Visvesvaraya Technological University
Belagavi-590 018, Karnataka

A MINI PROJECT REPORT ON

"DATA CONVERSION"

Mini Project Report submitted
in partial fulfillment of the requirement for The VI Semester File Structures Laboratory

[17CSL68]

Bachelor of Engineering
In
Information Science and Engineering

Submitted by
GAGAN.N.RAJ[1JT17IS010]

Department of Information Science and Engineering
Jyothy Institute of Technology
Tataguni, Bengaluru-560082

Jyothy Institute of Technology
Tataguni, Bengaluru-560082
Department of Information Science and Engineering

CERTIFICATE

Certified that the mini project work entitled **“Data Conversion”**carried out by **GAGAN.N.RAJ[1JT17IS010]** bonafide student of Jyothy Institute of Technology, in partial fulfillment for the award of **Bachelor of Engineering** in **Information Science and Engineering** department of the **Visvesvaraya Technological University, Belagavi** during the year **2020-2021**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

Prof.Vadiraja A
Guide,Asst.Professor
Dept. of ISE

Dr. Harshvardhan Tiwari
Associate Professor and HoD
Dept. OF ISE

External Viva Examiner

Signature with Date:

◆
□
◆ □

ACKNOWLEDGMENT

The satisfaction that accompanies the successful completion of my project Would be incomplete without mentioning the people who made it possible, I am thankful to **Dr. Gopalakrishna K**, Principal, JIT, Bangalore for Being kind enough to provide us an opportunity to work on a project in this Institution.

I am thankful to **Dr. Harshvardhan Tiwari** , HoD, Dept of ISE for his co-operation and encouragement at all moments of our approach and Whose constant guidance and encouragement crowns all the efforts with Success.

I would greatly mention enthusiastic influence provided by **Mr. Vadiraja A**, Asst. Professor, Dept. of ISE for his ideas and co-operation Showed on us during our venture and making this project a great success.

Finally, it’s pleasure and happiness to the friendly co-operation showed by all The staff members of Information Science Department, JIT.

GAGAN.N.RAJ
1JT17IS010

ABSTRACT

This project has been created using Spyder, with the platform Windows and language Python. The project title - “Data Conversion” is a process of converting one file format to another file format.

We come across many types of readable file formats. Sometimes there is a need of viewing a similar data in a different file format. For example, we have some data stored in a CSV file but it is inconvenient for a machine to read it easily. So we need to convert the same CSV to a machine readable language like JSON etc. so that no data is modified during the process. This is a very useful tool for general purposes as we can also convert the same JSON to CSV, which can be comprehended by the users easily.

In this project, the similar CSV to JSON conversion is offered alongside an easily understandable Graphical User Interface to the end user providing seamless conversions of the said file format.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

INTRODUCTION

1.1 Introduction to File Structures

In simple terms, a file is a collection of data stored on mass storage (e.g., disk or tape).But there is one important distinction that must be made at the outset when discussing file structures. And that is the difference between the logical and physical organization of the data.

On the whole a file structure will specify the logical structure of the data, that is the relationships that will exist between data items independently of the way in which these relationships may actually be realized within any computer. It is this logical aspect that we will concentrate on. The physical organization is much more concerned with optimizing the use of the storage medium when a particular logical structure is stored on, or in it. Typically for every unit of physical store there will be a number of units of the logical structure (probably records) to be stored in it.

For example, if we were to store a tree structure on a magnetic disk, the physical organization would be concerned with the best way of packing the nodes of the tree on the disk given the access characteristics of the disk.

Like all subjects in computer science the terminology of file structures has evolved higgledy-piggledy without much concern for consistency, ambiguity, or whether it was possible to make the kind of distinctions that were important.

It was only much later that the need for a well-defined, unambiguous language to describe file structures became apparent. In particular, there arose a need to communicate ideas about file structures without getting bogged down by hardware considerations.

1.2 Introduction to File System

In computing, a file system or file system controls how data is stored and retrieved. Without a file system, information placed in a storage medium would be one large body of data with noway to tell where one piece of information stops and the next begins. By separating the data into pieces and giving each piece a name, the information is easily isolated and identified. Taking its name from the way paper-based information systems are named, each groups of data is called a “file”. The structure and logic rules used to manage the groups of information and their names is called a “file system”.

There are many different kinds of file systems. Each one has different structure and logic, properties of speed, flexibility, security, size and more. Some file systems have been designed to be used for specific applications. For example, the ISO 9660 file system is designed specifically for optical discs.
File systems can be used on numerous different types of storage devices that use different kinds of media. The most common storage device in use today is a hard disk drive. Other kinds of media that are used include flash memory, magnetic tapes, and optical discs. In some cases, such as with tmpfs, the computer's main memory (random-access memory, RAM) is used to create a temporary file system for short-term use.

Some file systems are used on local data storage devices; others provide file access via a network protocol(for example, NFS, SMB, or 9P clients). Some file systems are “virtual”. Meaning that the supplied “files” (called virtual files) are computed on request (e.g. procfs) or are merely a mapping into a different file system used as a blacking store. The file system manages access to both the content of files and the metadata about those files. It is responsible for arranging storage space; reliability, efficiency, and tuning with regard to the physical storage medium are important design considerations .

1.3 Data Conversion

Throughout a computer environment, data is encoded in a variety of ways. For example, computer hardware is built on the basis of certain standards, which requires that data contains, for example, parity bit checks. Similarly, the operating system is predicated on certain standards for data and file handling. Furthermore, each computer program handles data in a different manner. Whenever any one of these variable is changed, data must be converted in some way before it can be used by a different computer, operating system or program.

Data Conversion is the process of translating data from one format to another. While the concept itself may seem simple, data conversion is a critical step in the process of data integration. This step enables the data to be read, altered, and executed in an application or database other than that in which it was created. A typical example of data conversion is the way in which many of us consume entertainment. In order to play a music or video file on a mobile device, the file needs to be converted from its source format (such as an MKV file) to one that can be read by the device (such as an MP4 file).

The goal of data conversion is to prevent data loss or corruption by maintaining the integrity of the data and embedded structures. This can be done easily if the destination format supports the same features and data structures as the source data. If the source formatting is not supported, however, businesses must correctly and comprehensively convert the format and structure to read, modify, and analyze data.

Data conversion is only possible if the target format is able to support the same data features and constructs of the source data. If the format specifications are not known, reverse engineering can be used to convert the data. In most cases, this would only lead to a close approximation of the original specification. In data conversion, information can easily be removed by application or conversion agent. However adding the information is an uphill task. Data conversion is performed in most cases in a rule-based fashion. The rules can be established by the operating system, application, the programming language or even by the programmer. Successful data conversion helps in the smooth functionality of applications dependent on the concerned data.

However, data conversion can be a painstaking task, especially in the case of complex data formats. If not done properly, it could also lead to loss of information. Data conversion can be a complex process, though it doesn't have to be. Tools for automating the process can improve both the accuracy and completeness of the converted data, while reducing development time. The basic steps that most data conversions incorporate are as follows:

- A comprehensive plan is developed based on user requirements.

- The character/symbol set is extracted from its source.

- That source data is converted to the format of the destination.

- The data is reviewed and loaded to the target system.

These basic steps vary based on several different factors. One important factor of data conversion is whether the source data type is converted to another data type or whether it is only reinterpreted as another data type.

Another aspect is whether it is implicit or explicit. With implicit conversion, a compiler automatically performs the conversion. This process is done by comparing one data type to another and then assigning the source data type to the proper destination data type. With explicit conversion, objects and data types are converted in one of three ways:

-A run \time check is performed prior to the conversion to determine if the destination data type can hold the source value. If it cannot, an error occurs.

-No checks are performed. If the destination data type cannot hold the source value, no error occurs. Rather, the resulting data type is left undefined. The raw bit pattern is copied without data being interpreted.

Each programming language uses its unique set of instructions for how the data types are converted. Strong-typing languages, which have stricter rules at compile time, typically use the explicit methodology to convert the data types. Weak-typing languages, which have looser rules and may produce unpredictable results, are more likely to arbitrarily interpret a data type as having different representations.

CHAPTER 2
REQUIREMENT ANALYSIS AND DESIGN

REQUIREMENT ANALYSIS AND DESIGN

2.1 Domain Understanding

The main objective of this project is to convert a given file. The outcome of this project is to ease the user for converting files to suitable formats with an easily understandable Graphical User Interface.

2.2 Classification of Requirements

System Requirements:

- Processor - Intel® Core™ i7 processor
- Operating Systems - Windows 7 or higher, Linux
- Python Version - 3.7.x or higher
- RAM - 4GB
- ROM - 75GB

Software and Hardware Requirements:

- Hard Disk: 4GB Hard Disk(Minimum)
- RAM: 75GB or more
- Python IDE
- pandas, tkinter packages

Programming Languages:

Primary Implementation - Python

Secondary Implementation - Front End : tkinter Back End : Python

System Analysis

There are basically two models built to constitute the entire project. The primary implementation shows how a conversion is done by displaying the final form of the converted file in the console. This is done to show how the conversions are usually carried out in the ETL Tools.

The secondary implementation is an extension of the primary implementation with a Graphical User Interface provided to the user to show how the large scale implementations of these ETL functions are done. A suitable input file can be chosen via GUI and once the name of output file alongside its directory is given, the CSV file will be converted to JSON and stored in the given directory.

CHAPTER 3
IMPLEMENTATION

IMPLEMENTATION

3.1 Primary Implementation:

The main conversion process/procedure is shown in this implementation. The CSV, Time libraries are imported to view the file and calculate time respectively. The CSV module implements classes to read and write tabular data in CSV format. It allows programmers to say, “write this data in the format preferred by Excel,” or “read data from this file which was generated by Excel,” without knowing the precise details of the CSV format used by Excel. Programmers can also describe the CSV formats understood by other applications or define their own special-purpose CSV formats. The pprint module provides a capability to “pretty-print” arbitrary Python data structures in a form which can be used as input to the interpreter. If the formatted structures include objects which are not fundamental Python types, the representation may not be loadable. This may be the case if objects such as files, sockets or classes are included, as well as many other objects which are not representable as Python literals. The timeit module provides a simple way to time small bits of Python code. It has both a Command-Line Interface as well as a callable one. It avoids a number of common traps for measuring execution times.

When the command is given to execute the program , the user has to enter the file location upon prompt. The time at that instance is recorded. The CSV file is read using csv.Dictreader. It creates an object which operates like a regular reader but maps the information read into a dict whose keys are given by the optional fieldnames parameter. The fieldnames parameter is a sequence whose elements are associated with the fields of the input data in order. These elements become the keys of the resulting dictionary.

If the fieldnames parameter is omitted, the values in the first row of the file f will be used as the fieldnames. If the row read has more fields than the fieldnames sequence, the remaining data is added as a sequence keyed by the value of restkey. If the row read has fewer fields than the fieldnames sequence, the remaining keys take the value of the optional restval parameter. Any other optional or keyword arguments are passed to the underlying reader instance. And for every line read, the ends are given '{' and '}' and field names and values are defined with ':' respectively. The time taken for the conversion is calculated as the difference between the instances at the points where the writing of JSON is completed and when the CSV file is imported for reading. The time taken is directly proportional to the size of the file. All the required output i.e, the JSON format and total time taken for the conversion are printed in the Terminal/Command Prompt where the same program is executed.

3.2 Secondary Implementation:

This implementation involves various built in functions to demonstrate the large scale implementation of File Conversions. One of the main libraries used are pandas.

pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open source data analysis / manipulation tool available in any language. It is already well on its way toward this goal. It is well suited for many different kinds of data:

- Tabular data with heterogeneously-typed columns, as in an SQL table or Excel spreadsheet

- Ordered and unordered (not necessarily fixed-frequency) time series data.

- Arbitrary matrix data (homogeneously typed or heterogeneous) with row and column labels
- Any other form of observational / statistical data sets. The data actually need not be labeled at all to be placed into a pandas data structure

The Graphical User Interface is built using Python's library called Tkinter. The tkinter package ("Tk interface") is the standard Python interface to the Tk GUI toolkit. Both Tk and tkinter are available on most Unix platforms, as well as on Windows systems. Tk itself is not part of Python; it is maintained at ActiveState(Open Source Languages Company). The following modules are imported from tkinter:

tkinter.filedialog
Common dialogs to allow the user to specify a file to open or save.

tkinter.messagebox
Access to standard Tk dialog boxes.

Initially, the whole layout of the Graphical User Interface is designed by setting it's dimensional parameters and the character parameters such as Title, background colour, text font etc. The basic functionalities' interfaces are created alongside the functions it should execute when it is selected during its execution. Each buttons created are assigned to specific functions.
There are three functions which constitute the whole program. One to fetch the CSV file, one to convert to JSON format and last one to exit the GUI.
In the getCSV() function, the user is asked to fetch a CSV file from the system manually. After this, with the help of pandas library, the imported file is read and stored in a

predefined global variable called ‘read_file’. This variable now contains the entire data of the imported CSV file. The time instances before and after reading the file is observed and recorded for the final calculation.

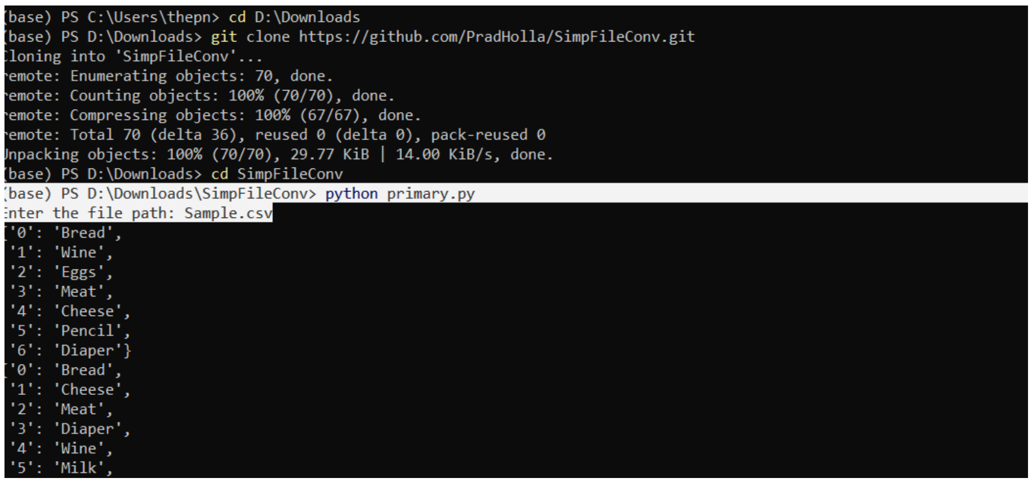
The second function is to convert the imported CSV file into a JSON file. In this function, a prompt appears asking the user to name the output file and also to define its location with the default extension of the said file set to .json format. Now, with the help of the global variable ‘read_file’, the entire CSV data is written to the new file with the help of pandas module called .to_json with the indentaion set to 4. Now, the new file contains the CSV file’s data in JSON format. Again, the time instances before and after writing the data to JSON file are recorded and subtracted. The same is done with the instances recorded in getCSV() function. Now, the two new values obtained are added in order to get the total time taken for the entire conversion I.e, the time taken for reading the CSV data and writing the said same to the new JSON file. The final time taken for conversion is then displayed in the Terminal/Command Prompt after the conversion.

The third minor function is created to exit the GUI. This basically contains the parameters of the quit box, and function assigned to it. After the button is clicked, the user gets a prompt whether to exit the application or not. All these basic features conclude the whole function.

CHAPTER 4
RESULT AND ANALYSIS

RESULT AND ANALYSIS

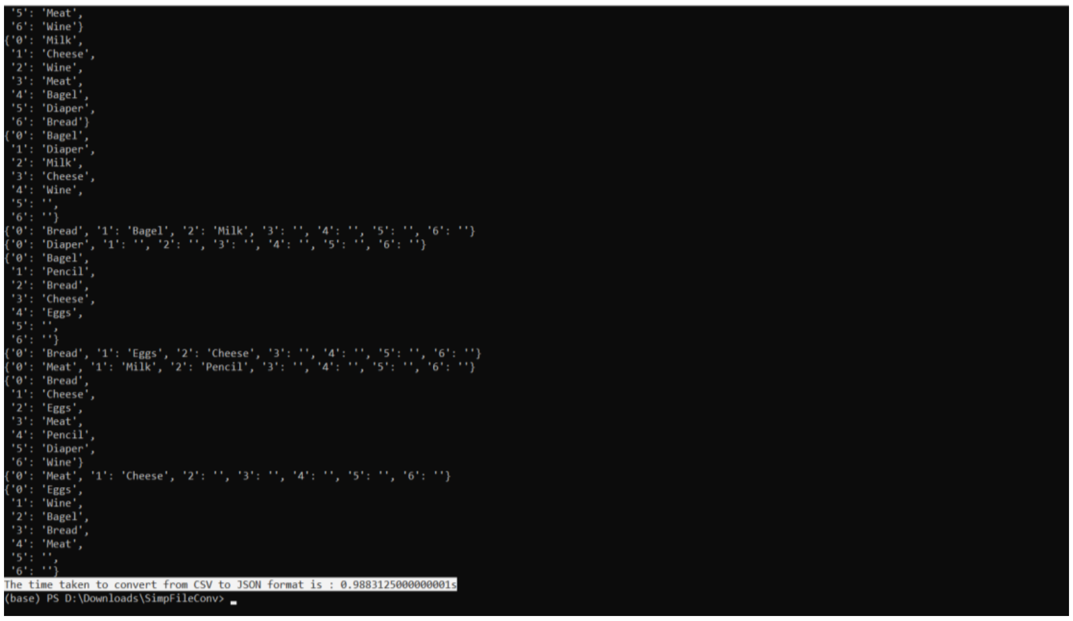
4.1.1 Primary Implementation in Windows Machine:



```
(base) PS C:\Users\thepn> cd D:\Downloads
(base) PS D:\Downloads> git clone https://github.com/PradHolla/SimpFileConv.git
loning into 'SimpFileConv'...
emote: Enumerating objects: 70, done.
emote: Counting objects: 100% (70/70), done.
emote: Compressing objects: 100% (67/67), done.
emote: Total 70 (delta 36), reused 0 (delta 0), pack-reused 0
npacking objects: 100% (70/70), 29.77 KiB | 14.00 KiB/s, done.
(base) PS D:\Downloads> cd SimpFileConv
(base) PS D:\Downloads\SimpFileConv> python primary.py
nter the file path: Sample.csv
{'0': 'Bread',
 '1': 'Wine',
 '2': 'Eggs',
 '3': 'Meat',
 '4': 'Cheese',
 '5': 'Pencil',
 '6': 'Diaper'}
{'0': 'Bread',
 '1': 'Cheese',
 '2': 'Meat',
 '3': 'Diaper',
 '4': 'Wine',
 '5': 'Milk',
```

Fig: 1.1

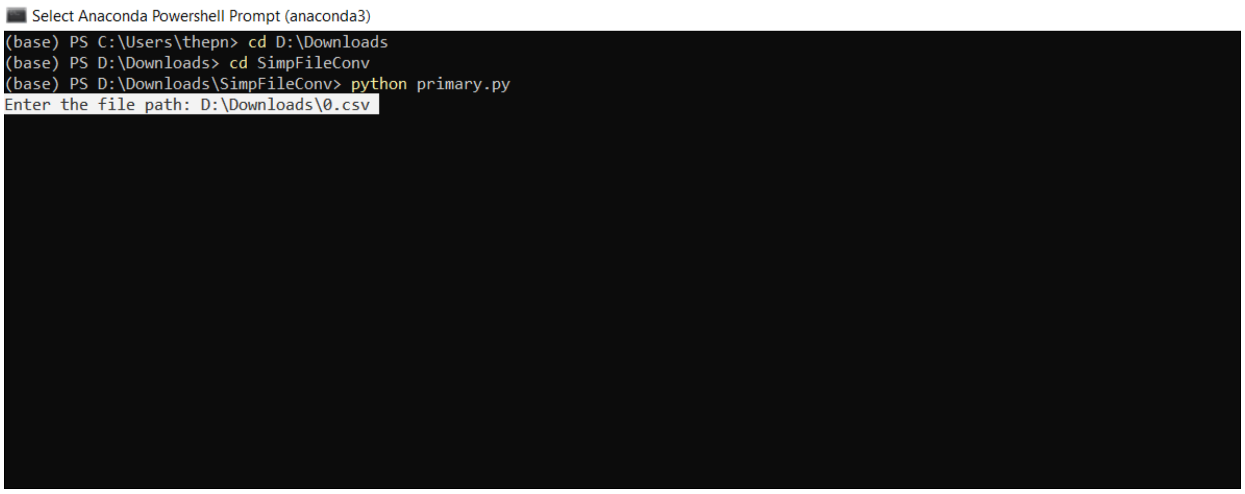
The above figure shows that upon program’s execution, user is asked to enter the CSV file



```
 '5': 'Meat',
 '6': 'Wine'}
{'0': 'Milk',
 '1': 'Cheese',
 '2': 'Wine',
 '3': 'Meat',
 '4': 'Bagel',
 '5': 'Diaper',
 '6': 'Bread'}
{'0': 'Bagel',
 '1': 'Diaper',
 '2': 'Milk',
 '3': 'Cheese',
 '4': 'Wine',
 '5': '',
 '6': ''}
{'0': 'Bread', '1': 'Bagel', '2': 'Milk', '3': '', '4': '', '5': '', '6': ''}
{'0': 'Diaper', '1': '', '2': '', '3': '', '4': '', '5': '', '6': ''}
{'0': 'Bagel',
 '1': 'Pencil',
 '2': 'Bread',
 '3': 'Cheese',
 '4': 'Eggs',
 '5': '',
 '6': ''}
{'0': 'Bread', '1': 'Eggs', '2': 'Cheese', '3': '', '4': '', '5': '', '6': ''}
{'0': 'Meat', '1': 'Milk', '2': 'Pencil', '3': '', '4': '', '5': '', '6': ''}
{'0': 'Bread',
 '1': 'Cheese',
 '2': 'Eggs',
 '3': 'Meat',
 '4': 'Pencil',
 '5': 'Diaper',
 '6': 'Wine'}
{'0': 'Meat', '1': 'Cheese', '2': '', '3': '', '4': '', '5': '', '6': ''}
{'0': 'Eggs',
 '1': 'Wine',
 '2': 'Bagel',
 '3': 'Bread',
 '4': 'Meat',
 '5': '',
 '6': ''}
The time taken to convert from CSV to JSON format is : 0.9883125000000001s
(base) PS D:\Downloads\SimpFileConv> _
```

Fig: 1.2

The converted JSON format is displayed in the command prompt along with the total time taken for conversion

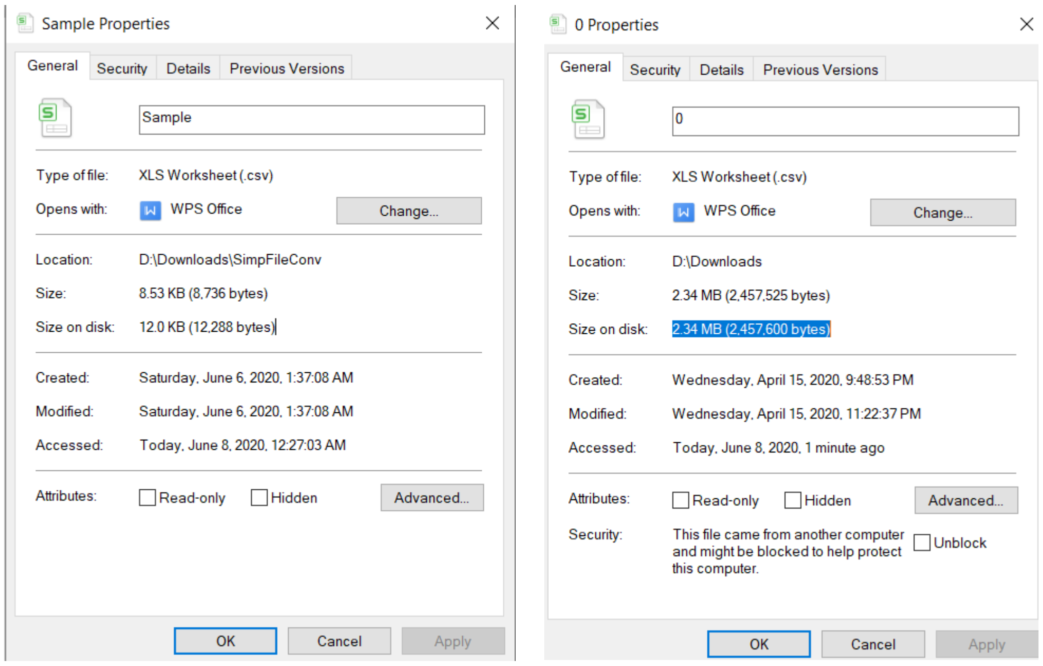


Select Anaconda Powershell Prompt (anaconda3)

```
(base) PS C:\Users\thepn> cd D:\Downloads
(base) PS D:\Downloads> cd SimpFileConv
(base) PS D:\Downloads\SimpFileConv> python primary.py
Enter the file path: D:\Downloads\0.csv
```

Fig: 1.3

As mentioned earlier, files from various locations of the system can be fetched for the conversion



Sample Properties

General | Security | Details | Previous Versions

	Sample
Type of file:	XLS Worksheet (.csv)
Opens with:	WPS Office
Location:	D:\Downloads\SimpFileConv
Size:	8.53 KB (8,736 bytes)
Size on disk:	12.0 KB (12,288 bytes)
Created:	Saturday, June 6, 2020, 1:37:08 AM
Modified:	Saturday, June 6, 2020, 1:37:08 AM
Accessed:	Today, June 8, 2020, 12:27:03 AM
Attributes:	☐ Read-only ☐ Hidden

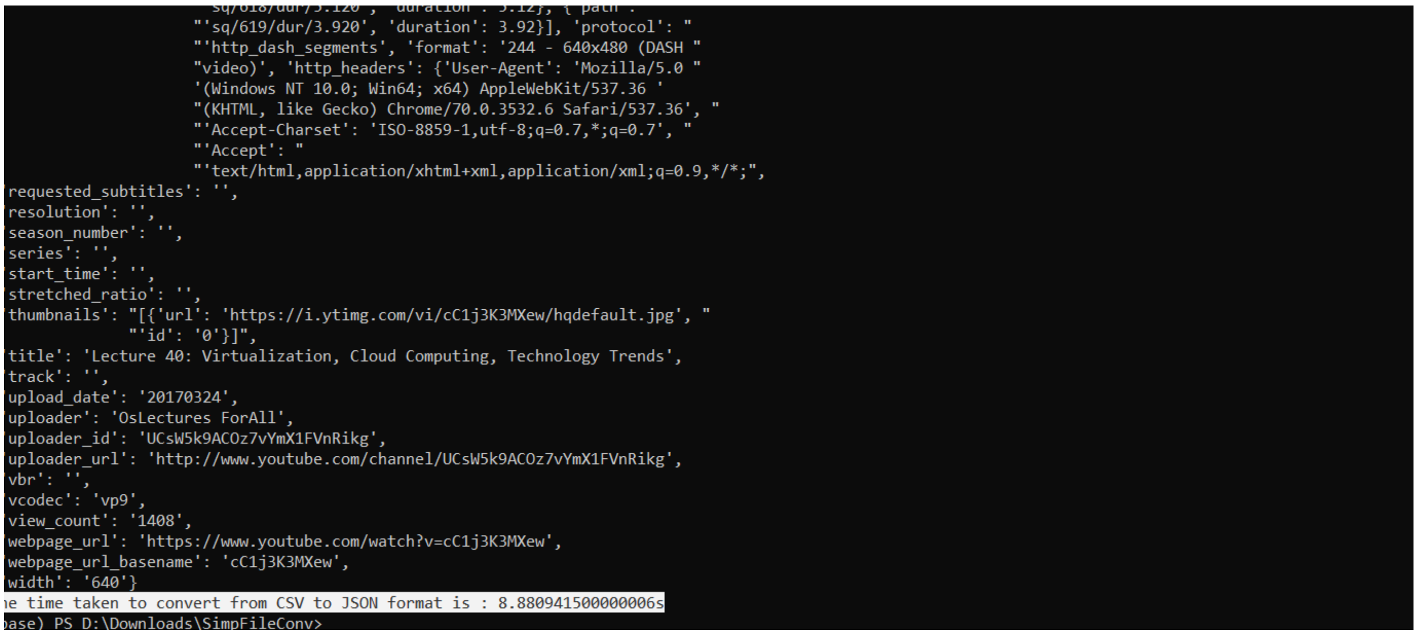
0 Properties

General | Security | Details | Previous Versions

	0
Type of file:	XLS Worksheet (.csv)
Opens with:	WPS Office
Location:	D:\Downloads
Size:	2.34 MB (2,457,525 bytes)
Size on disk:	2.34 MB (2,457,600 bytes)
Created:	Wednesday, April 15, 2020, 9:48:53 PM
Modified:	Wednesday, April 15, 2020, 11:22:37 PM
Accessed:	Today, June 8, 2020, 1 minute ago
Attributes:	☐ Read-only ☐ Hidden
Security:	This file came from another computer and might be blocked to help protect this computer. ☐ Unblock

Fig: 1.4 & 1.5

The above figures show the sizes of the two files used for this conversion. Fig 1.4 shows the size of sample file used for conversion whereas fig 1.5 shows the size of random file selected as shown in Fig 1.3.

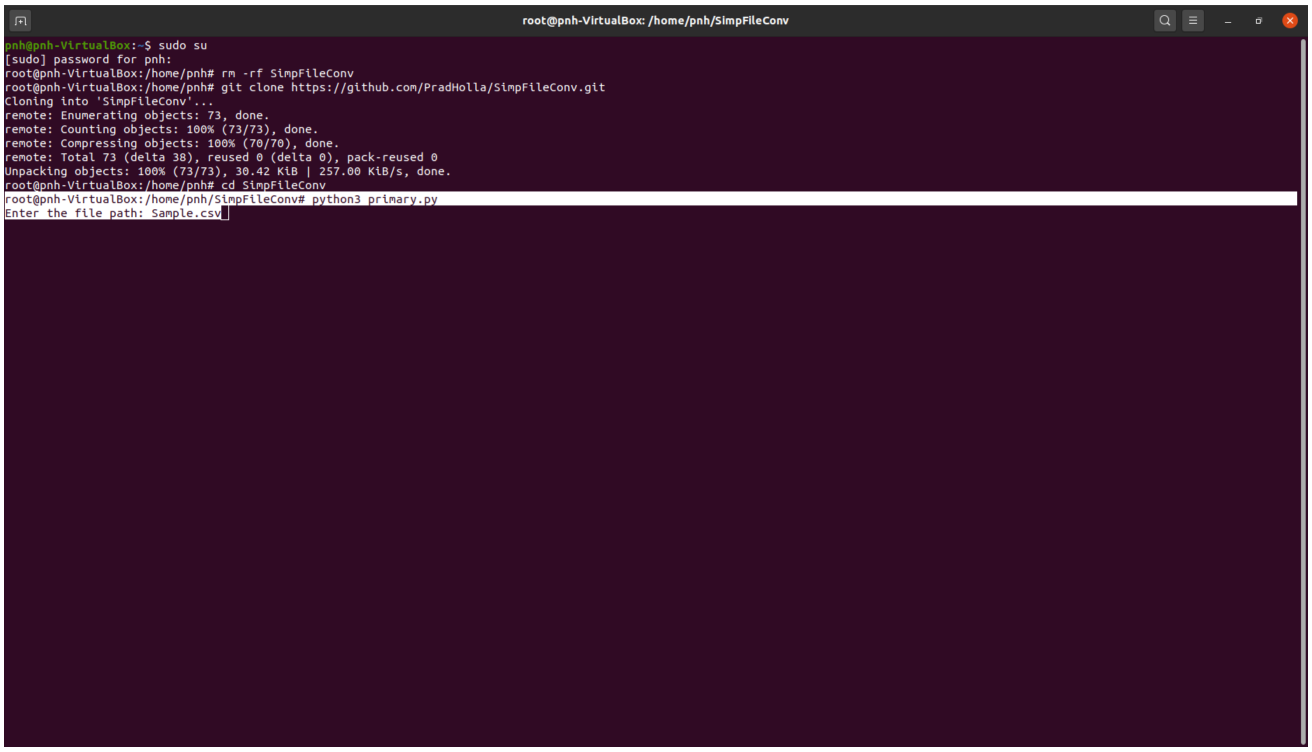


```
                  "'sq/619/dur/3.920', 'duration': 3.92}], 'protocol': "
                  "'http_dash_segments', 'format': '244 - 640x480 (DASH "
                  "video)', 'http_headers': {'User-Agent': 'Mozilla/5.0 "
                  '(Windows NT 10.0; Win64; x64) AppleWebKit/537.36 '
                  "(KHTML, like Gecko) Chrome/70.0.3532.6 Safari/537.36', "
                  "'Accept-Charset': 'ISO-8859-1,utf-8;q=0.7,*;q=0.7', "
                  "'Accept': "
                  "'text/html,application/xhtml+xml,application/xml;q=0.9,*/*;",
'requested_subtitles': '',
resolution': '',
season_number': '',
series': '',
start_time': '',
stretched_ratio': '',
thumbnails': "[{'url': 'https://i.ytimg.com/vi/cC1j3K3MXew/hqdefault.jpg', "
             "'id': '0'}]",
'title': 'Lecture 40: Virtualization, Cloud Computing, Technology Trends',
track': '',
upload_date': '20170324',
uploader': 'OsLectures ForAll',
uploader_id': 'UCsW5k9ACOz7vYmX1FVnRikg',
uploader_url': 'http://www.youtube.com/channel/UCsW5k9ACOz7vYmX1FVnRikg',
vbr': '',
vcodec': 'vp9',
view_count': '1408',
webpage_url': 'https://www.youtube.com/watch?v=cC1j3K3MXew',
webpage_url_basename': 'cC1j3K3MXew',
width': '640'}
ie time taken to convert from CSV to JSON format is : 8.880941500000006s
pase) PS D:\Downloads\SimpFileConv>
```

Fig: 1.6

As said previously and referring to figure 1.5, the above image shows how conversion takes longer when the file size increases.

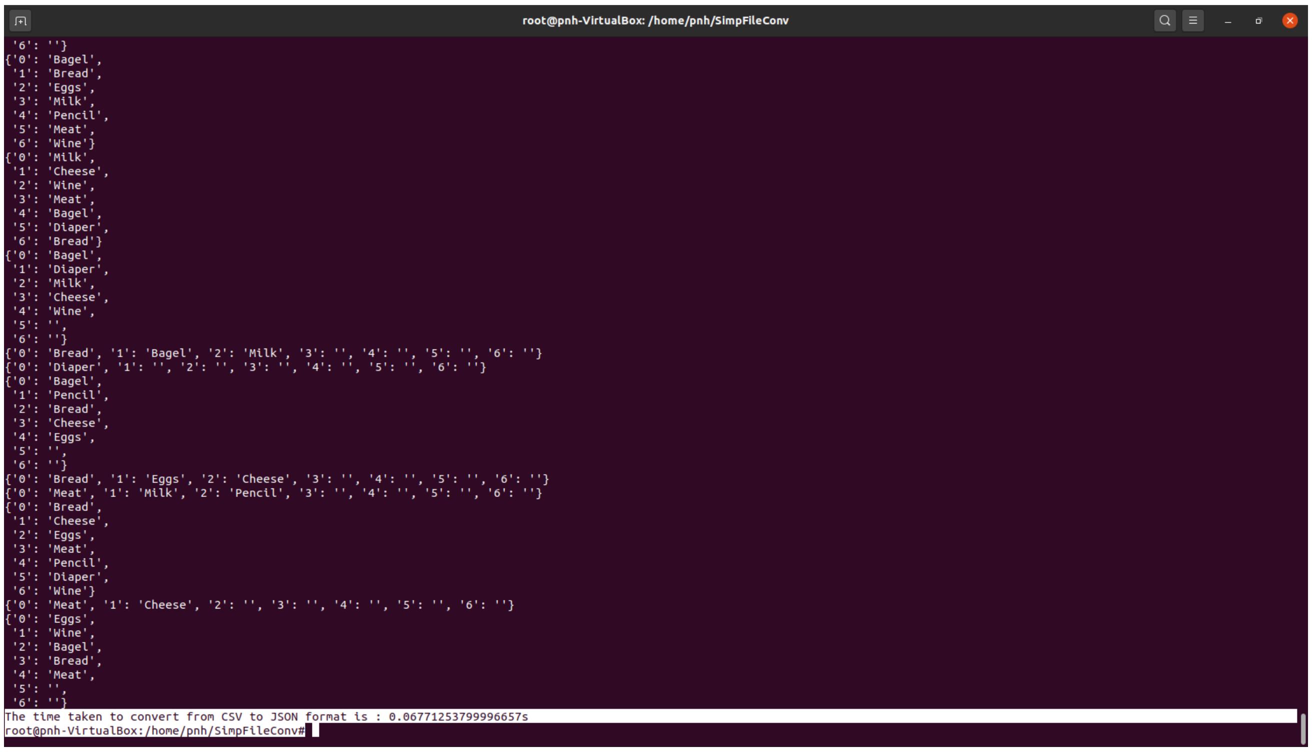
4.1.2 Primary Implementation in Linux Machine:



```
pnh@pnh-VirtualBox:~$ sudo su
[sudo] password for pnh:
root@pnh-VirtualBox:/home/pnh# rm -rf SimpFileConv
root@pnh-VirtualBox:/home/pnh# git clone https://github.com/PradHolla/SimpFileConv.git
Cloning into 'SimpFileConv'...
remote: Enumerating objects: 73, done.
remote: Counting objects: 100% (73/73), done.
remote: Compressing objects: 100% (70/70), done.
remote: Total 73 (delta 38), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (73/73), 30.42 KiB | 257.00 KiB/s, done.
root@pnh-VirtualBox:/home/pnh# cd SimpFileConv
root@pnh-VirtualBox:/home/pnh/SimpFileConv# python3 primary.py
Enter the file path: Sample.csv
```

Fig: 2.1

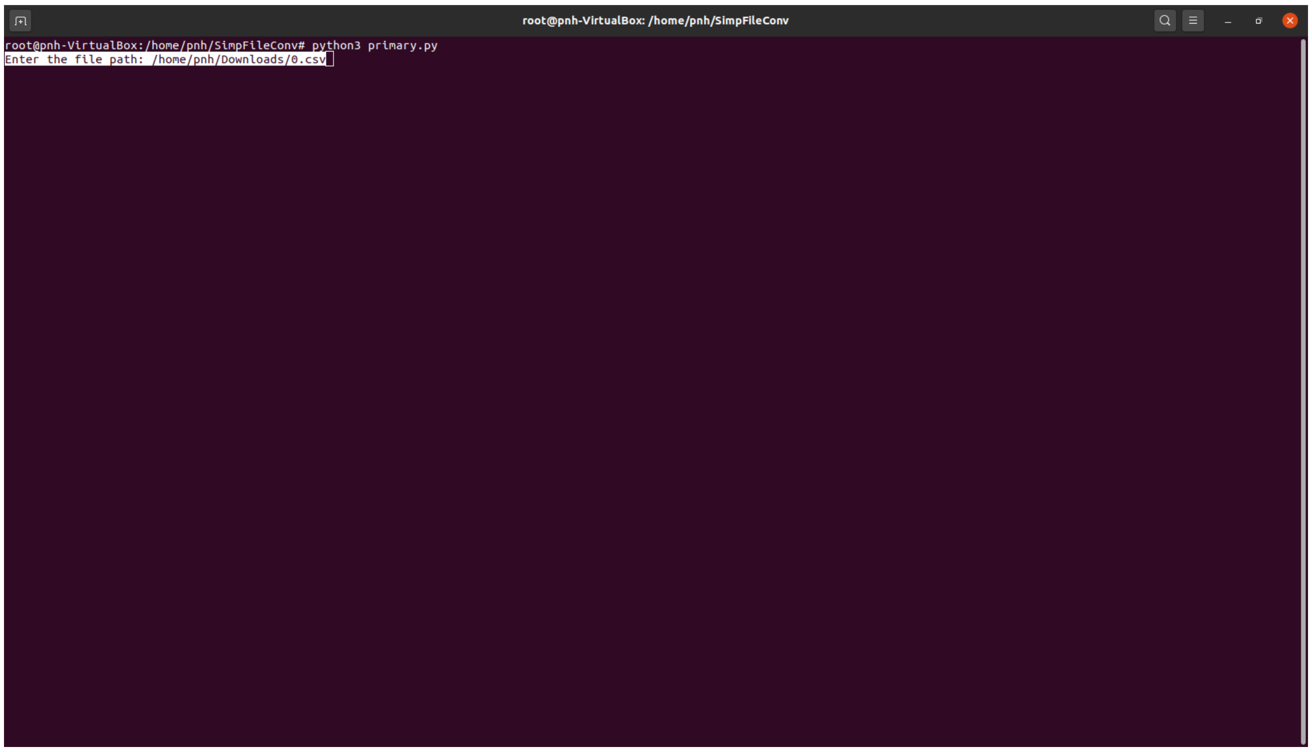
Implementation in Linux here is as same as Windows Machine's implementation



```
 '6': ''}
{'0': 'Bagel',
 '1': 'Bread',
 '2': 'Eggs',
 '3': 'Milk',
 '4': 'Pencil',
 '5': 'Meat',
 '6': 'Wine'}
{'0': 'Milk',
 '1': 'Cheese',
 '2': 'Wine',
 '3': 'Meat',
 '4': 'Bagel',
 '5': 'Diaper',
 '6': 'Bread'}
{'0': 'Bagel',
 '1': 'Diaper',
 '2': 'Milk',
 '3': 'Cheese',
 '4': 'Wine',
 '5': '',
 '6': ''}
{'0': 'Bread', '1': 'Bagel', '2': 'Milk', '3': '', '4': '', '5': '', '6': ''}
{'0': 'Diaper', '1': '', '2': '', '3': '', '4': '', '5': '', '6': ''}
{'0': 'Bagel',
 '1': 'Pencil',
 '2': 'Bread',
 '3': 'Cheese',
 '4': 'Eggs',
 '5': '',
 '6': ''}
{'0': 'Bread', '1': 'Eggs', '2': 'Cheese', '3': '', '4': '', '5': '', '6': ''}
{'0': 'Meat', '1': 'Milk', '2': 'Pencil', '3': '', '4': '', '5': '', '6': ''}
{'0': 'Bread',
 '1': 'Cheese',
 '2': 'Eggs',
 '3': 'Meat',
 '4': 'Pencil',
 '5': 'Diaper',
 '6': 'Wine'}
{'0': 'Meat', '1': 'Cheese', '2': '', '3': '', '4': '', '5': '', '6': ''}
{'0': 'Eggs',
 '1': 'Wine',
 '2': 'Bagel',
 '3': 'Bread',
 '4': 'Meat',
 '5': '',
 '6': ''}
The time taken to convert from CSV to JSON format is : 0.06771253799996657s
root@pnh-VirtualBox:/home/pnh/SimpFileConv#
```

Fig: 2.2

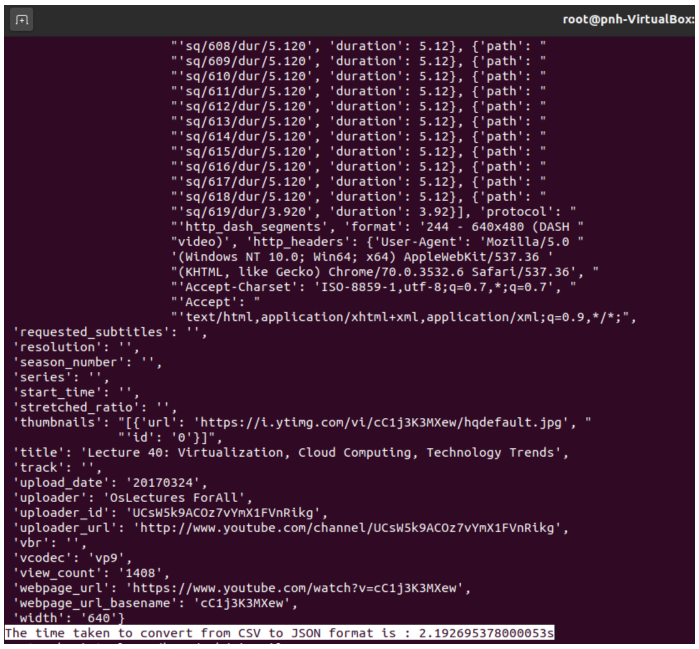
CSV file is converted to JSON format and displayed along with the time taken



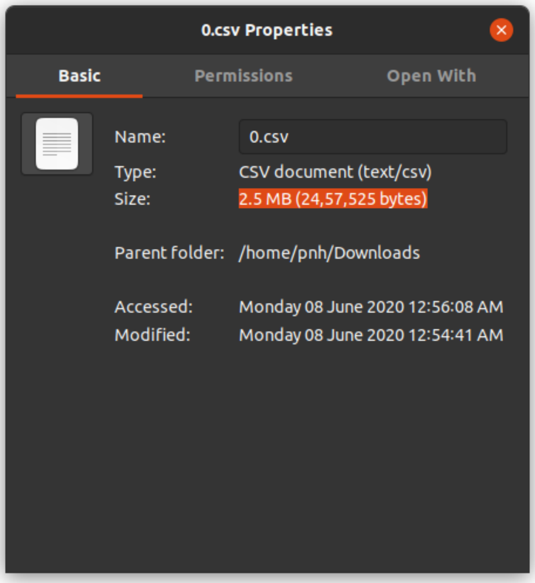
```
root@pnh-VirtualBox: /home/pnh/SimpFileConv
root@pnh-VirtualBox:/home/pnh/SimpFileConv# python3 primary.py
Enter the file path: /home/pnh/Downloads/0.csv
```

Fig: 2.3

Files from other locations can also be given for conversion



```
root@pnh-VirtualBox:
                     "'sq/608/dur/5.120', 'duration': 5.12}, {'path': "
                     "'sq/609/dur/5.120', 'duration': 5.12}, {'path': "
                     "'sq/610/dur/5.120', 'duration': 5.12}, {'path': "
                     "'sq/611/dur/5.120', 'duration': 5.12}, {'path': "
                     "'sq/612/dur/5.120', 'duration': 5.12}, {'path': "
                     "'sq/613/dur/5.120', 'duration': 5.12}, {'path': "
                     "'sq/614/dur/5.120', 'duration': 5.12}, {'path': "
                     "'sq/615/dur/5.120', 'duration': 5.12}, {'path': "
                     "'sq/616/dur/5.120', 'duration': 5.12}, {'path': "
                     "'sq/617/dur/5.120', 'duration': 5.12}, {'path': "
                     "'sq/618/dur/5.120', 'duration': 5.12}, {'path': "
                     "'sq/619/dur/3.920', 'duration': 3.92}], 'protocol': "
                     "'http_dash_segments', 'format': '244 - 640x480 (DASH "
                     "video)', 'http_headers': {'User-Agent': 'Mozilla/5.0 "
                     '(Windows NT 10.0; Win64; x64) AppleWebKit/537.36 '
                     "(KHTML, like Gecko) Chrome/70.0.3532.6 Safari/537.36', "
                     "'Accept-Charset': 'ISO-8859-1,utf-8;q=0.7,*;q=0.7', "
                     "'Accept': "
                     "'text/html,application/xhtml+xml,application/xml;q=0.9,*/*;",
 'requested_subtitles': '',
 'resolution': '',
 'season_number': '',
 'series': '',
 'start_time': '',
 'stretched_ratio': '',
 'thumbnails': "[{'url': 'https://i.ytimg.com/vi/cC1j3K3MXew/hqdefault.jpg', "
               "'id': '0'}]",
 'title': 'Lecture 40: Virtualization, Cloud Computing, Technology Trends',
 'track': '',
 'upload_date': '20170324',
 'uploader': 'OsLectures ForAll',
 'uploader_id': 'UCsW5k9ACOz7vYmX1FVnRikg',
 'uploader_url': 'http://www.youtube.com/channel/UCsW5k9ACOz7vYmX1FVnRikg',
 'vbr': '',
 'vcodec': 'vp9',
 'view_count': '1408',
 'webpage_url': 'https://www.youtube.com/watch?v=cC1j3K3MXew',
 'webpage_url_basename': 'cC1j3K3MXew',
 'width': '640'}
The time taken to convert from CSV to JSON format is : 2.192695378000053s
```



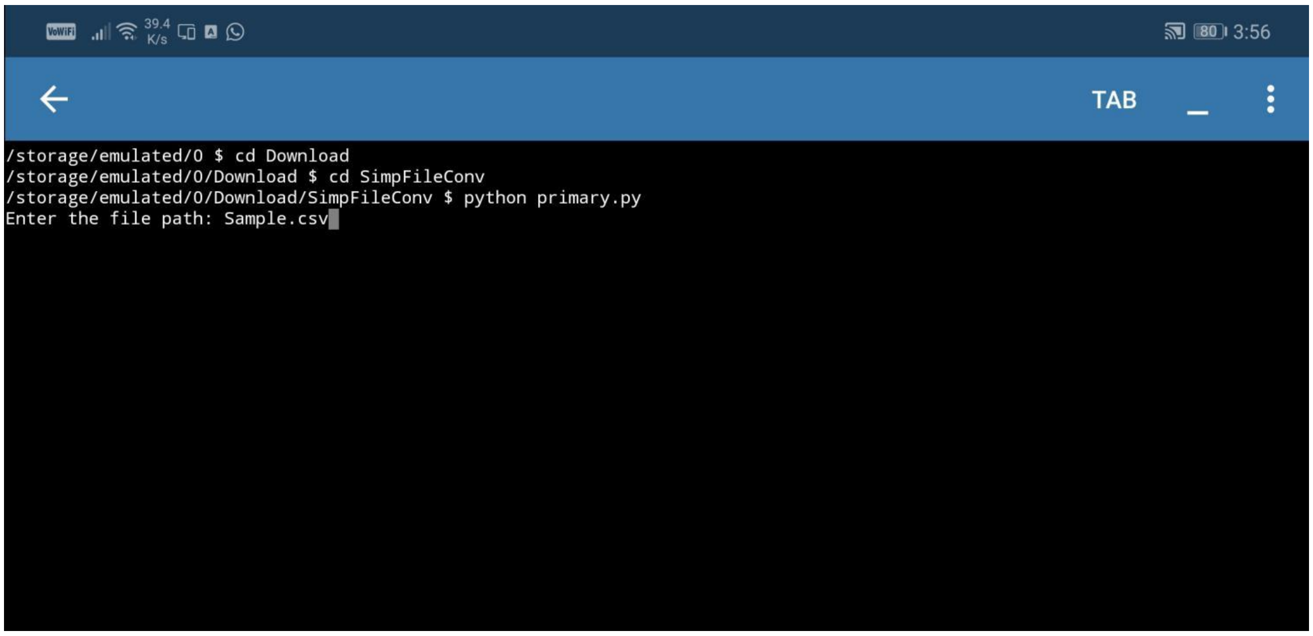
0.csv Properties

Name:	0.csv
Type:	CSV document (text/csv)
Size:	2.5 MB (24,57,525 bytes)
Parent folder:	/home/pnh/Downloads
Accessed:	Monday 08 June 2020 12:56:08 AM
Modified:	Monday 08 June 2020 12:54:41 AM

Fig: 2.4 & Fig: 2.5

Observing Fig: 2.4 & 2.5 and referring Fig: 1.5 & 1.6, it can be concluded that the Linux Machine is significantly faster in conversion

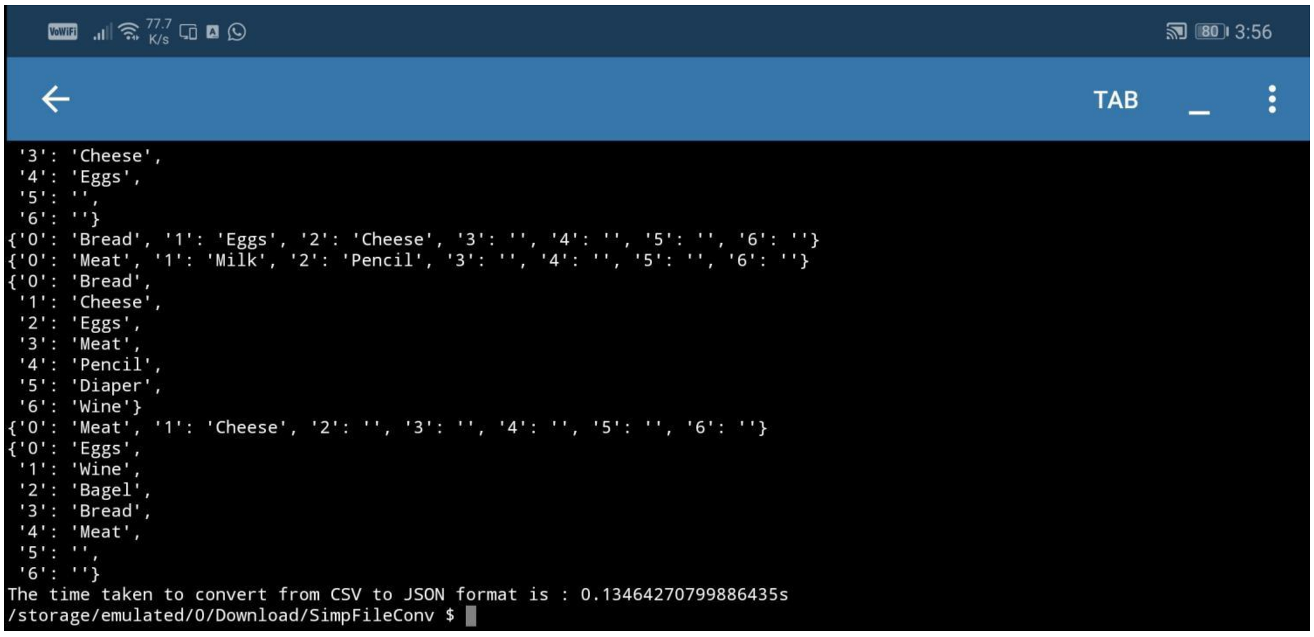
4.1.3 Primary Implementation in Android Device:



```
/storage/emulated/0 $ cd Download
/storage/emulated/0/Download $ cd SimpFileConv
/storage/emulated/0/Download/SimpFileConv $ python primary.py
Enter the file path: Sample.csv
```

Fig: 3.1

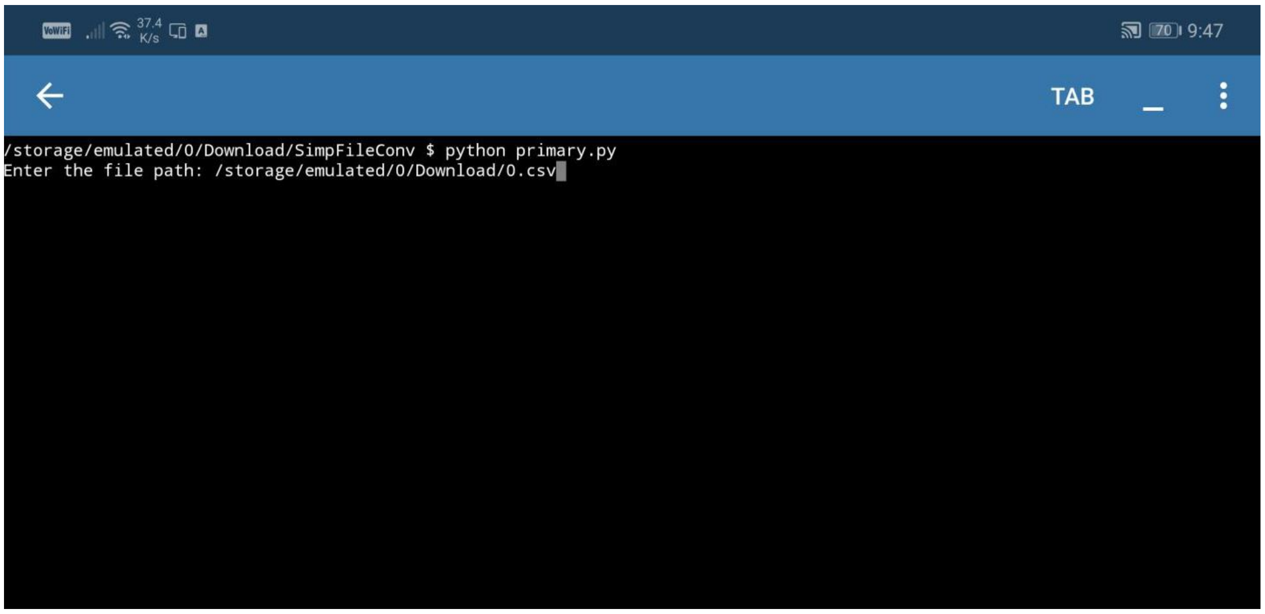
Similar implementation as Windows' and Linux's



```
 '3': 'Cheese',
 '4': 'Eggs',
 '5': '',
 '6': ''}
{'0': 'Bread', '1': 'Eggs', '2': 'Cheese', '3': '', '4': '', '5': '', '6': ''}
{'0': 'Meat', '1': 'Milk', '2': 'Pencil', '3': '', '4': '', '5': '', '6': ''}
{'0': 'Bread',
 '1': 'Cheese',
 '2': 'Eggs',
 '3': 'Meat',
 '4': 'Pencil',
 '5': 'Diaper',
 '6': 'Wine'}
{'0': 'Meat', '1': 'Cheese', '2': '', '3': '', '4': '', '5': '', '6': ''}
{'0': 'Eggs',
 '1': 'Wine',
 '2': 'Bagel',
 '3': 'Bread',
 '4': 'Meat',
 '5': '',
 '6': ''}
The time taken to convert from CSV to JSON format is : 0.13464270799886435s
/storage/emulated/0/Download/SimpFileConv $
```

Fig: 3.2

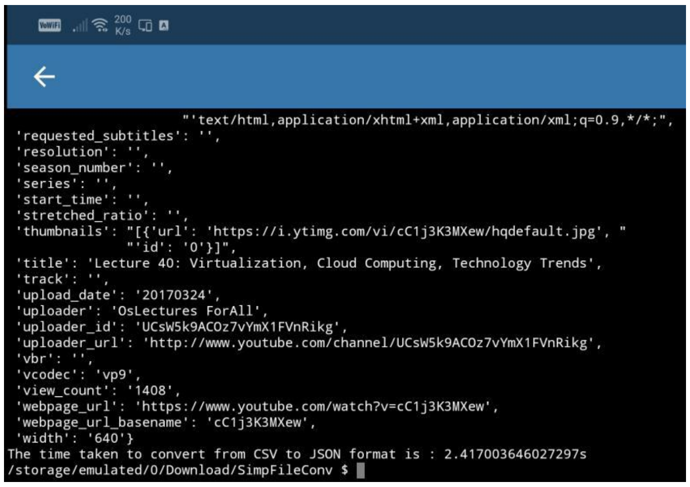
The CSV file is converted to JSON format and displayed along with the time taken for conversion



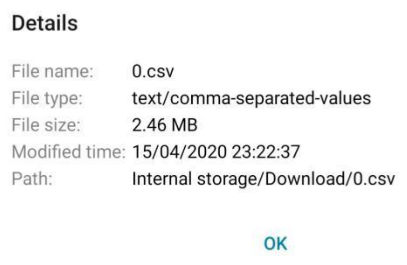
```
/storage/emulated/0/Download/SimpFileConv $ python primary.py
Enter the file path: /storage/emulated/0/Download/0.csv
```

Fig: 3.3

Entering the location of another CSV file present in the device



```
                       "'text/html,application/xhtml+xml,application/xml;q=0.9,*/*;",
 'requested_subtitles': '',
 'resolution': '',
 'season_number': '',
 'series': '',
 'start_time': '',
 'stretched_ratio': '',
 'thumbnails': "[{'url': 'https://i.ytimg.com/vi/cC1j3K3MXew/hqdefault.jpg', "
               "'id': '0'}]",
 'title': 'Lecture 40: Virtualization, Cloud Computing, Technology Trends',
 'track': '',
 'upload_date': '20170324',
 'uploader': 'OsLectures ForAll',
 'uploader_id': 'UCsW5k9ACOz7vYmX1FVnRikg',
 'uploader_url': 'http://www.youtube.com/channel/UCsW5k9ACOz7vYmX1FVnRikg',
 'vbr': '',
 'vcodec': 'vp9',
 'view_count': '1408',
 'webpage_url': 'https://www.youtube.com/watch?v=cC1j3K3MXew',
 'webpage_url_basename': 'cC1j3K3MXew',
 'width': '640'}
The time taken to convert from CSV to JSON format is : 2.417003646027297s
/storage/emulated/0/Download/SimpFileConv $
```



Details

File name:	0.csv
File type:	text/comma-separated-values
File size:	2.46 MB
Modified time:	15/04/2020 23:22:37
Path:	Internal storage/Download/0.csv

Fig: 3.4 & Fig: 3.5

Even here, the time taken to convert the same file is lesser than Windows' conversion

4.2.1 Secondary Implementation in Windows Machine:

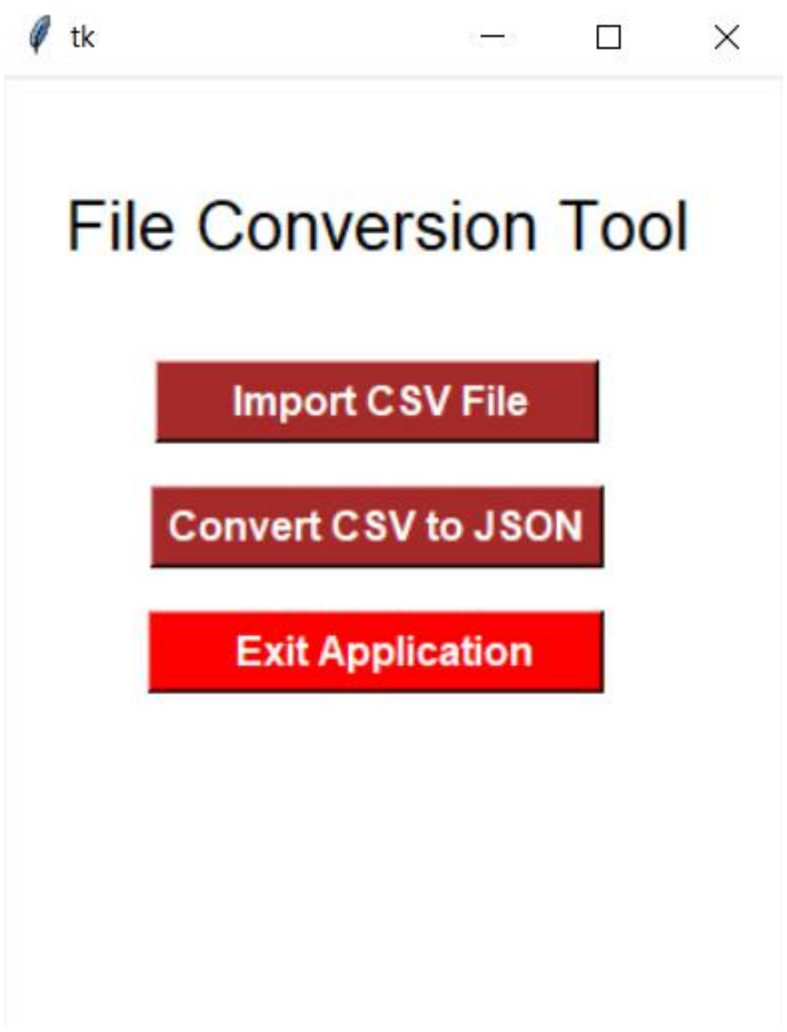


Fig: 4.1

This is the Graphical User Interface providing three functions to the user

As mentioned earlier, there are three functions that a user can make use of in the Graphical User Interface. The First option is to import a CSV file from the system. The second is to convert the selected/imported file to JSON format and store it in a file. The user has to specify name and location of the new JSON file. The third option is to exit from the GUI.

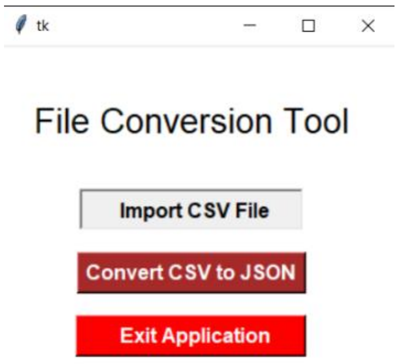
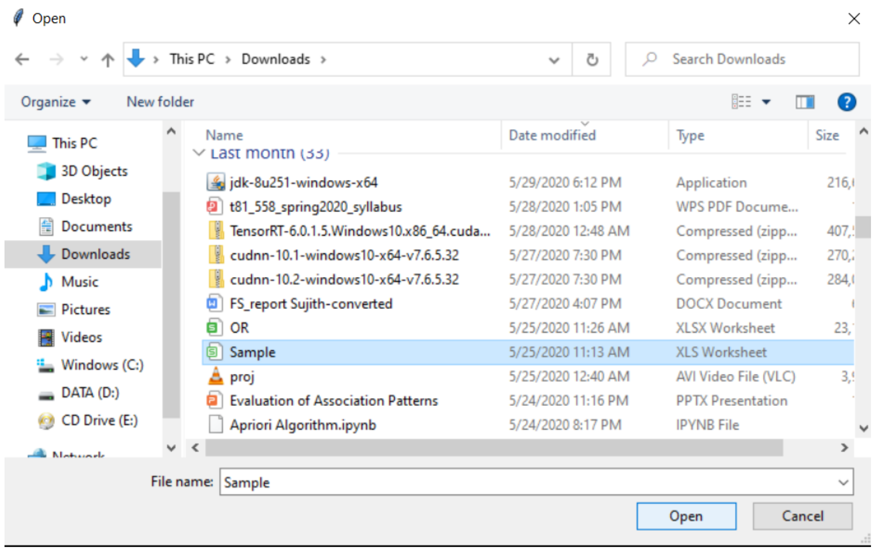


Fig: 4.2



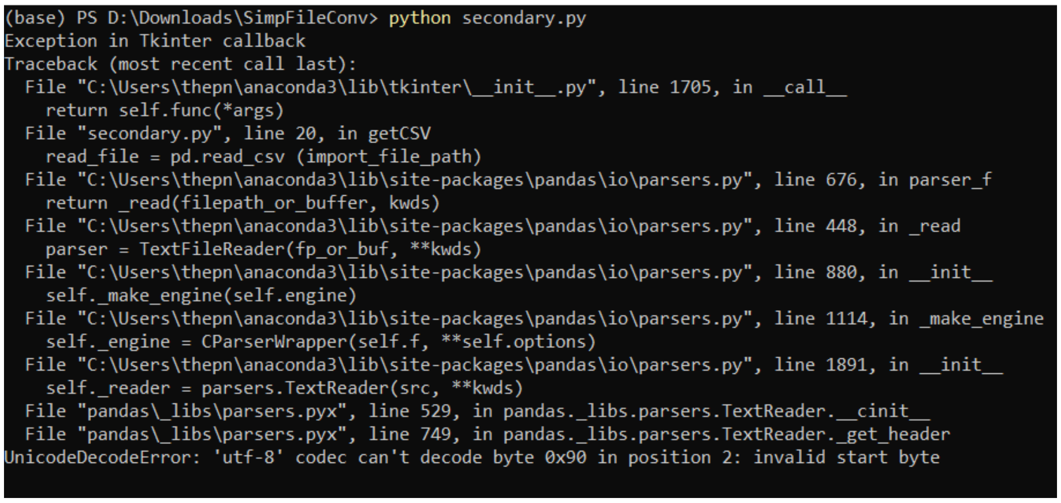
Last month (33)

Name	Date modified	Type	Size
jdk-8u251-windows-x64	5/29/2020 6:12 PM	Application	216,
t81_558_spring2020_syllabus	5/28/2020 1:05 PM	WPS PDF Docume...	
TensorRT-6.0.1.5.Windows10.x86_64.cuda...	5/28/2020 12:48 AM	Compressed (zipp...	407,
cudnn-10.1-windows10-x64-v7.6.5.32	5/27/2020 7:30 PM	Compressed (zipp...	270,
cudnn-10.2-windows10-x64-v7.6.5.32	5/27/2020 7:30 PM	Compressed (zipp...	284,
FS_report Sujith-converted	5/27/2020 4:07 PM	DOCX Document	
OR	5/25/2020 11:26 AM	XLSX Worksheet	23,
Sample	5/25/2020 11:13 AM	XLS Worksheet	
proj	5/25/2020 12:40 AM	AVI Video File (VLC)	3,
Evaluation of Association Patterns	5/24/2020 11:16 PM	PPTX Presentation	
Apriori Algorithm.ipynb	5/24/2020 8:17 PM	IPYNB File	

File name: Sample

Fig: 4.3

When ‘Import CSV File’ option is selected, a screen pops up asking the user to select the desired CSV File for conversion. The user has to select the appropriate file as shown in Fig: 4.3.

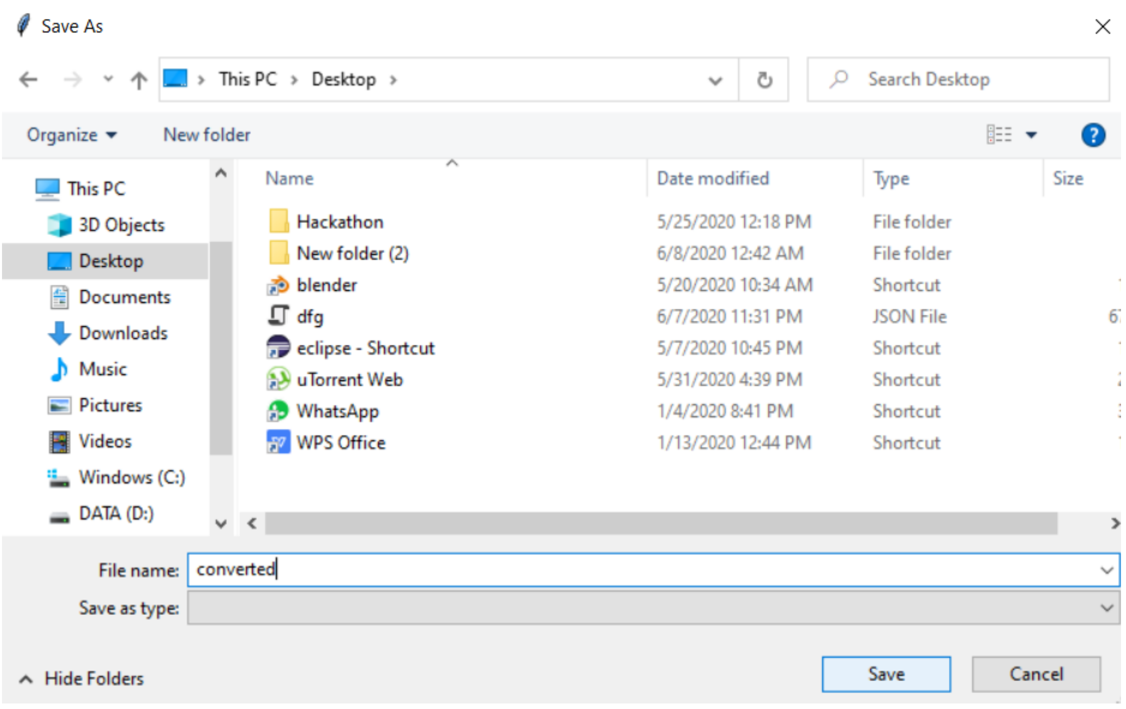


```
(base) PS D:\Downloads\SimpFileConv> python secondary.py
Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Users\thepn\anaconda3\lib\tkinter\__init__.py", line 1705, in __call__
    return self.func(*args)
  File "secondary.py", line 20, in getCSV
    read_file = pd.read_csv (import_file_path)
  File "C:\Users\thepn\anaconda3\lib\site-packages\pandas\io\parsers.py", line 676, in parser_f
    return _read(filepath_or_buffer, kwds)
  File "C:\Users\thepn\anaconda3\lib\site-packages\pandas\io\parsers.py", line 448, in _read
    parser = TextFileReader(fp_or_buf, **kwds)
  File "C:\Users\thepn\anaconda3\lib\site-packages\pandas\io\parsers.py", line 880, in __init__
    self._make_engine(self.engine)
  File "C:\Users\thepn\anaconda3\lib\site-packages\pandas\io\parsers.py", line 1114, in _make_engine
    self._engine = CParserWrapper(self.f, **self.options)
  File "C:\Users\thepn\anaconda3\lib\site-packages\pandas\io\parsers.py", line 1891, in __init__
    self._reader = parsers.TextReader(src, **kwds)
  File "pandas\_libs\parsers.pyx", line 529, in pandas._libs.parsers.TextReader.__cinit__
  File "pandas\_libs\parsers.pyx", line 749, in pandas._libs.parsers.TextReader._get_header
UnicodeDecodeError: 'utf-8' codec can't decode byte 0x90 in position 2: invalid start byte
```

Fig: 4.4

If an inappropriate file is chosen, an error prompts up in the CMD (Ref. Fig: 4.4)

After the suitable file is selected, the user has to select ‘Convert CSV to JSON’ and specify the name and location of the new JSON file that is yet to be created.



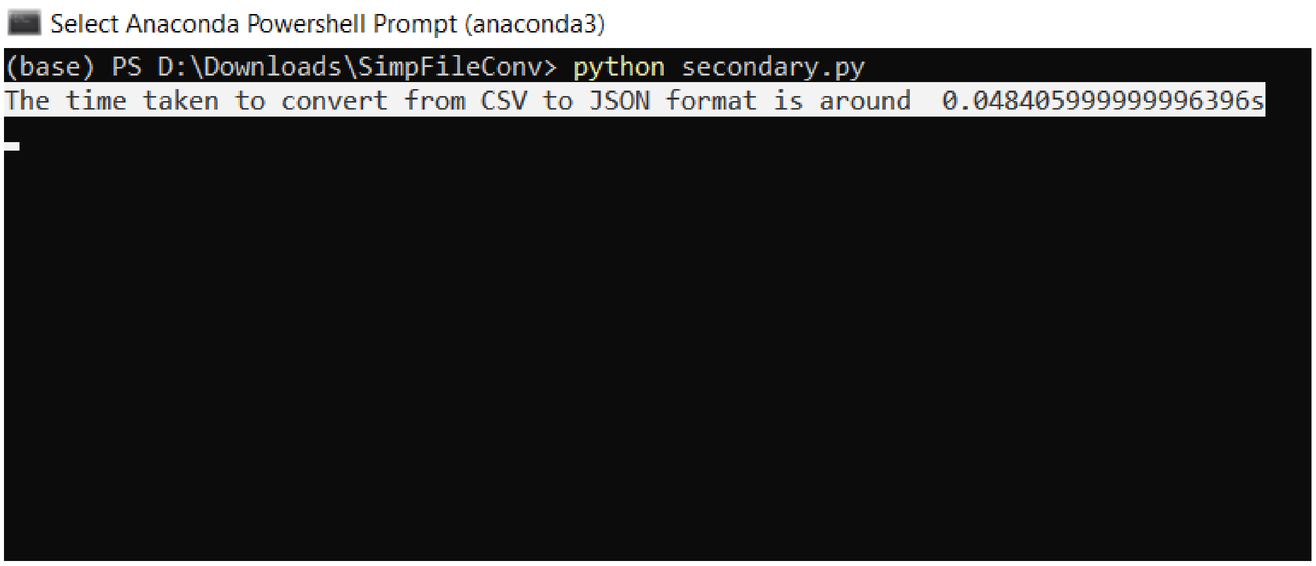
Name	Date modified	Type	Size
Hackathon	5/25/2020 12:18 PM	File folder	
New folder (2)	6/8/2020 12:42 AM	File folder	
blender	5/20/2020 10:34 AM	Shortcut	
dfg	6/7/2020 11:31 PM	JSON File	6
eclipse - Shortcut	5/7/2020 10:45 PM	Shortcut	
uTorrent Web	5/31/2020 4:39 PM	Shortcut	
WhatsApp	1/4/2020 8:41 PM	Shortcut	
WPS Office	1/13/2020 12:44 PM	Shortcut	

File name: converted
Save as type:

Fig: 4.5

Specifying name and location of the new file

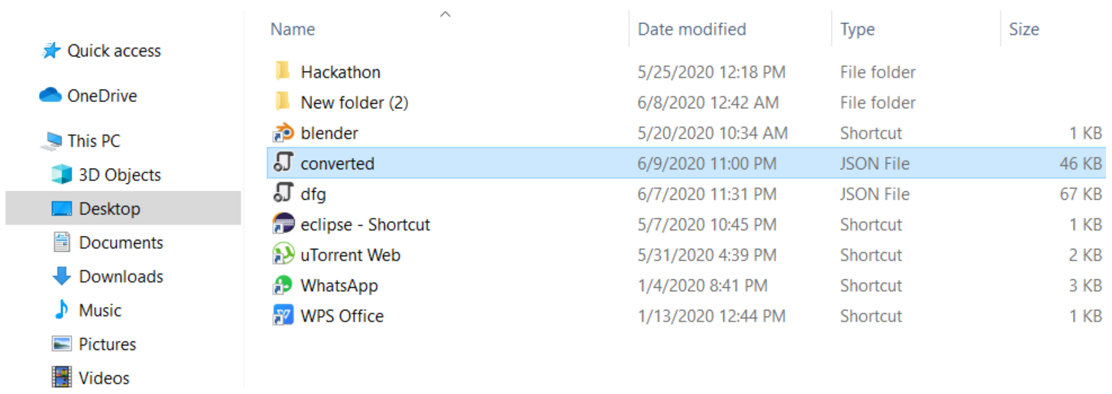
After the name and location are specified, the CSV file will be converted to JSON and the time taken to convert it is displayed in the CMD.



```
Select Anaconda Powershell Prompt (anaconda3)
(base) PS D:\Downloads\SimpFileConv> python secondary.py
The time taken to convert from CSV to JSON format is around  0.048405999999996396s
```

Fig: 4.6

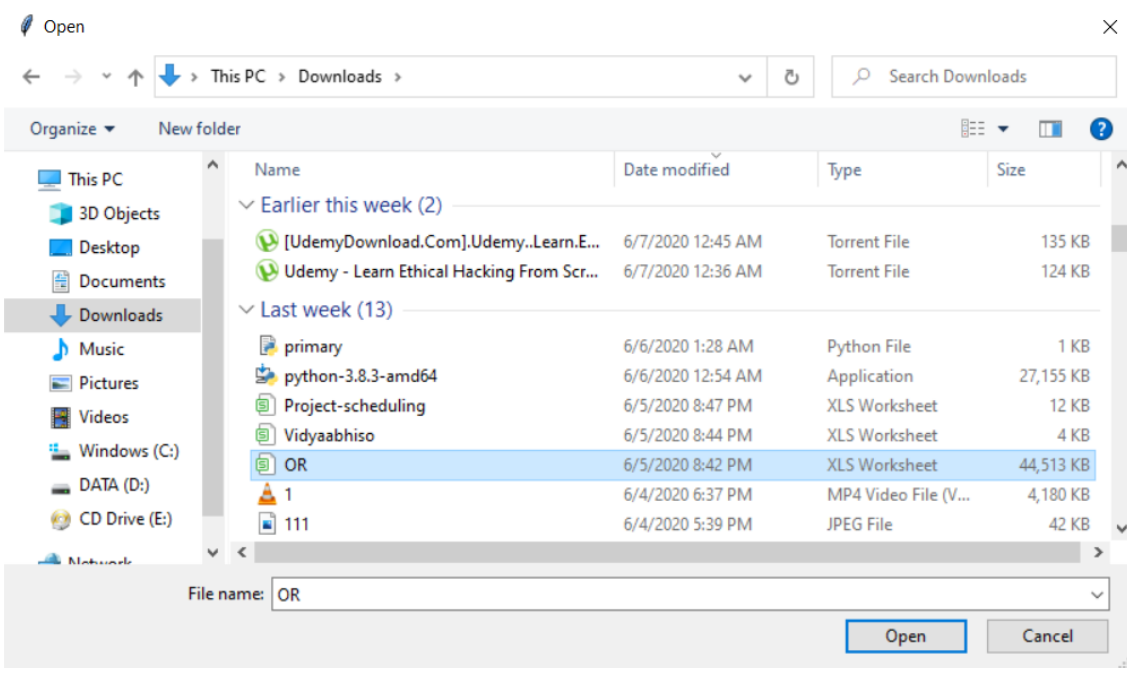
The time taken to convert is 0.0484s



Name	Date modified	Type	Size
Hackathon	5/25/2020 12:18 PM	File folder	
New folder (2)	6/8/2020 12:42 AM	File folder	
blender	5/20/2020 10:34 AM	Shortcut	1 KB
converted	6/9/2020 11:00 PM	JSON File	46 KB
dfg	6/7/2020 11:31 PM	JSON File	67 KB
eclipse - Shortcut	5/7/2020 10:45 PM	Shortcut	1 KB
uTorrent Web	5/31/2020 4:39 PM	Shortcut	2 KB
WhatsApp	1/4/2020 8:41 PM	Shortcut	3 KB
WPS Office	1/13/2020 12:44 PM	Shortcut	1 KB

Fig: 4.6

The converted file is present in the specified directory



Open

This PC › Downloads ›

Search Downloads

Organize ▾ New folder

This PC, 3D Objects, Desktop, Documents, Downloads, Music, Pictures, Videos, Windows (C:), DATA (D:), CD Drive (E:), Network

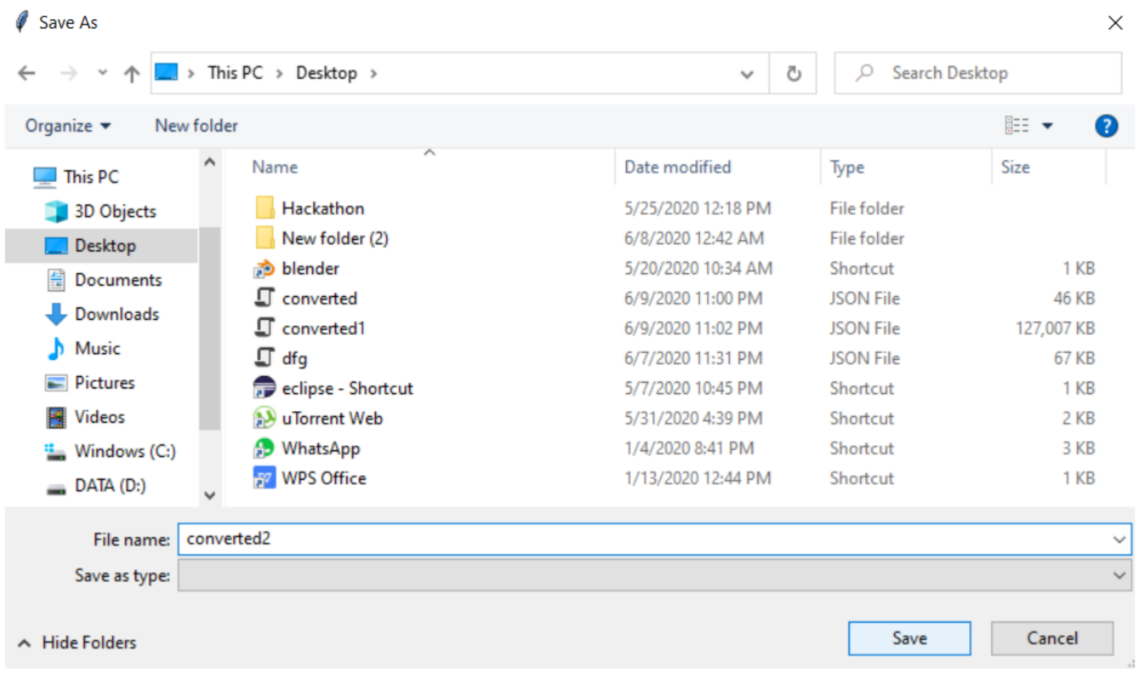
Name	Date modified	Type	Size
Earlier this week (2)			
[UdemyDownload.Com].Udemy..Learn.E...	6/7/2020 12:45 AM	Torrent File	135 KB
Udemy - Learn Ethical Hacking From Scr...	6/7/2020 12:36 AM	Torrent File	124 KB
Last week (13)			
primary	6/6/2020 1:28 AM	Python File	1 KB
python-3.8.3-amd64	6/6/2020 12:54 AM	Application	27,155 KB
Project-scheduling	6/5/2020 8:47 PM	XLS Worksheet	12 KB
Vidyaabhiso	6/5/2020 8:44 PM	XLS Worksheet	4 KB
OR	6/5/2020 8:42 PM	XLS Worksheet	44,513 KB
1	6/4/2020 6:37 PM	MP4 Video File (V...	4,180 KB
111	6/4/2020 5:39 PM	JPEG File	42 KB

File name: OR

Open Cancel

Fig: 4.7

Now, as seen in Fig: 4.6, a much larger file is selected for conversion. This is done to observe the time taken to convert a large file.



Save As

This PC › Desktop ›

Search Desktop

Organize ▾ New folder

This PC, 3D Objects, Desktop, Documents, Downloads, Music, Pictures, Videos, Windows (C:), DATA (D:)

Name	Date modified	Type	Size
Hackathon	5/25/2020 12:18 PM	File folder	
New folder (2)	6/8/2020 12:42 AM	File folder	
blender	5/20/2020 10:34 AM	Shortcut	1 KB
converted	6/9/2020 11:00 PM	JSON File	46 KB
converted1	6/9/2020 11:02 PM	JSON File	127,007 KB
dfg	6/7/2020 11:31 PM	JSON File	67 KB
eclipse - Shortcut	5/7/2020 10:45 PM	Shortcut	1 KB
uTorrent Web	5/31/2020 4:39 PM	Shortcut	2 KB
WhatsApp	1/4/2020 8:41 PM	Shortcut	3 KB
WPS Office	1/13/2020 12:44 PM	Shortcut	1 KB

File name: converted2

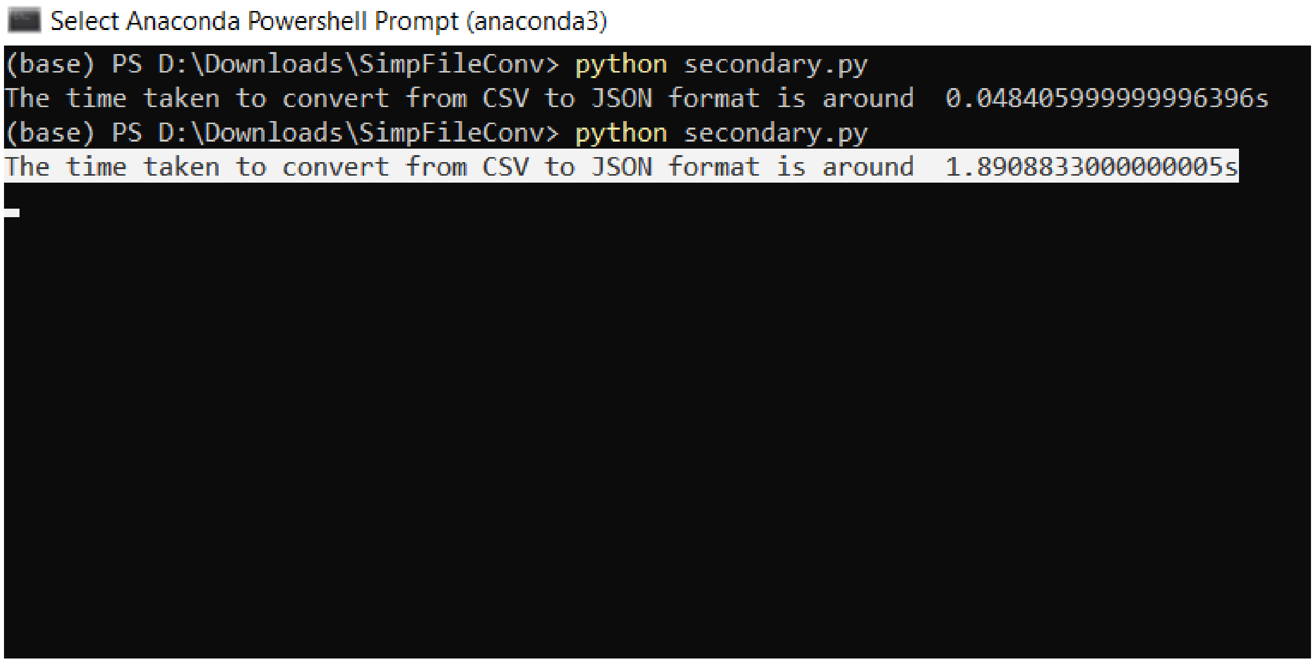
Save as type:

Hide Folders

Save Cancel

Fig: 4.8

The output file is named as converted2



Select Anaconda Powershell Prompt (anaconda3)

```
(base) PS D:\Downloads\SimpFileConv> python secondary.py
The time taken to convert from CSV to JSON format is around  0.048405999999996396s
(base) PS D:\Downloads\SimpFileConv> python secondary.py
The time taken to convert from CSV to JSON format is around  1.8908833000000005s
```

Fig: 4.9

The time taken to convert is 1.89s

Although OR.csv is much larger than the sample file, conversions happen quicker here as we make use of the pandas library.

4.2.2 Secondary Implementation in Linux Machine:

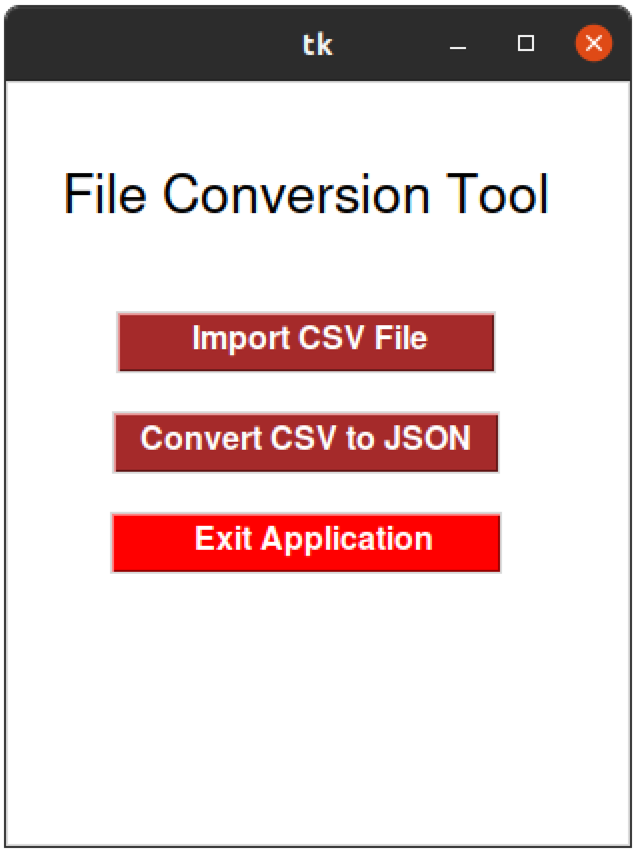


Fig: 5.1

The GUI appears this way in Linux System

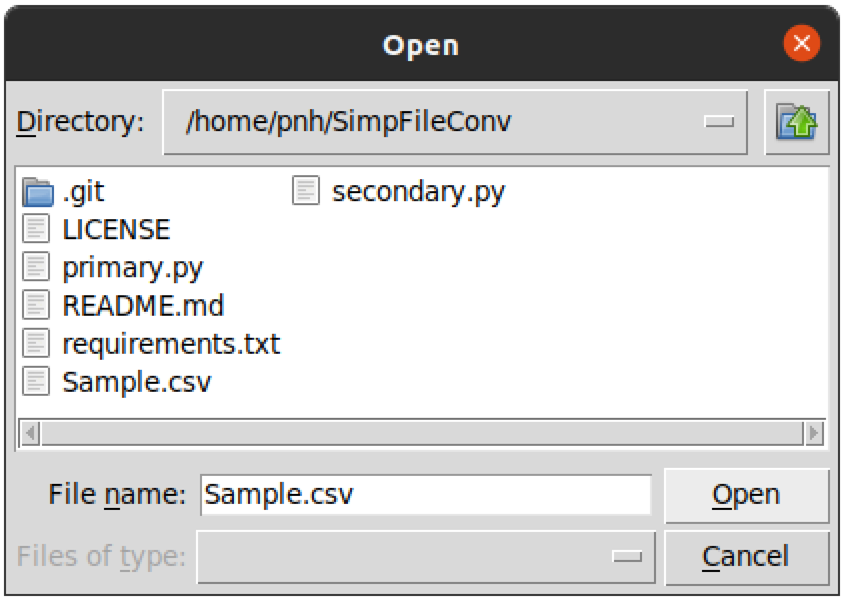


Fig: 5.2

Similar to the Windows' implementation, we select the desired file after choosing 'Import CSV File'. (Refer Fig: 5.1)

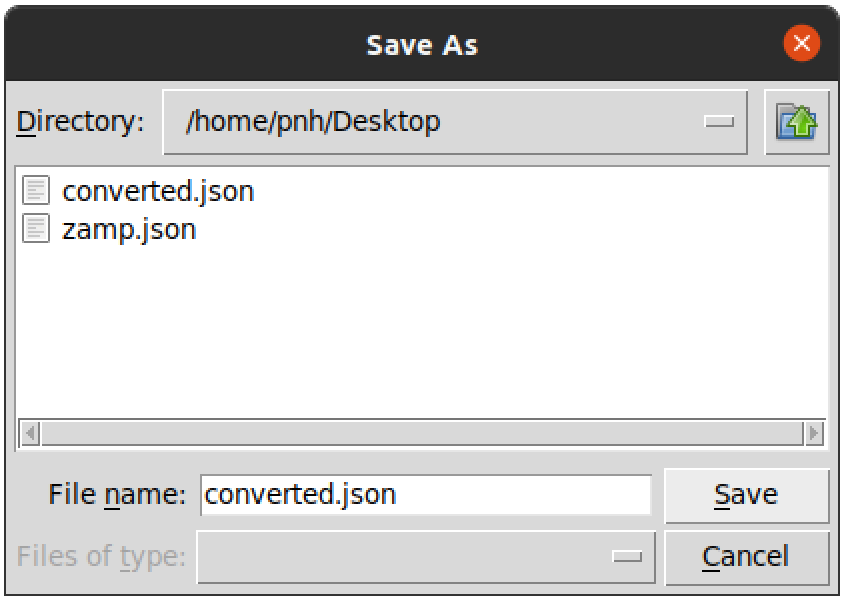
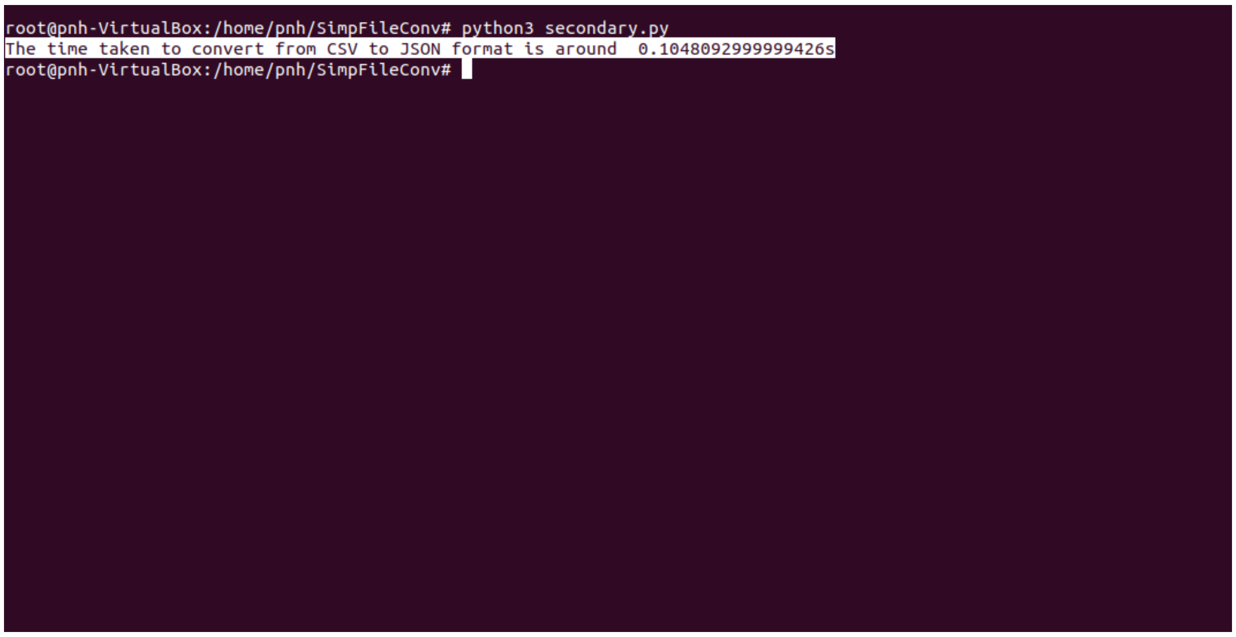


Fig: 5.3

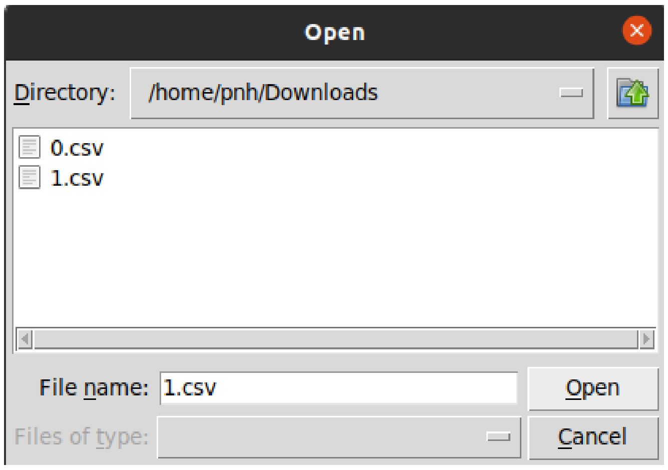
The file named as ‘converted’ located at the Desktop



```
root@pnh-VirtualBox:/home/pnh/SimpFileConv# python3 secondary.py
The time taken to convert from CSV to JSON format is around  0.10480929999999426s
root@pnh-VirtualBox:/home/pnh/SimpFileConv# 
```

Fig: 5.4

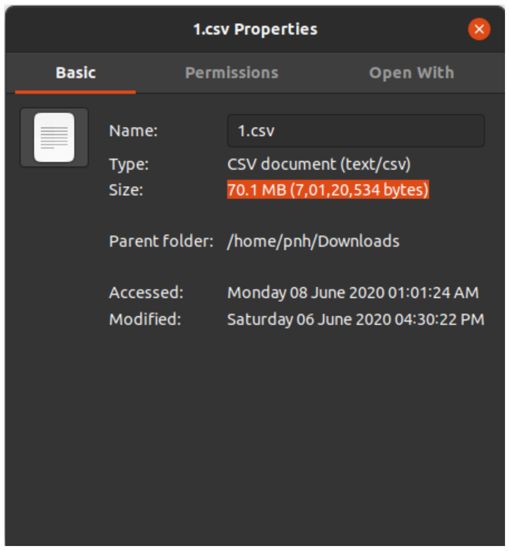
As mentioned before, time taken will be displayed in the Terminal after the conversion.



Directory: /home/pnh/Downloads

0.csv
1.csv

File name: 1.csv



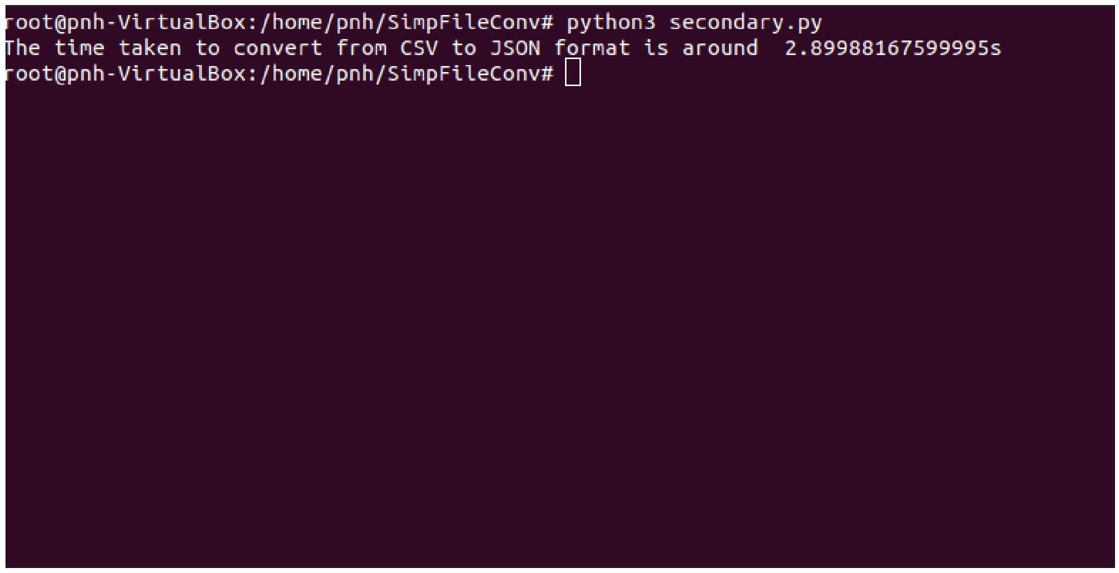
Name: 1.csv
Type: CSV document (text/csv)
Size: 70.1 MB (7,01,20,534 bytes)

Parent folder: /home/pnh/Downloads

Accessed: Monday 08 June 2020 01:01:24 AM
Modified: Saturday 06 June 2020 04:30:22 PM

Fig: 5.5 & Fig: 5.6

Now, a much bigger file is selected for conversion (See fig: 5.5 & 5.6) to observe the time taken for conversion. The newly selected file is around 70MB.



```
root@pnh-VirtualBox:/home/pnh/SimpFileConv# python3 secondary.py
The time taken to convert from CSV to JSON format is around  2.89988167599995s
root@pnh-VirtualBox:/home/pnh/SimpFileConv# 
```

Fig: 5.7

The reason for such long conversions is due to the large amount of characters that needs to be converted to the JSON format. Although it takes significantly less time than the primary implementation, the changes in time are still noticeable.

4.2.3 Secondary Implementation in Android Device:

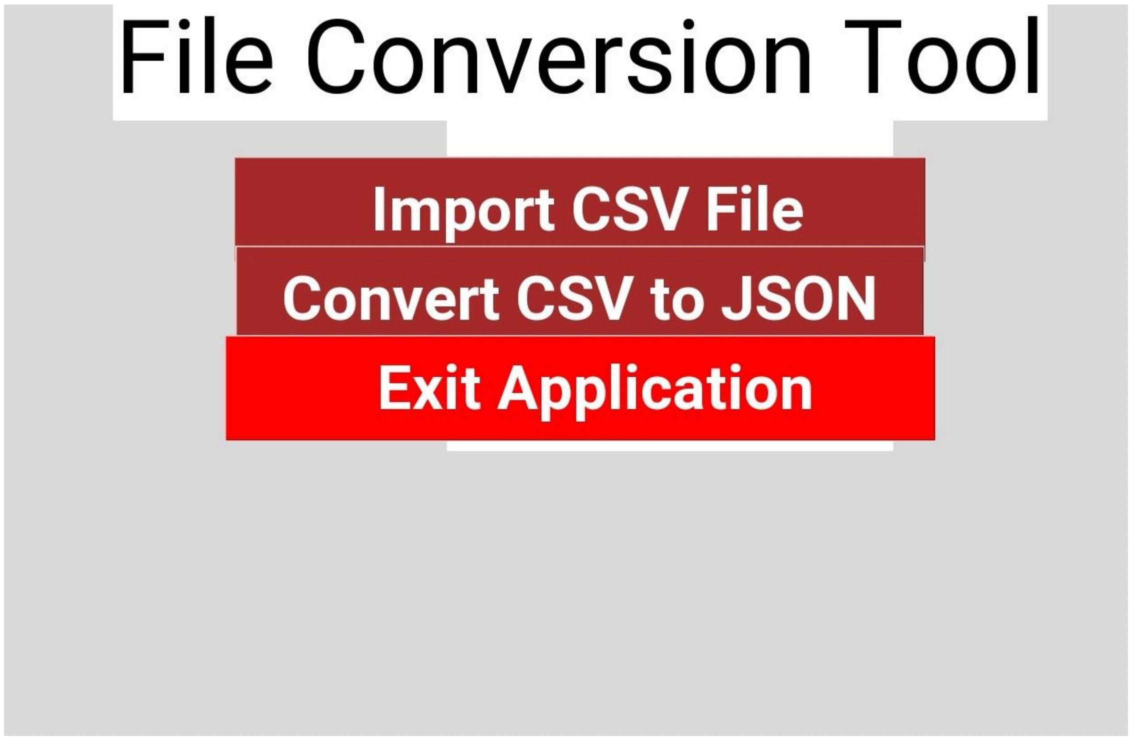


Fig: 6.1

The GUI appears this way in an Android Device

Unlike it’s primary implementation, the secondary implementation in Android cannot be run in Terminal. It has to be run in a suitable Python IDE(Pydroid3 here) as it supports GUI implementations. The Terminal however, does not support GUI applications yet.

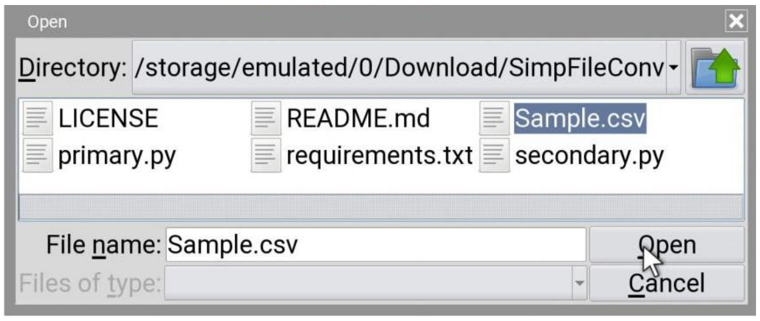


Fig: 6.2

The sample file is selected

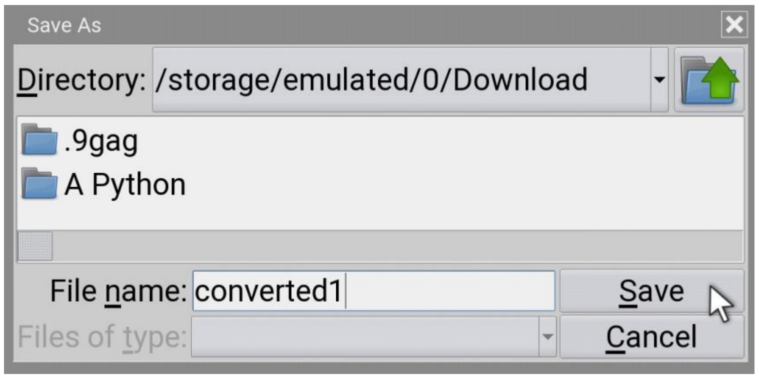
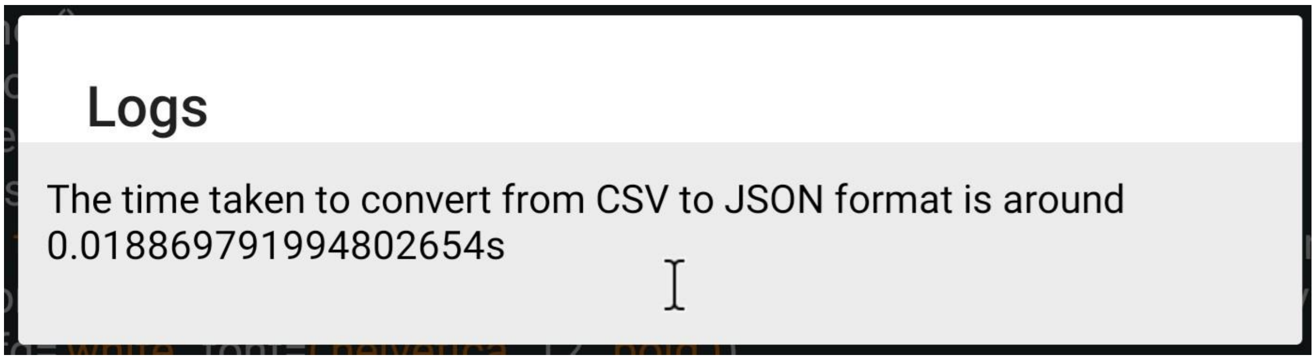


Fig: 6.3

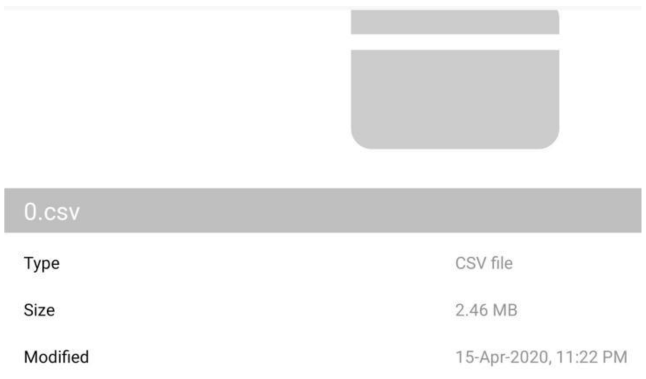
The output file is named as ‘converted1’


Logs

```
The time taken to convert from CSV to JSON format is around
0.018869791994802654s
```

Fig: 6.4

The time taken can be viewed from the Logs


0.csv

Type	CSV file
Size	2.46 MB
Modified	15-Apr-2020, 11:22 PM

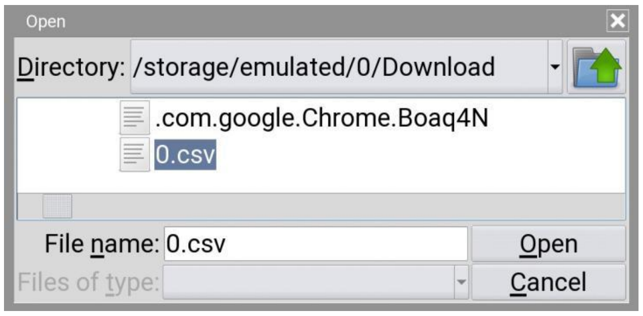

Open
Directory: /storage/emulated/0/Download
.com.google.Chrome.Boaq4N
0.csv
File name: 0.csv
Open
Files of type:
Cancel

Fig: 6.5 & Fig: 6.6

Much bigger files are selected

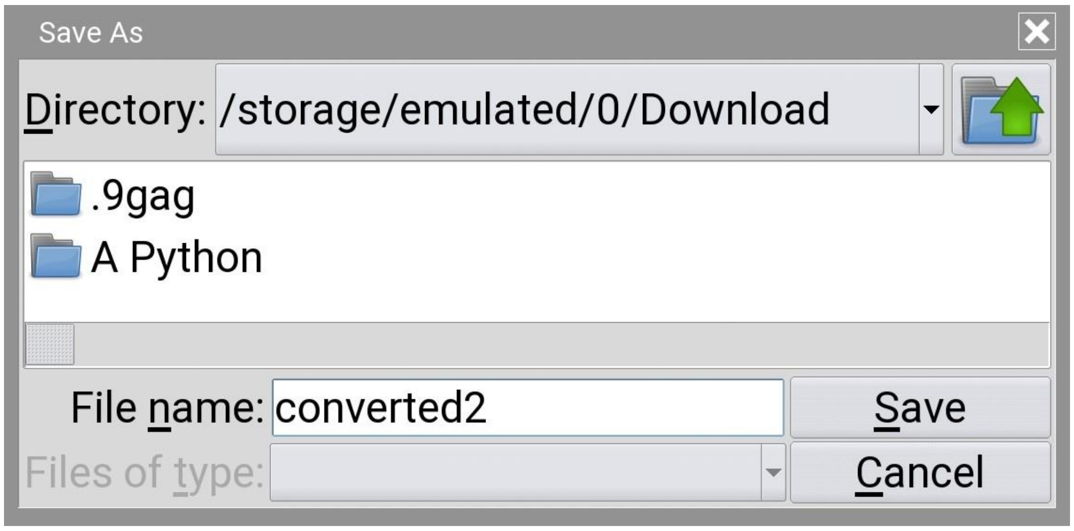
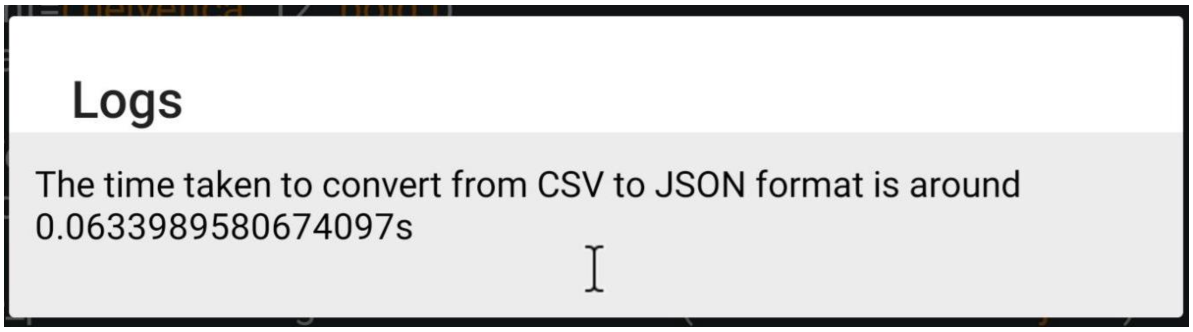


Fig: 6.7

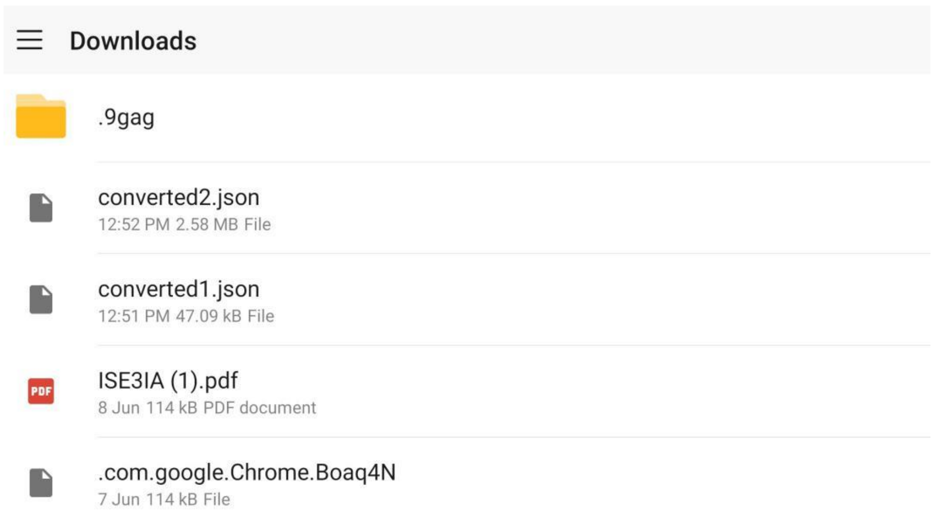
Output file is named as 'converted2'


Logs

```
The time taken to convert from CSV to JSON format is around
0.0633989580674097s
```

Fig: 6.8

Time taken here is longer than previous one even though there is no much difference


Downloads

.9gag

converted2.json
12:52 PM 2.58 MB File

converted1.json
12:51 PM 47.09 kB File

ISE3IA (1).pdf
8 Jun 114 kB PDF document

.com.google.Chrome.Boaq4N
7 Jun 114 kB File

Fig: 6.9

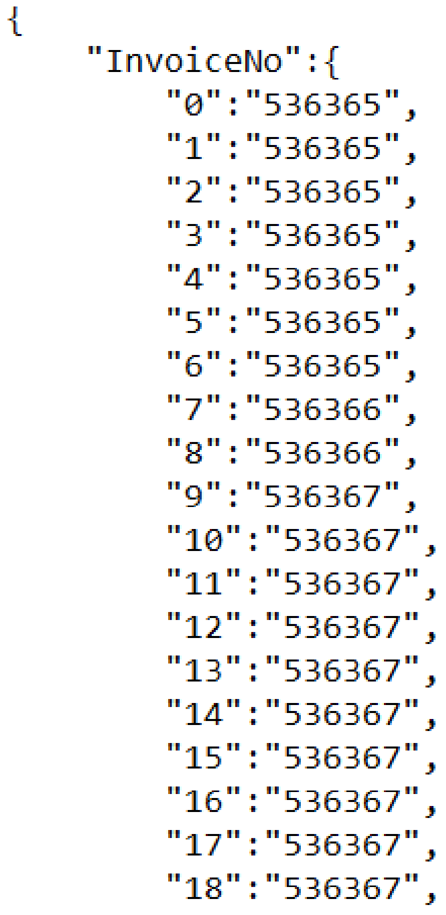
The converted files are saved in Downloads Folder

CHAPTER 5

OUTPUT

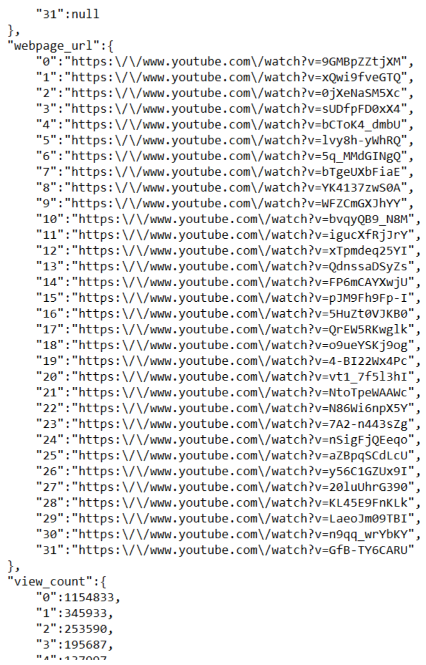
Output:

The below screenshots are some of the converted files done by GUI.



```
{
    "InvoiceNo":{
        "0":"536365",
        "1":"536365",
        "2":"536365",
        "3":"536365",
        "4":"536365",
        "5":"536365",
        "6":"536365",
        "7":"536366",
        "8":"536366",
        "9":"536367",
        "10":"536367",
        "11":"536367",
        "12":"536367",
        "13":"536367",
        "14":"536367",
        "15":"536367",
        "16":"536367",
        "17":"536367",
        "18":"536367",
```

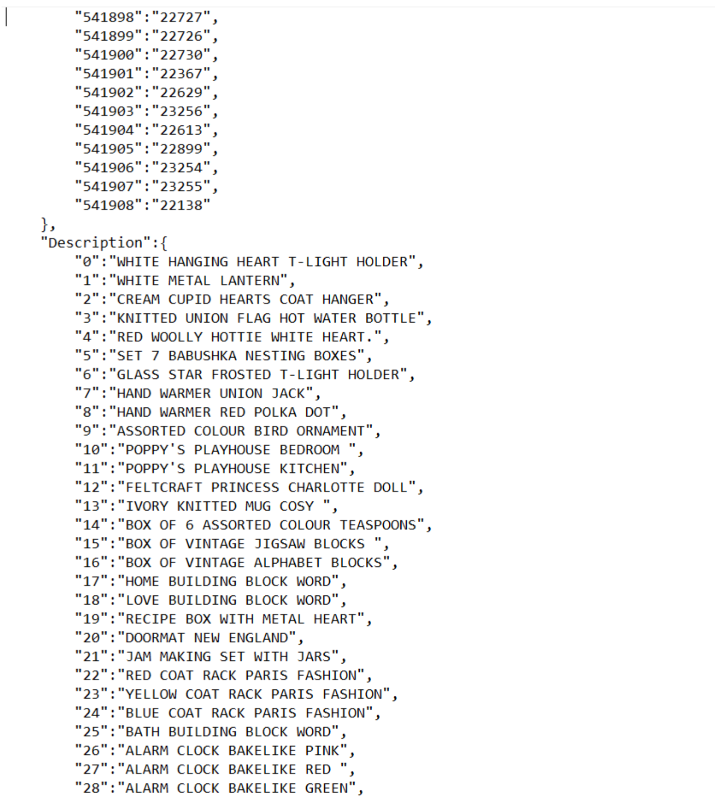
Fig: 7.1



```
        "31":null
    },
    "webpage_url":{
        "0":"https:\/\/www.youtube.com\/watch?v=9GMBpZZtjXM",
        "1":"https:\/\/www.youtube.com\/watch?v=xQwi9fveGTQ",
        "2":"https:\/\/www.youtube.com\/watch?v=0jXeNaSM5Xc",
        "3":"https:\/\/www.youtube.com\/watch?v=sUDfpFD0xX4",
        "4":"https:\/\/www.youtube.com\/watch?v=bCToK4_dmbU",
        "5":"https:\/\/www.youtube.com\/watch?v=lvy8h-yWhRQ",
        "6":"https:\/\/www.youtube.com\/watch?v=5q_MMdGINgQ",
        "7":"https:\/\/www.youtube.com\/watch?v=bTgeUXbFiaE",
        "8":"https:\/\/www.youtube.com\/watch?v=YK4137zwS0A",
        "9":"https:\/\/www.youtube.com\/watch?v=WFZCmGXJhYY",
        "10":"https:\/\/www.youtube.com\/watch?v=bvqyQB9_N8M",
        "11":"https:\/\/www.youtube.com\/watch?v=igucXfRjJrY",
        "12":"https:\/\/www.youtube.com\/watch?v=xTpmdeq25YI",
        "13":"https:\/\/www.youtube.com\/watch?v=QdnssaDSyZs",
        "14":"https:\/\/www.youtube.com\/watch?v=FP6mCAYXwjU",
        "15":"https:\/\/www.youtube.com\/watch?v=pJM9Fh9Fp-I",
        "16":"https:\/\/www.youtube.com\/watch?v=5HuZt0VJKB0",
        "17":"https:\/\/www.youtube.com\/watch?v=QrEW5RKwglk",
        "18":"https:\/\/www.youtube.com\/watch?v=o9ueYSKj9og",
        "19":"https:\/\/www.youtube.com\/watch?v=4-BI22Wx4Pc",
        "20":"https:\/\/www.youtube.com\/watch?v=vt1_7f5l3hI",
        "21":"https:\/\/www.youtube.com\/watch?v=NtoTpeWAAWc",
        "22":"https:\/\/www.youtube.com\/watch?v=N86Wi6npX5Y",
        "23":"https:\/\/www.youtube.com\/watch?v=7A2-n443sZg",
        "24":"https:\/\/www.youtube.com\/watch?v=nSigFjQEeqo",
        "25":"https:\/\/www.youtube.com\/watch?v=aZBpqSCdLcU",
        "26":"https:\/\/www.youtube.com\/watch?v=y56C1GZUx9I",
        "27":"https:\/\/www.youtube.com\/watch?v=20luUhrG390",
        "28":"https:\/\/www.youtube.com\/watch?v=KL45E9FnKLk",
        "29":"https:\/\/www.youtube.com\/watch?v=LaeoJm09TBI",
        "30":"https:\/\/www.youtube.com\/watch?v=n9qq_wrYbKY",
        "31":"https:\/\/www.youtube.com\/watch?v=GfB-TY6CARU"
    },
    "view_count":{
        "0":1154833,
        "1":345933,
        "2":253590,
        "3":195687,
        "4":137007
```

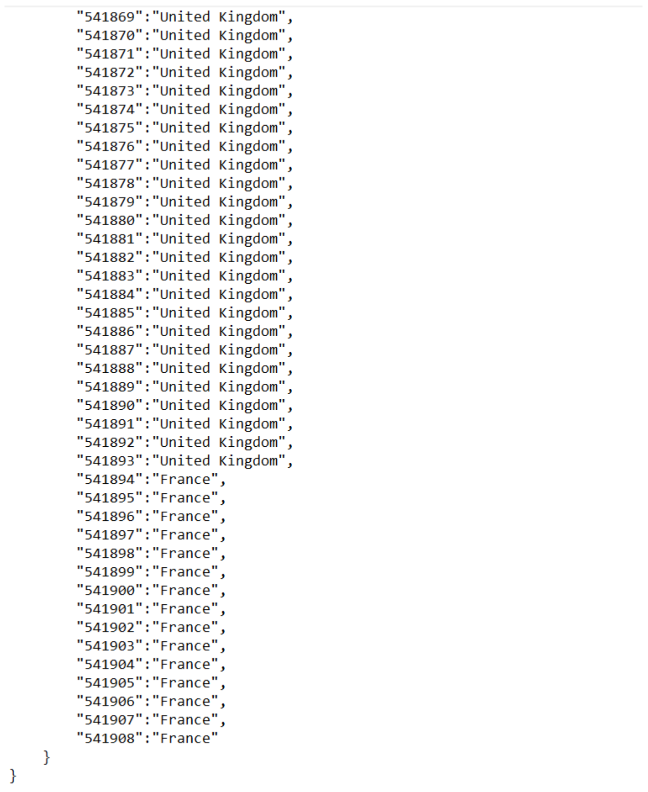
OUTPUT

Fig: 7.2



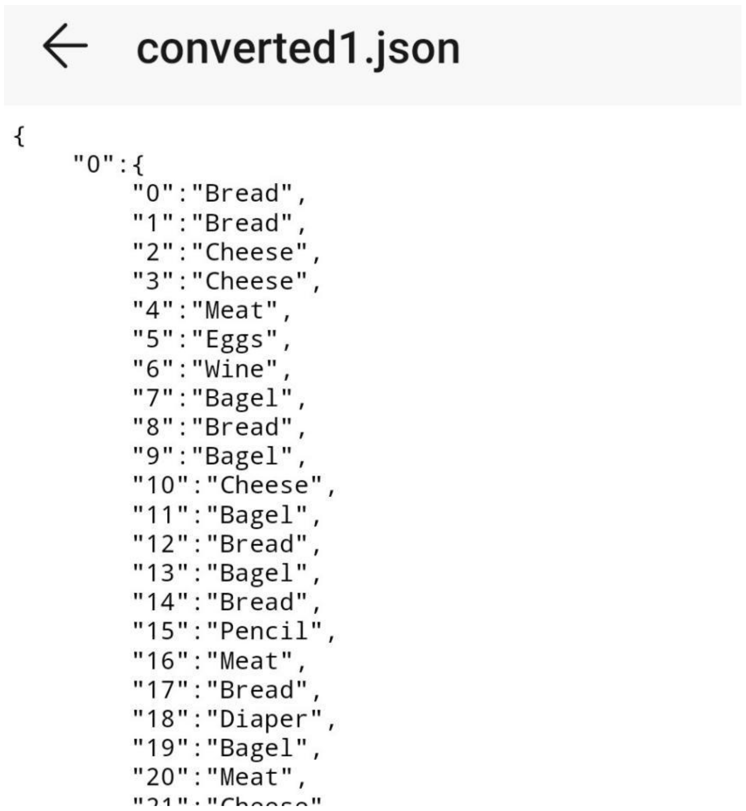
```
        "541898":"22727",
        "541899":"22726",
        "541900":"22730",
        "541901":"22367",
        "541902":"22629",
        "541903":"23256",
        "541904":"22613",
        "541905":"22899",
        "541906":"23254",
        "541907":"23255",
        "541908":"22138"
    },
    "Description":{
        "0":"WHITE HANGING HEART T-LIGHT HOLDER",
        "1":"WHITE METAL LANTERN",
        "2":"CREAM CUPID HEARTS COAT HANGER",
        "3":"KNITTED UNION FLAG HOT WATER BOTTLE",
        "4":"RED WOOLLY HOTTIE WHITE HEART.",
        "5":"SET 7 BABUSHKA NESTING BOXES",
        "6":"GLASS STAR FROSTED T-LIGHT HOLDER",
        "7":"HAND WARMER UNION JACK",
        "8":"HAND WARMER RED POLKA DOT",
        "9":"ASSORTED COLOUR BIRD ORNAMENT",
        "10":"POPPY'S PLAYHOUSE BEDROOM ",
        "11":"POPPY'S PLAYHOUSE KITCHEN",
        "12":"FELTCRAFT PRINCESS CHARLOTTE DOLL",
        "13":"IVORY KNITTED MUG COSY ",
        "14":"BOX OF 6 ASSORTED COLOUR TEASPOONS",
        "15":"BOX OF VINTAGE JIGSAW BLOCKS ",
        "16":"BOX OF VINTAGE ALPHABET BLOCKS",
        "17":"HOME BUILDING BLOCK WORD",
        "18":"LOVE BUILDING BLOCK WORD",
        "19":"RECIPE BOX WITH METAL HEART",
        "20":"DOORMAT NEW ENGLAND",
        "21":"JAM MAKING SET WITH JARS",
        "22":"RED COAT RACK PARIS FASHION",
        "23":"YELLOW COAT RACK PARIS FASHION",
        "24":"BLUE COAT RACK PARIS FASHION",
        "25":"BATH BUILDING BLOCK WORD",
        "26":"ALARM CLOCK BAKELIKE PINK",
        "27":"ALARM CLOCK BAKELIKE RED ",
        "28":"ALARM CLOCK BAKELIKE GREEN",
```

Fig: 7.3



```
        "541869":"United Kingdom",
        "541870":"United Kingdom",
        "541871":"United Kingdom",
        "541872":"United Kingdom",
        "541873":"United Kingdom",
        "541874":"United Kingdom",
        "541875":"United Kingdom",
        "541876":"United Kingdom",
        "541877":"United Kingdom",
        "541878":"United Kingdom",
        "541879":"United Kingdom",
        "541880":"United Kingdom",
        "541881":"United Kingdom",
        "541882":"United Kingdom",
        "541883":"United Kingdom",
        "541884":"United Kingdom",
        "541885":"United Kingdom",
        "541886":"United Kingdom",
        "541887":"United Kingdom",
        "541888":"United Kingdom",
        "541889":"United Kingdom",
        "541890":"United Kingdom",
        "541891":"United Kingdom",
        "541892":"United Kingdom",
        "541893":"United Kingdom",
        "541894":"France",
        "541895":"France",
        "541896":"France",
        "541897":"France",
        "541898":"France",
        "541899":"France",
        "541900":"France",
        "541901":"France",
        "541902":"France",
        "541903":"France",
        "541904":"France",
        "541905":"France",
        "541906":"France",
        "541907":"France",
        "541908":"France"
    }
}
```

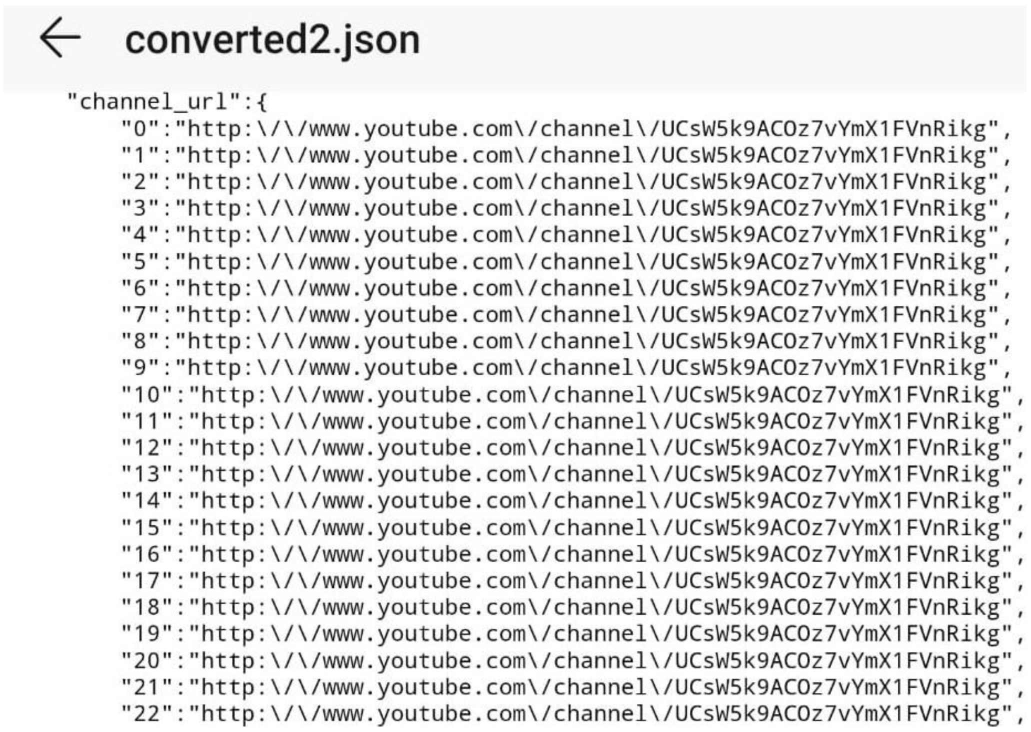
Fig: 7.4



converted1.json

```
{
    "0":{
        "0":"Bread",
        "1":"Bread",
        "2":"Cheese",
        "3":"Cheese",
        "4":"Meat",
        "5":"Eggs",
        "6":"Wine",
        "7":"Bagel",
        "8":"Bread",
        "9":"Bagel",
        "10":"Cheese",
        "11":"Bagel",
        "12":"Bread",
        "13":"Bagel",
        "14":"Bread",
        "15":"Pencil",
        "16":"Meat",
        "17":"Bread",
        "18":"Diaper",
        "19":"Bagel",
        "20":"Meat",
        "21":"Cheese"
```

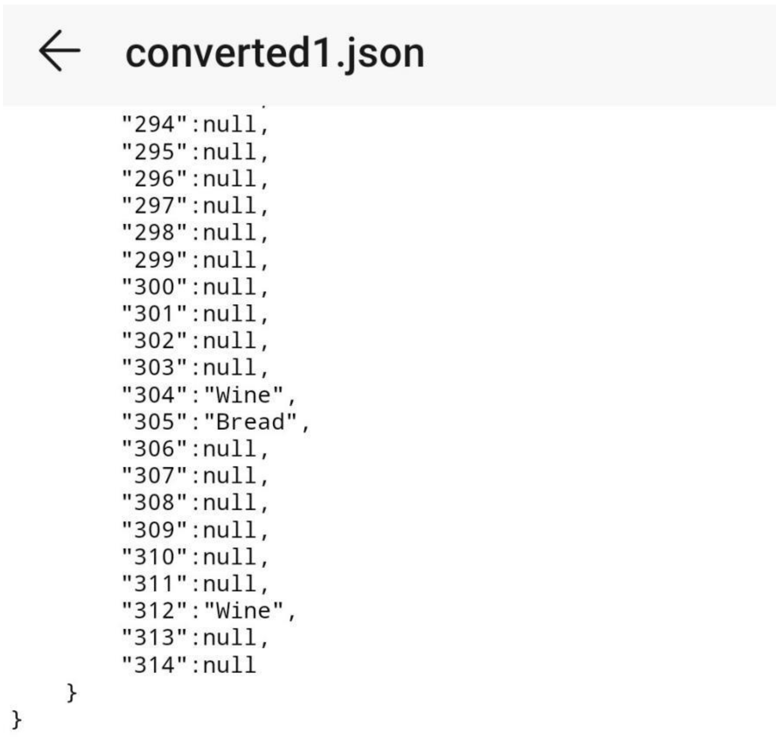
Fig: 7.5



converted2.json

```
"channel_url":{
    "0":"http:\/\/www.youtube.com\/channel\/UCsW5k9ACOz7vYmX1FVnRikg",
    "1":"http:\/\/www.youtube.com\/channel\/UCsW5k9ACOz7vYmX1FVnRikg",
    "2":"http:\/\/www.youtube.com\/channel\/UCsW5k9ACOz7vYmX1FVnRikg",
    "3":"http:\/\/www.youtube.com\/channel\/UCsW5k9ACOz7vYmX1FVnRikg",
    "4":"http:\/\/www.youtube.com\/channel\/UCsW5k9ACOz7vYmX1FVnRikg",
    "5":"http:\/\/www.youtube.com\/channel\/UCsW5k9ACOz7vYmX1FVnRikg",
    "6":"http:\/\/www.youtube.com\/channel\/UCsW5k9ACOz7vYmX1FVnRikg",
    "7":"http:\/\/www.youtube.com\/channel\/UCsW5k9ACOz7vYmX1FVnRikg",
    "8":"http:\/\/www.youtube.com\/channel\/UCsW5k9ACOz7vYmX1FVnRikg",
    "9":"http:\/\/www.youtube.com\/channel\/UCsW5k9ACOz7vYmX1FVnRikg",
    "10":"http:\/\/www.youtube.com\/channel\/UCsW5k9ACOz7vYmX1FVnRikg",
    "11":"http:\/\/www.youtube.com\/channel\/UCsW5k9ACOz7vYmX1FVnRikg",
    "12":"http:\/\/www.youtube.com\/channel\/UCsW5k9ACOz7vYmX1FVnRikg",
    "13":"http:\/\/www.youtube.com\/channel\/UCsW5k9ACOz7vYmX1FVnRikg",
    "14":"http:\/\/www.youtube.com\/channel\/UCsW5k9ACOz7vYmX1FVnRikg",
    "15":"http:\/\/www.youtube.com\/channel\/UCsW5k9ACOz7vYmX1FVnRikg",
    "16":"http:\/\/www.youtube.com\/channel\/UCsW5k9ACOz7vYmX1FVnRikg",
    "17":"http:\/\/www.youtube.com\/channel\/UCsW5k9ACOz7vYmX1FVnRikg",
    "18":"http:\/\/www.youtube.com\/channel\/UCsW5k9ACOz7vYmX1FVnRikg",
    "19":"http:\/\/www.youtube.com\/channel\/UCsW5k9ACOz7vYmX1FVnRikg",
    "20":"http:\/\/www.youtube.com\/channel\/UCsW5k9ACOz7vYmX1FVnRikg",
    "21":"http:\/\/www.youtube.com\/channel\/UCsW5k9ACOz7vYmX1FVnRikg",
    "22":"http:\/\/www.youtube.com\/channel\/UCsW5k9ACOz7vYmX1FVnRikg",
```

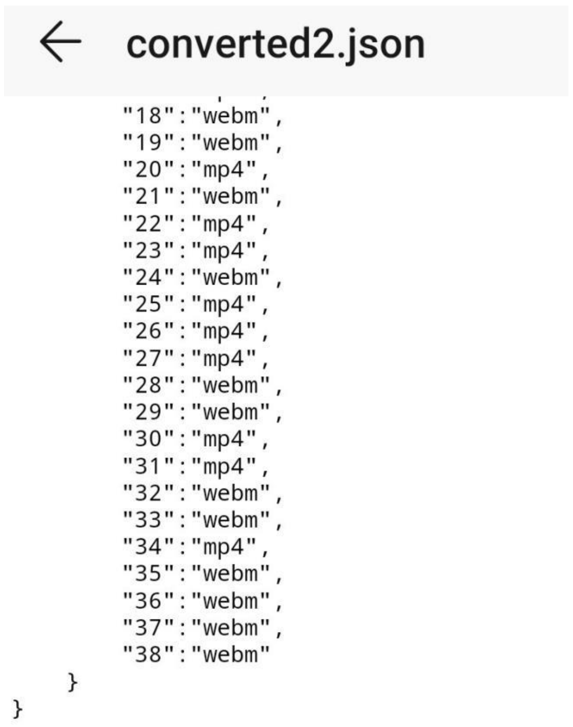
Fig: 7.6



converted1.json

```
        "294":null,
        "295":null,
        "296":null,
        "297":null,
        "298":null,
        "299":null,
        "300":null,
        "301":null,
        "302":null,
        "303":null,
        "304":"Wine",
        "305":"Bread",
        "306":null,
        "307":null,
        "308":null,
        "309":null,
        "310":null,
        "311":null,
        "312":"Wine",
        "313":null,
        "314":null
    }
}
```

Fig: 7.7



converted2.json

```
        "18":"webm",
        "19":"webm",
        "20":"mp4",
        "21":"webm",
        "22":"mp4",
        "23":"mp4",
        "24":"webm",
        "25":"mp4",
        "26":"mp4",
        "27":"mp4",
        "28":"webm",
        "29":"webm",
        "30":"mp4",
        "31":"mp4",
        "32":"webm",
        "33":"webm",
        "34":"mp4",
        "35":"webm",
        "36":"webm",
        "37":"webm",
        "38":"webm"
    }
}
```

Fig: 7.8

6.1 CONCLUSION:

In conclusion, data conversion is an integral concept in the computing world and knowingly/unknowingly, it happens everyday both internally and externally. In providing a general overview on how Data Conversion works, it is hoped that this article allows end users to understand and make use of this wonderful concept effectively both personally and professionally as how every good concept must be used.

6.2 REFERNCES:

1) Python 3 Documentation for various libraries:

https://docs.python.org/3/

2) Graphical User Interface for Conversion:

https://datatofish.com/csv-to-json-string-python/

3) Various issues were solved with the help of :

Stack Overflow - Where Developers Learn, Share, & Build Careers

https://stackoverflow.com/

4) Pandas Documentation:

https://pandas.pydata.org/pandas-docs/version/0.25.3/

5) Book:- File Organization by Binnur Kurt

6) Data Conversion Journals:

http://www.ijecs.in/index.php/ijecs/article/view/2574/2379

OUTPUT