|  |  |
| --- | --- |
| Leah Rampono | 20231562 |
| Guruprasaath Gopalakrishnan | 20231284 |
| Abinash Vibin | 20231246 |

Abstract

[Draw your reader in with an engaging abstract. It is typically a short summary of the document.   
When you’re ready to add your content, just click here and start typing.]

test plan

[Document subtitle]

Table of Contents

[Test Plan Overview 2](#_Toc201843935)

[Objectives 2](#_Toc201843936)

[Test Scope 2](#_Toc201843937)

[Out of Scope 3](#_Toc201843938)

[Test Strategy 3](#_Toc201843939)

[Test Environment 3](#_Toc201843940)

[Test Cases 4](#_Toc201843941)

[Test Summary 6](#_Toc201843942)

[Overview of Testing 6](#_Toc201843943)

[Test Execution Summary 6](#_Toc201843944)

[Key Findings & Defects Found 6](#_Toc201843945)

# Test Plan Overview

This test plan has been designed to ensure the functionality and the reliability of the as well as the accuracy of the CRUD operations for the Art, Mathematics, and Technology collections controllers within the API. By testing all the endpoints, we will be able to identify any potential bugs and ensure that the API behaves as it is expected to. The goal of testing is to be able to deliver a backend service that functions correctly and can be integrated with the frontend of the application.

# Objectives

* **Test all CRUD operations:**  
   Verify that Create, Read, Update, and Delete functions work correctly and perform the expected actions in the application.
* **Check data persistence in MongoDB:**  
   Ensure that all data changes are correctly saved in the MongoDB database and remain consistent across sessions.
* **Verify response structure and status codes:**  
   Confirm that each API endpoint returns the correct HTTP status codes (e.g., 200, 201, 400) and structured JSON responses.
* **Test error handling:**  
   Check that the system handles invalid inputs and errors gracefully, showing helpful messages without crashing or corrupting data.

# Test Scope

**Features that will be tested within the API:**

* Post/ API/ [Art, Technology,Mathematics]
* Get/ API/ [ Art, Technology,Mathematics]
* Put/API/ [ Art, Technology,Mathematics]
* Delete/API/ [ Art, Technology,Mathematics]

## Out of Scope

out of scope for this test plan document is the following:

* UI and integration testing for the mobile application
* Role based access or authentication features (firebase authentication, implemented in frontend).

# Test Strategy

Manual testing will be conducted using Postman to verify the behaviour of all API endpoints. Each CRUD operation (Create, Read, Update, Delete) will be tested individually to ensure it performs as expected.

The testing strategy will include:

* Sending requests with valid and invalid input data.
* Verifying database updates through MongoDB Atlas.
* Checking that proper HTTP status codes and response structures are returned.

# Test Environment

* Backend: Node.js with Express
* Database: MongoDB
* Testing Tool: Postman

# Test Cases

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test ID** | **Feature** | **Description** | **Endpoints** | **Method** | **Expected Result** | **Pass / Fail** |
| TC01 | Create Article | Add new article to art | /API/art | POST | 200 OK  Request successful | Pass |
| TC02 | Create Article | Add new article to Mathematics | /API/mathematics | POST | 200 OK  Request successful | Pass |
| TC03 | Create Article | Add new article to Technology | /API/technology | POST | 200 OK  Request successful | Pass |
| TC04 | Read Articles | Get all art articles | /API/art | GET | 200 OK  Request successful | Pass |
| TC05 | Read Articles | Get all mathematics articles | /API/mathematics | GET | 200 OK  Request successful | Pass |
| TC06 | Read Articles | Get all technology articles | /API/technology | GET | 200 OK  Request successful | Pass |
| TC07 | Update article | Update art Article by id | /API/art/:id | PUT | 200 OK  Request successful | Pass |
| TC08 | Update article | Update mathematics Article by id | /API/ mathematics /:id | PUT | 200 OK  Request successful | Pass |
| TC09 | Update article | Update technology Article by id | /API/ technology /:id | PUT | 200 OK  Request successful | Pass |
| TC10 | Delete Article | Delete art Article by id | /API/art/:id | DELETE | 200 OK  Request successful | Pass |
| TC11 | Delete Article | Delete mathematics Article by id | /API/ mathematics /:id | DELETE | 200 OK  Request successful | Pass |
| TC12 | Delete Article | Delete technology Article by id | /API/ technology /:id | DELETE | 200 OK  Request successful | Pass |
| TC13 | Error Handling | Create with invalid data | /API/art/ | POST | 404 Not Found | Fail |
| TC14 | Error Handling | Update with invalid data | /API/ technology /: invalid id | PUT | 500 Internal Server Error | Pass |
| TC15 | Error Handling | Delete with non-existent ID | / API/ mathematics /: invalid id | DELETE | 500 Internal Server Error | Pass |

# Test Summary

**Author:** Guruprasaath

**Date:** 26/06/2025

**Testing Period**: June 25, 2025 – June 26, 2025

## Overview of Testing

This test summary summarizes the results of the testing that has been conducted on the backend-api. The main objective is to validate the functionality of the API and identify any defects. Manual testing through postman was used to uncover bugs within the API.

## Test Execution Summary

|  |  |  |
| --- | --- | --- |
| **Total Test Cases** | **Passed** | **Failed** |
| 15 | 14 | 1 |

## Key Findings & Defects Found

Medium-Priority Defect:

**BUG\_TC13- Invalid Data Error Handling**

* **Issue:** the API accepts invalid or incomplete data without returning a proper validation error.
* **Impact:** This leads to poor data integrity and potential frontend issues across all article types.
* **Severity:** Medium
* **Steps to reproduce:** 
  1. Open Postman and create a POST request to: http://localhost:5000/api/art
  2. In the Body, select raw → JSON and enter:

{

"wrongField": "This should fail"

}

* 1. Click Send
  2. Observe the API response:

**Expected:** 500 Internal Server Error, Failed to create article

**Actual:** Respond with 200 OK