**Module 1: User Authentication**

**Test Script Overview**

* **Script Name:** user\_authentication\_test.jmx
* **Description:** This script tests the navigation and login functionality of the flight booking application under load.

**Test Configuration**

* **Thread Group:**
  + **Number of Threads (Users):** 10
  + **Ramp-Up Period:** 2 seconds
  + **Loop Count:** Infinite
  + **Runtime duration:** 300 Seconds

**HTTP Request Configuration**

* **HTTP Request Defaults:**
  + **Server Name or IP:** [server\_ip]
  + **Port Number:** 8888 (or the relevant port)
* **HTTP Request:**
  + **Path:** /login
  + **Method:** POST

**Assertions**

* **Response Assertion:**
  + **Field to Test:** Response Text
  + **Patterns to Test:** “Navigation Successful”
  + **source:** Boston
  + **destination:** New York

**Listeners**

* **View Results Tree**
* **Aggregate Report**
* **Graph Result**

**Module 2: Flight Search**

**Test Script Overview**

* **Script Name:** flight\_search\_test.jmx
* **Description:** This script tests the flight search functionality under load.

**Test Configuration**

* **Thread Group:**
  + **Number of Threads (Users):** 10
  + **Ramp-Up Period:** 2 seconds
  + **Loop Count:** Infinite
  + **Runtime duration:** 300 Seconds

**HTTP Request Configuration**

* **HTTP Request Defaults:**
  + **Server Name or IP:** [server\_ip]
  + **Port Number:** 8888 (or the relevant port)
* **HTTP Request:**
  + **Path:** /search
  + **Method:** GET
  + **Parameters:**
    - **Choosing the flights:** United Airlines
    - **date:** 31.05.2024

**Assertions**

* **Response Assertion:**
  + **Field to Test:** Response Code
  + **Pattern Matching Rules:** Equals
  + **Patterns to Test:** 200

**Listeners**

* **View Results Tree**
* **Aggregate Report**
* **Graph Result**

**Module 3: Booking Process**

**Test Script Overview**

* **Script Name:** booking\_process\_test.jmx
* **Description:** This script tests the booking process, including adding items to the cart, navigating to checkout, entering payment details, and confirming the purchase.

**Test Configuration**

* **Thread Group:**
  + **Number of Threads (Users):** 10
  + **Ramp-Up Period:** 2 seconds
  + **Loop Count:** Infinite
  + **Runtime duration:** 300 Seconds

**HTTP Request Configuration**

* **HTTP Request Defaults:**
  + **Server Name or IP:** [server\_ip]
  + **Port Number:** 8888
* **HTTP Requests:**
  + **Add to Cart:**
    - **Path:** /passenger details/payment page
    - **Method:** POST
    - **Parameters:**
      * **quantity:** 1
      * inputName sam
      * address s2, city center, brooklyn, newyork
      * city Brooklyn
      * state Newyork
      * zipCode 620000
  + **Navigate to Checkout:**
    - **Path:** /checkout
    - **Method:** GET
  + **Enter Payment Details:**
    - **Path:** /checkout/payment
    - **Method:** POST
    - **Parameters:**
      * cardType visa
      * creditCardNumber 4565135512656
      * creditCardMonth 11
      * creditCardYear 2027
      * CVV 655
      * nameOnCard sam
  + **Confirm Booking:**
    - **Path:** /checkout/confirm
    - **Method:** POST

**Assertions**

* **Response Assertion:**
  + **Field to Test:** Response Text
  + **Pattern Matching Rules:** Contains
  + **Patterns to Test:** "Booking confirmed"

**Listeners**

* **View Results Tree**
* **Aggregate Report**
* **Graph Result**

**Instructions for Running the Tests**

1. **Clone the Repository:**
   * Clone the GitHub repository containing the test scripts.
2. **Open JMeter:**
   * Open Apache JMeter and load the respective **.jmx** files for each module.
3. **Configure the Environment:**
   * Ensure the server IP, port, and other parameters are correctly set in the HTTP Request Defaults.
4. **Run the Tests:**
   * Execute the tests by clicking the green start button in JMeter.
5. **Monitor and Analyze Results:**
   * Use listeners like View Results Tree and Summary Report to monitor and analyze the test execution.