Task Scheduling System

We propose to develop a task scheduling system for a project management company. The system will manage tasks within a 5-day workweek, automatically generating unique IDs for tasks for better identification. The goal is to prioritize and optimize task completion to maximize total revenue while respecting deadlines and the constraint that only one task can be completed per day.

Develop a RESTful API using Express.js to manage task records, including titles, and provide the optimal task schedule and total revenue.

System Requirements

1. **Task Attributes**: Each task in the system will have the following details:
   - `id` (integer): An auto-incremented unique identifier for the task.
   - `title` (string): A descriptive title of the task.
   - `deadline` (integer): The number of days within which the task must be completed, considering a 5-day workweek.
   - `revenue` (integer): The revenue earned upon completing the task.
2. **Express.js API Endpoints**:
   - **POST /tasks**: Create a new task record. The request body should include `title`, `deadline`, and `revenue`. The response should confirm the creation of the task with an auto-generated `id`.
   - **GET /tasks**: Retrieve a list of all tasks.
   - **GET /schedule**: Retrieve the optimal schedule considering a 5-day workweek and the single-task-per-day constraint. Return the list of tasks in the order they should be completed and the total revenue.

Constraints

1. **Auto-Increment IDs**: Ensure that each task receives a unique `id` that is auto-incremented.
2. **5-Day Workweek**: The system will schedule tasks only within a 5-day workweek. Each day can accommodate only one task.
3. **Single Task Per Day**: Only one task can be completed each day.
4. **In-Memory Data**: The system will store all data in-memory with no persistence between server restarts.
5. **Deadline Compliance**: The scheduling must respect deadlines and avoid conflicts by ensuring each task is scheduled within its deadline.

Example Workflow

1. **Add a New Task**:

POST request to `/tasks` with the body:

```
{ "title": "Develop User Authentication Module", "deadline": 3,
"revenue": 10000 }
```

    ○

Response:

```
{ "id": 1, "title": "Develop User Authentication Module", "deadline":
3, "revenue": 10000 }
```

    ○

2. **Retrieve All Tasks**:
    - **GET** request to `/tasks`.

Response:

```
[
  { "id": 1, "title": "Develop User Authentication Module",
"deadline": 3, "revenue": 10000 },
  { "id": 2, "title": "Implement Payment Gateway", "deadline": 2,
"revenue": 9000 },
  { "id": 3, "title": "Design Data Analytics Dashboard", "deadline":
4, "revenue": 8000 },
  { "id": 4, "title": "Create API Documentation", "deadline": 5,
"revenue": 12000 },
  { "id": 5, "title": "Optimize Database Queries", "deadline": 2,
"revenue": 6000 },
  { "id": 6, "title": "Setup Continuous Integration", "deadline": 3,
"revenue": 7000 },
  { "id": 7, "title": "Conduct Security Audit", "deadline": 5,
"revenue": 11000 }
]
```

    ○

3. **Retrieve Optimal Task Schedule**:

    - **GET** request to `/schedule`.

Response:

```
{
  "schedule": [
    { "id": 2, "title": "Implement Payment Gateway", "deadline": 2,
"revenue": 9000 },
    { "id": 5, "title": "Optimize Database Queries", "deadline": 2,
"revenue": 6000 },
    { "id": 6, "title": "Setup Continuous Integration", "deadline": 3,
"revenue": 7000 },
    { "id": 1, "title": "Develop User Authentication Module",
"deadline": 3, "revenue": 10000 },
    { "id": 4, "title": "Create API Documentation", "deadline": 5,
"revenue": 12000 }
  ],
  "totalRevenue": 38000
}
```

    -
    - **Explanation**:
        - **Tasks Scheduled**: 5 tasks were scheduled within the 5-day workweek.
        - **Total Revenue**: ₹38,000
        - **Note**: Not all tasks could be scheduled due to the constraint of only having 5 days in the workweek and one task per day.