Kerala Blockchain Academy (KBA)

An initiative of the Government of Kerala under the Kerala University of Digital Sciences, Innovation and Technology (formerly IIITM-K)

Online Internship Project Report

on

Blockchain based National Scholarship Portal

Submitted in partial fulfillment of the requirements for Blockchain Internship Program offered by Kerala Blockchain Academy during the Academic Year 2024-25

by

Asmi Rajbhar

Atharva Hande

Lifna Challissery Samu

Project Mentor
Lekshmi M. B (Research Scientist, KBA)
Vishva Prasad (Research and Development Engineer @KBA)

Kerala Blockchain Academy
(AY 2024-25)

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea / data / fact / source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Asmi Rajbhar
Atharva Hande
Lifna Challissery Samu

Date: **2nd June 2025**

INDEX

Chapter 1: Brief overview of the project

The project undertaken during the internship was titled **"Intelligent Scholarship Disbursement Module for National Scholarship Portal (NSP)."** This project was conceptualized as part of the **Digital India Initiative** by the Government of India, aimed at ensuring a **quick and efficient process for the disbursement of government scholarships** to eligible students. The existing **National Scholarship Portal (NSP)** provides a platform where students can apply for government scholarships. However, to apply for **private scholarships**, students need to approach multiple organizations separately. This dilemma often leads to confusion, delays, and administrative burdens for the students. To address this issue, the objective of this project was to design an **Intelligent Scholarship Disbursement Module** that integrates both **Government and Private Scholarship schemes** into a single, unified platform. The module automates the entire scholarship process using **Aadhaar details provided during registration**. It streamlines processes such as **document verification, income verification via PAN**, and **fund disbursement**, ensuring **centralized monitoring and quick processing**.

Figure 1 illustrates the overall architecture of the proposed system. The main stakeholders are the NSP, applicants, EIs, SOs (both private and public bodies), the ITD, the UIDAI, and the banks involved in fund disbursement. Internally, the Hyperledger fabric network maintains two channels: one for communication between the EIs and another for the SOs. Each of these organizations maintains a ledger and a copy of the chaincode. The membership service provider (MSP) and certificate authority (CA) are key components in the Hyperledger fabric network. The MSP manages the identities of all participating organizations, like private SO and EIs, ensuring that only verified members can access the network and perform actions. The CA issues digital certificates to these organizations, which act as their identity proof within the blockchain network. Thus, the MSP and CA create a safe and reliable environment within the Hyperledger fabric network in the NSP to efficiently manage scholarships, ensuring that only authorized institutions and providers can participate in the process.

When an applicant submits a scholarship application through a decentralized application (DApp), the request is sent over the internet to the Hyperledger fabric network established by the NSP. The request first reaches special computers called endorser peers (EPs), which check the request against a smart contract (a set of rules) to ensure its validity. If everything is in order, these peers sign off on the transaction. The signed transaction is then sent to an orderer peer (O), which organizes these transactions into a block and sends it to all relevant computers in the network, called committer peers (CPs). These peers further validate the block and permanently

record the transactions on the distributed ledger. An anchor peer (AP) helps synchronize this information across different organizations in the network.

After the scholarship deadline, the NSP initiates a smart contract to verify the credentials of the candidates through their respective EIs. These institutions authorize the candidates, and the NSP generates a verified list of eligible candidates based on the criteria defined by the SO. The SO then shortlists the candidates and initiates a smart contract to trigger the banks to disburse the funds to the selected candidates. Once the funds are transferred, the applicant and other involved entities, such as the NSP, EIs, and SOs, are notified of the transaction's status. The DApp displays the updated status to the applicant, and all related data is securely stored in a decentralized storage system, ensuring it is safe and accessible.

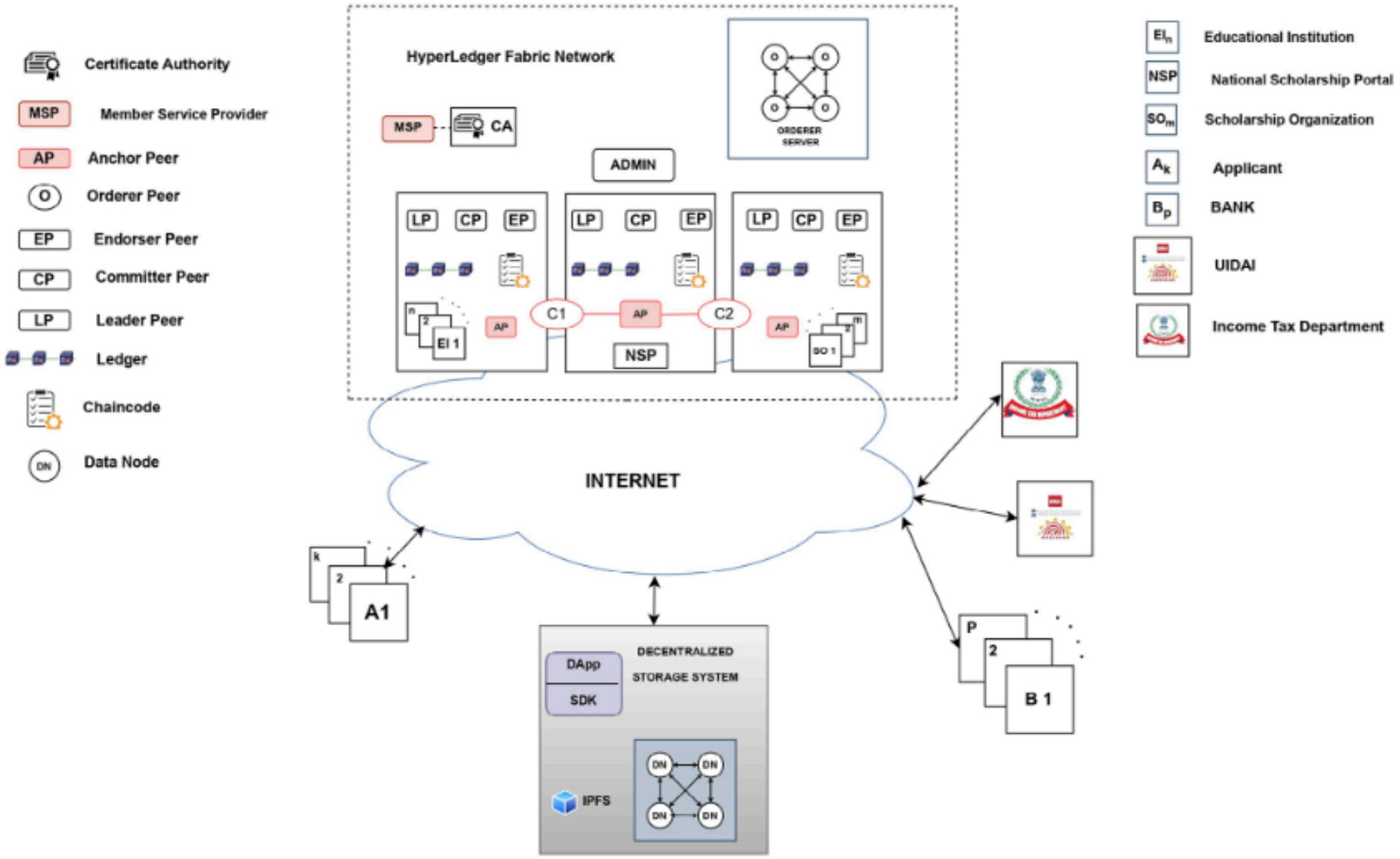

Figure - 1 : Proposed system architecture of the blockchain-based National Scholarship Portal.

Chapter 2: Detailed description of the tasks undertaken

During the internship, the main task was to transition the Intelligent Scholarship Disbursement Module for NSP project from a static, development-only network to a production-ready Hyperledger Fabric network. Initially, a **frontend application** was prepared and connected to a static test network, which could not be used for production. To create a **production-grade blockchain network**, **Fabric samples** provided by Hyperledger Fabric were used as the foundation. We focused on modifying and configuring critical files within the Fabric samples to adapt them to the project's needs. These included:

- **compose-ca.yaml** to define and configure **certificate authorities (CAs)** necessary for managing identities of participants in the network.
- **compose-test-net.yaml** to set up and configure the **network topology**, ensuring that it correctly reflected the organizations and peers in the project.
- **registerEnroll.sh** to manage the **registration and enrollment** processes for user identities across the three key organizations in the system: **NSPOrg** (National Scholarship Portal Organization), **EIOrg** (Educational Institution Organization), and **SchOrg** (Scholarship Organization).
- **configtx.yaml** to set the **channel configurations** and **orderer system parameters**, ensuring that the network could support the required transaction processing and governance rules.

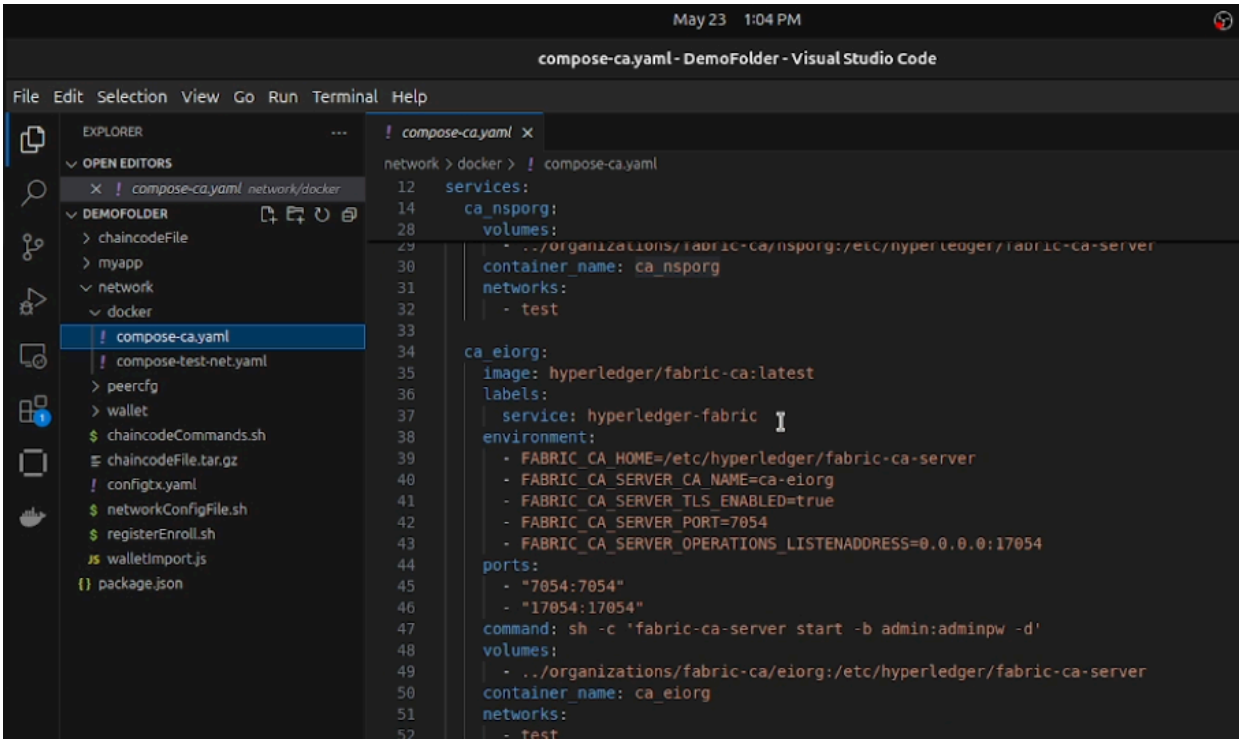

Figure - 2 : Screenshot depicting the editing on file - **compose-ca.yaml**

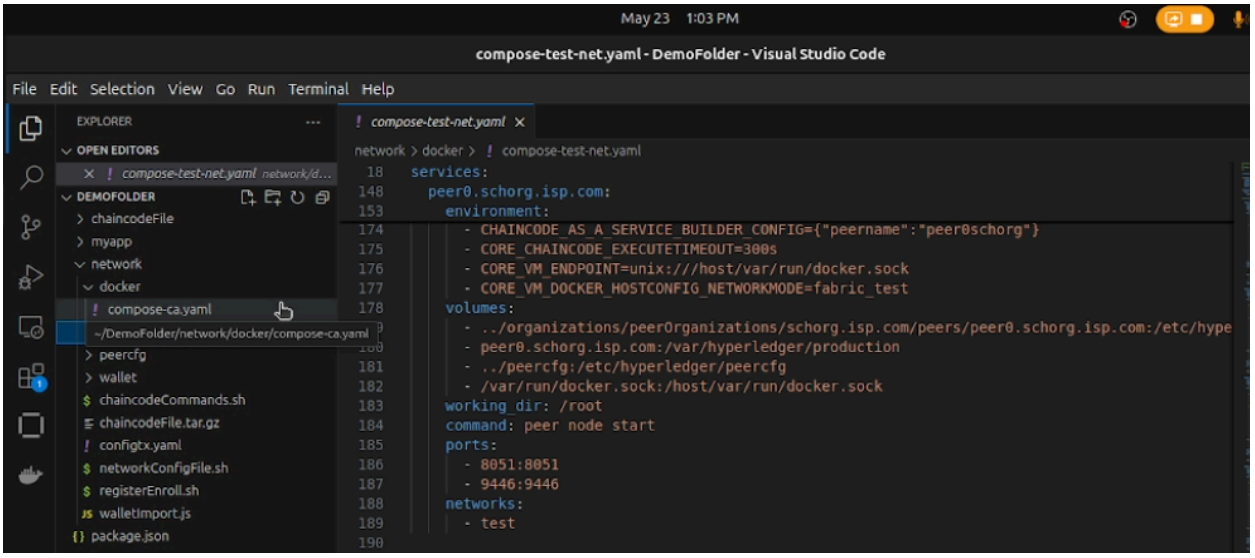

Figure - 3 : Screenshot depicting the editing on file - **compose-test-net.yaml**

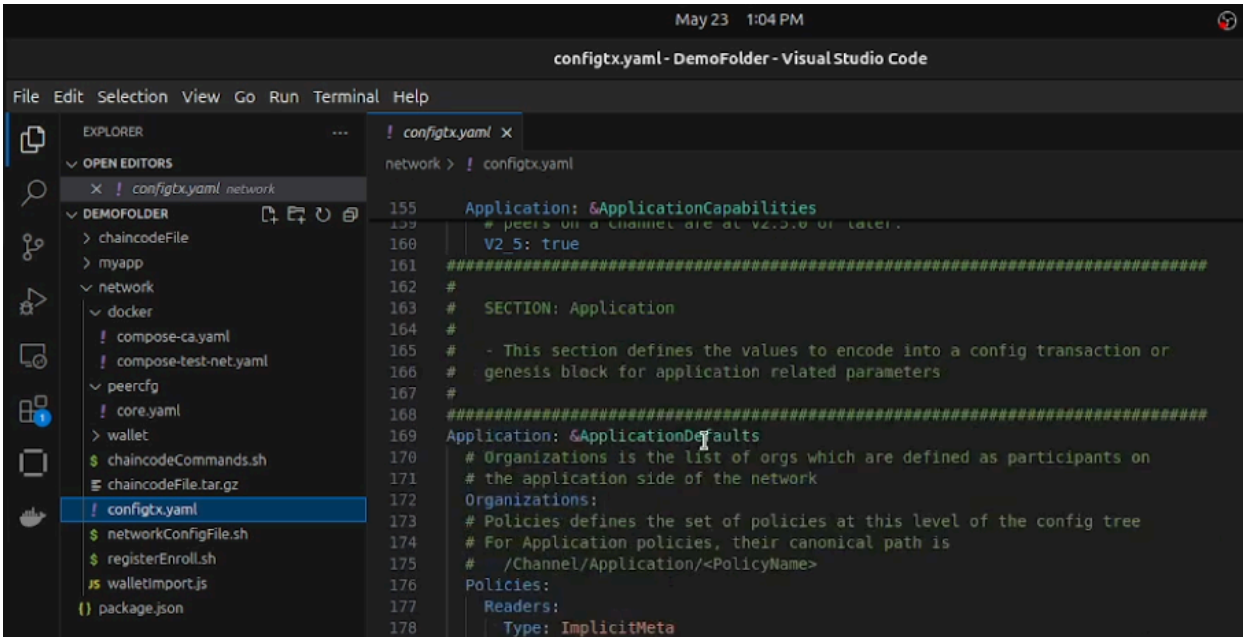

Figure - 4 : Screenshot depicting the editing on file - **configtx.yaml**

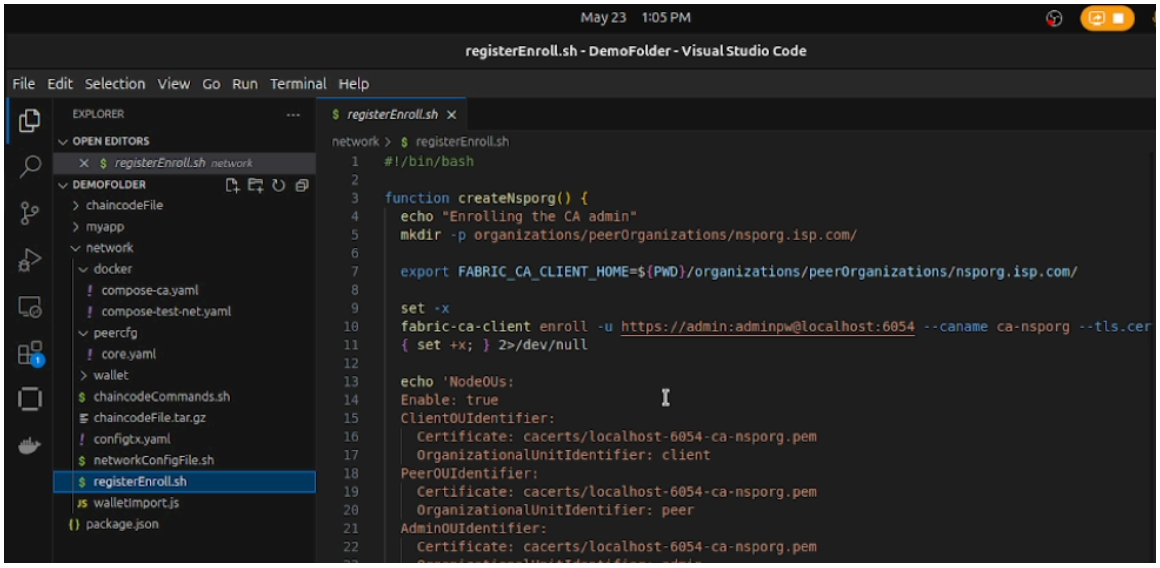

Figure - 5 : Screenshot depicting the editing on file - **registerEnroll.sh**

After these changes, a custom script called **networkConfigFile.sh** was created to streamline and automate the setup and configuration of the network based on the modified files. This script ensured that the network components, including the **peers, orderers, and CAs**, were deployed correctly. Once the network was operational, the focus shifted to deploying the **chaincode (smart contract)**, which defines the logic for the scholarship disbursement process. A second script, **chaincodeCommands.sh**, was created to automate the commands required to **install, approve, and commit the chaincode** to the production network. The result was a **fully functional, production-ready Hyperledger Fabric network** with a deployed chaincode, ready to support the **Intelligent Scholarship Disbursement Module** project.

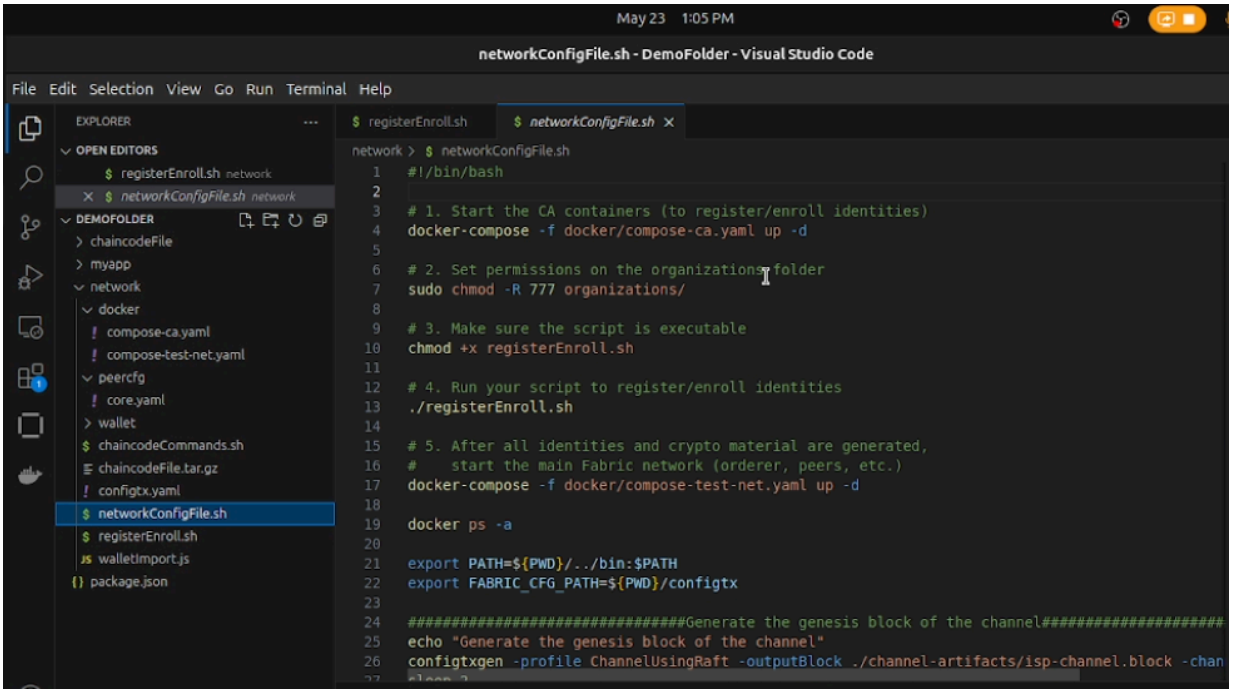

Figure - 6 : Screenshot depicting the editing on file - **networkConfigFile.sh**

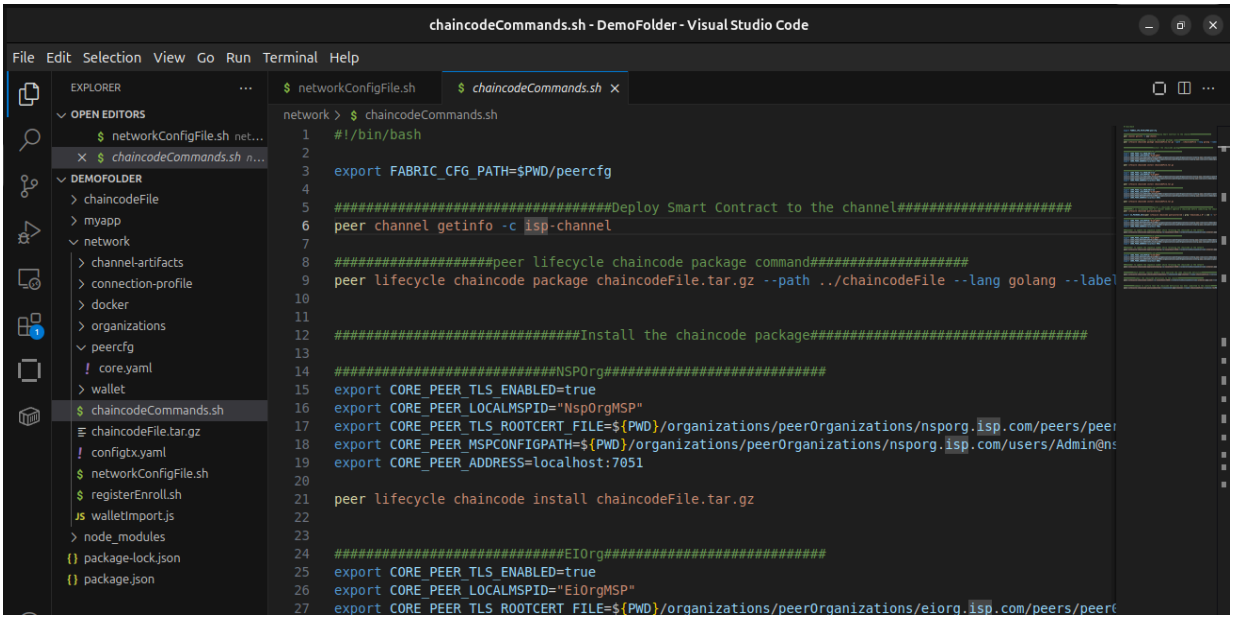

Figure - 7 : Screenshot depicting the editing on file - **chaincodeCommands.sh**

Chapter 3: Technologies/tools used

The experimental environment was configured on an Intel Core i3-2350 M CPU @ 2.30 GHz × 4 with 8GB of main memory, running on Ubuntu 20.04 LTS. Docker v28.1.0 was utilized for containerization, alongside Docker Compose v1.29.2 to manage multi-container applications. Node.js v22.12.0 and npm v10.9.0 were used for package management. The smart contracts were written in Go v1.20, and Hyperledger Fabric v2.5.4[1] components were deployed as the blockchain framework, facilitating the execution of the experiment. The entire documentation of the setup along with smart contracts code and related documents are available at the Github repository[2] for the project. The following technologies were used for the implementation of the Internship Project.

- **Hyperledger Fabric**: for creating and managing the blockchain network.
- **Docker**: for running network components such as peers, orderers, and certificate authorities in isolated containers.
- **Hyperledger Fabric Samples**: provided sample files and templates that were customized for the project
- **Shell Scripts (Bash)**: for automating the network configuration (**networkConfigFile.sh**) and chaincode deployment (**chaincodeCommands.sh**).
- **YAML Configuration Files**: such as **docker-ca.yaml**, **docker-test-net.yaml**, and **configtx.yaml** for defining the structure and behavior of the network.
- **Node.js / React.js**: used for building the frontend application (based on what you mentioned in earlier discussions).

Chapter 4: Key learnings and outcomes

The following are the outcomes of the Internship Project :

- **Learned how to configure and customize a Hyperledger Fabric network** for production use by modifying core files like **docker-ca.yaml** and **configtx.yaml**.
- **Understood the structure of a Fabric network** and the roles of different organizations (**NSPOrg, EIOrg, SchOrg**) in the scholarship system.
- **Gained hands-on experience in creating and executing shell scripts** to automate network setup and chaincode deployment.
- **Learned to integrate a production-ready blockchain network** with an existing frontend application for real-world use.
- **Developed problem-solving skills** in configuring and troubleshooting network components and chaincode deployment.
- Successfully **deployed a production-grade network** and **integrated chaincode logic** into the Intelligent Scholarship Disbursement Module project.

Chapter 5: Screenshots or links to the deployed code repository

The script file mentioned in chapter 2 were executed successfully and the screenshots are as follows:

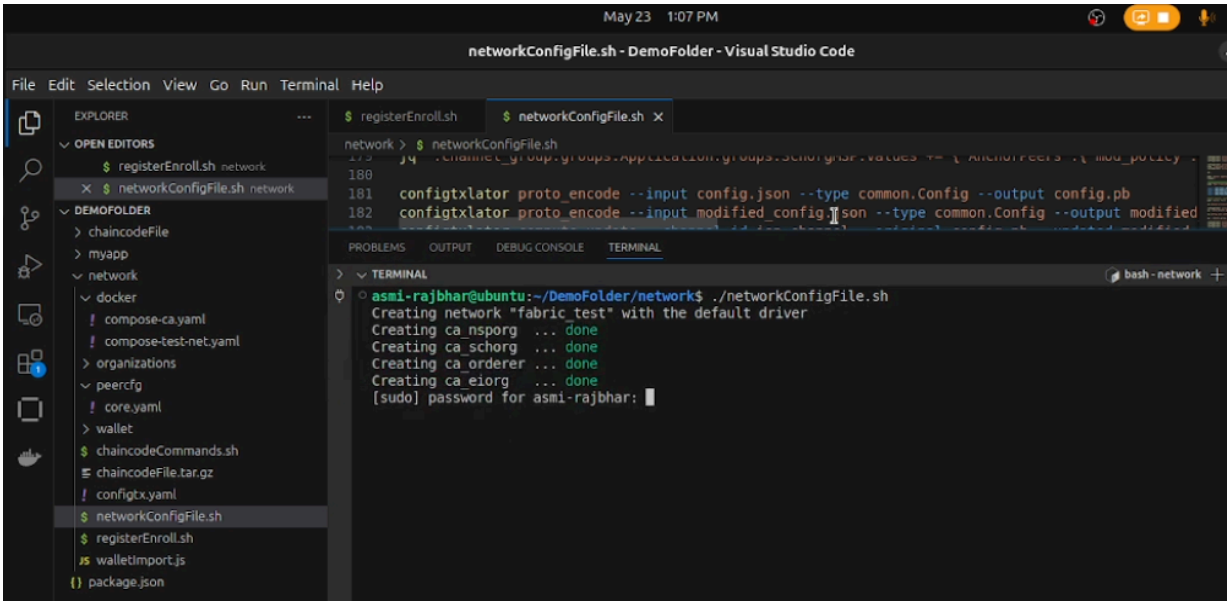

Figure - 8(a) : Depicts the execution of **networkConfigFile.sh** file creating the peer network

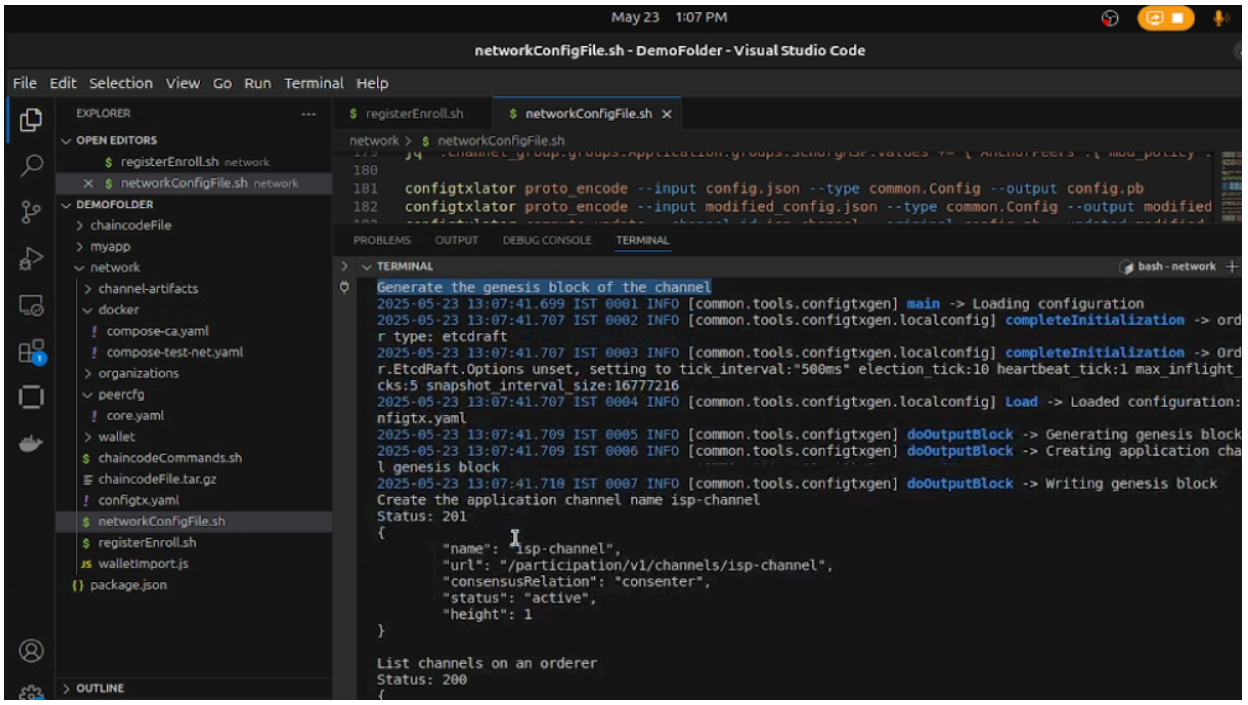

Figure - 8(b) : Depicts the execution of **networkConfigFile.sh** file where the Genesis block is generated for the channel

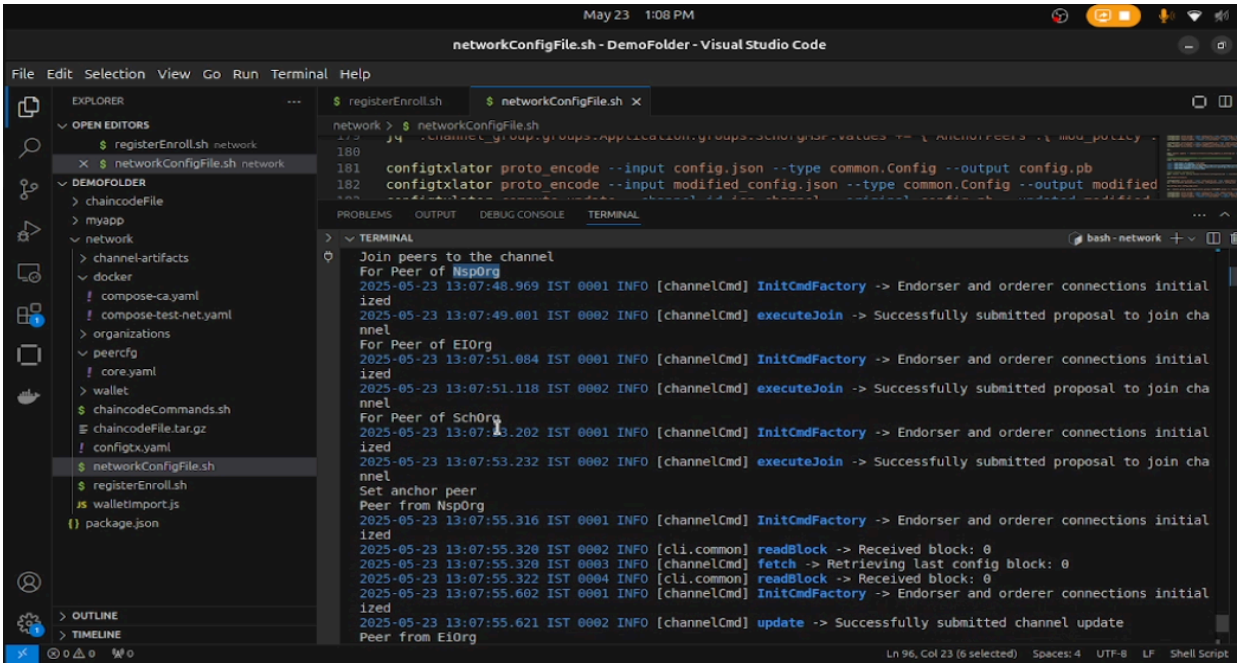

Figure - 8(c) : Depicts the execution of **networkConfigFile.sh** file where the peers are added to the channel

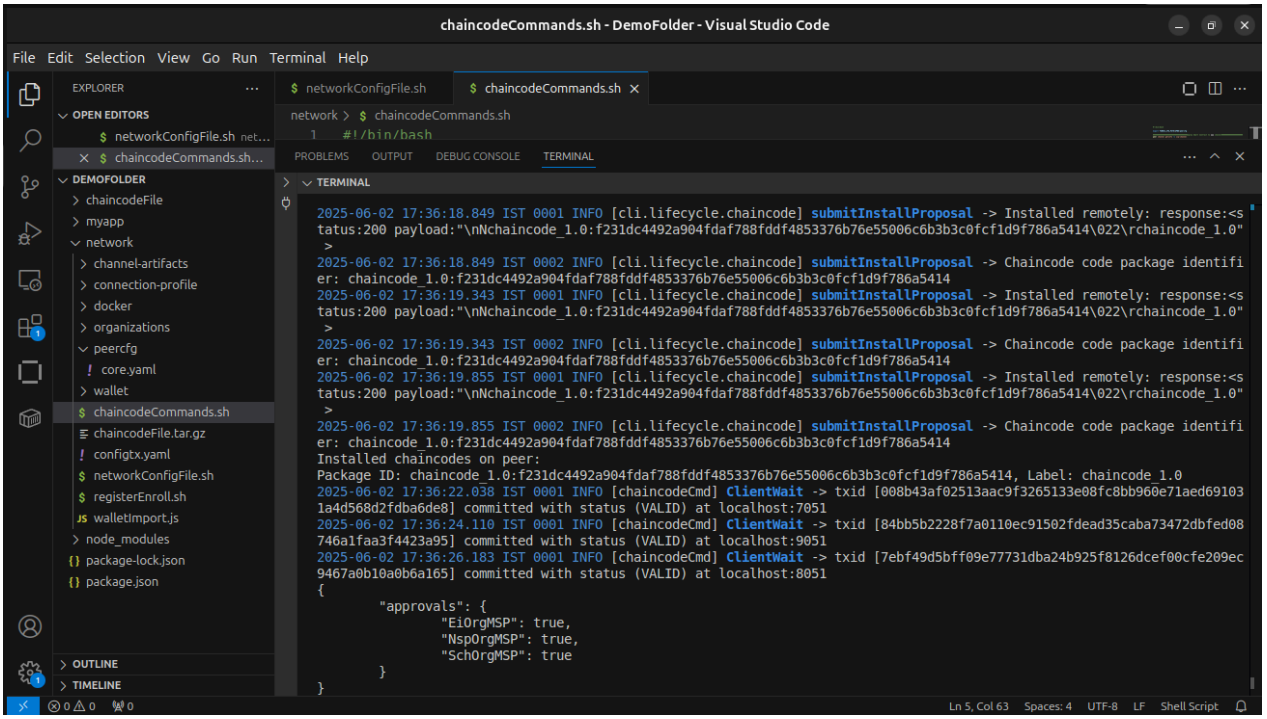

Figure - 9(a) : Executing the **chaincodeCommands.sh** file for installing the chaincode on to the peers remotely.

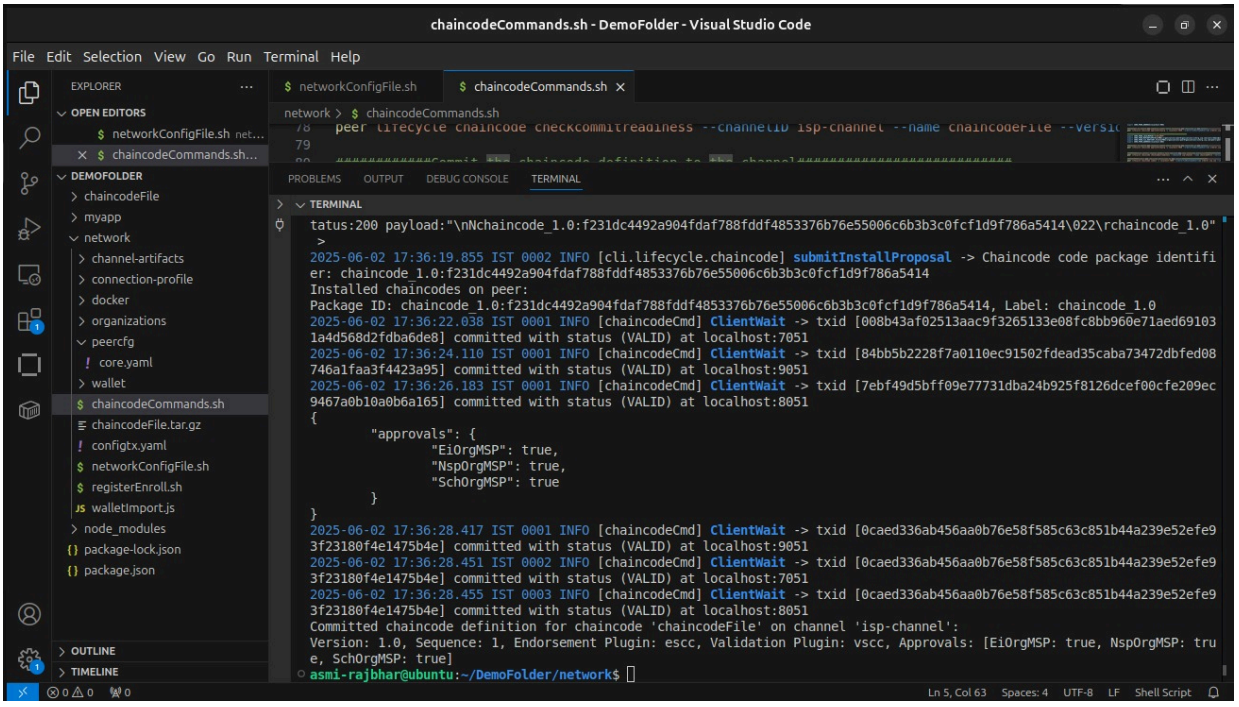

Figure - 9(b) : Executing the **chaincodeCommands.sh** file where the chaincode is approved by the peers remotely.

Figure - 10 : Depicts the invocation of chaincode for making an EIOrg entry into the ledger of the network.

Code Repository : Blockchain based National Scholarship Portal—Github Repository. https://github.com/LifnaJos/Blockchain-based-National-Scholarship-Portal.

References

1. Hyperledger fabric documentation. https://hyperledger-fabric.readthedocs.io/en/release-2.2/getting_started.html

2. Hyperledger GitLab from KBA https://gitlab.com/CHF_KBA/kba_chf_vb14