Response Evolution System

**Description:** The Amazon Feedback System is a Python program designed to collect feedback from customers about their product experience. It allows customers to rate various aspects of the service and provide comments. The program can be used by name, product quality, establishment to gather valuable feedback to improve their offerings and customer satisfaction.

**Requirements:**

**Functionality:**

1. **Submit Feedback**:
   * Users can submit feedback for products by providing their username, product ID, rating, and comments.
2. **View Feedback**:
   * Users can view feedback for a specific product by providing its ID.
3. **Feedback Storage**:
   * Each feedback entry is stored as a dictionary containing information such as username, product ID, rating, and comments
   * .

**Non functionality:**

1. **Performance**:
   * **Responsiveness**: The program should respond quickly to user interactions, ensuring minimal delay between submitting feedback and viewing it.
   * **Scalability**: It should be capable of handling a growing number of feedback submissions and large datasets without a significant decrease in performance.
2. **Usability**:
   * **User Interface**: The program should have a user-friendly interface, making it easy for users to submit feedback and view feedback entries.
   * **Accessibility**: The system should be accessible to users with disabilities, following relevant accessibility standards and guidelines.
3. **Security**:
   * **Data Confidentiality**: Feedback data should be kept confidential and protected from unauthorized access or disclosure.
   * **Authentication and Authorization**: Access to sensitive functionality or data should be restricted to authorized users only, using appropriate authentication and authorization mechanisms.

**Approach:**

1. **User Authentication**:
   * Implement user authentication to ensure that only registered users can submit feedback. This enhances security and prevents misuse of the feedback system.
2. **Web Development Framework**:
   * Utilize a web development framework such as Flask or Django to create a web-based interface for the feedback system. This enables users to access the feedback system through a web browser.
3. **Input Validation**:
   * Implement input validation to ensure that user-provided data is valid and within acceptable ranges. This prevents issues such as SQL injection attacks and data corruption.
4. **User Interface Enhancement**:
   * Improve the user interface to enhance usability and user experience. This may involve adding features such as pagination, search functionality, and visualizations of feedback data.
5. **Security Measures**:
   * Implement additional security measures such as HTTPS encryption, CSRF protection, and rate limiting to prevent unauthorized access and protect user data.

**Program:**