

HEALTH TRACKING SYSTEM

[Project report]



SUBMITTED FOR PARTIAL FULFILMENT OF
MASTER OF COMPUTER APPLICATIONS

- | | |
|--------------------------|--------------|
| • Ashutosh Kumar Yadav | 504121011006 |
| • Mujahid Ali Ansari | 504121011025 |
| • Shahil Kumar Chourasia | 504121011045 |
| • Sirsha Majumder | 504121021050 |
| • Sonu Routh | 504121011051 |

UNDER THE GUIDANCE OF **MRS.DOLA SAHA**

(ASSISTANT PROFESSOR OF GURUNANAK INSTITUTE OF TECHNOLOGY)

DEPARTMENT OF COMPUTER APPLICATIONS
GURUNANAK INSTITUTE OF TECHNOLOGY

GNIT

TABLE OF CONTENTS

1.	Certificate of Approval -----	01
2.	Acknowledgement -----	02
3.	Abstract -----	03
4.	Introduction -----	04
5.	Objective -----	05
6.	Flow chart Diagram of Health Tracking System -----	06
7.	Tools & Software -----	07
8.	Approach -----	08 - 21
9.	Outputs -----	22 - 23
10.	Applications -----	24
11.	Advantages -----	25
12.	Disadvantages -----	26
13.	Conclusion -----	27
14.	Future Development -----	28
15.	Bibliography -----	29
16.	Individual Contribution to the project -----	30

 **CLOUD
COMPUTING IN
HEALTHCARE**



CERTIFICATE OF APPROVAL

This is to certify that this report of our MCA Final Year Project entitled "Health Tracking System", carried out by -

- | | |
|--------------------------|--------------|
| • Ashutosh Kumar Yadav | 504121011006 |
| • Mujahid Ali Ansari | 504121011025 |
| • Shahil Kumar Chourasia | 504121011045 |
| • Sirsha Majumder | 504121021050 |
| • Sonu Routh | 504121011051 |

Under my supervision and guidance.

In my opinion, the report in its present form is in partial fulfilment of all the requirements, as specified by GuruNanak Institute of Technology (JISGroup) and as per the regulations of Maulana Abul Kalam Azad University of Technology. In fact, it has attained the standard and the necessary criteria for submission to the best of my knowledge. The result embodied in this report is original in nature and worthy of incorporation in the present version of the report of our MCA program.

X

Prof. Dola Saha
Project Mentor

X

Dr. Jib Dutta
HEAD, DEPARTMENT

X

Dr. Santanu Kumar Sen
Principal of GNIT



ACKNOWLEDGEMENT

We take this opportunity to express our profound gratitude and deep regards to our guide Mrs. Dola Saha for her exemplary guidance, monitoring, and constant encouragement throughout this course. The blessings, help and guidance given by her to us shall carry us to a long way in our journeys of life on which we are about to embark.

Signature of Students

Signature of Mentor



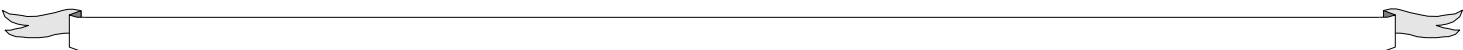


ABSTRACT

This project introduces a Cloud Based Health Tracking System that can track the records of our daily Health measures like our Fitness Plans and Diet Plans 24x7 suggested by our Nutritionist or Dietician. This application consists of several the parts based on - Python and AWS Cloud.

Data storing part is done through AWS Cloud, and Python has been used as a base language. We are monitoring and running the entire application through AWS Cloud. Through this application, our Nutritionist or Dietician can log onto our daily Fitness records and Diet Plans. We can also retrieve our health records at any point of time by using our User ID. Deleting the records is also a possible function here.

The most important part is that, with the help of this project, our Nutritionist or Dietician can maintain every Patient's data in a separate folder so that they can access and retrieve the records easily at any point of time with the help of the Patient's User ID.



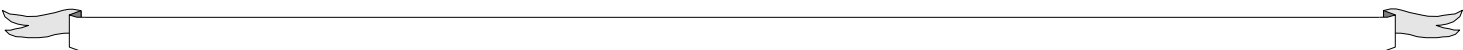


INTRODUCTION

Health is the most important part of any human's life. By 'health', we mean both our physical and mental health. Without a good health, it is useless to cherish any treasure of life. Sometimes, we are so busy that we are not even able to care of our own bodies and minds.

With proper care, like fitness, healthy and nutritional food, exercises and much more, we can take steps ahead towards our well-being. We often think that we need to make our diet plan or exercise plan to maintain our health and take proper guidance for the same.

In this program, our Nutritionist or Dietician add our diet and exercises that we need to do. We can also save our daily diet or exercise in a separate folder. Our Nutritionist or Dietician can retrieve it at any point of time by using the Patient's User ID. They can also delete the record if it is no longer required. It also helps our Nutritionist or Dietician to make another fitness plan or update the existing one, as per requirements.

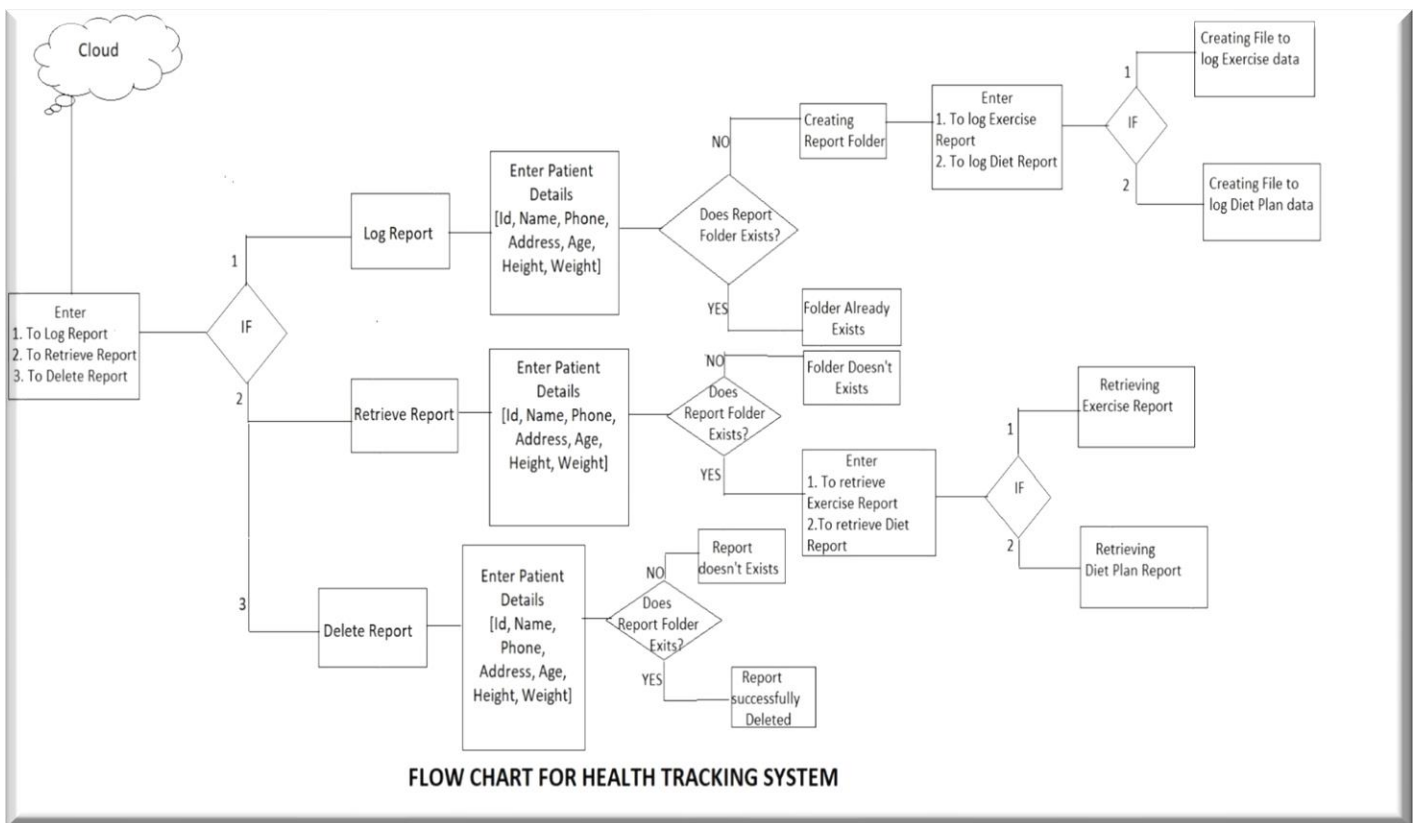


OBJECTIVE

- To provide reliable, latest and useful health nutrients information to everyone in anywhere in world with just single click.
- To amend health policies and working system on the basis of feedback received from health information system, and to combined all data and store in cloud.
- To make efficient and easy to use and crucially without losing any data
- To contribute towards achievement of objectives of health policies and programs. To increase efficiency and quality in health management.



FLOW CHART DIAGRAM OF HEALTH TRACKING SYSTEM



TOOLS AND SOFTWARES

Tools and Software's:

- Visual Studio Code:

Visual Studio Code is a code editor redefined and optimized for building and debugging modern web and cloud applications. Visual Studio Code is free and available on our favourite platforms - Linux, macOS, and Windows.



- AWS Cloud :

Amazon Web Services. Amazon Web Services (AWS) is a secure cloud services platform, offering compute power, database storage, content delivery and other functionality to help businesses scale and grow.



Programming Languages:

- Python :

Python is an interpreted language, which precludes the need to compile code before executing a program because Python does the compilation in the background. Because Python is a high-level programming language, it abstracts many sophisticated details from the programming code.





APPROACH

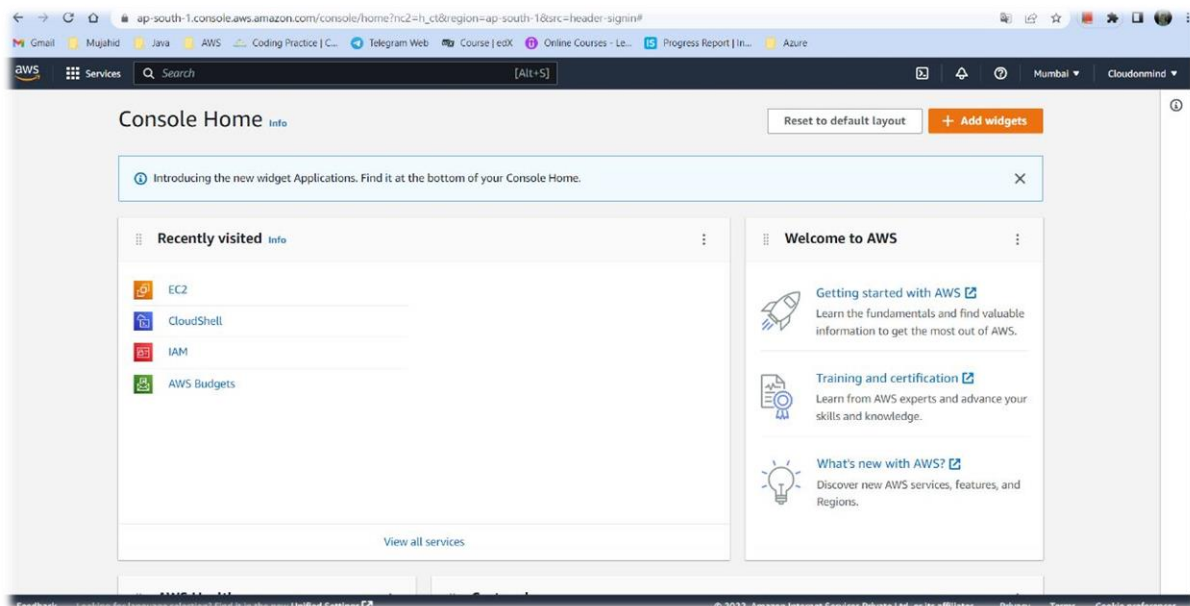
In this program, the user inputs his/her diet and exercise, and then we save the input in a separate Folder. So, whenever the user wants, they can Retrieve and Delete the Routine. Following operations that are eligible to be performed under this are :

- **`create_directory()`**
- **`create_record ()`**
- **`log_record ()`**
- **`show_directory ()`**
- **`show_record ()`**
- **`retrieve_record()`**
- **`delete_record()`**
- **`create a instance ()`**
- **`create your key pair ()`**



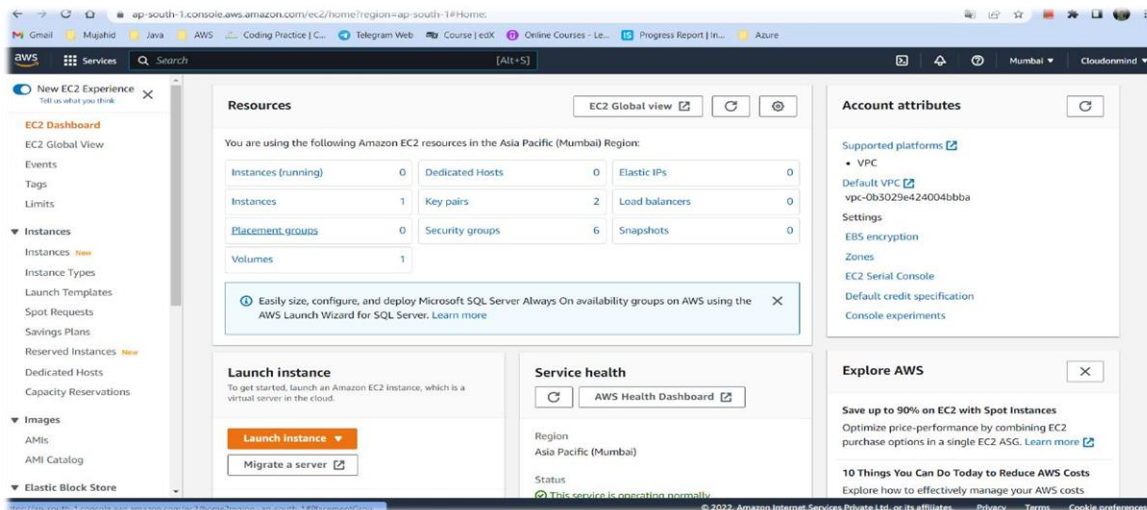
create your key pair ()

Step 1: Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.



Step 2: In the navigation pane, choose Key Pairs. 5Amazon Elastic Compute Cloud User Guide for Windows Instances Create a security group

Step 3: Choose Create key pair.



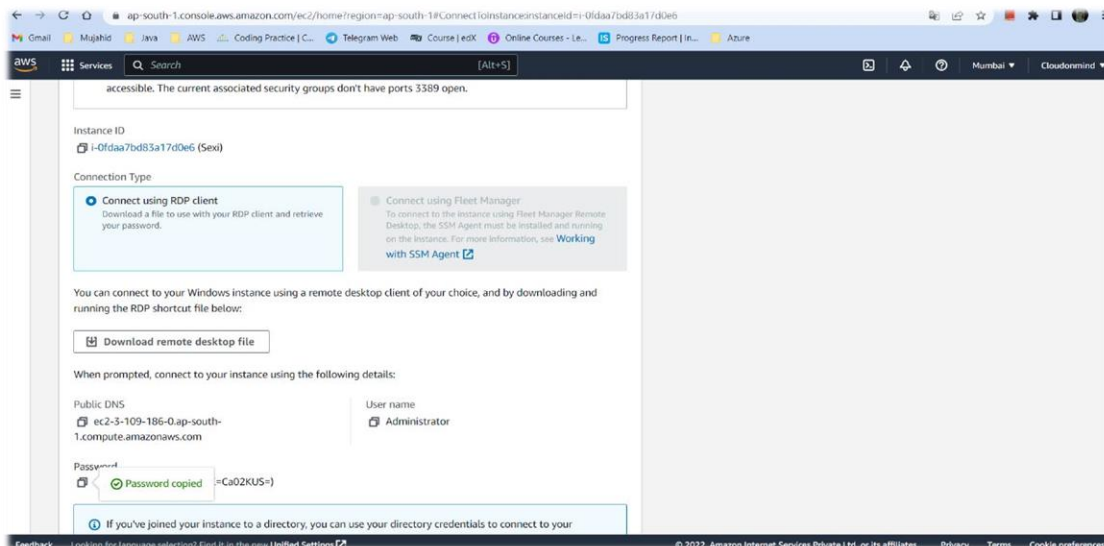
Step 4: For Name, enter a descriptive name for the key pair. Amazon EC2 associates the public key with the name that you specify as the key name. A key name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Step 5: For Key pair type, choose either RSA or ED25519. Note that ED25519 keys are not supported for Windows instances.

Step 6: For Private key file format, choose the format in which to save the private key. To save the private key in a format that can be used with OpenSSH, choose PEM. To save the private key in a format that can be used with Putty, choose PPK.

Step 7: Choose Create key pair.

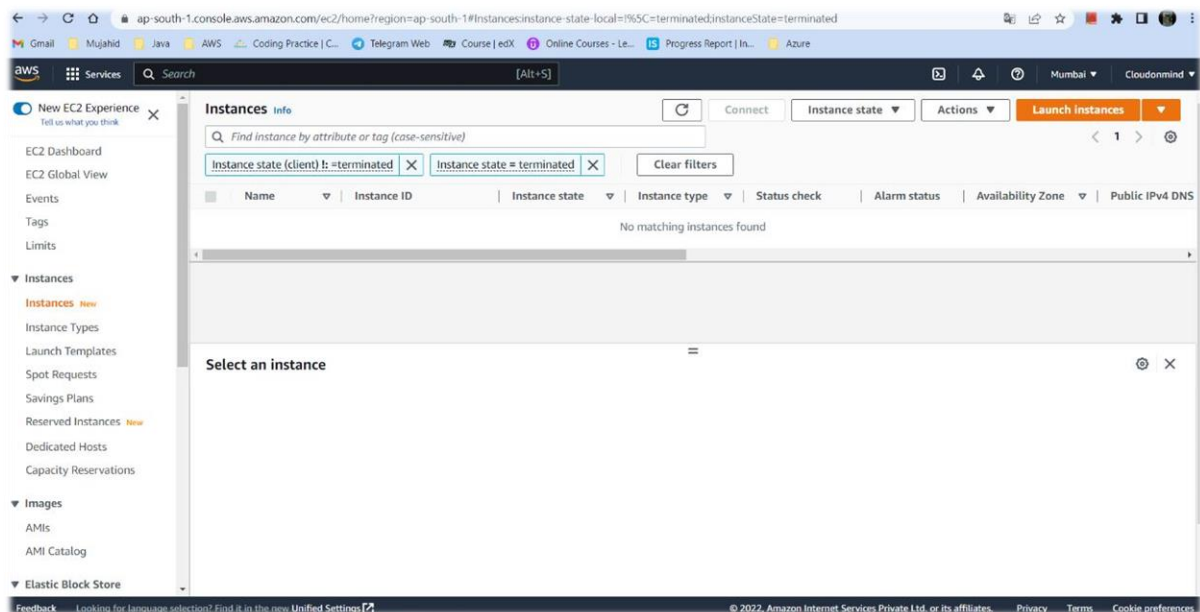
Step 8: The private key file is automatically downloaded by your browser. The base file name is the name you specified as the name of your key pair, and the file name extension is determined by the file format you chose. Save the private key file in a safe place. Important This is the only chance for you to save the private key file.



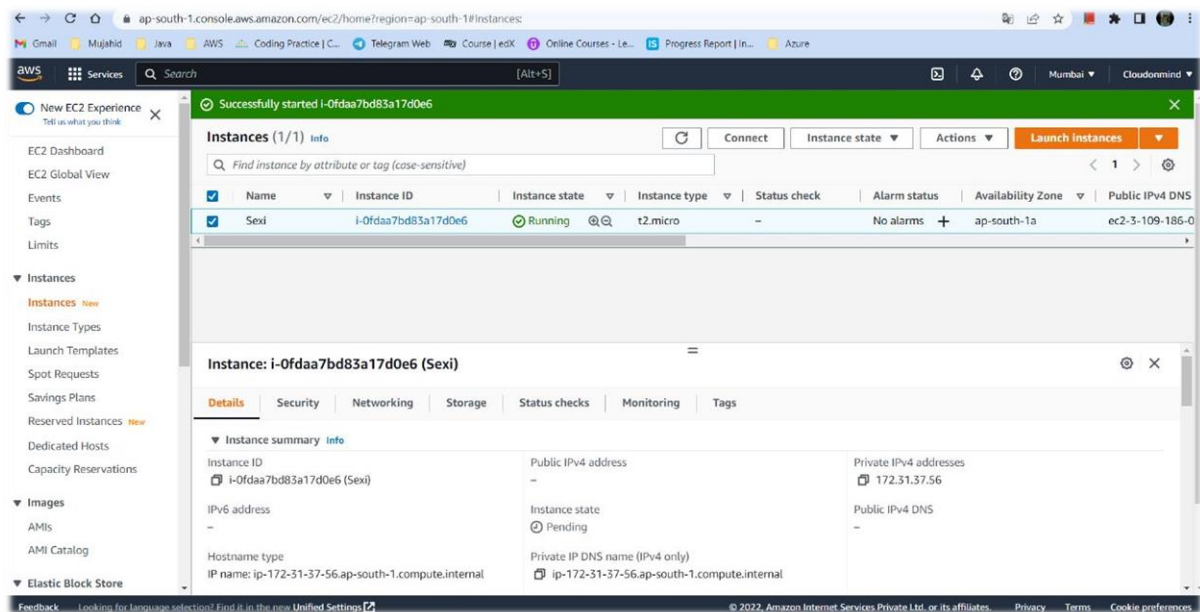
- **create a instance ()**

Step 1: Launch an instance You can launch a Windows instance using the AWS Management Console as described in the following procedure. This tutorial is intended to help you quickly launch your first instance, so it doesn't cover all possible options. For information about advanced options, see Launch an instance using the new launch instance wizard (p. 509). For information about other ways to launch your instance, see Launch your instance (p. 507). To launch an instance 1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.


Step 2: From the EC2 console dashboard, in the Launch instance box, choose Launch instance, and then choose Launch instance from the options that appear.



Step 3: Under Name and tags, for Name, enter a descriptive name for your instance.



Step 4: Under Application and OS Images (Amazon Machine Image), do the following: a. Choose Quick Start, and then choose Windows. This is the operating system (OS) for your instance. b. From Amazon Machine Image (AMI), select the AMI for Windows Server 2016 Base or later.. Notice that these AMIs are marked Free tier eligible. An Amazon Machine Image (AMI) is a basic configuration that serves as a template for your instance.



Step 5: Under Instance type, from the Instance type list, you can select the hardware configuration for your instance. Choose the t2.micro instance type, which is selected by default. The t2.micro instance type is eligible for the free tier. In Regions where t2.micro is unavailable, you can use a t3.micro instance under the free tier. For more information, see AWS Free Tier.

Step 6: Under Key pair (login), for Key pair name, choose the key pair that you created when getting set up. Note that you must select an RSA key. ED25519 keys are not supported for Windows instances. 10Amazon Elastic Compute Cloud User Guide for Windows Instances Step 2: Connect to your instance Warning Do not choose Proceed without a key pair (Not recommended). If you launch your instance without a key pair, then you can't connect to it.

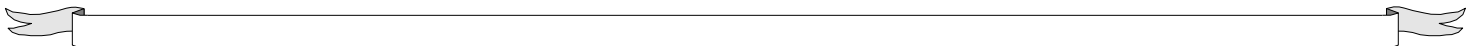
Step 7: Next to Network settings, choose Edit. For Security group name, you'll see that the wizard created and selected a security group for you. You can use this security group, or alternatively you can select the security group that you created when getting set up using the following steps: a. Choose Select existing security group. b. From Common security groups, choose your security group from the list of existing security groups.

Step 8: Keep the default selections for the other configuration settings for your instance.

Step 9: Review a summary of your instance configuration in the Summary panel, and when you're ready, choose Launch instance.

Step10: A confirmation page lets you know that your instance is launching. Choose View all instances to close the confirmation page and return to the console.

Step 11: On the Instances screen, you can view the status of the launch. It takes a short time for an instance to launch. When you launch an instance, its initial state is pending. After the instance starts, its state changes to running



and it receives a public DNS name. If the Public IPv4 DNS column is hidden, choose the settings icon () in the top-right corner, toggle on Public IPv4 DNS, and choose Confirm.

12. It can take a few minutes for the instance to be ready for you to connect to it. Check that your instance has passed its status checks; you can view this information in the Status check column.

create_directory() :

This function is used to create the user's Report directory.

```
def create_directory(directory_name, sub_dir_name) :  
    parent_dir = "C:/Users/kshah/Downloads/Final-Year-Python-Project (1)/Final-Year-Python-Project"  
    if(sub_dir_name):  
        parent_dir = parent_dir + '/' + directory_name  
        path = os.path.join(parent_dir, sub_dir_name)  
    else:  
        path = os.path.join(parent_dir, directory_name)  
    is_dir_exist = os.path.exists(path)  
    if(is_dir_exist):  
        return True  
    else:  
        os.mkdir(path)  
        return False
```

- **create_record() :**

This function is used to create the Patient's Report Text File where his/her Nutritionist or Dietician can store all their Exercise plans and Diet plans.

```
def create_record(final_dir_path, file_name, user_data, plan_details) :  
    final_file_path= final_dir_path+file_name+"-report.txt"  
    is_file_exist = os.path.exists(final_file_path)  
    if(is_file_exist):  
        return True  
    else:  
        with open(final_file_path, 'w') as fp:  
            fp.write("Patient Details:"+ "\n"  
                    +"ID: " +user_data.get("id")+"\n"  
                    +"Name: " +user_data.get("name")+"\n"  
                    +"Phone No: " +user_data.get("phone")+"\n"  
                    +"Address: " +user_data.get("address")+"\n"  
                    +"Age: " +user_data.get("age")+"\n"  
                    +"Height: " +user_data.get("height")+"\n"  
                    +"Weight: " +user_data.get("weight")+"\n"+" \n"  
                    +file_name+ " " +"Details: "+" \n"  
                    +plan_details )  
            fp.close()  
        return False
```

- **Log_record**

This function helps to specify data related to what task has to be performed. The choice here given is between food and exercise. This also accepts Complete User Details Dictionary i.e. User ID, Name, Address, Age, Height etc. and returns it for further processing. As per the Patient's choice it logs the record.

```

def log_record(patient_details):
    print("patients details: ",patient_details)
    directory_name = patient_details.get("id")+ "-report"
    is_dir_created = create_directory(directory_name, False)
    if(is_dir_created):
        print("""
=====
Record already exists for this Patient in our Database!!
=====
""")
    else:
        print("""
=====
Record doesn't exists for this Patient. Creating new Record!!
=====
""")

    print("""
=====
1.To LOG FITNESS Record:
2.To LOG HEALTH Record:
=====
""")
    log_option = int(input("Enter your choice: "))
    if(log_option == 1):
        print("""
=====
1.To LOG WORKOUT Details:
2.To LOG DIET PLAN Details:
=====
""")
        activity = int(input("Enter your choice: "))
        fitness_directory_name = patient_details.get("id") + "-fitness-report"

        is_fitness_dir_created = create_directory(directory_name, fitness_directory_name)
        if(is_fitness_dir_created):
            print("""
=====
FITNESS Directory for this Patient already exists in our Database!!
=====
""")
        else:
            print("""
=====
Fitness Record doesn't exists for this Patient. Creating new Record!!
=====
""")

```

```

parent_fitness_dir_path = r'C:/Users/kshah/Downloads/Final-Year-Python-Project (1)/Final-Year-Python-Project/'+directory_name+'/'+fitness_directory_name+'/'
if(activity == 1):
    # Directory
    workout_data = input("Type your WORKOUT Plan here....\n")
    workout_record_name='workout'
    create_workout_record=create_record(parent_fitness_dir_path, workout_record_name, patient_details, workout_data)
    print("""
        =====
        || Workout Details successfully Logged in the Database ||
        =====

    """)
elif(activity == 2):
    diet_data = input("Type your DIET Plan here....\n")
    diet_record_name='diet'
    create_diet_record=create_record( parent_fitness_dir_path, diet_record_name, patient_details, diet_data)
    print("""
        =====
        || Diet Plan Details successfully Logged in the Database ||
        =====

    """)

else:
    print("""
        =====
        Please Choose Valid Activity option
        =====

    """)
    os.system("pause")

elif(log_option == 2):
    print("""
        -----
        We will Add in Major project(diabetes prediction,breast cancer prediction, fever etc.)
        -----

    """)
else:
    print("""
        =====
        Please Choose Valid Fitness option
        =====

    """)

```

● retrieve_record() :

This function is used to retrieve the Patient Record. Nutritionist or Dietician can get specific Patient details by their Patient User ID to keep the track of their Fitness or Diet.

```
def retrieve_record(id):
    directory_name = id + "-report"
    is_dir_created=show_directory(directory_name, False)
    if(is_dir_created):
        print("""
        |
        | -----
        | =====
        | Record exists for this Patient in the Database!!
        | =====
        | -----
        |
        """)
    print("""
    |
    | 1.To RETRIEVE FITNESS Record
    | 2.To RETRIEVE HEALTH Record(We will Add in Major project):
    |
    """)
    log_option = int(input("Enter your choice: "))

    if(log_option == 1):
        print("""
        |
        | 1.To RETRIEVE WORKOUT Details
        | 2.To RETRIEVE DIET PLAN Details:
        |
        """)
        activity = int(input("Enter your choice: "))
        fitness_directory_name = id + "-fitness-report"

        is_fitness_dir_created = show_directory(directory_name, fitness_directory_name)

        if(is_fitness_dir_created):
            print("""
            |
            | =====
            | FITNESS Directory for this Patient exists in our Database!!
            | =====
            |
            """)
```

```
parent_fitness_dir_path=r'C:/Users/kshah/Downloads/Final-Year-Python-Project (1)/Final-Year-Python-Project/'+directory_name+'/'+fitness_directory_name+'/'
```

```
if(activity == 1):
```

```
    # Directory
```

```
    workout_record_name='workout-report.txt'
```

```
    show_workout_record=show_record(parent_fitness_dir_path, workout_record_name)
```

```
    print("""
```

```
        =====
        || Workout Details successfully Retrieved from the Database ||
        =====
    """)
```

```
elif(activity == 2):
```

```
    diet_record_name='diet-report.txt'
```

```
    show_diet_record=show_record(parent_fitness_dir_path, diet_record_name)
```

```
    print("""
```

```
        =====
        || Diet Plan Details successfully Retrieved from the Database ||
        =====
    """)
```

```
else:
```

```
    print("""
```

```
        =====
        Please Choose Valid Activity option
        =====
    """)
```

```
else:
```

```
    print("""
```

```
        =====
        Fitness Record doesn't exists for this Patient in our Database!!
        =====
    """)
```

```
elif(log_option == 2):
```

```
    print("""
```

```
        -----
        We will Add in Major project(diabetes prediction,breast cancer prediction, fever etc.)
        -----
    """)
```

```
else:
```

```
    print("""
```

```
        =====
        Please Choose Valid Fitness option
        =====
    """)
```

```
else:
```

```
    print("""
```

```
        =====
        Record doesn't exists for this Patient in the Database!!
        =====
    """)
```

```
os.system("pause")
```

- **show_directory() :**

This function is used to show the Patient's Report directory.

```
def show_directory(directory_name,sub_dir_name):
    parent_dir = "C:/Users/kshah/Downloads/Final-Year-Python-Project (1)/Final-Year-Python-Project"
    if(sub_dir_name):
        parent_dir= parent_dir+'/'+directory_name
        path = os.path.join(parent_dir, sub_dir_name)
    else:
        path = os.path.join(parent_dir, directory_name)
        # Path

    is_dir_exist = os.path.exists(path)
    if(is_dir_exist):
        return True
    else:
        return False
```

- **show_record() :**

This function is used to show the Patient's Report Text File where your Nutritionist or Dietician can store all of your Exercise and Diet Plans.

```
def show_record(final_dir_path,file_name):
    final_file_path= final_dir_path+file_name
    is_record_exist = os.path.exists(final_file_path)

    if(is_record_exist):
        print("""
        -----
        =====
        Record exists for this Patient in our Database!!
        =====
        -----
        """)
        with open(final_file_path, 'r') as fp:
            content = fp.read()
            print(content)
            fp.close()
            return True
    else:
        print("""
        -----
        =====
        No Record exists for this Patient in the Database!!
        =====
        -----
        """)
        return False
```

- **delete_record() :**

This function is used to Delete the Patient Details from the Record if it's no longer required.

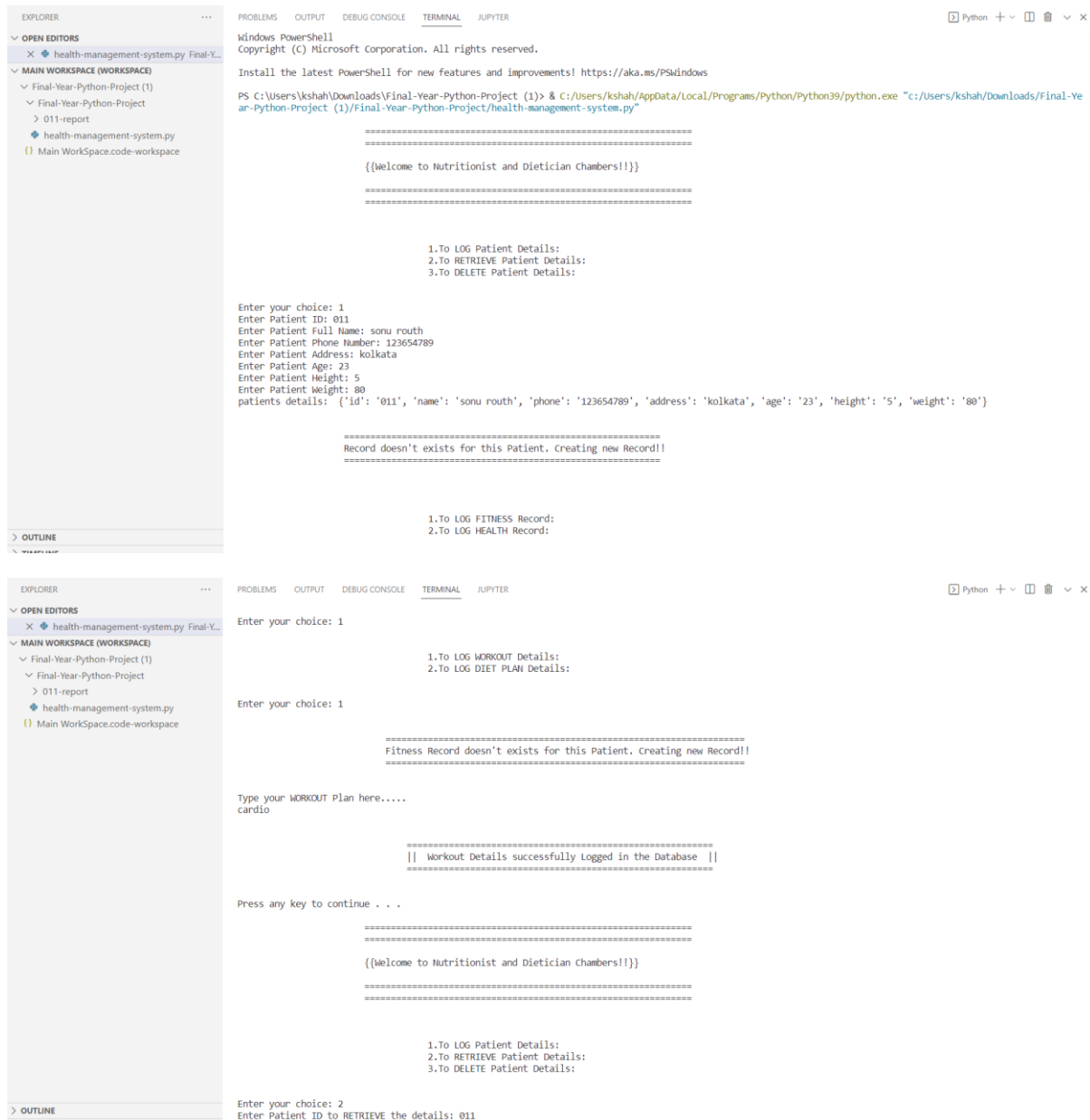
```
def delete_record(id):
    directory_name = id + "-report"
    parent_dir = "C:/Users/kshah/Downloads/Final-Year-Python-Project (1)/Final-Year-Python-Project"
    is_dir_Exists=show_directory(directory_name, False)
    if(is_dir_Exists):
        print("""
            -----
            =====
            Record exists for this Patient in our Database!!
            =====
            -----

            """)
        path = os.path.join(parent_dir, directory_name)
        # removing directory
        shutil.rmtree(path)
        print("""
            =====
            Record deleted successfully!!
            =====

            """)
    else:
        print("""
            =====
            No Record exists for this Patient in the Database!!
            =====

            """)
    os.system("pause")
```

OUTPUTS



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\kshah\Downloads\Final-Year-Python-Project (1)> & C:\Users\kshah\AppData\Local\Programs\Python\Python39\python.exe "c:\Users\kshah\Downloads\Final-Year-Python-Project (1)\Final-Year-Python-Project\health-management-system.py"

=====
{(Welcome to Nutritionist and Dietician Chambers!!)}
=====

1.To LOG Patient Details:
2.To RETRIEVE Patient Details:
3.To DELETE Patient Details:

Enter your choice: 1
Enter Patient ID: 011
Enter Patient Full Name: sonu routh
Enter Patient Phone Number: 123654789
Enter Patient Address: kolkata
Enter Patient Age: 23
Enter Patient Height: 5
Enter Patient Weight: 80
patients details: {'id': '011', 'name': 'sonu routh', 'phone': '123654789', 'address': 'kolkata', 'age': '23', 'height': '5', 'weight': '80'}

=====
Record doesn't exists for this Patient. Creating new Record!!
=====

1.To LOG FITNESS Record:
2.To LOG HEALTH Record:

Enter your choice: 1

1.To LOG WORKOUT Details:
2.To LOG DIET PLAN Details:

Enter your choice: 1

=====
Fitness Record doesn't exists for this Patient. Creating new Record!!
=====

Type your WORKOUT Plan here.....
cardio

=====
|| Workout Details successfully Logged in the Database ||
=====

Press any key to continue . . .

=====
{(Welcome to Nutritionist and Dietician Chambers!!)}
=====

1.To LOG Patient Details:
2.To RETRIEVE Patient Details:
3.To DELETE Patient Details:

Enter your choice: 2
Enter Patient ID to RETRIEVE the details: 011
```



```
EXPLORER  ...  PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  JUPYTER
OPEN EDITORS
  X health-management-system.py Final-Y...
MAIN WORKSPACE (WORKSPACE)
  Final-Year-Python-Project (1)
  Final-Year-Python-Project
    > 011-report
    health-management-system.py
    Main Workspace.code-workspace

Record exists for this Patient in the Database!!
=====

1.To RETRIEVE FITNESS Record
2.To RETRIEVE HEALTH Record(We will Add in Major project):

Enter your choice: 1

1.To RETRIEVE WORKOUT Details
2.To RETRIEVE DIET PLAN Details:

Enter your choice: 1

=====
FITNESS Directory for this Patient exists in our Database!!
=====

=====
Record exists for this Patient in our Database!!
=====
```

Patient Details:
ID: 011
Name: sonu routh
Phone No: 123654789
Address: kolkata
Age: 23
Height: 5
Weight: 80

workout Details:
cardio

```
=====
|| Workout Details successfully Retrieved from the Database ||
=====
```

Press any key to continue . . .

```
=====
=====
{{Welcome to Nutritionist and Dietician Chambers!!}}
=====
=====
```

- 1.To LOG Patient Details:
- 2.To RETRIEVE Patient Details:
- 3.To DELETE Patient Details:

Enter your choice: 3
Enter Patient ID to DELETE the details: 011

```
=====
Record exists for this Patient in our Database!!
=====
```

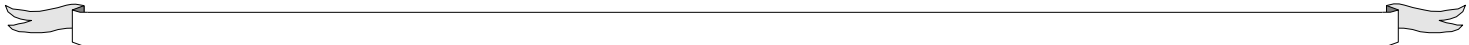
```
=====
Record deleted successfully!!
=====
```

Press any key to continue . . . █



APPLICATIONS

This project is based on a trending technology of the present times and has many applications –

- 1. One of the most important applications of this project is that, the accessibility to the directories and records will be available to users 24x7 due to the use of cloud service provider (here, AWS), making it more user friendly.**
 - 2. This project paves the path for a smooth guidance to all the health-conscious individuals irrespective of their age and health conditions.**
 - 3. The users are also given relief from the hassle of storage issues when it comes to using this application as it provides the opportunity of online data storage.**
 - 4. Data can be inserted, updated when required, deleted, and can also be saved separately in folders for uncomplicated distinguish between information of users.**
 - 5. This project ensures that users can access resources in real time from any location or device i.e., mobility is offered to the users in this project.**
- 

ADVANTAGES

1. **Cost – effective** : We only need to pay for the access services of the APIs this project procures as it is a Pay-Per-Use model and does not need any computing infrastructure or any other hardware or software.
2. **Backup & Restore** : There's a data backup plan incorporated in this project as it's using a cloud service provider AWS (here) which helps us to recover data even in case of any disasters. There is no tension for loss of data.
3. **Data Security** : All the data of the patients are securely stored and handled with the help of some advanced features that AWS offers.
4. **User – friendly** : Easy to use and reduces paperwork. Also, time consumption required for all the data work is reduced to the minimum. Patient details are also easily managed.
5. **Mobility** : The application used by the users can easily be accessed from any location in real-time and in mobiles/tablets/computers, etc.



DISADVANTAGES

1. **Internet Connectivity** : Internet connection is mandatory to access the records of the patients. There is no other way to access data/records of the users or patients from the cloud.
2. **Limited Control** : Cloud infrastructure is completely owned, managed, and monitored by the service provider AWS (here) and so we have less control over the function and execution of services within a cloud infrastructure.
3. **Downtime** : AWS may also face power loss, service maintenance, etc which might cause interruptions to the users.



CONCLUSION

As Healthcare Services are important part of our society, automating these services lessen the burden on humans and eases the measuring process. Also, the transparency of this system helps Doctors or Patients to trust it.

The objective of developing Health Tracking System is to helps our Nutritionist or Dietician to keep track of our daily Health Routines like Exercises and Diets and to provide instant suggestions whenever needed ensuring smooth communication.





FUTURE DEVELOPMENT

In near future, we are planning to make a website with help of SQL server on that server data will be saved for all students of Computer Applications Department, We're planning to develop a site that'll help to keep records of all students who're attending class and vice versa add on with the help of Python Programming language, We'll send notifications to all students via SMS or WhatsApp Medium for attendance and other crucial information

Like some student haven't filled some important document, we'll send notifications via SMS to complete the given task as per given timeline, and each and every data of students will be saved and recorded to our SQL database. That'll help to keep record of every students for filling applications. We promised to our

Prof. Dr Ananjan Mity, We'll complete this in our Major under the guidance of our Professor. And make sure this will be up to the expectations and useful applications to our Department.





BIBLIOGRAPHY

This project synopsis has been prepared by taking help from –

- **Our Project Guide**

- **Book : Cloud Computing – Concepts, Technology & Architecture (1st Edition) by Thomas Erl, Zaigham Mahmood, and Ricardo Putting**

- **Useful sites :**
 1. <http://www.wikipedia.org>
 2. cloud.google.com
 3. <https://www.instructables.com/id/NodeMCU-ESP8266-Details-and-Pinout/>
 4. <https://www.salesforce.com/in/products/platform/best-practices/>





INDIVIDUAL CONTRIBUTION TO THE PROJECT

NAME OF THE PROJECT MEMBER	CONTRIBUTION
MUJAHID ALI ANSARI	AWS CLOUD SERVICE
ASHUTOSH KUMAR YADAV	PRESENTATION
SHAHIL KUMAR CHOURASIA	PYTHON CODING
SONU ROUTH	PYTHON CODING
SIRSHA MAJUMDER	PROJECT REPORT
