



|| ಪ್ರದ್ಯುಮಿ ಪರಮಾ ಗತಿಃ ||

THE NATIONAL COLLEGE
Autonomous
Jayanagar, Bangalore-560070

PROJECT REPORT
ON
HYPERLEDGER FABRIC BLOCKCHAIN BASED
CBFTIG APPLICATION

BY

Sai Pavan A S

20NJB440

Under the guidance of

Prof. VARADARAJ

CBFTIG project report submitted in partial fulfilment of the requirements
of

VI Semester BCA, THE NATIONAL COLLEGE JAYANAGAR

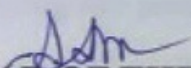


|| ಶ್ರದ್ಧಾಹಿ ಪರಮಾ ಗತಿಃ ||

THE NATIONAL COLLEGE
Autonomous
Jayanagar, Bangalore-560070

CERTIFICATE

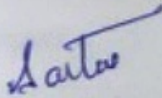
This is to certify the project report titled "CBFTIG Application" is a work done by **Sai Pavan A S** of THE NATIONAL COLLEGE, Jayanagar, Bengaluru, in partial fulfilment of the requirements of VI Semester BCA during the year 2022-2023.


HEAD OF THE DEPARTMENT

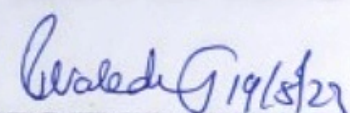
Head. Dept. of Comp. Science
The National Degree College
(Autonomous)
Jayanagar, Bangalore - 560 070

Examiners:

1.



2.


PROJECT GUIDE

Examination Centre

**The National College,
Jayanagar.**

Date of Examination: 22/8/23

ACKNOWLEDGEMENT

CBFTIG Application is the project of many hands from the team. Our tribute for the successful completion of the project goes to all those who helped through their constant guidance and encouragement. The satisfaction that accompanies the success would be incomplete without thanking the person who made it.

I am thankful to our beloved Principal **Dr.B SURESHA**, who encourages us to come with new and innovative ideas and for providing the environment with all facilities for completing the project.

I am also grateful to our Head of the Department **Prof. ASHA TS** Department of computer science for her valuable guidance and constant support during our project development.

I am also grateful to our project guide **Prof. VARADARAJ**, lecturer Department of computer science for his valuable guidance and constant support during our project development.

A special thanks to **MUTHURAM GOVINDARASU**, CEO and Founder of Indigeneous Tech Private Limited, Bangalore-32 with 10 years of experience in Blockchain for his valuable guidance and technical support for our project.

I extend our thanks to all our teaching staffs of the department of computer science. Finally, we thank one and all who helped us directly and indirectly for the completion of our project.

Table of Contents

| | |
|---|----|
| I) Project Goal (Problem Statement) | 7 |
| II) Solution Proposed | 7 |
| III) Input Data and Verification | 10 |
| IV) Project/ Solution (CBFTW) Design | 12 |
| V) Tools/ Technologies Used | 17 |
| VI) Project Team Members | 17 |
| VII) Referenced Documents: | 18 |
| VIII) Project / Solution Setup | 18 |
| 1) AWS free-tier Account and EC2 Instance creation and connection | 18 |
| a) Create an AWS free-tier account and further create an EC2 instance with Ubuntu Operating System AMI | 18 |
| b) Connect to the AWS EC2 Instance through "PuTTY" Software and open the Terminal. Change over to "root" directory with Admin permissions | 18 |
| 2) Download the required pre-requisite tools and Software into the EC2 Instance, transfer the "CBFTIG" Project files from Windows System to EC2 Instance using "FileZilla" and setup the project in the EC2 Instance..... | 18 |
| a) Download and install all the pre-required tools and software for the Hyperledger Fabric based Blockchain set-up in the created EC2 Instance | 19 |
| b) In the EC2 instance, clone the "fabric-samples" directory, download and install the Hyperledger Based Blockchain Binaries Version 1.4.6 and download the Hyperledger based Docker images from Docker hub. | 19 |
| c) Transfer these program and files from Windows System to the created AWS EC2 Instance using the "FileZilla" software | 19 |
| d) Go to "CBFTIG-network" directory and create Cryptographic based certificates and update the "docker-compose.yaml" file with the relevant CA certificate and save the file | 19 |
| e) Create a directory "channel-artifacts" and create genesis.block and channel.tx files under that directory and verify. Convert the "start.sh" file into an executable file | 19 |
| f) Go to "CBFTIG" directory and convert the "startCBFTIG.sh" and "teardownCBFTIG.sh" files into executable ones | 19 |
| g) Execute the command "npm install" and verify the creation of "node_modules" directory | 19 |
| IX) Project / Solution Execution | 20 |
| 1) In the "CBFTW" directory, execute the command "./startCBFTIG.sh" command and verify that the Blockchain Network is up and deployed | 20 |
| 2) Verify that all the required docker containers are up and running | 21 |
| 3) Enroll the "Admin" into Blockchain Network executing the "enrollAdmin.js" file | 22 |
| 4) Register "User1" into the Blockchain Network executing the "registerUser.js" file | 22 |

CBFTW APPLICATION

5) Query and list all the records (Bank , Customer and Forex records) that have got instantiated into the Blockchain by the Smart contract "banks.go" using "queryAll_Data.js" file 23

6) Query the specified Bank record and verify 23

7) Query the specified Customer record and verify 24

8) Query the specified Forex record and verify 24

9) Add a new Bank record into the Blockchain then query the created record and verify it 24

10) Add a new Customer data into the Blockchain and then query that record and verify it 25

11) Add a new Forex rate data into the Blockchain and then query that record and verify it 26

2) Pay the Customer US_John_Doe_123 with 1000 EURO units from Customer EU_Marcos_999 and verify 26

3) Query and Verify the Customers (EU_Marcos_999 and US_John_Doe_123) data and note down their new account balances and verify 27

4) Query the current reserves of Japanese_Bank and UK_Bank in the Blockchain.... 28

X) Setup and loading of "CBFTIG Frontend"..... 29

1) Open the "/CBFTW-front/src/App.js" file and update the Public IP address of the CBFTW Project EC2 Instance and save the file 29

2) From the "CBFTW" directory, execute the command "node CBFTW-backend.js" and verify the display of "Listening on port 4001" 32

3) Create a duplicate session of the EC2 Instance, navigate to "CBFTW-front" directory in that instance, execute the command "npm run start" and verify the successful starting of the React App Development Server 32

4) Load the "CBFTW-frontend" onto the Chrome browser by using the url: "http://Public IP of the Instance:3000/" and verify the successful display of the CBFTW Application frontend 35

XI) Interaction with the CBFTIG Blockchain Network using CBFTW Frontend..... 36

1) Click on "QUERY ALL" option and then "SEARCH ALL" button. Verify the display of all the available data under "Info" window 36

2) Query an existing Bank and verify: Click on QUERY, enter "US_Bank" then click on "SEARCH" button..... 37

3) Query an existing Customer and verify the data: Click on QUERY, enter "US_Alice_456" then click on "SEARCH" button 38

4) Query an existing Forex Pair and verify the value: Click on QUERY, enter "USD:JPY" then click on "SEARCH" button 39

5) Create a new Customer: Click on the CREATE CUSTOMER option, enter the new Customer details and then click on "CREATE CUSTOMER" button..... 40

6) Query the created Customer and verify: Click on the QUERY option, enter the created Customer details and then click on "SEARCH" button 41

7) Create a new Bank : Click on the CREATE BANK option, enter the new Bank details and then click on "CREATE BANK" button 41

8) Query the created Bank and verify: Click on the QUERY option, enter the created Bank details and then click on "SEARCH" button 42

9) Create a new Forex: Click on the CREATE FOREX option, enter the new Forex details and then click on "CREATE FOREX" button..... 43

10) Query the created Forex: Click on the QUERY option, enter the created Query details and then click on "SEARCH" button..... 44

11) Query and verify the Customers (IND_Ramu_111 and EU_Marcos_999) data and note down their current account balances (for verifying the successful transfer of fund between them)..... 44

12) Query and verify the Banks (IND_Bank and EU_BankRamu_111 and EU_Bank) data and note down their current reserve amounts (for verifying the successful change in their reserves)..... 45

13) Pay 1,000 units in Indian currency (INR) from the IND_Bank Customer "IND_Ramu_111" to the EU_Bank Customer "EU_Marcos_999" and verify 46

14) Query and Verify the Customers (IND_Ramu_111 and EU_Marcos_999) data and note down their current account balances (for verifying the successful changes in the accounts' balances) 47

15) Query and verify the Banks (IND_Bank and EU_Bank) data and note down their current reserve amounts (for verifying the successful change in the Banks' reserves)... 48

XII) Closing of "CBFTIG" Project 49

XIII) Project Summary 52

I) Project Goal (Problem Statement)

To design, develop and verify the Hyperledger Fabric based Blockchain application namely "CBFTW Application" having the following features:

- 1) Query from the Banks Network's data that is available as part of the Blockchain (Reading bulk data from the Blockchain)
- 2) Query any one of the Banks Network's data that is available as part of the Blockchain (Reading individual record or data from the Blockchain)
- 3) Add a new Bank, Customer, Forex Value data and verify that these values get added successfully into the Blockchain (Writing new records into the blockchain and its verification)
- 4) Pay from any one of the Banks' Customer Account to any other Bank's Customer Account across the available Banks in the network effectively using the applicable Forex rates without any Third-party involvement (Cross Border Funds Transfer across the World (CBFTW)). Further verifying the successful transfer of the fund across the Customer accounts.

Note: These verifications need to be done both in Command Line Interface (CLI) mode and also using browser based front-end mode.

II) Solution Proposed

- 1) We will design, develop, implement and verify the Hyperledger Fabric based Blockchain network namely "CBFTW Application" in an Ubuntu O/S based EC2 Instance in AWS.
- 2) We will write the required Blockchain configuration and other files so as to bring-up the Blockchain network
- 3) We will write a Golang program namely "banks.go" in which we will implement the main logic of the Solution.
- 4) In the banks.go program, we will design the below given Data structures: