**Procedure**

1. Set up the development environment: Before you start building the project, you need to have a development environment set up. This will involve installing the necessary software and tools on your system. Here are the steps:

* Install Node.js: Node.js is a JavaScript runtime that you'll need to run the project. You can download and install it from the official website: <https://nodejs.org/en/download/>
* Install Truffle: Truffle is a development framework for Ethereum. You can install it using the following command: npm install -g truffle
* Install Ganache: Ganache is a personal blockchain for Ethereum development. You can download and install it from the official website: <https://www.trufflesuite.com/ganache>

1. Create a new Truffle project: Once you have the development environment set up, you can create a new Truffle project using the following steps:

* Open your terminal and navigate to the directory where you want to create the project.
* Run the following command: truffle init

This will create a new Truffle project with a basic folder structure.

1. Write the smart contract: The next step is to write the smart contract that will power the payment system. Here are the steps:

* Navigate to the contracts directory in your Truffle project.
* Create a new file called PaymentSystem.sol.
* Write the smart contract code in this file. You can use Solidity, which is a programming language for Ethereum, to write the code.

1. Compile the smart contract: Once you've written the smart contract code, you need to compile it to generate the bytecode that will be deployed to the Ethereum network. Here are the steps:

* Run the following command in your terminal: truffle compile

This will compile the smart contract and generate the bytecode.

1. Configure the deployment: The next step is to configure the deployment of the smart contract to the Ethereum network. Here are the steps:

* Navigate to the migrations directory in your Truffle project.
* Create a new file called 2\_deploy\_contracts.js.
* Write the deployment code in this file. You'll need to specify the name of the smart contract and the bytecode that you generated in the previous step.

1. Deploy the smart contract: The final step is to deploy the smart contract to the Ethereum network. Here are the steps:

* Start Ganache and make sure it's running.
* Run the following command in your terminal: truffle migrate –reset

This will deploy the smart contract to the Ethereum network.

1. Build the frontend: Now that you have the smart contract deployed to the Ethereum network, you can build the frontend of the payment system. Here are the steps:

* Navigate to the client directory in your Truffle project.
* Create a new file called index.html.
* Write the HTML code for the frontend in this file.
* Create a new file called app.js.
* Write the JavaScript code for the frontend in this file. You'll need to use web3.js, which is a JavaScript library for interacting with Ethereum, to communicate with the smart contract.

1. Test the payment system: The final step is to test the payment system to make sure it's working as expected. Here are the steps:

* Start a local web server by running the following command in your terminal: python -m http.server
* Open your web browser and navigate to http://localhost:8000.
* Use the payment system to make a payment and verify that the transaction is processed correctly.