

# **Software Requirements Specifications**

**Project Name: TechXpress**

**Under the supervision of Eng. Mahmoud Al Shahaly**

**Jana Hani Mohamed**  
**Samira Khalid Anwar**  
**Nouran Salah Ibrahim**  
**Esraa Ramadan Atia**  
**Sandy Melad Philip**

## 1. Introduction

TechXpress is an E-Commerce platform designed with the purpose of selling Tech products such as Laptops, PCs, Smart Phones, etc..

## 2. User Requirements

These are the requirements stipulated by the user base:

- **User Login and Signup:**
  - The program should allow for the creation of a user account.
- **Shopping Cart Management:**
  - The program should have a functional 'shopping cart' where the user can add items they wish to buy.
- **Customer Service:**
  - The program should have customer service where users can go to settle disputes with the sellers, problems with the program, etc...
- **Order:**
  - The program should allow the user to purchase the items stored in their shopping cart.
- **Order History:**
  - The program should keep track of the purchases made by the user.

- **Bookmarks:**
  - The program should allow the user to save their favorite search results.
- **Checkout and Payment:**
  - The program should have an efficient and reliable system for checkout and payment.
- **Search and Filter:**
  - The program should allow users to look up items they wish to buy as well as allow for the filtering of the results.
- **Admin Dashboard:**
  - The program should contain admin controls.

### 3. Functional Requirements

- **User Login and Signup:**
  - **Description/Action:** Allows users to create an account by entering their email address, username, and password/login by entering their email and password.
  - **Requirements/Inputs:** Email, password, username.
  - **Source:** User Input via registration form.
  - **Pre-Condition:** The user is not already logged in/has an account.
  - **Post-Condition:** The user is logged in/a new account is created.
- **Shopping Cart Management:**

- **Description/Action:** Allows users to add items they wish to buy to a shopping cart.
- **Requirements/Inputs:** Product ID and quantity.
- **Source:** User input via “Add to cart” button.
- **Pre-Condition:** The user is logged in.
- **Post-Condition:** The item is added to the user’s cart.
- **Customer Service:**
  - **Description/Action:** Allows users to contact a program/company representative.
  - **Requirements/Inputs:** Complaint message.
  - **Source:** User input via “Customer Service” button.
  - **Pre-Condition:** The user is logged in.
  - **Post-Condition:** The user is in contact with a representative.
- **Contacting the Seller:**
  - **Description/Action:** Allows users to contact the seller of a product.
  - **Requirements/Inputs:** Text message depicting the request the buyer has for the seller.
  - **Source:** User input via “Contact the Seller” button.
  - **Pre-Condition:** The user is logged in.
  - **Post-Condition:** The user is in contact with the seller of a given item.
- **Order:**
  - **Description/Action:** Allows users to make orders through their shopping carts.
  - **Requirements/Inputs:** User is logged in and has at least one item in their shopping cart.
  - **Source:** User Input via “Order” button.

- **Pre-Condition:** The user is logged in.
- **Post-Condition:** The user is capable of reviewing the items they want to buy before making their purchase.
- **Order History:**
  - **Description/Action:** Allows users to have information about their past orders stored on their profile.
  - **Requirements/Inputs:** User is logged in and has performed at least one past purchase.
  - **Source:** User Input via “View Order History” button.
  - **Pre-Condition:** The user is logged in.
  - **Post-Condition:** The user is capable of viewing all the purchases they have made in the past.
- **Checkout and Payment:**
  - **Description/Action:** Allows the user to provide payment to the seller in either cash on delivery or credit format.
  - **Requirements/Inputs:** User is logged in and shopping cart has at least one item within.
  - **Source:** User Input via “Checkout” button.
  - **Pre-Condition:** The user is logged in.
  - **Post-Condition:** The user has paid/will pay and the product is flagged to be delivered.
- **Bookmark:**
  - **Description/Action:** Allows the user to store items they may not wish to immediately purchase for later perusal.
  - **Requirements/Inputs:** User is logged in and has performed at least one search in the website.
  - **Source:** User Input via “Save for Later” button.
  - **Pre-Condition:** The user is logged in.
  - **Post-Condition:** The user has an item stored in the Bookmarks portion of their profile.

- **Search and Filter:**
  - **Description/Action:** Allows the users to look up certain products and filter the results provided.
  - **Requirements/Inputs:** User is logged in.
  - **Source:** User Input via search bar.
  - **Pre-Condition:** The user is logged in and looking for products to buy.
  - **Post-Condition:** The user is viewing a page full of search results.
  
- **Admin Dashboard:**
  - **Description/Action:** Allows users with special permissions – admins – to control unseen aspects of the program.
  - **Requirements/Inputs:** User is logged in with special account.
  - **Source:** User Input via various admin controls.
  - **Pre-Condition:** The user is not logged in.
  - **Post-Condition:** The user is logged in and is capable of viewing the various changes they can affect in the program.

## 4. Non-Functional Requirements

- **Performance Requirements:**
  - **Response Time:** The system should provide search results in under 10 seconds under normal conditions.
  - **Execution Speed:** The checkout process should be completed in under 15 seconds under normal conditions.

- **Reliability Requirements:**

- **Uptime:** The program should have an uptime of 99.9%, ensuring that it is always available to users except during scheduled maintenance.
- **Error Handling:** The program should handle errors gracefully, displaying user-friendly error messages and logging errors for later debugging.

- **Usability Requirements:**

- **User Interface:** The system should have a responsive and user-friendly UI that functions on multiple interfaces.
- **Accessibility:** The system should be accessible and accommodating to all users.

- **Security Requirements:**

- **Data Encryption:** All sensitive user data should be encrypted under the conventional encryption standards.
- **Authentication:** The system should enforce strong password policies and support multi-factor authentication for added security.

- **Scalability Requirements:**

- **Load Handling:** The system should support up to 10,000 concurrent users without going down.
- **Storage:** The systems should support up to 1,000,000 products and scale the storage capacity as needed.

- **Compatibility Requirements:**

- **Browser Support:** The system should be compatible with the latest versions of Chrome, Firefox, Safari, and Edge.

- **Operating Systems:** The system should be supported on Windows, MacOS, iOS, and Android.

- **Maintainability Requirements:**

- **Code Quality:** The system should follow coding and clean code conventions.
- **Version Control:** Git should be used for version control.