Mini Collection Management System - Assignment

Overview

Create a simplified collection management system that helps manage customer payments and notifications. This system should demonstrate your ability to work with full-stack technologies, real-time updates, and basic AI integration.

Detailed Requirements

1. Authentication System

- Implement user registration and login functionality
- JWT-based authentication

2. Customer Management

Features:

- CRUD operations for customer details
  - Name
  - Contact information
  - Outstanding payment amount
  - Payment due date
  - Payment status
- Bulk customer upload via Excel
  - Provide a template for upload
  - Validate data before import
  - Show import success/error summary
- List view with filtering and sorting options

3. Payment Management

- Create a mock payment API endpoint
- Mark payments as completed/pending
- Real-time payment status updates using WebSocket

4. Notification System

- Real-time notifications using WebSocket
- Notification types:
  - Payment received
  - Payment overdue
  - New customer added
- Notification center to view all notifications

Technical Requirements

Frontend (React.js/Next.js)

- Clean and responsive UI
- State management (Redux/Context API)
- Form validation
- Error handling

- Loading states
- Real-time updates
- File upload handling

Backend (Node.js)

- RESTful API architecture
- WebSocket implementation
- File handling for Excel uploads
- JWT authentication
- Input validation
- Error handling middleware
- Logging system

Database:

- SQL/Elasticsearch (Preferred Elasticsearch)

Documentation

- Swagger/OpenAPI documentation

Evaluation Criteria

1. Code Quality

- Clean, maintainable code
- Proper error handling
- Use of design patterns
- Code organization
- Comments and documentation
- Testing (unit/integration)

2. Functionality

- All features working as specified
- Real-time updates working correctly
- Excel upload functioning properly
- Proper error handling and user feedback
- Performance optimization

3. Technical Implementation

- Proper use of specified technologies
- Database design and optimization
- API structure and documentation
- Security implementation

4. UI/UX Design

- Intuitive interface
- Responsive design
- Loading states
- Error feedback

Submission Requirements

GitHub repository with:

- Complete source code

- README.md with:
  - Project overview
  - Setup instructions
  - Architecture Diagram
  - Technical decisions explanation
  - Future improvements
- .env.example file
- Docker configuration