. *Form*: Enables users to add todo items to the list through the form by typing into the row located at the bottom of the todo list and clicking on the arrow button located at the right.
  3. *Search bar:* Will be a form, with an input and a button to submit the input.
- Todo list:
  1. *Appearance*: It will be represented in the form of a column, using table rows in HTML or CSS Grid.
  2. *The todo items in the list*: Will be represented using a table of rows; each row for each todo item in the list.
  3. *Tags*: Will be an array of strings, which will be empty by default.

Basic Use Cases:

- Main Page:
  1. *Search bar*: Users can use the search bar located at the top of the to-do list, to search for particular todo items by categories/tags.
  2. *Counter*: Have a counter located at the top left of the page to show the number of incomplete tasks due on that day(Additional Feature).
- Todo List:
  1. *Addition of tasks*: Use the form at the bottom of the to-do list to add a task to the list, by typing into the input row, and clicking on the arrow button at the right once done.
  2. *Deletion of tasks*: Click the trash button located at the right of each task(when hovering over the tasks).

3. *Crossing off tasks from the list*: Click the content(text) of the task if you would like to strike off completed tasks. (OR: Have a checkbox at the left side of the task. Upon clicking it, the task will be considered as completed)
4. *Order of tasks in a list*: By default, all tasks will be in the order of addition. If the user would like to reorder tasks, he can do so by manually moving the rows or change the settings to reorder by due date.
5. *Editing todo items*: Users can edit todo items by clicking directly on the text of a todo item.
6. *Addition of tags to todo items*: Users can choose to add tags to the todo item that they have added to the list by clicking on the row containing the todo item. Users can add tags to each todo item to allow for easier searching when the list becomes long. (OR: Have a + sign at the top right of the todo list, and upon clicking it, users can add more items to the list as an input row will appear at the top of the list, below the search bar)
7. *Deleting tags from todo items*: Click the row containing the tags and manually delete each unwanted tag. (OR: Click the 'x' located next to each tag to delete a tag.)

Challenges I'm facing:

- New to Ruby on Rails, React, SQL and Databases.
- Has been two years since I last touched HTML, CSS and web development so there were a lot of things I needed to read up on to refresh my memory, on top of doing a couple of new projects on my own to practice what I've learnt.
- Pre-defined functions used in Source(CS1101S) mostly did not exist in Javascript or were under different names. Object-oriented programming(OOP) was not taught in CS1101S. Had to take up a JS course on the basics of OOP and learnt DOM with the help of Javascript30 by Wes Bos and my own mini-projects like Etch-A-Sketch and a Calculator, created using HTML, CSS and JS.