**Quizify – API Testing Project (Test Plan Document)**

**1. Project Overview**

**Quizify** is an Online Quiz and Certification System developed using the **MERN stack** (MongoDB, Express, React, Node.js).

The purpose of this project is to perform **API Testing** on its live backend using **Postman** to verify the functionality, reliability, security, and performance of all REST endpoints.

**2. Objectives**

* **Validate** that all API endpoints return expected responses.
* **Ensure** authentication & authorization work correctly with **JWT** (JSON Web Tokens).
* **Check** data integrity for **CRUD** (Create, Read, Update, Delete) operations.
* **Perform** positive, negative, and security testing.
* **Automate** checks using Postman test scripts and environment variables.
* **Generate** professional reports and maintain test documentation.

**3. Scope of Testing**

**In Scope:**

* **Authentication APIs** (Register, Login, Reset Password)
* **User Profile APIs** (Get, Update)
* **Quiz APIs** (List Quizzes, Get Questions)
* **Result APIs** (Submit, Leaderboard, History)
* **Admin APIs** (Users, Questions, Stats)
* **Advanced Tests**: CRUD, Negative Scenarios, API Chaining

**Out of Scope:**

* Frontend UI testing
* Database performance tuning
* Load testing beyond Postman’s basic runner

**4. Test Environment**

|  |  |
| --- | --- |
| Component | Details |
| **Base URL** | https://quiziify-backend.onrender.com/api |
| **Frontend Reference** | https://quizify-phi.vercel.app |
| **Database** | **MongoDB Atlas** |
| **Hosting** | Backend – Render / Frontend – Vercel |
| **Tools Used** | **Postman v10+**, Newman, Excel/Sheets |
| **Auth Type** | **JWT Bearer Token** |
| **Environments** | Local / Production (configurable via Postman) |

**5. Testing Types Applied**

|  |  |
| --- | --- |
| Type | Description |
| **Functional Testing** | Verify endpoints behave as expected |
| **Regression Testing** | Ensure new changes don’t break existing APIs |
| **Security Testing** | Token validation, unauthorized access checks |
| **Negative Testing** | Invalid data, missing fields, wrong roles |
| **Data Validation** | JSON schema and field verification |
| **API Chaining** | Using data from one API in another |
| **Automation** | Postman tests executed via Collection Runner / Newman |

**6. Test Data**

* **Registered User** credentials
* **Admin** credentials (used for admin routes)
* **Dummy Quiz Categories** (e.g., Tech, Aptitude)
* Sample **questions & scores** for result validation

**7. Test Deliverables**

|  |  |
| --- | --- |
| Deliverable | Description |
| **Postman Collection** (.json) | All API requests + test scripts |
| **Environment File** (.json) | Base URL + tokens + dynamic variables |
| **Test Plan** (this doc) | Strategy and scope description |
| **Test Cases** (.xlsx) | Detailed API test cases with expected results |
| **Bug Report** (.xlsx) | Defects found during testing |
| **Final Test Report** (.docx) | Summary of results & coverage |

**8. Entry & Exit Criteria**

**Entry Criteria:**

* Backend server is **up and stable**.
* Valid user and admin accounts **available**.
* APIs documented and **accessible via Postman**.

**Exit Criteria:**

* All **critical API tests executed and passed**.
* **No high-severity defects** open.
* Test summary report **approved**.

**9. Test Metrics**

|  |  |
| --- | --- |
| Metric | Target |
| **Total APIs tested** | 20 + |
| **Functional Pass Rate** | ≥95% |
| **Negative Case Pass Rate** | ≥90% |
| **Automation Coverage** | 100% of functional flows |
| **Average Response Time** | <500 ms |

**10. Risks & Mitigations**

|  |  |
| --- | --- |
| Risk | Mitigation |
| **Server downtime** | Use mock or staging environment |
| **Token expiration** | Re-login and auto-update token via script |
| **Schema changes** | Regular sync with backend team |
| **Data dependency between APIs** | Apply API chaining and cleanup scripts |

**11. Roles & Responsibilities**

|  |  |
| --- | --- |
| Role | Responsibility |
| **QA Engineer (Furqanuddin)** | Design and execute API tests, create docs, report defects |
| **Developer (Team)** | Fix issues and update backend APIs |
| **Project Mentor** | Review test coverage and reports |

**12. Approval & Sign-off**

This Test Plan has been reviewed and approved by the QA Engineer and Project Mentor. Testing will be executed according to this strategy.

Prepared By: **Mohammad Furqanuddin**

Role: **QA Engineer (Trainee)**

Date: **October 2025**