

Blockchain

GROUP-4

MSCIT PART-1

Aim:Blockchain in Supply Chain Management

Description :

A supply chain in blockchain is a digital record-keeping system that uses blockchain technology to track the movement of goods and services through a supply chain. It is a distributed ledger that is shared by all participants in the supply chain, and it is immutable, meaning that data cannot be tampered with or altered.

Blockchain technology has the potential to revolutionize supply chain management by providing a number of benefits, including:

Transparency: Blockchain provides complete transparency into the supply chain, allowing all participants to see where goods and services are at any given time. This can help to identify and prevent fraud, counterfeiting, and other problems.

Traceability: Blockchain makes it easy to track the provenance of goods and services, from raw materials to finished products. This can help to ensure that products are sourced from ethical suppliers and that they meet quality and safety standards.

Efficiency: Blockchain can help to streamline supply chain processes and reduce costs by eliminating the need for intermediaries and by automating tasks such as payments and inventory management.

Security: Blockchain is a very secure technology, and it can help to protect supply chains from cyberattacks.

Code:

Server.py:

```
import shutil

from flask import Flask, render_template, request, redirect, url_for

import os

import json
```

```

import hashlib

import datetime


app = Flask(__name__)


def get_previous_hash():

    folder_path = 'emplogin'

    folder_len = len(os.listdir(folder_path))

    if folder_len == 0:

        return '0000000000000000'


    prev_block_filename = f"{folder_len}.json"

    prev_block_path = os.path.join('emplogin', prev_block_filename)


    # Read the previous block's data and return its hash

    with open(prev_block_path, 'r') as file:

        prev_block_data = json.load(file)

        return prev_block_data.get('hash', '0')


@app.route('/')

def home():

    return render_template("supply.html")


@app.route('/employee', methods=['POST'])

def cust():

```

```

#Taking form data

email = request.form.get('email')

password = request.form.get('password')


#Checking if any field is empty

if not all([email, password]):

    return "All fields must be filled."


#Checking if the provided email and password match the data in the
folder

folder_path = 'employlogin'

found = False


for file_name in os.listdir(folder_path):

    file_path = os.path.join(folder_path, file_name)

    with open(file_path, 'r') as file:

        data = json.load(file)

        if data['email'] == email and data['password'] == password:

            found = True

            break


if found:

    # If email and password matches, take the user to the next page

    return render_template('employee.html')

else:

```

```

        return "Login failed. Please check your email and password."

@app.route('/adminlogin')

def adminlogin():

    return render_template("adminlogin.html")

@app.route('/admin', methods=['POST'])

def admin():

    if request.method == 'POST':

        email = request.form.get('email')

        password = request.form.get('password')

        if not all([email, password]):

            return "All fields must be filled."

        # Check if the provided email and password match the data in
        the "admin" folder

        folder_path = 'admin'

        found = False

        for file_name in os.listdir(folder_path):

            file_path = os.path.join(folder_path, file_name)

            with open(file_path, 'r') as file:

                data = json.load(file)

                if data['email'] == email and data['password'] ==
password:

```

```
        found = True

        break

    if found:

        # If email and password matches, take the admin to the next
page

        return render_template('admin.html')

    else:

        return "Admin login failed. Please check your email and
password."


@app.route('/addemp')

def addemp():

    return render_template("addemp.html")


@app.route('/reg', methods=['POST'])

def register():

    #Taking form data

    email = request.form.get('email')

    mobile = request.form.get('mobile')

    password = request.form.get('password')

    re_password = request.form.get('repassword')

    #Checking if any field is empty
```

```
if not all([email, mobile, password, re_password]):

    return "All fields must be filled."


#If passwords are not same

elif password != re_password:

    return "Passwords do not match."


elif len(mobile) != 10:

    return "Enter valid number"


else:

    #Register and save data to a file

    timestamp = str(datetime.datetime.now())

    folder_path = 'emplogin'

    folder_len = len(os.listdir(folder_path)) + 1

    previous_hash = get_previous_hash()

    block_data = {

        'blockid': folder_len,

        'email': email,

        'mobile': mobile,

        'password': password,

        'timestamp': timestamp,

        'prevhash': previous_hash
```

```

    }

    block_hash = hashlib.sha256(json.dumps(block_data,
sort_keys=True).encode()).hexdigest()

    block_data['hash'] = block_hash

    #Saving data with email as file name

    file_name = f"{folder_len}.json"

    file_path = os.path.join('employlogin', file_name)

    #Saving data to a file

    try:

        with open(file_path, 'w') as file:

            json.dump(block_data, file, indent=4)

            return "Registered successfully"

    except Exception as e:

        return f"An error occurred while saving the data: {str(e)}"

@app.route('/viewemp', methods=['GET'])

def viewemp():

    if request.method == 'GET':

        folder_path = 'employlogin'

        customer_data = []

        for file_name in os.listdir(folder_path):

            file_path = os.path.join(folder_path, file_name)

```

```

        with open(file_path, 'r') as file:

            data = json.load(file)

            customer_data.append({

                'blockid': data['blockid'],

                'email': data['email'],

                'mobile': data['mobile'],

                'password': data['password']

            })

    return render_template("data.html", cust = customer_data)


@app.route('/addprod')

def addprod():

    return render_template("addprod.html")


@app.route('/add_prod', methods=['POST'])

def add_prod():

    if request.method == 'POST':

        name = request.form.get('name')

        temprature = request.form.get('temprature')

        price = request.form.get('price')

        brand = request.form.get('brand')

        addby = request.form.get('addby')

```



```

if not all([name, temprature, price, brand]):

    return "All fields must be filled."

else:

    #Registration successful, save data to a file

    timestamp = str(datetime.datetime.now())

    folder_path = 'Add_Product'

    folder_len = len(os.listdir(folder_path)) + 1

    # Hash the current block's data

    block_data = {

        'blockid': folder_len,

        'name': name,

        'temprature': temprature,

        'price': price,

        'brand': brand,

        'addby': addby,

        'status': 'Not Mined'

    }

    #Creates a unique file name based on the block ID

    file_name = f"{folder_len}.json"

    file_path = os.path.join('Add_Product', file_name)

```

```

#Saving block data to a file

try:

    with open(file_path, 'w') as file:

        json.dump(block_data, file, indent=4)

    return "Product Added successfully"

except Exception as e:

    return f"An error occurred while saving the data: {str(e)}"


@app.route('/viewprod', methods=['GET'])

def viewprod():

    if request.method == 'GET':

        folder_path = 'Add_Product'

        product_data = []

        for file_name in os.listdir(folder_path):

            file_path = os.path.join(folder_path, file_name)

            with open(file_path, 'r') as file:

                data = json.load(file)

                product_data.append({

                    'blockid': data['blockid'],

                    'name': data['name'],

                    'temprature': data['temprature'],

                    'price': data['price'],

                    'brand': data['brand'],

                    'addby': data['addby'],

                    'status': data['status']

```

```
}}
```

```
    return render_template("product.html", prod = product_data)

@app.route('/mineprod', methods=['GET'])

def mine_block():

    folder_path = 'Add_Product'

    mined_folder_path = 'mined_product'

    #Creates a 'mined_product' folder if it doesn't exist

    if not os.path.exists(mined_folder_path):

        os.makedirs(mined_folder_path)

    last_mined_hash = get_last_mined_hash()

    for file_name in os.listdir(folder_path):

        file_path = os.path.join(folder_path, file_name)

        with open(file_path, 'r+') as file:

            data = json.load(file)

            # Only mine products with status 'Not Mined'

            if data.get('status') == 'Not Mined':

                # Calculate block hash

                block_data = {
```

```

        'blockid': data['blockid'],

        'name': data['name'],

        'temprature': data['temprature'],

        'price': data['price'],

        'brand': data['brand'],

        'addby': data['addby'],

        'status': 'Mined'

    }

    block_data['prevhash'] = last_mined_hash

    block_hash = hashlib.sha256(json.dumps(block_data,
sort_keys=True).encode()).hexdigest()

    block_data['hash'] = block_hash

    # Update file content with block hash, previous hash,
and status

    file.seek(0)

    json.dump(block_data, file, indent=4)

    file.truncate()

    #copy the updated file to the 'mined_product' folder

    shutil.copy(file_path, os.path.join(mined_folder_path,
file_name))

    #Update the last mined hash for the next block

```

```

        last_mined_hash = block_hash

    return redirect(url_for('viewprod'))

def get_last_mined_hash():

    folder_path = 'mined_product'

    files = os.listdir(folder_path)

    if files:

        last_file_path = os.path.join(folder_path, sorted(files)[-1])

        with open(last_file_path, 'r') as file:

            last_block_data = json.load(file)

            return last_block_data['hash'] if 'hash' in last_block_data
    else '0'

    else:

        return '0000000000000000'

@app.route('/globalprod', methods=['GET'])

def globalprod():

    if request.method == 'GET':

        folder_path = 'mined_product'

        product_data = []

        for file_name in os.listdir(folder_path):

            file_path = os.path.join(folder_path, file_name)

```

```
with open(file_path, 'r') as file:

    data = json.load(file)

    product_data.append({

        'blockid': data['blockid'],

        'name': data['name'],

        'temprature': data['temprature'],

        'price': data['price'],

        'brand': data['brand'],

        'addby': data['addby']

    })
```

```
return render_template("empaddprod.html", prod = product_data)
```

```
@app.route('/rawmat', methods=['GET'])
```

```
def rawmat():
```

```
    return render_template("rawmat.html")
```

```
@app.route('/manufac', methods=['GET'])
```

```
def manufac():
```

```
    return render_template("manufac.html")
```

```
@app.route('/supplier', methods=['GET'])
```

```
def supplier():
```

```
    return render_template("supplier.html")
```

```

@app.route('/retailer', methods=['GET'])

def retailer():

    return render_template("retailer.html")


#Tracker

def save_data(role, data, prodname):

    #Getting the list of files in the "tracker" folder

    files = os.listdir('tracker')

    #Checking if there are any files in the "tracker" folder

    if not files or role == 'raw supplier':

        # If no files or role is raw supplier, create a new file with
the name as the length of files

        file_index = len(files) + 1

        file_name = f'tracker_data_{file_index}.json'

    else:

        # Get the last file in the "tracker" folder

        sorted_files = sorted(files)

        last_file = sorted_files[-1]

        file_name = last_file

    # Defining the file path

    file_path = os.path.join('tracker', file_name)

```

```

#Saving the data to the file

with open(file_path, 'a') as file:

    file.write(json.dumps(data, indent=4))

    file.write('\n')


return f>Data saved to {file_name}."


@app.route('/tracker', methods=['POST'])

def tracker():

    role = request.form.get('role')

    if role == 'raw supplier':

        prodname = request.form.get('name')

    else:

        prodname = None


    role = request.form.get('role')

    status = request.form.get('status')

    temperature = request.form.get('temperature')


#Creating a dictionary of received data

data = {

    'role': role,

    'status': status,

    'temperature': temperature

```



```
}
```

```
#Saving data to a file
```

```
result = save_data(role, data,prodname)
```

```
return result
```

```
#Reports
```

```
@app.route('/report_page')
```

```
def report_page():
```

```
    return render_template("report.html")
```

```
def save_report(report):
```

```
    #Creates a "reports" if not exist.
```

```
    if not os.path.exists('reports'):
```

```
        os.makedirs('reports')
```

```
    #Getting the list of files in the "reports" folder
```

```
    files = os.listdir('reports')
```

```
    #file index based on the length of existing files
```

```
    file_index = len(files) + 1
```

```
    #file name for the new report file
```

```

file_name = f'reports_data_{file_index}.json'

file_path = os.path.join('reports', file_name)

#Saving the report data to the file

with open(file_path, 'w') as file:

    file.write(json.dumps(report, indent=4))


return f"Report saved to {file_name}."


@app.route('/report', methods=['POST'])

def report():

    if request.method == 'POST':

        files = os.listdir('reports')

        file_index = len(files) + 1

        prodname = request.form.get('name')

        report_text = request.form.get('report')


        data = {

            'srno': file_index,

            'reportby': 'Retailer',

            'prodname': prodname,

            'report': report_text

        }

        #Saving the report to a file

```

```

        result = save_report(data)

    return result

@app.route('/viewreport', methods=['GET'])
def viewreport():

    if request.method == 'GET':

        folder_path = 'reports'

        report_data = []

        for file_name in os.listdir(folder_path):

            file_path = os.path.join(folder_path, file_name)

            with open(file_path, 'r') as file:

                data = json.load(file)

                report_data.append({

                    'srno': data['srno'],

                    'reportby': data['reportby'],

                    'prodname': data['prodname'],

                    'report': data['report']

                })

        return render_template("viewrep.html", rep_data=report_data)

if __name__ == '__main__':

    app.run(debug=True)

```

Html Pages:

Supply.html:(mainpage)

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width,
initial-scale=1.0">

    <title></title>

    <style>

        body {

            display: flex;

            flex-direction: column;

            align-items: center;

            justify-content: center;

            min-height: 100vh;

            margin: 0;

            background-color: skyblue;

            font-family: Arial, sans-serif;

        }

        .container {

            display: flex;

            flex-direction: column;

            align-items: center;
```

```
    margin-top: 20px;

    background-color: white;

    padding: 20px;

    border-radius: 10px;
}

.container h1 {

    margin-bottom: 10px;

    text-align: center;
}

.container h2 {

    margin-bottom: 10px;

    text-align: center;
}

.container input {

    margin-bottom: 7px;

    width: 100%; /* Make the inputs stretch to full width */

    padding: 5px; /* Add some padding for better appearance */
}

.container input[type="button"] {

    background-color: #2980b9;

    color: white;
}
```

```

        margin: 0 10px; /* Adjust margin to create space between
buttons */

        padding: 10px 20px;

        border: none;

        border-radius: 5px;

        cursor: pointer;

        margin-bottom: 3px;

    }

    .container input[type="submit"] {

        background-color: #2980b9;

        color: white;

        margin: 0 10px; /* Adjust margin to create space between
buttons */

        padding: 10px 20px;

        border: none;

        border-radius: 5px;

        cursor: pointer;

        margin-bottom: 3px;

    }

</style>

</head>

<body>

    <h1>BLOCKCHAIN IN SUPPLYCHAIN MANAGEMENT - EMPLOYEE LOGIN</h1>

    <form class="container" action="/employee" method="POST">

```

```
<div class="container">

    <label>Email</label>

    <input type="text" placeholder="Email" name="email"/><br>

    <label>Password</label>

    <input type="password" placeholder="Password"
name="password"/><br><br>

    <input type="submit" value="Login"/>

    <input type="button" value="Admin Login"
onclick="adminLogin()" /><br>

</div>

<script>

    function adminLogin() {

        window.location.href = "/adminlogin";

    }

</script>

</form>

</body>

</html>
```

Addemp.html:

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width,  
initial-scale=1.0">
```

```
<title>Registration</title>
```

```
<style>
```

```
    body {  
  
        display: flex;  
  
        flex-direction: column;  
  
        align-items: center;  
  
        justify-content: center;  
  
        min-height: 100vh;  
  
        margin: 0;  
  
        background-color: skyblue;  
  
        font-family: Arial, sans-serif;  
  
    }
```

```
.container {  
  
    display: flex;  
  
    flex-direction: column;  
  
    align-items: center;  
  
    justify-content: center;  
  
    background-color: white;  
  
    padding: 20px;  
  
    border-radius: 10px;  
  
}
```



```
.container h1 {  
  
    text-align: center;  
  
}
```

```
.container div {  
  
    display: flex;  
  
    flex-direction: column;  
  
    align-items: center;  
  
    margin-bottom: 10px;  
  
}
```

```
.container div label {  
  
    margin-bottom: 5px;  
  
}
```

```
.container div input {  
  
    width: 100%;  
  
    padding: 5px;  
  
}
```

```
.container button {  
  
    background-color: #2980b9;  
  
    color: white;  
  
    padding: 10px 20px;  
  
    border: none;
```

```
        border-radius: 5px;

        cursor: pointer;
    }

</style>

</head>

<body>

    <form class="container" action="/reg" method="POST">

        <h1>BLOCKCHAIN IN SUPPLYCHAIN MANAGEMENT - REGISTRATION</h1>

        <div>

            <label for="email">Email:</label>

            <input type="text" id="email" name="email"
placeholder="Email">

        </div>

        <div>

            <label for="mobile">Mobile:</label>

            <input type="text" id="mobile" name="mobile"
placeholder="Mobile">

        </div>

        <div>

            <label for="password">Password:</label>

            <input type="password" id="password" name="password"
placeholder="Password">

        </div>

        <div>

            <label for="repassword">Re-enter Password:</label>

            <input type="password" id="repassword" name="repassword"
placeholder="Re-enter Password">

    </form>

</body>

</html>
```

```
        </div>

        <button type="submit">Register</button>

    </form>

</body>

</html>
```

Addprod.html:

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width,
initial-scale=1.0">

    <title>Registration</title>

    <style>

        body {

            display: flex;

            flex-direction: column;

            align-items: center;

            justify-content: center;

            min-height: 100vh;

            margin: 0;

            background-color: skyblue;

            font-family: Arial, sans-serif;
```

```
}
```

```
.container {  
  
    display: flex;  
  
    flex-direction: column;  
  
    align-items: center;  
  
    justify-content: center;  
  
    background-color: white;  
  
    padding: 20px;  
  
    border-radius: 10px;  
  
}
```

```
.container h1 {  
  
    text-align: center;  
  
}
```

```
.container div {  
  
    display: flex;  
  
    flex-direction: column;  
  
    align-items: center;  
  
    margin-bottom: 10px;  
  
}
```

```
.container div label {  
  
    margin-bottom: 5px;
```

```

    }

    .container div input {

        width: 100%;

        padding: 5px;

    }

    .container button {

        background-color: #2980b9;

        color: white;

        padding: 10px 20px;

        border: none;

        border-radius: 5px;

        cursor: pointer;

    }

</style>

</head>

<body>

    <form class="container" action="/add_prod" method="POST">

        <h1>BLOCKCHAIN IN SUPPLYCHAIN MANAGEMENT - ADD PRODUCT</h1>

        <div>

            <label for="name">Product Name:</label>

            <input type="text" id="name" name="name"
placeholder="Product Name">

        </div>

```

```
<div>

    <label for="temprature">Temprature:</label>

    <input type="text" id="temprature" name="temprature"
placeholder="Temprature">

</div>

<div>

    <label for="price">Price:</label>

    <input type="text" id="price" name="price"
placeholder="Price">

</div>

<div>

    <label for="brand">Brand:</label>

    <input type="text" id="brand" name="brand"
placeholder="Brand">

</div>

<div>

    <label for="addby">Added By:</label>

    <input type="text" id="addby" name="addby"
placeholder="Added By">

</div>

<button type="submit">Register</button>

</form>

</body>

</html>
```

Admin.html:

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width,
initial-scale=1.0">

  <title>Document</title>

  <style>

    body {

      display: flex;

      flex-direction: column;

      align-items: center;

      justify-content: center;

      min-height: 100vh;

      margin: 0;

      background-color: skyblue;

      font-family: Arial, sans-serif;

    }

    .container {

      display: flex;

      align-items: center;

      flex-direction: column;

      justify-content: center;
```

```
}

.dropdown {

    margin-right: 20px; /* Adjust margin as needed */

}

.container input[type="button"] {

    background-color: #2980b9;

    color: white;

    margin: 0 10px; /* Adjust margin to create space between
buttons */

    padding: 10px 20px;

    border: none;

    border-radius: 5px;

    cursor: pointer;

    margin-bottom: 3px;

}

</style>

</head>

<body>

    <div>

        <h1>WELCOME ADMIN</h1>

    </div>
```



```

<div class="container">

    <div style="display: flex;flex-direction: row;">

        <div class="dropdown" >

            <input type="button" value="Add Product"
onclick="add_prod()" ">

            </select>

        </div>

        <div class="dropdown">

            <input type="button" value="View Product"
onclick="view_prod()" ">

        </div>

        <div class="dropdown">

            <input type="button" value="Add Employee"
onclick="add_emp()" ">

        </div>

        <div class="dropdown">

            <input type="button" value="View Employee"
onclick="view_emp()" ">

        </div>

        <div class="dropdown">

            <input type="button" value="Reports"
onclick="view_report()" ">

        </div>

    </div>

</div>

<script>

    function add_emp() {

```

```
        window.location.href = "/addemp";

    }

    function view_emp() {

        window.location.href = "/viewemp";

    }

    function add_prod() {

        window.location.href = "/addprod";

    }

    function view_prod() {

        window.location.href = "/viewprod";

    }

    function view_report() {

        window.location.href = "/viewreport";

    }

</script>

</body>

</html>
```

Adminlogin.html:

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width,
initial-scale=1.0">

    <title></title>
```

```
<style>
```

```
body {  
  
    display: flex;  
  
    flex-direction: column;  
  
    align-items: center;  
  
    justify-content: center;  
  
    min-height: 100vh;  
  
    margin: 0;  
  
    background-color: skyblue;  
  
    font-family: Arial, sans-serif;  
  
}
```

```
.container {  
  
    display: flex;  
  
    flex-direction: column;  
  
    align-items: center;  
  
    margin-top: 20px;  
  
    background-color: white;  
  
    padding: 20px;  
  
    border-radius: 10px;  
  
}
```

```
.container h1 {  
  
    margin-bottom: 10px;  
  
    text-align: center;  
  
}
```

```
}
```

```
.container h2 {  
  
    margin-bottom: 10px;  
  
    text-align: center;  
  
}
```

```
.container input {  
  
    margin-bottom: 7px;  
  
    width: 100%; /* Make the inputs stretch to full width */  
  
    padding: 5px; /* Add some padding for better appearance */  
  
}
```

```
.container input[type="submit"] {  
  
    background-color: #2980b9;  
  
    color: white;  
  
    margin: 0 10px; /* Adjust margin to create space between  
buttons */  
  
    padding: 10px 20px;  
  
    border: none;  
  
    border-radius: 5px;  
  
    cursor: pointer;  
  
    margin-bottom: 3px;  
  
}
```

```
</style>

</head>

<body>

    <h1>BLOCKCHAIN IN SUPPLYCHAIN MANAGEMENT - ADMIN LOGIN</h1>

    <form class="container" action="/admin" method="POST">

        <div class="container">

            <label>Email</label>

            <input type="text" placeholder="Email" name="email"/><br>

            <label>Password</label>

            <input type="password" placeholder="Password"
name="password"/><br><br>

            <input type="submit" value="Admin Login"/><br>

        </div>

    </form>

</body>

</html>
```

Data.html:

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
```

```
<title>Document</title>
```

```
<style>
```

```
    body {  
  
        text-align: center;  
  
        background-color: skyblue;  
  
    }
```

```
    .container {  
  
        display: flex;  
  
        flex-direction: column;  
  
        justify-content: center;  
  
        background-color: white;  
  
        padding: 20px;  
  
        border-radius: 30px;  
  
        margin-left: 30%;  
  
        margin-right: 30%;  
  
    }
```

```
    table {  
  
        display: flex;  
  
        align-items: center;  
  
        flex-direction: column;  
  
        justify-content: center;  
  
        margin-top: 10px;  
  
    }
```

```
</style>

</head>

<body>

<h1>EMPLOYEES IN BLOCKCHAIN</h1>

<div class="container">

    <table>

        <tr>

            <th>BlockId</th>

            <th>Email</th>

            <th>Mobile</th>

            <th>Password</th>

        </tr>

        {% for data in cust %}

        <tr>

            <td>{{ data['blockid'] }}</td>

            <td>{{ data['email'] }}</td>

            <td>{{ data['mobile'] }}</td>

            <td>{{ data['password'] }}</td>

        </tr>

        {% endfor %}

    </table>
```

```
        </div>

</body>

</html>
```

Empaddprod.html:

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width,
initial-scale=1.0">

    <title>Document</title>

    <style>

        body {

            text-align: center;

            background-color: skyblue;

        }

        .container {

            display: flex;

            flex-direction: column;

            justify-content: center;

            background-color: white;

            padding: 20px;

            border-radius: 30px;

            margin-left: 30%;
```



```
        margin-right: 30%;
    }

    table {

        display: flex;

        align-items: center;

        flex-direction: column;

        justify-content: center;

        margin-top: 10px;
    }

.container input[type="submit"] {

    background-color: #2980b9;

    color: white;

    margin: 0 10px; /* Adjust margin to create space between
buttons */

    padding: 10px 20px;

    border: none;

    border-radius: 5px;

    cursor: pointer;

    margin-bottom: 3px;
}

</style>

</head>
```

```
<body>

<h1>PRODUCTS IN BLOCKCHAIN</h1>

<form class="container" action="/globalprod" method="GET">

  <div class="container">

    <table>

      <tr>

        <th>BlockId</th>

        <th>Name</th>

        <th>Temprature</th>

        <th>Price</th>

        <th>Brand</th>

      </tr>

      {% for data in prod %}

      <tr>

        <td>{{ data['blockid'] }}</td>

        <td>{{ data['name'] }}</td>

        <td>{{ data['temprature'] }}</td>

        <td>{{ data['price'] }}</td>

        <td>{{ data['brand'] }}</td>

      </tr>

      {% endfor %}

    </table>
```

```
        </div>

    </form>

</body>

</html>
```

Employee.html:

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width,
initial-scale=1.0">

    <title>Document</title>

    <style>

        body {

            display: flex;

            flex-direction: column;

            align-items: center;

            justify-content: center;

            min-height: 100vh;

            margin: 0;

            background-color: skyblue;

        }

        .container {
```

```
        display: flex;

        align-items: center;

        justify-content: center;

        margin-top: 50px;
    }

    .dropdown {

        margin-right: 20px; /* Adjust margin as needed */
    }

    .container input[type="button"] {

        background-color: #2980b9;

        color: white;

        margin: 0 10px; /* Adjust margin to create space between
buttons */

        padding: 10px 20px;

        border: none;

        border-radius: 5px;

        cursor: pointer;

        margin-bottom: 3px;
    }

</style>

</head>

<body>
```

```
<div>

    <h1>WELCOME EMPLOYEE</h1>

</div>

<div class="container">

    <div class="dropdown">

        <input type="button" value="Add product"
onclick="add_prod()" ">

    </div>

    <div class="dropdown">

        <input type="button" value="View product"
onclick="view_prod()" ">

    </div>

    <div class="dropdown">

        <input type="button" value="Reports"
onclick="view_report()" ">

    </div>

</div>

<script>

    function add_prod() {

        window.location.href = "/addprod";

    }

    function view_prod() {

        window.location.href = "/globalprod";

    }

    function view_report() {

        window.location.href = "/viewreport";

    }

}
```

```
    }

    </script>

</body>

</html>
```

Manufac.html:

```
<!DOCTYPE html>

<html>

<head>

    <title>Manufacturer</title>

    <style>

        body {

            display: flex;

            flex-direction: column;

            align-items: center;

            justify-content: center;

            min-height: 100vh;

            margin: 0;

            background-color: skyblue;

            font-family: Arial, sans-serif;

        }

        .container {

            display: flex;

            flex-direction: column;
```

```
    align-items: center;

    margin-top: 20px;

    background-color: white;

    padding: 20px;

    border-radius: 10px;
}

.container input {

    margin-bottom: 7px;

    width: 100%; /* Make the inputs stretch to full width */

    padding: 5px; /* Add some padding for better appearance */
}

.container button[type="submit"] {

    background-color: #2980b9;

    color: white;

    margin: 0 10px; /* Adjust margin to create space between
buttons */

    padding: 10px 20px;

    border: none;

    border-radius: 5px;

    cursor: pointer;

    margin-bottom: 3px;
}
```

```

        .container button[type="button"] {

            background-color: #2980b9;

            color: white;

            margin: 0 10px; /* Adjust margin to create space between
buttons */

            padding: 10px 20px;

            border: none;

            border-radius: 5px;

            cursor: pointer;

            margin-bottom: 3px;

        }

    </style>

</head>

<body>

    <h1>Manufacturer</h1>

    <form class="container" action="/tracker" method="POST">

        <label for="status">Enter Role: </label>

        <input type="text" id="role" name="role" required><br><br>

        <input type="hidden" name="role" value="manufacturer">

        <label for="temperature">Enter Temperature: </label>

        <input type="text" id="temperature" name="temperature"
required><br><br>

        <label for="status">Enter Status (sent): </label>

        <input type="text" id="status" name="status" required><br><br>

        <button type="submit">Submit</button>

```



```
        <button type="button" onclick="view_prod()">View
Products</button>

</form>

<script>

    function view_prod() {

        window.location.href = "/globalprod";

    }

</script>

</body>

</html>
```

Product.html:

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width,
initial-scale=1.0">

    <title>Document</title>

    <style>

        body {

            text-align: center;

            background-color: skyblue;

        }

        .container {
```

```
display: flex;

flex-direction: column;

justify-content: center;

background-color: white;

padding: 20px;

border-radius: 30px;

margin-left: 30%;

margin-right: 30%;

}
```

```
table {

display: flex;

align-items: center;

flex-direction: column;

justify-content: center;

margin-top: 10px;

}
```

```
.container input[type="submit"] {

background-color: #2980b9;

color: white;

margin: 0 10px; /* Adjust margin to create space between
buttons */

padding: 10px 20px;

border: none;
```

```
        border-radius: 5px;

        cursor: pointer;

        margin-bottom: 3px;

    }

</style>

</head>

<body>

    <h1>PRODUCTS IN BLOCKCHAIN</h1>

    <form class="container" action="/mineprod" method="GET">

        <div class="container">

            <table>

                <tr>

                    <th>BlockId</th>

                    <th>Name</th>

                    <th>Temprature</th>

                    <th>Price</th>

                    <th>Brand</th>

                    <th>Added By</th>

                    <th>Status</th>

                </tr>

                {% for data in prod %}

                <tr>
```

```

        <td>{{ data['blockid'] }}</td>

        <td>{{ data['name'] }}</td>

        <td>{{ data['temprature'] }}</td>

        <td>{{ data['price'] }}</td>

        <td>{{ data['brand'] }}</td>

        <td>{{ data['addby'] }}</td>

        <td>{{ data['status'] }}</td>

    </tr>

    {% endfor %}

</table>

<input type="submit" value="Mine Product"></button>

</div>

</form>

</body>

</html>

```

Rawmat.html:

```

<!DOCTYPE html>

<html>

<head>

    <title>Raw Material Supplier</title>

    <style>

        body {

            display: flex;

```

```
    flex-direction: column;

    align-items: center;

    justify-content: center;

    min-height: 100vh;

    margin: 0;

    background-color: skyblue;

    font-family: Arial, sans-serif;
}
```

```
.container {

    display: flex;

    flex-direction: column;

    align-items: center;

    margin-top: 20px;

    background-color: white;

    padding: 20px;

    border-radius: 10px;
}
```

```
.container input {

    margin-bottom: 7px;

    width: 100%; /* Make the inputs stretch to full width */

    padding: 5px; /* Add some padding for better appearance */
}
```

```
.container button[type="submit"] {

    background-color: #2980b9;

    color: white;

    margin: 0 10px; /* Adjust margin to create space between
buttons */

    padding: 10px 20px;

    border: none;

    border-radius: 5px;

    cursor: pointer;

    margin-bottom: 3px;

}

.container button[type="button"] {

    background-color: #2980b9;

    color: white;

    margin: 0 10px; /* Adjust margin to create space between
buttons */

    padding: 10px 20px;

    border: none;

    border-radius: 5px;

    cursor: pointer;

    margin-bottom: 3px;

}

</style>

</head>

<body>

    <h1>Raw Material Supplier</h1>
```

```

<form class="container" action="/tracker" method="POST">

    <label for="status">Product name:</label>

    <input type="text" id="name" name="name" required><br><br>

    <label for="status">Enter Role: </label>

    <input type="text" id="role" name="role" required><br><br>

    <input type="hidden" name="role" value="manufacturer">

    <label for="temperature">Enter Temperature: </label>

    <input type="text" id="temperature" name="temperature"
required><br><br>

    <label for="status">Enter Status (sent): </label>

    <input type="text" id="status" name="status" required><br><br>

    <button type="submit">Submit</button>

    <button type="button" onclick="view_prod()">View
Products</button>

</form>

<script>

    function view_prod() {

        window.location.href = "/globalprod";

    }

</script>

</body>

</html>

```

Report.html:

```

<!DOCTYPE html>

<html>

```

```
<head>
```

```
<style>
```

```
body {  
  
    display: flex;  
  
    flex-direction: column;  
  
    align-items: center;  
  
    justify-content: center;  
  
    margin: 0;  
  
    background-color: skyblue;  
  
    font-family: Arial, sans-serif;  
  
}
```

```
.container {  
  
    display: flex;  
  
    flex-direction: column;  
  
    align-items: center;  
  
    background-color: white;  
  
    padding: 20px;  
  
    border-radius: 10px;  
  
}
```

```
.container input {  
  
    width: 100%; /* Make the inputs stretch to full width */  
  
    padding: 5px; /* Add some padding for better appearance */  
  
}
```



```
}
```

```
.container button[type="submit"] {
```

```
    background-color: #2980b9;
```

```
    color: white;
```

```
    padding: 10px 20px;
```

```
    border: none;
```

```
    border-radius: 5px;
```

```
    cursor: pointer;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>Retailer</h1>
```

```
<form class="container" action="/report" method="POST">
```

```
    <h1>Report Page</h1>
```

```
    <label>Product name</label>
```

```
    <input type="text" id="name" name="name" required><br><br>
```

```
    <label>Report</label>
```

```
    <input type="text" id="report" name="report" required><br><br>
```

```
    <button type="submit">Submit</button>
```

```
</form>
```

```
</body>
```

```
</html>
```

Retailer.html:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <title>Retailer</title>
```

```
  <style>
```

```
    body {
```

```
      display: flex;
```

```
      flex-direction: column;
```

```
      align-items: center;
```

```
      justify-content: center;
```

```
      margin: 0;
```

```
      background-color: skyblue;
```

```
      font-family: Arial, sans-serif;
```

```
    }
```

```
    .container {
```

```
      display: flexbox;
```

```
      flex-direction: column;
```

```
      align-items: center;
```

```
      background-color: white;
```

```
      padding: 20px;
```

```
      border-radius: 10px;
```

```
}

.container input {

    width: 95%; /* Make the inputs stretch to full width */

    padding: 5px; /* Add some padding for better appearance */

}

.container button[type="submit"] {

    background-color: #2980b9;

    color: white;

    padding: 10px 20px;

    border: none;

    border-radius: 5px;

    cursor: pointer;

}

.container button[type="button"] {

    background-color: #2980b9;

    color: white;

    padding: 10px 20px;

    border: none;

    margin-top: 5px;

    border-radius: 5px;

    cursor: pointer;

}

</style>
```

```
</head>

<body>

    <h1>Retailer</h1>

    <form class="container" action="/tracker" method="POST">

        <label for="status">Enter Role: </label>

        <input type="text" id="role" name="role" required><br><br>

        <input type="hidden" name="role" value="manufacturer">

        <label for="temperature">Enter Temperature: </label>

        <input type="text" id="temperature" name="temperature"
required><br><br>

        <label for="status">Enter Status (received): </label>

        <input type="text" id="status" name="status" required><br><br>

        <button type="submit">Submit</button>

        <button type="button" onclick="view_prod()">View
Products</button>

        <h1>Report</h1>

        <button type="button" onclick="reppage()">Report</button>

    </form>

    <script>

        function view_prod() {

            window.location.href = "/globalprod";

        }

        function reppage() {

            window.location.href = "/report_page";

        }

    </script>
```

```
</body>
```

```
</html>
```

Supplier.html:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <title>Supplier</title>
```

```
  <style>
```

```
    body {
```

```
      display: flex;
```

```
      flex-direction: column;
```

```
      align-items: center;
```

```
      justify-content: center;
```

```
      min-height: 100vh;
```

```
      margin: 0;
```

```
      background-color: skyblue;
```

```
      font-family: Arial, sans-serif;
```

```
    }
```

```
    .container {
```

```
      display: flex;
```

```
      flex-direction: column;
```

```
      align-items: center;
```

```
      margin-top: 20px;
```

```
      background-color: white;
```

```
padding: 20px;

border-radius: 10px;

}

.container input {

margin-bottom: 7px;

width: 100%; /* Make the inputs stretch to full width */

padding: 5px; /* Add some padding for better appearance */

}

.container button[type="submit"] {

background-color: #2980b9;

color: white;

margin: 0 10px; /* Adjust margin to create space between
buttons */

padding: 10px 20px;

border: none;

border-radius: 5px;

cursor: pointer;

margin-bottom: 3px;

}

.container button[type="button"] {

background-color: #2980b9;

color: white;
```

```

        margin: 0 10px; /* Adjust margin to create space between
buttons */

        padding: 10px 20px;

        border: none;

        border-radius: 5px;

        cursor: pointer;

        margin-bottom: 3px;

    }

</style>

</head>

<body>

    <h1>Supplier</h1>

    <form class="container" action="/tracker" method="POST">

        <label for="status">Enter Role: </label>

        <input type="text" id="role" name="role" required><br><br>

        <input type="hidden" name="role" value="manufacturer">

        <label for="temperature">Enter Temperature: </label>

        <input type="text" id="temperature" name="temperature"
required><br><br>

        <label for="status">Enter Status (sent): </label>

        <input type="text" id="status" name="status" required><br><br>

        <button type="submit">Submit</button>

        <button type="button" onclick="view_prod()">View
Products</button>

    </form>

    <script>

```

```
function view_prod() {  
  
    window.location.href = "/globalprod";  
  
}  
  
</script>  
  
</body>  
  
</html>
```

Viewrep.html:

```
<!DOCTYPE html>  
  
<html lang="en">  
  
<head>  
  
    <meta charset="UTF-8">  
  
    <meta name="viewport" content="width=device-width,  
initial-scale=1.0">  
  
    <title>View Reports</title>  
  
    <style>  
  
        body {  
  
            text-align: center;  
  
            background-color: skyblue;  
  
        }  
  
  
        .container {  
  
            display: flex;  
  
            flex-direction: column;  
  
            justify-content: center;
```



```

        background-color: white;

        padding: 20px;

        border-radius: 30px;

        margin-left: 30%;

        margin-right: 30%;
    }

    table {

        display: flex;

        align-items: center;

        flex-direction: column;

        justify-content: center;

        margin-top: 10px;
    }

    .container input[type="submit"] {

        background-color: #2980b9;

        color: white;

        margin: 0 10px; /* Adjust margin to create space between
buttons */

        padding: 10px 20px;

        border: none;

        border-radius: 5px;

        cursor: pointer;

        margin-bottom: 3px;

```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>Reports</h1>
```

```
<form class="container">
```

```
<div class="container">
```

```
<table>
```

```
<tr>
```

```
<th>SrNo</th>
```

```
<th>ReportBy</th>
```

```
<th>Product Name</th>
```

```
<th>Report</th>
```

```
</tr>
```

```
{% for data in rep_data %}
```

```
<tr>
```

```
<td>{{ data['srno'] }}</td>
```

```
<td>{{ data['reportby'] }}</td>
```

```
<td>{{ data['prodname'] }}</td>
```

```
<td>{{ data['report'] }}</td>
```

```
</tr>
```

```
{% endfor %}
```

```
</table>
```

```
</div>
```

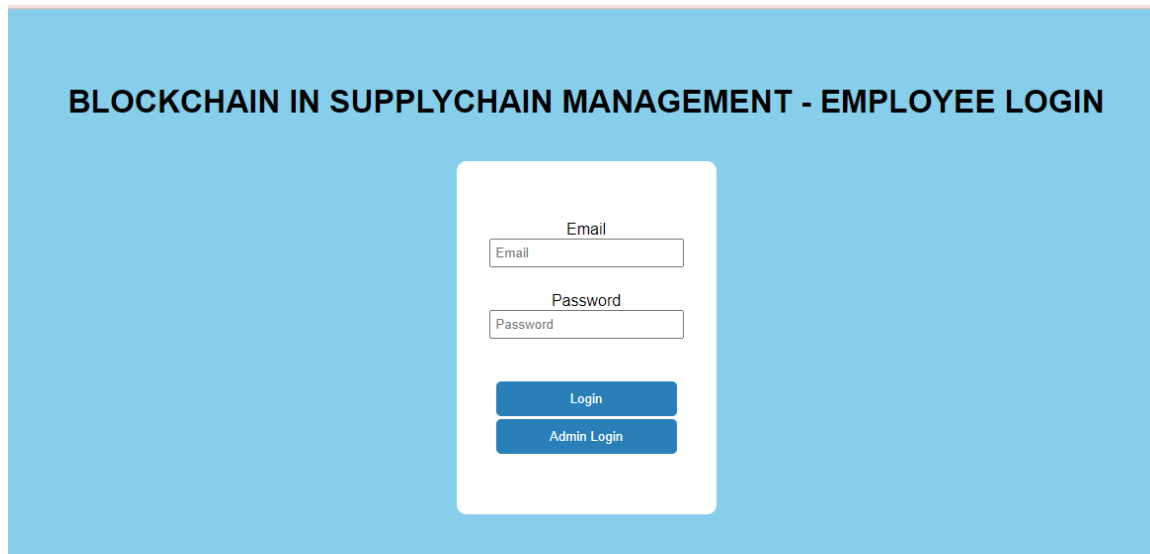
```
</form>

</body>

</html>
```

Output:

1)Mainpage:



BLOCKCHAIN IN SUPPLYCHAIN MANAGEMENT - EMPLOYEE LOGIN

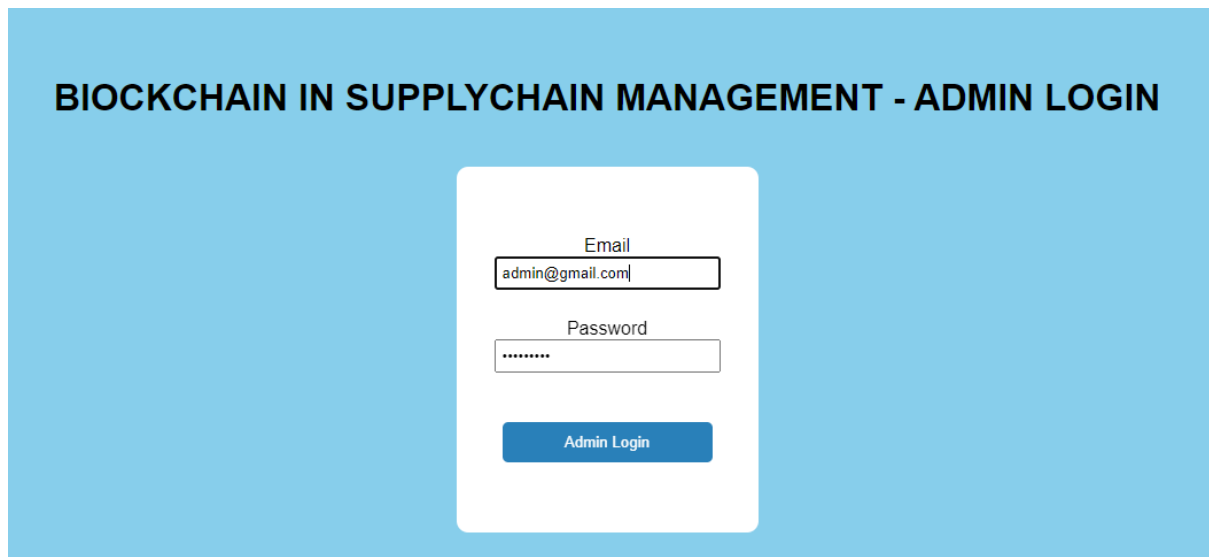
Email

Password

Login

Admin Login

2)Admin login:(if email and password are not correct)



BLOCKCHAIN IN SUPPLYCHAIN MANAGEMENT - ADMIN LOGIN

Email

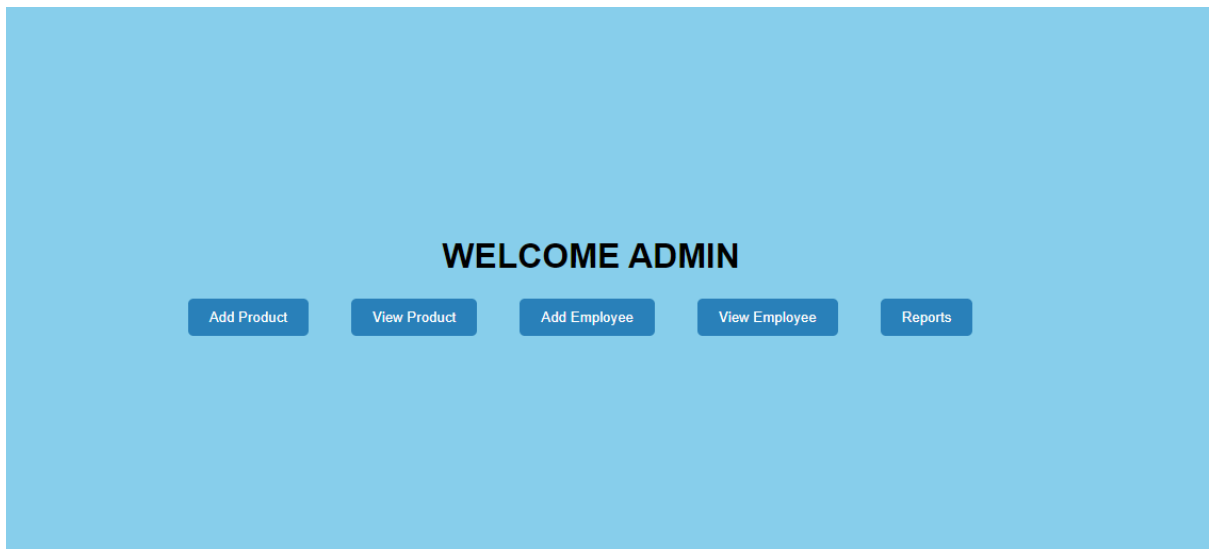
Password

Admin Login

Message:

Admin login failed. Please check your email and password.

Else:(LogIn)



3)Firstly we will be adding Employees so that we will not only be the one to add product

If any field is empty:

A screenshot of a registration form titled "BLOCKCHAIN IN SUPPLYCHAIN MANAGEMENT - REGISTRATION" in bold black font. The form is set against a light blue background. It contains five input fields: "Email:", "Mobile:", "Password:", and "Re-enter Password:". The "Mobile:" field contains the number "87965784". The "Password:" and "Re-enter Password:" fields are filled with asterisks. Below the input fields is a blue "Register" button.

Message:

All fields must be filled.

If password and re-enter password are not same

Passwords do not match.

If number is less than 10 digit:

Enter valid number

Else:(register and store data in file and save in folder emplogin)

Registered successfully

```
emplogin
{ } 1.json
{ } 2.json
```

4)Once employees are added we can check it in view employee:

EMPLOYEES IN BLOCKCHAIN

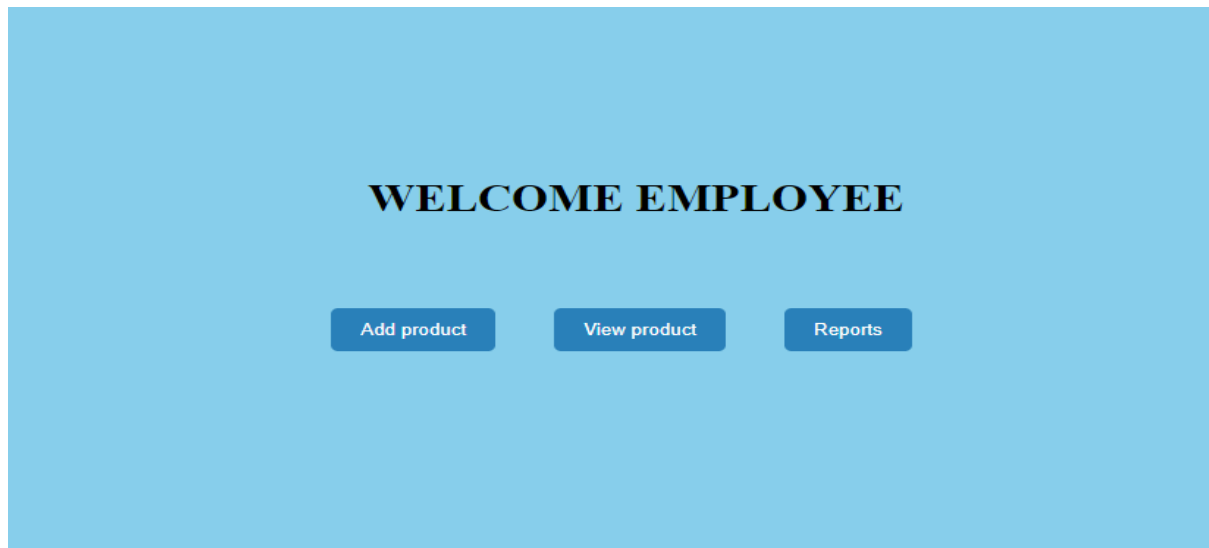
BlockId	Email	Mobile	Password
1	mangesh@gmail.com	8796578422	mangesh
2	pritam@gmail.com	9786578422	pritam

5)Logging a employee:

If email or password is wrong:

Login failed. Please check your email and password.

Else:



6)Adding product from admin and employee which will get stored in Add_Product folder:

From admin:

BLOCKCHAIN IN SUPPLYCHAIN MANAGEMENT - ADD PRODUCT

Product Name:

Icecream

Temperature:

18

Price:

100

Brand:

Amul

Added By:

Admin

Register

Message:

Product Added successfully

From employee:

BLOCKCHAIN IN SUPPLYCHAIN MANAGEMENT - ADD PRODUCT

Product Name:

Icecream2

Temprature:

18

Price:

200

Brand:

Amul

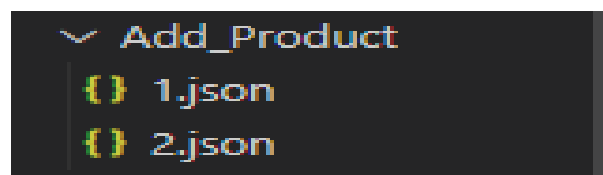
Added By:

Employee

Register

Message:

Product Added successfully



7)Once product is added it will be visible to admin with status not mined and who added it and will not be visible to employee or others:

For Admin:

PRODUCTS IN BLOCKCHAIN						
BlockId	Name	Temprature	Price	Brand	Added By	Status
1	Icecream	18	100	Amul	Admin	Not Mined
2	Icecream2	18	200	Amul	Employee	Not Mined
Mine Product						

For Employee and other:

PRODUCTS IN BLOCKCHAIN				
BlockId Name Temprature Price Brand				

8)When the admin click on mine all the products with ‘Not Mined’ status get mined and status changes to ‘Mined’ for admin and store file in mined_product folder,get added in blockchain.Now the products details will be visible to everyone:

PRODUCTS IN BLOCKCHAIN						
BlockId	Name	Temprature	Price	Brand	Added By	Status
1	Icecream	18	100	Amul	Admin	Mined
2	Icecream2	18	200	Amul	Employee	Mined
<button>Mine Product</button>						

mined_product	
{ }	1.json
{ }	2.json

For employee and everyone:

PRODUCTS IN BLOCKCHAIN					
BlockId	Name	Temprature	Price	Brand	
1	Icecream	18	100	Amul	
2	Icecream2	18	200	Amul	

9) Now product will be visible to raw supplier, manufacturer, supplier and retailer, now (raw supplier, manufacturer, supplier) can start process to deliver product to retailer in this retailer is the customer. Everyone's data will get store in tracker folder:

Raw Material Supplier

Product name:

Icecream

Enter Role:

aw supplier

Enter Temperature:

18

Enter Status (sent):

Material sent

Submit

View Products

Manufacturer

Enter Role:

Manufacturer

Enter Temperature:

18

Enter Status (sent):

Material sent

Submit

View Products

Supplier

Enter Role:

Supplier

Enter Temperature:

26

Enter Status (sent):

Material sent

Submit

View Products

Retailer

Enter Role:

Retailer

Enter Temperature:

32

Enter Status (received):

Received

Submit

View Products

Report

Report

10) If the product condition is not good, retailers can report (which will get stored in reports folder). In our case it is not good

Retailer

Report Page

Product name

Report

Submit

```
▼ reports
| {} reports_data_1.json
```

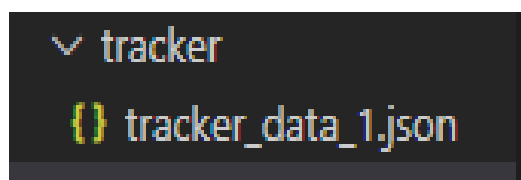
11)Once the retailer reports it gets displayed to the employee who will contact the admin to track who made the mistake.So the status and temperature of raw supplier, manufacturer,supplier and retailer are stored in a file(different for each product) in the tracker folder.

For employee:

Reports			
SrNo	ReportBy	Product Name	Report
1	Retailer	Icecream	Icecream Melted

For admin:

Reports			
SrNo	ReportBy	Product Name	Report
1	Retailer	Icecream	Icecream Melted



Admin can open the file and check who made mistake:

```
{
  "role": "raw supplier",
  "status": "Material sent",
  "temperature": "18"
}
{
  "role": "Manufacturer",
  "status": "Material sent",
  "temperature": "18"
}
{
  "role": "Supplier",
  "status": "Material sent",
  "temperature": "26"
}
{
  "role": "Retailer",
  "status": "Received",
  "temperature": "32"
}
```

Conclusion:In our case the mistake was made by the manufacturer while sending the product to the supplier, the temperature is 26 when the supplier received it as it must be 18.

