Project Proposal Information

Title: E-commerce Platform with Kubernetes and AWS Services.

Context:

I need to recommend the development of a modern e-commerce platform. The platform will be built the usage of Java, taking advantage of the Spring Framework for its sturdy functions and simplicity of improvement. To make certain scalability, flexibility, and green aid control, the software might be deployed on Kubernetes, permitting seamless field orchestration and automated scaling primarily based on demand. Additionally, AWS services will play a pivotal role on this project, offering a reliable, steady, and fantastically to-be-had cloud infrastructure. AWS Elastic Beanstalk or Amazon EKS will host the utility, even as Amazon RDS or Amazon DynamoDB might be applied for coping with the database. Amazon S3 will save media files, and Amazon CloudFront will supply content material successfully.

The platform will be characteristic a consumer-friendly frontend constructed the use of cutting-edge web technology like React, making sure an unbroken and intuitive buying revel in. We will also incorporate important functionalities along with person registration and authentication, product catalog surfing, searching, cart management, steady checkout, and personal opinions. In addition to assembling these requirements, the challenge will put in force design patterns like Singleton and Observer, providing an extensible and maintainable codebase. We will adhere to industry first-class practices in database layout, making sure a scalable and green records tier. Rigorous unit testing will be applied using JUnit, and complete documentation will accompany the mission to facilitate destiny improvements and maintenance. The completion of this project will empower us to create an innovative and characteristic e-commerce presence, ultimately improving customer satisfaction in the virtual marketplace.

Detailed Use Cases:

- User Registration and Authentication
- Browse Product Catalog
- Search Products
- Add/Remove Products to/from Cart.
- Checkout and Payment Processing
- Order History and Tracking
- User Reviews and Ratings
- Admin Panel for Product Management

Roles and Responsibilities: Administrator

Manage the product catalog:
- Adding new products.
- Refresh product details such name, description, cost, and availability.
- Eliminate items from the catalog.

Manage orders:
- viewing and handling incoming ones.
- Provide shipment tracking information and update the order status (processed, dispatched, delivered, etc.).
- Manage cancellations of orders

user management:
- Create and manage user accounts
- Take care of user account-related concerns, such as account deactivation and password resets.

Support for customers:
- Respond to their questions and grievances, fixing problems.
- Help with technical issues, order tracking, and general questions.

Security and Compliance:
- Check to see that the platform complies with data protection laws and security standards.
- Take action to stop data breaches and illegal access.

Roles and Responsibilities: User (Customer)

Searching and Browsing:
- Look through the product catalog.
- Use filters, categories, or keywords to search for specific products.

Account management:
- Set up a user profile for more individualized shopping.
- Update the account's information, including your shipping and payment information.
- Add items to the shopping basket so you may buy them later.
- Proceed to the checkout page and enter your payment information to complete your order.

Order tracking:
- Keep track of the status of placed orders, including the movement of shipments.

Reviews and ratings of products:
- Post reviews and ratings of the goods you've bought.

Favorites and Wishlist's:
- Add your favorite products to a Wishlist for future use.

Support and Feedback:
- For questions, help, or problem solving, contact customer support.

Privacy and Security:
- Make sure that account credentials and personal data are secure.

Technology/tool to be used:

Services and Cloud Technologies:

• AWS Elastic Beanstalk or Amazon EKS (Kubernetes): Used for setting up, controlling, and handling auto-scaling for application containers.
• Amazon RDS (Relational Database Service) or Amazon DynamoDB: Used for database management, storing user data, order information, and product information.
For storing and serving media files, such as product photographs, use Amazon S3 (Simple Storage Service).
• Amazon CloudFront: To accelerate website performance through content caching and deliver content more effectively.
• AWS Lambda: This platform offers serverless functions that can be used for certain backend activities.
For delivering transactional emails like order confirmations and password reset emails, use Amazon SES (Simple Email Service).
• Amazon CloudWatch: For keeping track of infrastructure and application performance.

Backend Technologies:

For creating the backend business logic and services, use Java.
• Spring Framework: To take advantage of Spring's strong capabilities, such as data access, web development, and dependency injection.
• Object-Relational Mapping (ORM) and database interactions can be accomplished using Hibernate or JPA (Java Persistence API).
• Spring Boot: To quickly construct and set up the backend application.
• RESTful APIs: For frontend and backend service communication.
• Swagger/OpenAPI: To facilitate integration and provide API documentation.
• JUnit: For creating functional and quality unit tests for code.
For secure user authentication and authorization, utilize JWT (JSON Web Tokens).

Frontend Technologies:

• React: For creating the interactive and dynamic user interface.
For front-end markup, styling, and interactivity, use CSS and JavaScript.
• Redux: For the front-end application's state management.
• For sending HTTP queries to the backend services, use the Axios or Fetch API.

Development and Testing Tools:

• Git: used for collaborative development and version control.
• Maven or Gradle: For automating build processes and dependency management.
• Containerizing application components with Docker will ensure consistency between environments.
• Swagger UI or Postman: For testing APIs while they are being developed.