### **Key Features and Functionality:**

1. **Marking Past Days**:
   * Days that are in the past are automatically marked with a class called .past-day, which can be styled (e.g., grayed out) to indicate that they have already passed.
2. **Highlighting the Current Day**:
   * The current day is identified by comparing it to today's date and is highlighted using the .current-day class (e.g., with a blue background).
3. **Task Status Colors**:
   * **Incomplete tasks**: Days with tasks that are not yet completed are marked with the class .incomplete-tasks (e.g., with a red background).
   * **All tasks completed**: Days where all tasks have been completed are marked with the class .completed-tasks (e.g., with a green background).
4. **Adding Tasks**:
   * Users can add tasks through a modal form. Each task has a title and a completed status (initially set to false).
   * Tasks are associated with specific dates, and when added, the calendar reloads to reflect them.
5. **Marking Tasks as Complete**:
   * When a day is clicked, a popup appears with the list of tasks for that day.
   * Each task in the popup has a button to mark it as complete or incomplete. When a task's status changes, the calendar updates dynamically to reflect whether the day is marked as completed or incomplete based on the tasks.
6. **Navigation**:
   * You can move between months using the "Previous" and "Next" buttons. The calendar will update accordingly, and past days, the current day, and tasks are all recalculated for the new month.

### **How it Works:**

* **Task Storage**: Tasks are stored in a JavaScript object, where each key is a date, and each value is an array of tasks associated with that date.
* **Dynamic Rendering**: The calendar is rendered dynamically, where each day is generated, and tasks are displayed within the corresponding day cells. The colors (past, current, incomplete, or completed) are applied based on the conditions at that moment.