**ASSIGNMENT 01**

**QUESTION 01(A)**

Many software problems arise during the development of web-based systems. Among the common ones are listed below.

* **Scalability Issues**:the system cannot handle increasing traffic and, thus e.g. performance declines This translates into a real challenge for web-based applications with unpredictable ways in which users will behave.
* **Security Vulnerabilities**: Web applications can contain multiple security vulnerabilities. Data security is a constant effort.
* **Compatibility Errors:**A web application is expects to work on multiple web browsers, as well as from your mobile devices. While implementing their projects, developers might ignore the differences in rendering engines and device specs causing compatibility issues.
* **Lack Of User Experience (UX**):If a website takes time to load, have confusing navigation, and difficult to reach features; users simply lose interest.
* **Installation Issues:** Web systems usually have external APIs, services, or databases integrated into them. It takes away the ability to integrate those systems in a timely manner, and in worse case scenarios makes it impossible when systems are incompatible or documentation is poor.
* **Versioning and Deployment Mess:**Especially with large teams, managing the code version and deployment pipeline becomes a nightmare which leads to inconsistent functionality across various environments

**QUESTION 01(C)**

The following categories software problems in web-based system development contain root causes:  
  
**Inadequate Proper Planning and** Architecture: Though one of the most significant factors leading to poor performance as traffic grows is not developing for proper scalability planning, poor architecturalchoices tend to create brittle, very hard to maintain code.

**Solution:** Proper architectural design, load testing, and scalability planning early in the development phase can avoid potential performance issues. Elastic scaling will also be beneficial in cloud platforms like AWS or Azure.

**Security Oversights** : Security vulnerabilities are often due to neglect of proper security practices, including input validation, data encryption, or secure session management.  
**Solution:**Implementation of secure coding practices, regular vulnerability assessments, adoption of frameworks that automatically handle security concerns (Django or Laravel), and so on tend to minimize the risk involved in safety.

**Inadequate Testing and Browser Compatibility Checks**: The diversity in user devices and browsers can result in compatibility issues if not carefully tested across multiple platforms.

**Solution**: Automated cross-browser testing tools and employing responsive design practices ensure that the web application works seamlessly across different devices and browsers.

**Poor Communication and Documentation in Integrations**: Problems often arise when integrating third-party services or APIs due to inadequate documentation or changing API specifications.

**Solution**: Developers have to use accurate fully explained APIs and set up clear integration protocols. Maintaining compatibility needs suitable contract testing and API versioning.

**Version Control and Deployment Mismanagement**: Issues with inconsistent environments during deployment or version control conflicts often arise in larger teams.

**Solution**: Version control systems like Git, when combined with pipelines for continuous integration (CI) and continuous installation , are useful in simplifying deployment methods, promising consistency across environments and decreasing human error.

**QUESTION 01(B)**

Below is a simple representation of the sources of problems in web-based system development

+-----------------------------+

| Sources of Software Problems|

+-----------------------------+

| 1. Scalability Problems |

| 2. Security Vulnerabilities |

| 3. Load Handling |

| 4. Load Balancing |

| 5. Compatibility Issues |

| 6. Incomplete Requirements |

| 7.Data Breaches And Attacks |

| 8. Integration Challaneges |

+-----------------------------+

**QUESTION 02**

For example, take an **E-commerce System** like Amazon, eBay, or a small online store. E-commerce systems have two primary stakeholders: the **client/organization** (business owner) and the **end user** (customer). Both have different requirements and expectations from the system, and the system should be designed to meet both the requirements to ensure its successful operation and satisfaction.

**1. Needs and Expectations of the Client/Organization (Business Owner)**

The client (that is, the business or organisation) has functional and operational requirements from the e-commerce system so that it be fully functional, grow in time, and compete at the market.

**-Scalability:** With growth, the system must scale up more products, customers, and transactions. It further provides for high performance regarding system traffic, such as the Black Friday or holiday seasons.

* **Expected Characteristics:** It should be cloud-based, or have a scalable architecture such that the increased traffic can be easily handled and an increased load on the databases.

**- Inventory Handling**: Track the product in real-time so that over-selling doesn't happen; it automatically handles the level of stock, back order, and restocking levels of products.

* **Expected Characteristics:** It should be able to integrate with the back-end system in such a way that the inventory of every transaction gets updated while sending alerts for low stock products.

**- Order Management:** Should handle orders efficiently. This is updating orders in real time and shipment tracking while handling the return or refund from a customer.

* **Expected Characteristics:** Order tracking through an easy-to-use admin panel, automatic email notifications to customers and admins, and clear return or exchange management.

**- Analytics and Reporting**: Business owners will need insight into customer behavior, sales performance, and inventory health to make data-driven decisions.

* **Expected Characteristics**: Analytics tools that integrate dashboards for sales trends, customer demographics, and product performance.

**- Security and Compliance:** E-commerce systems are also anticipated to comply with all relevant data protection laws regarding data security of a customer. They must also be secure while processing online payments from customers for the better interest of consumer privacy.

* **Expected Features:** SSL certificates, compliance to PCI-DSS regarding the payment gateways, and multi-factor authentication for all user accounts.

**2. Requirements and Expectations of the End User (Customer)**

The end users will be the customers of the system who shall browse the products, make their purchases, and seek help post-purchase. Therefore, the outcome of the e-commerce system will depend directly on the satisfaction of the end users.

**- User Experience (UX):** The interface of the system shall be seamless and easy to use for the customers while they browse, search for, and buy the products.

* The main expected **characteristic** is the simplistic nature of easy navigation, quickly access the search, responsiveness even up to smartphones, and easily checkout, few steps of checkout as well.

**- Security and Privacy**: Their personal data such as Credit Card Details, Shipping Information, password of the accounts will get stored and forwarded in absolute security.

* Expected **Characteristics** for Security and privacy are :

HTTPS encryption in the frontend

Safe Gateways like Paypal and Stripe at the payments

Data protection legislations or CCPA, GDPR).

**- Speed and Performance**: Consumers are impatient and will not sit for long on a slow page when they are trying to check out. Speed is paramount in reducing cart abandonment.

* **Expected Characteristics:** Optimized images, caching, fast and reliable hosting infrastructure, and a low load time on product pages and checkout.

**- Customer Support:** Easy access to help is expected for issues, whether related to inquiries regarding the product or returning a product or even concerns over delivery.

* **Expected Characteristics**: Live chat with an integrated feature, the availability of a help desk and ticketing system, FAQs, and accessible contacts through customer service via the chat, email, phone, etc.

**- Flexibility in Payment Methods:** In return, the ease to pay using credit/debit cards, digital wallets, or transferring from a bank account.

* **ExpectedCharacteristics**: Multiple payment methods (Visa, Mastercard, PayPal, Apple Pay), and finance/instalment option for more expensive products.

**3. Use Case Diagram System**

A **Use Case Diagram** is a graphic representation of the way in which a system interacts with its users, or actors, to carry out particular tasks or functions, use cases. The following is a very simple Use Case Diagram of the **E-commerce System**, which is a simplified depiction based on the needs and expectations of both partie

**Admin**

**Customer**

**The Customer Use Cases include:**

Browse Products Customer searches and filters products within specified categories, price, among others.

**- Add to Cart:** Users can add selected products into their cart for future purchasing.

**- Checkout:** Customers view their cart, fill in shipping information, and then make the payment.

**- Make Payment**: Customers will pay for their orders through a secure payment gateway.

**- View Order:** Customers can trace orders, review the order history, or request returns.

**Admin Use Cases:**

**Manage inventory**-Admin users will be able to add, update, or delete products in the inventory.

**Manage orders-**Admins can see and process orders coming from customers and shipping and delivering.

**Manage returns-**Return customer goods and issue refund based on the organization's policy.

**View analytics-**The admins view data on the sales trends, demographics of customers, and the products' performance.

**Manage Discounts:** Administrators can set up promotional codes, discounts, and special offers for customers.

- Payment Gateway Use Cases:

**Process Payment:** The system will interact with the external payment gateways (PayPal, Stripe, etc.) to securely process payments.

**Secure Payment:** Secure handling of credit card and transaction details.

**Process Refunds:** Process payment reversals when there is a return or cancellation of an order.

**Confirmed Payment**: It sends a message as soon as the transaction gets accepted which also sends back alert to the customer as well as administrator side.

**Conclusion:**

The **E-Commerce System** will also be responsive to every unique need of an end **user/customer** as well as that required of a business owner /clients.

The developed **e-commerce system** is well secure, easy to use, extensible as well as provides perfect on-the-spot transaction/communication methods in case of stock administration without misplacing an accurate customers updating system.

We can define and visualize the way various stakeholders interact with the system to achieve their goals by using a **Use Case Diagram,** and thereby ensure that the functionalities of the system are in accordance with stakeholder expectations.

It helps the business grow in terms of business, customers, and operational efficiency.