PART 1: TEST PLAN DESIGN

1. **Scope:**
   - *Backend (C# APIs)*
     - User Authentication:
       - Login validation with valid and invalid credentials.
       - Authentication token generation, validity and expiration tests.
     - Product Management:
       - CRUD (Create, Read, Update, Delete) operations for products.
       - Error handling for invalid product data.
       - Search tests using id.
     - Order Processing:
       - Order creation and verification.
       - Canceling and updating order status.
       - Verification of error handling in invalid order transactions.
   - *Frontend (ReactJS)*
     - User Authentication and Dashboard:
       - Verification of login and logout functionalities.
       - Testing of data display and update in the user's control panel.
       - Testing of the responsiveness and accessibility of the user interface.
     - Product Listing and Order Processing:
       - Verification of product listing display and filtering.
       - Testing of order creation, update and cancellation workflows.
       - Testing of the responsiveness and accessibility of the user interface.
2. **Objectives:**
   - **Software Quality:** Ensure that all critical backend and frontend functionalities work correctly and are robust to invalid input.
   - **Usability:** Ensure that the user interface is accessible, easy to use and displays correctly on different devices and resolutions.

- **Security:** Verify that the authentication process is secure and that sensitive data is protected.
- **Reliability:** Ensure that the system adequately handles errors and provides clear feedback to the user.

3. **Resources**
   - **Tools:**
     - Postman: For backend API testing.
     - Cypress: For automated frontend testing.
     - Git: For version control and code storage.
   - **Environments:**
     - Local development environment: local server for backend testing.
     - Browser: Google Chrome for frontend testing.
     - Devices: Desktop, tablet and mobile computers for responsiveness testing.
   - **Test data:**
     - Test users with different credentials (correct and incorrect).
     - Sample products with different attributes.
     - Test orders with different status and configurations.
4. **Risks**
   - **Technical risks:**
     - Possible version incompatibilities between test tools and the development environment.
     - Errors in the test environment configuration that could affect test execution.
   - **Mitigation strategies:**
     - Configuration and verification of controlled and well-documented environments before testing begins.
     - Use of a separate, replicable test environment to avoid interference with the production environment.
     - Running tests on multiple browsers and devices to ensure complete coverage.

5. Results

- Test Plan Document: It will contain the test strategy, scope, objectives, resources, risks and deliverables.
- Test Cases: A complete set of test cases designed to cover all the functionalities described in the scope.
- Test Execution Report: Detailed documentation of test results, including cases passed, failed, and any defects found.
- Automation Scripts: Automated test scripts for key frontend and backend tests, accompanied by execution instructions.