



|| ಶ್ರದ್ಧಾಢು ಪರಮಾ ಗತಿಃ ||

**THE NATIONAL COLLEGE**  
**Autonomous**  
**Jayanagar, Bangalore-560070**

**PROJECT REPORT**  
**ON**  
**HYPERLEDGER FABRIC BLOCKCHAIN BASED**  
**FABCAR APPLICATION**

**BY**

**Yadunandana D Kulkarni**

**19NCJB433**

**Under the guidance of**

**Prof. VARADARAJ R**

**FabCar project report submitted in partial fulfilment of the requirements**  
**of**  
**VI Semester BCA, THE NATIONAL COLLEGE JAYANAGAR**



|| ಶ್ರದ್ಧಾಹು ಪರಮಾ ಗತಿ: ||

**THE NATIONAL COLLEGE**  
**Autonomous**  
**Jayanagar, Bangalore-560070**

**CERTIFICATE**

This is to certify the project report titled "FabCar Application" is a work done by **Yadunandana D Kulkarni(19NCJB433)** of **THE NATIONAL COLLEGE, Jayanagar, Bengaluru**, in partial fulfilment of the requirements of VI Semester BCA during the year 2021-2022.

**HEAD OF THE DEPARTMENT**

**PROJECT GUIDE**

Examiners:

Dept. Of Comp. Science

**VALUED**

Examiner

(1)

(2)

2

Authorised Signatory

Examination Centre

The National College,  
Jayanagar.

Date of Examination:

## ABSTRACT

FabCar is a database of car records stored in the ledger of a Fabric network. We can consider this as a traditional database storing the data: it is like a table, indexed with a Car Identifier (CarID), and the information of Maker, Model, Colour and Owner is recorded for this car.

The data are stored in the world state database inside the ledger. Interaction with the data is through chaincode. FabCar comes with a Chaincode, containing the functions which can interact with the stored data in the ledger. They are for database (ledger) initiation, query and update. The world state is queried or updated only through these chaincode functions, and any update is logged in the blockchain inside the ledger as tamper-resistant record.

## **ACKNOWLEDGEMENT**

FabCar 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.

We are thankful to our beloved Principal Dr.KAMALA Y.C, who encourages us to come with new and innovative ideas and for providing the environment with all facilities for completing the project.

We are also grateful to our Head of the Department Prof. SHALINI.C Department of computer science for her valuable guidance and constant support during our project development.

We are also grateful to our project guide Prof. VARADARAJ.R, 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.

We 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.

## INDEX

	Content	Page No.
I	<b>Project Goal (Problem Statement)</b>	1
II	<b>Solution Proposed</b>	2
III	<b>Input Data and Verification</b>	4
IV	<b>Project/ Solution (FabCar) Design</b>	5
V	<b>Tools/ Technologies Used</b>	7
VI	<b>Referenced Documents</b>	8
VII	<b>Project / Solution Setup</b>	8
	<p><b>1) Account and EC2 Instance creation and connection</b></p> <p>a) Create an AWS free account and further create an EC2 instance with Ubuntu Operating System <span style="float: right;">Error!</span> Bookmark not defined.</p> <p>b) Connect to the AWS EC2 Instance through “Putty” Software and open the Terminal. Change over to “root” directory with Admin permissions</p>	8
	<p><b>2) Download the required pre-requisite tools and Software into the EC2 Instance, transfer the “FabCar” Project files from Windows System to EC2 Instance using “FileZilla” and setup the project in the EC2 Instance</b></p> <p>a) Download and install all the pre-required tools and software for the Hyperledger Fabric based Blockchain set-up in the created EC2 Instance</p> <p>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</p>	9

	<p>c) Transfer these program and files from Windows System to the created AWS EC2 Instance using the "FileZilla" software</p> <p>d) Go to "fabcar-network" directory and create Cryptographic based certificates and update the "docker-compose.yaml" file with the relevant CA certificate and save the file</p> <p>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</p> <p>f) Go to "fabcar" directory and convert the "startFabCar.sh" and "teardownFabCar.sh" files into executable ones</p> <p>g) Execute the command "npm install" and verify the creation of "node_modules" directory</p>	
VIII	<b>Project / Solution Execution</b>	10
	<p>1) In the "fabcar" directory, execute the command <code>./StartFabCar.sh</code> command and verify that the Blockchain Network is up and deployed</p> <p>2) Execute the command <code>node enrollAdmin.js</code> and verify that the "Admin" is enrolled successfully</p> <p>3) Execute the command <code>node registerUser.js</code> and verify that the "user1" is registered and enrolled successfully</p> <p>4) Execute the command <code>node query-All-Cars.js</code> and verify that the all the car data stored inside the blockchain are read and displayed successfully</p> <p>5) Execute the command <code>node query-CAR4.js</code> and verify that the CAR4 details are read from the blockchain and displayed successfully</p> <p>6) Execute the command <code>node invoke-CAR10.js</code> and check that the CAR10 has got stored into the blockchain successfully</p> <p>7) Execute the command <code>node query-CAR10.js</code> and check that the CAR10 data is read from the blockchain and displayed successfully</p> <p>8) Execute the command <code>node invoke-CAR10-Owner-change.js</code> and check that the new owner of CAR10 has got stored into the blockchain successfully</p> <p>9) Execute the command <code>node query-CAR10.js</code> and check that the CAR10 data is read from the blockchain and displayed successfully with new car owner</p>	<p>10</p> <p>13</p> <p>14</p> <p>14</p> <p>15</p> <p>15</p> <p>16</p> <p>16</p> <p>16</p>
IX	<b>Setup and loading of "Fabcar Frontend"</b>	17

	<p>1) Open the “fabcar-front/src/App.js” file and update the Public IP address of the fabcar EC2 Instance and save the file</p> <p>2) From the “fabcar” directory, execute the command “node fabcar-backend.js” and verify the display of “Listening on port 4001”</p> <p>3) Create the duplicate session of the EC2 Instance, navigate to “fabcar-front” directory in that instance, execute the command “npm run start” and verify the successful starting of the React App Development Server</p> <p>4) Load the “fabcar-frontend” onto the Chrome browser by using the url: “http://Public IP of the Instance:3000/” and verify the successful display of the fabcar frontend</p>	17
		21
		21
		23
X	<b>Interaction with the Fabcar Blockchain Network using Fabcar Frontend from Windows System Browser</b>	24
	1) Click on “QUERY ALL” option and then “SEARCH ALL” button	24
	2) Click on QUERY, enter CAR4 and then click on “SEARCH” button	25
	3) Click on CREATE option, enter the new Car details and then click on “CREATE” button	26
	4) Click on QUERY option, enter CAR11 and click on “SEARCH” button	27
	5) Click on TRANSFER option, enter CAR11 for CAR ID and Prakash for New Owner and then click on “TRANSFER” button	28
	6) Click on QUERY option, enter CAR11 and click on “SEARCH” button	29
XI	<b>Closing of “FabCar” Project</b>	30
XII	<b>Project Summary</b>	34
XIII	<b>Bibliography</b>	34