**Project Documentation**

**Overview**

This project is a task management system that allows users to create and manage projects and tasks. Users can register, create projects, and assign tasks to those projects, providing a straightforward interface for managing their work.

Logical Design of the Database

**Entities**

1. User: Represents individuals who can register, create projects, and manage tasks.
2. Project: Represents a collection of tasks associated with a specific user.
3. Task: Represents an individual task that belongs to a project.

**Table and Entity Relation Design**

* **User Table:**
  + Fields: username(unique), email(unique), password
* **Project Table:**
  + Fields: user\_id (FK),name(unique),, description
* **Task Table:**
  + Fields: project\_name (FK), username (FK), title, description, status ,name (unique)

**The relationships are as follows:**

* A User can have multiple Projects.
* A Project can have multiple Tasks.

**API Descriptions**

* User Registration: Creates a new user and ensures no duplicate usernames or emails exist.
* Create Project: Creates a new project if the user exists; checks for duplicate project names.
* Get User Projects: Retrieves projects linked to a specific user.
* Delete Project: Deletes a project associated with a user if it exists.
* Create Task: Creates a task for a specific project if the project exists; checks for duplicate task names.
* Delete Task: Deletes a task only if it exists for the specified user and project.
* Get Tasks by User and Project: Retrieves tasks for a specific user and project

**Note :-please goto the db.js in config file and set the mongo server url perfectly to make the pi working and also check the port of api before running it**

**API Documentation**

**User Registration**

* **Method: POST**
* **URL: /api/users/register**
* **Body: { "username": "testuser", "email": "testuser@example.com", "password": "password123" }**

**Create Project**

* **Method: POST**
* **URL: /api/projects**
* **Body: { "username": "testuser", "name": "Test Project", "description": "This is a test project" }**

**Get User Projects**

* **Method: GET**
* **URL: /api/projects/**
* **Body: { "username": "testuser"}**

**Delete Project**

* **Method: DELETE**
* **URL: /api/projects/**
* **Body: { "username": "testuser", "projectName": "Test Project",**

**Create Task**

* **Method: POST**
* **URL: /api/tasks**
* **Body: { "project\_name": "Test Project", "username": "testuser", "title": "Test Task", "description": "This is a test task", "task\_name": "Test Task", }**

**Delete Task**

* **Method: DELETE**
* **URL: /api/tasks**
* **Body: { "project\_name": "Test Project", "username": "testuser”,"task\_name": "Test Task", }**

**Get Tasks by User and Project**

* **Method: get**
* **URL: /api/tasks**
* **Body: { "username": "testuser", "project\_name": "Test Project" }**

**Conclusion**

This project offers a simple API for managing users, projects, and tasks. Key features include:

* User Management: Ensures unique usernames and emails.
* Project Management: Allows creation and management of projects.
* Task Management: Enables adding, updating, and deleting tasks.

Overall, it provides a clear and efficient solution for project management needs.