Assignment: Contract Management Platform

Objective

Build a frontend-based Contract Management Platform from scratch that demonstrates product thinking, UI design, state management, and clean code architecture.

No UI designs or backend will be provided. You must design and implement everything yourself.

Deadline

Submission by **Sunday, 11:59 PM IST**
Late submissions will not be evaluated.

Deliverables

1. Public GitHub repository
2. Working frontend application
3. `README.md` including:
    - Setup instructions
    - Architecture and design decisions
    - Assumptions and limitations

Functional Requirements

1. Blueprint Creation

A Blueprint represents a reusable contract template.

- Create a blueprint with configurable fields
- Supported field types: Text, Date, Signature, Checkbox
- Ability to place fields on a page (basic positioning is sufficient)
- Store field metadata:
    - Type
    - Label
    - Position
- Blueprint data may be stored locally or via mocked persistence

2. Contract Creation from Blueprint

- Select an existing blueprint
- Generate a contract from it
- Contract should inherit all fields
- Allow users to fill values for fields

3. Contract Lifecycle

Each contract must follow the lifecycle below:

Created → Approved → Sent → Signed → Locked
Revoked (can occur after creation or sending)

Rules:

- State transitions must be controlled (no skipping steps)
- UI should clearly display current status and available actions
- Locked contracts cannot be edited
- Revoked contracts cannot proceed further

4. Contract Listing Dashboard

Display contracts in a table view, grouped or filterable by status:

- Active
- Pending
- Signed

Table should include:

- Contract name
- Blueprint name
- Status
- Created date
- Action buttons (view, change status)

UI Guidelines

- No design assets will be provided
- UI must be created by the candidate
- Focus on clarity, usability, and logical flow
- Visual polish is secondary to structure and usability

Tech Stack

You may choose your own stack. Justify your choices in the README.

Preferred (not mandatory):

- React or Next.js
- TypeScript
- Component-based architecture
- Proper state management
- Clean folder structure

Backend is not required. Mock data is acceptable.

Evaluation Criteria

- Code structure and architecture
- Contract lifecycle handling
- UI clarity and UX flow
- State management approach
- Code readability and consistency
- GitHub commit quality and documentation

Restrictions

- No copied code without understanding
- No monolithic or poorly structured files
- Missing README will lead to rejection
- Broken or incomplete flows will not be evaluated

Optional Enhancements

- Drag-and-drop field placement
- Status indicators or timeline
- Reusable component library
- Basic unit tests

Submission

Share the public GitHub repository link at: https://forms.gle/U8gZW7bKiWbFd6RU9