HLD(HIGH-LEVEL DESIGN) DOCUMENT for FLIPKART:

High-Level Design (HLD) refers to the process of designing the architecture of a software system at a conceptual level. It is a critical phase in the software development life cycle that occurs after the system requirements have been gathered and analyzed. During the HLD phase, the focus is on creating a high-level representation of the entire system, defining its major components, their interactions, and the overall structure.

Login Page:

Functional Requirements and Non-Functional Requirements:

**1.User Authentication:**

* The login page must authenticate users based on their provided credentials (username/email and password).

**2.User Registration:**

* If users don't have existing credentials, they should be able to register through the login page.

**3.Forget Password:**

* Users should have an option to rest their passwords in case they forget them.

**4.Session Management**:

* The Login page should manage user sessions securely , including session timeout and session termination features.

**5.Access Control:**

* The system must restrict access to authenticated users only and may implement role-based access control if necessary.

**6.Error Handling:**

* The login page must handle various error scenarios gracefully, such as incorrect credentials, expired sessions, and system errors.

**1.Security:**

* The login page must ensure the confidentiality, integrity, and availability of user credentials and sessions. It should employ measure like encryption, hashing passwords, and protection against common attacks like brute force and SQL injection.

**2.performance:**

* The login page should load quickly and respond promptly to user actions, even under high traffic conditions.

**3.Usability:**

* The login page must have an intuitive and user-friendly interface, with clear instructions and error messages. It should be accessible to users with disabilities.

**4.Compatibility:**

* The Login page should be compatible with different web browsers and devices, ensuring a consistent experience for all users.

**5.Scalability:**

* The system should be able to handle a growing number of users and sessions without compromising performance or security.

**6.Logging and monitoring:**

* The system should login attempts, including successful and failed ones, for security and audit purposes. it should also provide monitoring capabilities to detect unusual login patterns or security branches.

**Product CatLog:**

**1.Product Display:**

* Display product images, names, descriptions, and prices.
* Categorize products into relevant categories and subcategories.
* Provide filtering options to allow users to refine product searches.

**2.Search Functionality:**

* Implement a robust search engine to enable users to find products efficiently.
* Support keyword-based searches and provide relevant suggestions.
* And supports image search.

**3.Product Details Page:**

* Allow users to view detailed information about a product when selected.
* Display product specifications, reviews, and related items.

**4.User Authentication and Authorization**:

* Require user authentication to access certain features.
* Implement authorization to ensure that users have appropriate access levels.

**5.Shopping Cart:**

* Enable users to add products to their shopping cart.
* Allow users to view and modify the contents of their cart.

**6.Checkout Process:**

* Implement a secure and user-friendly checkout process.
* Include options for different payment methods.
* Provide order confirmation and tracking information.

**7.Wishlist:**

* Allow users to create and manage a wishlist of desired products.

**8.User Reviews and Ratings:**

* Enable users to leave reviews and ratings for products.
* Display average ratings and relevant feedback on product pages.

**9.Integration with Inventory Management:**

* Ensure real-time synchronization with the inventory system to reflect accurate product availability.

**10.Promotions and Discounts:**

* Support promotional offers, discounts, and coupon codes.
* Display discounted prices and promotion details on product listings.

**1.Performance:**

* Ensure fast loading times for product pages and search results.
* Handle a large number of simultaneous users and product listings.

**2.Scalability:**

* Design the system to scale horizontally to accommodate growth in user base and product CatLog.

**3.Reliability:**

* Implement measures to ensure high availability and minimal downtime.
* Have mechanisms for data backup and recovery.

**4.Security:**

* Implement secure user authentication and data encryption.
* Protect against common security threats such as SQL injection and cross-site scripting.

**5.Usability:**

* Design an intuitive and user-friendly interface for both desktop and mobile platforms.
* Ensure accessibility for users with disabilities.

**6.Compatibility**:

* Support multiple web browsers and devices to ensure a consistent experience.

**7.Maintainability:**

* Design the system with modularity and code maintainability in mind.
* Provide documentation for developers and administrators.

**8.Compliance:**

* Ensure compliance with data protection regulations and industry standards.
* Regularly update the system to meet evolving compliance requirements.

**9.Monitoring and Logging:**

* Implement monitoring tools to track system performance and detect anomalies.
* Maintain comprehensive logs for troubleshooting and auditing purposes.

**10.Interoperability:**

* Integrate with external systems, such as payment gateways and inventory management systems.

**Wishlist and save for later:**

**1.User Authentication:**

* Users must be authenticated to utilize the Wishlist and Save for Later features.
* Users should be able to log in and log out securely.
* The system must verify user identity before allowing access to the Wishlist and Save for Later functionalities.

**2.Wishlist Management:**

* Users should be able to create, view, edit, and delete items in their wish list.
* Users can add products to their wish list from product pages.
* The system should allow users to organize and categorize wish list items.
* Users can set quantities and priority levels for wish list items.

**3.Save for Later Functionality:**

* Users can move items from the shopping cart to a "Save for Later" list.
* Users should be able to easily transfer items between the shopping cart and "Save for Later."
* The system should retain the state of items saved for later across user sessions.

**4.Notifications:**

* Notify users about changes in their wish list or saved items.
* Users receive notifications for successful additions, deletions, or edits in their wish list.
* Optionally, users can opt to receive alerts about price drops or availability changes for wish list items.

**5.Integration with Checkout:**

* Seamless integration with the checkout process.
* Users can directly move items from the wish list or "Save for Later" to the shopping cart during the checkout process.
* The system should update the cart and reflect changes in real-time.

**1.Performance:**

* Ensure fast response times for wish list and "Save for Later" operations.
* The system should load and display wish list items quickly.
* Operations like adding or removing items should be near-instantaneous.

**2.Scalability:**

* Ability to handle a large number of wish list and "Save for Later" items.
* The system should scale horizontally to accommodate an increasing user base and growing lists.

**3.Security:**

* Protect user data and wish list information.
* Implement secure authentication and authorization mechanisms.
* Use encryption for sensitive user data.

**4.Reliability:**

* Ensure the availability and reliability of wish list and "Save for Later" features.
* Minimize system downtime during updates or maintenance.
* Implement measures for data backup and recovery.

**5.User Experience**:

* Provide an intuitive and user-friendly interface.
* The UI should be easy to navigate, allowing users to manage their wish list effortlessly.
* Support responsive design for a consistent experience across devices.

**Shopping cart and checkout:**

**1.Add and Remove Items:**

* Users should be able to add products to the shopping cart and remove items.
* Users can add products to the cart from product pages.
* The system should allow users to modify the quantity of items in the cart.
* Users can easily remove items from the cart.

**2.View Cart:**

* Users can view the contents of their shopping cart.
* The system should display product details, quantities, and total prices.
* Users can see a summary of the items in the cart at any time.

**3.Save Cart State:**

* Persist the state of the shopping cart across user sessions.
* The system should save the user's shopping cart to provide a consistent experience across visits.
* Ensure that the cart state is retrievable even after a user logs out and logs back in.

**4.Integration with Wishlist and Save for Later:**

* Seamless integration with the Wishlist and Save for Later features.
* Users can easily move items between the shopping cart and Wishlist/Save for Later.
* The system should update the cart and reflect changes in real-time.

**5.Proceed to Checkout:**

* Users can initiate the checkout process from the shopping cart.
* The system should provide a clear and user-friendly path to proceed to checkout.
* Users can review and confirm their order before payment.

**6.Multiple Payment Options:**

* Support various payment methods for checkout.
* Users can choose from different payment options such as credit/debit cards, digital wallets, net banking, and Cash on Delivery (COD).
* Ensure secure and encrypted payment transactions.

**7.Order Summary:**

* Display a comprehensive summary before finalizing the order.
* Users can review the list of items, quantities, and the total cost before confirming the order.
* Include any applicable taxes, shipping costs, and discounts in the order summary.

**1.Performance:**

* Ensure fast response times during cart operations and checkout.
* The system should process additions, removals, and updates to the shopping cart quickly.
* Checkout processes should be efficient to provide a smooth user experience.

**2.Scalability:**

* Ability to handle a large number of simultaneous users and transactions.
* The system should scale horizontally to accommodate increased user traffic during peak times.

**3.Security:**

* Protect user data, especially during payment transactions.
* Implement secure authentication and authorization mechanisms.
* Use encryption for sensitive user and payment information.

**4.Reliability:**

* Ensure the availability and reliability of the shopping cart and checkout features.
* Minimize system downtime during updates or maintenance.
* Implement measures for data backup and recovery.

**5.User Experience:**

* Provide an intuitive and user-friendly interface.
* The UI should make it easy for users to navigate the shopping cart and checkout processes.
* Support responsive design for a consistent experience across devices.

**Promotional And Discounts:**

**1.Promotion Display:**

* Display promotions and discounts prominently on the platform.
* Promotions should be visible on product listings and details pages.
* The system should highlight discounted prices and applicable promotions.

**2.Coupon Code Redemption:**

* Allow users to redeem coupon codes during the checkout process.
* Users can enter coupon codes to avail additional discounts.
* The system should validate and apply valid coupon codes.

**3.Deal of the Day:**

* Feature a daily deal with special discounts.
* The system should showcase a prominent "Deal of the Day" section.
* Discounts for the deal should be automatically applied during checkout.

**4.Flash Sales and Limited-Time Offers:**

* Implement flash sales and limited-time offers.
* The system should display products with limited-time discounts.
* Users should be informed about the remaining time for flash sales.

**5.Bundle Deals and Package Discounts:**

* Provide discounts for bundled products or packages.
* The system should offer discounts when users buy specific combinations of products.
* Bundle deals should be clearly communicated to users.

**1.Performance:**

* Ensure fast loading times for pages displaying promotions and discounts.
* Promotions should load quickly to enhance user experience.
* The system should handle high traffic during peak promotion periods.

**2.Scalability:**

* Ability to handle increased user activity during promotion events.
* The system should scale horizontally to accommodate a surge in users.
* Handle simultaneous requests for promotions and discounts without degradation in performance.

**3.Security:**

* Ensure secure validation and application of discounts.
* The system should secure coupon code transactions to prevent abuse.
* Validate and apply discounts securely during checkout.

**4.Reliability:**

* Ensure the reliability of promotions and discounts features.
* Minimize system downtime during updates or maintenance.
* The system should recover gracefully from failures to avoid disruptions during promotions.

**5.User Experience:**

* Provide an intuitive and positive user experience.
* Promotions and discounts should be clearly communicated with relevant visual elements.
* The system should present users with personalized promotions based on their preferences and purchase history.

**6.Integration with Inventory Management:**

* Ensure accurate tracking of inventory during promotions.
* The system should update inventory levels in real-time as products are purchased during promotions.
* Prevent overselling of products that are part of limited-time offers.

**Costing Limitations:**

**1.Development Costs:**

* Labour costs for developers, designers, and other team members involved in building and maintaining the platform.
* Costs for development tools, software licenses, and resources.

**2.Infrastructure Costs:**

* Expenses related to servers, databases, and cloud services required to host and manage the platform. Costs associated with data storage, processing, and network infrastructure.

**3.Security Costs:**

* Costs associated with implementing robust security measures to protect user data and the platform from cyber threats. Expenses for regular security audits and compliance with data protection standards.

**4.Scalability Costs:**

* Expenses for designing the platform to scale horizontally or vertically to accommodate increased user traffic and data volume. Costs for additional hardware, load balancing, and other scalability measures.

**5.Maintenance and Support Costs:**

* Ongoing costs for maintaining and updating the platform.
* Expenses for customer support, bug fixes, and updates.

**6.Integration Costs:**

* Expenses related to integrating the platform with external systems, APIs, or third-party services.
* Costs for ensuring seamless communication between different components.

**7.User Interface (UI) and User Experience (UX) Costs:**

* Costs for designing and implementing an intuitive and visually appealing user interface.
* Expenses for usability testing and user feedback analysis.

**8.Training Costs:**

* Costs associated with training staff on platform usage, maintenance, and support.
* Training costs for customer support teams.

**9.Compliance Costs:**

* Expenses related to ensuring that the platform complies with legal and regulatory requirements in various regions.
* Costs for staying updated on changing compliance standards.

**10.Backup and Disaster Recovery Costs:**

* Expenses for implementing and maintaining backup and disaster recovery mechanisms to protect against data loss.

**11.Monitoring and Analytics Costs:**

* Costs for implementing monitoring tools and analytics to track the performance and usage of the platform.
* Expenses for deriving insights from user behaviour and system metrics.

**12.Upgrade and Technology Refresh Costs:**

* Expenses for upgrading technologies and frameworks to stay current with industry standards.
* Costs associated with adopting new technologies for improved performance or features.

**13.Testing Costs:**

* Expenses for testing activities, including automated testing tools, testing environments, and resources for manual testing.

**Architecture Overview**

**1.Client-Side :**

* The client-side represents the user interface that customers interact with.
* Web Interface: Flipkart's website for desktop and mobile browsers.
* Mobile Applications: Native or hybrid mobile apps for iOS and Android.

**2. Web Server:**

* The web server handles user requests and serves web pages.
* Web Application: Manages user sessions, handles product searches, and serves dynamic content.
* Load Balancer: Distributes incoming traffic across multiple servers for scalability and fault tolerance.

**3. Application Server:**

* The application server handles business logic, user authentication, and communication with databases.
* User Authentication: Manages user login sessions, authentication, and authorization.
* Product Catalog: Manages product information, categories, and inventory.
* Shopping Cart and Checkout: Handles user interactions related to shopping carts and the checkout process.
* Promotions and Discounts: Manages promotional offers, discounts, and coupon codes.

**4. Database Management System (DBMS):**

* The DBMS stores and retrieves data related to products, users, orders, and more.
* Product Database: Stores information about products, including details, prices, and availability.
* User Database: Contains user profiles, preferences, and order history.
* Order Database: Stores information about customer orders, including items, prices, and delivery details.

**5. Cache:**

* Caching mechanisms to improve performance and reduce database load.
* Content Cache: Stores frequently accessed product details, reducing response times.
* Session Cache: Caches user session information for faster authentication checks.

**6. External Services:**

* Integration with external services for functionalities like payment processing and shipping.
* Payment Gateway: Handles secure payment transactions.
* Shipping Services: Integrates with logistics providers to manage order fulfillment and delivery.

**7. Security Layer:**

* Ensures the security of user data, transactions, and the overall platform.
* Encryption: Encrypts sensitive data during transmission and storage.
* Security Protocols: Implements secure protocols for user authentication and authorization.

**8. Analytics and Monitoring:**

* Tools for tracking user behavior, monitoring system health, and generating insights.
* Analytics Engine: Analyzes user data to derive insights for business decisions.
* Monitoring Tools: Track system performance, identify issues, and ensure reliability.

**9. Content Delivery Network (CDN):**

* Optimizes content delivery by caching static assets closer to users.
* CDN Servers: Distribute and cache static content (images, stylesheets) for faster loading times.

**10. Search and Recommendation Engine:**

* Powers the product search functionality and provides personalized recommendations.
* Search Engine: Enables users to search for products based on keywords.
* Recommendation System: Analyzes user behavior to suggest relevant products.

**11. Infrastructure:**

* The underlying infrastructure supporting the entire system.
* Servers: Physical or virtual machines hosting web, application, and database servers.
* Networking: Manages communication between different components.

**12. Load Balancing and Scalability:**

* Ensures even distribution of user requests and scalability to handle increased traffic.
* Load Balancers: Distribute incoming traffic across multiple servers.
* Scaling Mechanisms: Horizontal and vertical scaling to accommodate varying loads.

**13.Backend:**

* The backend can be built using a microservices architecture with technologies like

Java,python,node.js (JavaScript),Ruby, PHP, Scala, C#.

**Architecture diagram:A paper with text on it

Description automatically generated**