Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

FACULTY KIT

Objective –

The goal of this faculty kit is to outline the roles, responsibilities, and tools necessary for successful project implementation of the **Online Toll Payment System**. This system allows users to pay tolls for highways or roads through an online platform, streamlining the payment process, reducing physical traffic at toll booths, and integrating payment gateways for secure transactions. The kit provides guidance for faculty to manage and oversee the development, progress, and testing of the system.

Requirements Specification –

The **Online Toll Payment System** will include:

- **User Authentication:** Secure login and user management for customers, toll booth operators, and administrators.
- **Toll Fee Calculation:** Automated calculation of toll charges based on vehicle type, distance traveled, and applicable discounts or charges.
- **Payment Integration:** A secure payment gateway to process toll payments using various methods like credit/debit cards, net banking, and wallets.
- **Transaction History:** Users can view their toll payment history and download receipts.
- **Admin Control:** Admins can manage toll rates, track payments, monitor system usage, and generate reports.
- **Notification System:** Users will receive notifications about toll charges, payment confirmations, and balance updates

Technology Familiarization –

The project will use **NodeJS** for backend development, enabling the implementation of REST APIs and database interactions. The frontend will be developed using **React.js**, providing a dynamic, responsive user interface. For the database, **SQL** and **NoSQL** databases will be used to store user information, transaction records, and toll fee configurations. Payment gateway integration will be implemented using or similar services.

Database Creation –

The **Online Toll Payment System** will utilize both relational and non-relational databases to handle different data types:

- **SQL Database** for structured data (user information, toll rates, transaction details).
- **NoSQL Database** for unstructured data, such as user logs or logs of system events.
- **User:** User details, role (Admin, Operator, Customer).
- **Transaction:** Information on toll payments, including payment status and receipts.
- **Toll:** Toll rate details, vehicle classifications, and distance-based charges.

High-Level and Detailed Design –

System Overview:

The system will consist of three main layers:

- **Frontend:** A React-based user interface allowing users to make toll payments, check history, and view toll rates.
- **Backend:** Nodejs will handle business logic, process transactions, and communicate with external services like payment gateways.
- **Database:** SQL (for user, transaction, and toll data) and NoSQL (for logs and non-structured data) will ensure scalability, performance, and reliability.

Detailed Design:

The backend will expose RESTful APIs for:

- User authentication.
- Toll calculation based on vehicle type, route, and applicable charges.
- Payment processing and receipt generation.
- Transaction and payment history retrieval.

The frontend will interact with the backend through HTTP requests, which will be handled dynamically update the user interface based on real-time data stored in the database.

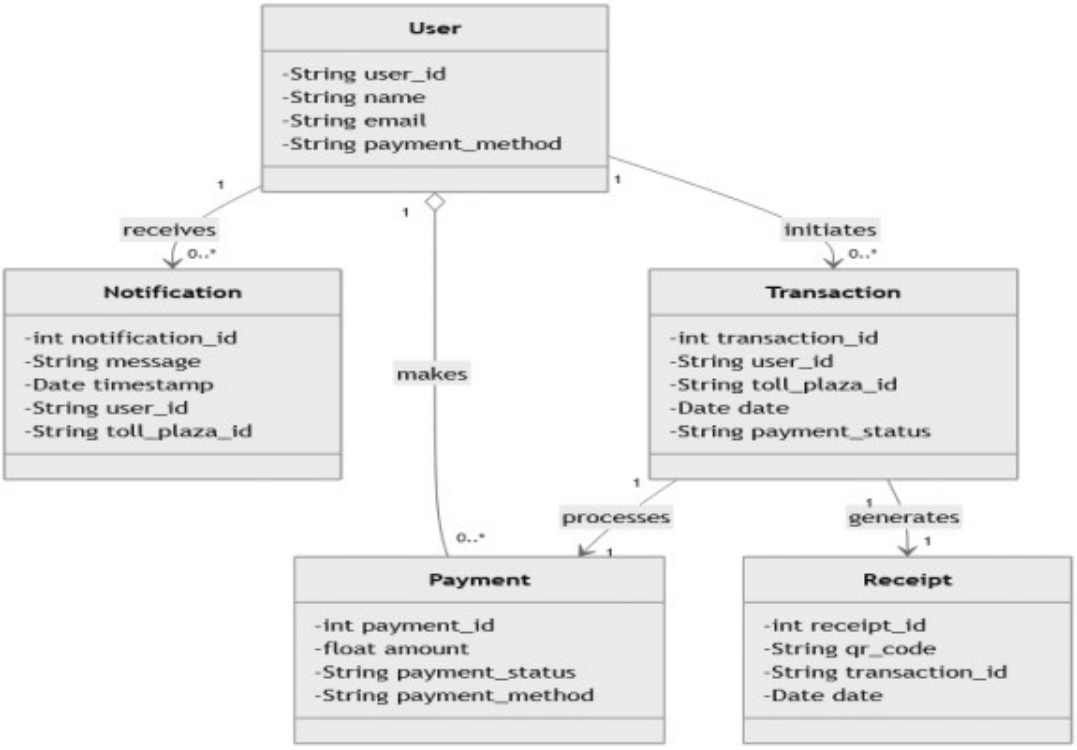

Fig :-Detailed Design

Frontend Implementation –

React.js will be used for building the user interface. The UI will have sections for login, toll fee calculation, payment gateway integration, transaction history, and notifications. Different user roles will have specific permissions and views:

- **Customers:** Pay tolls, view history, check balances, and manage payment methods.
- **Admins:** Manage toll rates, monitor transactions, and generate reports.
- **Toll Operators:** Track vehicle entries and exits, verify payments, and update toll data.

Integrating the Frontend with the Database –

To integrate the frontend with the backend and database, **Node.js** will expose REST APIs. The frontend will use **Axios** to make HTTP requests to the backend, enabling CRUD operations for fetching toll rates, processing payments, and retrieving transaction history. The backend will interact with both SQL and NoSQL databases to provide accurate and up-to-date data to the frontend

Test Plan Review -

Testing will be conducted at various stages to ensure the system functions as expected:

- **Unit Testing:** Backend logic for payment calculations, user authentication, and data validation.
- **Integration Testing:** Testing the communication between frontend and backend.
- **UI/UX Testing:** Ensuring the frontend is user-friendly and easy to navigate.
- **Performance Testing:** Verifying that the system can handle high traffic during peak toll payment times

Final Review -

At the conclusion of the project, a final review will be conducted to ensure the system meets all specified requirements. This will include:

- Validating user authentication and role management.
- Ensuring toll fee calculations are accurate.
- Testing the payment gateway and ensuring secure transactions.
- Ensuring transaction history and notifications function as expected.

User feedback will be gathered to refine the system for future improvements

Documents/References that May Aid the Process of Evaluation-

- **Node.js Documentation:** For backend implementation and API development.
- **React.js Documentation:** For frontend development and UI handling.
- **SQL and NoSQL Database Tutorials:** For designing and integrating databases.
- **Stripe API Documentation:** For integrating payment gateway functionality.
- **Agile Methodology Resources:** For project management and sprint planning.

Conclusion –

This **Faculty Kit** serves as a comprehensive guide for faculty members involved in the development and deployment of the **Online Toll Payment System**. Through its structured approach, it ensures that all aspects of the project, from initial planning to final review, are thoroughly covered. The kit helps faculty understand the various technologies, systems, and modules used in the project, such as database creation, frontend development, and backend integration. It ensures that the system is secure, user-friendly, and scalable, providing a valuable learning experience for students working on this project.