The **Leitner System** is a powerful spaced repetition technique, often used for flashcards. Your task is to incorporate it into the **MERN stack task** by making a **Flashcard Learning App** instead of a generic task manager.

📌 Task: Build a Flashcard Learning App with Leitner System

Objective: Create a web app where users can **create, review, and progress through flashcards** using the **Leitner System**.

📜 Requirements:

✅ Backend (Node.js, Express, MongoDB, Mongoose)

- Create an API with the following endpoints:
  - `POST /flashcards` → Add a new flashcard
  - `GET /flashcards` → Get all flashcards
  - `PUT /flashcards/:id` → Update a flashcard (move to the next level if answered correctly)
  - `DELETE /flashcards/:id` → Delete a flashcard
- Implement the **Leitner System logic**:
  - Flashcards start in **Box 1**.
  - If answered correctly, they move to the next box.
  - If answered incorrectly, they go back to Box 1.
  - Higher boxes have **longer review intervals**.
- Store flashcard **level (box number), question, answer, and next review date** in MongoDB.

✅ Frontend (React, React Hooks, Axios, Tailwind/Bootstrap)

- Display flashcards with options:
  - **"Show Answer"** button
  - **"Got it right"** and **"Got it wrong"** buttons
- Update the flashcard level based on the user's response.
- Fetch flashcards based on their **next review date** (implement spaced repetition logic).
- Show progress (e.g., “You have 5 flashcards due today”).
- **Simple & clean UI** with minimal distractions.

✅ Bonus (Optional, for extra points)

- 🔹 **Login System (JWT Auth)** – Let users save their progress
- 🔹 **Dark Mode Toggle** – Better UX for late-night study sessions
- 🔹 **Animations (Framer Motion)** – Smooth transitions when answering flashcards
- 🔹 **Deploy on Vercel/Render** – Bonus points for making it live

---

⏳ Submission

- Push code to **GitHub with a README** explaining setup & thought process.

📊 Evaluation Criteria:

✔️ **Code Quality & Best Practices**
✔️ **Leitner System Implementation**
✔️ **UI/UX Simplicity & Usability**
✔️ **Proper API Integration & State Management**
✔️ **Bonus Features (if implemented)**