API Assignment Testing Report

Student Name:	[Your Name]
Student ID:	[Your ID]
Project Name:	Books Management CRUD API
Date:	February 13, 2026
Base URL:	http://169.239.251.102:280/~your_username/books-api/actions

Project Overview

This report documents the testing of a RESTful CRUD API built with PHP and MySQL. The API implements all required operations: Create, Read (All and One), Update, and Delete. All endpoints return JSON responses and are located in the 'actions/' subfolder as required.

Test 1: Read All Records

Endpoint: read_all.php

Method: GET

Purpose: Retrieve all books from the database with their IDs.

curl Command:

```
curl "http://169.239.251.102:280/~your_username/books-api/actions/read_all.php"
```

Expected Response Format:

```
{"success": true, "data": [{"id": 1, "name": "...", "phone": "..."}]}
```

Screenshot:

[INSERT SCREENSHOT HERE - showing curl command and JSON response]

Test 2: Read One Record

Endpoint: read_one.php

Method: GET

Purpose: Retrieve a single book by its ID.

curl Command:

```
curl
"http://169.239.251.102:280/~your_username/books-api/actions/read_one.php?id=1"
```

Expected Response Format:

```
{"success": true, "data": {"id": 1, "name": "...", "phone": "..."}}
```

Screenshot:

[INSERT SCREENSHOT HERE - showing curl command and JSON response]

Test 3: Create Record

Endpoint: create.php

Method: POST

Purpose: Insert a new book into the database.

curl Command:

```
curl -X POST
"http://169.239.251.102:280/~your_username/books-api/actions/create.php" \ -d
"name=New Book" \ -d "phone=0240555555"
```

Expected Response Format:

```
{"success": true, "data": {"id": 10}}
```

Screenshot 1 - Create:

[INSERT SCREENSHOT HERE - showing create command and response]

Verification - Read All:

```
curl "http://169.239.251.102:280/~your_username/books-api/actions/read_all.php"
```

Screenshot 2 - Verification:

[INSERT SCREENSHOT HERE - showing read_all with new record]

Test 4: Update Record

Endpoint: update.php

Method: POST

Purpose: Update an existing book's information.

curl Command:

```
curl -X POST
"http://169.239.251.102:280/~your_username/books-api/actions/update.php" \ -d
"id=1" \ -d "name=Updated Book" \ -d "phone=0240999999"
```

Expected Response Format:

```
{"success": true}
```

Screenshot 1 - Update:

[INSERT SCREENSHOT HERE - showing update command and response]

Verification - Read All:

```
curl "http://169.239.251.102:280/~your_username/books-api/actions/read_all.php"
```

Screenshot 2 - Verification:

[INSERT SCREENSHOT HERE - showing read_all with updated record]

Test 5: Delete Record

Endpoint: delete.php

Method: POST

Purpose: Remove a book from the database.

curl Command:

```
curl -X POST
"http://169.239.251.102:280/~your_username/books-api/actions/delete.php" \ -d
"id=1"
```

Expected Response Format:

```
{"success": true}
```

Screenshot 1 - Delete:

[INSERT SCREENSHOT HERE - showing delete command and response]

Verification - Read All:

```
curl "http://169.239.251.102:280/~your_username/books-api/actions/read_all.php"
```

Screenshot 2 - Verification:

[INSERT SCREENSHOT HERE - showing read_all without deleted record]

Conclusion

All CRUD operations have been successfully implemented and tested. The API correctly handles Create, Read (All and One), Update, and Delete operations, returning JSON responses as required. All endpoints are accessible via the 'actions/' subfolder, and the database integration works as expected.

Requirements Checklist:

- ✓ Read All endpoint implemented with record IDs
- ✓ Read One endpoint implemented
- ✓ Create endpoint implemented
- ✓ Update endpoint implemented
- ✓ Delete endpoint implemented
- ✓ MySQL database used (no static data)
- ✓ All endpoints in 'actions/' subfolder
- ✓ All responses in JSON format
- ✓ Error responses in JSON format