

# **PROJECT REPORT**

*on*

## **Medical Test Recommender**

*(CSE VI Semester Mini project PCS-604)*

**2021-2022**



***Submitted to:***

*Mr. Samir Rana*

*(CC-CSE-D-VI-Sem)*

***Submitted by:***

*Mr. Chinmay Tiwari*

*Roll. No.: 1918306*

*CSE-D-VI-Sem*

*Session: 2021-2022*

**DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY**

**GRAPHIC ERA HILL UNIVERSITY, DEHRADUN**

# **CERTIFICATE**

Certified that Mr. Chinmay Tiwari (Roll No.- 1918306) has developed mini project on “Credit Card Fraud Detection” for the CS VI Semester Mini Project Lab (PCS-604) in Graphic Era Hill University, Dehradun. The project carried out by Students is their own work as best of my knowledge.

Date:30/06/2022

(Mr. Samir Rana)

**Class Co-Ordinator**

**CSE-D-VI-Sem**

(CSE Department)

GEHU Dehradun

(Mr. Sushant Chamoli)

**Project Guide**

Resource Person

(CSE Department)

GEHU Dehradun

# **ACKNOWLEDGMENT**

We would like to express our gratitude to The Almighty Shiva Baba, the most Beneficent and the most Merciful, for completion of project.

We wish to thank our parents for their continuing support and encouragement. We also wish to thank them for providing us with the opportunity to reach this far in our studies.

We would like to thank particularly our project Co-Ordinator Mr. Samir Rana and our Project Guide Mr. Sushant Chamoli for his patience, support, and encouragement throughout the completion of this project and having faith in us.

At last but not the least we greatly indebted to all other persons who directly or indirectly helped us during this work.

**Mr. Chinmay Tiwari**

**Roll No.- 1918306**

**CSE-D-VI-Sem**

**Session: 2021-2022**

**GEHU, Dehradun**

# **Table Of Contents**

- 1. Abstract**
- 2. Motivation**
- 3. Software Requirements**
- 4. Hardware Requirements**
- 5. Language Used**
- 6. Conclusion**

## **ABSTRACT:**

The goal of this project is to build a Medical Test Recommender System with the help of questionnaire.

Medical test recommendation system based on the analysis of patients' symptoms and anamneses. The exact test selection for a specific patient can be time consuming and error-prone due to the huge amount of information to be considered like: the number of tests, patients, long working hours, exceptional cases, etc.

The number of medical tests that are applicable in the hospitals is too high, therefore only 20 most frequently required ones are selected. The promising results of the study indicated that the symptoms given as plain text can be efficiently utilized by the experts for medical test selection.

## **MOTIVATION:**

In this project, I have designed, implemented, and analyzed a Medical Test Recommender System using questionnaire on what symptoms a patient is having. By Building this project I got to learn about various diseases and what symptoms can lead to serious risk issues if ignored.

## **SOFTWARE REQUIREMENTS:**

- Jupyter Notebook
- Python

## **HARDWARE REQUIREMENTS:**

- 2 GHz Intel or high processor
- Minimum of 180 GB HDD
- At least should have 2 GB RAM

## **LANGUAGE USED:**

- Python

## **MEDICAL TEST RECOMMENDER SYSTEM:**

*A Medical Recommender System* is a specialization of an Recommendation system. In the context of an HRS, a recommendable item of interest is a piece of non-confidential, scientifically proven or at least generally accepted medical information, which in itself is not linked to an individual's medical history. However, an HRS's suggestions are driven by individualized health data such as documented in a *personal health record* (PHR). This source of information is considered the user profile of an *recommender system*.

Recommender systems are employed in many fields to help users to find important products and services for them. Similar approaches can be headed for providing diagnosis, thus supporting physicians in their work.

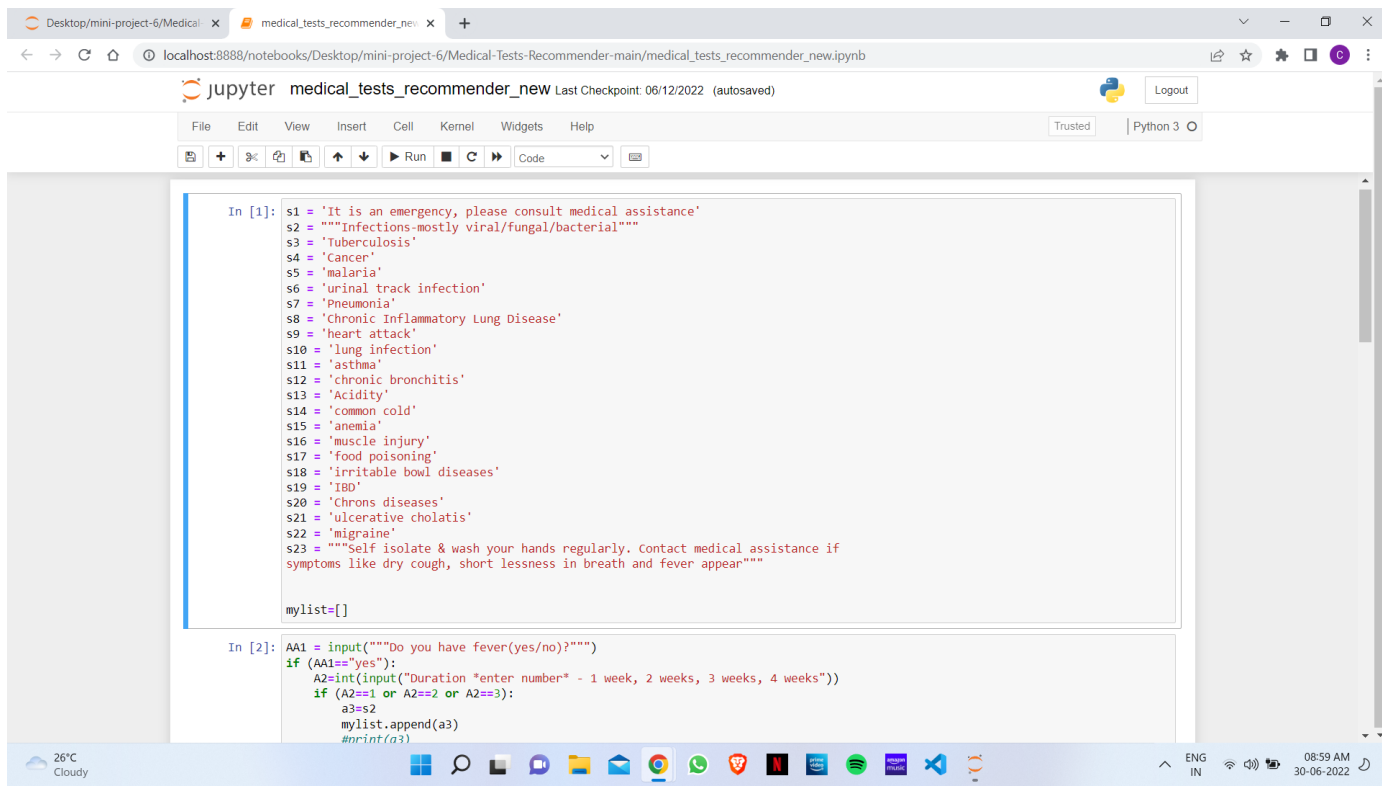
I presented a content-based recommender system within the medical domain, by providing an overview of recent information retrieval and semantic enrichment tools we employed. Our work addressed the challenge to find out which types of information can be directly processed by machines on large collections of symptoms to return the reliable results.

## **CODE IMPLEMENTATION:**

1. Defining all the diseases: There are in total 23 diseases in our system for which we are making a medical recommendation system.

2. After that we are taking the input from the user that for which disease, he wants the diagnosis.

## Screenshot Of Project:



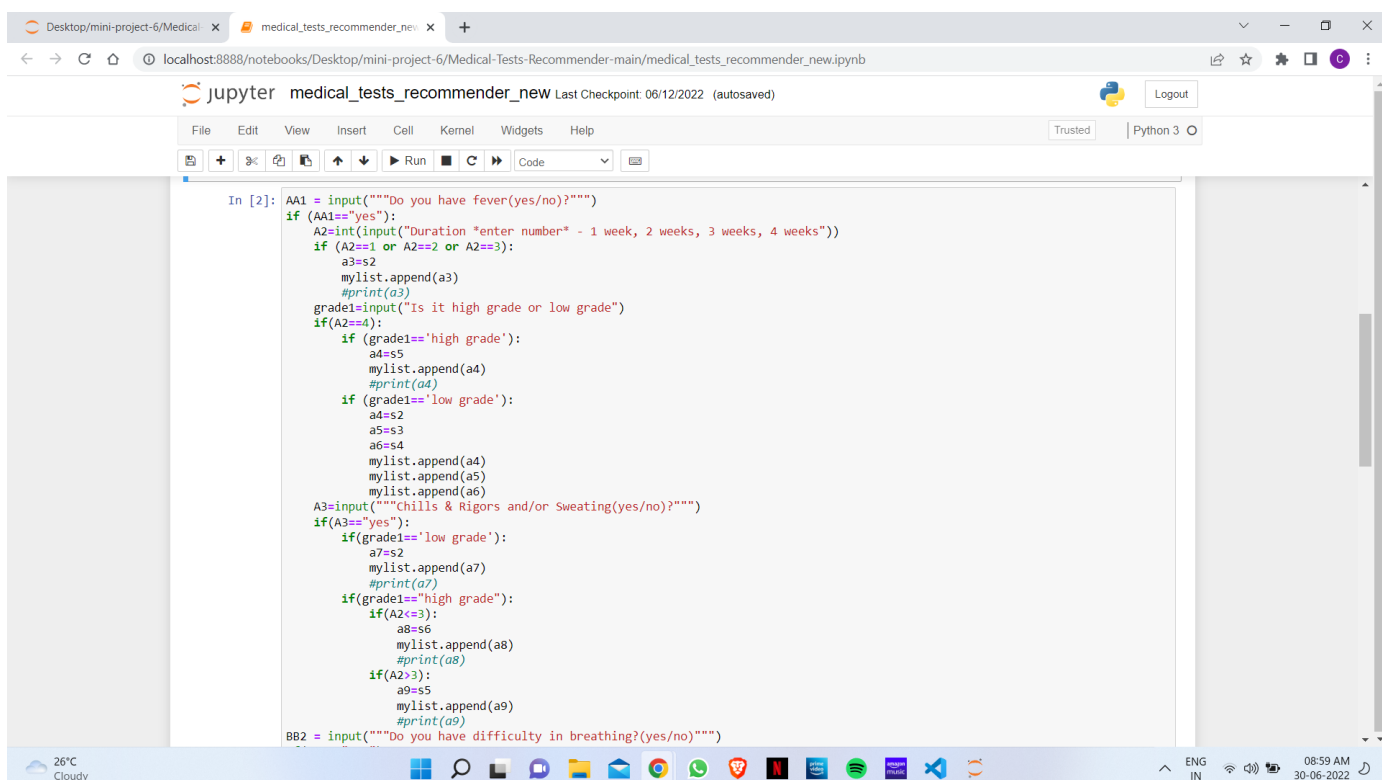
The screenshot shows a Jupyter Notebook interface with two code cells. The first cell (In [1]) defines a list of medical conditions and symptoms. The second cell (In [2]) starts the user input logic for fever and duration.

```
In [1]: s1 = 'It is an emergency, please consult medical assistance'
s2 = 'Infections-mostly viral/fungal/bacterial'
s3 = 'Tuberculosis'
s4 = 'Cancer'
s5 = 'malaria'
s6 = 'urinal track infection'
s7 = 'Pneumonia'
s8 = 'Chronic Inflammatory Lung Disease'
s9 = 'heart attack'
s10 = 'lung infection'
s11 = 'asthma'
s12 = 'chronic bronchitis'
s13 = 'Acidity'
s14 = 'common cold'
s15 = 'anemia'
s16 = 'muscle injury'
s17 = 'food poisoning'
s18 = 'irritable bowl diseases'
s19 = 'TBD'
s20 = 'Chrons diseases'
s21 = 'ulcerative cholatis'
s22 = 'migraine'
s23 = 'Self isolate & wash your hands regularly. Contact medical assistance if
symptoms like dry cough, short lessness in breath and fever appear'

mylist=[]

In [2]: AA1 = input("Do you have fever(yes/no)?")
if (AA1=="yes"):
    A2=int(input("Duration *enter number* - 1 week, 2 weeks, 3 weeks, 4 weeks"))
    if (A2==1 or A2==2 or A2==3):
        a3=s2
        mylist.append(a3)
        #print(a3)
```

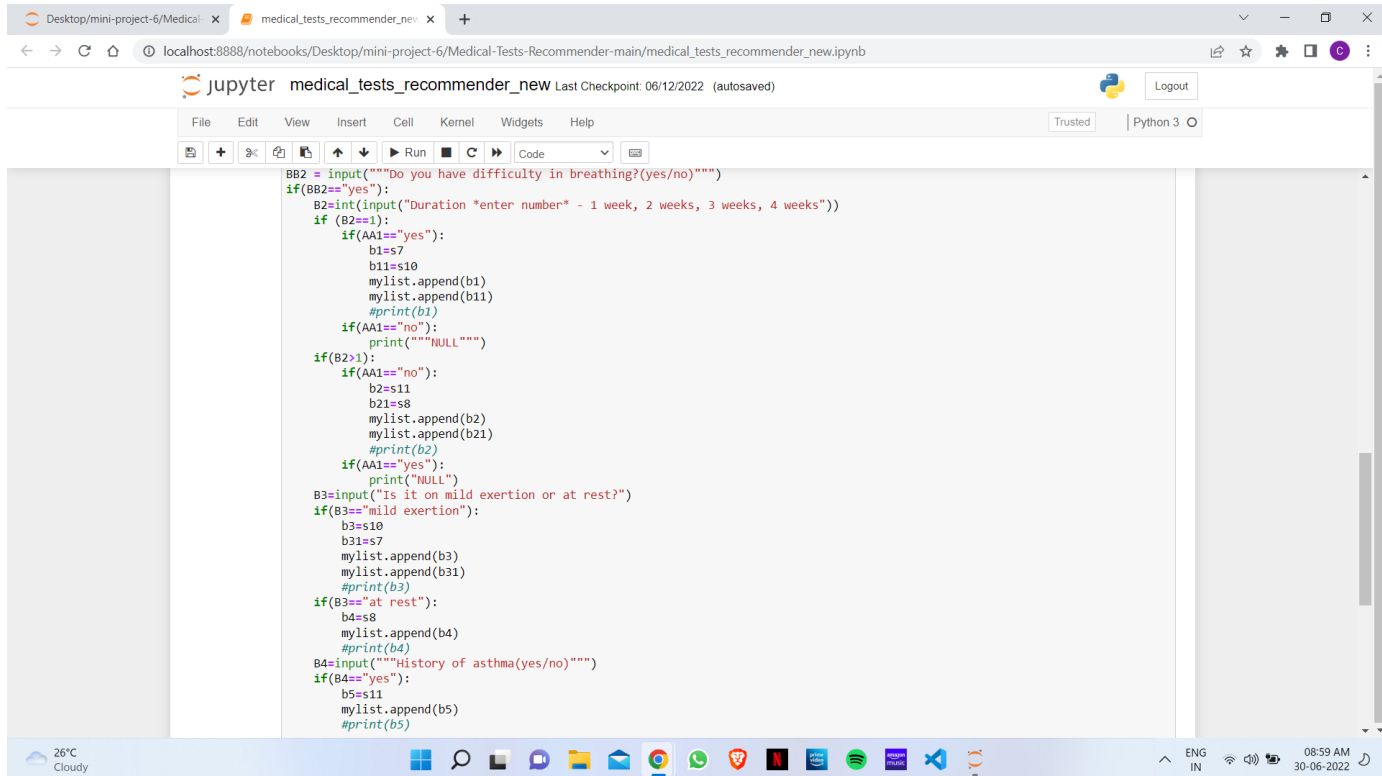
3. Inputting other informations from user like the duration of the disease and the stage of the disease like if the patient is having the diease for higher degree or normal degree.



The screenshot shows the continuation of the Jupyter Notebook code. The third cell (In [2]) continues the logic for high and low grade symptoms, and adds input for chills and rigors.

```
In [2]: #print(a3)
grade1=input("Is it high grade or low grade")
if(A2==4):
    if (grade1=='high grade'):
        a4=s5
        mylist.append(a4)
        #print(a4)
    if (grade1=='low grade'):
        a4=s2
        a5=s3
        a6=s4
        mylist.append(a4)
        mylist.append(a5)
        mylist.append(a6)
A3=input("chills & Rigors and/or Sweating(yes/no)?")
if(A3=="yes"):
    if(grade1=='low grade'):
        a7=s2
        mylist.append(a7)
        #print(a7)
    if(grade1=='high grade'):
        if(A2<=3):
            a8=s6
            mylist.append(a8)
            #print(a8)
        if(A2>3):
            a9=s5
            mylist.append(a9)
            #print(a9)
BB2 = input("Do you have difficulty in breathing?(yes/no)")
```

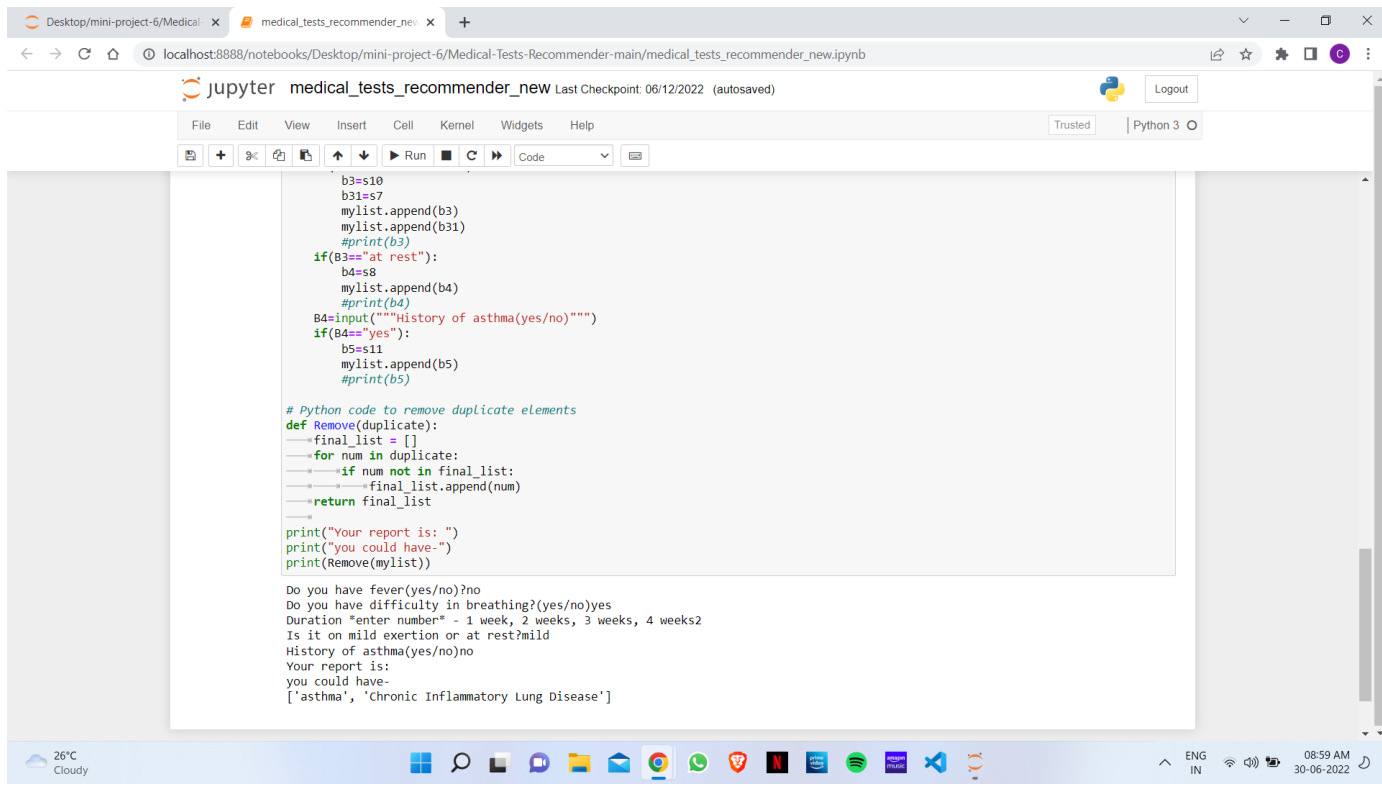
4. Checking if the customer is having other diseases like if he is having history of asthma then we can detect the right disease for him.



```
BB2 = input("Do you have difficulty in breathing?(yes/no)")
if(BB2=="yes"):
    B2=int(input("Duration *enter number* - 1 week, 2 weeks, 3 weeks, 4 weeks"))
    if (B2==1):
        if(AA1=="yes"):
            b1=s7
            b11=s10
            mylist.append(b1)
            mylist.append(b11)
            #print(b1)
        if(AA1=="no"):
            print("NULL")
    if(B2>1):
        if(AA1=="no"):
            b2=s11
            b21=s8
            mylist.append(b2)
            mylist.append(b21)
            #print(b2)
        if(AA1=="yes"):
            print("NULL")
B3=input("Is it on mild exertion or at rest?")
if(B3=="mild exertion"):
    b3=s10
    b31=s7
    mylist.append(b3)
    mylist.append(b31)
    #print(b3)
if(B3=="at rest"):
    b4=s8
    mylist.append(b4)
    #print(b4)
B4=input("History of asthma(yes/no)")
if(B4=="yes"):
    b5=s11
    mylist.append(b5)
    #print(b5)
```

9. After various diagnosis we can predict that the patient is suffering from which disease.





```
b3=s10
b31=s7
mylist.append(b3)
mylist.append(b31)
#print(b3)
if(b3=="at rest"):
    b4=s8
    mylist.append(b4)
    #print(b4)
B4=input("History of asthma(yes/no)")
if(B4=="yes"):
    b5=s11
    mylist.append(b5)
    #print(b5)

# Python code to remove duplicate elements
def Remove(duplicate):
    final_list = []
    for num in duplicate:
        if num not in final_list:
            final_list.append(num)
    return final_list

print("Your report is: ")
print("you could have-")
print(Remove(mylist))

Do you have fever(yes/no)?no
Do you have difficulty in breathing?(yes/no)yes
Duration *enter number* - 1 week, 2 weeks, 3 weeks, 4 weeks2
Is it on mild exertion or at rest?mild
History of asthma(yes/no)no
Your report is:
you could have-
['asthma', 'Chronic Inflammatory Lung Disease']
```

## **FUTURE ENHANCEMENTS:**

Recommender systems are employed in many fields to help users to find important products and services for them. Similar approaches can be headed for providing diagnosis, thus supporting physicians in their work.

I presented a content-based recommender system within the medical domain, by providing an overview of recent information retrieval and semantic enrichment tools we employed. Our work addressed the challenge to find out which types of information can be directly processed by machines on large collections of symptoms to return the reliable results.

## **REFERENCES:**

- Wikipedia- To get most common disease and their symptoms.

## **CONCLUSION:**

This work contributes to simplify administrative functions and boost the quality of management of patients improving the quality of healthcare with models that are both transparent and safe. Our methodology can be extended to different clinical scenarios where recommender systems can be applied. The acceptance and further development of the app is one of the next important steps and still requires further development depending on specific requirements of the health management, the physicians or health professionals.

# **Thank You!**