**Assignment Management System**

The purpose of the project is to create a micro system for that can handle the assignment management system in which student can access and submit the assignment for the enrolled course and faculty can create assignment for the course and assign them points according to their submission. Even student gets an instant update about assignment on email at time of assignment creation by faculty.

**Table of Contents**

* [Project Description](https://chat.openai.com/#project-description)
* [Technologies Used](https://chat.openai.com/#technologies-used)
* [Features](https://chat.openai.com/#features)
* [Getting Started](https://chat.openai.com/#getting-started)
  + [Prerequisites](https://chat.openai.com/#prerequisites)
  + [Installation](https://chat.openai.com/#installation)
* [Usage](https://chat.openai.com/#usage)
* [API Endpoints](https://chat.openai.com/#api-endpoints)
* [Database Schema](https://chat.openai.com/#database-schema)
* Project Deployment Link

**Project Description**

The Project consists of mainly two routes:

1. Student Routes
2. Faculty Routes

The student routes and faculty routes are different and are authenticated by their respective middleware. Even JWT authentication is included which authenticates that route during the different request and responses. The Project consists of 6 tables Student, Faculty, Course, Assignment, Enrollment, Assignment Submission. These tables are interrelated to each other which help to flow the data as per the requirement of the project. Deployment of Project is done on render. For the deployment of the project is used cloud MySQL server.

**Technologies Used**

List the technologies and frameworks you used in your project. For example:

* MySQL
* Express.js
* Multer
* Node.js
* JsonWebToken
* Redis
* Sequelize
* Nodemailer
* Bcryptjs

**Features**

Highlight the key features of your project. For example:

* User authentication and authorization
* File upload and storage using Multer
* RESTful API for CRUD operations on specific resources
* Database integration with MySQL
* Implementation of Redis for faster access of the data

**Getting Started**

Explain how to set up and run your project locally. Include steps for prerequisites and installation.

**Prerequisites**

List any software or tools that need to be installed before running the project. For example:

* Node.js and npm
* MySQL database

**Installation**

Provide step-by-step instructions on how to install and configure your project. Include any necessary commands.

 Install dependencies: npm install

 Configure your MySQL database settings and SMTP credentials for Nodemailer in a .env file:

**NOTE**: No need to create Tables or Schema in MySQL as Sequelize will create the table on npm start

PORT=8000

JWT\_SECRET = YOUR\_SECRET

ROOT = MYSQL\_ROOT

PASSWORD = MYSQL\_PASSWORD

HOST = MYSQL\_HOST

DATABASENAME = MYSQL\_SCHEMA\_NAME

NODEMAILERHOST = SMTP\_HOST\_NAME

NODEMAILERUSER = SMTP\_USER\_NAME

NODEMAILERPASS = SMTP\_USER\_PASSWORD

 Start the application: npm start

**Usage**

Explain how to use your application. Provide examples of common use cases. For example:

1. Register a new student.
2. Register a new faculty.
3. Login to student and faculty.
4. Creation of course by registered faculty.
5. Enrollment of student in the course.
6. Creation of assignment for the course created by the faculty and sending alert email to the enrolled students.
7. Submission of assignment by student.
8. Grading of assignment by faculty

**API Endpoints**

Document for the API Endpoints is available in the folder refer the file “Classroom.postman\_collection.json”.

**Database Schema**

**NOTE**: No need to create Tables or Schema in MySQL as Sequelize will create the table on npm

**Deployment Link**

<https://classroom-assignment.onrender.com>