**Unit Test Plan**

|  |  |
| --- | --- |
| Module #: | Application #: |
| Tester: Nguekam Raphael | Test Manager: Willy |
| **Module Overview** | |
| Describe the purpose of the module.  The purpose of these modules is to handle discount calculations and exchange rate retrievals. They provide functions to apply discounts based on percentage or fixed amount, generate referral codes, and fetch exchange rates using an external provider. | |
| **Module Inputs** | |
| Describe the inputs to the modules   * Discount Module:   + Discount code   + Percentage   + Minimum spend   + Current price   + Fixed discount amount * Exchange Rate Module:   + Currency code   + Callback function | |
| **Module Outputs** | |
| Describe the outputs from the module.   Discounted price based on discount code   Referral code   Exchange rate information | |
| **Logic Flow** | |
| Include a logic flow diagram.  (Include a flowchart here illustrating the interactions between functions like getDiscount, applyDiscount, calculatePercentageDiscount, calculateMoneyOff, and getExchangeRate.) | |
| **Test Data** | |
| List all test cases to be executed. | |
|  **TC\_POS\_01: Verify percentage discount calculation**   * **Test to be performed:** Call calculatePercentageDiscount with valid inputs. * **Expected Outcome:** The function should return the correct discounted price based on the percentage.    **TC\_POS\_02: Verify fixed amount discount calculation**   * **Test to be performed:** Call calculateMoneyOff with valid inputs. * **Expected Outcome:** The function should return the correct discounted price based on the fixed amount.    **TC\_POS\_03: Generate referral code**   * **Test to be performed:** Call generateReferralCode with a valid user ID. * **Expected Outcome:** The function should return a correctly formatted referral code.    **TC\_POS\_04: Apply a valid discount code**   * **Test to be performed:** Call applyDiscount with a valid discount code and current total. * **Expected Outcome:** The function should return the discounted price based on the discount code.    **TC\_POS\_05: Get exchange rate**   * **Test to be performed:** Call getExchangeRate with a valid currency code and callback function. * **Expected Outcome:** The function should return the correct exchange rate information via the callback. | |
| Number each test case. Indicate the test to be performed and expected outcome | |
| Negative Test Cases   1. **TC\_NEG\_01: Apply an invalid discount code**    * **Test to be performed:** Call applyDiscount with an invalid discount code.    * **Expected Outcome:** The function should return the original total without any discount. 2. **TC\_NEG\_02: Calculate discount below minimum spend**    * **Test to be performed:** Call calculatePercentageDiscount and calculateMoneyOff with current price below minimum spend.    * **Expected Outcome:** The functions should return the original current price. 3. **TC\_NEG\_03: Get exchange rate for unsupported currency**    * **Test to be performed:** Call getExchangeRate with an unsupported currency code.    * **Expected Outcome:** The function should return an error or null via the callback. | |
| Listin valid data selections | |
| Interface Modules | |
|  **Output data:** Discounted prices, Exchange rate information   **Data input:** Discount code, Percentage, Minimum spend, Current price, Fixed discount amount, Currency code, Callback function   **Internal program interface:** Function calls within the modules   **External program interface:** API calls to external services for discount validation and exchange rate retrieval | |
| Test Tools   * **Unit Testing Framework:** Jest * **Mocking Library:** axios-mock-adapter (for mocking axios calls) | |
| Identify software used for unit testing.  Identify names and locations of software for future regression testing.   * **Regression Testing Framework:** Cypress (located in tests/regression directory) | |
| Archive Plan | |
| Specify location of archived data.  Define procedures required to obtain access to this data.  **Archive Location:** /mnt/data/archived\_tests   * Access can be requested through the project manager. Archived data will be accessible through a version control system. | |
| Updates | |
| Identify how unit test plan will be updated.  The unit test plan will be updated whenever new features are added or changes are made to the existing functionality. Updates will be reviewed and approved by the test manager before implementation. | |