A Project Report on

“Virtual Manager for Medical Records”

Submitted in partial fulfilment of the requirement for
Degree in Bachelor of Engineering (Information Technology)

By

Sakshi Khose (501727)
Varad More (501733)
Sonal Sarode (501776)

Guided by:

Prof. Smita Rukhande

Department of Information Technology
Fr. Conceicao Rodrigues Institute of Technology
Sector 9A, Vashi, Navi Mumbai – 400703

University of Mumbai
2019-2020

CERTIFICATE

This is to certify that the project entitled

"VIRTUAL MANAGER FOR MEDICAL RECORDS"

Submitted By

Sakshi Khose
Varad More
Sonal Sarode

In partial fulfilment of degree of **B.E**. in **Information Technology** for term work of the project is approved.

External Examiner	**Internal Examiner**
External Guide	**Internal Guide**
Head of the Department	**Principal**
Date: -	**College Seal**

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)

Sakshi Khose – 501727

Varad More – 501733

Sonal Sarode – 501776

Date:

ABSTRACT

In parallel with the advancement of innovation in Medical fields, different control techniques are additionally created. Voice control chatbot is one of these control techniques. We aim to propose a model with the help of machine learning and AI which will help to search, analyze and record clinical data and laboratory data in a patient's electronic health record and will also record patient history. Development using Machine learning and Artificial Intelligence is immensely popular. Our model will first record conversations between doctors and patients. Then using NLP, we can retrieve a patient's history and then classify them into different sections such as symptoms, allergies, etc. Prescriptions given to each patient will also be stored in the form of a soft copy. We also aim to provide notifications regarding immunizations and appointments. The innovation utilizes characteristic language handling (NLP) that perceives the voice and deciphers what the client talks utilizing calculations. These conversational operators imitate human discourse to stimulate a conversation that resembles speaking to a real person. Moreover, it will play a vital role in reminding patients with deadlines for vaccinations and follow-ups.

INDEX

LIST OF FIGURES

Abbreviation

NLP - Natural Language Processing
CTRT - Cardiotoxicity of Radiation Therapy
SQL - Structured Query Language
LSTM - Long-Short Term Memory
EHR - Electronic Health Record
RL - Reinforcement Learning
HTML - Hypertext Markup Language
TTS - Text to Speech
IFTTT - If This Then That
PDF - Portable Document Format
JSON - JavaScript Object Notation
NLTK - Natural Language ToolKit
MRI - Magnetic Resonating Image
CT/RT-Chemotherapy-Thoracic Radiotherapy

Introduction

1.1 Background

Medical records are the document that explains all details about the patient’s history, clinical findings, diagnostic test results, pre, and postoperative care, patient’s progress and medication [1]. If written correctly, notes will support the doctor about the correctness of treatment. In Spite of knowing the importance of proper record keeping in India, it is still in the initial stages. A good medical record serves the interest of the medical practitioner as well as his patients. It is very important for the treating doctor to properly document the management of the patient under his care. Moreover, even today these documents are stored, maintained and preserved in a file system manually. Hence, sometimes it is tedious to preserve these documents for patients as well as for the doctor. Even today, prescribing faults and prescription errors are major problems among medication errors. They occur both in general practice and in hospitals, and although they are rarely fatal, they can affect patients' safety and quality of healthcare.

1.2 Motivation

As the coronavirus pandemic spreads throughout the entire world, some health experts are warning that hospitals are not prepared to manage the anticipated number of patients, if there is a large spike in severe cases. In such pandemic scenarios, it becomes difficult to maintain patient records in a file system. Moreover, studies reveal that illegible handwriting and transcription errors are responsible for as much as 61 percent of medication errors in hospitals [2].

1.3 Problem Definition

The patient record is the principal repository for information concerning a patient's health care. It affects, in some way, virtually everyone associated with providing, receiving, or reimbursing health care services. Despite the many technological advances in health care over the past few decades, the typical patient record of today is remarkably similar to the patient record of 50 years ago. This failure of patient records to evolve is now creating additional stress within the already burdened health care system as the information needs of practitioners, patients, administrators, third-party payers, researchers, and policymakers often go unmet. Hence, we aim to provide a solution to this problem where in medical records, Blood tests and other reports, prescriptions of patients will be managed and preserved effectively. Moreover, symptoms of patients will be collectively extracted from conversations and stored effectively.

1.4 Scope / Assumptions

The medical field has reduced manpower due to upcoming technologies but somewhere it requires manpower to write prescriptions. NLP and text summarizers play an important role to overcome this weakness. This project will focus on analysing and recording a patient's conversation as well as the extraction of useful information from the patient's report. The conversation carried out between patients and doctors should be done in the English language only. Doctors and patients should use normal accents while speaking. Data Extraction is carried out from sources such as lab report formats where the data is extracted and stored in a database. Diseases and symptoms analysed by extractors are classified by the model. Finally, it will help doctors to maintain records of patients in sequential order and helps patients to visit doctors without any file.

1.5 Issues / Limitations

The virtual manager focuses on some important parameters such as it provides an easy interface for referring to a patient's history, resolving doctor’s handwritten issues by producing readable prescriptions and also it reduces strain on doctors for writing prescriptions. Along with these core factors, our product has some limitations such as doctors and patients have to speak compulsory in the English language. Actions of the patient and doctors are not recorded. There will be difficulty in finding knowledge from the database as the search returns information in large volumes. The main problem is about medical term variations while speaking as the term can be many forms. For example, chemotherapy can be chemoradiotherapy, chemoradiation, CT/RT, CTRT, and so on.

Literature Survey and Analysis

The project “Virtual Manager for Medical Records” makes use of various text summarization methods. Hence, in order to understand our project effectively it is necessary to know these technologies. Moreover, some existing systems are analysed to get a broader view of our project.

2.1 Related work:

1. **Finding Clinical Knowledge from MEDLINE Abstracts by Text Summarization Technique.**

Chumsak Sibunruang and Jantima Polpinij applied the concept of Text Summarizer which helps us to gather important information from the original text. Their approach is used for finding useful knowledge from the MEDLINE abstract of cancer patients. Gathering and understanding clinical knowledge requirements. However, two cancer experts help to read and select the most relevant MEDLINE abstracts in clinical trials relating to cervical cancer which is divided into different sections where text selection and cleaning using NLP is done [3]. This consequence of clinical information synopsis is assessed by looking at the outcomes from the area specialists.

2. **Formation of SQL from Natural Language query using NLP.**

Uma M, Sneha V, Sneha G, BHuvana J, Bharathi B [4] Proposed a system for extraction of data from databases using structured natural language which is further converted into SQL query using NLP as yield to the framework as an output to the system. Where a regular expression is used to map queries in the English Language. Conversion of regular structured natural language expression follows some steps such as tokenization, lemmatization, part of speech conversion tagging, parsing and mapping on data. NLP plays a tremendous job of analyzing large amounts of data. The accuracy achieved by this model is 98.89 percent.

3. **Text Summarization Based on Double-Attention Pointer Network.**

ZHIXIN LI 1, ZHI PENG1, SUQIN TANG1, CANLONG ZHANG1, AND HUIFANG MA on January 10, 2020 presents an encoder-decoder model based on a double attention pointer network for automatic text-summarization. They have used an improved coverage mechanism to remove repetition problems and improve quality of generated summaries. Scheduled sampling and reinforcement learning (RL) are combined to generate new training methods to optimize the model. Gate mechanism has been used to construct a dual attention pointer network. The paper uses the Reinforcement Learning Method to optimize the coverage mechanism. In the described model, they have built a bidirectional LSTM encoder for processing input text and a unidirectional LSTM decoder for outputting summaries. Self-attention mechanism is achieved by taking the input sequence of the source text and converting it to word embedding format and then using the bidirectional LSTM encoder for processing, thereby obtaining the sequence of the hidden states of the encoder. Probability distribution is used to obtain the key information [5]. The dual pointer network is a variant of sequence to sequence model with an attention mechanism. The bidirectional LSTM encoders are shared and can generate key information or context information by matching a self-attention mechanism or a soft attention mechanism. A truncation parameter has been introduced which ultimately optimizes the loss function and thereby improves the performance of textsummarisers. Teacher-forcing algorithm has been used for training the LSTM encoder and policy-gradient which is one of the algorithms of RL that has been used to calculate the mixed loss function.

4. Evidence Based Treatment.

According to a paper published by Kavita Lohani and Debajyoti Mukhopadhyay in 2017 presented an Evidence Based Treatment discusses the use of hive and big data warehousing tools to store the data in order to build an Electronic Health Record (EHR). These records were later used to refer for medical prescriptions in future, forming a clinical decision support system [6]. The solution used to store complete data in hive and deal with the big data using Hadoop.

2.2 Existing System

Various existing systems related to “Virtual Manager for Medical Records” have been analysed and studied. “Virtual Manager for Medical Records” has various advantages over the existing systems such as storing the medical records from conversations and producing readable prescriptions.

1. CIBS: A biomedical text summarizer using topic-based sentence clustering: -

Automatic text summarizers can reduce the time required to read lengthy text documents by extracting the most important parts. This summarization method named Clustering and Item set mining based Biomedical Summarizer (CIBS) which extracts biomedical contents from the input document and employs an itemset mining algorithm to discover main topics. It applies a clustering algorithm to put the sentences into clusters such that those in the same cluster share similar topics. Selecting sentences from all the clusters and producing a summary that covers a wide range of topics of the input text. The performance is evaluated on single- and multi-document biomedical text summarization [7].

2. Osplabs

OSP is a leading insights-as-a-service provider harnessing the dynamism of global health data to address today's healthcare challenges. Various modules covered by osplabs in the healthcare sector are as follows:

- patient management system
- online appointment scheduling
- inpatient management
- billing and discharge summary
- laboratory and test management.
- Hospital Analytics
- Remote Patient Monitoring

Working is as follows:

For hospital management system, osplabs collects various information from DICOM Imaging modalities, central laboratory, patient data and pharmacy distribution and stores it in an enterprise data warehouse. Then by applying various data mining technologies the information is collectively stored in the form of electronic health records.

2.3 Requirements Analysis

At present, most of the medical related documents are stored in files. Hence it is tedious to maintain patient records both for patient and doctor. There are various issues pertaining in the system due to negligence of handling important documents. They are improper scientific evaluation and review of patients, delay in getting medical claims. Moreover, it even becomes difficult for the government to plan strategies for future medical care. Hence, it has become a necessity to store these documents in a digital format in order to curb these issues. Though EHR (Electronic Health Records) are present but there are no interlinks where the information related to patients can be shared from one hospital to another. When a user switches doctors due to travel or any other issue it will be easier for the user to share their entire medical history with the help of digital documents. Also, our project supports the ‘Digital India’ initiative which aims to transform India into a digitally empowered society and knowledge economy. We also plan to add specialized newsfeed for medical practitioners which will give them current medical-related news from all over the globe in a concise manner. In pandemic-like situations, records can be maintained precisely in our system.

System Design

In the project “Virtual Manager for Medical Records” various system diagrams have been created in order to define the structure of the project. Architecture Diagram represents the various principles, elements and components used in the system. Flow chart represents the flow or set of dynamic relationships in a system.

3.1 Architecture Diagram/ block diagram

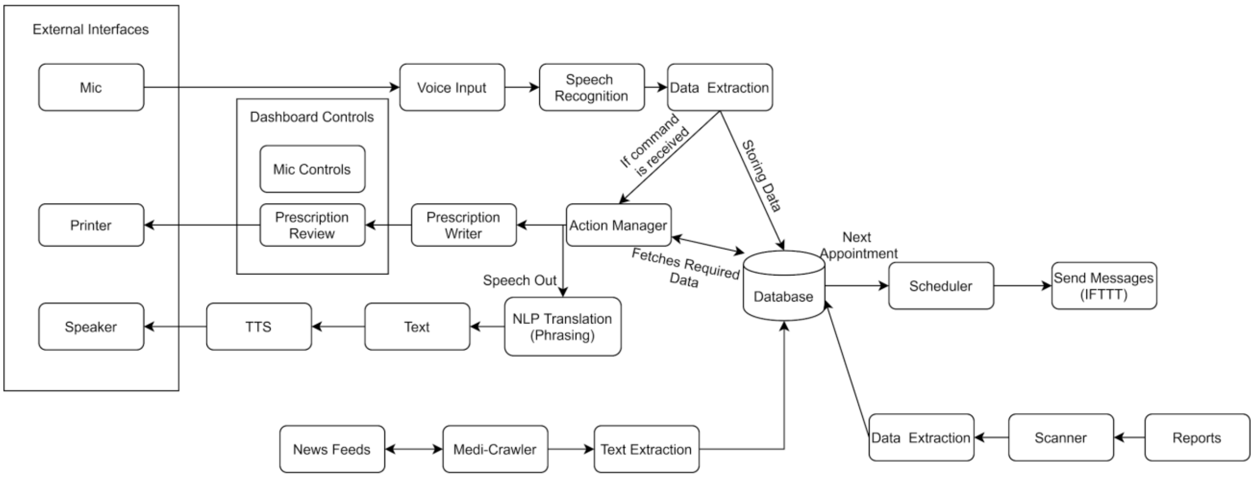


Fig 3.1.1- Architecture Diagram/Block Diagram

Virtual Assistant is designed with aim easing operations of doctors. There are three external interfaces to this mic, for taking input of conversations and commands. Printer for Printing prescriptions. Speaker/Headphone for listening to output.
Mic gets voice signals as input to speech recognizer which converts speech to machine readable format forwards to data extractor. Data extractor recognises and extracts required textual data. In case of a command the request is forwarded to the action manager. Regular conversation data is summarized and stored in data for future reference.
Action manager queues all the commands and is developed with specialization in two domains - writing prescription and responding doctor with speech output.
Scheduler reads appointments and sends reminder/confirmation messages to patients regarding appointments. A report scanner is developed for scanning and storing highlighted results to the database. Finally, for updating doctors about the happenings (only medical news) all around the world a web crawler is developed providing live and accurate news feeds to them. eg: Vaccine development news, disease occurrences, etc.

3.2 Flow chart

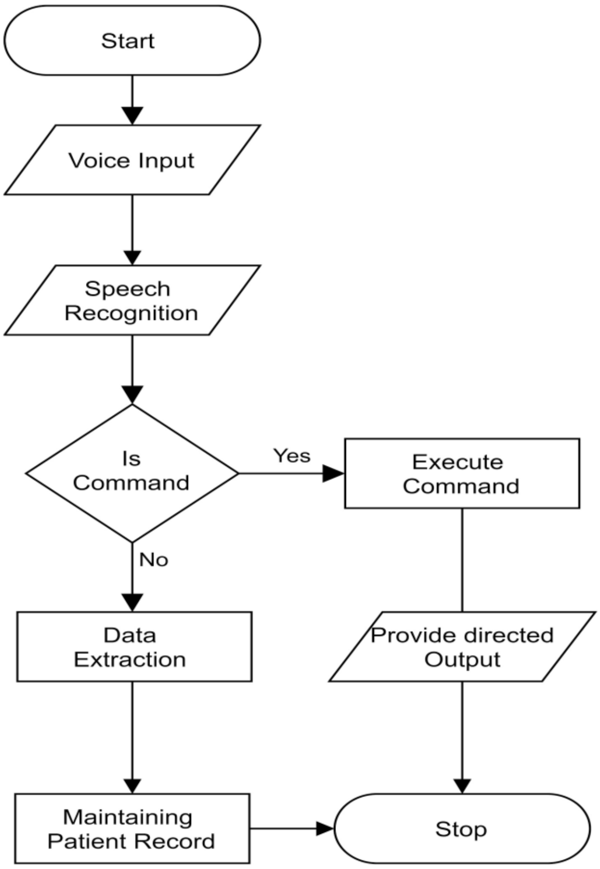


Fig 3.2.1-Flowchart

The flowchart defines the workflow of the project. Our project ‘Virtual Manager for Medical Records’, starts with receiving the input as external conversation. This feed is given to Speech Recogniser to convert the provided output to text or machine understandable format. Then the machine understands the input and categorises if it’s an input or not. In case of input required actions are performed and output is generated. If it is a conversation then details are summarized into a record register and prescription is generated.

Implementation details

4.1 System Requirements

- **Software**
  1. Python programming language and supporting Library - tika, gTTS, Speech-Recognition, etc.
  2. Natural Language Processing
  3. Basic Web Development Technologies - HTML, CSS, JavaScript.
- **Hardware**
  1. Operating System (Windows 7+, Ubuntu)
  2. RAM - 8GB
  3. Built-in Mic

4.2 Solution Approach/ Methodology

The system focuses on the development of a complete assistant for doctors, to ease their working time. The system will be developed using python and dependent libraries like NLTK, speech recognition, etc. The project will provide a dashboard (Web interface) for controlling the system functionalities - the activity of mic and finalizing prescription. The recorded voice would be classified by a Long Short-term Memory (LSTM) network model. The required text will be extracted and added to the database. Uniquely identified by patient ID. All pathology reports will be also passed through a data extractor and important details will be saved in the database for future reference. On the next scheduled appointment, the patient will receive a text reminder through telegram using If this then that (IFTTT), a freeware web-based service. Furthermore, the doctor will be provided with the patient's profile for prescribing drugs or medicines. Spoken prescription is recorded and typed by the system and left for the doctor's review. To update doctors with news alerts in the medical domain, a specialized web crawler is devised for the purpose of extracting information about latest happenings and displaying them as news feeds on the dashboard.

4.3 Modules

1. Voice Identifier & Data Extractor
   This is the first entry point of the system; an external peripheral microphone is used to take audio input signals. The input is processed by speech recognizer and converted to text/machine understandable format, which is further analysed by data extractor. Data extractor components classify input into command or conversation. In case of conversation the data is summarized and stored for future reference.

2. Prescription Generator
   This module is activated on receiving a trigger for writing a prescription. Spoken prescription is written by the system and displayed on Dashboard for doctor's review.

3. Appointment Reminder
   This maintains a calendar of appointments, it provides a reminder message to patients regarding their appointment, a day prior. If This Then That (IFTTT) is used for sending reminder messages to patients.

4. Report Scanner
   All pathology lab reports are in PDF formats, for maintaining the profile important data will be stored for future reference. This will virtually maintain a file of lab reports patient wise and could be heard on demand.

5. Responder
   This module replies to the doctor on receiving command. The output could be for various types - Patient history, Reading out schedule, etc. Helps in providing hands-free updates. Natural Language Processing is used for phrasing the generating statement on data points fetched from the database. A python library - NLTK is used as core of this module. The generated text statement is converted into speech and given output through speakers.

6. News feed
   Web-crawlers are used to crawl news websites for the latest news updates in the medical domain. News web pages will be scrapped for news articles tagged with medical and updated as notification on the dashboard. E.g.: News updates on spread of corona, reports on vaccinations, etc.

Experimental Results

Data Extraction Module was implemented. In this module, data from medical reports in a pdf format was extracted and then various parameters of blood reports were stored effectively.

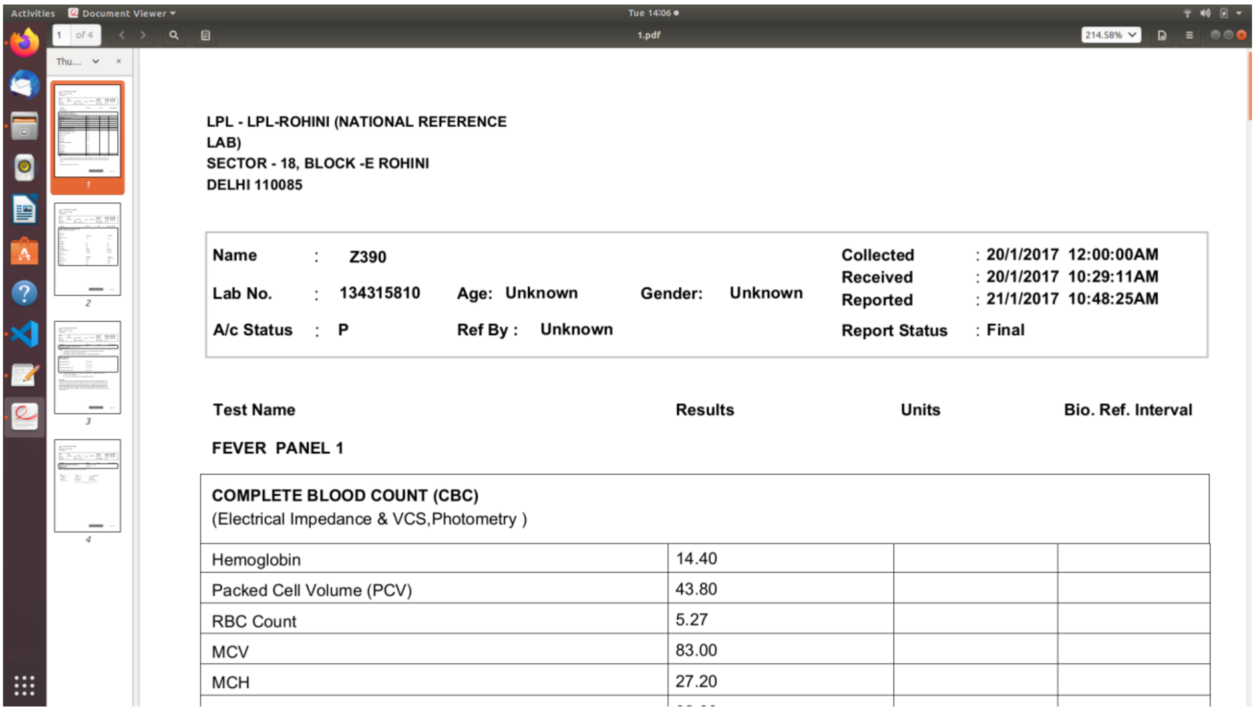
5.1 Results

We implemented the data extraction module. The data from reports (in PDF) was migrated in python. In three steps:

1. PDF was read and raw data was extracted and flush in a txt file (as shown in fig 5.2.2).
2. Tabula was used to extract tables from the PDF and obtained table data was stored.
3. The obtained data was converted into JSON format and extracted for individual data fields and its information as shown in fig 5.2.4

5.2 Screenshots of implementation

This image shows a regular report PDF of a person. Before sending to the data extractor.



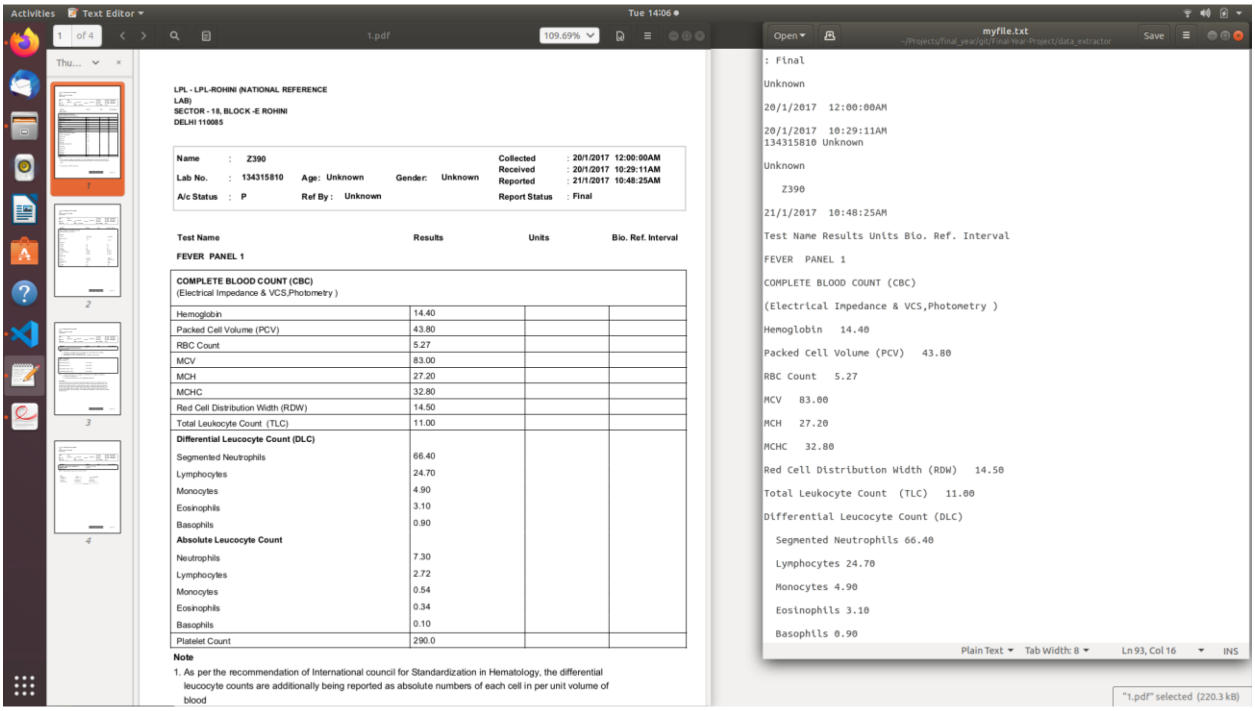
LPL - LPL-ROHINI (NATIONAL REFERENCE
LAB)
SECTOR - 18, BLOCK -E ROHINI
DELHI 110085

Name	:	Z390			Collected	:	20/1/2017 12:00:00AM
					Received	:	20/1/2017 10:29:11AM
Lab No.	:	134315810	Age: Unknown	Gender: Unknown	Reported	:	21/1/2017 10:48:25AM
A/c Status	:	P	Ref By : Unknown		Report Status	:	Final

Test Name	Results	Units	Bio. Ref. Interval
FEVER PANEL 1			
COMPLETE BLOOD COUNT (CBC) (Electrical Impedance & VCS,Photometry)			
Hemoglobin	14.40		
Packed Cell Volume (PCV)	43.80		
RBC Count	5.27		
MCV	83.00		
MCH	27.20		

Fig:5.2.1 Patient report

The data extractor, extracts the raw data texts in a file (this is a temporary file). The raw file has only the text extracted from PDF which is irregularly arranged and filling blanks as unknown.



LPL - LPL-ROHINI (NATIONAL REFERENCE
LAB)
SECTOR - 18, BLOCK -E ROHINI
DELHI 110085

Name : Z390
Lab No. : 134315810 Age: Unknown Gender: Unknown
A/c Status : P Ref By : Unknown
Collected : 20/1/2017 12:00:00AM
Received : 20/1/2017 10:29:11AM
Reported : 21/1/2017 10:48:25AM
Report Status : Final

Test Name	Results	Units	Bio. Ref. Interval
FEVER PANEL 1			
COMPLETE BLOOD COUNT (CBC) (Electrical Impedance & VCS,Photometry)			
Hemoglobin	14.40		
Packed Cell Volume (PCV)	43.80		
RBC Count	5.27		
MCV	83.00		
MCH	27.20		
MCHC	32.80		
Red Cell Distribution Width (RDW)	14.50		
Total Leukocyte Count (TLC)	11.00		
Differential Leucocyte Count (DLC)			
Segmented Neutrophils	66.40		
Lymphocytes	24.70		
Monocytes	4.90		
Eosinophils	3.10		
Basophils	0.90		
Absolute Leucocyte Count			
Neutrophils	7.30		
Lymphocytes	2.72		
Monocytes	0.54		
Eosinophils	0.34		
Basophils	0.10		
Platelet Count	290.0		

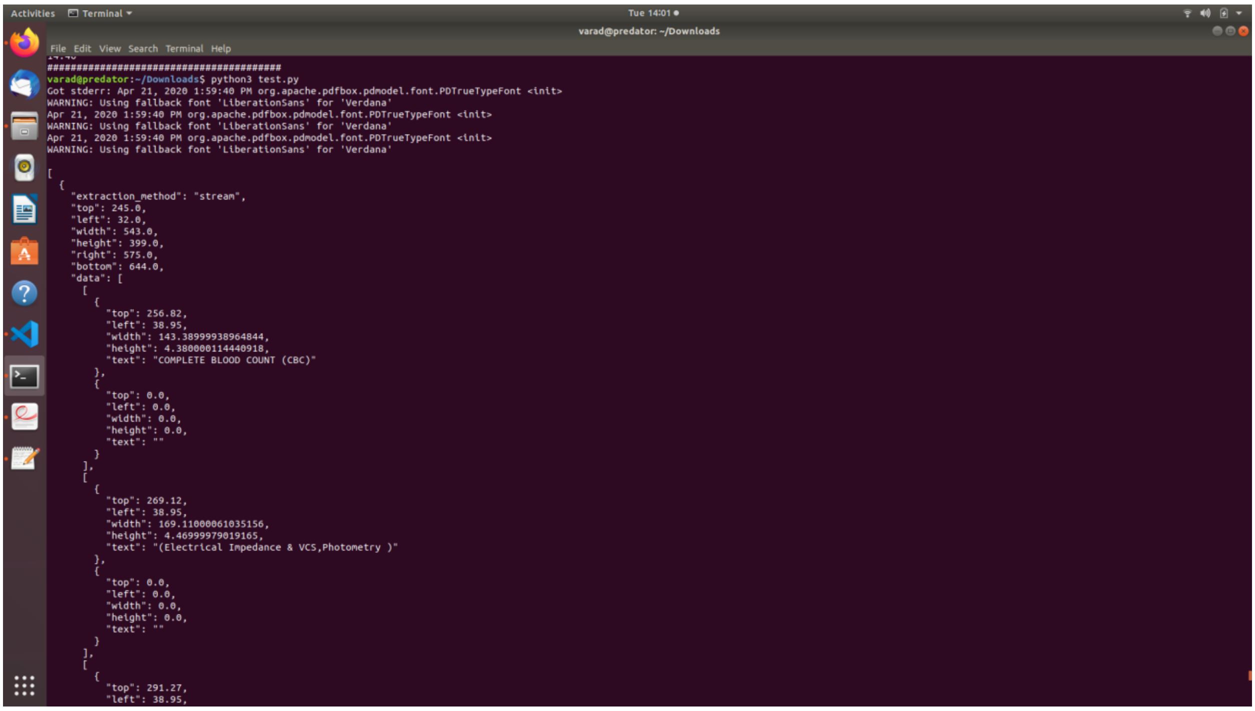
Note

1. As per the recommendation of International council for Standardization in Hematology, the differential leucocyte counts are additionally being reported as absolute numbers of each cell in per unit volume of blood

```
: Final

Unknown

20/1/2017  12:00:00AM

20/1/2017  10:29:11AM
134315810 Unknown

Unknown

   Z390

21/1/2017  10:48:25AM

Test Name Results Units Bio. Ref. Interval

FEVER  PANEL 1

COMPLETE BLOOD COUNT (CBC)

(Electrical Impedance & VCS,Photometry )

Hemoglobin   14.40

Packed Cell Volume (PCV)   43.80

RBC Count   5.27

MCV   83.00

MCH   27.20

MCHC   32.80

Red Cell Distribution Width (RDW)   14.50

Total Leukocyte Count  (TLC)   11.00

Differential Leucocyte Count (DLC)

  Segmented Neutrophils 66.40

  Lymphocytes 24.70

  Monocytes 4.90

  Eosinophils 3.10

  Basophils 0.90
```

Fig:5.2.2 After extracting report to raw text file

Using python libraries - pandas and tabula helps to extract the tabular data. The data is displayed in JSON with location in the page (indentations - right, left, top) and data fields. These are useful for mapping particulars to its observed values.

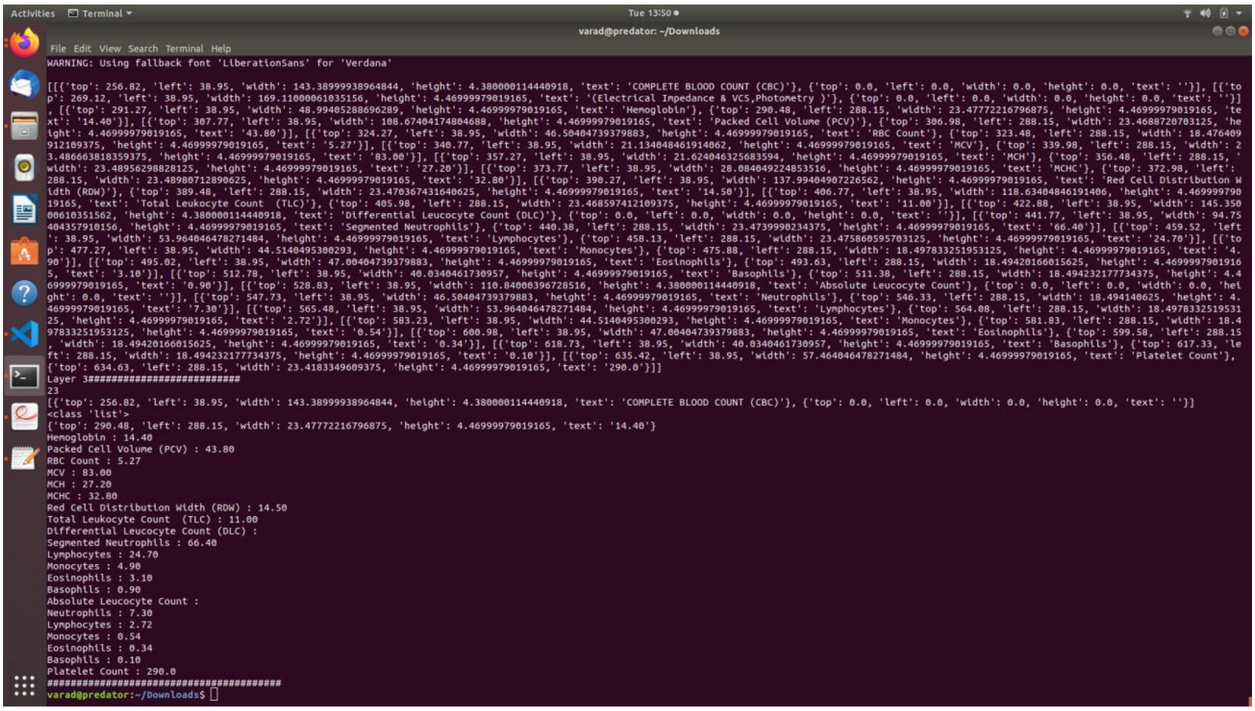


Activities
Terminal
Tue 14:01
varad@predator: ~/Downloads
File Edit View Search Terminal Help

```
14.40
########################################
varad@predator:~/Downloads$ python3 test.py
Got stderr: Apr 21, 2020 1:59:40 PM org.apache.pdfbox.pdmodel.font.PDTrueTypeFont <init>
WARNING: Using fallback font 'LiberationSans' for 'Verdana'
Apr 21, 2020 1:59:40 PM org.apache.pdfbox.pdmodel.font.PDTrueTypeFont <init>
WARNING: Using fallback font 'LiberationSans' for 'Verdana'
Apr 21, 2020 1:59:40 PM org.apache.pdfbox.pdmodel.font.PDTrueTypeFont <init>
WARNING: Using fallback font 'LiberationSans' for 'Verdana'

[
  {
    "extraction_method": "stream",
    "top": 245.0,
    "left": 32.0,
    "width": 543.0,
    "height": 399.0,
    "right": 575.0,
    "bottom": 644.0,
    "data": [
      [
        {
          "top": 256.82,
          "left": 38.95,
          "width": 143.38999938964844,
          "height": 4.380000114440918,
          "text": "COMPLETE BLOOD COUNT (CBC)"
        },
        {
          "top": 0.0,
          "left": 0.0,
          "width": 0.0,
          "height": 0.0,
          "text": ""
        }
      ],
      [
        {
          "top": 269.12,
          "left": 38.95,
          "width": 169.11000061035156,
          "height": 4.46999979019165,
          "text": "(Electrical Impedance & VCS,Photometry )"
        },
        {
          "top": 0.0,
          "left": 0.0,
          "width": 0.0,
          "height": 0.0,
          "text": ""
        }
      ],
      [
        {
          "top": 291.27,
          "left": 38.95,
```

Fig:5.2.3 Table Data conversion to JSON format

After processing the table data of particulars were extracted and mapped to its respective values, as shown in the figure below.


Activities
Terminal
Tue 13:50
varad@predator: ~/Downloads
File Edit View Search Terminal Help

```
WARNING: Using fallback font 'LiberationSans' for 'Verdana'

[[{'top': 256.82, 'left': 38.95, 'width': 143.38999938964844, 'height': 4.380000114440918, 'text': 'COMPLETE BLOOD COUNT (CBC)'}, {'top': 0.0, 'left': 0.0, 'width': 0.0, 'height': 0.0, 'text': ''}], [{'to
p': 269.12, 'left': 38.95, 'width': 169.11000061035156, 'height': 4.46999979019165, 'text': '(Electrical Impedance & VCS,Photometry )'}, {'top': 0.0, 'left': 0.0, 'width': 0.0, 'height': 0.0, 'text': ''}]
, [{'top': 291.27, 'left': 38.95, 'width': 48.99405288696289, 'height': 4.46999979019165, 'text': 'Hemoglobin'}, {'top': 290.48, 'left': 288.15, 'width': 23.47772216796875, 'height': 4.46999979019165, 'te
xt': '14.40'}], [{'top': 307.77, 'left': 38.95, 'width': 108.67404174804688, 'height': 4.46999979019165, 'text': 'Packed Cell Volume (PCV)'}, {'top': 306.98, 'left': 288.15, 'width': 23.4688720703125, 'he
ight': 4.46999979019165, 'text': '43.80'}], [{'top': 324.27, 'left': 38.95, 'width': 46.50404739379883, 'height': 4.46999979019165, 'text': 'RBC Count'}, {'top': 323.48, 'left': 288.15, 'width': 18.476409
912109375, 'height': 4.46999979019165, 'text': '5.27'}], [{'top': 340.77, 'left': 38.95, 'width': 21.134048461914062, 'height': 4.46999979019165, 'text': 'MCV'}, {'top': 339.98, 'left': 288.15, 'width': 2
3.486663818359375, 'height': 4.46999979019165, 'text': '83.00'}], [{'top': 357.27, 'left': 38.95, 'width': 21.624046325683594, 'height': 4.46999979019165, 'text': 'MCH'}, {'top': 356.48, 'left': 288.15, '
width': 23.48956298828125, 'height': 4.46999979019165, 'text': '27.20'}], [{'top': 373.77, 'left': 38.95, 'width': 28.084049224853516, 'height': 4.46999979019165, 'text': 'MCHC'}, {'top': 372.98, 'left':
288.15, 'width': 23.48980712890625, 'height': 4.46999979019165, 'text': '32.80'}], [{'top': 390.27, 'left': 38.95, 'width': 137.99404907226562, 'height': 4.46999979019165, 'text': 'Red Cell Distribution W
idth (RDW)'}, {'top': 389.48, 'left': 288.15, 'width': 23.470367431640625, 'height': 4.46999979019165, 'text': '14.50'}], [{'top': 406.77, 'left': 38.95, 'width': 118.63404846191406, 'height': 4.469999790
19165, 'text': 'Total Leukocyte Count  (TLC)'}, {'top': 405.98, 'left': 288.15, 'width': 23.468597412109375, 'height': 4.46999979019165, 'text': '11.00'}], [{'top': 422.88, 'left': 38.95, 'width': 145.350
00610351562, 'height': 4.380000114440918, 'text': 'Differential Leucocyte Count (DLC)'}, {'top': 0.0, 'left': 0.0, 'width': 0.0, 'height': 0.0, 'text': ''}], [{'top': 441.77, 'left': 38.95, 'width': 94.75
404357910156, 'height': 4.46999979019165, 'text': 'Segmented Neutrophils'}, {'top': 440.38, 'left': 288.15, 'width': 23.4739990234375, 'height': 4.46999979019165, 'text': '66.40'}], [{'top': 459.52, 'left
': 38.95, 'width': 53.964046478271484, 'height': 4.46999979019165, 'text': 'Lymphocytes'}, {'top': 458.13, 'left': 288.15, 'width': 23.475860595703125, 'height': 4.46999979019165, 'text': '24.70'}], [{'to
p': 477.27, 'left': 38.95, 'width': 44.5140495300293, 'height': 4.46999979019165, 'text': 'Monocytes'}, {'top': 475.88, 'left': 288.15, 'width': 18.497833251953125, 'height': 4.46999979019165, 'text': '4.
90'}], [{'top': 495.02, 'left': 38.95, 'width': 47.00404739379883, 'height': 4.46999979019165, 'text': 'Eosinophils'}, {'top': 493.63, 'left': 288.15, 'width': 18.49420166015625, 'height': 4.4699997901916
5, 'text': '3.10'}], [{'top': 512.78, 'left': 38.95, 'width': 40.0340461730957, 'height': 4.46999979019165, 'text': 'Basophils'}, {'top': 511.38, 'left': 288.15, 'width': 18.494232177734375, 'height': 4.4
6999979019165, 'text': '0.90'}], [{'top': 528.83, 'left': 38.95, 'width': 110.84000396728516, 'height': 4.380000114440918, 'text': 'Absolute Leucocyte Count'}, {'top': 0.0, 'left': 0.0, 'width': 0.0, 'hei
ght': 0.0, 'text': ''}], [{'top': 547.73, 'left': 38.95, 'width': 46.50404739379883, 'height': 4.46999979019165, 'text': 'Neutrophils'}, {'top': 546.33, 'left': 288.15, 'width': 18.494140625, 'height': 4.
46999979019165, 'text': '7.30'}], [{'top': 565.48, 'left': 38.95, 'width': 53.964046478271484, 'height': 4.46999979019165, 'text': 'Lymphocytes'}, {'top': 564.08, 'left': 288.15, 'width': 18.4978332519531
25, 'height': 4.46999979019165, 'text': '2.72'}], [{'top': 583.23, 'left': 38.95, 'width': 44.5140495300293, 'height': 4.46999979019165, 'text': 'Monocytes'}, {'top': 581.83, 'left': 288.15, 'width': 18.4
97833251953125, 'height': 4.46999979019165, 'text': '0.54'}], [{'top': 600.98, 'left': 38.95, 'width': 47.00404739379883, 'height': 4.46999979019165, 'text': 'Eosinophils'}, {'top': 599.58, 'left': 288.15
, 'width': 18.49420166015625, 'height': 4.46999979019165, 'text': '0.34'}], [{'top': 618.73, 'left': 38.95, 'width': 40.0340461730957, 'height': 4.46999979019165, 'text': 'Basophils'}, {'top': 617.33, 'le
ft': 288.15, 'width': 18.494232177734375, 'height': 4.46999979019165, 'text': '0.10'}], [{'top': 635.42, 'left': 38.95, 'width': 57.464046478271484, 'height': 4.46999979019165, 'text': 'Platelet Count'},
{'top': 634.63, 'left': 288.15, 'width': 23.4183349609375, 'height': 4.46999979019165, 'text': '290.0'}]]
Layer 3###########################
23
[{'top': 256.82, 'left': 38.95, 'width': 143.38999938964844, 'height': 4.380000114440918, 'text': 'COMPLETE BLOOD COUNT (CBC)'}, {'top': 0.0, 'left': 0.0, 'width': 0.0, 'height': 0.0, 'text': ''}]
<class 'list'>
{'top': 290.48, 'left': 288.15, 'width': 23.47772216796875, 'height': 4.46999979019165, 'text': '14.40'}
Hemoglobin : 14.40
Packed Cell Volume (PCV) : 43.80
RBC Count : 5.27
MCV : 83.00
MCH : 27.20
MCHC : 32.80
Red Cell Distribution Width (RDW) : 14.50
Total Leukocyte Count  (TLC) : 11.00
Differential Leucocyte Count (DLC) :
Segmented Neutrophils : 66.40
Lymphocytes : 24.70
Monocytes : 4.90
Eosinophils : 3.10
Basophils : 0.90
Absolute Leucocyte Count :
Neutrophils : 7.30
Lymphocytes : 2.72
Monocytes : 0.54
Eosinophils : 0.34
Basophils : 0.10
Platelet Count : 290.0
########################################
varad@predator:~/Downloads$ 
```

Fig:5.2.4 Final output assigned actual numbers to respective blood count

Conclusion and Future Scope

6.1 Conclusion

In a nutshell, the development of the system will be a big boon to doctors in doing physical and needless work. The system is being built by using open source technologies, which are easily portable and easier to set up on various machines. The system has a dashboard (Web interface) for controlling the system functionalities - the activity of mic and finalizing prescription. The recorded voice would be classified by a Long Short-term Memory (LSTM) network model. Conversations are summarized and stored in the database for future reference. The system is supported with a report scanner and appointment reminder for scanning pathology reports and reminding patients regarding appointments using IFTTT respectively. After meeting, the prescription is auto-generated. This system will help the government to formulate various strategies regarding future health care. It will also help medical insurance companies to keep a proper record in order to provide patient's demand for medical expenses. Since records can be retrieved easily with our system it will help doctors to evaluate a patient's health and plan treatment accordingly. Moreover, it also supports the government initiative "Digital India". A medi-crawler is employed to provide news feeds about the medical domain. The latest news regarding the medical domain will be updated on the dashboard.

6.2 Future Scope

The developed system could be kept as a baseline for personal assistants for doctors.
Moreover, the system has a huge potential to expand itself in the market in the following areas:

- Appointment Booking system.
- News-Feeds for medical-related information around the globe.
- Data Analysis of Disease occurrence and patients.
- Interpreting verbal conversations with the use of a camera.
- Multi-lingual support for extracting data from every interaction.
- Reading and spotting errors in X-Ray & MRI scans, etc

References

1. J Maxillofac Oral Surg“Management for Medical records”, Sept,2011.
2. SHERIN HIJAZI 1,3, NADIM OBEID 2, AND KHAIR EDDIN SABRI 3 “The Logical Foundation of a Personalized Medical Prescription System”, December 31, 2019.
3. Chumsak Sibunruang and Jantima Polpinij, “Finding Clinical Knowledge from MEDLINE Abstracts by Text Summarization Technique.”,2018.
4. Uma M, Sneha V, Sneha G, BHuvana J, Bharathi B, “Formation of SQL from Natural Language query using NLP”, 2019.
5. ZHIXIN LI 1, ZHI PENG1, SUQIN TANG1, CANLONG ZHANG1, AND HUIFANG MA “Text Summarization Based on Double-Attention Pointer Network”, January 10, 2020.
6. Kavita Lohani & Debajyoti Mukhopadhyay “Reduction of Medication Errors while Prescribing Using Evidence Based Treatment” IEEE, 2017.
7. Milad Moradi “CIBS: A biomedical text summarizer using topic-based sentence clustering”, December, 2018.

Appendix A: Code Sample

Python Code

```
#Importing required libs
import pandas as pd
import tabula
import json

#Using tabula to read the PDF and extract tables from it.
df = tabula.read_pdf("1.pdf", pages = '1', multiple_tables = True, output_format="json")

#Tabula data table can be converted into JSON format. Making extraction easy.
data =json.loads(json.dumps(df))
# print (json.dumps(data,indent=2))

# Extracting Data field(table) in particular
for d in data:
    d1= (d["data"])
print (d1)

#For individual rows in table
d3 = json.loads(json.dumps(d1))
print ("Layer 3#########################")
print(len(d3))


print (d3[0])

# Individual value and data extractions as per our needs
a=json.loads(json.dumps(d3[2]))
print (type(a))
print (a[1])

#Printing extracted attributes

for i in range (2,23):
    a=json.loads(json.dumps(d3[i]))
    b=json.loads(json.dumps(a[0]))
    c=json.loads(json.dumps(a[1]))
    print (b["text"]+" : " + c["text"])
```

Acknowledgements

The making of the project “Virtual manager for medical records” involves the contribution of many people.

We owe a great thanks to many people who have helped and supported us during the course of our project. We express our gratitude to Dr. S. M. Khot, Principal of Fr. C. Rodrigues Institute of Technology, HOD in-charge Ms. Dhanashree Hadsul and mini project coordinators as Mrs. Lakshmi Gadhikar and Mrs. Kalpana Wani extending their inevitable and valuable support to us.

We would take this opportunity to thank our guide Mrs. Smita Rukhande for guiding and correcting various documents of ours with attention and care. This synopsis could not have been completed without their knowledge and expertise.

We would also thank our Institution and our faculty members without whom this project would have been a distant reality. We also extend our heartfelt thanks to our families and well-wishers.

Yours sincerely,

Sakshi Khose

Varad More

Sonal Sarode