**Backend Task**

**Functional Requirements:**

**1. Core Entities:**

**Users:** Manage users of the system (Admin, Manager, Member).

**Projects:** Each project can have multiple tasks and assigned members.

**Tasks:** Tasks have different statuses and must be completed within due dates.

**1. API Endpoint: Get Task Completion Summary**

Description: Fetch the count of tasks by status (To Do, In Progress, Completed) for all tasks in the system.

**2. API Endpoint: Get User Performance Report**

Description: Get the total number of tasks assigned to a user, grouped by their status (To Do, In Progress, Completed).

**3. API Endpoint: Get Overdue Tasks Summary**

Description: Fetch the total number of tasks that are overdue, grouped by project.

**4. API Endpoint: Get Project Task Summary with Members' Contribution**

Description: Get the total number of tasks in a project, grouped by status, along with how many tasks each member has completed.

**Pagination and Filtering:**

Add pagination to the analytics APIs to handle large datasets efficiently

**Cache Layer:**

Implement Redis caching for analytics results to improve performance.

**Evaluation Criteria:**

1. **Advanced MongoDB Aggregation:**

**2)** Ability to design complex aggregation pipelines with $lookup, $facet, $group, and $project.

**3)** **Performance Optimization.**

**4)** **Efficient querying and pagination strategies.**

**5) API Design & Code Quality:**

**6) Modular design using NestJS modules, controllers, and services.**

**7)** **Security & Caching:**

**8)** Use of Redis for caching analytics results and JWT for protecting routes.

**Expected Deliverables:**

1) Complete NestJS project code hosted on a GitHub repository.

2) Postman collection for testing the APIs.

3) Instructions on how to run the project locally