**Project Report: Kings PC Services E-commerce Website**

**i. Introduction and Business Concept**

This project is an academic and practical web development assignment centered around creating a functional e-commerce website for a real business: **Kings PC Services**. The business sells tech gadgets including laptops, gaming PCs, accessories, SSDs, hard drives, and computer components. The objective of this assignment was to apply web development concepts practically by designing and implementing a digital storefront with basic e-commerce functionality.

Kings PC Services targets Tanzanian customers and aims to provide a local, accessible, and convenient digital platform where customers can browse, shop, and simulate payments for tech products. Additionally, an admin dashboard was implemented to allow product management and user control. This included the ability to register, log in, simulate checkout, and dynamically manage products — thereby creating a realistic mini e-commerce system.

**ii. Platform/Technology Used**

The project uses the following tools and technologies:

* **HTML5:** For page structure and layout.
* **CSS3:** For styling the site with a consistent black-and-white color theme.
* **JavaScript (ES6):** For interactive functionality including user registration, login, cart system, and admin features.
* **LocalStorage API:** For persistent data storage within the browser (products, users, cart items).
* **Visual Studio Code (VS Code):** As the main development environment.
* **Live Server (VS Code Extension):** For running the website locally and testing changes in real time.
* **Mobile Phone Access via Local IP:** For testing cross-device performance and responsiveness.
* **Placeholder image hosting:** Images were either base64 encoded or served via URL for testing.

The solution was built entirely using **frontend technologies**, without any backend framework or database server, which allowed for simplicity and accessibility while still demonstrating core programming skills.

**iii. Site Architecture and Navigation**

The website follows a modular page structure:

1. **index.html (Home Page):**
   * Displays categories, introduction, about us, and contact details.
   * Allows login/register with modal forms.
   * Navigation menu and hamburger side menu for mobile.
2. **products.html (Product Listing Page):**
   * Dynamically displays available products from localStorage.
   * Allows customers to add products to cart.
3. **cart.html (Cart Page):**
   * Lists items added to cart.
   * Allows clearing cart.
4. **checkout.html (Payment Simulation):**
   * Displays cart total.
   * Simulates payment form submission.
   * Clears cart after "payment".
5. **admin.html (Admin Dashboard):**
   * Only accessible by logging in as "admin".
   * Allows adding, editing, deleting products.
   * Allows viewing and deleting users.
   * Admin navigation menu hides user navigation links.
6. **profile.html (User Profile Page):**
   * Displays the logged-in user’s information.

Navigation is handled using:

* A **header navigation bar** for desktop users.
* A **hamburger side menu** available across all pages for mobile-friendly navigation.
* Links are conditionally shown based on user type (admin vs regular user).

**iv. Design Elements (UI/UX considerations)**

The UI/UX design prioritizes **simplicity, accessibility, and consistency**:

* **Color Theme:** A consistent black and white color scheme to align with a professional tech brand identity.
* **Responsive Design:** Uses media queries and Flex/Grid layouts to adapt to various screen sizes.
* **Navigation:** Both top menu and hamburger side menu ensure smooth access across all devices.
* **Button Styling:** Rounded corners, hover effects, and consistent button sizes across all interactions.
* **Forms:** Styled input fields with placeholders and validation to enhance usability.
* **Visual Feedback:** Alerts for login, registration, cart addition, and successful checkout.
* **Image Preview:** Admins see image previews before saving products.
* **Greeting Message:** Personal welcome message shown to logged-in users.
* **Conditional UI:** Login/register buttons disappear once a user is logged in; admin gets redirected or shown a dashboard link.

**v. Security Features Implemented**

Although the application is built entirely on the frontend, basic security-like features were implemented:

* **Role-based Access:**
  + Admin dashboard is only accessible after logging in as “admin”.
  + Users cannot type in admin.html manually and gain access without logging in as admin.
* **LocalStorage Credential Checks:**
  + Login and registration validate credentials using stored user data.
  + Passwords are stored in plaintext (for educational purposes), but usernames and passwords are validated before access is granted.
* **Hidden Navigation for Admin:**
  + When admin logs in, customer navigation options (products, cart, checkout) are hidden both from the top menu and side menu.
  + Admin-only links like “Add Product” and “Edit Product” are displayed.
* **Session Simulation:**
  + LocalStorage is used to simulate session persistence.
  + Logged-in users are remembered until they manually log out or clear the browser cache.

In a production-grade site, password hashing, database storage, and secure authentication protocols would replace localStorage.

**vi. Digital Payment Simulation**

The **checkout page** includes:

* A summary of cart total (in Tanzanian shillings, Tzs).
* A styled **payment form** collecting:
  + Full name
  + Email
  + Card number
  + Expiry date
  + CVV

When a user submits the form:

* JavaScript validates inputs
* If successful, it simulates “payment success”
* Clears the cart
* Shows a success message

This simulation mimics the behavior of real-world online checkouts, without processing any actual payments.

**vii. Testing and Debugging Processes**

Testing was done manually across different devices and browsers:

**✔ Desktop Testing:**

* Chrome, Firefox on Windows
* Tested login, cart flow, and admin panel

**✔ Mobile Testing:**

* Connected via local IP address (e.g., 192.168.x.x:5500)
* Opened site using Chrome on Android
* Validated responsiveness and performance

**✔ Key Debugging Scenarios:**

* 🐛 Admin login broke after browser refresh — fixed by ensuring admin account is always created in localStorage.
* 🐛 Product duplication in cart — fixed by isolating cart logic and avoiding multiple event listener bindings.
* 🐛 Old products appearing on phone — fixed by syncing localStorage and clearing outdated data.
* 🐛 Admin access showing user navigation — resolved with conditional UI rendering.

**✔ Validation:**

* User input is validated (non-empty, password match, etc.).
* Invalid login shows error messages.
* Forms block submission if required fields are empty.

**viii. Limitations and Areas for Improvement**

**❗ Current Limitations:**

* **Security:** All data is stored in plain localStorage — not secure for real-world use.
* **No Backend:** No server-side database, which means data is lost if localStorage is cleared.
* **No image hosting:** Product images are base64-encoded or linked via placeholders.
* **Single Admin:** Only one hardcoded admin account exists.
* **No real payment processing:** The checkout is a simulated experience.(mockup)
* **Single-device data separation:** Changes made on one device (like laptop) are not reflected on another (like phone) unless manually synced.

**🚀 Possible Improvements:**

* Integrate a backend using Node.js and Express or Firebase
* Store user/product data in a database (e.g., MongoDB or MySQL)
* Encrypt passwords using hashing (e.g., bcrypt)
* Add real authentication and login sessions
* Use file hosting or image CDN for product images
* Implement product search and filters by category
* Improve UI with animations and polish for production readiness
* Convert into a Progressive Web App (PWA) or mobile app

**ix. Conclusion**

The Kings PC Services e-commerce project provided valuable hands-on experience in designing and developing a functional web application. The project successfully integrates key features of a digital storefront, including product display, cart functionality, checkout simulation, user authentication, and an admin dashboard.

From a learning perspective, this assignment allowed the application of HTML, CSS, and JavaScript in a real-world context, reinforcing concepts like DOM manipulation, data storage, user interaction, and responsive design. While limited by the absence of a backend or secure data handling, the system demonstrates a working prototype of a tech gadget store that could be extended into a full e-commerce platform in the future.

This project also highlights the importance of user roles, conditional UI rendering, and data persistence. Future improvements would aim at adding backend support, advanced security, and real payment gateways.

**x. Screenshots of the Interface**

📌 **Note**: attached are images of how my website looks like

* Home page showing hero section, categories, and contact info
* Products page with 3-column product layout
* Cart page listing selected products with total
* Checkout form showing payment simulation
* Admin dashboard with:
  + User list and delete option
  + Add product form with image preview
  + Edit/delete product list
* Mobile view