Class 18 - Kanban Board - 1

1. Brief description about the Project (requirement doc)
    a. The project is a web-based Kanban board application designed to enhance task management and productivity for individuals.
    b. At its core, the Kanban board facilitates organizing tasks into a visual format, allowing users to easily manage their workloads and prioritize tasks according to urgency and importance.
    c. The application is built using HTML, CSS, and JavaScript, ensuring a responsive and interactive user experience across various devices and screen sizes.

2. Overview Features and Demo of the Project
    a. Dynamic Task Addition:
        i. Users can create new tasks via a modal pop-up, which becomes visible upon clicking the add (+) button. This feature is essential for inputting new tasks into the board, allowing users to detail what needs to be done.
    b. Task Priority Color Coding:
        i. Each task can be assigned a priority level represented by colors (light pink, light green, light blue, black). This visual differentiation helps users quickly identify task

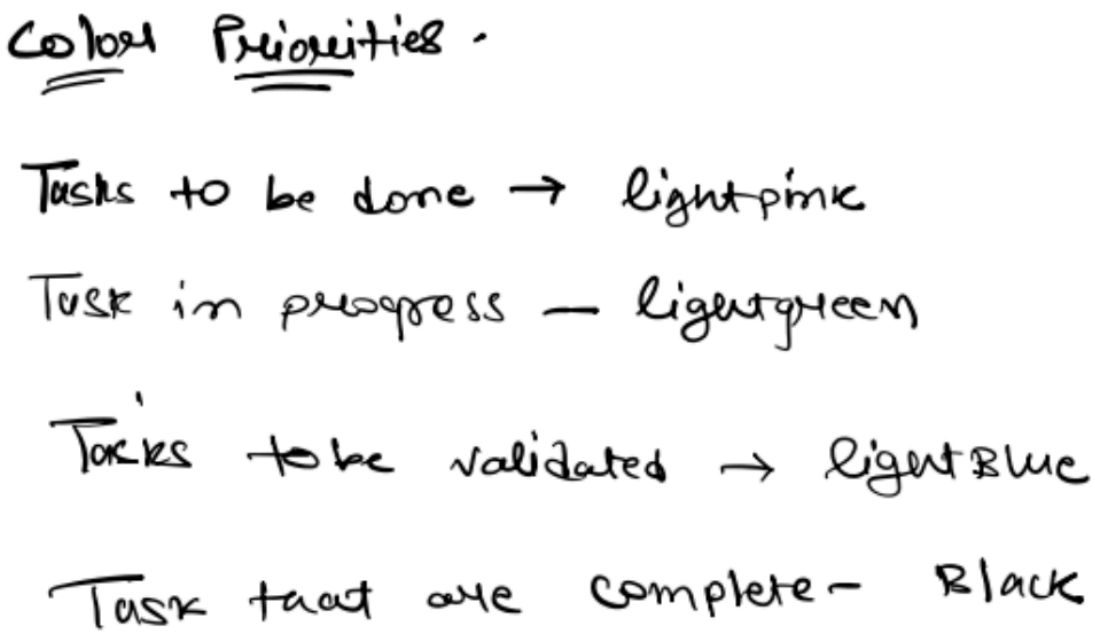
importance or categorize them based on priority, enhancing task management efficiency.

c. Task Filtering by Color:
    i. The Kanban board provides functionality to filter tasks based on their priority color. Clicking on a color in the toolbox shows only tasks of that color, while a double-click resets the filter, showing all tasks.
    ii. This feature aids in focusing on tasks of a specific priority or category, making it easier to manage large numbers of tasks.
d. Editable Task Descriptions:
    i. Tasks are editable after their creation. Users can unlock a task to edit its description directly on the board and lock it back once done. This inline editing feature makes it convenient to update task details without needing a separate edit mode or pop-up.
e. Local Storage for Persistence:
    i. The application uses the web browser's local storage to save tasks, ensuring that user data is preserved between sessions. This means that tasks remain on the board even after the browser is closed or refreshed, providing a persistent user experience.
f. Task Deletion:
    i. Users can delete tasks from the board. A delete (or remove) mode can be toggled, after which clicking on a task will remove it. This feature allows for the easy

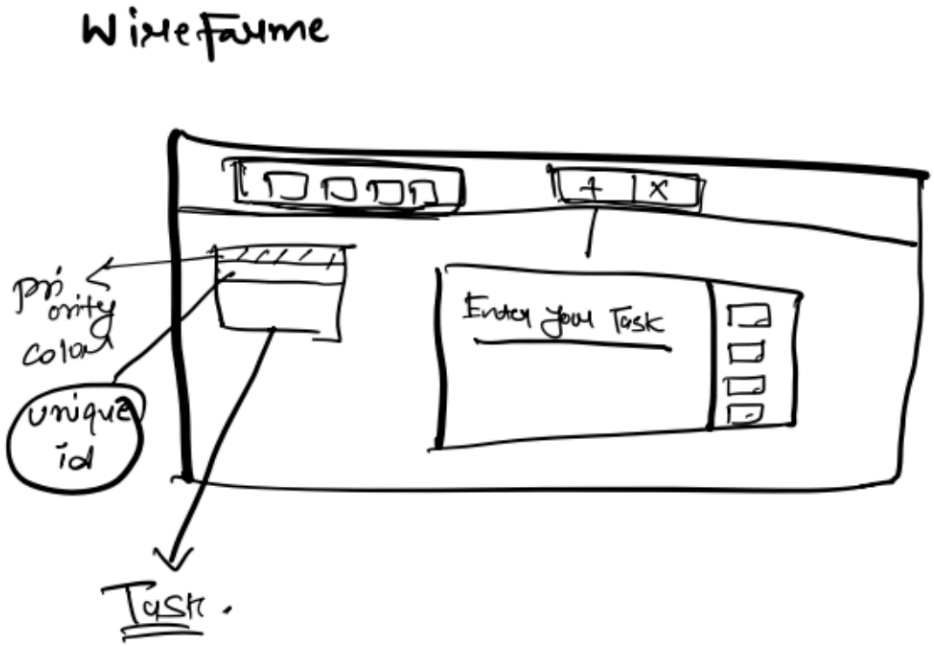
removal of completed or unwanted tasks, maintaining a clean and up-to-date task list.

g. Priority Color Selection in Modal:
    i. When adding a new task, users can select its priority color within the modal through a set of color options. This selection process lets users categorize or prioritize the task right at the point of creation.
h. Unique Task Identification:
    i. Each task is assigned a unique ID, ensuring that tasks can be individually identified. This is particularly useful for managing tasks programmatically, such as updating or deleting specific tasks based on user actions.
i. Interactive UI Elements:
    i. The application includes interactive UI elements such as buttons for adding and removing tasks and color selections for priority. These elements are designed to provide a user-friendly experience, making task management both efficient and enjoyable.
j. Responsive Design:
    i. The project is styled with responsiveness in mind, meaning it's designed to look and function well on a variety of devices and screen sizes. This adaptability enhances accessibility and user experience across different platforms.

Wireframe of the project:

1. 

2.

3. 

Desiging the toolBox

1. Start a blank HTML Page and start designing the first section of the app (The toolbox Container) basically the Navbar
2. The Toolbox will have-
    a. 4 diffrent colored boxes(buttons) clicking on them our tasks will get filtered , the color which will be clicked that task will be visible
    b. Two buttons one Add (+) and one Cross (X) we will get the icons form font awesome
3. Let's create the toolbox

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <div class="toolbox-cont"></div>
</body>
</html>
```

4. Inside the toolbox, there are two main sections: one for priority colors (toolbox-priority-cont) and one for action buttons (action-btn-cont).
    a. Priority Colors Section
    b. code

```
<div class="toolbox-cont">
      <div class="toolbox-priority-cont">

      </div>

   </div>
```

    c. A container for priority color indicators. It's styled to visually differentiate between various priority levels through color.
    d. Code

```
<div class="toolbox-cont">
     <div class="toolbox-priority-cont">
       <div class="lightpink color"></div>
       <div class="lightgreen color"></div>
       <div class="lightblue color"></div>
       <div class="black color"></div>
     </div>
   </div>
```

e. Action Buttons Section
f. code

```
<div class="action-btn-cont"></div>
```

g. A container for the action buttons, including add and remove buttons. It's designed to allow user interactions with the toolbox.
h. We will be using font awesome to get Icons
    i. Using Font Awesome icons in a web project involves a few straightforward steps. Font Awesome provides a vast library of icons that can be easily integrated and styled directly within your HTML and CSS. Here's how you can use Font Awesome icons
    ii. Step 1: Include Font Awesome Library
    iii. . Font Awesome offers a CDN (Content Delivery Network) link that you can use to include the latest version of the icons directly from their servers.
    iv. code

```
<script
    src="https://kit.fontawesome.com/589957875e.js"
    crossorigin="anonymous"
  ></script>
</head>
```

    v. Step 2: Use Font Awesome Icons in HTML
    vi. After including the Font Awesome library, you can start using the icons in your HTML. To do this, you

add an <i> (or) element with the class names that correspond to the specific icon you want to use. Font Awesome icons have two parts to their class names: the style prefix and the icon name.

vii. code

```
<div class="action-btn-cont">
       <div class="add-btn">
         <i class="fa-solid fa-plus"></i>
       </div>
       <div class="remove-btn">
         <i class="fa-solid fa-xmark"></i>
       </div>
     </div>
```

viii. fa-solid is the style prefix indicating you're using the solid style of the icons.

ix. fa-plus and fa-xmark are the icon names for the plus and cross icons, respectively.

x. Inside, individual div elements for the add and remove buttons are defined, each containing an icon from FontAwesome (plus and x-mark icons). The classes add-btn and remove-btn are used for specific styling and interactions.

5. CSS - Styling the Toolbox

   a. code

```
* {
 box-sizing: border-box;
}
body {
```

```
 margin: 0;
 padding: 0;
}
.toolbox-cont {
 height: 5rem;
 background-color: #4b4b4b;
 display: flex;
 align-items: center;
}
.toolbox-cont > * {
 margin-left: 4rem;
}
```

b. The .toolbox-cont class styles the toolbox container, setting its height, background color, and centering its child elements using Flexbox.
c. The > * selector increases the left margin of all direct children, spacing out the priority color and action button containers.
d. Priority Colors Container
    i. code

```
.toolbox-priority-cont {
 height: 3.5rem;
 width: 18rem;
 background-color: #3d3d3d;
 display: flex;
 align-items: center;
 justify-content: space-evenly;
}
.toolbox-priority-cont > *:hover {
 background-color: #485460;
```

```
}
```

ii. Styles the priority color container, setting dimensions, background color, and evenly spacing the color indicators. The :hover pseudo-class changes the background color of any color indicator on hover, enhancing user interaction.

e. Color indicators

i. code

```
.color {
 height: 1.5rem;
 width: 3rem;
}
.lightpink {
 background-color: lightpink;
}
.lightgreen {
 background-color: lightgreen;
}
.lightblue {
 background-color: lightblue;
}
.black {
 background-color: black;
}
```

f. Action Buttons Container

i. Now i want to make my action buttons in same row

ii. code

```
.action-btn-cont {
```

```
  height: 3.5rem;
  width: 8rem;
  background-color: #3d3d3d;
  display: flex;
}
.action-btn-cont > * {
  display: flex;
  width: 50%;
  font-size: 2rem;
  color: white;
  justify-content: center;
  align-items: center;
}
```

iii. Defines the styling for the action button container and its children. Each button is designed to take up half the container's width, centering the FontAwesome icons with white color and significant font size for visibility.

iv. Button Hover Effects

v. code

```
.add-btn:hover,
.remove-btn:hover {
 background-color: #4bb543;
}
```

6. Complete html so far

```
<!DOCTYPE html>
<html lang="en">
 <head>
   <meta charset="UTF-8" />
   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

```
    <title>Document</title>
    <script
      src="https://kit.fontawesome.com/589957875e.js"
      crossorigin="anonymous"
    ></script>
    <link rel="stylesheet" href="style.css" />
  </head>
  <body>
    <div class="toolbox-cont">
      <div class="toolbox-priority-cont">
        <div class="lightpink color"></div>
        <div class="lightgreen color"></div>
        <div class="lightblue color"></div>
        <div class="black color"></div>
      </div>
      <div class="action-btn-cont">
        <div class="add-btn">
          <i class="fa-solid fa-plus"></i>
        </div>
        <div class="remove-btn">
          <i class="fa-solid fa-xmark"></i>
        </div>
      </div>
    </div>
  </body>
</html>
```

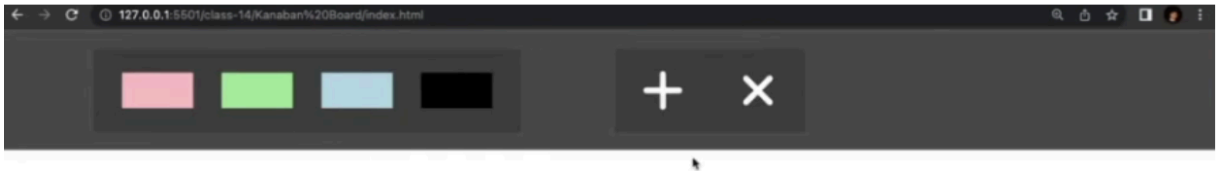
8. Complete css till now

```
* {
  box-sizing: border-box;
}
body {
  margin: 0;
  padding: 0;
}
.toolbox-cont {
  height: 5rem;
  background-color: #4b4b4b;
  display: flex;
  align-items: center;
```

```
}
.toolbox-cont > * {
 margin-left: 4rem;
}

.toolbox-priority-cont {
 height: 3.5rem;
 width: 18rem;
 background-color: #3d3d3d;
 display: flex;
 align-items: center;
 justify-content: space-evenly;
}
.toolbox-priority-cont > *:hover {
 background-color: #485460;
}

.color {
 height: 1.5rem;
 width: 3rem;
}
.lightpink {
 background-color: lightpink;
}
.lightgreen {
 background-color: lightgreen;
}
.lightblue {
 background-color: lightblue;
}
.black {
 background-color: black;
}

.action-btn-cont {
 height: 3.5rem;
 width: 8rem;
 background-color: #3d3d3d;
 display: flex;
}
.action-btn-cont > * {
 display: flex;
```

```
 width: 50%;
 font-size: 2rem;
 color: white;
 justify-content: center;
 align-items: center;
}

.add-btn:hover,
.remove-btn:hover {
 background-color: #4bb543;
}
```

Output-



Designing the Ticket

1. This will be a task ticket that will appear in the application, including elements like a color-coded priority indicator, a ticket ID, a task description area, and a lock icon for editing control.
2. HTML - add it next to tool box div

```
<div class="main-cont">
     <div class="ticket-cont">
       <div class="ticket-color"></div>
       <div class="ticket-id">12345</div>
       <div class="task-area">Random Task</div>
       <div class="ticket-lock">
         <i class="fa-solid fa-lock"></i>
       </div>
```

```
    </div>
  </div>
```

3. Ticket related css

```
/* ticket css */
.main-cont {
 display: flex;
 gap: 2rem;
 justify-content: center;
 padding: 2rem;
 flex-wrap: wrap;

 /* As many tickets can be added here so this will make sure
that the design is neat to accomodate multiple tickets */
}
```

   a. Sets up a flexible container with a gap between items for spacing, centered alignment, and wrapping behavior. This ensures that no matter how many tickets are added, the layout remains neat and scalable.

4. Ticket container

```
.ticket-cont {
 height: 12rem;
 width: 15rem;
 background-color: coral;
}
```

   a. Defines the size and background color of each ticket container. These dimensions ensure that each ticket has sufficient space to display its content clearly.

5. Ticket color (header of the ticket)

```
.ticket-color {
 height: 1rem;
}
```

    a. Sets the height of the priority color bar. The width is not specified, so it extends to the full width of its parent container (.ticket-cont).

6. Ticket id

```
.ticket-id {
 background-color: yellow;
 height: 2rem;
}
```

7. Ticket lock

```
.ticket-lock {
 display: flex;
 justify-content: flex-end;
 margin-top: 90px;
 margin-right: 5px;
 font-size: 1.5rem;
}
```

    a. Positions the lock icon at the bottom-right corner of the ticket, with a margin at the top to push it down and a smaller margin on the right. The display: flex and justify-content: flex-end properties align the icon to the right. The font size increases the icon's size for better visibility.

HTML so far

```html
<!DOCTYPE html>
<html lang="en">
 <head>
   <meta charset="UTF-8" />
   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
   <title>Document</title>
   <script
     src="https://kit.fontawesome.com/589957875e.js"
     crossorigin="anonymous"
   ></script>
   <link rel="stylesheet" href="style.css" />
 </head>
 <body>
   <div class="toolbox-cont">
     <div class="toolbox-priority-cont">
       <div class="lightpink color"></div>
       <div class="lightgreen color"></div>
       <div class="lightblue color"></div>
       <div class="black color"></div>
     </div>
     <div class="action-btn-cont">
       <div class="add-btn">
         <i class="fa-solid fa-plus"></i>
       </div>
       <div class="remove-btn">
         <i class="fa-solid fa-xmark"></i>
       </div>
     </div>
   </div>
   <div class="main-cont">
     <div class="ticket-cont">
       <div class="ticket-color"></div>
       <div class="ticket-id">12345</div>
       <div class="task-area">Random Task</div>
       <div class="ticket-lock">
         <i class="fa-solid fa-lock"></i>
       </div>
     </div>
   </div>
 </body>
</html>
```

CSS so far

```
* {
 box-sizing: border-box;
}
body {
 margin: 0;
 padding: 0;
}
.toolbox-cont {
 height: 5rem;
 background-color: #4b4b4b;
 display: flex;
 align-items: center;
}
.toolbox-cont > * {
 margin-left: 4rem;
}

.toolbox-priority-cont {
 height: 3.5rem;
 width: 18rem;
 background-color: #3d3d3d;
 display: flex;
 align-items: center;
 justify-content: space-evenly;
}
.toolbox-priority-cont > *:hover {
 background-color: #485460;
}

.color {
 height: 1.5rem;
 width: 3rem;
}
.lightpink {
 background-color: lightpink;
}
.lightgreen {
 background-color: lightgreen;
}
.lightblue {
 background-color: lightblue;
```

```
}
.black {
 background-color: black;
}

.action-btn-cont {
 height: 3.5rem;
 width: 8rem;
 background-color: #3d3d3d;
 display: flex;
}
.action-btn-cont > * {
 display: flex;
 width: 50%;
 font-size: 2rem;
 color: white;
 justify-content: center;
 align-items: center;
}

.add-btn:hover,
.remove-btn:hover {
 background-color: #4bb543;
}

/* ticket css */
.main-cont {
 display: flex;
 gap: 2rem;
 justify-content: center;
 padding: 2rem;
 flex-wrap: wrap;

 /* As many tickets can be added here so this will make sure that the design is neat
to accomodate multiple tickets */
}

.ticket-cont {
 height: 12rem;
 width: 15rem;
 background-color: coral;
}
```

```css
.ticket-color {
 height: 1rem;
}

.ticket-id {
 background-color: yellow;
 height: 2rem;
}

.ticket-lock {
 display: flex;
 justify-content: flex-end;
 margin-top: 90px;
 margin-right: 5px;
 font-size: 1.5rem;
}
```

Designing the Pop-up Modal

The modal component is used for adding or editing tasks in a Kanban board application. It includes a text area for entering the task's content and a side panel for selecting the task's priority through color codes.

```html
<div class="modal-cont"></div>
```

1. This div acts as the container for the entire modal. It's designed to provide a focused area for inputting task details and selecting a priority.
2. Text area for task input

```html
<div class="modal-cont">
     <textarea class="textArea-cont" placeholder="Enter Your
Task"></textarea>
```

```
  </div>
```

   a. A textarea element where users can type the task they wish to add or edit. The placeholder attribute provides a hint about what to enter.

3. Priority colors container

```
<div class="modal-cont">
    <textarea class="textArea-cont" placeholder="Enter Your Task"></textarea>
    <div class="priority-colors-container"></div>
  </div>
```

   a. This div houses the priority color options. It's a distinct section within the modal, allowing users to choose a priority for their task visually.

4. Priority color options

```
<div class="modal-cont">
    <textarea class="textArea-cont" placeholder="Enter Your Task"></textarea>
    <div class="priority-colors-container">
      <div class="lightpink priority-color"></div>
      <div class="lightgreen priority-color"></div>
      <div class="lightblue priority-color"></div>
      <div class="black priority-color active"></div>
    </div>
  </div>
```

   a. Individual div elements represent different priority levels through color (lightpink, lightgreen, lightblue, black).
   b. The active class on the last element indicates the current selection or default value.

5. CSS
    a. Modal container

```css
.modal-cont {
 height: 50vh;
 width: 45vw;
 display: flex;
 background-color: lightsalmon;
 position: absolute;
 top: 30%;
 left: 27%;
}
```

    b. Styles the modal container with a fixed size relative to the viewport (vh and vw units), centers it on the screen using position: absolute and offsets from the top and left.
    c. The display: flex property indicates its children (the text area and priority colors container) will be laid out in a row.
6. Text area

```css
.textArea-cont {
 height: 100%;
 width: 75%;
 resize: none;
 outline: none;
 border: none;
 background-color: #dfe4ea;
 font-size: 2rem;
 color: black;
}
```

    a. Configures the task input area to take up the majority of the modal's space, removing the default border and resize handle for a cleaner look. The large font size improves readability.

7. Priority Colors Container

```
.priority-colors-container {
 height: 100%;
 width: 25%;
 display: flex;
 flex-direction: column;
 background-color: #4b4b4b;
 align-items: center;
 justify-content: space-around;
}
```

a. Sets up a sidebar for priority color selection, occupying a smaller portion of the modal. It's designed to display color options in a column, centered and evenly spaced.

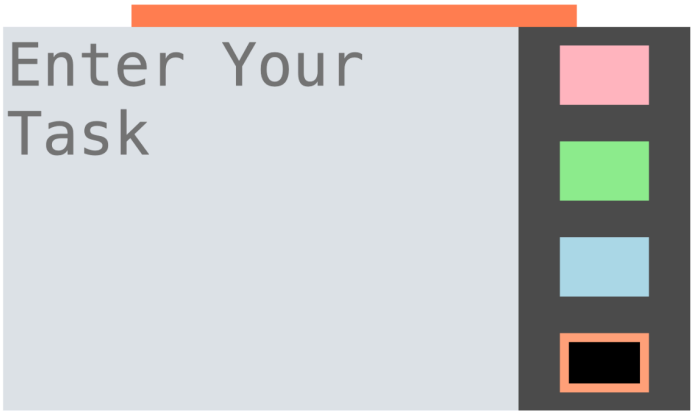
8. Priority color options

```
.priority-color {
 height: 2rem;
 width: 3rem;
}
```

9. Active color

```
.active {
 border: 5px solid lightsalmon;
}
```

o/p achieved so far


Enter Your
Task