Requirement Specification

Shopping Web Application

# Executive Summary

This document outlines the user requirements of a shopping web application t be developed as part of an observational case study. The application will include typical functions of shopping web site including a home page, product list, shopping cart, and customer enquiry form. The application will be built in a two-hour session using agentic tools to observe Developer-AI collaboration in software development. Upon completion, functional requirements will be evaluated against the test cases specified in a test case specification.

The structure of this document follows the Guide to the Software Engineering Body of Knowledge (SWEBOK) v4a (Washizaki, 2024) to ensure alignment with recognised software engineering standards.

The attributes of test Case is defined in accordance with the guidelines presented in ISO/IEC/IEEE 29119-3:2021 (Test documentation, Section 8.3), which includes a unique identifier, objective, priority, traceability, preconditions, inputs, and expected results (IEEE, 2021). For practicality, the test cases are presented in an Excel sheet format and include additional columns titled Actual Result and Test Result. Two copies of it will be made for each observation session.

# Technology Stack

The web application will be implemented using a lightweight, flexible, and open-source technology stack that is compatible with the capabilities of Cursor, which was chosen as the agentic AI tools for this case study. The selected stack is designed to ensure ease of development, maintainability, and sufficient functionality to support observation and evaluation in this study.

* 1. Programming language
     1. **Python (version 3.11 or higher)**Python is chosen as it is widely used programming language in both academia and industry. It is well-supported by various AI-assisted code editor such as Cursor and offers extensive libraries for web development, testing, and data handling (Python software foundation, 2025; Cursor, 2025).
  2. Web Framework
     1. **Flask (version 2.x)**Flask is a lightweight and flexible web application framework based on Python that enables rapid development of web applications (Pallets, 2025).
  3. Development Environment and Tools
     1. **Cursor**

Cursor is an AI-powered code editor designed for ease of adoption. It offers various levels of autonomy including Ask Mode, Agent Mode, Background Mode. This allows observation of AI–developer collaboration under different interaction styles (Cursor, 2025).

* 1. Deployment Environment
     1. **The a**pplication will be hosted and tested locally to support observation.
  2. Version Control
     1. GitHub will be used to store the project files and manage version control.

# Functional Requirements

## Navigation

* + 1. The application shall provide a navigation menu that allows users to move to any page from any location.

## Home page

* + 1. The application shall start with a home page displaying the shop information and navigation to product list and enquiry form.
    2. The shop information shall display the shop’s name, logo, contact details as shown below.

|  |  |
| --- | --- |
| Shop Name | IKW store |
| Logo | Figure 1. Logo design for IKW store (created by the researcher using AI-assisted image generation). |
| Contact Name | Waichi Ikeda |
| Email | [W.Ikeda@liverpool.ac.uk](mailto:W.Ikeda@liverpool.ac.uk) |
| Phone Number | +81 00-000-0000 |

## Product List

* + 1. The product list shall display items in a grid format, with each product showing a thumbnail image, name, brief description, and price.
    2. The product list shall be automatically generated and contains 20 computer accessory products. The price shall be presented in Japanese Yen (JPY) with the range from ¥1000 – ¥15000.
    3. The application shall provide a case-insensitive search function, enabling users to locate products by name.
    4. The application shall allow users to rate products using a star rating from 1 to 5.

## Shopping Cart

* + 1. The application shall allow users to add products to a shopping cart directly from the product list by clicking the button “Add to Cart” shown for each product.
    2. The application shall provide a dedicated shopping cart page where users can view cart contents, update product quantities, and remove items.
    3. The shopping cart shall display a running subtotal and overall total.
    4. Cart contents shall persist for the duration of the user session or until checkout is initiated.
    5. The cart shall include a “Checkout” button for demonstration purposes; however, the checkout process itself is out of scope for this study and shall not be implemented.

## Enquiry Form

* + 1. The application shall provide an enquiry form accessible from the home page and navigation menu.
    2. The form shall collect the following information: user’s name, email address, subject, and message content. All these fields are mandatory.
    3. The system shall validate that all fields are completed and that the email address is entered in a valid format.
    4. The system shall validate that all fields are protected from injection attacks.
    5. Upon submission, the system shall display a confirmation page showing the information entered by the user. No email transmission shall occur, as this functionality is outside the scope of the study.

# Non-Functional Requirements

* 1. **Usability**: The interface shall be simple to use and responsive.
  2. **Performance**: Pages shall load within 2 seconds under normal conditions. Cart actions and enquiry form actions shall finish within 1 second.
  3. **Security**: All user input shall be checked and cleaned to avoid injection attacks.
  4. **Scalability**: The application shall be designed so it can connect with payment gateways in the future.

# Assumptions and Constraints

* 1. Users are expected to have access to modern web browsers (Chrome, Firefox, Edge, Safari).
  2. The application will not include payment processing; checkout will be limited to order summary and enquiry.

1. Acceptance Criteria

All functional requirements defined in this document are “must-have” criteria. Every test case specified in the accompanying Test Case Specification shall be executed and pass without error.

**References**

Cursor (2025) *Cursor Documentation*. Available at: [**https://docs.cursor.com/en/welcome**](https://docs.cursor.com/en/welcome) (Accessed: August 1, 2025).

IEEE (2021) 'IEEE/ISO/IEC International Standard for Software and systems engineering--Software testing--Part 3:Test documentation', *ISO/IEC/IEEE 29119-3:2021(E)*, pp. 1-98.

Pallets (2025) *Flask’s documentation*. Available at: <https://flask.palletsprojects.com/en/stable/> (Accessed: October 2, 2025).

Python software foundation (2025) *Python home page*. Available at: <https://www.python.org/about/> (Accessed: October 5, 2025).

Washizaki, H. (2024) *Guide to the Software Engineering Body of Knowledge (SWEBOK Guide), Version 4.0*: IEEE Computer Society. Available at: [www.swebok.org](https://d.docs.live.net/7c695e43d0e25aed/MasterDegree/CSCK700_ComptuerScienceCaptoneProject/deliverables/IT%20artefacts/www.swebok.org) (Accessed: July 16, 2025).