Problem Statement: Loan management system

Design and develop a comprehensive loan management system using Java, Spring Boot, and React to streamline the process of managing loan applications, approvals, disbursements, and repayments for a financial institution.

Key Features:

1. **User Authentication and Authorization**: Implement secure user authentication and authorization mechanisms to ensure only authorized users can access the system.
2. **Loan Application Management**: Allow borrowers to submit loan applications through a user-friendly interface. Capture essential details such as personal information, loan amount, purpose, and supporting documents.
3. **Loan Approval Workflow**: Design a workflow for loan approval, including stages such as application review, credit assessment, and approval/rejection by authorized personnel.
4. **Document Management**: Enable document upload and management, ensuring all necessary documents are securely stored and easily accessible throughout the loan approval process.
5. **Interest Calculation**: Implement an interest calculation module to determine the interest amount payable based on the loan amount, tenure, and applicable interest rates.
6. **Repayment Scheduling**: Generate repayment schedules for approved loans, specifying installment amounts, due dates, and payment methods.
7. **Payment Processing**: Integrate payment gateways to facilitate loan disbursements and automate installment collections from borrowers.

Problem Statement: Forex Management System

Problem Statement: Forex Management System

Design and develop a comprehensive Forex (Foreign Exchange) management system using Java, Spring Boot, and React to facilitate efficient management of currency exchange transactions, rates, accounts, and reporting for financial institutions, currency traders, and businesses involved in international trade.

Key Features:

1. **Currency Exchange Transactions:** Enable users to perform currency exchange transactions, including buying and selling foreign currencies, at real-time exchange rates. Provide a user-friendly interface for executing transactions securely and efficiently.
2. **Exchange Rate Monitoring:** Integrate with external APIs or financial data providers to fetch and display real-time exchange rates for various currency pairs. Allow users to monitor exchange rate fluctuations and trends over time.
3. **Account Management:** Enable users to manage their Forex accounts, including viewing account balances, transaction history, open positions, and account statements. Implement features for depositing funds, withdrawing funds, and transferring balances between accounts.
4. **Risk Management:** Implement risk management tools to help users assess and mitigate currency exchange risks. Provide features such as stop-loss orders, limit orders. Manage order lifecycle, including order placement, modification, and cancellation.
5. **Compliance and Regulatory Reporting:** Ensure compliance with regulatory requirements and reporting obligations related to Forex trading activities. Implement features for KYC (Know Your Customer) verification
6. **Market Analysis and Insights:** Provide users with access to market analysis tools, technical indicators, and financial news to support informed decision-making in Forex trading. Generate insights into market trends, volatility, and trading opportunities.
7. **Alerts and Notifications:** Set up alerts and notifications for users to stay informed about significant market events, price movements, and account activities. Notify users about margin calls, trade executions, and other important updates in real-time.

Project Title: Insurance Management System

Objective:

Develop a comprehensive web application for managing insurance policies, claims, and customer information. The backend will be built using Java Spring Boot, and the frontend will use React. The system should facilitate seamless interaction between customers, agents, and administrators, ensuring efficient and secure management of insurance-related operations.

Problem Statement:

Your team is tasked with creating an insurance management system that provides a robust and user-friendly platform for both customers and insurance administrators. The application should include functionalities for policy management, claims processing, customer support, and administrative oversight.

Functional Requirements:

1. **User Registration and Authentication:**
    - **Customers and agents should be able to register by providing necessary details (name, address, email, phone number, etc.).**
    - **Implement secure authentication mechanisms allowing users to log in using their credentials.**
    - **Enable password reset functionality via email verification.**
2. **Policy Management:**
    - **Users should be able to browse and purchase different types of insurance policies (health, auto, home, life).**
    - **Display detailed information about each policy, including coverage, premium, terms, and conditions.**
    - **Allow users to view their purchased policies, update personal information, and renew policies.**
3. **Claims Processing:**
    - **Enable customers to file insurance claims by submitting required details and documents.**
    - **Implement a workflow for claims processing, including claim submission, verification, approval/rejection, and settlement.**
    - **Notify users about the status of their claims via email and dashboard alerts.**
4. **Customer Support:**
    - **Integrate a support ticket system where users can raise issues or queries related to their policies or claims.**
    - **Allow users to view the status of their support tickets and communicate with support agents.**
    - **Implement an admin interface for managing and responding to support tickets.**

5. **Notifications and Alerts:**
    - **Send email notifications for important events such as policy renewal reminders, claim status updates, and support ticket responses.**
    - **Allow users to configure their notification preferences.**
6. **Administrative Functions:**
    - **Implement an admin dashboard to manage users, policies, claims, and support tickets.**
    - **Ensure administrators can generate reports on various aspects of the system, such as policy sales, claim statistics, and user engagement.**

Project Title: Payroll Management System with Tax Calculator

Objective:

Develop a comprehensive web application for managing payroll processes and calculating taxes for employees. The backend will be built using Java Spring Boot, and the frontend will use React. The system should streamline payroll management, automate tax calculations, and ensure compliance with relevant regulations.

Problem Statement:

Your team is tasked with creating a payroll management system that provides a robust and user-friendly platform for HR administrators and employees. The application should include functionalities for managing employee information, processing payroll, calculating taxes, and generating reports.

Functional Requirements:

1. **User Registration and Authentication:**
    - HR administrators should be able to register and manage employee accounts.
    - Implement secure authentication mechanisms allowing users to log in using their credentials.
    - Enable password reset functionality via email verification.
2. **Employee Management:**
    - HR administrators should be able to add, update, and remove employee details (name, address, email, phone number, job title, salary, etc.).

- Employees should be able to view and update their personal information.

3. **Payroll Processing:**
    - Automate payroll calculation based on employee salaries, hours worked, overtime, bonuses, and deductions.
    - Generate pay slips for each pay period and allow employees to view their pay slips.
    - Allow administrators to set up payroll schedules (weekly, bi-weekly, monthly).
4. **Tax Calculation:**
    - Implement tax calculation logic based on employee salary and applicable tax rates (federal, state, and local taxes).
    - Ensure compliance with current tax laws and regulations.
    - Allow employees to view their tax deductions and net salary after tax.
5. **Leave Management:**
    - Employees should be able to apply for leave and view their leave balance.
    - Administrators should be able to approve or reject leave requests.
    - Integrate leave balances and deductions into payroll calculations.
6. **Reporting and Analytics:**
    - Generate reports for payroll summaries, tax deductions, and employee earnings.
    - Provide analytics dashboards for administrators to monitor payroll expenses and tax liabilities.
    - Export reports in various formats (PDF, Excel).
7. **Notifications and Alerts:**
    - Send email notifications for important events such as payroll processing, tax deductions, and leave approvals.
    - Allow users to configure their notification preferences.

Problem Statement: Online Grocery E-commerce Platform

Design and develop an online grocery e-commerce platform using Java, Spring Boot, and React to provide customers with a convenient and seamless shopping experience for purchasing groceries and household essentials.

Key Features:

1. **Product Catalog:** Create a comprehensive catalog of grocery items, including fresh produce, pantry staples, beverages, snacks, and household essentials.
2. **User Registration and Authentication:** Allow users to register accounts and log in securely to access personalized shopping experiences, order history, and saved preferences.
3. **Browsing the products:** Enable users to browse products by categories, brands, and special offers.
4. **Shopping Cart:** Provide users with a virtual shopping cart to add, remove, and modify items before proceeding to checkout. Display real-time updates on cart contents and total order value.
5. **Checkout Process:** Streamline the checkout process with a step-by-step flow for entering delivery details, selecting payment methods, and confirming orders. Offer multiple payment options, including credit/debit cards.(Dummy Payments)
6. **Promotions and Discounts:** Implement promotional campaigns, discount coupons, and reward on each purchase to customers.
7. **Inventory Management:** Manage inventory levels and product availability in real-time to prevent out-of-stock situations and ensure accurate order fulfillment.
8. **Order Management:** Enable administrators to view and manage incoming orders, process payments, update order statuses, and handle order cancellations or refunds.

Problem Statement: Online Sportswear E-commerce Platform

Create an online sportswear e-commerce platform using Java, Spring Boot, and React to offer customers a convenient and immersive shopping experience for purchasing athletic apparel, footwear, and accessories.

Key Features:

1. **Product Catalog:** Develop a comprehensive catalog of sportswear products, including activewear, footwear, equipment, and accessories, from various brands and categories.

2. **User Registration and Authentication:** Implement user registration and authentication functionalities to enable customers to create accounts, log in securely, and access personalized features such as order history, wishlists, and recommendations.
3. **Browsing and Search:** Provide intuitive browsing functionalities that allow customers to explore products by categories (e.g., men's, women's, kids'), sports types (e.g., running, yoga, basketball), brands, sizes, and price ranges.
4. **Product Details and Reviews:** Display detailed product information, including descriptions, images, pricing, sizes, colors, and customer reviews. Allow customers to leave reviews and ratings to help others make informed purchase decisions.
5. **Shopping Cart and Checkout:** Enable customers to add items to their shopping carts, view cart contents, adjust quantities, and proceed to checkout seamlessly. Implement a secure checkout process with multiple payment options, order summaries, shipping address inputs, and order confirmation pages.
6. **Promotions and Discounts:** Create promotional campaigns, discount codes, seasonal sales, and loyalty programs to attract customers and incentivize purchases. Display promotional banners, pop-ups.
7. **Inventory Management:** Manage product inventory levels, stock availability, and product variants (e.g., sizes, colors) in real-time to prevent overselling and backorders. Implement automated inventory replenishment and low stock alerts to ensure timely restocking.
8. **Social Media Integration:** Integrate social media platforms (e.g., Facebook, Instagram, Twitter) linking to social media page.

Project Title: Expense Management System (Similar to Splitwise)

Objective:

Develop a comprehensive web application for managing and splitting expenses among friends, families, or groups. The backend will be built using Java Spring Boot, and the frontend will use React. The system should facilitate easy tracking, sharing, and settlement of expenses, ensuring transparency and simplicity.

Problem Statement:

Your team is tasked with creating an expense management system that provides a robust and user-friendly platform for individuals to manage and split expenses. The application should include functionalities for adding expenses, splitting costs, settling debts, and generating reports.

Functional Requirements:

1. **User Registration and Authentication:**
    - Users should be able to register by providing necessary details (name, email, password).
    - Implement secure authentication mechanisms allowing users to log in using their credentials.
    - Enable password reset functionality via email verification.
2. **Group Management:**
    - Users should be able to create groups and invite other users to join.
    - Allow users to join multiple groups and manage expenses within each group separately.
    - Provide functionality for group admins to manage group members (add, remove).
3. **Expense Management:**
    - Users should be able to add expenses with details like amount, date, description, and payer.
    - Enable users to specify how the expense should be split among group members (equally, unequally, by shares).
    - Allow users to attach receipts or images to expenses.
4. **Expense Splitting and Settlement:**
    - Automatically calculate each member's share of the expense and update balances.
    - Provide a clear summary of who owes whom and how much.
    - Enable users to record payments made to settle debts.
    - Allow users to settle multiple debts at once and track the settlement history.
5. **Notifications and Alerts:**
    - Send email notifications for important events such as added expenses, payments, and group invitations.
    - Allow users to configure their notification preferences.
6. **Reports and Analytics:**

- Generate reports for expenses and settlements within a group or between specific users.
- Provide analytics dashboards for users to monitor their spending patterns.
- Allow users to export reports in various formats (PDF, Excel).

7. **User Interface and Experience:**
    - Provide a dashboard for users to quickly view their balances, groups, and recent activities.