ShuttleSync
Railway Management System

August 15, 2025

1 Project Name

ShuttleSync-Railway-App

2 Overview

ShuttleSync is a comprehensive railway management system built using the MERN (MongoDB, Express.js, React.js, Node.js) stack. This web application provides a complete solution for railway ticket booking with secure user authentication, train search functionality, payment processing, and real-time notifications.
The system is designed to streamline the train booking process for passengers while offering a modern, user-friendly interface. It handles the entire booking workflow from user registration to ticket confirmation, including secure payment processing and booking management.

3 Project Features

3.1 User Authentication

- **JWT Authentication**: Secure login and signup system using JSON Web Tokens
- User registration with form validation
- Secure password hashing and storage
- Protected routes and session management

3.2 Train Management

- **Search Trains**: Search functionality to find trains by origin, destination, and date
- Real-time seat availability checking
- Train schedule and fare information display
- Filter options for different train types and classes

3.3 Booking System

- **Book Train**: Complete booking process with seat selection
- Passenger information collection and validation
- Booking confirmation with unique transaction id
- Digital ticket generation

3.4 Payment Gateway

- Integrated payment processing system using SSL Commerz
- **Successful Payment**: Automatic booking confirmation
- **Payment Failure**: Error handling with retry options and booking cancellation
- Secure transaction processing and verification

3.5 Booking Management

- **Show Bookings**: Display user's booking history and current reservations
- Booking status tracking (confirmed, cancelled, pending)
- Option to cancel or modify bookings

3.6 Real-time Features

- **Real-time Notices**: Live updates and announcements
- Train delay notifications
- Platform and gate change alerts
- Emergency notices and important updates
- Instant booking confirmations

4 How to Run

4.1 Prerequisites

Before running the project, ensure you have the following installed:

- Node.js (version 14.x or higher)
- MongoDB (version 4.x or higher)
- npm or yarn package manager
- Git

4.2 Installation Steps

Step 1: Clone the Repository

```
git clone https://github.com/yourusername/shuttlesync.git
cd ShuttleSync-Railway-App
```

Step 2: Setup Backend

```
# Navigate to backend directory
cd backend

# Install dependencies
npm install

# Create environment file
cp .env.example .env
# Edit .env file with your configuration
```

Step 3: Setup Frontend

```
# Navigate to frontend directory
cd ../frontend

# Install dependencies
npm install
```

Step 4: Configure Environment Variables Create a `.env` file in the backend directory and frontend directory as well with the following:

```
# Database
MONGODB_URI=mongodb://localhost:27017/shuttlesync

# JWT Secret
JWT_SECRET=your-secret-key-here

# SSLCommerz Payment Gateway Configuration
SSLCOMMERZ_STORE_ID=your-store-id
SSLCOMMERZ_STORE_PASSWD=your-store-password


# Server Configuration
PORT=8080
NODE_ENV=development
FRONTEND_URL=http://localhost:3000



#frontend .env
REACT_APP_API_URL=http://localhost:8080
REACT_APP_FRONTEND_URL=http://localhost:3000
```

Step 5: Start the Application

```
# Start MongoDB service (if not running)
mongod

# Start backend server (in backend directory)
cd backend
node seedTrains.js
node index.js
# Backend will run on http://localhost:8080

# Start frontend (in new terminal, from frontend directory)
cd frontend
npm start
# Frontend will run on http://localhost:3000
```

4.3 Alternative: Run with Concurrently

If you have concurrently installed, you can run both frontend and backend together:

```
# From project root directory
npm install concurrently --save-dev
npm run dev
```

4.4 Access the Application

Once both servers are running:

- Frontend: http://localhost:3000
- Backend API: http://localhost:8080
- MongoDB: Default connection on port 27017

4.5 Default Test Credentials

If you have seed data, you can use these credentials to test the application:

```
Email: test@example.com
Password: password123
```

4.6 Troubleshooting

- Ensure MongoDB is running before starting the backend
- Check that all environment variables are properly configured
- Verify that ports 3000 and 8080 are not being used by other applications
- Run `npm install` if you encounter dependency issues