Hackathon Day 2 Planning the Technical Foundation.

Name: Ayesha Nazeer

Roll Number: 00383242

Technical Planning Documents Overvie

Key Technology

Frontend: Next.jsBackend: APIs

• Database: MongoDB

• CMS: Sanity

• Payment Gateway: Stripe

• Shipping Integration: ShipEngine

Technical Architecture

- Microservices architecture
- API-first approach
- Scalable and secure design

System Overview

- E-commerce platform for buying and selling furniture.
- Key components:
 - 1. User management
 - 2. Order management
 - 3. Payment management
 - 4. Shipment management
 - 5. Integration with Sanity CMS for content management
 - 6. Integration with ShipEngine for shipping management
 - 7. Integration with Stripe for payment processing

Database (MongoDB)

- NoSQL database for storing user data, orders, products, and delivery zones.
- Designed to be scalable and secure.

CMS (Sanity)

• Headless CMS for managing content

• Integration with frontend for displaying content

Order Tracking (ShipEngine)

• Integrated with ShipEngine for real-time tracking of orders.

Authentication (MongoDB)

• Secure and scalable user authentication using MongoDB.

Deployment

- Cloud-based deployment using AWS or Google Cloud.
- Designed for scalability and security.

System Components and Workflow

- User Management
- Order Management
- Payment Management
- Shipment Management
- Integration with Sanity CMS and ShipEngine

API Endpoints

- User Management:
 - o /users,/users/{id}
- Order Management:
 - o /orders,/orders/{id}
- Payment Management:
 - o /payments,/payments/{id}
- Shipment Management:
 - o /shipments,/shipments/{id}
- Product Management:
 - o /products,/products/{id}
- Category Management:
 - o /categories,/categories/{id}

Additional Features

1. Performance Optimization

- Caching Mechanisms: Implement caching (e.g., Redis) to speed up frequently accessed data.
- Lazy Loading: Enable lazy loading for images and content to improve frontend performance.
- **Efficient Pagination and Filtering:** Optimize data retrieval for large datasets, such as products and orders.

2. Scalability Planning

- Autoscaling: Configure AWS/Google Cloud for autoscaling to handle traffic spikes efficiently.
- **Independent Scaling:** Ensure microservices like users, orders, and payments can scale independently.

3. Analytics and Reporting

- Real-Time Analytics:
 - o Sales Reports: Track daily, weekly, and monthly sales.
 - User Activity Reports: Monitor user engagement and purchasing patterns.
 - o Shipping Performance: Analyze delivery times and delays.

4. Advanced Security Measures

- **OAuth 2.0:** Secure user login and session management.
- Payment Tokenization: Ensure Stripe uses tokenized payments to avoid storing sensitive card details.
- Rate Limiting: Implement API rate limiting to protect against DDoS attacks.
- Role-Based Access Control (RBAC): Set distinct access levels for admins, sellers, and customers.

5. Customer Experience Enhancements

- Advanced Search: Add smart search functionality with filters like category, price range, and brand.
- Recommendation Engine: Use AI or rule-based logic to suggest relevant products to users.
- Multilingual Support: Make the platform accessible in multiple languages.
- Mobile Optimization: Prioritize a mobile-first design for enhanced user experience.

6. Testing Strategy

- **Unit Testing:** Ensure individual components function as expected.
- Integration Testing: Validate the seamless connection between APIs and services.
- Load Testing: Test system performance under heavy traffic.
- User Acceptance Testing (UAT): Gather feedback from real users for improvements.

7. Future Enhancements (Roadmap)

- Loyalty Program: Introduce a rewards system for frequent buyers.
- **Subscription Model:** Add furniture rentals or maintenance services.
- Augmented Reality (AR): Enable users to virtually place furniture in their space.

Third-Party Integrations: Expand integrations with logistics partners or marketing tools.

8. Environmental Impact

- Sustainable Shipping: Offer eco-friendly delivery options.
- Carbon Offsetting: Dedicate a percentage of revenue to environmental initiatives.
- Recycling Programs: Encourage users to recycle old furniture responsibly.

Deployment Plan

- Deploy the platform on AWS or Google Cloud.
- Ensure the deployment is scalable, secure, and optimized for high availability.

Database Security Considerations

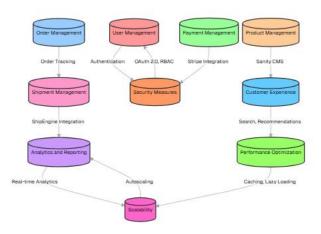
- **Data Encryption:** Encrypt sensitive data such as passwords and payment details.
- Access Control: Restrict database access to authorized personnel only.
- Security Audits: Conduct regular security audits and penetration tests.

Monitoring and Maintenance

- **Real-Time Monitoring:** Use monitoring tools to track system performance and uptime.
- **Regular Updates:** Implement continuous updates for security and functionality improvements.
- **Backup and Recovery:** Have a robust backup and disaster recovery plan.

Timeline

- 1. **Development:** 6 weeks.
- 2. **Testing and Debugging:** 2 weeks.
- 3. **Deployment:** 1 week.
- 4. Maintenance and Updates: Ongoing.



E-commerce Furniture Platform - System Architecture

