Case Study – GLAM Online Shopping Website

Objective: The goal of the GLAM E-Commerce platform is to create an online shopping experience for clothing targeted at both women and men.

Functional Requirements:

1. User Authentication: Customers must log into the platform using their valid username and password to shop.
2. User Registration: Customers are required to register before logging in and making purchases. However, guests can browse products without registering.
3. Admin Interface: Administrators should have the capability to add, modify, or delete items from the inventory.
4. Product Search: Customers must be able to search for specific products easily.
5. Product Showcase: A comprehensive display of products available for purchase should be provided.
6. Promotions: Items in the promotions section should feature special discounts.
7. Inventory Check: The platform must verify the availability of products in stock.
8. Pricing: The system should maintain accurate pricing information for all displayed products.
9. Filtering Options: Customers should have the ability to sort products based on availability, price, ratings, color, and brand.
10. Shopping Cart Functionality: Users must be able to add products directly to their shopping cart.
11. Payment Methods: The system should support various secure payment options, including direct bank transfers, checks, and drafts, with third-party security from providers like PayPal, Razorpay.
12. Order Confirmation: Upon completing a purchase, customers should be logged out, and a confirmation email should be dispatched.
13. Additional Pages: The platform will include features such as Order History, Customer Profiles, Review Sections, New Arrivals, Testimonials, and a Contact Us page.

Non-Functional Requirements:

1. The solution must include measures to protect against security vulnerabilities (e.g., OWASP Top 10). All system components should adhere to recognized security standards.
2. Ensure compatibility with the latest versions of major browsers (Chrome, Firefox, Safari, Edge) and popular mobile devices (iOS, Android).
3. The platform should be capable of accommodating 10 million users and support 10,000 simultaneous users.
4. Aim for a page load time of less than 2 seconds for standard pages and under 5 seconds for more complex pages.
5. Target an uptime of 99.9%.
6. The system should support both on-premises and cloud deployment, featuring auto-scaling capabilities to meet user demand.
7. Design the application to manage a 50% increase in concurrent users without compromising performance.
8. The proposed solution must include monitoring and maintenance capabilities, along with testability, configurability, and upgradeability.
9. The architecture should be adaptable to accommodate future increases in load.

10. The system will initially support only the English language, with no multilingual features required.

Deliverables:

Develop a High-Level Design (HLD) that includes the following components:

1. A suitable architectural framework for the application, illustrated with a diagram and detailed descriptions of each component.
2. Provide descriptions and scopes for each identified service.
3. Technology stack chosen for the application.
4. A deployment diagram that includes a thorough explanation of all involved components.