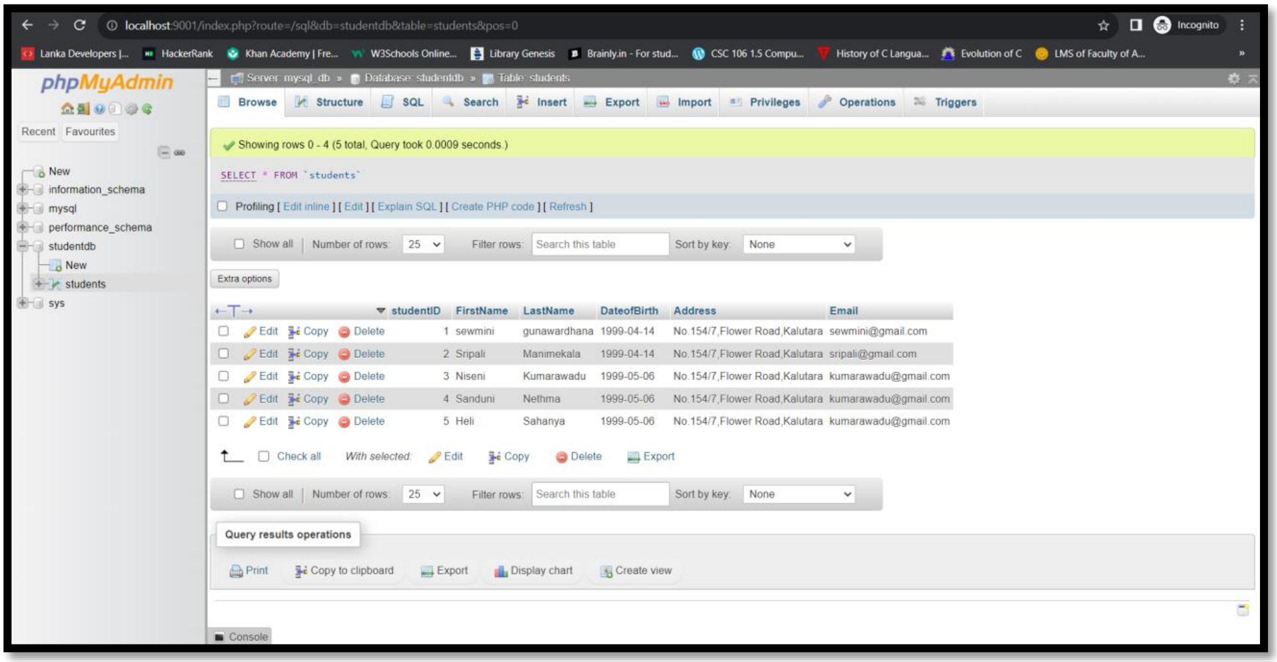
CSC 469 2.0 – Mobile Computing

Assignment 2023

K G S Manimekala
AS2019445

Setup the database



Showing rows 0 - 4 (5 total, Query took 0.0009 seconds.)

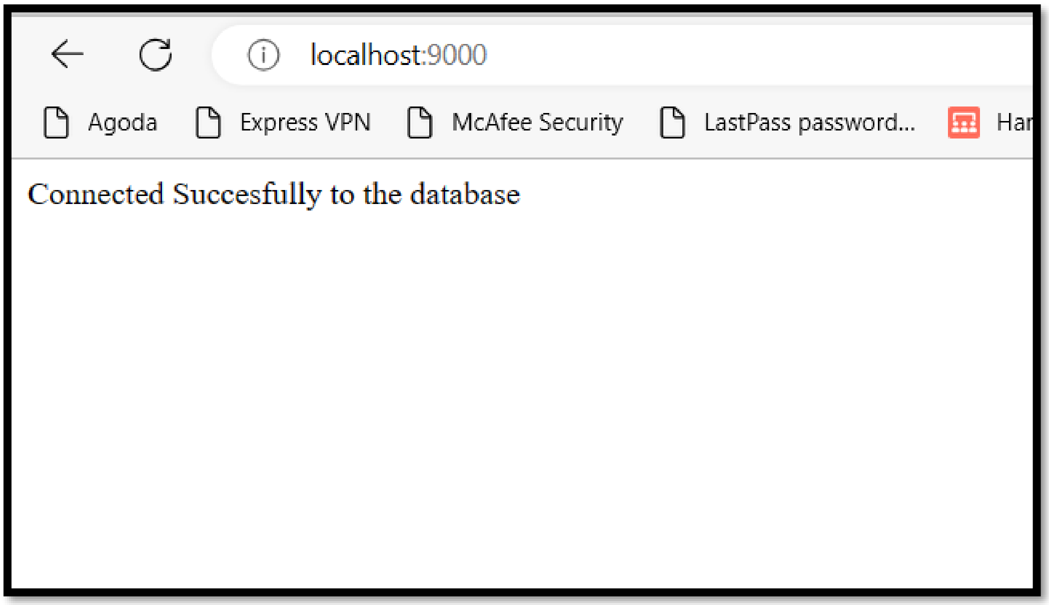
```
SELECT * FROM `students`
```

studentID	FirstName	LastName	DateofBirth	Address	Email
1	sewmini	gunawardhana	1999-04-14	No.154/7,Flower Road,Kalutara	sewmini@gmail.com
2	Sripali	Manimekala	1999-04-14	No.154/7,Flower Road,Kalutara	sripali@gmail.com
3	Niseni	Kumarawadu	1999-05-06	No.154/7,Flower Road,Kalutara	kumarawadu@gmail.com
4	Sanduni	Nethma	1999-05-06	No.154/7,Flower Road,Kalutara	kumarawadu@gmail.com
5	Heli	Sahanya	1999-05-06	No.154/7,Flower Road,Kalutara	kumarawadu@gmail.com

Figure 1 - phpMyAdmin workbench student Table Schema.

index.php

```php
<?php
$host = "mysql_db";
$username = "root";
$password = "root";
$db_name = "studentdb";

// Create connection
$conn = mysqli_connect($host,$username, $password,$db_name);

if(!$conn){
    die("Connection failed: ".mysqli_connect_error());
}
else{
    echo "Connected Succesfully to the database";
}
?>
```



Connected Succesfully to the database

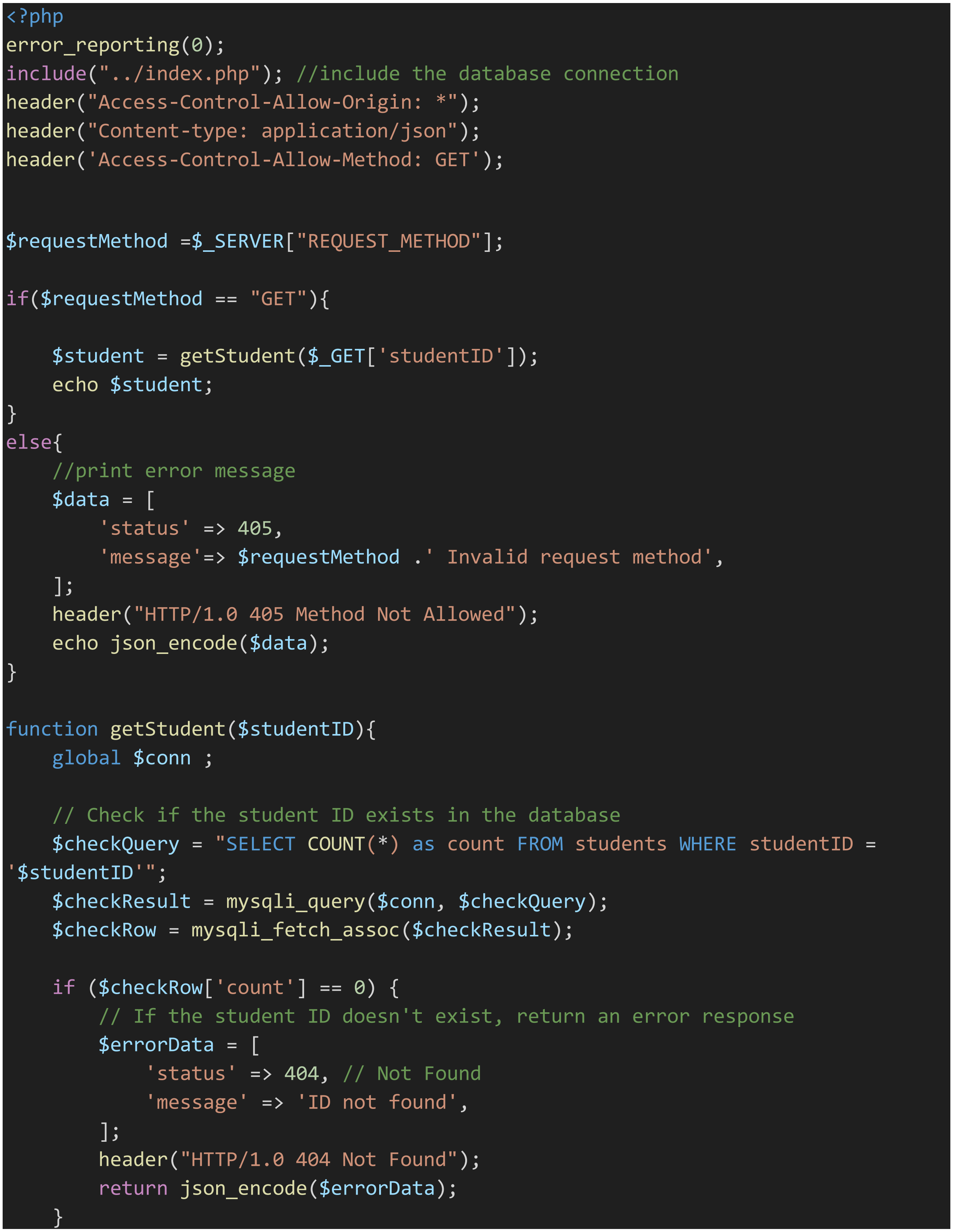
Figure 2 – Database connectivity confirmation.

create.php

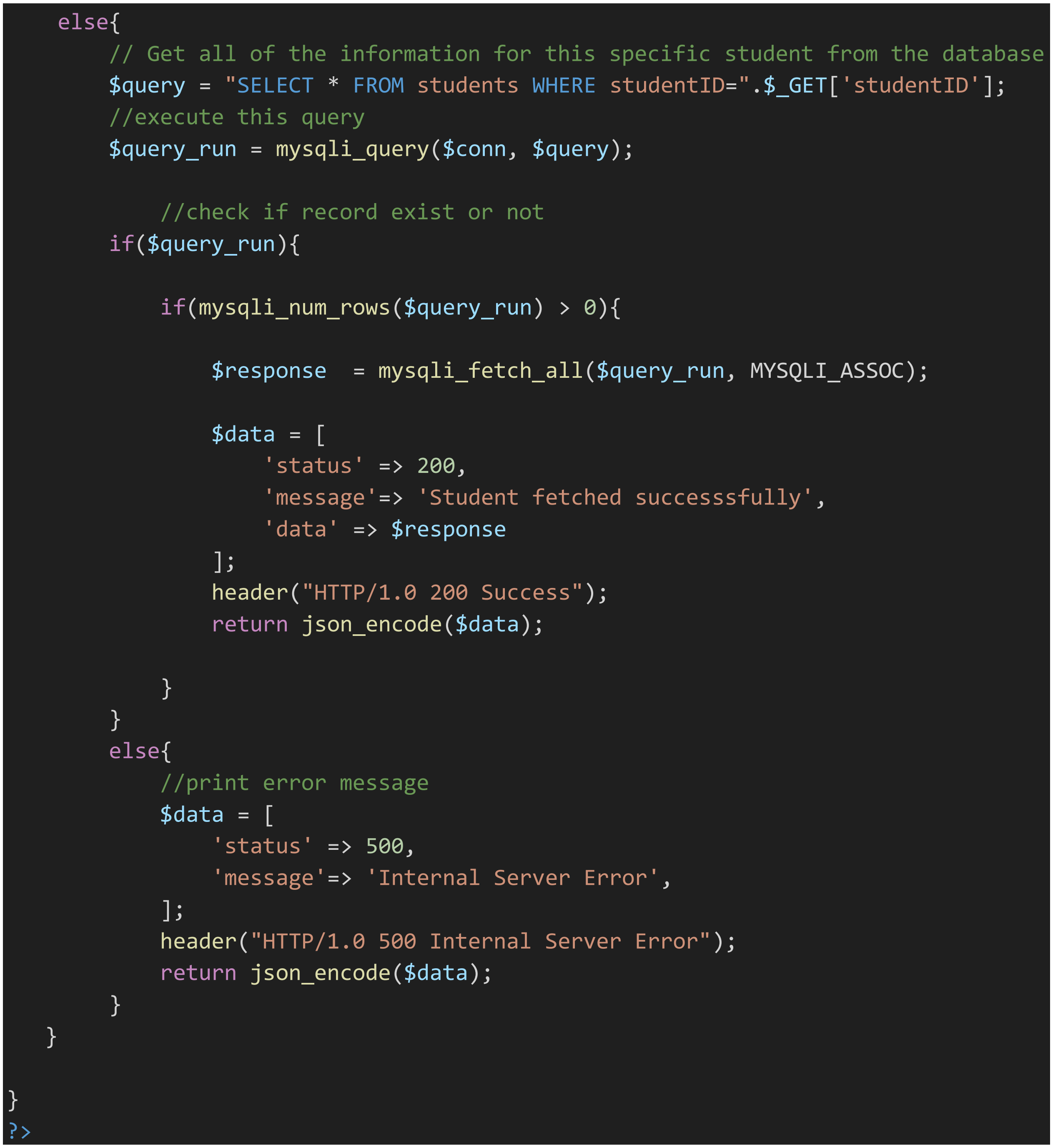
```php
<?php
error_reporting(0);
include("../index.php"); //include the database connection
header("Access-Control-Allow-Origin: *");
header("Content-type: application/json");
header('Access-Control-Allow-Method: POST');


$requestMethod =$_SERVER["REQUEST_METHOD"];

if($requestMethod == "POST"){

    $input_data= json_decode(file_get_contents("php://input"), true); //for any
data(raw data format)
    //if raw data is empty check post method
    if(empty($input_data)){
        //echo "not from raw data\n";
        $created_student = createStudent($_POST);
    }
    else{
        $created_student = createStudent($input_data);
    }
    echo $created_student;

}
else{
    //print error message
    $data = [
        'status' => 405,
        'message'=> $requestMethod .' Invalid request method',
    ];
    header("HTTP/1.0 405 Method Not Allowed");
    echo json_encode($data);
}

function createStudent($student){
    global $conn ;

    $studentID = mysqli_real_escape_string($conn, $student['studentID']);
    $FirstName = mysqli_real_escape_string($conn, $student['FirstName']);
    $LastName = mysqli_real_escape_string($conn, $student['LastName']);
    $DateofBirth = mysqli_real_escape_string($conn, $student['DateofBirth']);
    $Address = mysqli_real_escape_string($conn, $student['Address']);
    $Email = mysqli_real_escape_string($conn, $student['Email']);

    //validate
    //check all entries entered
    if(empty(trim($studentID)) || empty(trim($FirstName)) ||
empty(trim($LastName)) || empty(trim($DateofBirth)) || empty(trim($Address)) ||
empty(trim($Email))){
        return inputValidationError('Empty field detected');
    }
    else{
        $query = "INSERT INTO
students(studentID,FirstName,LastName,DateofBirth,Address,Email) VALUE
('$studentID','$FirstName','$LastName','$DateofBirth','$Address','$Email')";
        $query_run = mysqli_query($conn, $query);
        if($query_run){

            $responseData= array(
                "student ID" =>$studentID,
                "first Name" =>$FirstName ,
                "last Name" => $LastName,
                "date OfBirth" => $DateofBirth,
                "address" =>$Address,
                "email" =>$Email
            );
            echo json_encode($responseData, JSON_PRETTY_PRINT |
JSON_UNESCAPED_UNICODE);
            echo "\n";
            $data = [
                'status' => 201,
                'message'=> 'Student created successfully',
            ];
            header("HTTP/1.0 201 Created successfully");
            return json_encode($data);
        }
        else{
            $data = [
                'status' => 500,
                'message'=> 'Internal Server Error',
            ];
            header("HTTP/1.0 500 Internal Server Error");
            return json_encode($data);
        }
    }
}
```

```
function inputValidationError($message){
    $data = [
        'status' => 422,
        'message'=> $message,
    ];
    header("HTTP/1.0 422 unprocessable entity");
    echo json_encode($data);
}

?>
```

get.php



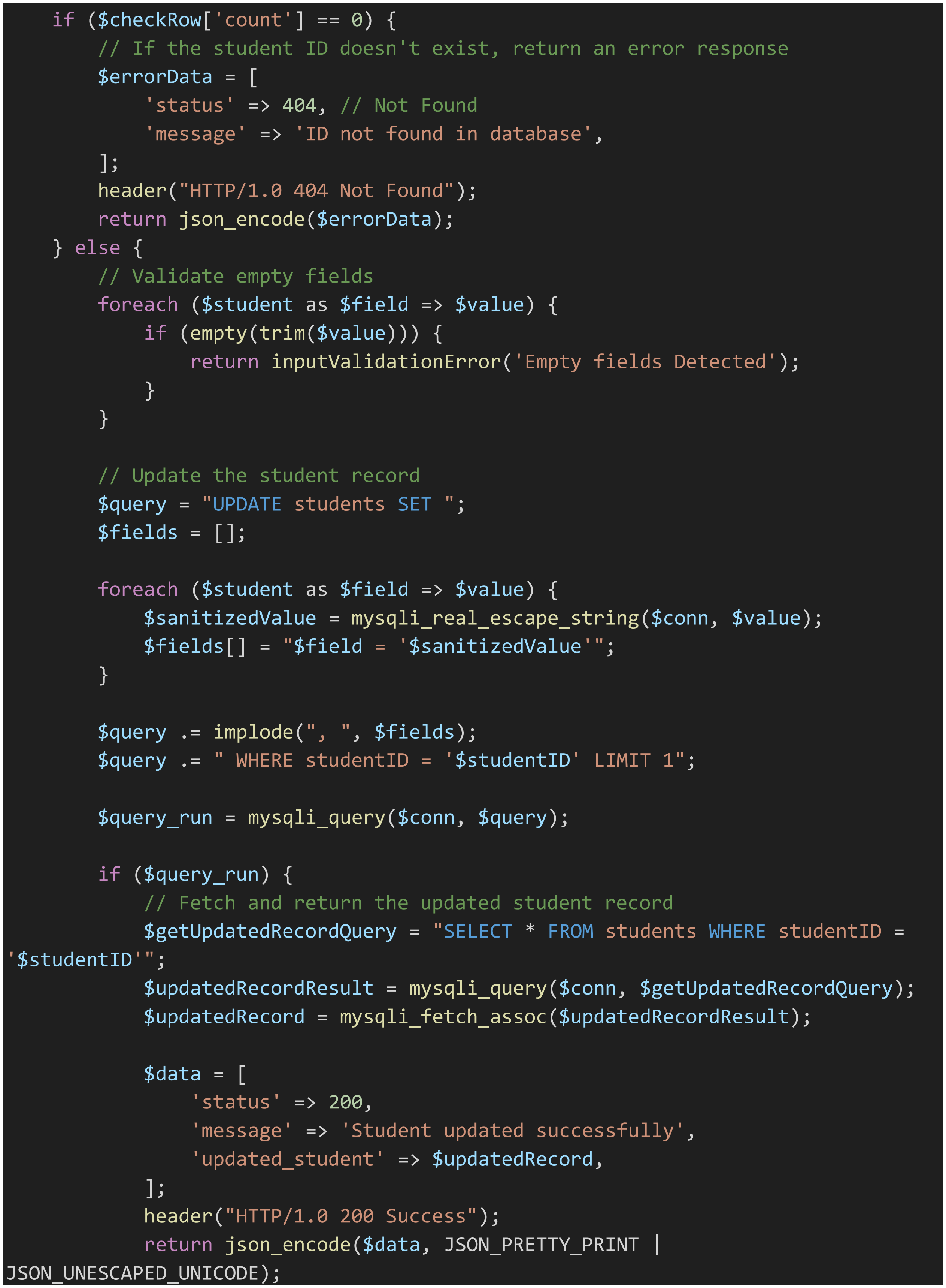
```php
<?php
error_reporting(0);
include("../index.php"); //include the database connection
header("Access-Control-Allow-Origin: *");
header("Content-type: application/json");
header('Access-Control-Allow-Method: GET');


$requestMethod =$_SERVER["REQUEST_METHOD"];

if($requestMethod == "GET"){

    $student = getStudent($_GET['studentID']);
    echo $student;
}
else{
    //print error message
    $data = [
        'status' => 405,
        'message'=> $requestMethod .' Invalid request method',
    ];
    header("HTTP/1.0 405 Method Not Allowed");
    echo json_encode($data);
}

function getStudent($studentID){
    global $conn ;

    // Check if the student ID exists in the database
    $checkQuery = "SELECT COUNT(*) as count FROM students WHERE studentID =
'$studentID'";
    $checkResult = mysqli_query($conn, $checkQuery);
    $checkRow = mysqli_fetch_assoc($checkResult);

    if ($checkRow['count'] == 0) {
        // If the student ID doesn't exist, return an error response
        $errorData = [
            'status' => 404, // Not Found
            'message' => 'ID not found',
        ];
        header("HTTP/1.0 404 Not Found");
        return json_encode($errorData);
    }
```



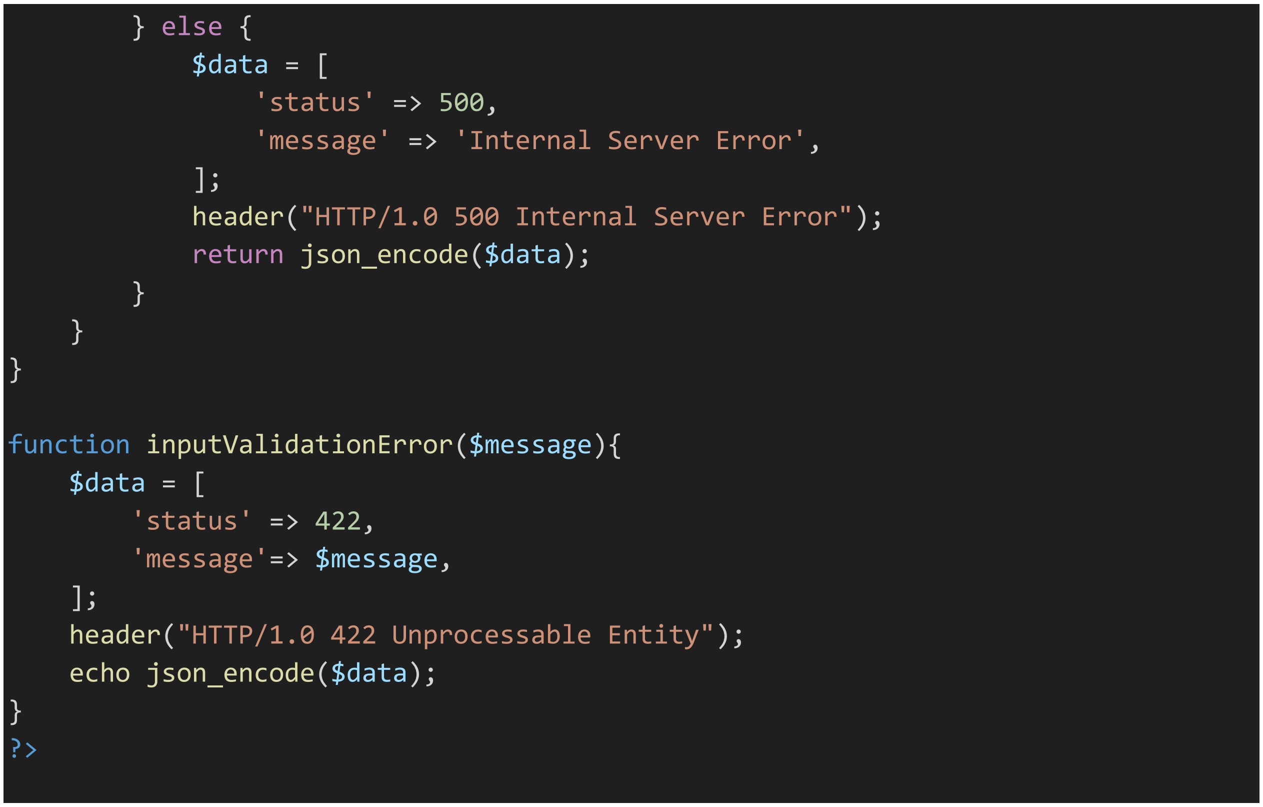
```
    else{
        // Get all of the information for this specific student from the database
        $query = "SELECT * FROM students WHERE studentID=".$_GET['studentID'];
        //execute this query
        $query_run = mysqli_query($conn, $query);

            //check if record exist or not
        if($query_run){

            if(mysqli_num_rows($query_run) > 0){

                $response  = mysqli_fetch_all($query_run, MYSQLI_ASSOC);

                $data = [
                    'status' => 200,
                    'message'=> 'Student fetched successsfully',
                    'data' => $response
                ];
                header("HTTP/1.0 200 Success");
                return json_encode($data);

            }
        }
        else{
            //print error message
            $data = [
                'status' => 500,
                'message'=> 'Internal Server Error',
            ];
            header("HTTP/1.0 500 Internal Server Error");
            return json_encode($data);
        }
   }

}
?>
```

update.php

```php
<?php
error_reporting(0);
include("../index.php"); //include the database connection
header("Access-Control-Allow-Origin: *");
header("Content-type: application/json");
header('Access-Control-Allow-Method: PUT');


$requestMethod = $_SERVER["REQUEST_METHOD"];

if ($requestMethod == "PUT") {

    $input_data = json_decode(file_get_contents("php://input"), true);
    $updated_student = updateStudent($input_data, $_GET);
    echo $updated_student;

} else {
    //print error message
    $data = [
        'status' => 405,
        'message' => $requestMethod .' Invalid request method',
    ];
    header("HTTP/1.0 405 Method Not Allowed");
    echo json_encode($data);
}

function updateStudent($student, $student_params){
    global $conn ;

    if (!isset($student_params['studentID'])){
        return inputValidationError('Student ID parameter not Found!');
    } elseif ($student_params['studentID'] == null) {
        return inputValidationError('Enter the student ID value');
    }

    $studentID = mysqli_real_escape_string($conn, $student_params['studentID']);

    // Check if the student ID exists in the database
    $checkQuery = "SELECT COUNT(*) as count FROM students WHERE studentID =
'$studentID'";
    $checkResult = mysqli_query($conn, $checkQuery);
    $checkRow = mysqli_fetch_assoc($checkResult);
```

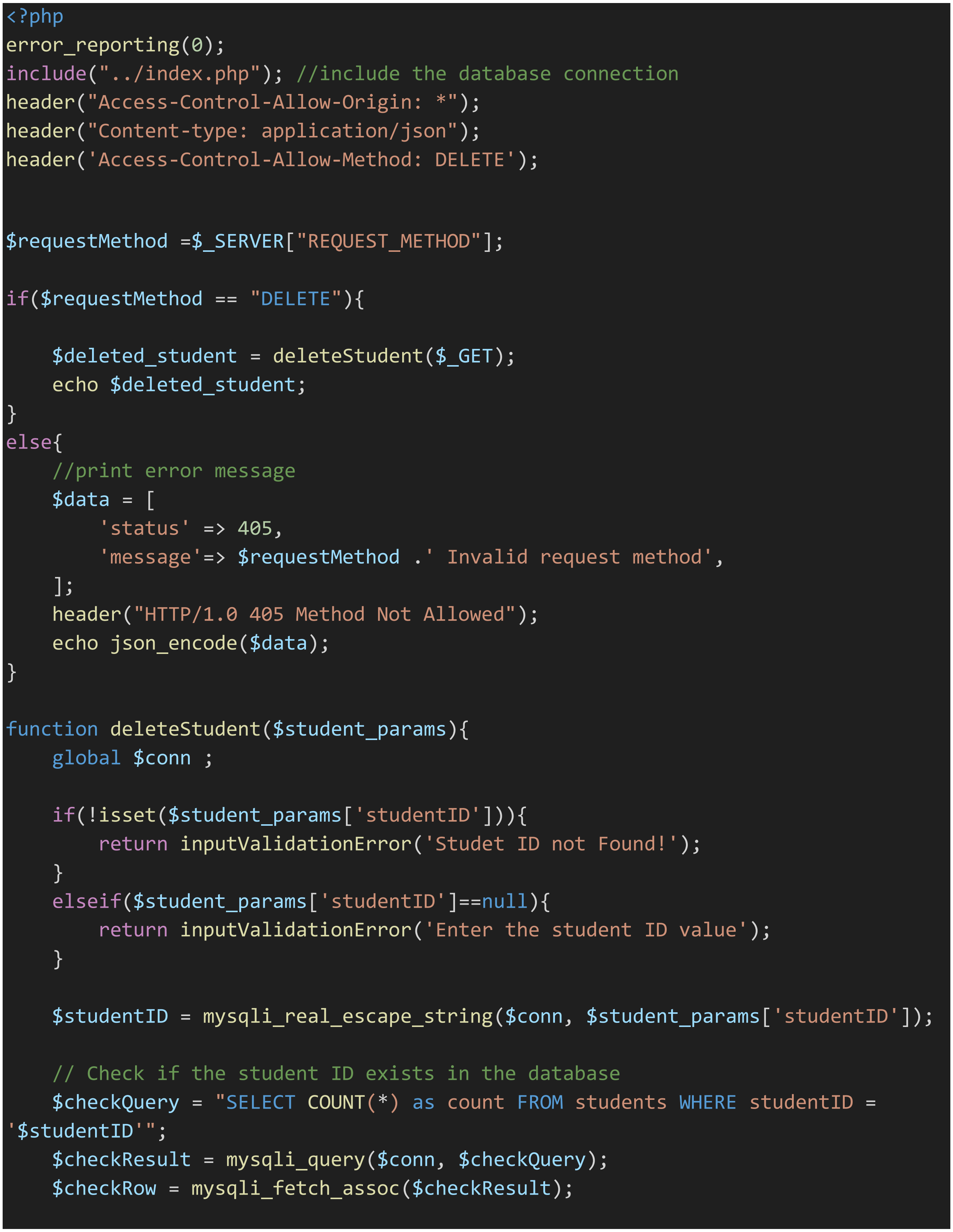


```
    if ($checkRow['count'] == 0) {
        // If the student ID doesn't exist, return an error response
        $errorData = [
            'status' => 404, // Not Found
            'message' => 'ID not found in database',
        ];
        header("HTTP/1.0 404 Not Found");
        return json_encode($errorData);
    } else {
        // Validate empty fields
        foreach ($student as $field => $value) {
            if (empty(trim($value))) {
                return inputValidationError('Empty fields Detected');
            }
        }

        // Update the student record
        $query = "UPDATE students SET ";
        $fields = [];

        foreach ($student as $field => $value) {
            $sanitizedValue = mysqli_real_escape_string($conn, $value);
            $fields[] = "$field = '$sanitizedValue'";
        }

        $query .= implode(", ", $fields);
        $query .= " WHERE studentID = '$studentID' LIMIT 1";

        $query_run = mysqli_query($conn, $query);

        if ($query_run) {
            // Fetch and return the updated student record
            $getUpdatedRecordQuery = "SELECT * FROM students WHERE studentID =
'$studentID'";
            $updatedRecordResult = mysqli_query($conn, $getUpdatedRecordQuery);
            $updatedRecord = mysqli_fetch_assoc($updatedRecordResult);

            $data = [
                'status' => 200,
                'message' => 'Student updated successfully',
                'updated_student' => $updatedRecord,
            ];
            header("HTTP/1.0 200 Success");
            return json_encode($data, JSON_PRETTY_PRINT |
JSON_UNESCAPED_UNICODE);
```

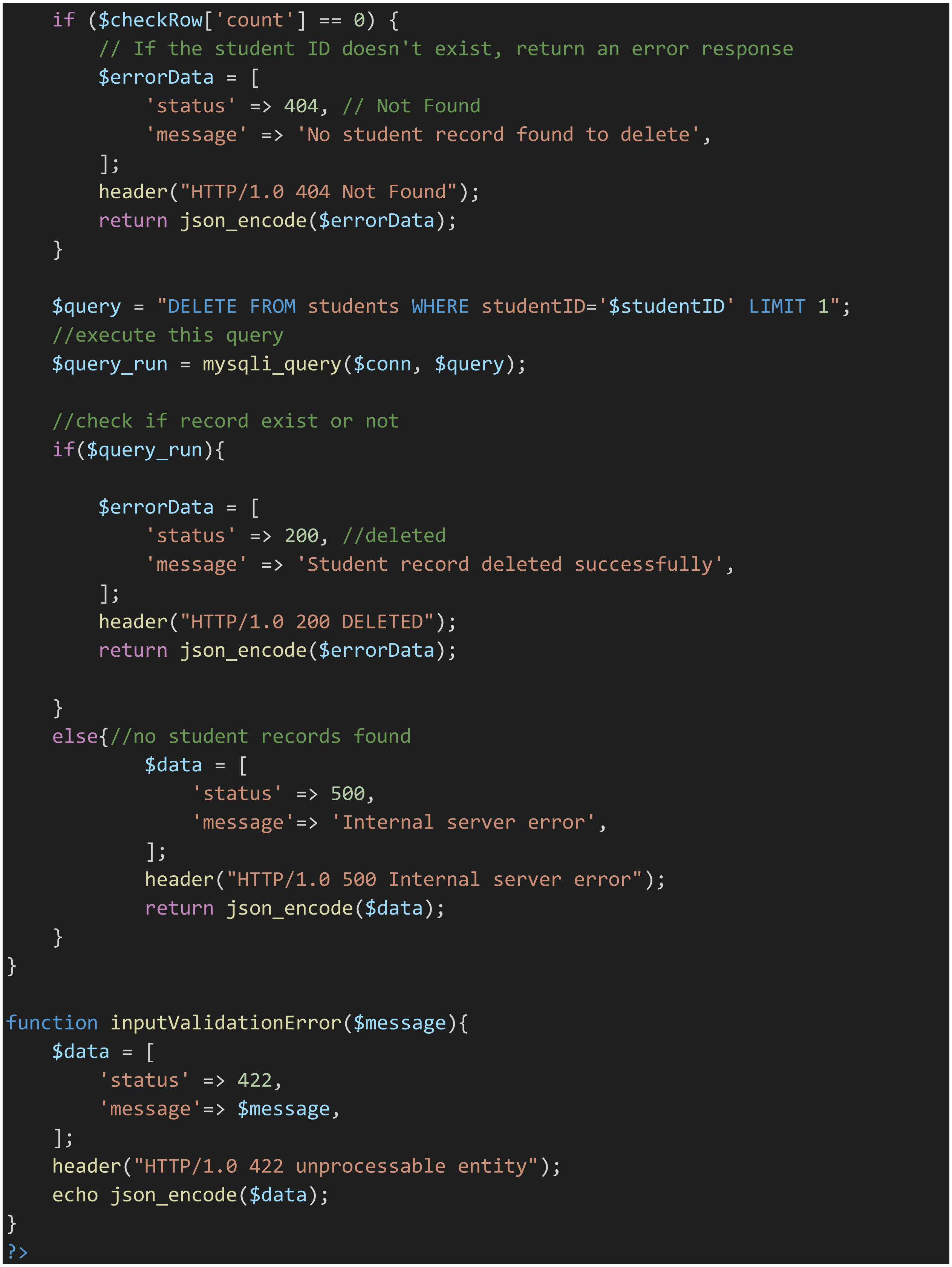


```
        } else {
            $data = [
                'status' => 500,
                'message' => 'Internal Server Error',
            ];
            header("HTTP/1.0 500 Internal Server Error");
            return json_encode($data);
        }
    }
}

function inputValidationError($message){
    $data = [
        'status' => 422,
        'message'=> $message,
    ];
    header("HTTP/1.0 422 Unprocessable Entity");
    echo json_encode($data);
}
?>
```

delete.php



```
<?php
error_reporting(0);
include("../index.php"); //include the database connection
header("Access-Control-Allow-Origin: *");
header("Content-type: application/json");
header('Access-Control-Allow-Method: DELETE');


$requestMethod =$_SERVER["REQUEST_METHOD"];

if($requestMethod == "DELETE"){

    $deleted_student = deleteStudent($_GET);
    echo $deleted_student;
}
else{
    //print error message
    $data = [
        'status' => 405,
        'message'=> $requestMethod .' Invalid request method',
    ];
    header("HTTP/1.0 405 Method Not Allowed");
    echo json_encode($data);
}

function deleteStudent($student_params){
    global $conn ;

    if(!isset($student_params['studentID'])){
        return inputValidationError('Studet ID not Found!');
    }
    elseif($student_params['studentID']==null){
        return inputValidationError('Enter the student ID value');
    }

    $studentID = mysqli_real_escape_string($conn, $student_params['studentID']);

    // Check if the student ID exists in the database
    $checkQuery = "SELECT COUNT(*) as count FROM students WHERE studentID =
'$studentID'";
    $checkResult = mysqli_query($conn, $checkQuery);
    $checkRow = mysqli_fetch_assoc($checkResult);
```



```
    if ($checkRow['count'] == 0) {
        // If the student ID doesn't exist, return an error response
        $errorData = [
            'status' => 404, // Not Found
            'message' => 'No student record found to delete',
        ];
        header("HTTP/1.0 404 Not Found");
        return json_encode($errorData);
    }

    $query = "DELETE FROM students WHERE studentID='$studentID' LIMIT 1";
    //execute this query
    $query_run = mysqli_query($conn, $query);

    //check if record exist or not
    if($query_run){

        $errorData = [
            'status' => 200, //deleted
            'message' => 'Student record deleted successfully',
        ];
        header("HTTP/1.0 200 DELETED");
        return json_encode($errorData);

    }
    else{//no student records found
            $data = [
                'status' => 500,
                'message'=> 'Internal server error',
            ];
            header("HTTP/1.0 500 Internal server error");
            return json_encode($data);
    }
}

function inputValidationError($message){
    $data = [
        'status' => 422,
        'message'=> $message,
    ];
    header("HTTP/1.0 422 unprocessable entity");
    echo json_encode($data);
}
?>
```

Title	CREATE Student
End point URL	http://localhost:9000/api/create.php
Method	POST
URL params	N/A

POST Params
(request format)
	{
	"studentID": "2",
	"FirstName": "Sripali",
	"LastName": "Manimekala",
	"DateofBirth": "1999-04-14",
	"Address": "No.154/7,Flower Road,Kalutara",
	"Email": "sripali@gmail.com"
	}

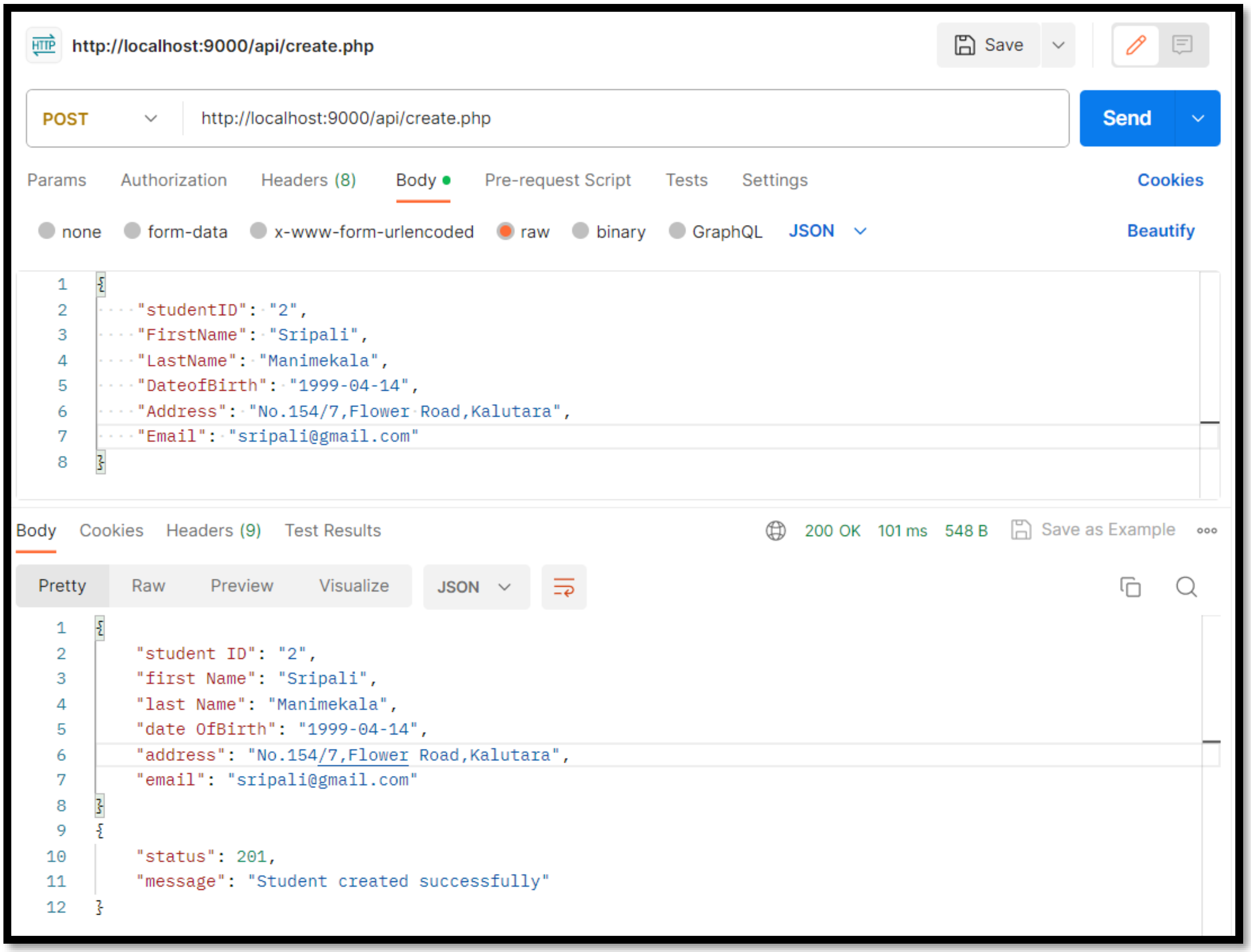
Success response
(Output format)
	{
	"student ID": "2",
	"first Name": "Sripali",
	"last Name": "Manimekala",
	"date OfBirth": "1999-04-14",
	"address": "No.154/7,Flower Road,Kalutara",
	"email": "sripali@gmail.com"
	}
	{
	"status": 201,
	"message": "Student created successfully"
	}

Error response
(Output format)
	{
	"status": 405,
	"message": "GET Invalid request method"
	}

	{
	"status": 422,
	"message": "Empty field detected"
	}

Notes	This API creates a new student entry in the students table in the student database when passing student details in JSON format or as form-data.

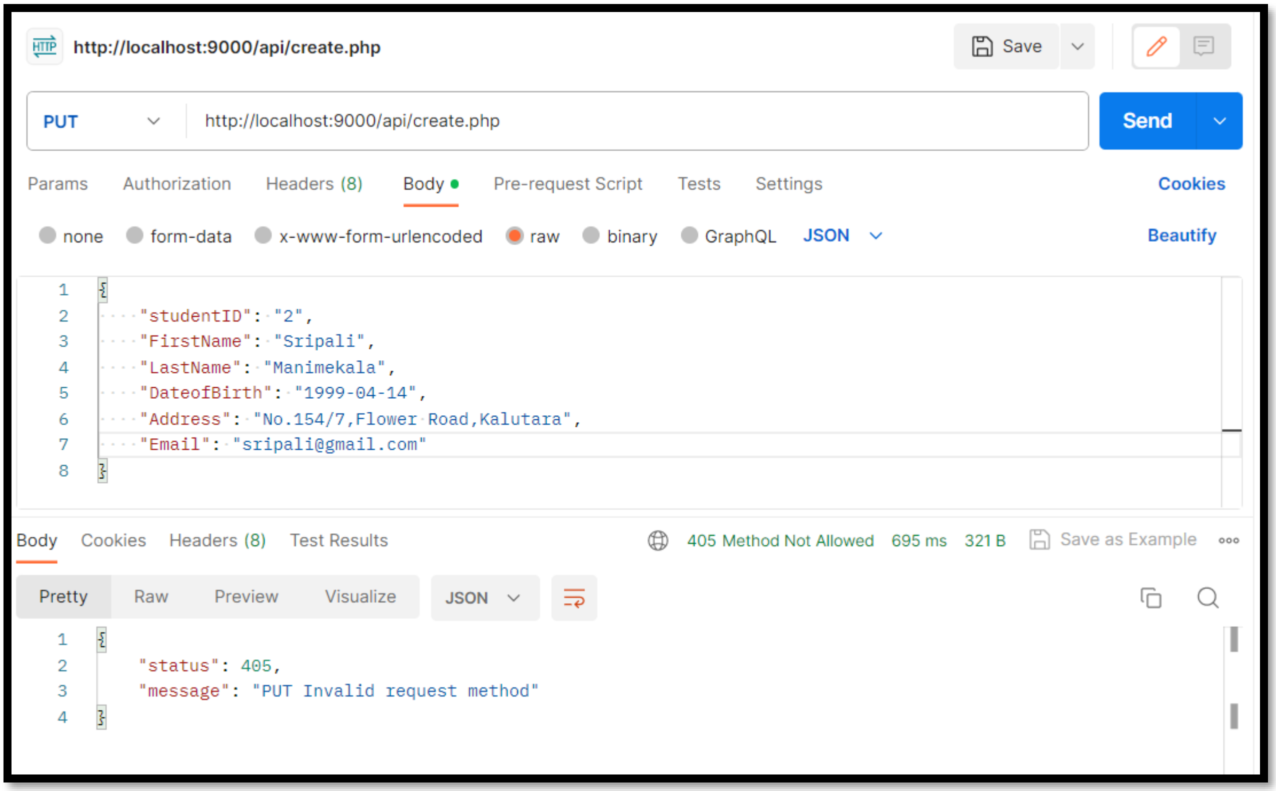
Table 1 – CREATE API details.



```
{
    "studentID": "2",
    "FirstName": "Sripali",
    "LastName": "Manimekala",
    "DateofBirth": "1999-04-14",
    "Address": "No.154/7,Flower Road,Kalutara",
    "Email": "sripali@gmail.com"
}
```

```
{
    "student ID": "2",
    "first Name": "Sripali",
    "last Name": "Manimekala",
    "date OfBirth": "1999-04-14",
    "address": "No.154/7,Flower Road,Kalutara",
    "email": "sripali@gmail.com"
}
{
    "status": 201,
    "message": "Student created successfully"
}
```

Figure 3 – Success Response.



http://localhost:9000/api/create.php

PUT http://localhost:9000/api/create.php

Send

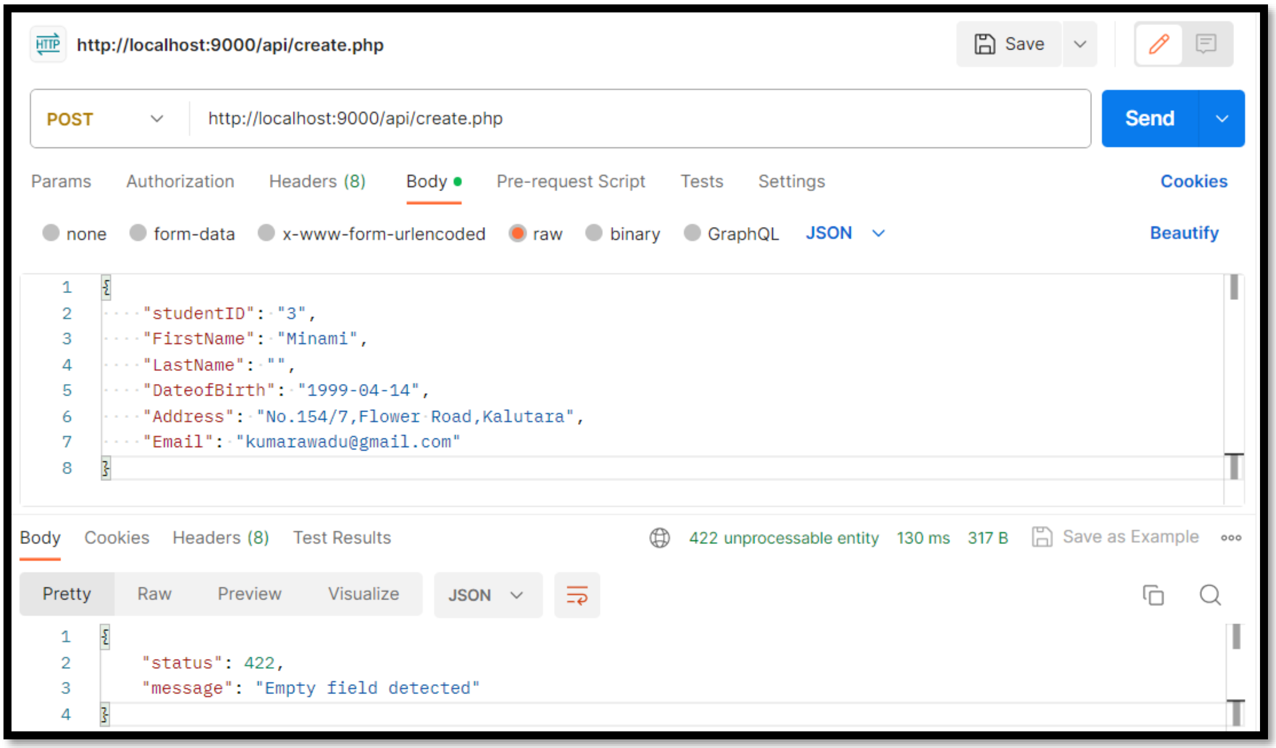
Body: raw JSON

```
{
    "studentID": "2",
    "FirstName": "Sripali",
    "LastName": "Manimekala",
    "DateofBirth": "1999-04-14",
    "Address": "No.154/7,Flower Road,Kalutara",
    "Email": "sripali@gmail.com"
}
```

Body — 405 Method Not Allowed 695 ms 321 B — JSON

```
{
    "status": 405,
    "message": "PUT Invalid request method"
}
```

Figure 4 – Error Response: Invalid Method PUT.



http://localhost:9000/api/create.php

POST http://localhost:9000/api/create.php

Send

Body: raw JSON

```
{
    "studentID": "3",
    "FirstName": "Minami",
    "LastName": "",
    "DateofBirth": "1999-04-14",
    "Address": "No.154/7,Flower Road,Kalutara",
    "Email": "kumarawadu@gmail.com"
}
```

Body — 422 unprocessable entity 130 ms 317 B — JSON

```
{
    "status": 422,
    "message": "Empty field detected"
}
```

Figure 5 – Error Response: Empty Field.

Title	READ Student
End point URL	http://localhost:9000/api/get.php
Method	GET
URL params	studentID --> (?studentID=2)
POST Params (Request format)	N/A

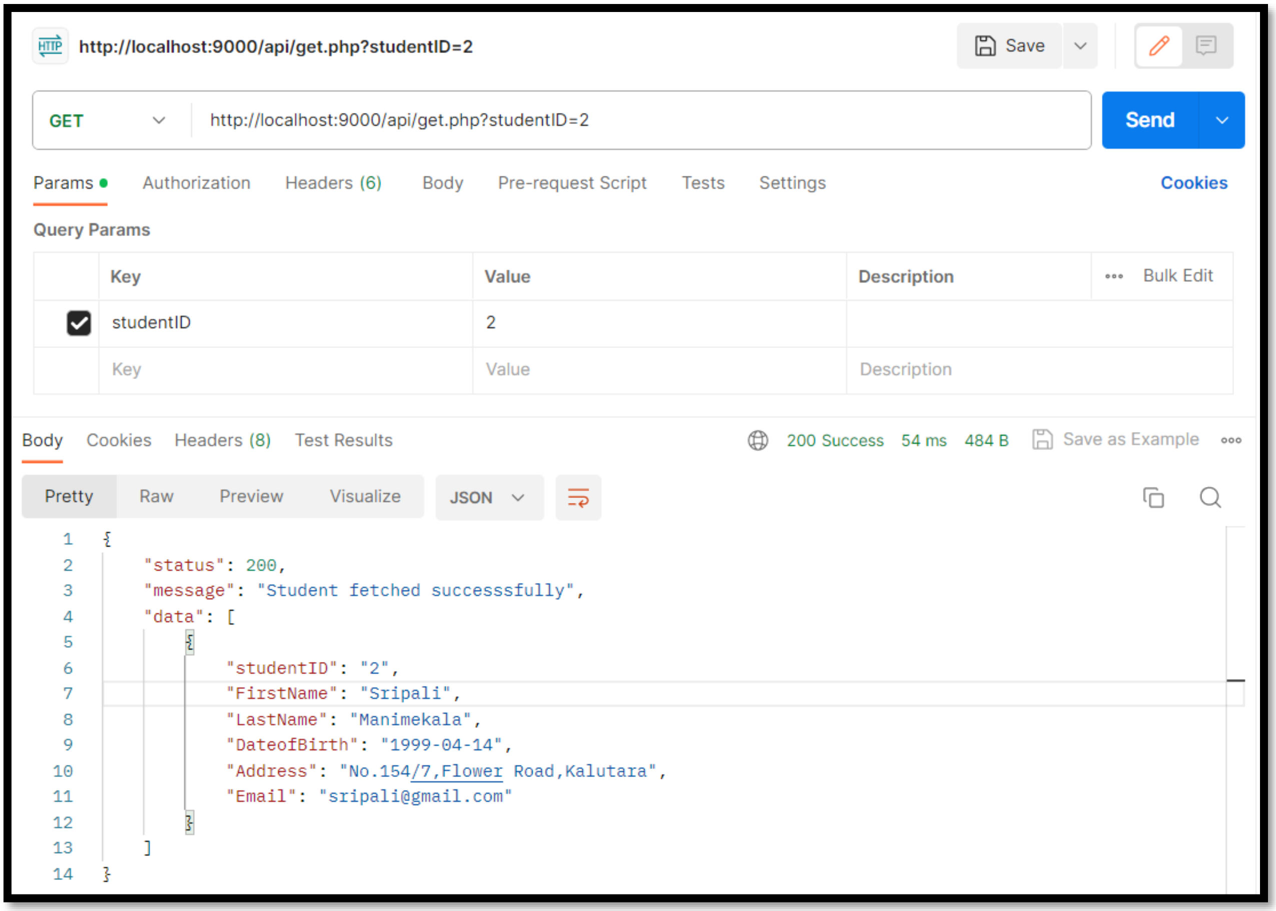
Success response
(Output format)
	{
	"status": 200,
	"message": "Student fetched successsfully",
	"data": [
	{
	"studentID": "2",
	"FirstName": "Sripali",
	"LastName": "Manimekala",
	"DateofBirth": "1999-04-14",
	"Address": "No.154/7,Flower Road,Kalutara",
	"Email": "sripali@gmail.com"
	}
	]
	}

Error response
(Output format)
	{
	"status": 405,
	"message": "POST Invalid request method"
	}

	{
	"status": 404,
	"message": "ID not found"
	}

Notes	This API reads a existing student record in the students table in the student database when given the student ID as query parameter in the request URL.

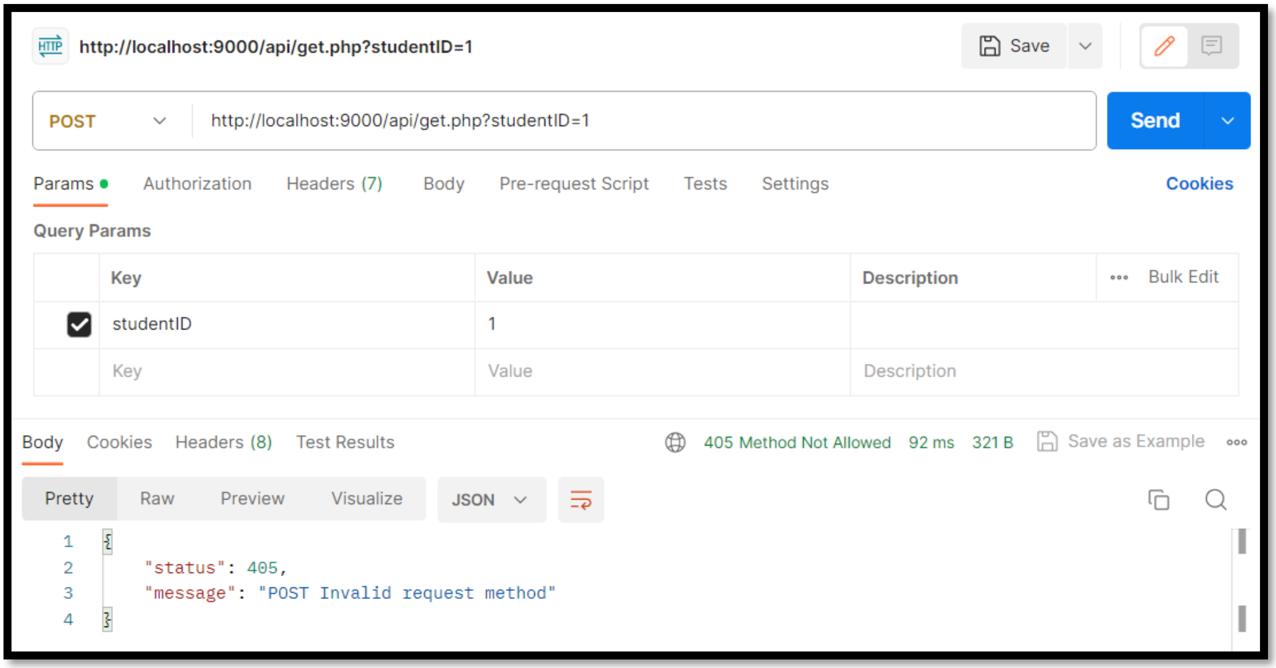
Table 2 – READ API details.



	Key	Value	Description
☑	studentID	2	
	Key	Value	Description

```
{
    "status": 200,
    "message": "Student fetched successsfully",
    "data": [
        {
            "studentID": "2",
            "FirstName": "Sripali",
            "LastName": "Manimekala",
            "DateofBirth": "1999-04-14",
            "Address": "No.154/7,Flower Road,Kalutara",
            "Email": "sripali@gmail.com"
        }
    ]
}
```

Figure 6 – Success Response.

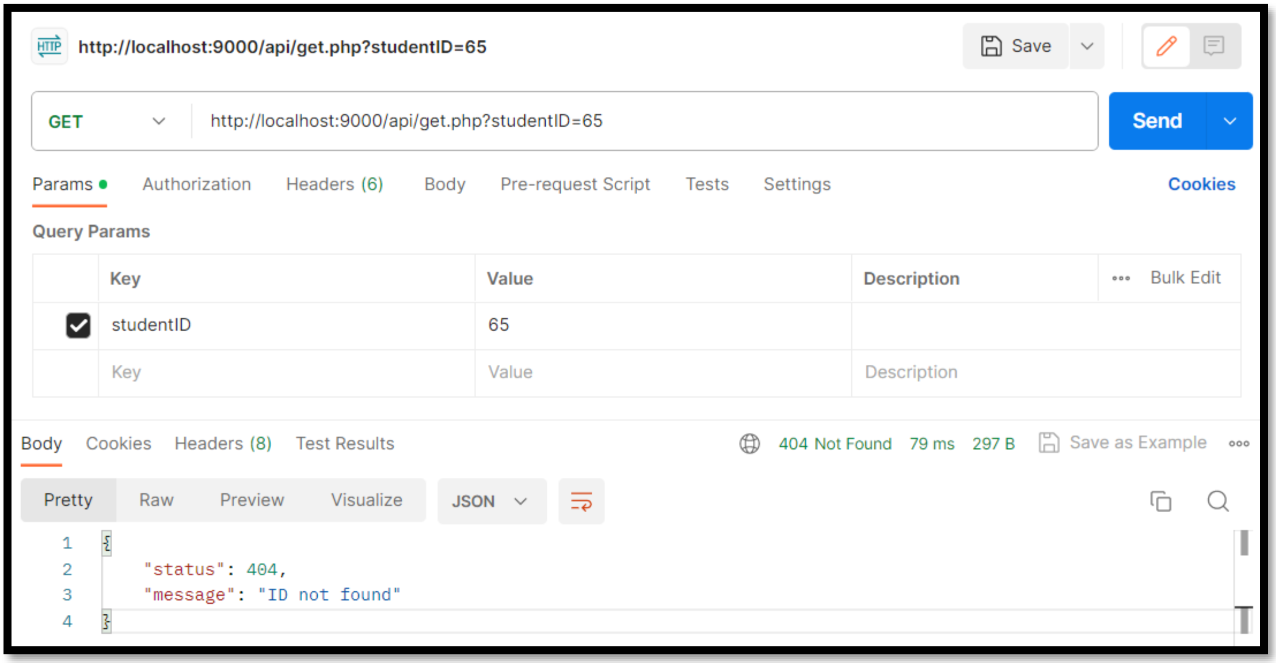


	Key	Value	Description
☑	studentID	1	
	Key	Value	Description

405 Method Not Allowed 92 ms 321 B

```
{
    "status": 405,
    "message": "POST Invalid request method"
}
```

Figure 7 – Error Response: Invalid Method POST.



	Key	Value	Description
☑	studentID	65	
	Key	Value	Description

404 Not Found 79 ms 297 B

```
{
    "status": 404,
    "message": "ID not found"
}
```

Figure 8 – Error Response: studentID Not Found.

Title	UPDATE Student
End point URL	http://localhost:9000/api/update.php
Method	PUT
URL params	studentID --> (?studentID=2)

POST Params
(request format)
	{
	  "FirstName": "Sripali",
	  "LastName": "Manimekala",
	  "Email": "xyz@gmail.com"
	}

Success response
(Output format)
	{
	    "status": 200,
	    "message": "Student updated successfully",
	    "updated_student": {
	        "studentID": "4",
	        "FirstName": "Sripali",
	        "LastName": "Manimekala",
	        "DateofBirth": "1999-05-06",
	        "Address": "No.154/7,Flower Road,Kalutara",
	        "Email": "xyz@gmail.com"
	    }
	}

Error response
(Output format)
	{
	    "status": 405,
	    "message": "DELETE Invalid request method"
	}

	{
	    "status": 422,
	    "message": "Student ID parameter not Found!"
	}

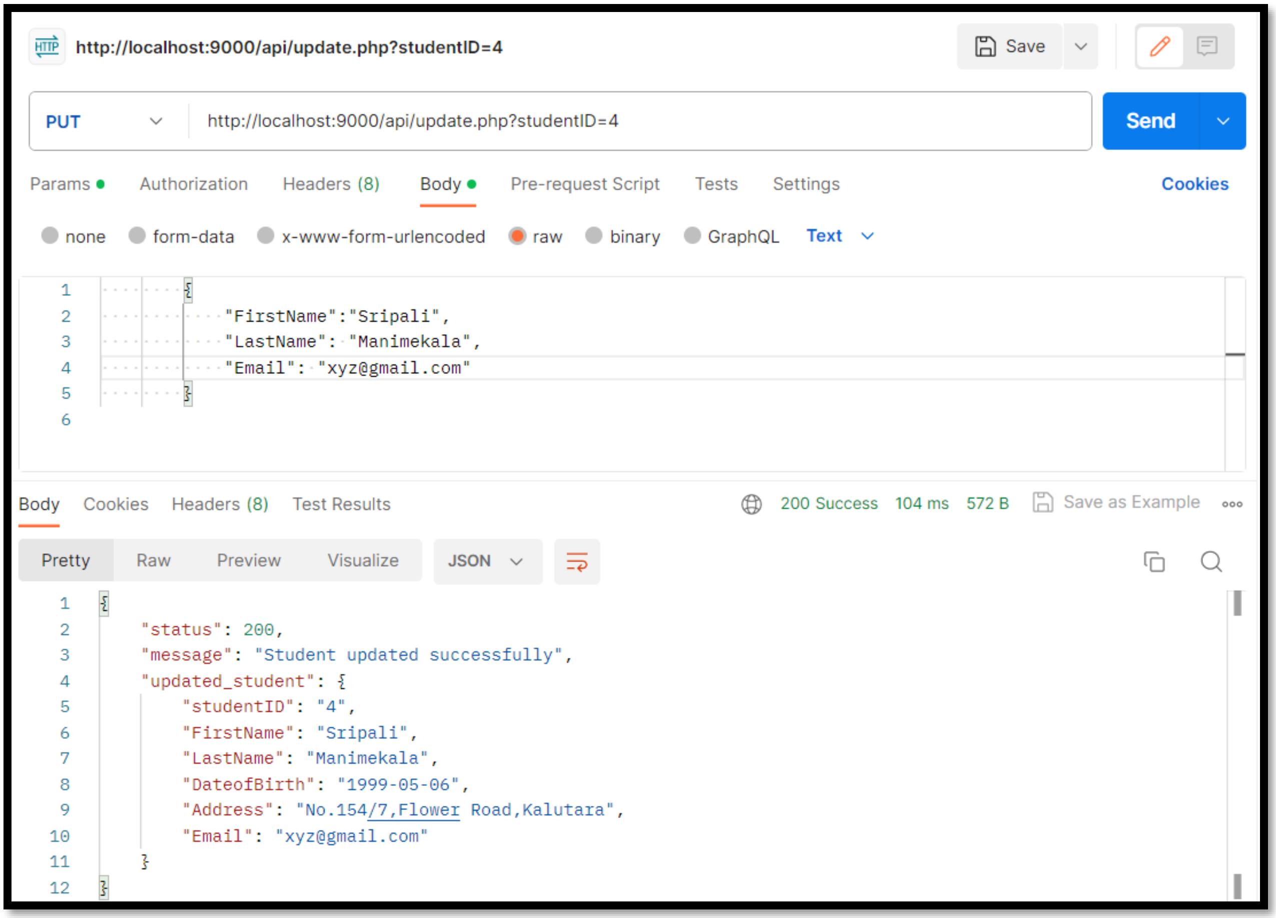
	{
	    "status": 422,
	    "message": "Enter the student ID value"
	}

	{
	    "status": 404,
	    "message": "ID not found in database"
	}

	{
	    "status": 422,
	    "message": "Empty fields Detected"
	}

Notes	This API updates student record in student table when passing only needed student details in JSON format. Student ID is passed through URL as query parameter.

Table 3 – UPDATE API details.

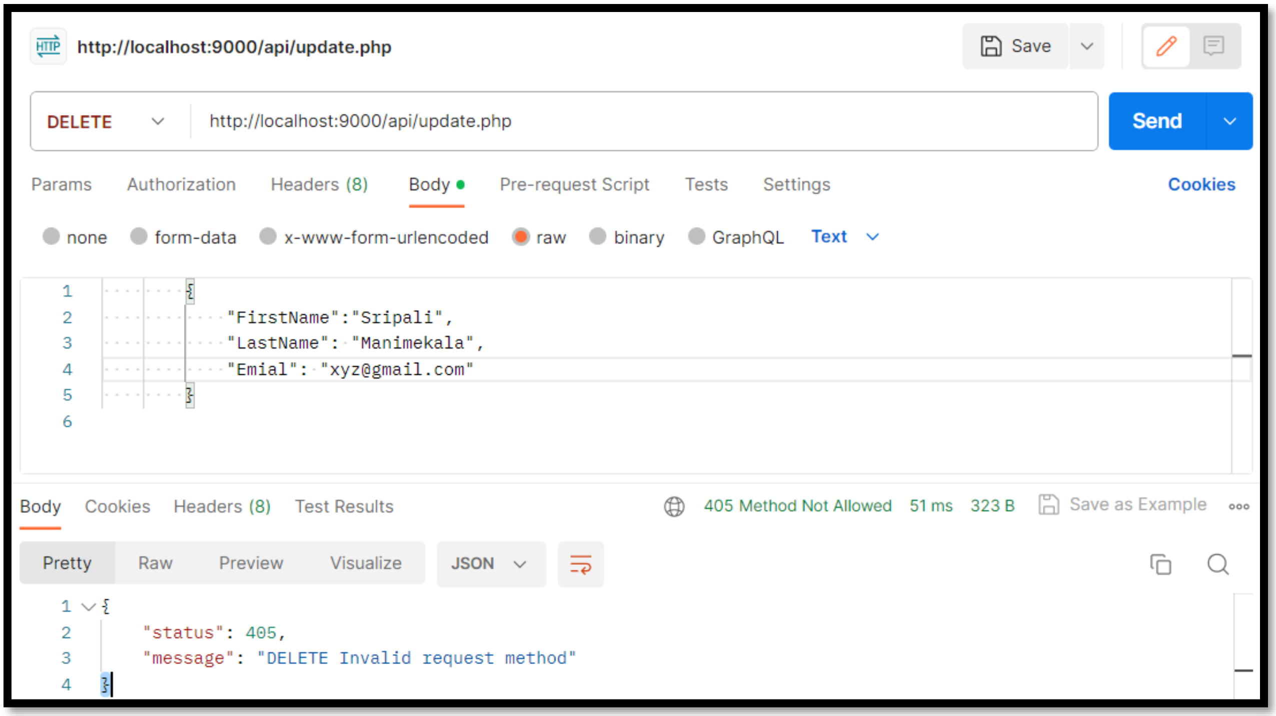


PUT http://localhost:9000/api/update.php?studentID=4

```
{
    "FirstName":"Sripali",
    "LastName": "Manimekala",
    "Email": "xyz@gmail.com"
}
```

```
{
    "status": 200,
    "message": "Student updated successfully",
    "updated_student": {
        "studentID": "4",
        "FirstName": "Sripali",
        "LastName": "Manimekala",
        "DateofBirth": "1999-05-06",
        "Address": "No.154/7,Flower Road,Kalutara",
        "Email": "xyz@gmail.com"
    }
}
```

Figure 9 – Success Response.



http://localhost:9000/api/update.php

DELETE http://localhost:9000/api/update.php

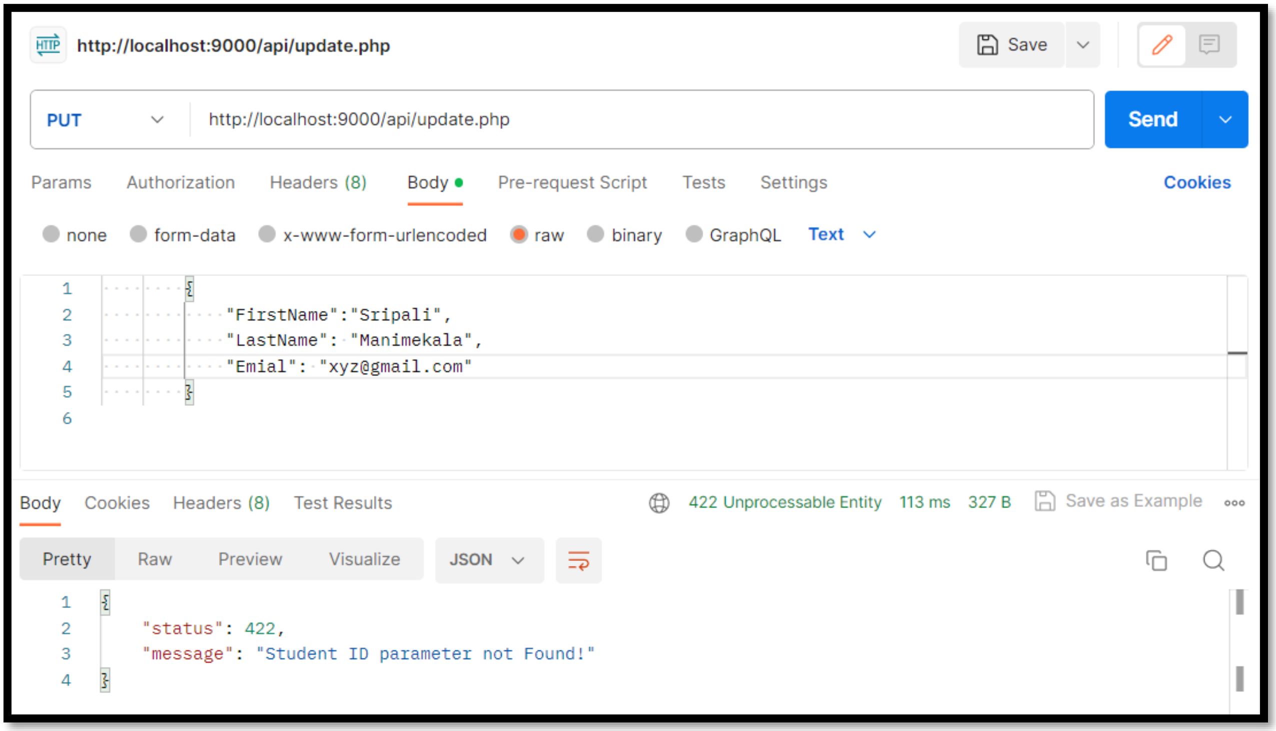
Body: raw, Text

```
{
    "FirstName":"Sripali",
    "LastName": "Manimekala",
    "Emial": "xyz@gmail.com"
}
```

405 Method Not Allowed 51 ms 323 B

```
{
    "status": 405,
    "message": "DELETE Invalid request method"
}
```

Figure 10 – Error Response: Invalid Method DELETE.



http://localhost:9000/api/update.php

PUT http://localhost:9000/api/update.php

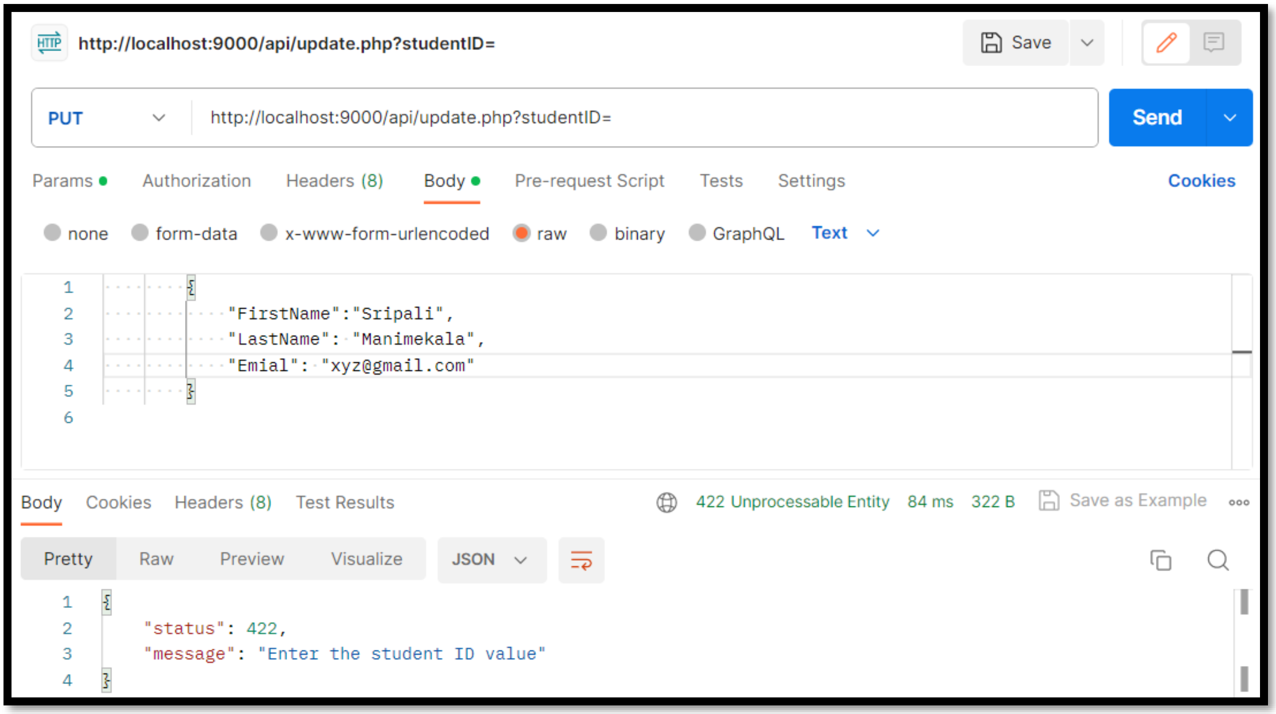
Body: raw, Text

```
{
    "FirstName":"Sripali",
    "LastName": "Manimekala",
    "Emial": "xyz@gmail.com"
}
```

422 Unprocessable Entity 113 ms 327 B

```
{
    "status": 422,
    "message": "Student ID parameter not Found!"
}
```

Figure 11 – Error Response: studentID not given.



http://localhost:9000/api/update.php?studentID=

PUT http://localhost:9000/api/update.php?studentID=

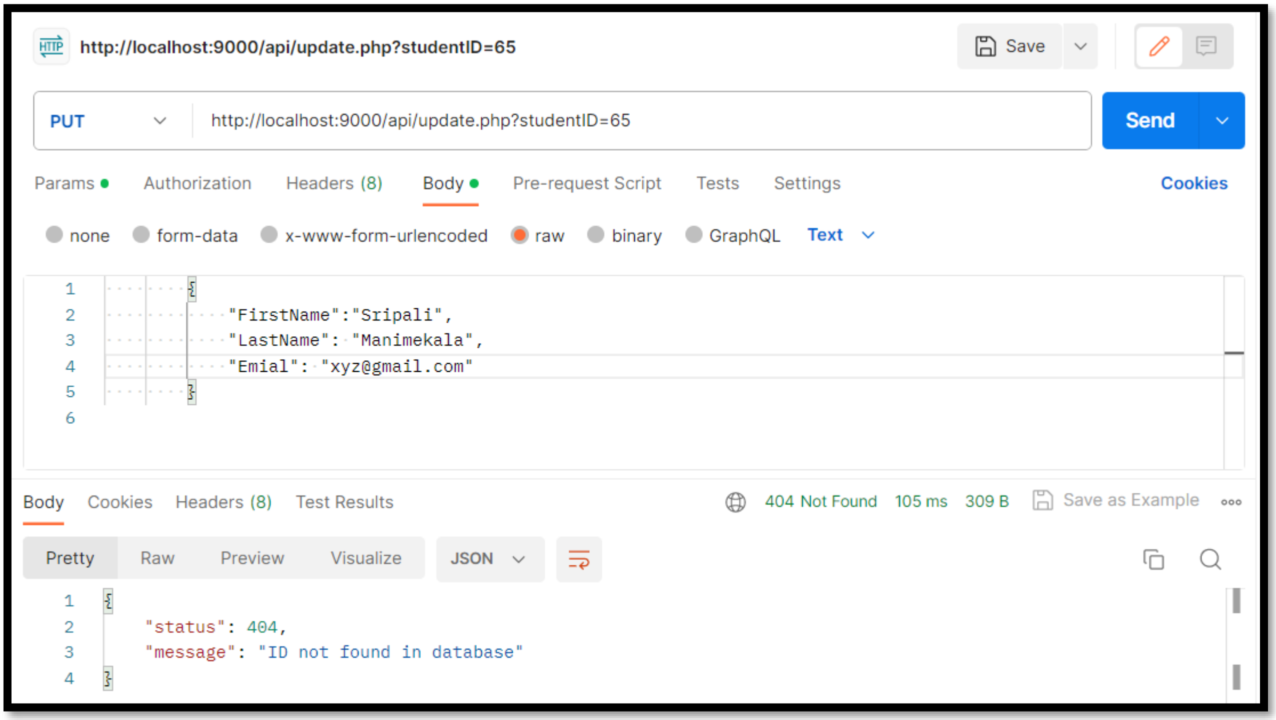
Send

```
{
    "FirstName":"Sripali",
    "LastName": "Manimekala",
    "Emial": "xyz@gmail.com"
}
```

422 Unprocessable Entity 84 ms 322 B

```
{
    "status": 422,
    "message": "Enter the student ID value"
}
```

Figure 11 – Error Response: studentID value not given.



http://localhost:9000/api/update.php?studentID=65

PUT http://localhost:9000/api/update.php?studentID=65

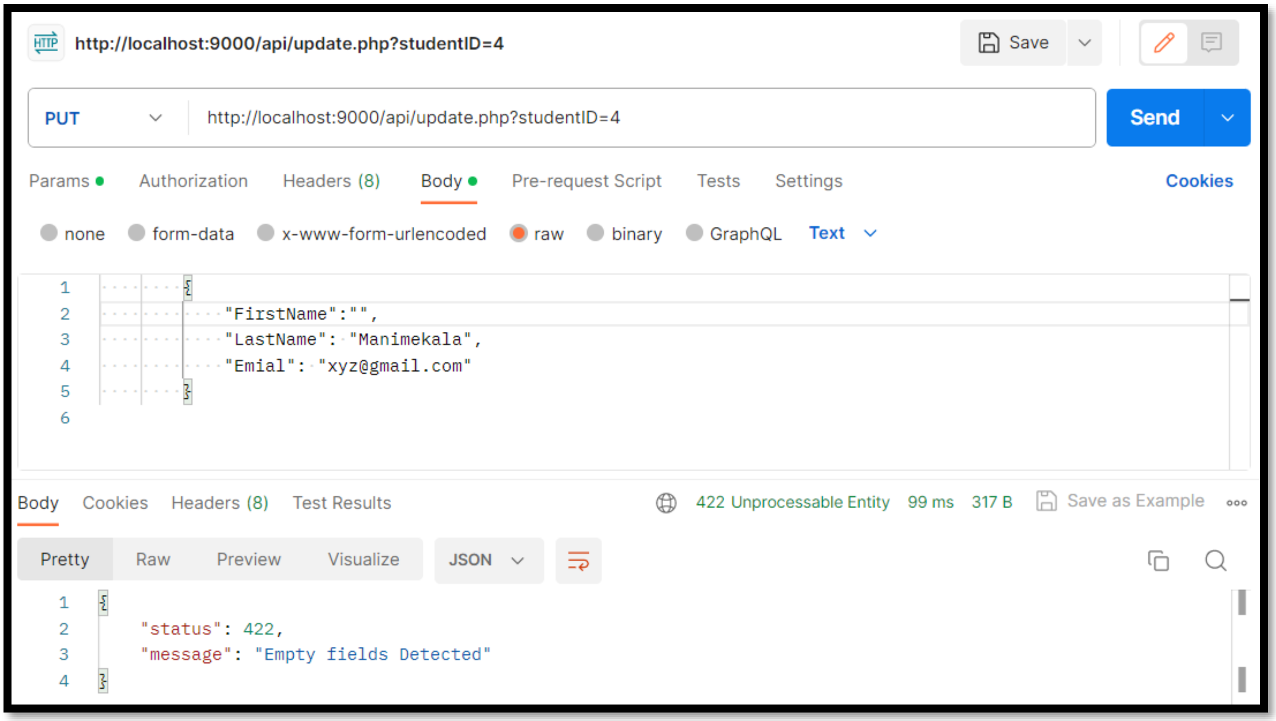
Send

```
{
    "FirstName":"Sripali",
    "LastName": "Manimekala",
    "Emial": "xyz@gmail.com"
}
```

404 Not Found 105 ms 309 B

```
{
    "status": 404,
    "message": "ID not found in database"
}
```

Figure 11 – Error Response: Invalid student ID Empty Field.



PUT http://localhost:9000/api/update.php?studentID=4

```
{
    "FirstName":"",
    "LastName": "Manimekala",
    "Emial": "xyz@gmail.com"
}
```

422 Unprocessable Entity 99 ms 317 B

```
{
    "status": 422,
    "message": "Empty fields Detected"
}
```

Figure 11 – Error Response: Empty Field.

Title	**DELETE Student**
End point URL	http://localhost:9000/api/delete.php
Method	DELETE
URL params	studentID --> (?studentID=2)
POST Params (request format)	N/A

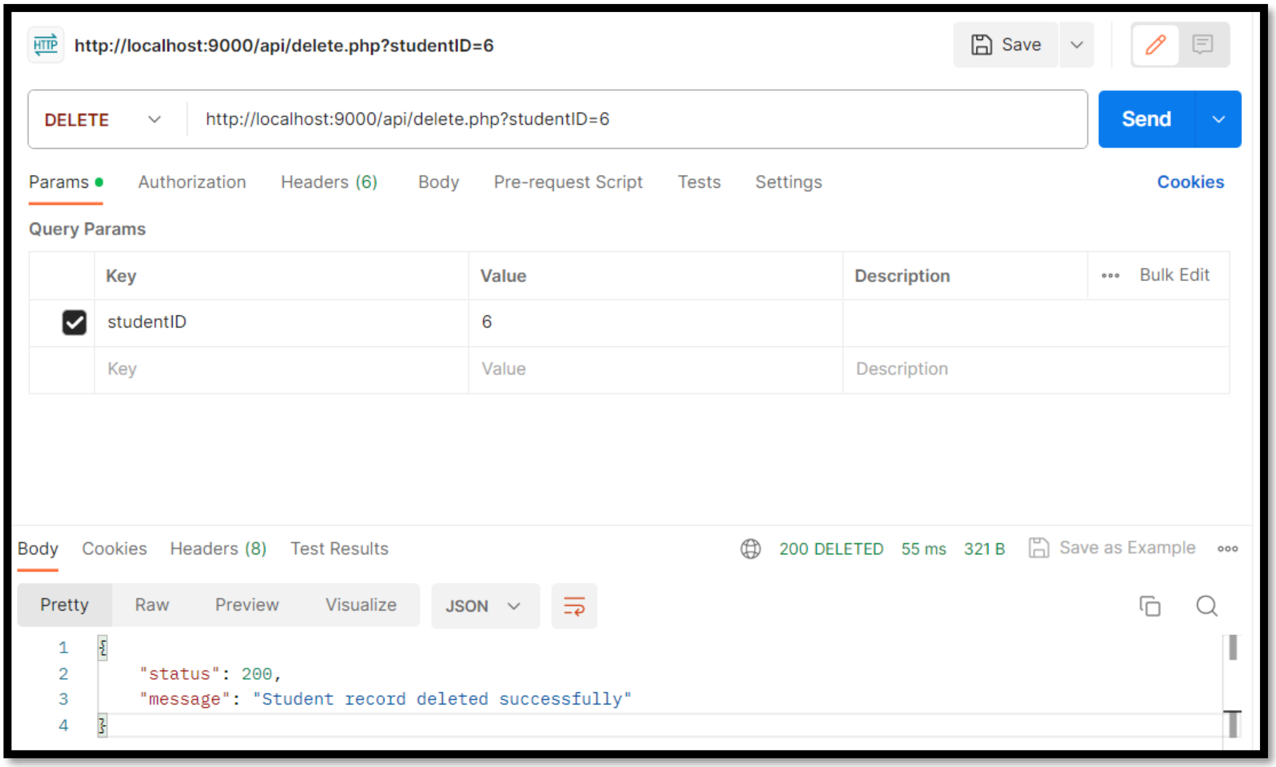
Success response (Output format)	{
	"status": 200,
	"message": "Student record deleted successfully"
	}

Error response (Output format)	{
	"status": 405,
	"message": "POST Invalid request method"
	}

	{
	"status": 404,
	"message": "No student record found to delete"
	}

Notes	This API deletes a student record entry in the database specified by the student ID. Student ID is passed as a query parameter along with the URL.

Table 4 – DELETE API details.



http://localhost:9000/api/delete.php?studentID=6

DELETE http://localhost:9000/api/delete.php?studentID=6

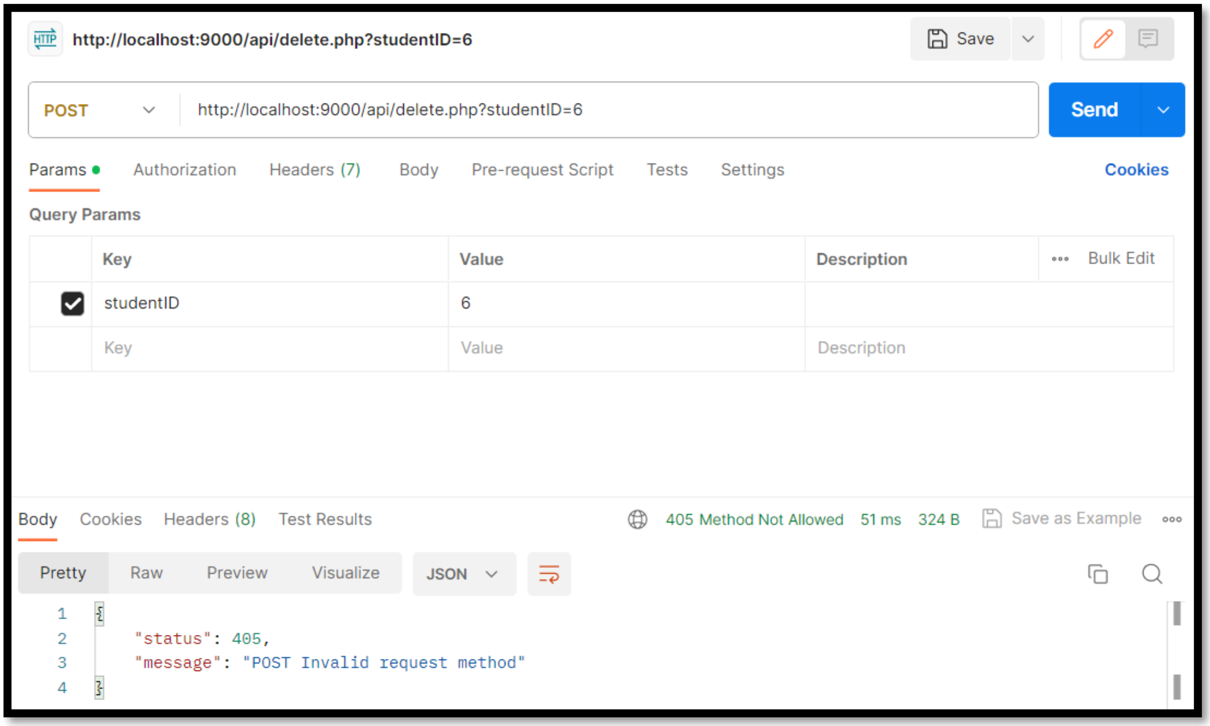
Query Params

	Key	Value	Description
☑	studentID	6	
	Key	Value	Description

200 DELETED 55 ms 321 B

```
{
    "status": 200,
    "message": "Student record deleted successfully"
}
```

Figure 12 – Success Response.



http://localhost:9000/api/delete.php?studentID=6

POST http://localhost:9000/api/delete.php?studentID=6

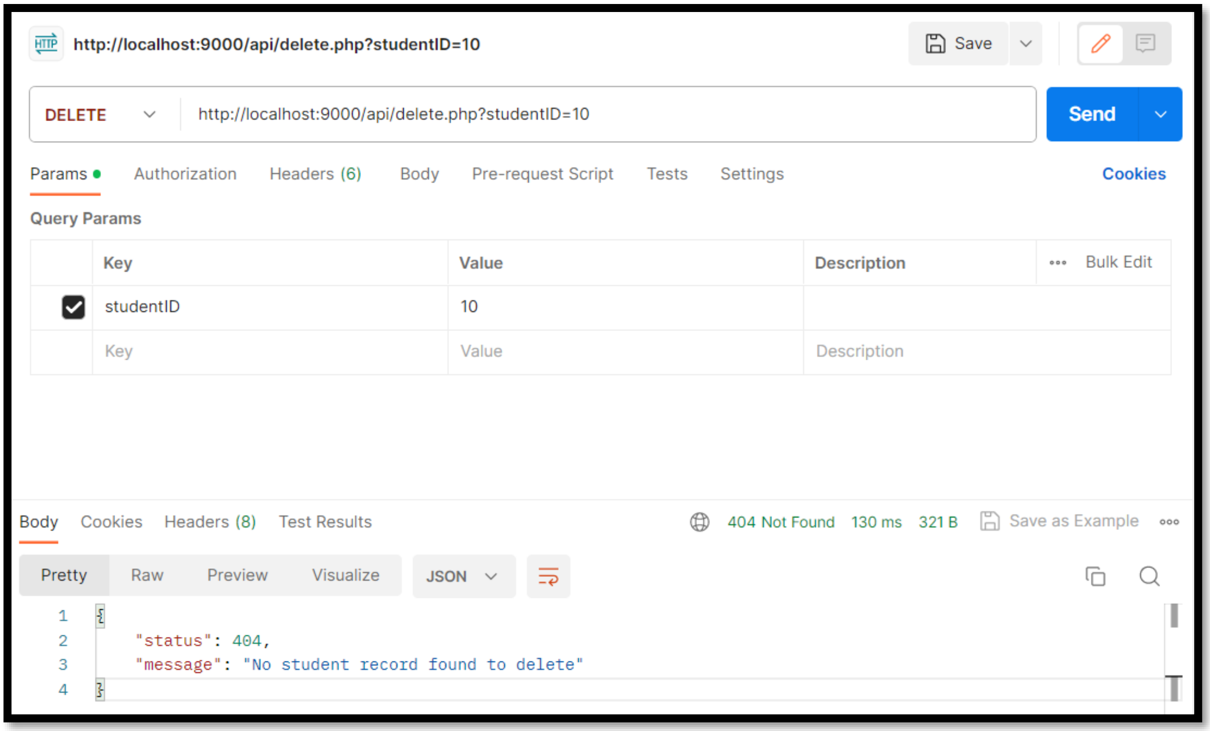
Query Params

	Key	Value	Description
☑	studentID	6	
	Key	Value	Description

405 Method Not Allowed 51 ms 324 B

```
{
    "status": 405,
    "message": "POST Invalid request method"
}
```

Figure 13 – Error Response: Invalid Method POST.



DELETE http://localhost:9000/api/delete.php?studentID=10

	Key	Value	Description
☑	studentID	10	
	Key	Value	Description

404 Not Found 130 ms 321 B

```
{
    "status": 404,
    "message": "No student record found to delete"
}
```

Figure 14 – Error Response: Invalid studentID.

Note: Since we are using the student ID as a primary key in our database, it means that each student ID must be unique. When try to add a new student with an ID that already exists in the database, the database management system will prevent the insertion because it violates the uniqueness constraint imposed by the primary key.