1. **Requirements**:
   - Node.js (v14.0 or higher)
   - Testnet cryptocurrency wallets (e.g., Bitcoin, Ethereum)
   - PayPal Developer account and sandbox credentials

2. **Set up the project folder and initialize it:**

apt install npm

mkdir crypto-exchange
cd crypto-exchange

npm init -y

3. Install required packages:

npm install express body-parser axios dotenv web3
```

4. Go to **Paypal Developer Dashboard to create a new sandbox account**. In the dashboard, create a new sandbox app and retrieve the client ID and secret for the app.

5. Choose a **cryptocurrency testnet wallet** (Bitcoin or Etherum), and obtain an API key from the testnet provider like Infura or **BlockCypher**

6. **Create a `.env` file to store your credentials:**

PAYPAL_CLIENT_ID=<your_paypal_client_id>
PAYPAL_SECRET=<your_paypal_secret>
```

7. **Create an `app.js` file and add the following code:**

const express = require("express");
const bodyParser = require("body-parser");
const axios = require("axios");
const dotenv = require("dotenv");

```
dotenv.config();

const app = express();
app.use(bodyParser.json());

// PayPal API URL and credentials
const PAYPAL_API_URL = "https://api-m.sandbox.paypal.com";
const PAYPAL_CLIENT_ID = process.env.PAYPAL_CLIENT_ID;
const PAYPAL_SECRET = process.env.PAYPAL_SECRET;

// Testnet cryptocurrency wallets
const wallets = {
  bitcoin: "your_testnet_bitcoin_wallet_address",
  ethereum: "your_testnet_ethereum_wallet_address",
};

// Helper function to get a PayPal access token
async function getPayPalAccessToken() {
  const response = await axios.post(
    `${PAYPAL_API_URL}/v1/oauth2/token`,
    "grant_type=client_credentials",
    {
      headers: {
        "Content-Type": "application/x-www-form-urlencoded",
        Authorization: `Basic ${Buffer.from(
          `${PAYPAL_CLIENT_ID}:${PAYPAL_SECRET}`
        ).toString("base64")}`,
      },
    }
  );

  return response.data.access_token;
}

// API endpoint for creating a PayPal order
app.post("/create-order", async (req, res) => {
  try {
```

```
    const { currency, amount } = req.body;
    const access_token = await getPayPalAccessToken();

    const response = await axios.post(
      `${PAYPAL_API_URL}/v2/checkout/orders`,
      {
        intent: "CAPTURE",
        purchase_units: [
          {
            amount: {
              currency_code: currency,
              value: amount,
            },
          },
        ],
      },
      {
        headers: {
          "Content-Type": "application/json",
          Authorization: `Bearer ${access_token}`,
        },
      }
    );

    res.json(response.data);
  } catch (error) {
    console.error(error);
    res.status(500).json({ error: "Failed to create order" });
  }
});

// API endpoint for capturing a PayPal order
app.post("/capture-order", async (req, res) => {
  try {
    const { orderID } = req.body;
    const access_token = await getPayPalAccessToken();

    const response = await axios.post(
```

```
      `${PAYPAL_API_URL}/v2/checkout/orders/${orderID}/capture`,
      {},
      {
        headers: {
          "Content-Type": "application/json",
          Authorization: `Bearer ${access_token}`,
        },
      }
    );

    // Simulate sending cryptocurrency to the user's wallet
    console.log(
      `Sent ${response.data.purchase_units[0].amount.value}
${response.data.purchase_units[0].amount.currency_code} to wallet:
${wallets.bitcoin}`
    );

    res.json(response.data);
  } catch (error) {
    console.error(error);
    res.status(500).json({ error: "Failed to capture order" });
  }
});

// Start the server
const port = process.env.PORT || 3000;
app.listen(port, () => {
  console.log(`Server running on port ${port}`);
});
```

8) **Test your exchange system**

Start your server by running node server.js

Now you have a simple cryptocurrency exchange system with e-payment through PayPal using testnet and sandbox accounts. You can expand this

example by adding more functionality, such as handling different cryptocurrencies, checking wallet balances, and integrating with real cryptocurrency APIs.