Jira Board Clone

Intended For Beginners

CodeSandbox Link :

jira-clone-aj-aug

https://codesandbox.io/p/sandbox/jira-clone-aj-aug-5zxvrh

We'll create a simple Jira Board using HTML, CSS, and JavaScript. The board will allow you to add tasks, move them between different columns (Planned, In-Progress, and Completed), and auto-delete them if they remain empty.

1. HTML Structure

Let's start by setting up the basic structure of our application with HTML.

```
<!DOCTYPE html> <html> <head> <title>Jira Board</title> <meta charset="UT
F-8" /> </head> <body> <div id="app"></div> <br /> <br /> <button id="add
Task">Add New Task</button> <br /> <br /> <div class="jira"> <div id="pla
nned" class="column"> <br /> <h3>Planned</h3> </div> <div id="inProgress"
class="column"> <br /> <h3>In-Progress</h3> </div> <div id="completed" cl
ass="column"> <br /> <h3>Completed</h3> </div> </div> <script src="./inde
x.js" type="module"></script> </body> </html>
```

- **Explanation:**
  - We have three main sections represented as columns: `Planned`, `In-Progress`, and `Completed`.
  - The `app` div will hold the title, which can be edited.
  - The "Add New Task" button will allow us to add tasks to the "Planned" column.

2. CSS Styling

Next, let's style the board using CSS.

```
* { padding: 0; margin: 0; box-sizing: border-box; } .jira { display: fle
x; justify-content: space-around; margin-left: 6px; } body { font-family:
sans-serif; background: linear-gradient(to bottom, #ff7757, #fefb4b); } h
1 { text-align: center; border: 2px solid rgb(252, 250, 250); padding: 6p
x; } h3 { border-bottom: solid grey; padding-bottom: 6px; } button { back
ground-color: red; color: whitesmoke; border: none; border-radius: 6px; p
adding: 12px 24px; font-size: 12px; cursor: pointer; transition: all 0.4s
ease; } button:hover { background-color: #ec1a4f; box-shadow: 0 6px 8px r
gba(0, 0, 0, 0.2); transform: translateY(-2px); } button:active { backgro
und-color: #bb1aec; box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1); transform:
translateY(1px); } .taskCard { background-color: bisque; color: tomato; b
order: 2px solid greenyellow; margin: 6px; padding: 6px; } .column { widt
h: 30%; border: 2px solid grey; margin-right: 6px; min-height: 400px; tex
t-align: center; }
```

- Explanation:
  - The `.jira` class is used to style the board's container, ensuring the columns are evenly spaced.
  - The `.taskCard` class styles each task card with a background color, border, and padding.
  - The `.column` class styles each column with a fixed width, border, and minimum height.

3. JavaScript Functionality

Finally, we'll add the interactivity using JavaScript.

```
import "./styles.css"; const heading = document.getElementById("app"); he
ading.innerHTML = `<h1>!Jira Board!</h1>`; heading.setAttribute("contente
ditable", true); const DEFAULT_TASK_CARD_MESSAGE = "Write Your Task Her
e"; const BLANK_TEXT = ""; let taskButton = document.getElementById("addT
ask"); let plannedColumn = document.getElementById("planned"); let count
= 0; taskButton.addEventListener("click", () => { let taskCard = documen
t.createElement("div"); taskCard.setAttribute("class", "taskCard"); taskC
ard.setAttribute("id", `taskCard-${++count}`); taskCard.setAttribute("con
tenteditable", true); taskCard.setAttribute("draggable", true); taskCard.
innerHTML = DEFAULT_TASK_CARD_MESSAGE; plannedColumn.appendChild(taskCar
d); // When we click, the existing text should get removed taskCard.addEv
entListener("click", (event) => { if (taskCard.innerText === DEFAULT_TASK
_CARD_MESSAGE) { taskCard.innerHTML = BLANK_TEXT; } }); // Removes the Ca
rd which was created but didn't have any value entered taskCard.addEventL
istener("blur", (event) => { if (taskCard.innerHTML === BLANK_TEXT) { tas
kCard.remove(); } }); taskCard.addEventListener("dragstart", (dragEvent)
=> { let selectedCardId = dragEvent.target.id; dragEvent.dataTransfer.set
Data("text", selectedCardId); taskCard.style.opacity = 0.8; }); taskCard.
addEventListener("dragend", (dragEvent) => { taskCard.style.opacity = 1;
}); let dragEvents = ["dragover", "dragenter", "drop"]; dragEvents.forEac
h((value) => { let cols = document.getElementsByClassName("column"); for
(let col of cols) { col.addEventListener(value, (event) => { event.preven
tDefault(); if (value === "drop") { let selectedCardId = event.dataTransf
er.getData("text"); let selectedCard = document.getElementById(selectedCa
rdId); col.append(selectedCard); } }); } }); });
```

- **Explanation:**
  - The code begins by setting up the title as editable content using `contenteditable`.