

Blockchain Lab

Assignment-7

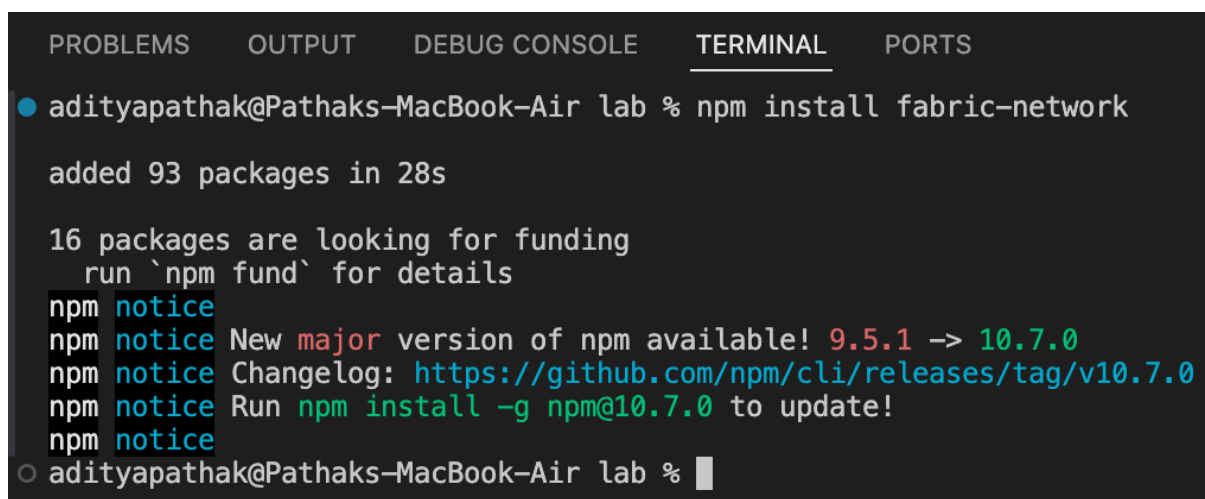
Name: Aditya Pathak

Subgroup: 3NC2

Roll Number: 102115044

The Hyperledger Fabric Software Development Kit (SDK) enables developers to build a wide range of applications for enterprise blockchain solutions.

Here's a simplified example of an identity management application using the Hyperledger Fabric SDK in Node.js. This example demonstrates how to enroll users, register new users, and perform identity-related operations on a Hyperledger Fabric network.



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
● adityapathak@Pathaks-MacBook-Air lab % npm install fabric-network

added 93 packages in 28s

16 packages are looking for funding
  run `npm fund` for details
npm notice
npm notice New major version of npm available! 9.5.1 -> 10.7.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.7.0
npm notice Run npm install -g npm@10.7.0 to update!
npm notice
○ adityapathak@Pathaks-MacBook-Air lab %
```

Installed Hyperledger Fabric Software Development Kit

Using VS Code Terminal

Code

JS hyperledgerFabric_task7.js × {} package-lock.json

JS hyperledgerFabric_task7.js > ...

```
1  const { Wallets, Gateway } = require('fabric-network');
2  const FabricCAServices = require('fabric-ca-client');
3
4  async function main() {
5      const gateway = new Gateway();
6
7      try {
8          const wallet = await Wallets.newFileSystemWallet('./wallet');
9
10         const caURL = 'http://localhost:7054';
11         const ca = new FabricCAServices(caURL);
12
13         const adminIdentity = await wallet.get('admin');
14         if (!adminIdentity) {
15             console.log('An identity for the admin user does not exist in the wallet');
16             return;
17         }
18
19         const gatewayOptions = {
20             identity: 'admin',
21             wallet: wallet,
22             discovery: { enabled: true, asLocalhost: true }
23         };
24
25         await gateway.connect('mychannel', gatewayOptions);
26
27         const network = await gateway.getNetwork('mychannel');
28
29         const contract = network.getContract('basic');
30
31         // enrolling a user
32         const enrollment = await ca.enroll({ enrollmentID: 'user1', enrollmentSecret: 'user1pw' });
33         const userIdentity = {
34             credentials: {
35                 certificate: enrollment.certificate,
36                 privateKey: enrollment.key.toBytes(),
37             },
38             mspId: 'Org1MSP',
39             type: 'X.509',
40         };
41
42         await wallet.put('user1', userIdentity);
43
44         console.log('Successfully enrolled user1 and imported it into the wallet');
```

```
45
46     // registering a new user
47     const registerRequest = {
48         enrollmentID: 'user2',
49         affiliation: 'org1.department1',
50         role: 'client',
51     };
52
53     const secret = await ca.register(registerRequest, adminIdentity);
54     console.log(`Successfully registered user2 with enrollment secret: ${secret}`);
55
56     } catch (error) {
57         console.error(`Error: ${error}`);
58         process.exit(1);
59     } finally {
60         gateway.disconnect();
61     }
62 }
63
64 main();
```