Q1: You have a shoe-selling e-commerce platform. Now, the client wants to implement a new feature "Search". During the client meeting, they mentioned they wanted to make the search dynamic, with any keywords, or specific text. Now you have a chance to get into more details on the feature. You can ask 10 questions to the client. List the questions according to priority.

Ans:

I have an opportunity to gather more details about the "Search" feature the client wants to implement. I can ask the following 10 questions in order of priority:

1. **What specific types of data can users search for (e.g., product name, brand, price, category)?**
2. **How should search results be ranked or sorted (e.g., relevance, price, popularity)?**
3. **Should the search support filtering or faceting (e.g., by price range, size, color)?**
4. **How should partial keyword matches be handled? Should the search use fuzzy matching?**
5. **Do you want real-time search suggestions as users type, or should results appear after a full keyword is entered?**
6. **Should the search support natural language queries (e.g., "cheap black shoes under $50")?**
7. **What should happen if no results are found (e.g., display alternative suggestions, show "no results" message)?**
8. **Should the search be case-sensitive, or handle synonyms or alternate spellings (e.g., "sneaker" vs. "trainers")?**
9. **How should the search handle misspellings or typos (e.g., auto-correction, suggestions)?**
10. **Should the search take into account product availability (e.g., exclude out-of-stock items from results)?**

Q2: Write down the test cases according to your questions.

Ans:

Test cases are derived from the client's requirements. Here's how i could write test cases based on the questions:

Test Case ID	Test Case Description	Expected Result	Priority

TC001	Verify if the user can search for products by product name	Relevant products matching the search query are displayed	High
TC002	Verify if the user can search for products by brand name	Relevant brand products are displayed	High
TC003	Verify if search results can be filtered by price range	Results are correctly filtered based on the chosen price range	High
TC004	Verify if partial keyword matches return relevant results	Partial matches are displayed	Medium
TC005	Verify if real-time suggestions are shown as the user types	Dynamic suggestions are displayed as the user types	High
TC006	Verify if the search can handle misspellings or typos	Suggestions or auto-corrections are provided for typos	Medium
TC007	Verify if the search supports natural language queries	Products relevant to natural language queries are displayed	Medium
TC008	Verify if no results page displays alternative suggestions when no match is found	Suggestions are displayed when no results are found	High
TC009	Verify if the search is case-insensitive	Search results are the same regardless of letter case	Low
TC010	Verify if out-of-stock items are excluded from the search results	Only available items are shown in the results	High

Q3: Let's say, your development team implemented the feature on the **Evershop** site (https://demo.evershop.io/). Now execute your test case into the URL and generate a report.

Ans:

1. Test Case Execution on Evershop Site

Open this site in my browser: https://demo.evershop.io/. Here, we'll execute the following test cases.

Test Case 1: Search by Product Name

- **Action**: Enter "Nike" in the search bar.
- **Expected Result**: Products related to Nike should appear in the results.
- Actual Result: [Search results for "Nike"].
- **Status**: Pass

Test Case 2: Search by Brand

- **Action**: Enter "Adidas" in the search bar.
- **Expected Result**: Adidas brand-related products should appear.
- **Actual Result**: [There is no product to display].
- **Status**:Fail.

Test Case 3: Filtering Results by Price

- **Action**: Search for "Nike" and then apply a price filter (e.g., filter for prices between $100 and $200).
- **Expected Result**: Only Nike products within the price range of $100 to $200 should be displayed.
- **Actual Result**: [it's work].
- **Status**: Pass

Test Case 4: Partial Keyword Search

- **Action**: Enter "Nik" in the search bar (a partial word for "Nike").
- **Expected Result**: Results for "Nike" should appear.
- **Actual Result**: [There is no product show in display].
- **Status**: Fail.

Test Case 5: Misspelled Search Query

- **Action**: Enter a misspelled word like "Nkie" instead of "Nike."

- **Expected Result**: The search system should suggest the correct word or display related products.
- **Actual Result**: [System should not suggest the correct word].
- **Status**: Fail.

Test Case 6: No Results Found Handling

- **Action**: Enter a random string like "xyzabc" in the search bar.
- **Expected Result**: The system should display a "no results found" message and possibly suggest other searches.
- **Actual Result**: [no result found].
- **Status**: Pass.

Test Case 7: Real-time Search Suggestions

- **Action**: Start typing "Nike" in the search bar, but pause after each letter.
- **Expected Result**: Real-time search suggestions should appear as i type.
- **Actual Result**: [Real-time search suggestions shouldn't appear].
- **Status**: Fail.

2. Tracking Defects

As I execute the test cases, any bugs or unexpected behaviors should be logged. Here's how i can do that:

Defect ID	Test Case ID	Steps to Reproduce	Expected Result	Actual Result	Severity
DEF001	TC002	1. Enter "Adidas" in the search bar. 2. Press Enter.	Adidas products should appear.	No products were displayed even though Adidas is available.	Major
DEF002	TC004	1. Enter "Nik" in the search bar 2. Press Enter.	Results for "Nike" should appear	There is no product show in display	Minor
DEF003	TC005	1. Enter a misspelled word like "Nkie" 2. Press Enter.	The search system should suggest the correct word or display related products.	System should not suggest the correct word	Critical

Defect ID	Test Case ID	Steps to Reproduce	Expected Result	Actual Result	Severity
DEF004	TC007	1. Start typing "Nike" in the search bar. 2. pause after each letter	Real-time search suggestions should appear as i type	Real-time search suggestions shouldn't appear	Major

3. Generating a Test Execution Report

Test Case ID	Test Case Description	Status	Defect ID
TC001	Verify product search by product name	Passed	N/A
TC002	Verify product search by brand	Failed	DEF001
TC003	Verify filtering search results by price	Passed	N/A
TC004	Verify partial keyword search functionality	Failed	DEF002
TC005	Verify misspelled word handling in search	Failed	DEF003
TC006	Verify no results page behavior	Passed	N/A
TC007	Verify real-time search suggestions	Failed	DEF004

Test Execution Summary Key Points:

- **Total Test Cases Executed**: 7
- **Test Cases Passed**: 3
- **Test Cases Failed**: 4
- **Defects Found**: 4 (DEF001, DEF002, DEF003, DEF004)