

Bug Report-API

Project: ITQA Module –Group -29

Bug ID: BUG_API_001

Date Reported: 20/12/2024

Reported by: Mr Jeewantha Kodikara

Assigned to: 204037E - Dharmawanshe H.S.M

Status: New

Priority: High

Severity: Critical

Component: API

Summary: Unauthorized user able to delete books

Description:

In the "Delete Book" API, users with valid user credentials are incorrectly authorized to delete books. When a DELETE request is sent with a valid book ID, the API returns a 200 OK status code instead of the expected 403 Forbidden, allowing unauthorized users to perform delete actions. This is a major security flaw.

Project: ITQA Module –Group -29

Bug ID: BUG _ API _002

Date Reported: 20/12/2024

Reported by: Mr Jeewantha Kodikara

Assigned to: 204037E - Dharmawanshe H.S.M

Status: New

Priority: Medium

Severity: Minor

Component: API

Summary: Incorrect status code for non-existing book deletion by a user

Description:

When a user who doesn't possess the permission to perform delete operations attempts to delete a non-existing book by providing a non-existing book ID, the API returns a 404 Not Found status code instead of the expected 403 Forbidden. Only authorized persons are allowed to perform any action and since this user doesn't have the permission he/she shouldn't be allowed into the system. This mismatch leads to incorrect error handling for unauthorized users.

Project: ITQA Module –Group -29

Bug ID: BUG _ API _003

Date Reported: 20/12/2024

Reported by: Mr Jeewantha Kodikara

Assigned to: 204037E - Dharmawanshe H.S.M

Status: New

Priority: High

Severity: Major

Component: API

Summary: Admin unauthorized to delete existing books

Description:

The "Delete Book" API fails to authorize admins to delete valid existing books. When an admin sends a DELETE request with valid admin credentials and a valid book ID, the API incorrectly responds with a 403 Forbidden status code instead of the expected 200 OK, preventing the intended operation even the user is authorized and using valid book details.

Project: ITQA Module –Group -29

Bug ID: BUG _ API _004

Date Reported: 20/12/2024

Reported by: Mr Jeewantha Kodikara

Assigned to: 204037E - Dharmawanshe H.S.M

Status: New

Priority: Medium

Severity: Minor

Component: API

Summary: Admin unauthorized to delete non-existing books

Description:

The "Delete Book" API incorrectly denies admin users permission to delete non-existing books. When a DELETE request is sent with valid admin credentials and a valid but non-existing book ID, the API returns a 403 Forbidden status code instead of the expected 404 Not Found.

Project: ITQA Module –Group -29

Bug ID: BUG _ API _005

Date Reported: 20/12/2024

Reported by: Mr Jeewantha Kodikara

Assigned to: 204136G - Nedungamuwa P.D.M.C.C.B

Status: New

Priority: High

Severity: Major

Component: API

Summary: Invalid data type for "author" is accepted

Description:

When an admin attempts to add a book with an integer as the author in the POST request to the "/api/books" endpoint, the API incorrectly accepts the request and returns a 201 Created status code instead of the expected 400 Bad Request. This allows invalid data to be stored in the system, violating data validation rules.

Project: ITQA Module –Group -29

Bug ID: BUG _ API _006

Date Reported: 20/12/2024

Reported by: Mr Jeewantha Kodikara

Assigned to: 204136G - Nedungamuwa P.D.M.C.C.B

Status: New

Priority: Medium

Severity: Minor

Component: API

Summary: Unexpected attributes in request body accepted with incorrect status code

Description: The "Create Book" API fails to validate the presence of required fields for new book entries. When an admin sends a POST request to the "/api/books" endpoint with unexpected attributes. The API incorrectly responds with a 201 Created status code instead of the expected 400 Bad Request, allowing incomplete and invalid data to be stored in the database.

Project: ITQA Module –Group -29

Bug ID: BUG _ API _007

Date Reported: 20/12/2024

Reported by: Mr Jeewantha Kodikara

Assigned to: 204136G - Nedungamuwa P.D.M.C.C.B

Status: New

Priority: High

Severity: Major

Component: API

Summary: Incorrect error message for missing required fields

Description:

When an admin sends a POST request to the "/api/books" endpoint with missing values for required fields (title and author), the API responds with a 201 Created status code instead of the expected 400 Bad Request

Project: ITQA Module –Group -29

Bug ID: BUG _ API _008

Date Reported: 20/12/2024

Reported by: Mr Jeewantha Kodikara

Assigned to: 204049R - Dissanayake J.W.

Status: New

Priority: High

Severity: Major

Component: API

Summary: Empty strings accepted for "title" and "author" fields

Description:

When a PUT request is sent to update a book with an id using empty strings for the title and author fields, the API responds with a 200 OK status code instead of the expected 400 Bad Request. This allows invalid data to be stored in the system, bypassing validation checks.

Project: ITQA Module –Group -29

Bug ID: BUG _ API _009

Date Reported: 20/12/2024

Reported by: Mr Jeewantha Kodikara

Assigned to: 204049R - Dissanayake J.W.

Status: New

Priority: High

Severity: Major

Component: API

Summary: Numeric values accepted in "title" and "author" fields

Description:

When a PUT request is sent to update a book with an id using numeric values for the title and author fields, the API incorrectly responds with a 200 OK status code instead of the expected 400 Bad Request. This behavior allows invalid data types to be accepted, compromising data integrity.

Project: ITQA Module –Group -29

Bug ID: BUG _ API _010

Date Reported: 20/12/2024

Reported by: Mr Jeewantha Kodikara

Assigned to: 204061V - Githmi P.H.

Status: New

Priority: Medium

Severity: Major

Component: API

Summary: Unauthorized access error for valid user fetching a book by ID

Description:

When a valid user sends a GET request to the "/api/books/{id}" endpoint to fetch a book by ID, the API responds with a 403 Forbidden status code instead of the expected 200 OK (if the book exists) or 404 Not Found (if the book does not exist). This bug prevents valid users from accessing book details and disrupts normal application behavior.