Requirement Gathering and Analysis Phase
Solution Requirements (Functional & Non-functional)

Date	22 July 2024
Team ID	SWTID1720760336
Project Name	BookStore- Where Stories NEstle
Maximum Marks	

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Site
FR-2	User Confirmation	Confirmation via Email Confirmation via Password
FR-3	User Authentication	Login with Email and Password
FR-4	User Profile Management	View Profile Edit Profile
FR-5	Product Catalog	Browse Products Search Products
FR-6	Product Details	View Product Details
FR-7	Shopping Cart	Add Products to Cart View Cart Update Cart (Change Quantity, Remove Items)
FR-8	Checkout and Payment	Enter Shipping Information Choose Payment Method Order Summarisation
FR-9	Order Management	View Order History View Order Details Cancel Order
FR-10	Admin Product Management	Add New Products Edit Existing Products Delete Products
FR-11	Admin Order Management	View All Orders Update Order Status
FR-12	Admin User Management	View All Users Edit User Details Delete Users
FR-13	Security	Role-Based Access Control Data Encryption
FR-17	Wishlist	Add Products to Wishlist View Wishlist Remove Products from Wishlist

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The React-based front end should provide an intuitive and responsive user interface, ensuring ease of use and a smooth user experience across various devices and screen sizes. It should follow best practices in web design and accessibility standards to cater to a diverse user base, including those with disabilities.
NFR-2	**Security**	Utilizing AWS's security features, the application must ensure data protection through secure communication protocols (e.g., HTTPS), encryption of sensitive data at rest and in transit (e.g., using AWS KMS for encryption keys), and secure authentication mechanisms (e.g., AWS Cognito). The system should protect against common web vulnerabilities such as SQL injection, XSS, and CSRF. Role-based access control should be implemented to restrict access to sensitive administrative functions.
NFR-3	**Reliability**	The application should leverage AWS's infrastructure to ensure high reliability, including the use of multiple availability zones for MongoDB to prevent data loss and maintain data integrity. The system should handle errors gracefully and include mechanisms for automatic backup and recovery, utilizing AWS Backup services for regular backups and automated recovery procedures.
NFR-4	**Performance**	The application should provide a fast and responsive user experience, with optimized page load times and server response times. AWS services like CloudFront for content delivery and EC2 instances for scalable compute power should be utilized to handle a large number of concurrent users and transactions efficiently. Performance metrics should be monitored using AWS CloudWatch to ensure they remain within acceptable thresholds.
NFR-5	**Availability**	High availability should be ensured using AWS's infrastructure, with features like auto-scaling groups for handling traffic spikes, and multi-region deployments for failover and disaster recovery. The application should be accessible to users 24/7 with minimal downtime, utilizing AWS services for redundancy and failover mechanisms. Scheduled maintenance should be conducted with minimal disruption to users, leveraging rolling updates and blue/green deployment strategies.

NFR-6	**Scalability**	The application should be designed to scale horizontally and vertically, leveraging AWS's scalable infrastructure. MongoDB on AWS should be configured to handle increasing data volumes and user loads using sharding and replica sets. The application should support auto-scaling for both the web servers and the database, ensuring it can handle increased demand seamlessly. The architecture should allow for easy addition of new features and functionalities without significant rework.