Case Study: Event Management App using Angular

Topics covered:

- Angular Components
- Custom Pipes
- Custom Directives

User Story

You are developing an **Event Management Portal** for a company that organizes conferences, workshops, and music concerts.
The portal should display upcoming events, format ticket prices, and visually highlight premium events.

Requirements

1. **Event List Component** (EventListComponent)
    - Displays a list of events with details: Name, Date, Ticket Price, and Category (Conference, Workshop, Concert).
    - Should use *ngFor to iterate over event data.
2. **Custom Pipe** (PriceFormatPipe)
    - Formats ticket prices into a standard format:
      Example: 500 → ₹500.00
      Example: 1200 → ₹1,200.00
3. **Custom Directive** (HighlightDirective)
    - Highlights premium events (ticket price above ₹2000) by changing the background color to light gold.

Sample Data

```
events = [
  { name: 'Tech Innovators Conference', date: '2025-09-12', price: 3500, category: 'Conference' },
  { name: 'Creative Writing Workshop', date: '2025-10-05', price: 800, category: 'Workshop' },
  { name: 'Rock Music Concert', date: '2025-11-20', price: 2500, category: 'Concert' },
  { name: 'AI & Machine Learning Summit', date: '2025-12-02', price: 5000, category: 'Conference' }
];
```

Demo Flow

1. **Event List Component** fetches and displays event details.
2. **PriceFormatPipe** formats ticket prices.
3. **HighlightDirective** automatically applies a background style for premium events.

Example Output (UI)

Event Name	Date	Ticket Price	Category
Tech Innovators Conference	12-Sep-2025	₹3,500.00	Conference
Creative Writing Workshop	05-Oct-2025	₹800.00	Workshop

Rock Music Concert	20-Nov-2025	₹2,500.00	Concert
AI & Machine Learning Summit	02-Dec-2025	₹5,000.00	Conference

---

Submission Guidelines

1. **Code Structure & Organization**
   - All components, directives, and pipes should be placed in appropriately named folders (components, directives, pipes, etc.).
   - Use **meaningful file names** following Angular's naming conventions (e.g., product-list.component.ts, highlight.directive.ts).
2. **Coding Standards**
   - Follow **Angular Style Guide** for naming conventions, indentation, and folder structure.
   - Ensure **consistent formatting** using tools like Prettier or Angular CLI's ng lint.
   - Use **TypeScript features** effectively (e.g., strong typing, interfaces).
3. **Functionality Requirements**
   - All features described in the case study **must be implemented and functional**.
   - Ensure **data binding**, **custom pipes**, and **custom directives** work as intended.
   - Add at least **one animation** for improved user experience.

4. **Domain Adaptation**
    - Replace the original domain content with the new domain scenario given in the case study.
    - Maintain **realistic sample data** relevant to the domain.
5. **Testing & Verification**
    - Run the application locally using ng serve and ensure **no compilation errors or warnings**.
    - Verify that all **pipes, directives, and components** display expected outputs.
6. **Submission Format**
    - Submit the **entire Angular project folder** as a **.zip** file (excluding node_modules).
    - Include a **README.md** with:
        - Project title & description
        - Installation steps (npm install, ng serve)
        - Brief explanation of implemented features
    - Include **screenshots** of key functionality (e.g., list display, formatting, highlighting).
7. **Deadline & Late Submission Policy**
    - Submit by the given deadline.
    - Late submissions will incur a penalty unless approved with valid reasons.
8. **Evaluation Criteria**
    - **Functionality:** 40%
    - **Code quality & structure:** 30%
    - **UI/UX & animations:** 20%
    - **Documentation & submission compliance:** 10%