**Group 1**

**Student LMS BREAD using Postman**

Kaito Yamasaki

Christian Lei Sulit

Mark Justine Asuncion

Renz Fabon

Khalid Buenafe

John Lara

Ben Anselmo

Andrea Serrano

Marielle Talisik

Base URL

http://localhost/api/index.php

Process

1. **Create (POST)**

Method: POST

Description: Create new student in database

URL: http://localhost/api/POST/index.php

Student Create (POST) Student: This step involves adding a new student to the database. You send a POST request to http://localhost/api/POST/index.php with details like student\_id, name, email, and course\_code in the request body.

Request Body:

{

"student\_id": "2001-3005",

"name": "new student",

"email": "new student email",

"course\_code": "ITEC-111"

}

1. **Read (GET) Student**

Method: GET

Description: Retrieve all information of the students

URL: http://localhost/api

Read (GET) Student: To retrieve all student information, send a GET request to http://localhost/api. To fetch the details of all students stored in the database.

**3.Update (PUT) Student**

Method: PUT

Description: Update an existing student on database

URL: http://localhost/api/PUT/index.php?id=7

Update (PUT) Student: To modify an existing student's information, send a PUT request to http://localhost/api/PUT/index.php?id=7 (where '7' is the student ID to be updated) with the updated details (student\_id, name, email, course\_code) in the request body.

Request body:

{

"id": "designated id",

"student\_id": "updated id",

"name": "updated name",

"email": "updated email",

"course\_code": "updated course-code"

}

**4.Delete (DELETE) student**

Method: DELETE

Description: Delete an existing student on database

URL: http://localhost/api/DELETE/index.php?id="what id you want to delete"

Delete (DELETE) Student: To remove a student from the database, send a DELETE request to http://localhost/api/DELETE/index.php?id=<student\_id> (replace <student\_id> with the actual ID of the student to be deleted).

**Response Format**

Successful execution the response will have a status of 1 along with a message indicating the operation was successful. If there's an issue, such as invalid input data, the response will have a status of 0 and an error message specifying the nature of the error ("Error: Invalid input.").

Success Response

{

"status": 1,

"message": "Operation successful."

}

Error Response

{

"status": 0,

"message": "Error: Invalid input."

}