**Test Planning for Student Registration System**

**1. Introduction**

The Student Registration System is designed to facilitate user registration, course enrollment, and payment processing. This test plan outlines the structured approach to validating the system’s functionality, security, and performance to ensure a seamless user experience. The objective is to identify and resolve defects before deployment, ensuring compliance with business requirements and system efficiency.

**2. Scope**

**In-Scope:**

* **Functional Testing**:
  + Registration form (e.g., input validation, error messages).
  + Payment processing via Stripe.
* **Integration Testing**:
  + Interaction between the registration module and Stripe API.
  + Interaction between the frontend, backend, and database.
* **System Testing**:
  + End-to-end testing of the entire system, including registration, payment, and database updates.
* **Database Testing**:
  + Ensure student and payment data is stored correctly.

**Out-of-scope:**

* **Security Testing**:
  + Validate secure handling of sensitive data (e.g., credit card information).
* **Performance Testing**:
  + Test system behavior under peak load conditions.
* **Usability Testing**:
  + Ensure a seamless and intuitive user experience.

**3. Test Objectives**

* Detect and resolve bugs or inconsistencies to maintain system stability.
* Ensure the system functions correctly, securely, and efficiently.
* Validate user registration, authentication, course enrollment, and payment processing work as expected.
* Test the integration between different modules of the system
* Test the application’s ability to handle errors
* Perform **system testing** to validate end-to-end functionality of the entire system.

**4. Test Deliverables**

* Test Plan Document
* Test Cases and Test Scripts
* Test Execution Reports
* Defect Reports
* Final Test Summary Report

**5. Test Strategy**

**5.1 Levels of Testing**

**5.1.1 Unit Testing**

* Focuses on testing individual components such as models, forms, and views if they function correctly alone.
* Ensures that each function and method performs as expected.

**5.1.2 Integration Testing**

* Validates the interaction between different system modules.
* Ensures that user registration, authentication, course enrollment, and Stripe payment processing work together seamlessly.
* Tests API calls, database interactions, and third-party integrations.

**5.1.3 System Testing**

* Evaluates the entire system to ensure compliance with functional and non-functional requirements.
* Conducts end-to-end testing of user workflows.
* Includes security, performance, and compatibility testing.

**5.1.4 Acceptance Testing(we’re not going to do that 😅)**

* Ensures the system meets business and user requirements.
* Conducts user acceptance testing (UAT) with stakeholders.
* Confirms that registration, course selection, and payment processing are user-friendly and error-free.

**5.2 Test Environments**

**5.2.1 Hardware:** my laptop

**5.2.2 Software**

* **Operating System**: Windows 10
* **Browsers**: Chrome, Brave
* **Database**: SQLite
* **Backend**: Python (Django framework)

**5.2.3 Tools**

* **Testing Framework**: Django Test Suite, Pytest
* **Version Control**: GitHub

**6. Entry and Exit Criteria**

**Entry Criteria**

* The system is developed and stable.
* Test cases and scripts are prepared and reviewed.
* Required test data is available.

**Exit Criteria**

* All test cases have been executed.
* All critical and major defects are resolved.
* Test execution report is reviewed and approved.

**7. Reasoning**

A well-defined test plan ensures a systematic approach to testing, reducing risks and improving system reliability. Testing will focus on preventing defects before deployment, ensuring users can seamlessly register, enroll in courses, and make payments.

**8. Schedule**

|  |  |  |
| --- | --- | --- |
| **Phase** | **Start Date** | **End Date** |
| Test Planning |  |  |
| Test Case Design |  |  |
| Test Execution |  |  |
| Final Test Report |  |  |

**9. Risk Management**

|  |  |  |
| --- | --- | --- |
| **Risk** | **Impact** | **Mitigation** |
| Delayed development completion | High | Align testing schedule with development progress |
| Critical defects during testing | High | Prioritize defect resolution and retesting before deployment |
| Inconsistent test environments | Medium | Maintain standardized test environments |
| Payment gateway issues | High | Perform multiple test transactions with different scenarios |

**10. Approval**

The test plan must be reviewed and approved by the following stakeholders:

* Project Manager
* QA Lead
* Development Lead
* Business Analyst