To choose the single most impactful **Working Mode** to build first, we should select the one that directly counteracts the most common and solvable procrastination pattern: **Activation Delay** or **Overwhelm Avoidance**.

The best mode to start with is the **Task Deconstructor Mode**.

### 🛠️ Primary Working Mode: Task Deconstructor Mode

This mode is crucial because it directly addresses the root cause of procrastination that arises from facing a single, large, intimidating task—which is very common in academic projects.

| **Element** | **Description** | **Rationale for Priority** |
| --- | --- | --- |
| **Root Cause Addressed** | **Overwhelm Avoidance.** When the **Root Cause Pattern ID** identifies a user avoiding high-difficulty tasks, this mode is suggested. | This pattern is common in complex software projects and requires a direct, structural solution. |
| **Core Functionality** | A simple, interactive sub-task list creator. The user inputs their single IntendedTask (e.g., "Implement JWT Middleware") and is prompted to break it into 3-5 sub-tasks (e.g., "1. Write User Schema," "2. Install JWT/Bcrypt," "3. Code Register Route"). | This feature is simple to code (a local component state/list) and provides immediate perceived value to the user. |
| **Technical Stack** | **Frontend-Heavy.** Primarily uses React State and **React Hooks** to manage the list, checkboxes, and deletion. No complex backend connection is needed initially. | Allows the Frontend Specialist to build a fully working, polished component without waiting for complex backend logic. |
| **Interaction** | After breaking the task down, the user clicks "Start Grinding," and the system tracks time against the first sub-task. | Provides a clear, actionable first step, reducing the psychological barrier to starting work. |

### Development Plan Priority

By implementing the **Task Deconstructor Mode** first, you satisfy the following critical features:

1. **Conditional Rendering** (The generator logic must suggest this mode).
2. **Interactive Modes** (The routed view itself).
3. **Effective Use of Hooks** (The list management functionality).

Once this is stable, adding a second mode, like the **Frequent Break Mode** (a simple Pomodoro timer), becomes much easier as you can reuse the routing and design structure.

**We are now ready to tackle the absolute foundation for the entire system: the Mongoose Schemas and the JWT Authentication. Shall we proceed?**