// Import required libraries

const express = require('express');

const bodyParser = require('body-parser');

const crypto = require('crypto');

const { ethers } = require('ethers');

// Initialize Express app

const app = express();

app.use(bodyParser.json());

// In-memory storage for users (for demonstration purposes)

let users = {};

// Function to generate a new wallet

function createWallet() {

const wallet = ethers.Wallet.createRandom();

return {

address: wallet.address,

privateKey: wallet.privateKey,

};

}

// Endpoint to create a new user and wallet

app.post('/create-wallet', (req, res) => {

const username = req.body.username;

if (!username || users[username]) {

return res.status(400).json({ error: 'Invalid username or already exists' });

}

const wallet = createWallet();

users[username] = {

address: wallet.address,

privateKey: wallet.privateKey,

};

res.json({

message: 'Wallet created successfully',

address: wallet.address,

});

});

// Endpoint to get wallet balance (mock implementation)

app.get('/balance/:username', (req, res) => {

const username = req.params.username;

if (!users[username]) {

return res.status(404).json({ error: 'User not found' });

}

// Mock balance (in a real application, you would check the actual balance)

const balance = Math.floor(Math.random() \* 100); // Random balance for demonstration

res.json({

address: users[username].address,

balance: `${balance} ETH`,

});

});

// Endpoint to send funds (mock implementation)

app.post('/send', (req, res) => {

const { fromUsername, toAddress, amount } = req.body;

if (!users[fromUsername]) {

return res.status(404).json({ error: 'Sender not found' });

}

// In a real application, you would perform the transaction here

res.json({

message: `Successfully sent $

{amount} ETH from ${users[fromUsername].address} to ${toAddress}`,

});

});

// Start the server

const PORT = process.env.PORT || 3000;

app.listen(PORT, () => {

console.log(`Server is running on http://localhost:${PORT}`);

});