SRS – API (POSTMAN and ‘request’ Library)

1. Introduction – This document details the interactions between the python library ‘request’ and the API (Application Programming Interface) POSTMAN as it interacts with our Fruit Stand application.  
   1. Scope – This document will outline functional and non-functional requirements of the POSTMAN API and ‘request’ Python libraries as they interact with our Fruit Stand application
2. System Overview
   1. Objectives of POSTMAN API and ‘request’ lab
      1. Familiarize the users with POSTMAN and ‘request’ library HTTP requests and formatting.
      2. Further compliment users’ knowledge of the testing methodologies and testing suites available in Python.
   2. Technologies Used
      1. IDE of choice
      2. Python
      3. Flask
      4. POSTMAN
      5. ‘Request’ Library within Python
3. Functional Requirements
   1. Adding Fruit to the cart
      1. Users should be able to add fruit to their cart and have POSTMAN API send a POST request to update the user’s cart with the correct fruit.
   2. Removing Fruit from the cart
      1. Users should be able to remove a selected fruit from their carts and the POSTMAN API will send a DELETE request to correct the items in the cart.
   3. Completing their purchase with product in cart.
      1. Users should be able to make a purchase with items in their cart and POSTMAN API should create a POST request to confirm the purchase.
   4. Sorting items within Users cart
      1. Users should be able to sort the items in their cart by price in either ascending or descending order the ‘request’ library will sort the items in the order specified.
4. Non-Functional Requirements
   1. Performance requirements
      1. The system should be able to process user requests (adding or removing items and completing purchases) within a reasonable time.
   2. Scalability Requirements
      1. The system should be able to handle multiple user requests without creating recognizable degradation in performance.
   3. Security Requirements
      1. API Endpoints should be protected from unauthorized access.
5. Setup and Environments
   1. Supported web browser for Fruit Stand application launch.
   2. Python version 3.8 or newer.
   3. Preferred IDE software
   4. POSTMAN API
   5. ‘requests’ Python Library
6. Deliverables
   1. Familiarization with basic POSTMAN HTTP requests such as POST, and DELETE and its interactions with web applications.
   2. Generating a report to verify requests between API endpoints and application interact in a consistent and predictable way.
7. Conclusions
   1. This lab will be a basic walkthrough utilizing the functionalities of the POSTMAN API and the Python ‘request’ library and their requests used in conjunction with a Fruit Stand web application.
8. Appendices
   1. Reference Materials Used
      1. Python Documentation - https://www.python.org/doc/
      2. Flask Documentation - https://flask.palletsprojects.com/en/3.0.x/
      3. POSTMAN Documentation - https://learning.postman.com/docs/introduction/