**Project Description: Family Task Scheduler**

The **Family Task Scheduler** is a Python-based application using MongoDB to help multiple families organize and schedule their shared and individual tasks efficiently. It supports managing events like birthdays, medical appointments, recreational outings, and major shopping trips for various families while ensuring privacy and tailored scheduling for each. The system emphasizes scalability, user-friendliness, and collaborative task management.

**Features**

1. **Family Management**:
   * Register and manage multiple families independently.
   * Assign unique identifiers for each family to maintain data separation.
2. **Member Management**:
   * Add family members with roles (adults, children, etc.) specific to each family.
   * Update and manage profiles for members within their family group.
3. **Task Scheduling**:
   * Create tasks scoped to specific family members, groups within a family, or the entire family.
   * Set recurring tasks (e.g., annual birthdays, weekly outings).
   * Assign task priority levels (low, medium, high).
4. **Notifications and Alerts**:
   * Notify family members of their upcoming or overdue tasks.
   * Send updates when tasks are modified or completed.
5. **Calendar View**:
   * Provide a personalized calendar view for each family showing their tasks.
   * Offer weekly, monthly, and yearly perspectives.
6. **Collaboration and Notes**:
   * Allow family members to collaborate on tasks and mark them as completed.
   * Add shared notes or attachments to tasks (e.g., shopping lists).
7. **Data Security and Isolation**:
   * Use MongoDB to store data securely, ensuring families' data is isolated.
   * Enable role-based access within families to ensure privacy.
8. **Synchronization and Backup**:
   * Synchronize tasks across multiple devices.
   * Regularly back up data for all families.

**Data Structure**

**User**

{

"id": "unique\_user\_id",

"username": "string",

"password": "string",

"admin": “boolean”,

"memberId": "unique\_member\_id"

}

**Family**

{

"id": "unique\_family\_id",

"name": "string",

"members": ["member\_id1", "member\_id2"],

"tasks": ["task\_id1", "task\_id2"]

}

**Family Member**

{

"id": "unique\_member\_id",

"name": "string",

"role": "adult | child",

"email": "string",

"phone": "string",

"familyId": "unique\_family\_id",

"tasks": ["task\_id1", "task\_id2"]

}

**Task**

{

"id": "unique\_task\_id",

"title": "string",

"description": "string",

"assignedTo": ["member\_id1", "member\_id2"],

"familyId": "unique\_family\_id",

"date": "ISODate",

"priority": "low | medium | high",

"recurring": {

"frequency": "daily | weekly | monthly | yearly",

"endDate": "ISODate"

},

"status": "pending | completed",

"notes": "string",

"attachments": ["file1", "file2"]

}

**Notification**

{

"id": "unique\_notification\_id",

"taskId": "unique\_task\_id",

"recipient": "unique\_member\_id",

"familyId": "unique\_family\_id",

"sentAt": "ISODate",

"read": "boolean"

}

**API Design**

**Family Endpoints**

* **POST /family**
  + Register a new family.
* **GET /families**
  + Retrieve a list of all families.
* **GET /family/{id}**
  + Retrieve details of a specific family.
* **PUT /family/{id}**
  + Update family details.
* **DELETE /family/{id}**
  + Remove a family and its associated data.

**Member Endpoints**

* **POST /family/{familyId}/member**
  + Add a new member to a family.
* **GET /family/{familyId}/members**
  + Retrieve all members of a family.
* **PUT /member/{id}**
  + Update member details.
* **DELETE /member/{id}**
  + Remove a family member.

**Task Endpoints**

* **POST /family/{familyId}/task**
  + Create a new task for a family.
* **GET /family/{familyId}/tasks**
  + Retrieve tasks for a family, with optional filters (e.g., date range, priority).
* **PUT /task/{id}**
  + Update task details.
* **DELETE /task/{id}**
  + Remove a task.

**Notification Endpoints**

* **GET /family/{familyId}/notifications/{memberId}**
  + Retrieve notifications for a specific family member.
* **PATCH /notification/{id}**
  + Mark a notification as read.

**Utility Endpoints**

* **GET /family/{familyId}/calendar-view**
  + Fetch tasks for a family in a calendar format.