**Test Case Design for Student Registration System**

1. **Test Case Details**
   1. **Unit Testing (Using pytest - test\_views.py)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Title** | **Module/Feature** | **Author** | **Created** | **Updated** |
| TC-UNIT-001 | Validate Student Model | Models | QA Team |  |  |
| TC-UNIT-002 | Validate Student Registration Form | Forms | QA Team |  |  |
| TC-UNIT-003 | Validate Registration View | Views | QA Team |  |  |
| TC-UNIT-004 | Validate Payment View | Views | QA Team |  |  |
| TC-UNIT-005 | Validate Payment Success View | Views | QA Team |  |  |

* 1. **Integration testing**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Title** | **Module/Feature** | **Author** | **Created** | **Updated** |
| TC-INT-001 | Validate Registration Flow | Integration | QA Team |  |  |
| TC-INT-002 | Validate Payment Flow | Integration | QA Team |  |  |
| TC-INT-003 | Validate Payment Failure Flow | Integration | QA Team |  |  |

**1.3 System Testing**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Title** | **Module/Feature** | **Author** | **Created** | **Updated** |
| TC-SYS-001 | Validate Student Registration and Payment Flow | System testing | QA team |  |  |

1. **Test Case Description**
   1. **Unit Testing (Using pytest - test\_views.py)**

**TC-UNIT-001: Validate Student Model:**

* **Description**: Test the creation and string representation of the Student model.
* **Objectives:** Ensure student model stores and retrieves data correctly.
* **Preconditions:** Database is set up.
* **Dependencies:** Django ORM must be functional.

**TC-UNIT-002: Validate Student Registration Form**

* **Description:** Test the StudentRegistrationForm with valid data.
* **Objectives:** Ensure the form accepts valid inputs and rejects invalid ones.
* **Preconditions:** Registration page is accessible.
* **Dependencies:** Form validation must be implemented.

**TC-UNIT-003: Validate Registration View**

* **Description:** Test the registration view for creating a new student and redirecting to the payment page.
* **Objectives:** Ensure user registration is successful and redirects properly.
* **Preconditions:** Registration page is accessible.
* **Dependencies:** Database and form validation must be active.

**TC-UNIT-004: Validate Payment View**

* **Description:** Test the payment view to ensure it renders the payment page correctly.
* **Objectives:** Ensure payment page loads correctly.
* **Preconditions:** Student must be registered.
* **Dependencies:** Payment system must be integrated.

**TC-UNIT-005: Validate Payment Success View**

* **Description:** Test the payment\_success view to ensure it updates the payment\_status to True.
* **Objectives:** Ensure payment status updates correctly after successful payment.
* **Preconditions:** Student must attempt payment.
* **Dependencies:** Database update must work correctly.
  1. **Integration Testing**

**TC-INT-001: Validate Registration Flow**

* **Description:** Test the student registration flow: Submit the registration form, verify the student is saved to the database, and verify the user is redirected to the payment page.
* **Objectives:** Ensure the registration process completes successfully.
* **Preconditions:** System is running with all dependencies.
* **Dependencies:** Form validation, database, and redirection must function properly.

**TC-INT-002: Validate Payment Flow**

* **Description:** Test the payment flow: Mock Stripe's API to simulate a successful payment, verify the user is redirected to Stripe's checkout page, simulate a successful payment callback, and verify the payment status is updated in the database.
* **Objectives:** Ensure payment processing and redirection to Stripe works.
* **Preconditions:** A registered student must attempt payment.
* **Dependencies:** Stripe API integration must be functional.

**TC-INT-003: Validate Payment Failure Flow**

* **Description:** Mock Stripe's API to simulate a payment failure, verify the user is redirected to the payment failure page, and verify the payment status remains unchanged.
* **Objectives:** Ensure failed payments are handled correctly.
* **Preconditions:** Payment attempt must fail.
* **Dependencies:** Stripe API integration must be simulated for failure.
  1. **System Testing**
* **Description:** Test the complete student registration and payment process, ensuring proper redirection, data validation, and Stripe integration.
* **Objectives:**
* Ensure that students can successfully register.
* Verify redirection to the payment page.
* Validate successful payment processing via Stripe.
* Ensure the database updates payment status correctly.
* **Preconditions:**
* The registration page must be accessible.
* Stripe test mode must be enabled.
* The system must be running with all dependencies.
* **Dependencies:**
* Form validation, database, and payment integration must function properly.

1. **Test Case Steps**
   1. **Unit Testing (Using pytest - test\_views.py)**

**TC-UNIT-001: Validate Student Model**

|  |  |  |
| --- | --- | --- |
| **Step No** | **Action** | **Expected Result** |
| 1 | Create a Student object with test data | Object is created successfully |
| 2 | Retrieve student data from database | Data matches expected values |
| 3 | Validate string representation | Returns correct student name |

**TC-UNIT-002: Validate Student Registration Form**

|  |  |  |
| --- | --- | --- |
| **Step No** | **Action** | **Expected Result** |
| 1 | Submit valid registration data | Form is submitted successfully |
| 2 | Submit invalid registration data | Form shows validation errors |

**TC-UNIT-003: Validate Registration View**

|  |  |  |
| --- | --- | --- |
| **Step No** | **Action** | **Expected Result** |
| 1 | Navigate to the registration page | Registration form is displayed |
| 2 | Enter valid user details and submit | User is registered successfully |
| 3 | Verify user entry in the database | Student record is created |
| 4 | Ensure redirection to the payment page | User is redirected correctly |

**TC-UNIT-004: Validate Payment View**

|  |  |  |
| --- | --- | --- |
| **Step No** | **Action** | **Expected Result** |
| 1 | Navigate to the payment page with a valid student ID | Payment page loads successfully |
| 2 | Verify student details displayed | Correct student information is shown |
| 3 | Click on proceed to payment | User is redirected to Stripe |

**TC-UNIT-005: Validate Payment Success View**

|  |  |  |
| --- | --- | --- |
| **Step No** | **Action** | **Expected Result** |
| 1 | Complete payment through Stripe | Stripe processes the payment |
| 2 | Verify payment success page loads | Payment success page is displayed |
| 3 | Check database for updated payment status | Student’s payment status is updated to 'Paid' |

* 1. **Integration Testing**

**TC-INT-001: Validate Registration Flow**

|  |  |  |
| --- | --- | --- |
| **Step No** | **Action** | **Expected Result** |
| 1 | Navigate to the registration page | Registration form is displayed |
| 2 | Submit valid user details | User is registered and redirected to payment page |
| 3 | Complete payment through Stripe | Payment is processed successfully |
| 4 | Verify student entry in database | Student record exists and payment status is 'Paid' |

**TC-INT-002: Validate Payment Flow**

|  |  |  |
| --- | --- | --- |
| **Step No** | **Action** | **Expected Result** |
| 1 | Initiate payment for a registered student | Stripe checkout page loads |
| 2 | Complete the payment | Payment is successfully processed |
| 3 | Verify redirection to success page | User is redirected to payment success page |
| 4 | Confirm payment status update in database | Student payment status changes to 'Paid' |

**TC-INT-003: Validate Payment Failure Flow**

|  |  |  |
| --- | --- | --- |
| **Step No** | **Action** | **Expected Result** |
| 1 | Initiate payment with an invalid transaction | Stripe returns a failure response |
| 2 | Verify redirection to failure page | User is redirected to payment failure page |
| 3 | Check database for payment status | Payment status remains 'Unpaid' |

* 1. **System Testing**

|  |  |  |
| --- | --- | --- |
| **Step No.** | **Action** | **Expected Result** |
| 1 | Navigate to the registration page | The registration page loads successfully. |
| 2 | Enter valid student details (Name, Email, Course) | Fields accept input and allow form submission. |
| 3 | Click the "Register" button | User is redirected to the payment page. |
| 4 | Click the "Proceed to Payment" button | The Stripe payment page loads. |
| 5 | Enter Stripe test card details (4242 4242 4242 4242, Expiry: 12/28, CVC: 123) | Card details are entered correctly. |
| 6 | Click the "Pay" button | Payment is processed, and user is redirected to the success page. |
| 7 | Verify redirection to the "Payment Successful" page | The success page is displayed with a confirmation message. |
| 8 | Check the database for payment confirmation (if applicable) | Payment status is updated in the database. |

1. **Test Data**

|  |  |  |
| --- | --- | --- |
| **Field** | **Valid Input** | **Invalid Input** |
| Name | John Doe, Johnathan Christopher Doe the Second | (empty), 12345, @invalid\_name |
| Email | john@example.com, user.test@domain.com | invalid-email, user@.com, user@domain |
| Course | Python Programming, Django Web Development | (empty), 12345, @invalid\_course |
| Card Number | 4242 4242 4242 4242 (valid Stripe test card) | 1111 1111 1111 1111 (invalid) |
| CVC | 123 (valid) | 12 (invalid) |
| Expiration Date | 12/28 (future date) | 01/20 (expired date) |

1. **Expected and Actual Results**
   1. **Unit Testing (Using pytest - test\_views.py)**

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case ID | Expected Result | Actual Result | Pass/Fail |
| TC-UNIT-001 | Student object is created, stored, and retrieved correctly from the database | Student object created successfully and retrieved from database without errors | Passed |
| TC-UNIT-002 | Form successfully validates correct inputs and displays appropriate errors for invalid inputs | Form accepted valid inputs and showed correct validation errors for invalid inputs | Passed |
| TC-UNIT-003 | User successfully registers, entry is created in the database, and redirection occurs correctly | User was registered, data stored in database, and redirected without issues | Passed |
| TC-UNIT-004 | Payment page displays correctly, showing accurate payment details and student information | Payment page loaded successfully with correct student details | Passed |
| TC-UNIT-005 | Payment status updates to 'Paid' after successful Stripe transaction | Payment was processed successfully and status updated in database | Passed |

* 1. **Integration testing**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| TC-INT-001 | Complete registration flow works smoothly from form submission to database entry and redirection | Registration workflow executed correctly, with proper database entries and redirection | Passed |
| TC-INT-002 | Payment process completes successfully, and user is redirected to Stripe and back to success page | User was redirected to Stripe, payment processed, and returned to success page | Passed |
| TC-INT-003 | System detects failed payments and redirects user to the appropriate failure page | Payment failure was correctly detected, and user was redirected to failure page | Passed |

* 1. **System Testing**

|  |  |  |  |
| --- | --- | --- | --- |
| Step No. | Expected Result | Actual Result | Pass/Fail |
| 1 | The registration page loads successfully. | The registration page loaded successfully. | Pass |
| 2 | Fields accept input and allow form submission. | Fields accepted input and allowed form submission. | Pass |
| 3 | User is redirected to the payment page. | User was redirected to the payment page. | Pass |
| 4 | The Stripe payment page loads. | The Stripe payment page loaded successfully. | Pass |
| 5 | Card details are entered correctly. | Card details were not entered correctly. | Fail |
| 6 | Payment is processed, and user is redirected to the success page. |  |  |
| 7 | The success page is displayed with a confirmation message. |  |  |
| 8 | Payment status is updated in the database. |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |