A

Major Project

On

DEFENCE'S DATA SECURITY

(Submitted in partial fulfilment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

Yashwanth sagar Koluguri
(18E41A05E8)

Under the Guidance of

Mr. P. SRINIVASA RAO

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SREE DATTHA INSTITUTE OF ENGINEERING AND SCIENCE

(Accredited by NAAC, Affiliated to JNTUH, Approved by AICTE, New Delhi)

Recognized Under Section 2(f) of the UGCAct.1956,

Sheriguda (V), Ibrahimpatnam (M), Ranga Reddy – 501510.

2018-22

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**DEFENCE'S DATA SECURITY**" is being submitted by **YASHWANTH SAGAR KOLUGURI(18E41A05E8)** in partial fulfilment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by him/her under our guidance and supervision during the year 2020-21.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Dr. P. Srinivasa Rao
Internal Guide

Dr. Md Sameeruddin Khan
Principal

Dr. M. Varaprasad Rao
HoD

External Examiner

Submitted for viva voice Examination held on _____

ACKNOWLEDGEMENT

Apart from our efforts, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard for my guide **Mr. P. Srinivasa Rao,** Assistant Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We are also thankful to **Dr. M. Varaprasad Rao,** Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. Md. Sameeruddin Khan,** Principal for being cooperative throughout this project. We would like to express our sincere gratitude to **Dr. GNV Vibhav Reddy**, Vice-Chairman and **Sri. G. Panduranga Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout this project.

The guidance and support were received from all the members of **Sree Dattha Institute of Engineering and Science** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

ABSTRACT

This project is titled " **DEFENCE'S DATA SECURITY** ".Blockchain is set to radically change our way of life in the coming decades. "one of the most important technologies since the advent of the Internet." Based on a peer-to-peer (P2P) topology, blockchain is a distributed ledger technology that allows data to be stored globally on thousands of servers – while letting anyone on the network see everyone else's entries in near-realtime . In other words, blockchain can be described as a global online database, that anyone, anywhere in the world, with an internet connection, can use. As a consequence, a blockchain doesn't belong to anyone, and it stores information permanently across a network of personal computers. Consequently, it can be seen as a revolutionary technology, thanks to its decentralised nature and its ability to distribute information among its participants, in total transparency evenly.

LIST OF FIGURES

LIST OF SCREENSHOTS

TABLE OF CONTENTS

1. INTRODUCTION

1. INTRODUCTION

1.1 PROJECT SCOPE

This project is titled as “Defence’s Data Security”. This software provides facility to upload the soldiers data and highly confidential data of missiles and attacks.This project uses ethereum smart contract for interacting with blockchain and storing data and keeping it secure.

1.2 PROJECT PURPOSE

Using the “untamperable” feature of blockchain technology, we can provide a solution to the problem of “hard to maintain evidence” in sensitive data management in military inspection and supervision, human resources and medical and healthcare.“Truthful record” of all information can be realized through “whole-network witness”, thus avoiding document counterfeits, file missing and information tampering.

1.3 PROJECT FEATURES

The main features of this project

a) Stores soldiers data and missile information

b) For storing data it requires like (aadhar number , batch no , age , identification no, included in the mission)

c) Admin verification

2. SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

SYSTEM ANALYSIS

System Analysis is the important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, “what must be done to solve the problem?” The system is viewed as a whole and the inputs to the system are identified. Once the analysis is completed the analyst has a firm understanding of what is to be done.

2.1 PROBLEM DEFINITION

As we have seen in daily life that many citizens data or information getting leaked by cyber threats , In the same way many hackers who are working for their countries they are trying to stole the defence information of other countries , if that happens the country will get into a risk because their military , airspace and navy information with them.

2.2 EXISTING SYSTEM

Currently, it is the Blockchain in Defense and Security. Blockchain Technology in Defense, Blockchain for Military Defense, Blockchain for Aerospace and Defense, etc are widely examined and explored today for several reasons. It is seen that the military and defense sector are into implying the Blockchain Technology for maximizing their capacity in their different field of action.

Blockchain military use cases are many like the Military Drone Technology, Blockchain Battleships, Decentralized Weapon Control Systems, etc. In Military Drone Technology, the Artificial Intelligence and Blockchain Technology are combined together with drone technology, opening immense possibilities of data collection.

2.2.1 LIMITATIONS OF EXISTING SYSTEM

- More classification.
- Time consuming.
- Needs manual calculations.

To avoid all these limitations and make the working more accurately the system needs to be implemented efficiently.

2.3 PROPOSED SYSTEM

The aim of proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of the existing system. The system provides higher accuracy and reduces the classification work. The existing system has several disadvantages and many more difficulties to work well. The proposed system that introduces blockchain technology in defence like storing information of a solider in one separate block which is not accessible for everyone , only higher officials can have access to it with admin login , in the same way secret operations missiles and attacks information should be stored in blocks which cannot be edited or stolen

2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has got following features

- Ensure data accuracy's.
- Minimum time needed for the various processing.
- Greater efficiency.
- Better service.
- User friendliness and interactive.
- Minimum time required.

2.4 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. Three key considerations involved in the feasibility analysis are

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

2.4.1 ECONOMIC FEASIBILITY

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development.

2.4.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.4.3 BEHAVIORAL FEASIBILITY

This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS

Hardware interfaces specifies the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

- Processor : Pentium IV 2.4GHz.
- Hard disk : 40GB and Above.
- RAM : 2GB and Above.
- Monitor : 15 VGA colour.

2.5.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements,

- Operating system : Windows 10
- Languages : Solidity, HTML, CSS, javascript
- IDE : visual studio code

3. ARCHITECTURE

3. ARCHITECTURE

3.1 PROJECT ARCITECTURE

This project architecture shows the procedure followed for providing security for Defence's data using Blockchain technology, starting from input to final prediction.

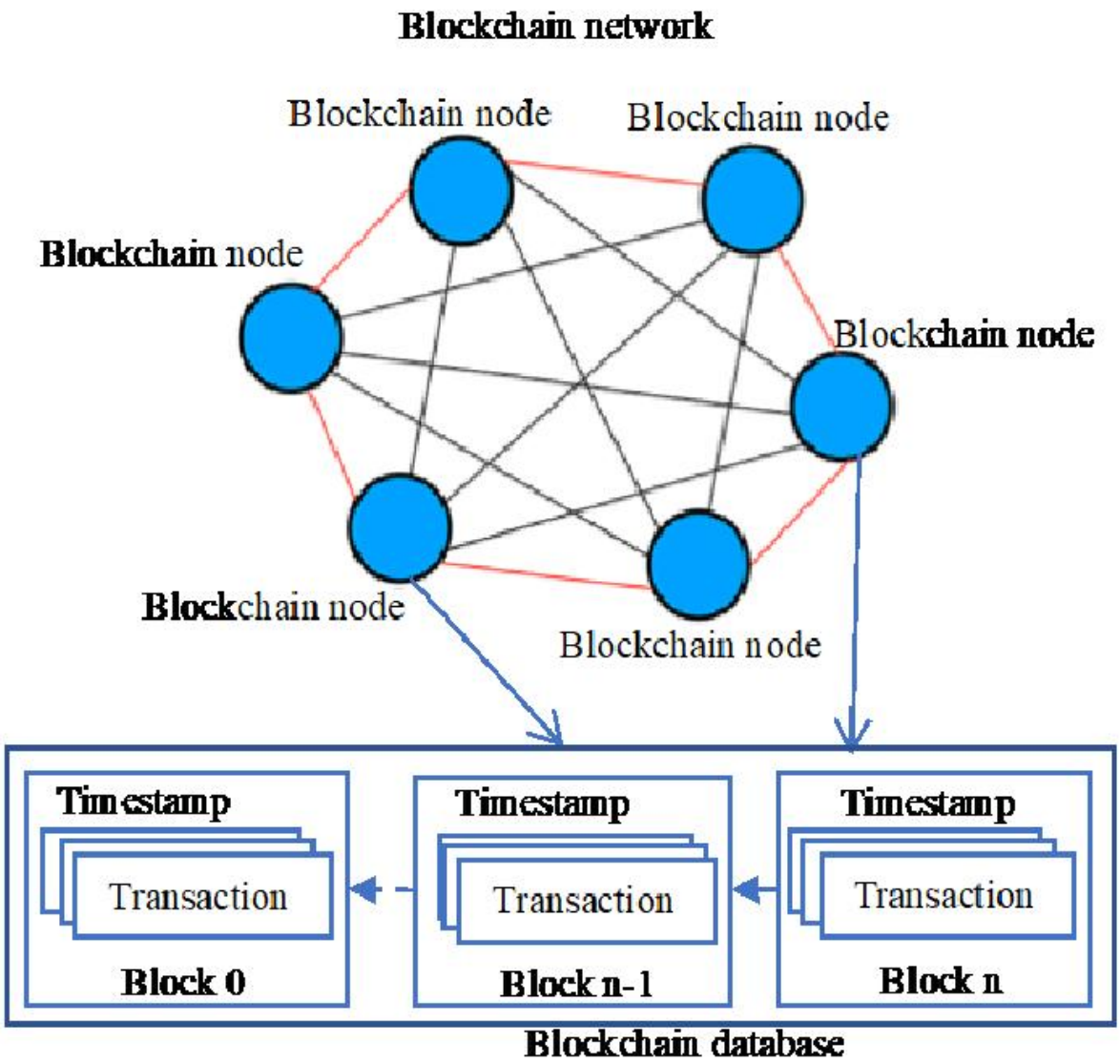


Figure 3.1: Project Architecture of Defence Data security

3.2 USE CASE DIAGRAM

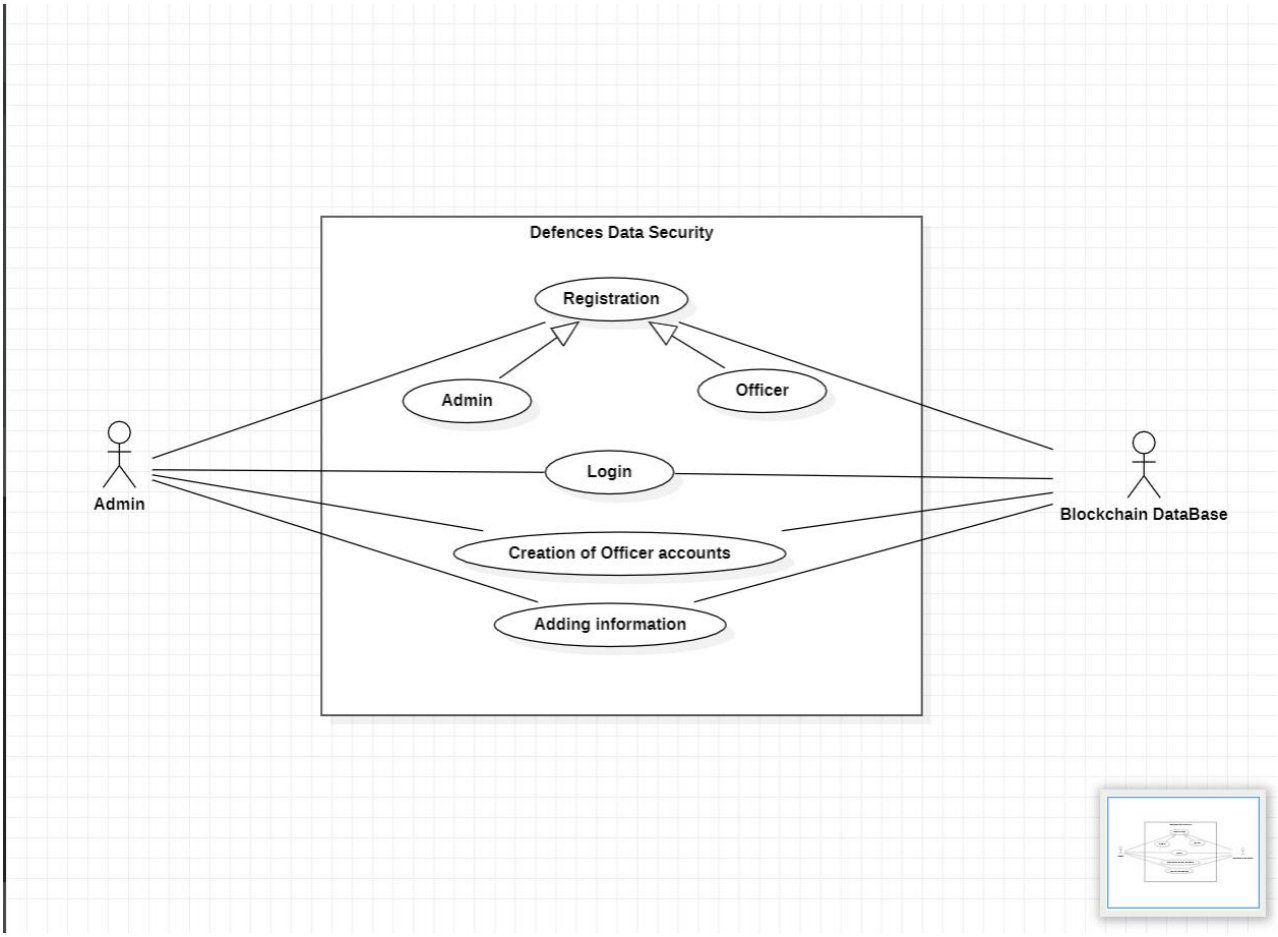


Figure 3.2: Use Case Diagram for user for Image Classifier to Identify Dog Breeds

3.3 CLASS DIAGRAM

Class Diagram is a collection of classes and objects.

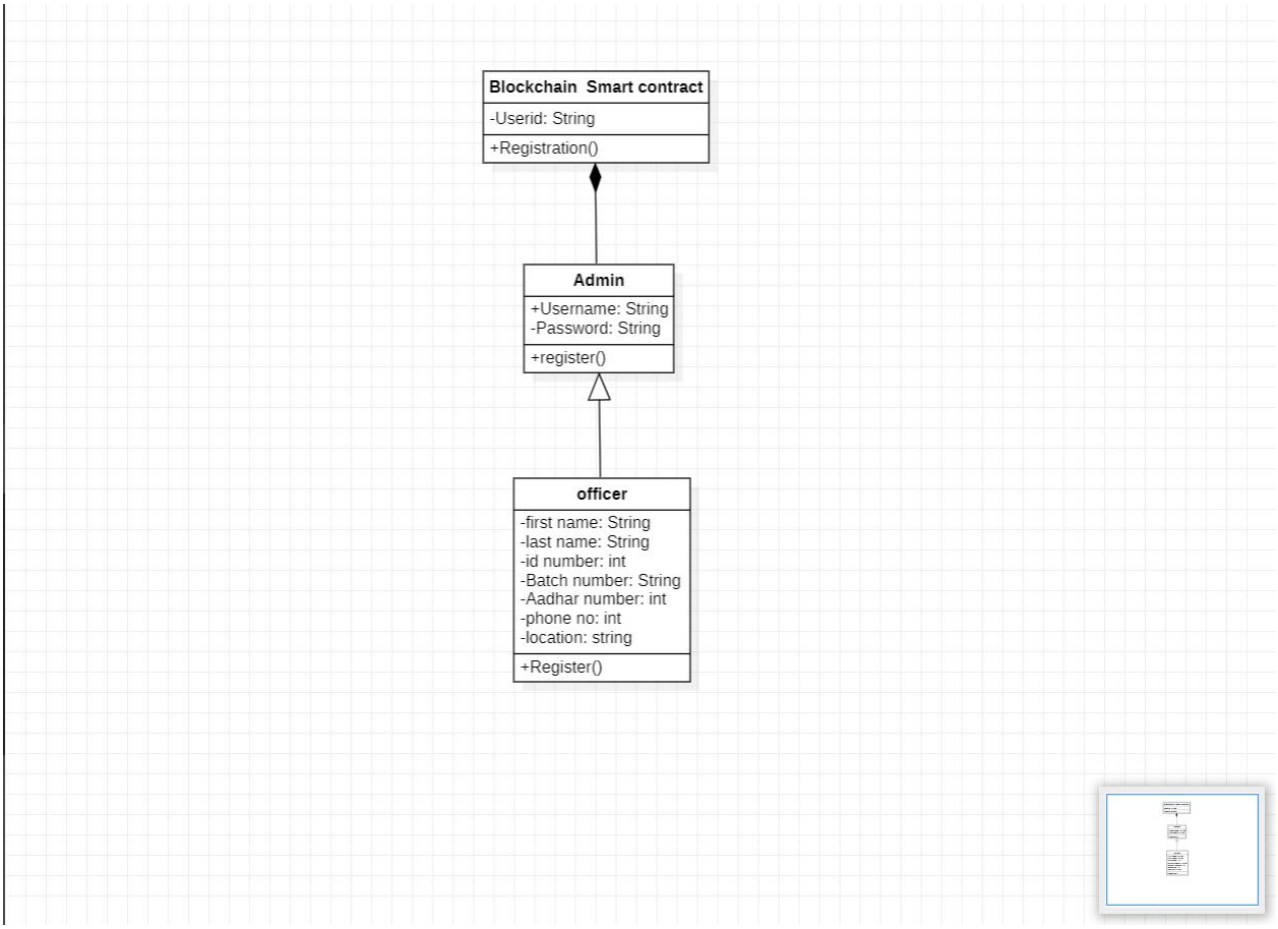


Figure 3.3: Class Diagram for Admin and Officer details

3.4 SEQUENCE DIAGRAM

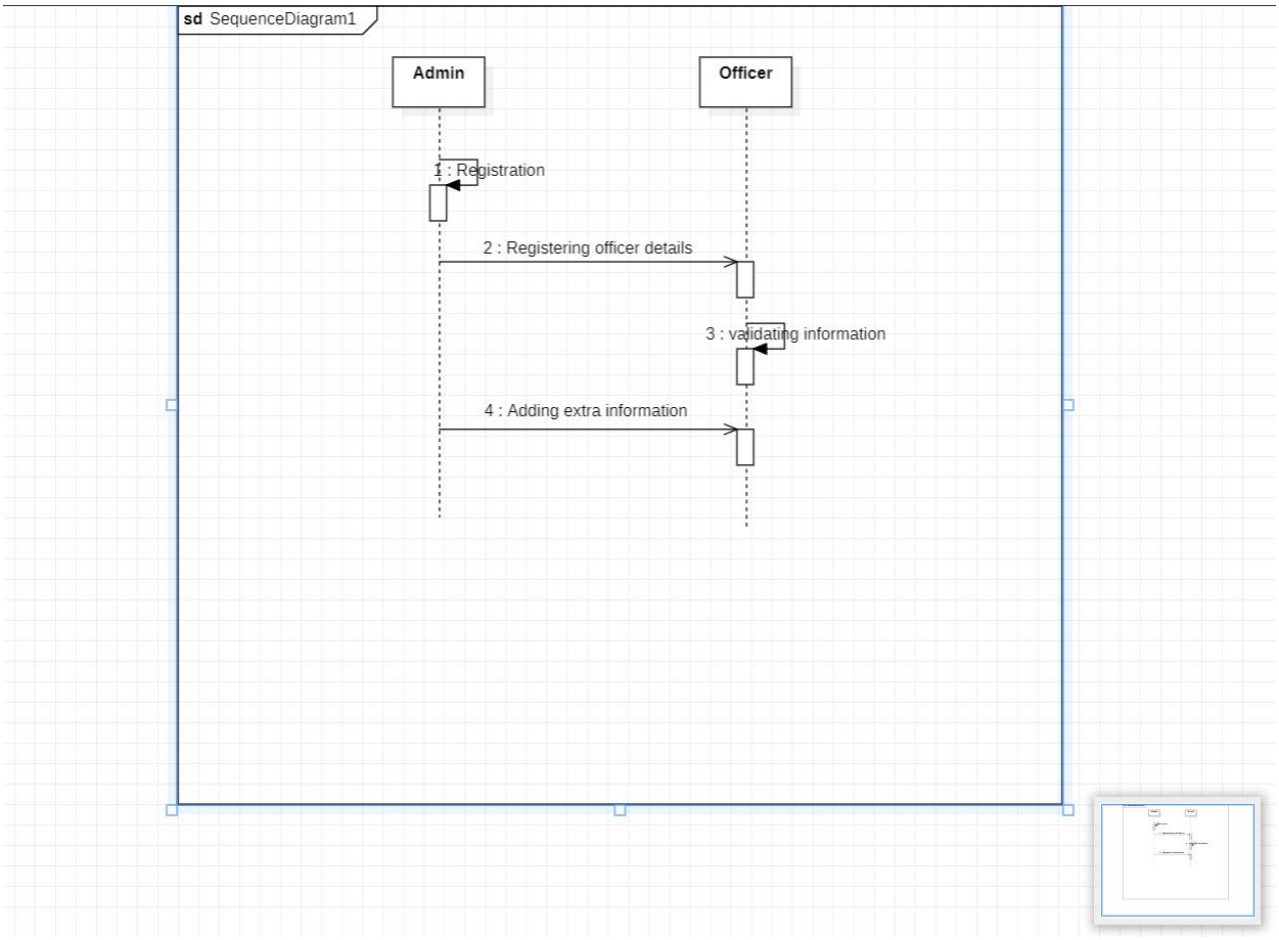


Figure 3.4: Sequence Diagram foradmin and officer details

4. IMPLEMENTATION

4. IMPLEMENTATION

4.1 INTRODUCTION

In this project we have used solidity to write smart contract , the web page is designed in HTML 5 and CSS 3.Web 3 scripts are written in node.js(javascript).

4.2 SAMPLE CODE

```
// SPDX-License-Identifier: MIT

pragma solidity ^0.8.10;

contract DefencesDataSecurity

{

   //Officer Details _1

  struct officerDetails_1

  {

     string firstName;

     string lastName;

     uint idNumber;

     string rank;

     uint aadharNumber;

     uint phoneNumber;

     string location;

  }

mapping (uint => officerDetails_1)officerMap;

//set Officer Details _1

   function setOfficerDetails_1(string memory _firstName,

                                string memory _lastName,
```

```
                    uint _idNumber,
                    string memory _rank,
                    uint _aadharNumber,
                    uint  _phoneNumber,
                    string memory _location
                    )
    public
    {
        officerMap[_idNumber].firstName = _firstName;
        officerMap[_idNumber].lastName = _lastName;
        officerMap[_idNumber].idNumber = _idNumber;
        officerMap[_idNumber].rank = _rank;
        officerMap[_idNumber].aadharNumber = _aadharNumber;
        officerMap[_idNumber].phoneNumber = _phoneNumber;
        officerMap[_idNumber].location = _location;
    }
//Get Officer Details _1
        function getOfficerDetails_1(uint _idNumber)
view public returns (string memory,
                    string memory,
                    uint,
                    string memory,
                    uint,
                    uint,
                    string memory
```

```
                    )
    {
        return
        (
            officerMap[_idNumber].firstName,
            officerMap[_idNumber].lastName,
            officerMap[_idNumber].idNumber,
            officerMap[_idNumber].rank,
            officerMap[_idNumber].aadharNumber,
            officerMap[_idNumber].phoneNumber,
            officerMap[_idNumber].location
        );
    }
//Admin And Officer  Verification

    struct verificationDetails
    {
        string publicKey;
        string privateKey;
    }
        mapping (string => verificationDetails) officerVerificationDetailsMap;
        mapping (string => verificationDetails) adminVerificationDeatilsMap;

//The Main Admin To Everything, Can Access Everything
    constructor()
```

```
    public
    {
        keccak256(abi.encodePacked((officerVerificationDetailsMap["5f5"].publicKey = "5f5")));

        keccak256(abi.encodePacked((officerVerificationDetailsMap["5f5"].privateKey = "5b8")));
  keccak256(abi.encodePacked((adminVerificationDeatilsMap["5f5"].publicKey = "5f5")));
keccak256(abi.encodePacked((adminVerificationDeatilsMap["5f5"].privateKey = "5b8")));
    }
\//Set Verification Details
    function setOfficerVerificationDetails(string memory _publicKey, string memory _privateKey)
 public
 {

keccak256(abi.encodePacked((officerVerificationDetailsMap[_publicKey].publicKey =
_publicKey)));
keccak256(abi.encodePacked((officerVerificationDetailsMap[_publicKey].privateKey =
_privateKey)));
                        }
//Get verifying Admin And Officer
  function getVerifyingAdminAndOfficer(string memory _publicKey, string memory _privateKey)
    view public returns(uint)
    {
        uint confirm = 0;
         if((keccak256(abi.encodePacked((_publicKey))) == keccak256(abi.encodePacked(("")))) ||
(keccak256(abi.encodePacked((_privateKey))) == keccak256(abi.encodePacked(("")))))
        {
```

```
            confirm = 3;


        }
            else
            {
                if((keccak256(abi.encodePacked((_publicKey))) ==
keccak256(abi.encodePacked((officerVerificationDetailsMap[_publicKey].publicKey))))
                || (keccak256(abi.encodePacked((_publicKey))) ==
keccak256(abi.encodePacked((adminVerificationDeatilsMap[_publicKey].publicKey)))))
                {
                    if((keccak256(abi.encodePacked((_privateKey))) ==
keccak256(abi.encodePacked((officerVerificationDetailsMap[_publicKey].privateKey))))
                    || (keccak256(abi.encodePacked((_privateKey))) ==
keccak256(abi.encodePacked((adminVerificationDeatilsMap[_publicKey].privateKey)))))
                    {
                        confirm = 1;
                    }
                }
            }


        return (confirm);
    }
//Set Admin Verification Details


  function setAdminVerificationDetails(string memory _publicKey, string memory _privateKey)
                                public
```

```
                    {

keccak256(abi.encodePacked((adminVerificationDeatilsMap[_publicKey].publicKey =
_publicKey)));
keccak256(abi.encodePacked((adminVerificationDeatilsMap[_publicKey].privateKey =
_privateKey)));


                    }
//Get verifying Admin
function getVerifyingAdmin(string memory _publicKey, string memory _privateKey)
   view public returns(uint)
   {
      uint confirm = 0;
     if((keccak256(abi.encodePacked((_publicKey))) == keccak256(abi.encodePacked(("")))) ||
(keccak256(abi.encodePacked((_privateKey))) == keccak256(abi.encodePacked(("")))))
      {
         confirm = 3;
        }
         else
         {
            if(keccak256(abi.encodePacked((_publicKey))) ==
keccak256(abi.encodePacked((adminVerificationDeatilsMap[_publicKey].publicKey))))
            {
               if(keccak256(abi.encodePacked((_privateKey))) ==
keccak256(abi.encodePacked((adminVerificationDeatilsMap[_publicKey].privateKey))))
               {
```

```
                confirm = 1;
            }
        }
    }
    return (confirm);
  }

}
```

Index.js

```
// Import Web3 JS library
const Web3 = require('web3');
const web3 = new Web3("HTTP://127.0.0.1:7545");
// Import the ABI definition of the DefencesDataSecurity
const artifact = require('../../build/contracts/DefencesDataSecurity.json');
const deployedContract = artifact.networks[5777];
const contractAddress = deployedContract.address;
const MIN_GAS = 1000000;
const App =
{
  web3: null,
  contractInstance: null,
  accounts: null,

  start: async function()
```

```
  {
     const { web3 } = this;
     // Get the accounts
     this.accounts = await web3.eth.getAccounts();
console.log(this.accounts);
this.contractInstance = new web3.eth.Contract(
        artifact.abi,
        contractAddress
     );
  },


   //SET OFFICER DETAILS_1


     setOfficerDetails_1: async function()
     {
       const _firstName = document.getElementById('setFirstName').value;
        const _lastName = document.getElementById('setLastName').value;
        const _idNumber = document.getElementById('setidNumber').value;
        const _rank = document.getElementById('setRank').value;
        const _aadharNumber = document.getElementById('setAadharNumber').value;
        const _phoneNumber = document.getElementById('setPhoneNumber').value;
        const _location = document.getElementById('setLocation').value;



        console.log(_firstName,_lastName,
```

```
            _idNumber,
            _rank,
            _aadharNumber,
            _phoneNumber,_location);
          const gas = await this.contractInstance.methods.setOfficerDetails_1(_firstName,_lastName,
                    _idNumber,
                    _rank,
                    _aadharNumber,
                    _phoneNumber,_location).estimateGas
          ({
            from: this.accounts[0]
          });
          await this.contractInstance.methods.setOfficerDetails_1(_firstName,_lastName,
            _idNumber,
            _rank,
            _aadharNumber,
            _phoneNumber,_location).send
          ({
            from: this.accounts[0], gas: Math.max(gas, MIN_GAS)
          });
        },

  //GET OFFICER DETAILS_1
getOfficerDetails_1: async function()
        {
```

```
   const _idNumber = document.getElementById('inputIdNumber').value;

  await
this.contractInstance.methods.getOfficerDetails_1(_idNumber).call().then(function(officerDetails_
1)

          {

             document.getElementById("getFirstName").innerHTML = "First Name: " +
officerDetails_1[0];

             document.getElementById("getLastName").innerHTML = "Last Number: " +
officerDetails_1[1];

             document.getElementById("getIdNumber").innerHTML = "ID Number: " +
officerDetails_1[2];

             document.getElementById("getRank").innerHTML = "Rank: " + officerDetails_1[3];

             document.getElementById("getAadharNumber").innerHTML = "Aadhar Number: " +
officerDetails_1[4];

             document.getElementById("getPhoneNumber").innerHTML = "Phone Number: " +
officerDetails_1[5];

             document.getElementById("getLocation").innerHTML = "Location: " +
officerDetails_1[6];

          });

       },









   //SET Officer VERIFICATION DETAILS

     setOfficerVerificationDetails: async function()

       {
```

```
        const _publicKey = document.getElementById('setOfficerUserName').value;
         const _privateKey = document.getElementById('setOfficerPassword').value;
          console.log(_publicKey,_privateKey);

         const gas = await
this.contractInstance.methods.setOfficerVerificationDetails(_publicKey,_privateKey).estimateGas
         ({
            from: this.accounts[0]
         });

         await
this.contractInstance.methods.setOfficerVerificationDetails(_publicKey,_privateKey).send
         ({
            from: this.accounts[0], gas: Math.max(gas, MIN_GAS)
         });
      },
  //GET VERIFYING ADMIN And OFFICER

      getVerifyingAdminAndOfficer: async function()
      {
         const _publicKey = document.getElementById('inputAdminOrOfficerUserName').value;
const _privateKey = document.getElementById('inputAdminOrOfficerPassword').value;
         if(_publicKey == "" || _privateKey == "")
         {
        window.alert("enter user name and password..!!");
```

```
        }
          else
        {
            await this.contractInstance.methods.getVerifyingAdminAndOfficer(_publicKey,
_privateKey).call().then(function(getVerifyingAdminAndOfficer)
            {
               const confirm = getVerifyingAdminAndOfficer[0];
 if(confirm == 0)
               {
                    window.alert("enter correct User Name and Password..!!");
               }
 else
               {
                  if(confirm == 1)
                  {
                     window.location = "html/officerDetails_1.html";
                  }
               }
            });
        }
          },
   //SET Admin VERIFICATION DETAILS
  setAdminVerificationDetails: async function()
     {
       const _publicKey = document.getElementById('setAdminUserName').value;
```

```
  const _privateKey = document.getElementById('setAdminPassword').value;
          console.log(_publicKey,_privateKey);
        const gas = await
this.contractInstance.methods.setAdminVerificationDetails(_publicKey,_privateKey).estimateGas
        ({
           from: this.accounts[0]
        });
        await
this.contractInstance.methods.setAdminVerificationDetails(_publicKey,_privateKey).send
        ({
           from: this.accounts[0], gas: Math.max(gas, MIN_GAS)
        });
      },
   //GET VERIFYING Admin

      getVerifyingAdmin: async function()
      {
        const _publicKey = document.getElementById('inputAdminUserName').value;

        const _privateKey = document.getElementById('inputAdminPassword').value;
        if(_publicKey == "" || _privateKey == "")
        {
       window.alert("enter user name and password..!!");
        }
        else
```

```
        {
            await this.contractInstance.methods.getVerifyingAdmin(_publicKey,
_privateKey ).call().then(function(getVerifyingAdmin)
            {
                const confirm = getVerifyingAdmin[0];
 if(confirm == 0)
                {
          window.alert("enter correct User Name and Password..!!");
                }
else
                {
                    if(confirm == 1)
                    {
                        window.location = "adminAndOfficerVerificationDetails.html";
                    }
                }
            });
        }

      },
}
window.App = App;
window.addEventListener("load", function() {
   App.web3 = new Web3(
     new Web3.providers.HttpProvider("http://127.0.0.1:7545"),
```

```
    );
 App.start();
});
```

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link href="css/styles.css" rel="stylesheet">
  <link href="css/bootstrap.css" rel="stylesheet">
  <!-- fontawesome link -->
  <link rel="stylesheet" href="https://pro.fontawesome.com/releases/v5.10.0/css/all.css"
integrity="sha384-
AYmEC3Yw5cVb3ZcuHtOA93w35dYTsvhLPVnYs9eStHfGJvOvKxVfELGroGkvsg+p"
crossorigin="anonymous" />

  <title>Defences Data Security</title>
</head>
<body style="background-color: #c4c4c4;">
<!--Navigation Bar-->
  <nav class="navbar navbar-light">
    <div class="container-fluid">
```

```
    <a class="navbar-brand" href="#">
      <i class="fas fa-cubes fundIcon"></i>
      <p class="reliefFund">
        Defences Data Security
      </p>
    </a>
  </div>
</nav>
<!-- body -->
<div class="container">
  <div class="row">
    <div class="col">
      <img src="images/bcicon.png" alt="icon">
    </div>
      <div class="col">
        <div class="registerform">
          <div class="title">
            Enter Login Credentials
          </div>
          <div class="form">
            <div class="inputfield">
              <label>User Name</label>
              <input type="password" class="input" id="inputAdminOrOfficerUserName"
placeholder="Enter Your User Name">
            </div>
```

```
            <div class="inputfield">
              <label>Password</label>
              <input type="password"  class="input" id="inputAdminOrOfficerPassword"
placeholder="Enter Your Password">
            </div>
            <div class="text-center">
              <button type="submit" id="loginCredentialsSubmitButton" class="btn btn-dark "
onclick="App.getVerifyingAdminAndOfficer()">Submit</button>
            </div>
           </div>
          </div>
         </div>
 <script src="index.js"></script>
</body>
</html>
```

Admin.verification.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link href="../css/styles.css" rel="stylesheet">
  <link href="../css/bootstrap.css" rel="stylesheet">
```

```
  <!-- fontawesome link -->
  <link rel="stylesheet" href="https://pro.fontawesome.com/releases/v5.10.0/css/all.css"
integrity="sha384-
AYmEC3Yw5cVb3ZcuHtOA93w35dYTsvhLPVnYs9eStHfGJvOvKxVfELGroGkvsg+p"
crossorigin="anonymous" />
 <title>Set Officer Details </title>
 <script src="../index.js"></script>
</head>
<body style="background-color: #c4c4c4" ;>
  <!-- Navigation bar -->
<nav class="navbar navbar-light">
   <div class="container-fluid">
     <a class="navbar-brand" href="#">
       <i class="fas fa-cubes fundIcon"></i>
       <p class="reliefFund">
         Admin Verification
       </p>
     </a>
   </div>
 </nav>

<div class="container">
   <div class="row">
        <div class="col">
          <div class="registerform">
            <div class="title">
              Enter Login Credentials
            </div>
            <div class="form">
              <div class="inputfield">
                <label>User Name</label>
```

```
            <input type="password" class="input" id="inputAdminUserName" placeholder="Enter Your User Name">
            </div>
            <div class="inputfield">
              <label>Password</label>
              <input type="password"  class="input" id="inputAdminPassword" placeholder="Enter Your Password">
            </div>
            <div class="text-center">
              <button type="submit" id="adminLoginCredentialsSubmitButton" class="btn btn-dark " onclick="App.getVerifyingAdmin()">Submit</button>
            </div>
           </div>
          </div>
       </div>
  <script src="../index.js"></script>
  </script>
</body>
</html>
```

adminAndOfficerVerificationDetails.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <link href="../css/styles.css" rel="stylesheet">
  <link href="../css/bootstrap.css" rel="stylesheet">
  <!-- <link href="html/officerDetails_1.html" rel="Registration"> -->
  <!-- fontawesome link -->
```

```
 <link rel="stylesheet" href="https://pro.fontawesome.com/releases/v5.10.0/css/all.css"
integrity="sha384-
AYmEC3Yw5cVb3ZcuHtOA93w35dYTsvhLPVnYs9eStHfGJvOvKxVfELGroGkvsg+p"
crossorigin="anonymous" />
 <title>Create New Account</title>
 <script src="../index.js"></script>
</head>
<body style="background-color: #c4c4c4" ;>
 <!-- Navigation bar -->
<nav class="navbar navbar-light">
   <div class="container-fluid">
     <a class="navbar-brand" href="#">
       <i class="fas fa-cubes fundIcon"></i>
       <p class="reliefFund">
         Admin And Officer Verification Details
       </p>
     </a>
   </div>
 </nav>

<!-- Registration Form -->
 <div class="container">
   <div class="row">
     <div class="col">
       <div class="registerform">
         <div class="title">
           Set Officer Verification Details
         </div>
         <div class="form">
           <div class="inputfield">
             <label>User Name</label>  <!--Public Key -->
```

```
          <input type="password" class="input" id="setOfficerUserName" placeholder="Enter Public Key..!!">
          </div>
          <div class="inputfield">
            <label>Password</label> <!--Private Key -->
          <input type="password"  class="input" id="setOfficerPassword" placeholder="Enter Private Key..!!">
          </div>
          <div class="text-center">
          <button type="submit" class="btn btn-dark " id="setOfficerVerificationDetailsSubmitButton"
onclick="App.setOfficerVerificationDetails()">Submit</button>
          </div>
        </div>
      </div>
    </div>
  <!-- Registration Form -->
  <div class="container">
    <div class="row">
      <div class="col">
        <div class="registerform">
          <div class="title">
            Set Admin Verification Details
          </div>
          <div class="form">
            <div class="inputfield">
              <label>User Name</label> <!--Public Key -->
            <input type="password" class="input" id="setAdminUserName" placeholder="Enter Public Key..!!">
          </div>
          <div class="inputfield">
            <label>Password</label> <!--Private Key -->
```

```
          <input type="password" class="input" id="setAdminPassword" placeholder="Enter Private Key..!!">
          </div>
          <div class="text-center">
           <button type="submit" class="btn btn-dark " id="setAdminVerificationDetailsSubmitButton" onclick="App.setAdminVerificationDetails()">Submit</button>
          </div>
         </div>
        </div>
      </div>
  <script src="../index.js"></script>
  </script>
</body>
</html>
```

oficerDetails_1.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
<link href="../css/styles.css" rel="stylesheet">
  <link href="../css/bootstrap.css" rel="stylesheet">
<!-- <link href="../html/adminVerification.html" rel="registerform"> -->
<!-- fontawesome link -->
  <link rel="stylesheet" href="https://pro.fontawesome.com/releases/v5.10.0/css/all.css" integrity="sha384-AYmEC3Yw5cVb3ZcuHtOA93w35dYTsvhLPVnYs9eStHfGJvOvKxVfELGroGkvsg+p" crossorigin="anonymous" />
```

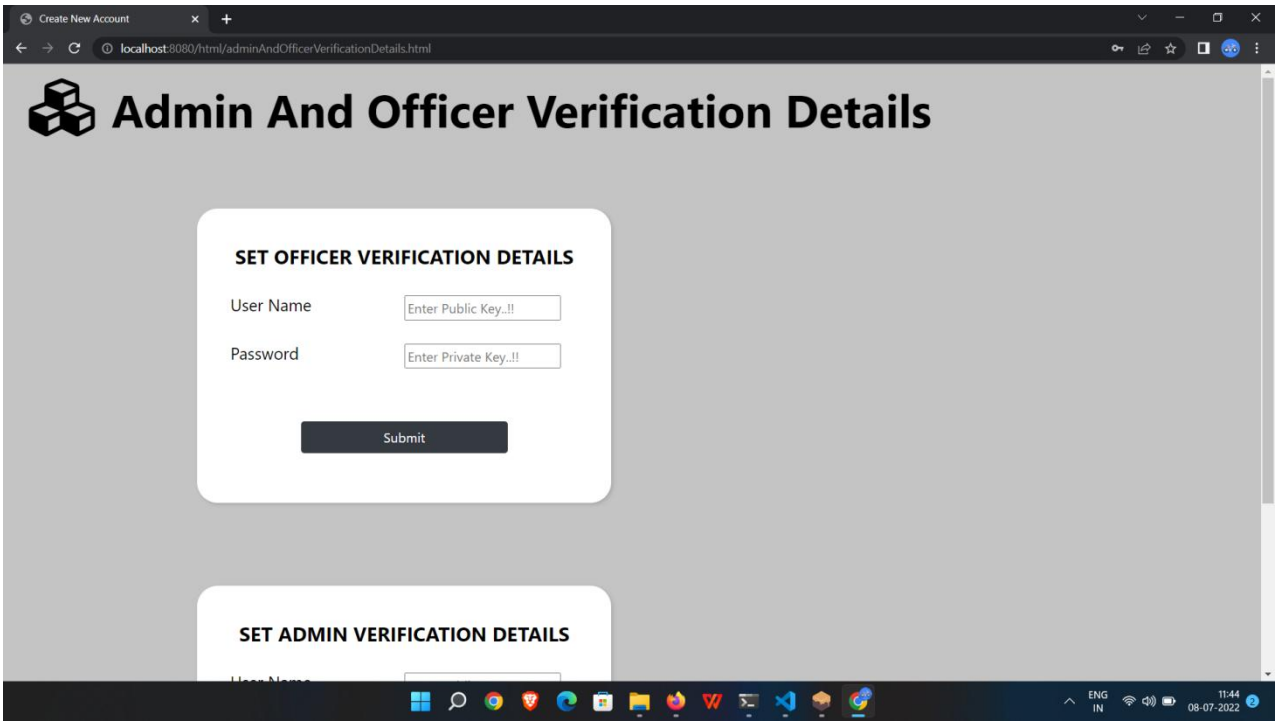
```
  <title>Set Officer Details </title>
 <script src="../index.js"></script>
</head>
<body style="background-color: #c4c4c4" ;>
 <!-- Navigation bar -->
<nav class="navbar navbar-light">
   <div class="container-fluid">
     <a class="navbar-brand" href="#">
       <i class="fas fa-cubes fundIcon"></i>
       <p class="reliefFund">
         Defences Data Security
       </p>
     </a>
   </div>
 </nav>
 <!-- Registration Form -->
 <div class="container">
   <div class="row">
     <div class="col">
       <div class="registerform">
         <div class="title">
           Set Officer Details 1
         </div>
         <div class="form">
           <div class="inputfield">
             <label>First Name</label>
             <input type="text" style="text-transform:lowercase" class="input" id="setFirstName" placeholder="enter your first name">
           </div>
           <div class="inputfield">
             <label>Last Name</label>
```

```
          <input type="text"  class="input" id="setLastName" placeholder="enter last name">
        </div>
        <div class="inputfield">
          <label>ID Number</label>
          <input type="number" class="input" id="setidNumber" placeholder="enter id number">
        </div>
        <div class="inputfield">
          <label>Rank</label>
          <input type="text" style="text-transform:lowercase" class="input" id="setRank" placeholder="enter rank">
        </div>
        <div class="inputfield">
          <label>Aadhar Number</label>
          <input type="number" class="input" id="setAadharNumber" placeholder="enter aadhar number">
        </div>
        <div class="inputfield">
          <label>Phone Number</label>
          <input type="number" class="input" id="setPhoneNumber" placeholder="enter phone number">
        </div>
        <div class="inputfield">
          <label>Location</label>
          <input type="text" class="input" id="setLocation" placeholder="enter location">
        </div>
    <div class="text-center">
          <button type="submit" class="btn btn-dark " onclick="App.setOfficerDetails_1()">Submit</button>
        </div>
       </div>
      </div>
     </div>
```

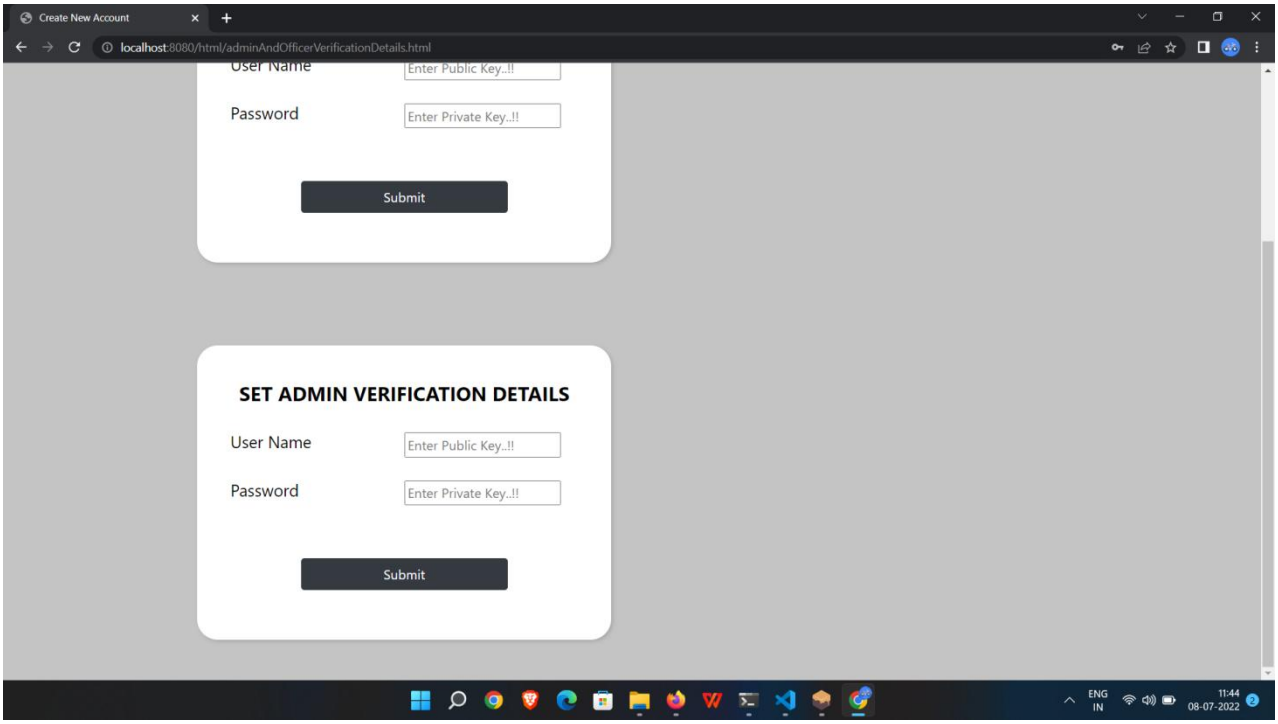
```
<!-- Get Officer Details_1 -->
 <div class="col">
      <div class="registerform">
        <div class="title">
          Get Officer Details 1
        </div>
        <div class="form">
          <div class="inputfield">
            <label>ID Number</label>
            <input type="number" class="input" id="inputIdNumber" placeholder="enter id number" required>
            <br>
          </div>
          <div class="form">
            <p class="textSize" id="getFirstName"></p>
            <p class="textSize" id="getLastName"></p>
            <p class="textSize" id="getIdNumber"></p>
            <p class="textSize" id="getRank"></p>
            <p class="textSize" id="getAadharNumber"></p>
            <p class="textSize" id="getPhoneNumber"></p>
            <p class="textSize" id="getLocation"></p>
            </div>
          <div class="text-center">
            <button type="submit" class="btn btn-dark " onclick="App.getOfficerDetails_1()">Get Details</button>
          </div>
        </div>
      </div>
    </div>
  <div class="col">
      <div class="buttonDivision">
```

```
        <a href="adminVerification.html">
          <button type="button" class="btnForMainPage">Give Access To New Officer</button>
        </a>
       </div>
     </div>
 <script src="../index.js"></script>
  </script>
</body>
</html>
```

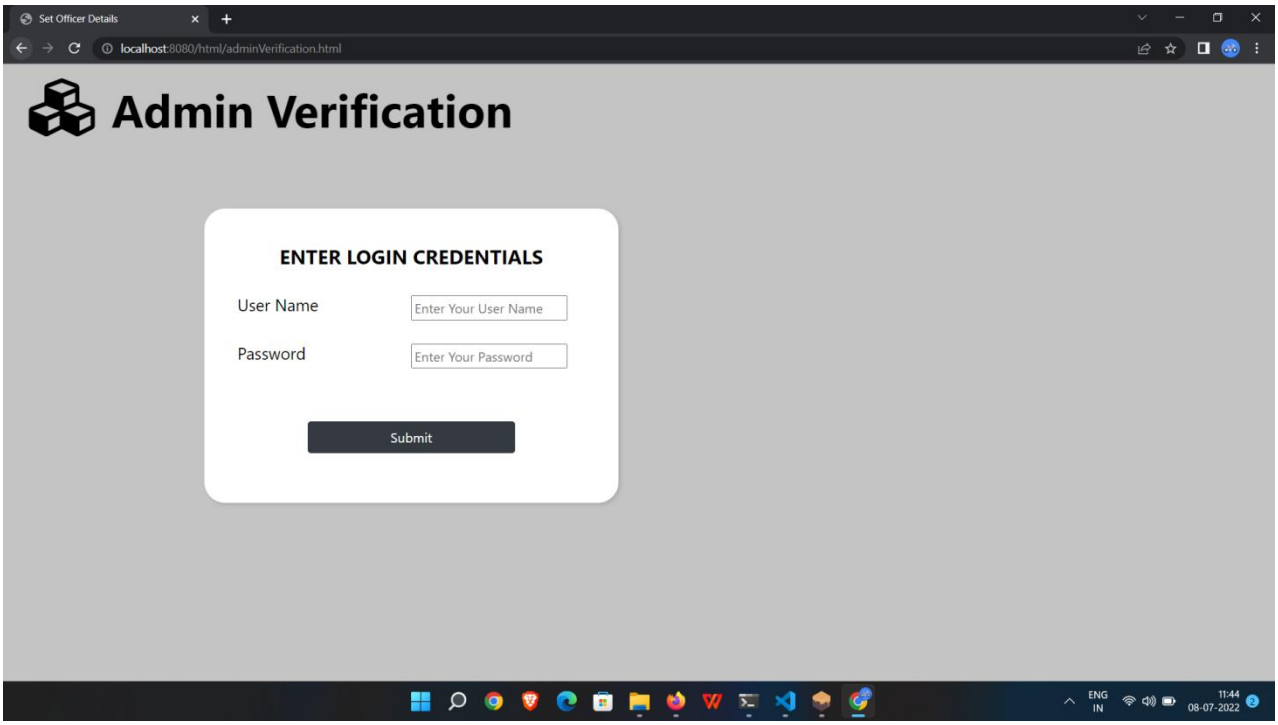
5. SCREENSHOTS



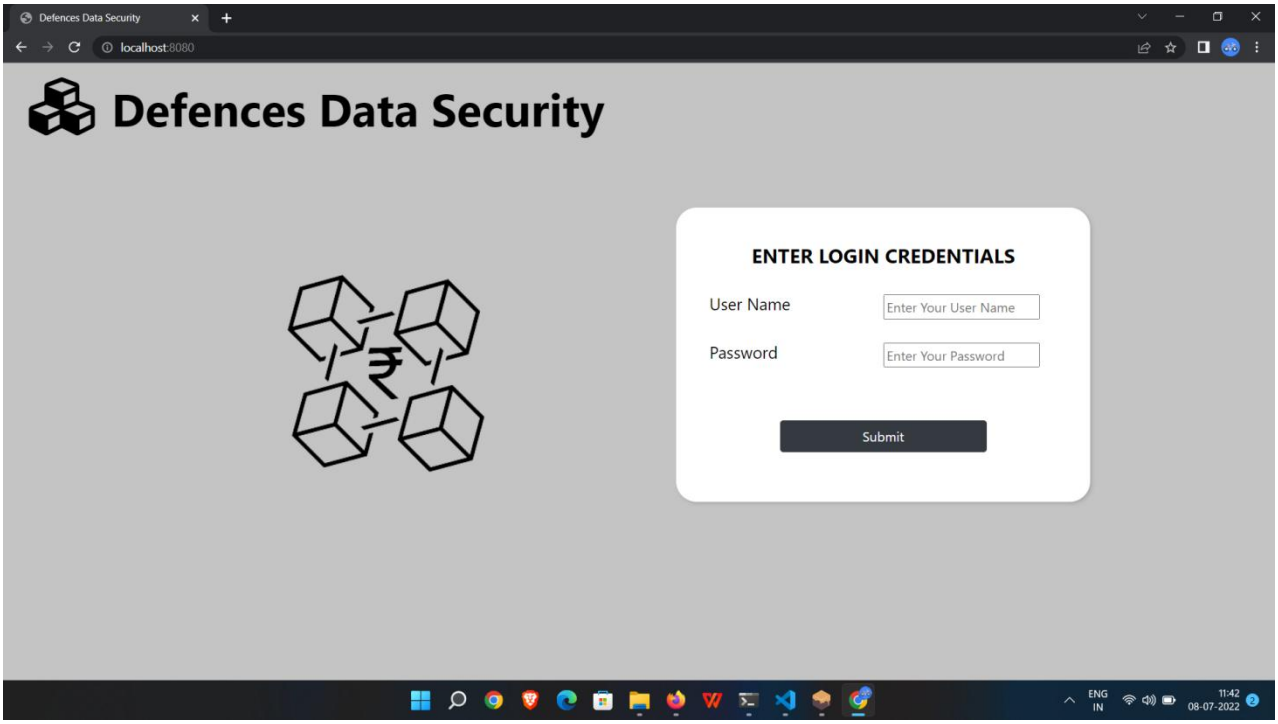
Screenshot 5.1: Admin And Officer Verification Details



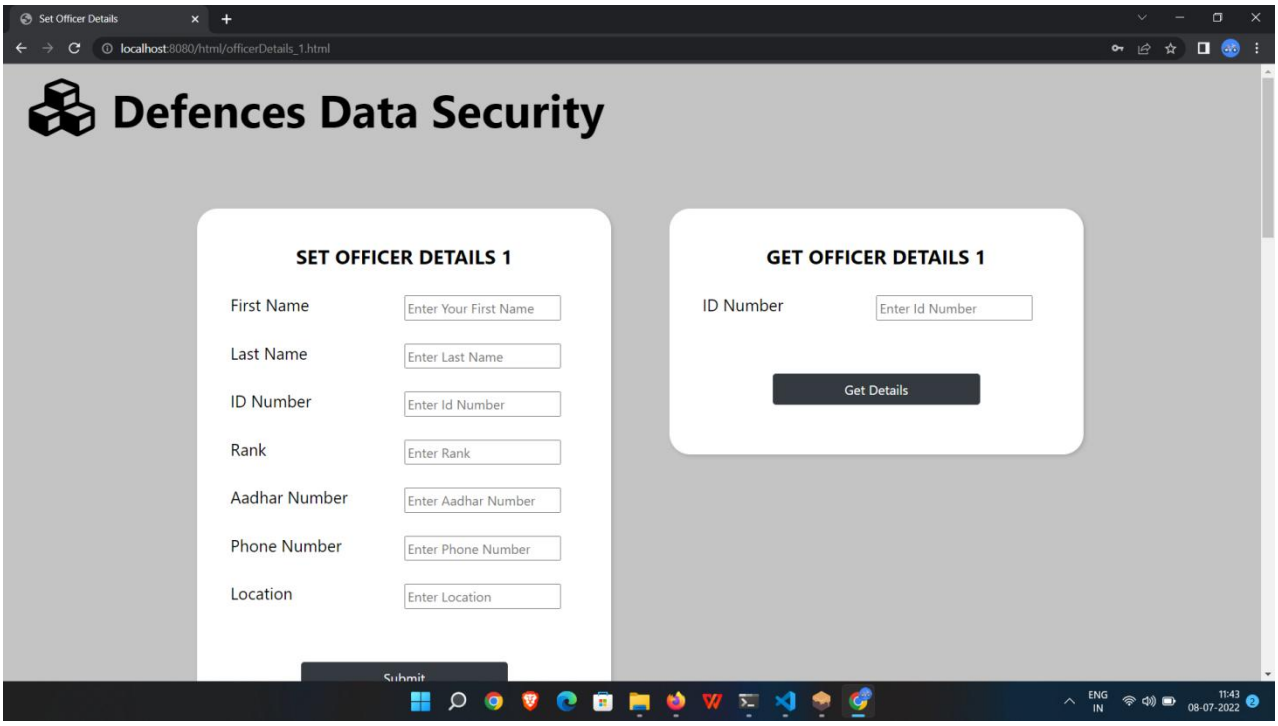
Screenshot 5.2: Set Admin Verification Details



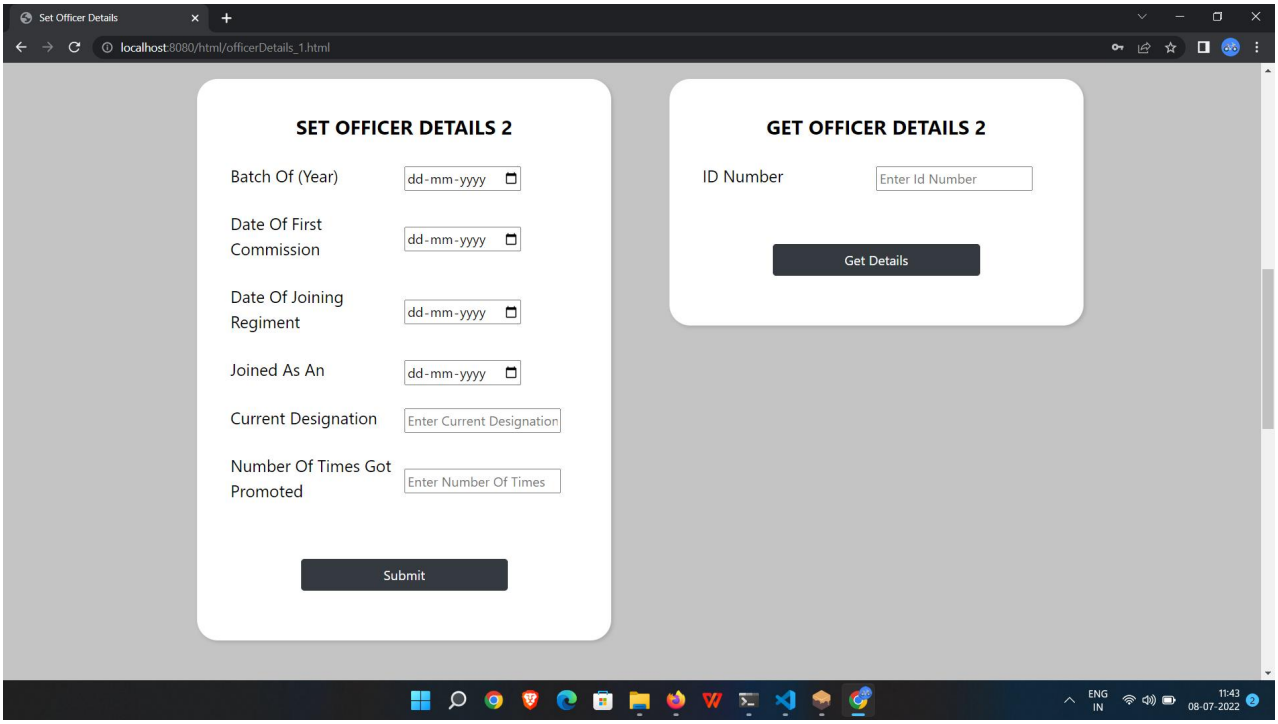
Screenshot 5.3: Admin Verification



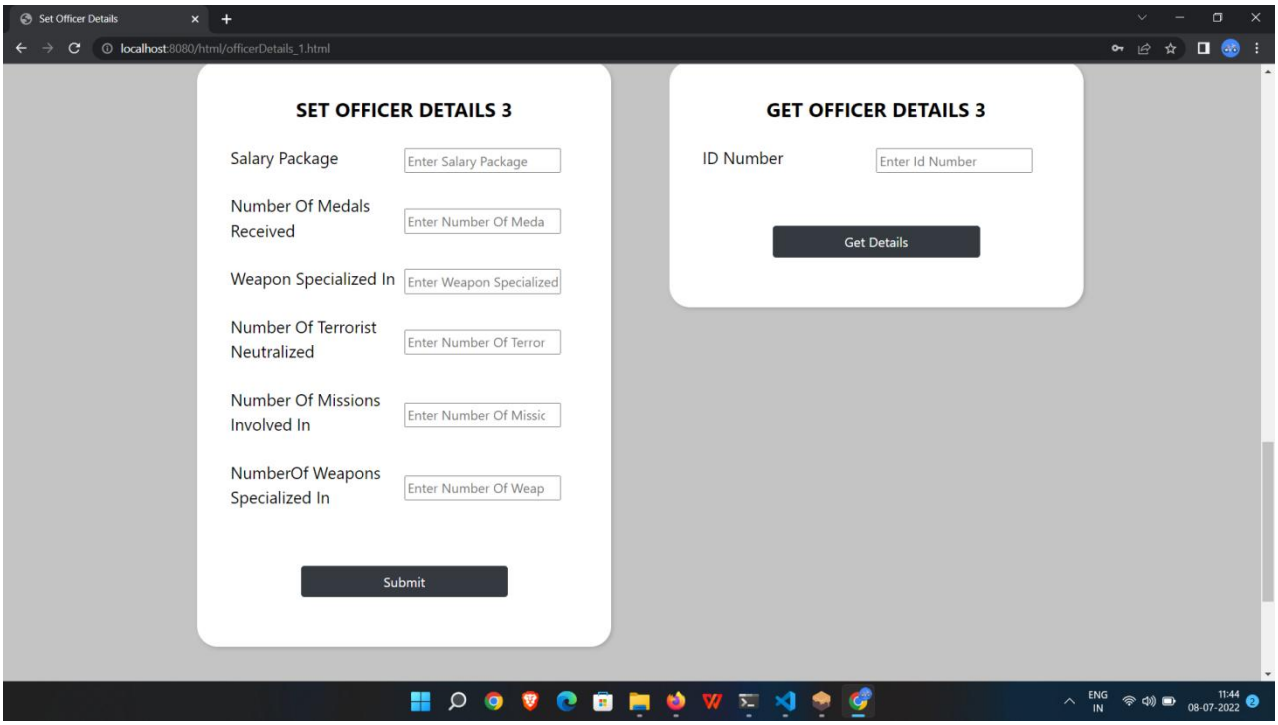
Screenshot 5.4: Enter Login Credentials



Screenshot 5.5: Setting and Getting Officer Details 1



Screenshot 5.5: Setting and Getting Officer Details 2



Screenshot 5.4: Setting and Getting Officer Details 3

6. TESTING

6. TESTING

6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discoverevery conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.2 TYPES OF TESTING

6.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes.

6.3 TEST CASES

6.3.1 UPLOADING IMAGES

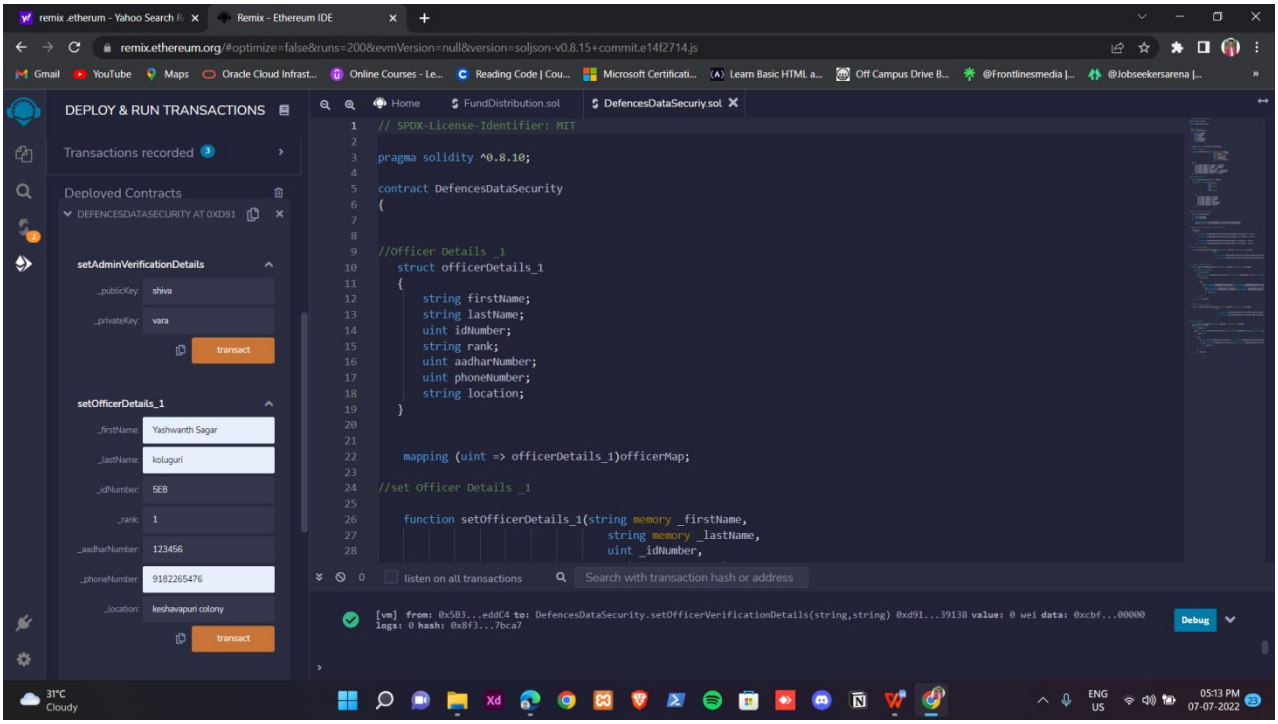


Fig 6.3.1: setting admin verification details and officer detail

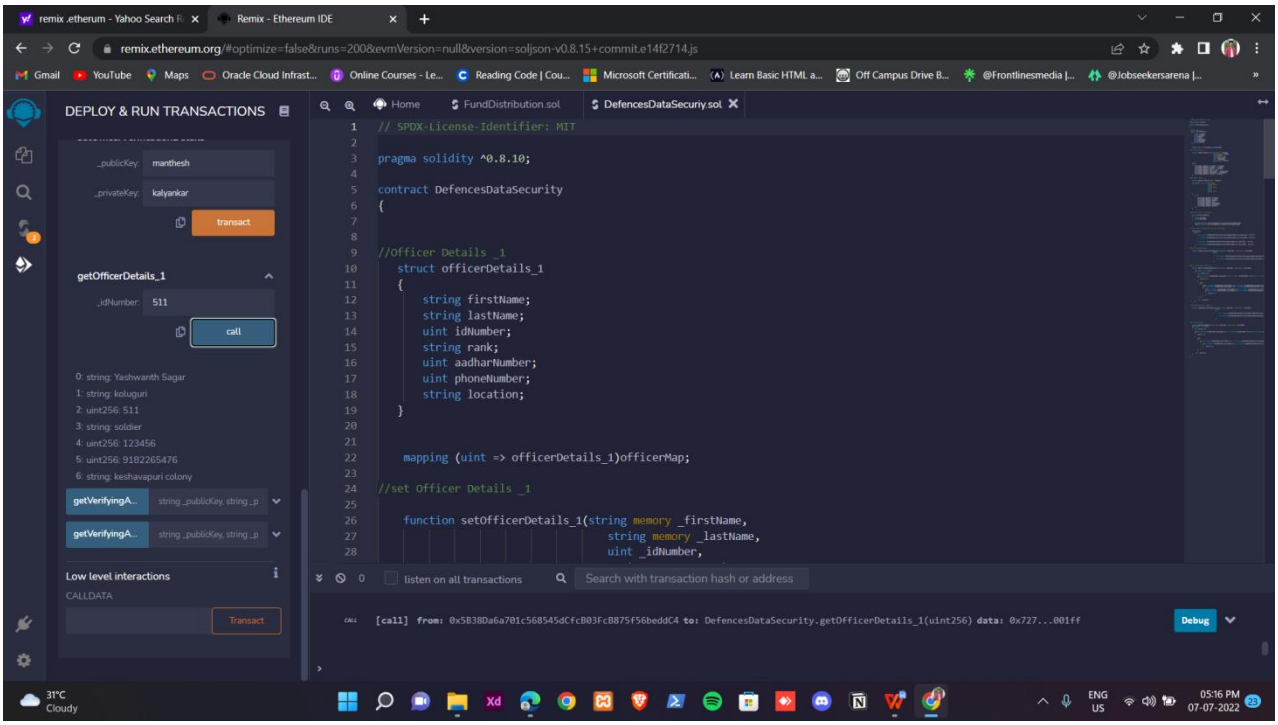


Fig 6.3.2:Getting officer details

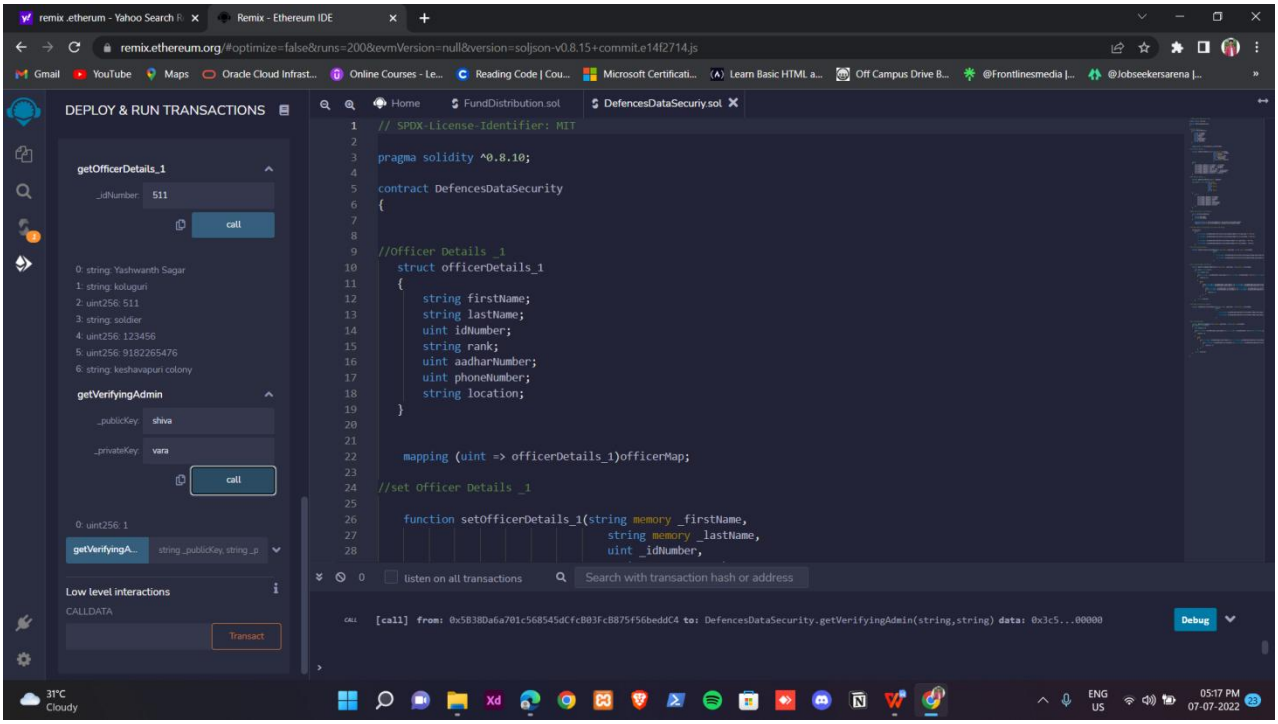


Fig 6.3.3 :Verifiying admin and officer

7. CONCLUSION

7. CONCLUSION & FUTURE SCOPE

7.1 PROJECT CONCLUSION

As discussed before, blockchain technology has the potential to radically change our way of life, and the way we conduct military operations, both on an operational and support level. Thanks to its decentralised and transparent nature, it could improve the decisions taken by military officials, while enhancing outcomes for military deployments. The development of blockchain technology offers increased data confidence and data availability that can help shape future military logistics and planning. As we saw, the US intends to use it for secured databases, logistics and 3D printing, similarly to China and Russia. The EU is also eager to invest in blockchain and will have the possibility to directly fund blockchain technology related to military fields thanks to the upcoming Horizon Europe framework and the European Defence Fund.

7.2 FUTURE SCOPE

So, is blockchain in the military an evolution or a revolution? An evolution would simply modify how the current military tools are used, while a revolution would dramatically change the tools themselves. Realistically speaking, at present, the evidence suggests an evolution, but not yet a "Revolution in Military Affairs" scale seachange. Blockchain will make communications more secure and facilitate military logistics. Henceforth, blockchain will strengthen and make armed forces more efficient. In the long term, blockchain in the military will be a revolution if it is well implemented and many more military applications are found, in addition to being used wisely and at affordable costs. Still, right now, we are instead witnessing a significant evolution than a real revolution of how armed forces and national DoD operate. Slowly but surely, we can agree that blockchain is becoming a game-changer for the security and efficiency of current military tools, especially if the biggest military actors start implementing it widely; as we saw above, the race between state actors in this field has already begun.

8. BIBLIOGRAPHY

8. BIBLIOGRAPHY

8.1 REFERENCES

• Beck, R; Müller-Bloch, C. (2017). Blockchain as Radical Innovation: A Framework for

aging with Distributed Ledgers. Proceedings of the 50th Hawaii International Conference

on System Sciences. pp. 5390-5399. Available at: https://pdfs.semanticscholar.org/cdc3/a80f5c77270bd36f1a0212bccea8651de3d4.pdf

• Caron, F. (2017). Blockchain: Identifying risk on the road to distributed ledgers, Isaca Journal. Vol. 5.

• Fitzi, M., Gazi, P., Kiayias, A., & Russell, A. (2018). Parallel Chains: Improving Throughput and Latency of blockchain Protocols via Parallel Composition. IACR Cryptology ePrint Archive, 2018, 1119.

• Linkov, I.; Wells, E.; Trump, B.; Collier, Z.; Goerger, S.; Lambert, J. (2018). Blockchain Benefits and Risks. Military Engineer. 110.

• McAbee, A. S. M.; Tummala, M.; McEachen, J. C. (2019), "Military Intelligence Applications for blockchain Technology", Proceedings of the 52nd Hawaii International Conference on System Sciences. pp. 6031-6040.

• Schrepel, T. (2018). "Is blockchain the Death of Antitrust Law? The blockchain Antitrust Paradox", Georgetown Law Technology Review, Vol. 3, N°281, June 11, 2018, available at: https://ssrn.com/abstract=3193576.