**High-Level Design (HLD) for API Lab**

**1. Overview**

The High-Level Design (HLD) presents the modular architecture of the API Lab. The design will go over significant components in a software system that interacts and exchanges data. Testing in the API development is significant for the following reasons: confirms that API functions meet the expected requirements, handle the API’s responses, evaluating performance and security of the API.

**2. System Architecture   
2.1. Modular Architecture**

* **Authentication Module:** Verifies the identity of users who access the API. API keys, usernames, or passwords are examples of validating credentials used by authorized users to access specific resources.
* **Data Validation Module:** Checks the integrity and validates the data being received by the API. The module ensures that the data meets requirements such as data types, formats, and constraints.
* **Request Handling Module:** Receives and processes requests from the users. This module handles the operations to execute actions such as manipulating data, performing calculations, or invoking other services.
* **Response Generation Module:** Creates and formats the API’s responses. It takes data from the previous module and changes it into the correct format. This can depend on the specifications of the API.
* **Error Handling:** Handles and manages errors that occur with users’ API requests. This module detects the following errors: invalid requests, server issues, or authentication failures. Error messages are prompted to let the client know the current issue.
* **Database Integration:** Connects and interacts with a database. It tackles tasks such as inserting, updating, and deleting data from the database based on the requests received from users.

**3. Components  
3.1. Web Application**

* Testing endpoints to add, update, retrieve, and delete fruits from the application.
* Performance and security tests are conducted on the API to test under different conditions and vulnerabilities.

**3.2. Postman (API Development and Testing Tool)**

* Postman can be used to test and validate the functionality of the application’s API endpoint.
* Requests can be made to simulate actions such as adding fruits to the cart, updating the quantity of specific fruit, or sorting fruit based on the low and high prices.
* In Postman, one is allowed to set headers, parameters, and authentication procedures to correctly simulate real world scenarios.

**3.3. Snipping Tool**

* The snipping tool can capture screenshots of Postman while testing API’s or analyzing responses. One can perform specified actions on Postman and capture them. The use of this tool can be used for documentation purposes.

**4. Data Flow**

1. **Send API Requests:** Postman is utilized to send API requests to the fruit stand application’s endpoints. Requests that were mentioned prior can include the following actions.

2. **Receive API Responses:** The application processes the API requests and sends back responses. Responses may include information about specific fruits, error messages, or messages prompted after the user makes a purchase.

3. **Validate Responses:** API responses that were received needs to be validated to check whether the responses contain the expected data, correct status codes, and abide by the specifications of the API **(Postman).**

4. **Perform Assertions:** Assertions can be implemented to verify that the API responses meet the expected requirements. This can be whether the responses match the expected values or if it’s formatted correctly.

5. **Analyze Test Results:** Test results are analyzed after the assertions have been performed. If the assertions pass, the API functions as expected. However, if assertions fail, there could possibly be issues regarding the API’s functionality or data.

**5. Integrations**

* **Slack:** In conjunction with Postman, one can receive notifications and updates about API testing in Slack channels. This promotes collaboration among team members efficiently.
* **Azure DevOps:** In conjunction with Postman, API testing is enabled as part of the application development and delivery process within the Azure DevOps environment.

**6.** **Scalability and Maintenance**

* Reusable scripts and snippets can be created in Postman to utilize in API tests. Code reuse is promoted while duplication is reduced just as much. Tests are maintainable and scalable due to this.
* API tests can be organized into collections and test suites with the help of Postman. The purpose of this is to manage and maintain tests as the API grows. Tests can be grouped together and executed as a suite to maintain scalability in testing.

**7. Conclusion**

In brief, Postman provides several benefits for API testing. The creation and execution of API requests are simplified to make testing efforts efficient. Features such as test suites and environment variables are utilized for testing purposes. This HLD document delves into the software’s architecture of how APIs are conducted. The data flow is presented for one to follow along with how to conduct API testing.