Development Environment Setup and Implementation Guide for Travel Agency Website

Overview

This document outlines the setup and implementation for a full-stack web development project designed for a **travel agency website**, where users can both **book travel** and **transport goods**. The technologies used in the project are:

- **Backend**: Node.js with Express.js
- **Frontend**: Next.js
- **Database**: PostgreSQL
- **Version Control**: GitHub
- **Deployment**: Render (for basic deployment)

1. Backend Setup: Node.js with Express.js

The backend of the travel agency website is built using **Node.js** and **Express.js** to handle the server-side logic, ensuring that the platform is scalable and efficient. This is particularly important since the site will have to handle bookings, payments, and logistics for goods transportation.

API Design for the Travel Agency

Design RESTful APIs to handle the following types of requests:

- **Travel bookings**: Manage routes, availability, seat reservations, and payment processing.
- **Goods transportation**: Allow customers to specify pickup/drop-off locations, size/weight of items, and choose transportation modes.
- **Customer management**: Create, update, and manage customer profiles, including travel preferences and booking history.

2. Frontend Setup: Next.js

The frontend of the travel agency website is built using **Next.js**, which is well-suited for building modern, high-performance web applications. It allows for server-side rendering, essential for fast loading of pages and improving the user experience, especially during peak usage times.

Key Features

- **Travel Booking Pages**: Display available routes, schedules, and prices with options to book.

- **Goods Transport Pages**: Allow users to input details like package weight, destination, and type of transport.
- **Customer Dashboard**: Users can log in to view their previous bookings, track goods, and update personal details.

3. Database Setup: PostgreSQL

For managing data, **PostgreSQL** is the ideal choice due to its SQL support, ability to handle complex queries, and ease of integration with Node.js. PostgreSQL will store data related to users, bookings, available travel routes, and goods transportation details.

Key Data Entities

- **Users**: Customer details, including personal information, preferences, and booking history.
- **Bookings**: Both travel and goods transport bookings, with details like destination, dates, payment information.
- **Travel tracking logs**: Enables users to view previous transportation information along with live tracking of active transportation activities.

4. Version Control: GitHub

GitHub will be used for version control, allowing multiple developers to collaborate efficiently and track changes to the codebase. It's particularly helpful for managing feature branches for travel bookings, goods transport features, and more

5. Deployment: Render

For deploying the travel agency website, **Render** is a great option for its simplicity and reliability. Both the backend and frontend will be deployed separately on Render.

Deployment Steps

- **Backend Deployment**: Deploy the Node.js server, ensuring that PostgreSQL database credentials and other environment variables are set up correctly in Render.
- **Frontend Deployment**: Deploy the Next.js application and connect it to the backend API.