**Setup instructions for both backend and frontend**

**1. Backend Setup**

Install Node.js and npm:

Make sure you have Node.js and npm installed. You can download and install them from Node.js official website.

**1.1 Create a New Project Directory:**

mkdir fintech-platform

cd fintech-platform

**1.2 Initialize Node.js Project:**

npm init -y

**1.3 Install Required Packages:**

npm install express axios jsonwebtoken body-parser

**1.4 Create the Backend Files:**

// server.js

const express = require('express');

const axios = require('axios');

const jwt = require('jsonwebtoken');

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

const app = express();

const port = 3000;

app.use(bodyParser.json());

const SECRET\_KEY = 'your\_secret\_key'; // Replace with a strong secret key

// Dummy in-memory storage for demonstration

const users = [];

let transactions = [];

// Helper functions

const authenticateToken = (req, res, next) => {

const authHeader = req.headers['authorization'];

const token = authHeader && authHeader.split(' ')[1];

if (token == null) return res.sendStatus(401);

jwt.verify(token, SECRET\_KEY, (err, user) => {

if (err) return res.sendStatus(403);

req.user = user;

next();

});

};

// Routes

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

const { username, password } = req.body;

if (users.find(user => user.username === username)) {

return res.status(400).send('User already exists');

}

users.push({ username, password, balance: 0 });

res.status(201).send('User registered');

});

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

const { username, password } = req.body;

const user = users.find(user => user.username === username && user.password === password);

if (!user) return res.status(400).send('Invalid credentials');

const token = jwt.sign({ username: user.username }, SECRET\_KEY);

res.json({ token });

});

app.get('/user', authenticateToken, (req, res) => {

const user = users.find(user => user.username === req.user.username);

if (!user) return res.sendStatus(404);

res.json({ username: user.username, balance: user.balance });

});

app.post('/transaction/deposit', authenticateToken, (req, res) => {

const { amount } = req.body;

const user = users.find(user => user.username === req.user.username);

if (!user) return res.sendStatus(404);

user.balance += amount;

transactions.push({ user\_id: user.username, type: 'deposit', amount, created\_at: new Date() });

res.sendStatus(200);

});

app.post('/transaction/withdraw', authenticateToken, (req, res) => {

const { amount } = req.body;

const user = users.find(user => user.username === req.user.username);

if (!user) return res.sendStatus(404);

if (user.balance < amount) return res.status(400).send('Insufficient funds');

user.balance -= amount;

transactions.push({ user\_id: user.username, type: 'withdrawal', amount, created\_at: new Date() });

res.sendStatus(200);

});

app.get('/transactions', authenticateToken, (req, res) => {

const userTransactions = transactions.filter(t => t.user\_id === req.user.username);

res.json(userTransactions);

});

app.listen(port, () => {

console.log(`Server running at http://localhost:${port}`);

});

Create a file named server.js:

**Run the Backend Server:**

**2. Frontend Setup**

**2.1 Create a New Project Directory for the Frontend:**

sh

Copy code

mkdir fintech-frontend

cd fintech-frontend

Create the Frontend Files:

Create a file named index.html:

html

Copy code

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Fintech Platform</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<div id="app">

<h1>Fintech Platform</h1>

<div id="login-form">

<h2>Login</h2>

<input type="text" id="username" placeholder="Username">

<input type="password" id="password" placeholder="Password">

<button onclick="login()">Login</button>

</div>

<div id="register-form">

<h2>Register</h2>

<input type="text" id="reg-username" placeholder="Username">

<input type="password" id="reg-password" placeholder="Password">

<button onclick="register()">Register</button>

</div>

<div id="user-details" style="display: none;">

<h2>User Details</h2>

<p id="user-info"></p>

<h3>Transactions</h3>

<div id="transactions"></div>

<h3>Perform Transaction</h3>

<input type="number" id="amount" placeholder="Amount">

<button onclick="deposit()">Deposit</button>

<button onclick="withdraw()">Withdraw</button>

</div>

</div>

<script src="app.js"></script>

</body>

</html>

Create a file named styles.css:

css

Copy code

body {

font-family: Arial, sans-serif;

padding: 20px;

}

#app {

max-width: 600px;

margin: 0 auto;

}

h1, h2, h3 {

text-align: center;

}

input {

display: block;

width: 100%;

margin-bottom: 10px;

padding: 8px;

box-sizing: border-box;

}

button {

display: block;

width: 100%;

padding: 10px;

background-color: #007BFF;

color: white;

border: none;

cursor: pointer;

}

button:hover {

background-color: #0056b3;

}

Create a file named app.js:

js

Copy code

let token = '';

function register() {

const username = document.getElementById('reg-username').value;

const password = document.getElementById('reg-password').value;

fetch('http://localhost:3000/register', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify({ username, password })

})

.then(response => response.text())

.then(data => alert(data));

}

function login() {

const username = document.getElementById('username').value;

const password = document.getElementById('password').value;

fetch('http://localhost:3000/login', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify({ username, password })

})

.then(response => response.json())

.then(data => {

token = data.token;

getUserDetails();

});

}

function getUserDetails() {

fetch('http://localhost:3000/user', {

headers: {

'Authorization': `Bearer ${token}`

}

})

.then(response => response.json())

.then(user => {

document.getElementById('user-info').innerText = `Username: ${user.username}, Balance: ${user.balance}`;

document.getElementById('login-form').style.display = 'none';

document.getElementById('register-form').style.display = 'none';

document.getElementById('user-details').style.display = 'block';

getTransactions();

});

}

function getTransactions() {

fetch('http://localhost:3000/transactions', {

headers: {

'Authorization': `Bearer ${token}`

}

})

.then(response => response.json())

.then(transactions => {

const transactionList = document.getElementById('transactions');

transactionList.innerHTML = '';

transactions.forEach(transaction => {

const transactionItem = document.createElement('div');

transactionItem.innerText = `Type: ${transaction.type}, Amount: ${transaction.amount}, Date: ${new Date(transaction.created\_at).toLocaleString()}`;

transactionList.appendChild(transactionItem);

});

});

}

function deposit() {

const amount = document.getElementById('amount').value;

fetch('http://localhost:3000/transaction/deposit', {

method: 'POST',

headers: {

'Content-Type': 'application/json',

'Authorization': `Bearer ${token}`