**Payment System Documentation**

**Table of Contents**

1. Introduction
2. System Overview
3. Prerequisites
4. Payment Gateway Server
5. Installation and Setup
6. Database
7. MySQL Setup
8. Database Schema
9. Merchant Application Server
10. Installation and Setup
11. Merchant Server Registration
12. Payment UI Web
13. Installation and Setup
14. Conclusion

**Introduction**

This document provides step-by-step instructions for setting up and understanding the Payment System, which consists of three main components:

Payment Gateway Server,

Merchant Application Server,

and Payment UI Web.

**System Overview**

The Payment System is designed to facilitate payment processing between merchants and users. The system comprises the following components:

**Payment Gateway Server:** Manages payment processing logic and communication between the Merchant Application and Payment UI.

**Merchant Application Server:** Handles merchant-specific logic and interactions.

**Payment UI Web:** A React-based web application for users to make payments and check payment status.

**Prerequisites**

Before you begin, make sure you have the following tools and software installed:

1. Node.js
2. npm
3. MySQL

**Payment Gateway Server**

**Installation and Setup**

1. Clone the repository.

git clone <https://github.com/deepak-lamba/Payment_Project.git>

1. Navigate to the Payment Gateway Server directory:

cd Payment\_Project\Payment\_Gateway\_SpringBoot

1. Install dependencies:

npm install

1. Start the Payment Gateway Server:

cd Payment\_Project\Payment\_Gateway\_SpringBoot\target

run java -jar Payment\_Gateway\_SpringBoot-0.0.1-SNAPSHOT.jar

The server will be running on http://localhost:8080 by default.

**Database**

**MySQL Setup**

1. Ensure you have MySQL installed.
2. Create a new database for the payment system:

CREATE DATABASE payment\_gateway\_db;

1. Update the database configuration settings in the config.js files of Payment\_Gateway\_SpringBoot.

**Database Schema**

The Payment Gateway uses a MySQL database. The schema tables are defined in the SpringBoot Application will be generated automatically.

**Merchant Application Server**

**Installation and Setup**

1. Navigate to the Merchant Application Server directory:

cd Payment\_Project\Merchant\_Application

1. Install dependencies:

npm install

1. Start the Merchant Application Server:

cd Payment\_Project\Merchant\_Application\target

run java -jar Merchant\_Application-0.0.1-SNAPSHOT.jar

The server will be running on http://localhost:8081 by default.

**Merchant Server Registration**

To integrate your Merchant Application Server with the Payment Gateway:

1. Use the following endpoint to register your server:

GET <http://localhost:8080/api/registration/register-server>

1. Include the following Authorization header:

Authorization: hash@access@register

1. Upon successful registration, the Payment Gateway Server will respond with:

{

"merchantId": "your-merchant-id",

"serverKey": "your-server-key"

}

Store these values in your application.properties file.

**Payment UI Web**

**Installation and Setup**

1. Navigate to the Merchant Application Server directory:

cd Payment\_Project\web\_payment\_ui

1. Install dependencies:

npm install

1. Start the Payment UI Web

npm start

The web application will be accessible at http://localhost:3000 by default.

**Conclusion**

Follow these instructions to set up and run the Payment System successfully. If you encounter any issues, you can reach out to me.