**Research on Testing Methodologies**

**1. Manual Testing vs Automated Testing**

***Manual Testing***

* Human tester checks the app step by step without scripts.
* Example: You log in as an organizer, click “Create Event,” fill in the details, and check if the event shows up.
* Best for: Exploratory testing, UI/UX checks, small projects.

***Automated Testing***

* Computer scripts/tools automatically test features.
* Example: Using **Selenium**, you write a script that opens your platform, clicks “Create Event,” fills in fields, and submits it.
* Best for: Repeated tests, regression testing, large projects.

**2. Functional vs Non-Functional Testing**

***Functional Testing = Tests what the system should do.***

* Example: In your project, when a user clicks “Register for Event,” the system should add their name to the attendee list.
* Questions answered: *Does this button/feature work correctly?*

***Non-Functional Testing = Tests how well the system works.***

* Example: 200 users try to register for the same event at the same time → system must handle the load.
* Other checks: performance, usability, security.

**3. Regression Testing**

* Definition: After adding a new feature, check if the old features still work.
* Example: After you add the “Filter Events” function, test again if “Create Event” and “Register for Event” still work as before.

**4. User Acceptance Testing (UAT)**

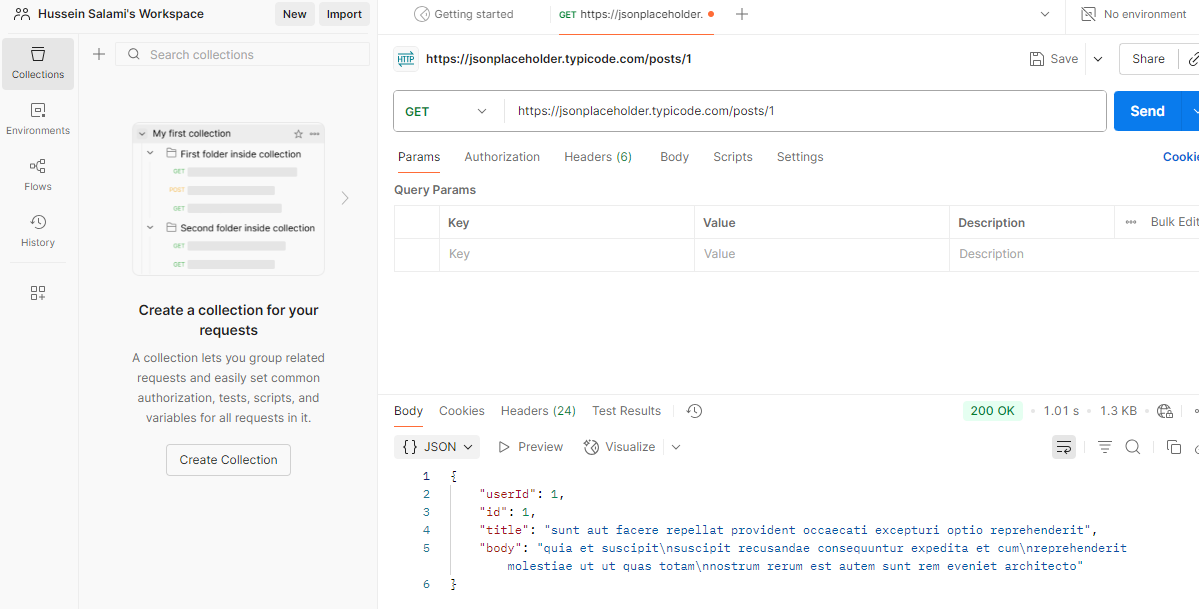
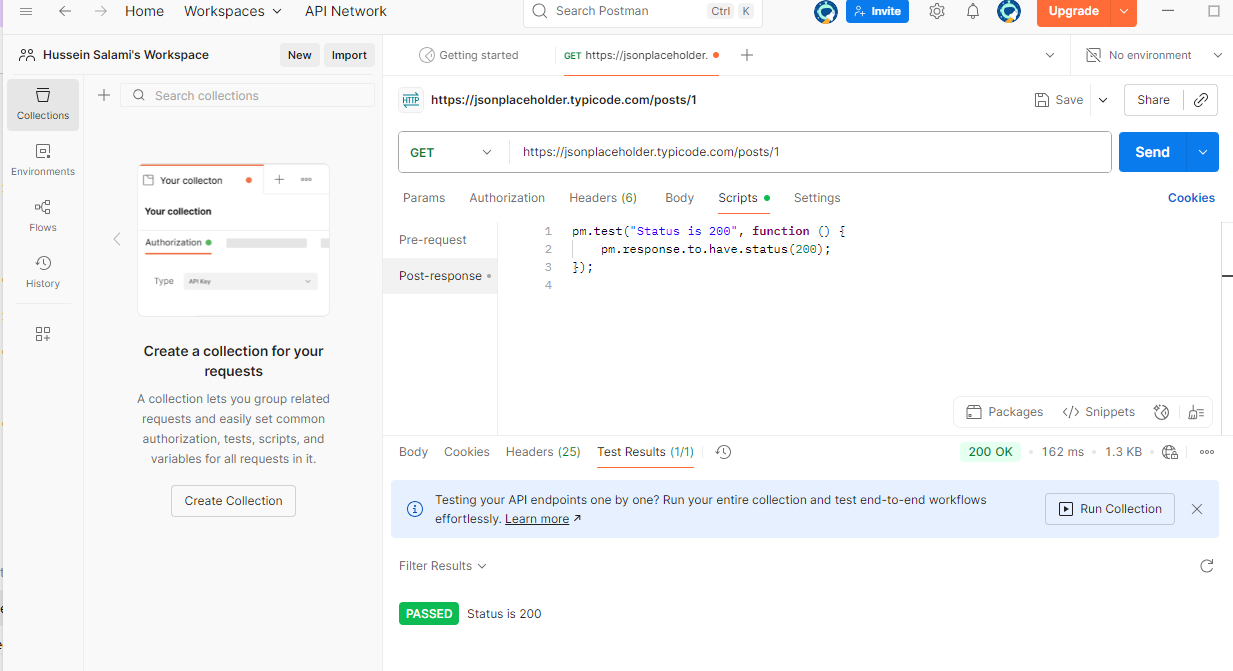
* Definition: Final testing by end-users (real people) before going live.
* Example: Invite 3–4 Darwin residents to test the app. Ask them: “Can you find and register for a cultural event easily?” If they can, UAT passes.

**Research & Write Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test ID | User story | Scenario | Preconditions | steps | Expected result | Priority |
| P-TC-001 | |  | | --- | |  |  |  | | --- | | Event Discovery | | Browse then Filter (Family + Weekend) then View Details | At least 3 events exist with different categories/dates | 1) Open **Events** page  2) Set **Category=Family**, **Date=Weekend**  3) Click an event card to open details | List only shows matching events; clicking opens details with title, time (local), location/map, description | **P1** |
| P-TC-002 | |  | | --- | |  |  |  | | --- | | Event Submission | | Organizer tries to create an event with a **blank title** | |  | | --- | | Organizer logged in |  |  | | --- | |  | | |  | | --- | | 1) Click Create Event  2) Leave Title empty  3) Fill other fields  4) Submit | | Error “**Title is required**”; event is not saved; user remains on form | P1 |
| P-TC-003 | |  | | --- | | Save/Share |  |  | | --- | |  | | |  | | --- | | Add event to calendar (Google/.ics) |  |  | | --- | |  | | |  | | --- | | Event is public; user in browser session |  |  | | --- | |  | | 1) Open event details  2) Click **Add to Calendar**  3) Choose Google or download .ics  4) Confirm | Calendar entry created with correct title, start/end time, and location | P2 |
| US-TC-001 | Organizer submits cultural events | Create event with valid information | |  | | --- | | Organizer verified & logged in |  |  | | --- | |  | | 1) Click **Create Event**  2) Fill required fields (title, date in future, venue, category)  3) **Submit** | Event status = **Submitted**; appears in **Admin Review** queue; organizer sees success message/notification | P1 |
| US-TC-002 | Attendee browses & filters events | No-match filters show friendly message | Events exist but none match selected filters | 1) Open **Events**  2) Apply Category=Nightlife + Date=Past 7 days | Message “**No events match your filters**” + **Clear filters** button displayed; no incorrect results | P2 |
| US-TC-003 | Admin reviews submitted events | Admin rejects event with a reason | At least one event in **Submitted** state | 1) Login as **Admin**  2) Open **Review Queue**  3) Open event  4) Click **Reject**, enter reason, **Confirm** | Status becomes **Rejected**; organizer receives rejection reason; event is **not public** | P1 |

**Learn Automation Tool**

Installed Postman:



Completed Course about Postman: [file:///C:/Users/hp/Downloads/CertificateOfCompletion\_Introducing%20Postman.pdf](C:\\Users\\hp\\Downloads\\CertificateOfCompletion_Introducing Postman.pdf)

**Identify Testing Tools**

**Postman**

* What it is: A tool to check if APIs (the “back-end connections”) are working.
* What it does: You can send requests like “give me all events” and see the reply from the system.
* Simple example: Type the event API link in Postman → click Send → it shows event details in JSON format.

**Selenium**

* What it is: A tool to control a web browser automatically.
* What it does: You write small scripts that click buttons, type in forms, and check results.
* Simple example: Selenium opens Chrome, logs in as an organizer, fills out the event form, clicks submit, and checks that the event was created.

**JMeter**

* What it is: A tool to test performance (speed under heavy load).
* What it does: Pretends to be many users using the system at the same time.
* Simple example: You set 200 fake users in JMeter → all apply filters at once → JMeter measures how fast the system responds.

**Fiddler**

* What it is: A tool to watch and record network traffic (the messages going between your computer and the server).
* What it does: Helps you see what’s sent and what’s received.
* Simple example: You click “Register for Event” → Fiddler shows the exact request your browser sent and the server’s reply.

**Cypress**

* What it is: A modern tool for testing websites from start to finish.
* What it does: Runs in the browser and checks that pages, buttons, and flows work correctly.
* Simple example: Cypress opens your site, applies “Family Weekend” filter, and checks that only Family events show in the list.