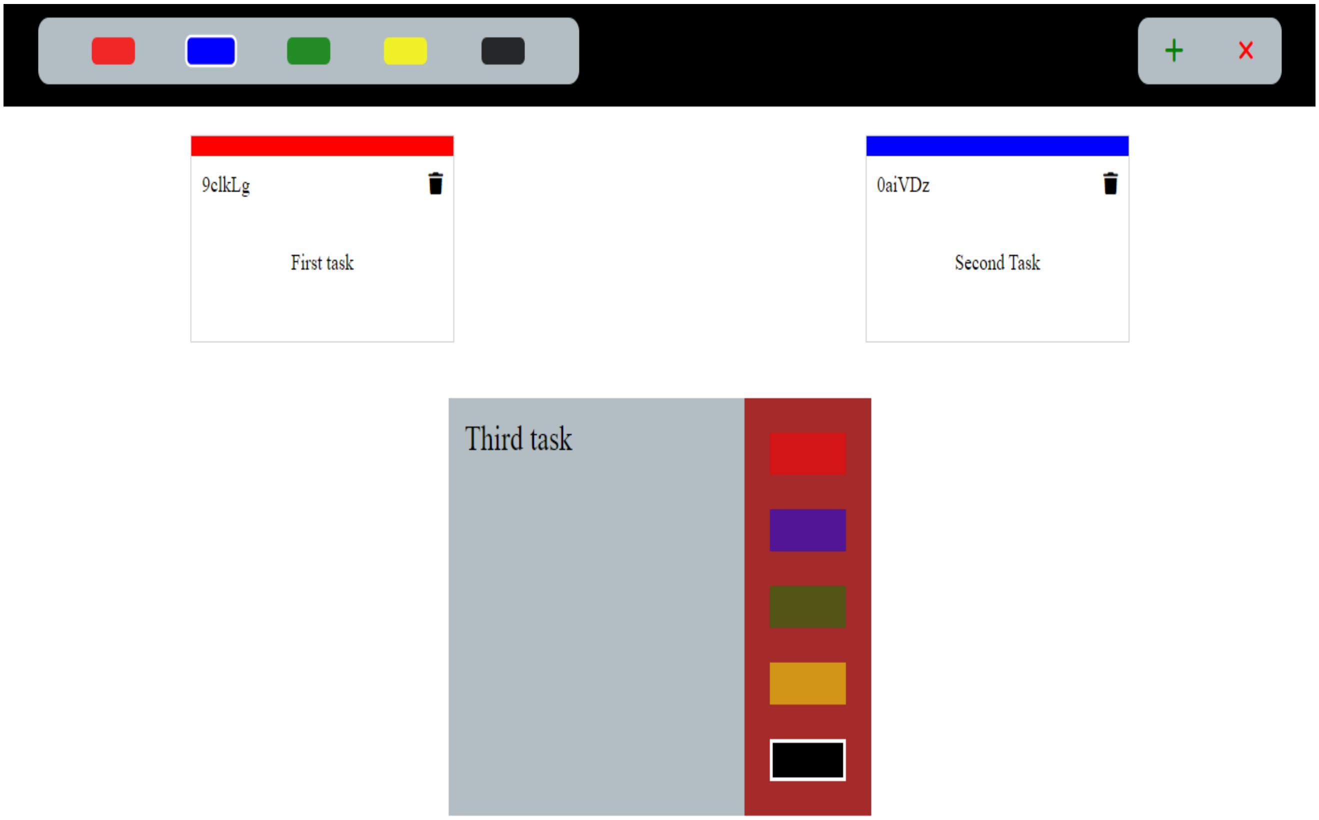

9clkLg

First task

0aiVDz

Second Task

Third task

Advanced To-Do

Project Link : https://github.com/jaydip1235/placewit-js/tree/master/Ticket-App

Jaydip Dey
Jadavpur University
https://linktr.ee/jaydipdey

This application allows users to create, view, filter, edit and delete tickets based on their requirements. The ticket data is stored in the browser's local storage, meaning that tickets persist across browser sessions.

Functionalities

Ticket Creation

The ticket creation process is started by clicking on an element with class "open-modal". The openTicketModal function is triggered which creates a modal for ticket input. The modal contains editable text and color filters. When text is inputted into the text area and the Enter key is pressed, the ticket is created with the selected filter color, a unique id, and the text content.

Ticket Storage

Tickets are stored in the browser's LocalStorage. A ticket is represented as an object with a unique id, a filter color, and its text value. All ticket objects are stored as an array in LocalStorage, serialized into a JSON string.

Ticket Deletion

Each ticket contains a delete button represented by a trash icon. Clicking on this button triggers an event listener, which removes the ticket element from the DOM and also removes its associated data from LocalStorage.

Ticket Filtering

The code has an event listener setup on the ".filter" elements. Clicking on these elements toggles the visibility of tickets based on their associated filter color. The color of a ticket can also be changed by clicking on its header.

User Interface (UI) elements

- Filters: These are UI elements that allow users to view tickets based on their color classifications. The colors used are red, blue, green, yellow, and black.
- Modal: The modal is a dialog box/pop-up window that is used to create new tickets. It is displayed on the click of the "openModal" button and closed by the "closeModal" button.
- Tickets: Each ticket is a UI element that displays the ticket content and includes a header colored according to its filter classification.

Events

The following events are handled in the code:

- Opening and closing the ticket modal: The click event on "openModal" and "closeModal" buttons triggers the opening and closing of the ticket creation modal respectively.
- Filter selection: The click event on each filter element allows for the selection and deselection of filters.
- Ticket creation: A keypress event is used within the ticket modal. If the 'Enter' key is pressed when text is present in the modal's text field, a new ticket is created.
- Filter change on ticket: The click event on the ticket header allows users to change the ticket's filter color by cycling through the available colors.
- Ticket deletion: The click event on the ticket delete button removes the ticket from the view and local storage.

Data Storage

The application uses the browser's LocalStorage as a simple database to store ticket data. Each ticket is stored as an object, with properties for the ticket filter color, ticket value (text), and a unique ticket identifier. The ticket data is retrieved and parsed from LocalStorage whenever tickets are loaded or filtered.

Functions

The following are the key functions implemented in the code:

- selectFilter(): This function implements the logic for selecting a filter, deselecting any currently selected filter, and loading the corresponding tickets.
- loadSelectedTickets(): This function fetches the tickets from LocalStorage that match a particular filter and calls the function to display them.
- loadTickets(): This function fetches all tickets from LocalStorage and calls the function to display them.
- openTicketModal(): This function opens the ticket creation modal and sets up the required event listeners.
- closeTicketModal(): This function closes the ticket creation modal.
- handleKeyPress(): This function handles the 'Enter' keypress event within the ticket modal's text field, initiating ticket creation.
- saveTicketToDb(): This function saves a newly created ticket to LocalStorage.

- appendTicket(): This function creates a new ticket DOM element from ticket data and appends it to the ticket container, also setting up the necessary event listeners for filter color change and deletion.

Libraries Used

The uuid function, which is used to create unique ids, comes from a library uuid.

Note : You may also highlight the point on Edit Tickets that you did in Task 5