**API Endpoints Documentation**

**Overview**

This document provides detailed information on the API endpoints for the assignment management system. The API is built using Node.js and Express.js, with MySQL as the database and JSON Web Tokens (JWT) for authentication. We have a postman api doucmation files that can execute directly in the postman.

You can access postman documentation of APIs here: <https://documenter.getpostman.com/view/24028901/2sA3QqhD4x>

**The base URL for all endpoints is** [**https://student-assignment.onrender.com/**](https://student-assignment.onrender.com/)

**Authentication**

All endpoints that modify or retrieve assignments require authentication via a JWT token. The token must be included in the Authorization header of the request in the following format:

**Authorization: <jwtToken>**

**Endpoints**

**1. User Authentication**

**POST /api/login**

Mock authentication service that accepts any username and password, and returns a JWT token for subsequent requests.

Request Body:

{ "username": "string", "password": "string" }

Response:

{ message: “user logged in successfully", jwtToken": "string" }

Description:

• No validation on username and password.

• Returns a JWT token for authenticated requests.

**2. Assignment Management**

**POST /api/assignments**

Creates a new assignment.

Middlewares:

• verifyToken

Request Body:

{ "title": "string", "description": "string", "due\_date": "date" }

Response:

{ "message": "Assignment created successfully.", "id": "integer" }

Description:

• Validates the user using verifyToken middleware.

• Creates an assignment with the provided title, description, and due date.

**GET /api/assignments**

Retrieves all assignments of the authenticated teacher.

Middleware:

• verifyToken

Response:

[ { "id": "integer", "title": "string", "description": "string", "due\_date": "date"} ]

Description:

• Validates the user using verifyToken middleware.

• Returns all assignments created by the authenticated teacher.

**GET /api/assignments/:id**

Retrieves a specific assignment by ID.

Middleware:

• verifyToken

Request Parameters:

• id (integer): Assignment ID

Response:

{ "id": "integer", "title": "string", "description": "string", "due\_date": "date"}

Description:

• Validates the user using verifyToken middleware.

• Returns the assignment with the specified ID.

PUT /api/assignments/:id

Updates an existing assignment by ID.

Middleware:

• verifyToken

• teacherAuth

Request Parameters:

• id (integer): Assignment ID

Request Body:

{ "title": "string", "description": "string", "due\_date": "date" }

Response:

{ "message": "Assignment updated successfully." }

Description:

• Validates the user using verifyToken middleware.

• Ensures that only the teacher who created the assignment can update it using teacherAuth

Allows partial or full update of the assignment.

**DELETE /api/assignments/:id**

Deletes a specific assignment by ID.

Middleware:

• verifyToken

• teacherAuth

Request Parameters:

• id (integer): Assignment ID

Response:

{ "message": "Assignment deleted successfully." }

Description:

• Validates the user using verifyToken middleware.

• Ensures that only the teacher who created the assignment can delete it using teacherAuth

.

**DELETE /api/assignments**

Deletes all assignments of the authenticated teacher.

Middleware:

• verifyToken

• teacherAuth

Response:

{ "message": "All assignments deleted successfully." }

Description:

• Validates the user using verifyToken middleware.

• Ensures that only the teacher who created the assignments can delete them using teacherAuth middleware.

**POST /api/assignments/:id**

Submits an assignment.

Middleware:

• verifyToken

Request Parameters:

• id (integer): Assignment ID

Request Body:

{ "response": "string" }

Response:

{ "message": "Assignment submitted successfully." }

Description:

• Validates the user using verifyToken middleware.

• Allows any authenticated user/student to submit a response to the assignment with the specified ID.

**Middlewares**

* **verifyToken**

Verifies the JWT token provided in the Authorization header of the request. If the token is valid, the user is authenticated and can proceed with the request.

* **teacherAuth**

Ensures that only the teacher who created a specific assignment can update or delete it. This middleware checks the assignment's created\_by field against the authenticated user's ID.

**Error Handling**

All endpoints return appropriate HTTP status codes and error messages for various failure scenarios such as authentication failures, missing required fields, and invalid data.

**Contact**

For any questions or issues regarding the API, please contact the API developer at [ataul.computer786@gmail.com](mailto:ataul.computer786@gmail.com) or phone: 7983732026.