**Written Test Cases in Manual Testing**

**TEST CASE FOR LOGIN/REGISTER PAGE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Expected Output** | **Actual Output** | **Tester Name** | **Date** |
| To validate register details of the user or chef. | To be directed to the home page. | An error message of login unsuccessful | Muhib Mohammed | 04-11-2023 |
| To validate login details of the user. | To be directed to the home page. | Display home page. | Sirajul Hasan Syed | 04-11-2023 |

**5.1.2 TEST CASE FOR HOME PAGE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Expected Output | Actual Output | Tester Name | Date |
| To check the interface is linked between the home page and Chef/user Profile page. | Redirect to profile page successfully. | Directed to the wrong page. | Muhib Mohammed | 10-11-2023 |
| To check the interface is linked between the home page and Chef/user Profile page. | Redirect to profile page successfully. | Directed successfully. | Sirajul Hasan Syed | 10-11-2023 |

**5.1.3 TEST CASE FOR PROFILE PAGE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Expected Output | Actual Output | Tester Name | Date |
| To edit profile. | Edit page displayed. | Page not displayed. | Sirajul Hasan Syed | 11-11-2023 |
| To edit profile. | Edit page displayed. | Page displayed successfully and profile is edited successfully. | Muhib Mohammed | 11-11-2023 |

**5.1.4 TEST CASE FOR MENU PAGE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Expected Output | Actual Output | Tester Name | Date |
| To search meal on menu page | List of meals displayed. | Error meals not found | Sirajul Hasan Syed | 15-11-2023 |
| To search meal on menu page | List of meals displayed. | Page displayed successfully and list of wide appears | Muhib Mohammed | 15-11-2023 |
| Order track and meals status | Notify like meal  order,preparing | Displayed successfully order tracking | Sirajul Hasan Syed | 15-11-2023 |

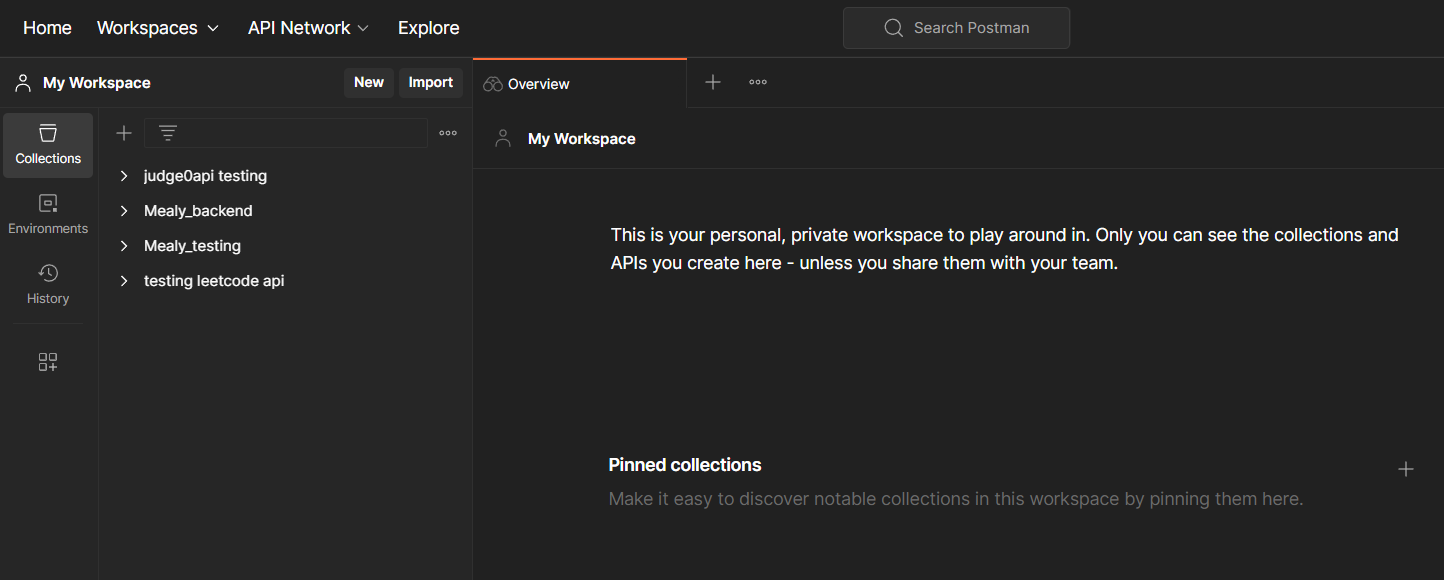
**AUTOMATED TESTING**

**Automated Testing in Mealy Using Postman**

**Overview:**

**- Objective:** Validate Homey Feasts's API functionality and ensure seamless integration.

**- Tool Used:** Postman for API testing.



**Test Cases:**

   - Login API:

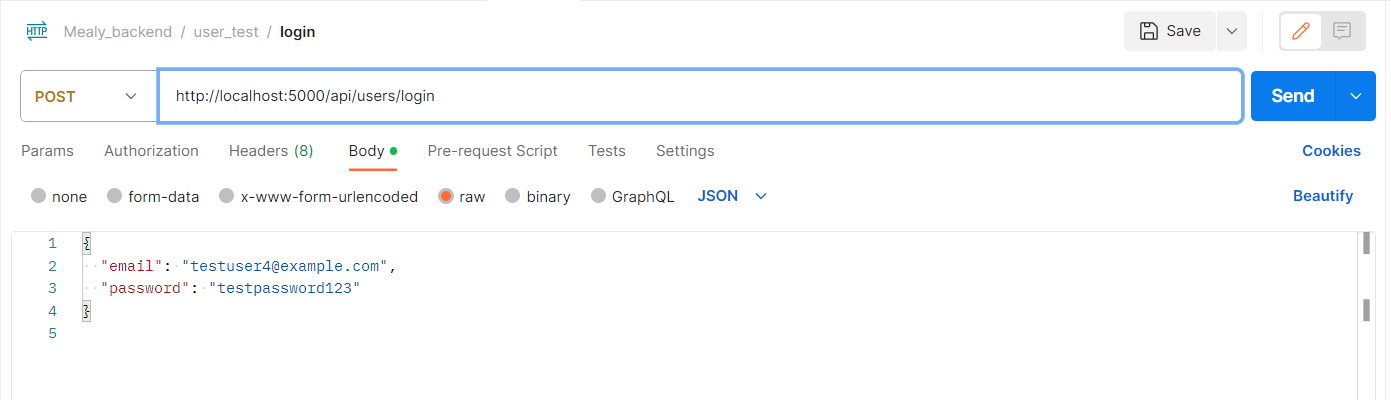
     - Description: Validates the user login functionality.

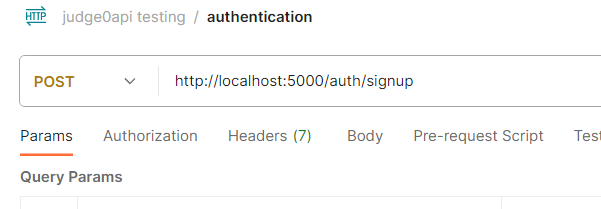
     - Steps:

       1. Send a POST request to the login endpoint with valid user credentials.

       2. Validate the response status code (200 for success).

  3.Check if the response contains the expected data (user details, authentication token).





   - Order Placement API:

     - Description: Ensures a smooth order placement process.

     - Steps:

       1. Send a POST request to the order placement endpoint with sample order data.

       2. Verify that the response status code is as expected.

       3. Confirm the order details in the response match the submitted data.

   - Payment API:

     - Description: Validates the payment processing functionality.

     - Steps:

       1. Send a POST request to the payment endpoint with mock payment data.

       2. Verify the response status code and payment confirmation details.

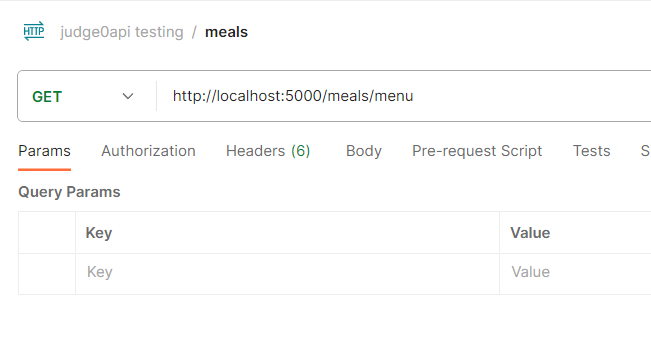
   - Community Interaction API:

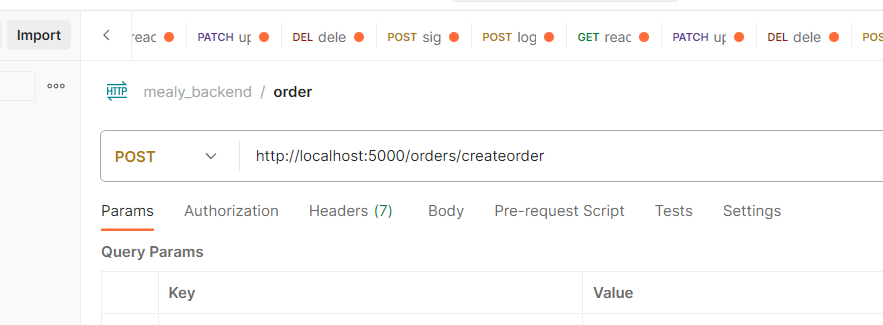
     - Description: Tests interactions with the Mealy community.

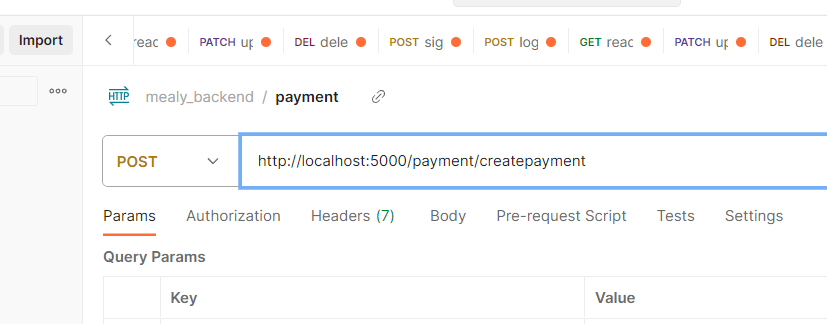
     - Steps:

       1. Send GET and POST requests to the community endpoints.

       2. Validate the response status codes and community data.







**Test Execution:**

   - \*\*Environment Setup:\*\*

     - Define environment variables for different testing environments (development, staging, production).

     - Configure base URLs for each environment.

   - Collections:

     - Create Postman collections for each functional area (Login, Orders, Payments, Community).

The below json data is used to test the routes.  
{

"Username":"blank",

"Password":"1234",

"email":"muhib.mohammed732[@gmail.com](mailto:abdullahbasarvi@gmail.com)",

"experienceLevel":{"$numberInt":"10"}

}

{

"upiPayment":{"InteractID":""},

"paymentDate":12-11-2023

}

{

"title":"SpaghettiCarbonara",

"image":"<https://static01.nyt.com/images/2021/02/14/dining/carbonara-horizontal/carbonara-horizontal-threeByTwoMediumAt2X-v2.jpg>",

"description":"Classic Italian pasta dish with creamy sauce",

"chefName":{"$oid":"652ea18d34d0f045341ce68e"},

"Price":"12.99"

}

   - Test Scripts:

     - Utilize Postman scripts for dynamic testing (e.g., extracting authentication tokens from login responses for subsequent requests).

{

"Username":"user1",

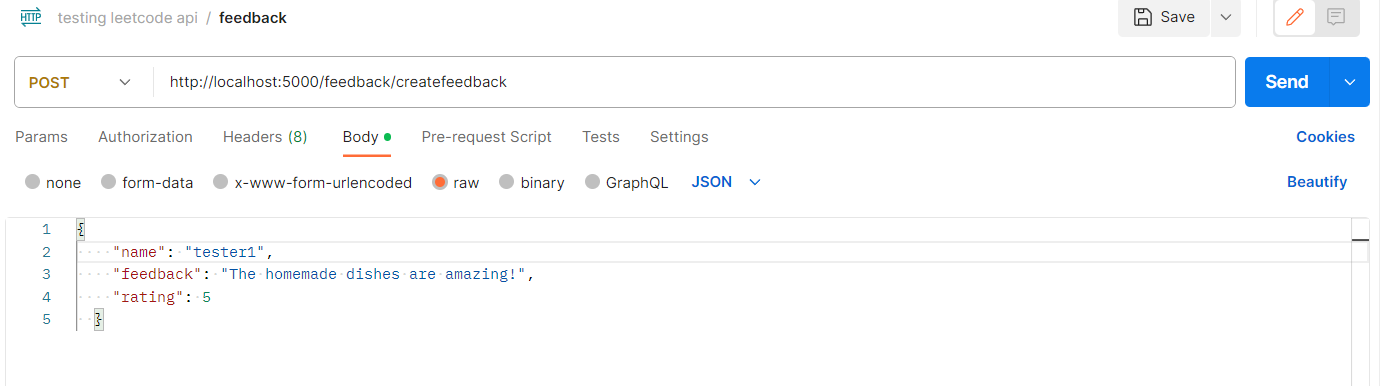
"email":"[user1@test.com](mailto:user1@test.com)",

"Password":"user123”

}

   - Assertions:

     - Include assertions in test scripts to validate expected outcomes.



**Test Reports:**

   - Postman generates detailed test reports for each test run.

   - Reports include information on test cases, request details, response data, and assertion results.

**Continuous Integration:**

   - Integrate Postman tests into the Homey Feasts CI/CD pipeline.

   - Schedule automated test runs after code deployments.

**Maintenance:**

   - Regularly update test scripts to align with changes in the Homey Feasts API.

   - Add new test cases as features are introduced or modified.

**Conclusion:**

   - Postman facilitates comprehensive and efficient automated testing, ensuring the reliability and functionality of Homey Feasts 's API endpoints.

This structure provides a comprehensive guide on how automated testing is organized and executed in Homey Feasts using Postman, covering different functional aspects of the application.