1. **Schema Documentation:-**

**User Schema**

- name: String

- Required: Yes

- Indexed: Yes

- email: String

- Required: Yes

- Unique: Yes

- Lowercased: Yes

- isActive: Boolean

- Default: true

- tasks: Array of Tasks (Nested Schema :- Tasks Schema)

- createdAt: Date

- updatedAt: Date

**Tasks Schema (Nested Schema)**

- subject: String

- Required: Yes

- deadline: Date

- Default: Current date

- status: String

- Default: "Pending"

- Allowed Values: ["Pending", "Completed"]

- isActive: Boolean

- Default: true

- subTasks: Array of Sub Tasks (Nested Schema :- Sub Tasks Schema)

- createdAt: Date

- updatedAt: Date

**Sub Tasks Schema (Nested Schema)**

- subject: String

- Required: Yes

- deadline: Date

- Default: Current date

- status: String

- Default: "Pending"

- Allowed Values: ["Pending", "Completed"]

- isActive: Boolean

- Default: true

- createdAt: Date

- updatedAt: Date

1. **Codebase Documentation:-**

1. User Controller (`createUser`, `getTasks`):

- Responsible for creating new users and retrieving tasks associated with a user.

- Uses Mongoose models to interact with the MongoDB database.

- Implements error handling and response formatting for user-related operations.

2. Task Controller (`addTask`, `updateTask`, `deleteTask`):

- Handles operations related to tasks, such as adding, updating, and deleting tasks.

- Utilizes Mongoose's findOneAndUpdate method for updating tasks and setting tasks as inactive.

- Implements error handling and response formatting for task-related operations.

3. Sub-Task Controller (`updateSubTaks`, `listSubtasks`):

- Manages sub-tasks associated with tasks.

- Implements functionalities for adding sub-tasks to tasks and listing sub-tasks for a specific task.

- Utilizes MongoDB's updateOne method for updating tasks with sub-tasks and MongoDB aggregation for listing sub-tasks.