# Task management system to manage their college assignments and projects.

## Key Features:

**1. Event Creation (Admin Feature)**

* **Event Creation Form (Local Storage):**
* Admins will create events using a form that collects:
  + **Event Title**, **Description**, **Date**, **Time**, **Location**, and **Registration Limit**.
* This information will be stored in **local storage** to simulate a database.
* **Drag-and-Drop Gantt Chart (Optional):**
* We’ll use a simple front-end drag-and-drop library (such as dragula.js) to simulate a GANTT-style calendar view where admins can visually adjust event timings. If you can implement it very good, just google it and you should be able to do it.

**2. User Registration for Events (Student Feature)**

* **Event Registration Form:**
* Students can search for events using filters (date, event type).
* Registration will be simulated using a button to reserve a seat. The number of available seats for each event will decrease with each registration, and when full, the registration button becomes disabled.
* Registrations and remaining seat counts will be stored in **local storage**.

**3. Payment Integration Simulation for Paid Events**

* **Simulated Payment Process:**
* For paid events, a **"Pay Now"** button will simulate the payment process. Clicking the button will trigger a JavaScript function that simulates successful payment and updates the local storage to reflect the student’s registration.
* No real payment gateway will be involved. Just show a payment success message.

**4. Post-Event Feedback (Feedback and Rating Feature)**

* **Feedback Form:**
* After attending an event, students can provide feedback through a simple form.
* Feedback will include a rating system (e.g., 1-5 stars) and an optional comments section.
* Feedback is stored in **local storage** and displayed on the event's page.
* Average ratings can be calculated and displayed in real-time using JavaScript.

**5. Event Calendar (Centralized Calendar with Filters)**

* **Client-Side Calendar View:**
* We will implement a front-end-only calendar using a JavaScript calendar library (such as **FullCalendar.js**).
* Events will be dynamically displayed based on data stored in **local storage**.
* Students can filter events on the calendar by date and type.

You can use your own creativity to add more features too, above are the minimum features that you need to add.

* To store information, use Local Storage.
* Please ensure that it should have good UI.