



Estd. 2000

ABESEC Ghaziabad
Department of Computer Science & Engineering
Mini Project REPORT- KCS-354
(2019-20)

Project Title: UNIT CONVERTER				
Project Type Application				
	Name	Roll Number	Section	Signature
Group member	Isha Goel	1803210067	CS-A	
Project Guide	Mr.Priyansh Singh Mr.Rohit Rastogi	Remarks:		
Signature				
Date of submission				

Contents

1.1. Problem Introduction.....	3
1.1.1. Motivation.....	3
1.1.2. Project Objective.....	3
1.1.3. Scope of the project.....	3
1.2. Related previous work.....	4
1.3. Software and Hardware requirements.....	6
1.4. Proposed Method.....	7
1.5. Deliverables.....	11
1.6. Stakeholders.....	20
1.7. Gantt chart.....	20
1.8. References.....	21

PROBLEM INTRODUCTION

- **MOTIVATION**

The main motivation to develop this program is the summer training that was undertaken under the guidance of college management.

- **PROJECT OBJECTIVE**

Conversion of units is the conversion between different units of measurement for the same quantity through multiplicative conversion factors and this program serves the purpose. It would require the user to input the unit he wishes to convert and would convert it into different standard units.

A conversion factor is used to change the units of measured quantity without changing its value.

- **SCOPE OF THE PROJECT**

The program consists of eight different converters as follows:

- LENGTH CONVERTER
- WEIGHT CONVERTER
- VOLUME CONVERTER
- AREA CONVERTER
- CURRENCY CONVERTER
- TIME CONVERTER
- SPEED CONVERTER
- FORCE CONVERTER

Each one of them requires the user to input the magnitude and unit of the desired value he wants to convert and would produce the converted most commonly used standard units.

The input can be either entered through keyboard or through the SPEECH RECOGNITION SYSTEM. The speech recognition system facilitates the user to get the required output merely by using the speech to text feature. The user may also see the search history whenever he requires the previously searched values as it stores all the output along with the input provided by the user.

Related Previous work

- **UNIT CONVERTER:** (Smart tools co.)

Unit converter is the 6th set of Smart Tools collection. This app includes Currency (money, bitcoin) exchange rates.

There are a lot of unit conversion apps on the market. However, most are inconvenient and difficult to use because of poor and complicated UI. This app has intuitive and simple UI, that is designed for anybody to use.

There are 4 categories.

- Basic : length (distance), area, weight (mass), volume (capacity)
- Living : exchange rate, temperature, time, speed, shoes, clothing, hat, ring
- Science : pressure, force, work (energy), power, torque, flow, current, voltage, density, viscosity, concentration, astronomy
- Misc. : angle, data, fuel efficiency, cooking, illuminance, radiation, prefix, binary, time zone, blood sugar, hardness, AWG

- **UNIT CONVERTER:** (Digital Grove Tool)

Built-In real time Currency Converter with 150 world currencies and their latest exchange rates

Smart Tools - Bubble Level, Compass, Protractor, Resistor codes, Stop Watch, Ruler, World Time, Date Converter and much more.

New Tools - Battery Monitor, Notes, Expression Evaluation, Equation Solver, Induction Color codes, Scientific Calculator

Financial calculators - Loan calculator, Compound interest calculator, Retirement calculator, Service tax calculator, Stock calculator

Math calculators - Roman numeral converter, Number base converter, Number series generator, Ratio, Fraction, Proportion, etc.

It has most important conversion tools that are used in daily life including Fuel Calculations, Temperature, Volume, Speed, Weight, Computer Storage, Angle, Power, Viscosity, Force, Energy, Torque, Density.

It has conversion tools like radiation, electric resistance, electric capacitance, inductance, inertia, specific heat density, specific heat capacity, and illumination.

- **FULL UNIT CONVERTER:** (Haytham Ayyash Tools)

-It contain many unit conversions and includes the following conversions:

-Distance.

-Area.

-Volume.

-Weight.

-Time.

-Temperature.

-Power.

-Speed.

-Energy.

-Force.

-Pressure.

-Cooking.

-Fuel.

-Digital Storage.

-Number of decimal : you can control of number of decimal .

HARDWARE REQUIREMENT:

For this project nothing but a PC is required.

SOFTWARE REQUIREMENT:

To begin with this project, a system well versed with Python and Anaconda navigator is required. However any other IDE would also do.

Mine is installed with Python version 3.7.4 and anaconda navigator as it supports Jupyter notebook, it makes the code presentable and increases readability of the code.

It also supports Spyder which is an interactive IDE for python programs.

IDE USED:

In developing this project Anaconda(Jupyter Notebook) was used.

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment.

ANACONDA NAVIGATOR:

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, macOS and Linux.

The following applications are available by default in Navigator:

- JupyterLab
- Jupyter Notebook
- QtConsole
- Spyder
- Glueviz and others....

PROPOSED METHOD:

ALGORITHM / APPROACH

Step 1: User is given choices as to which converter he wants to use.

Step 2: The choice is input through keyboard and stored in a variable, say ch.

Step 3: if $ch==1$, direct him to the length converter

- Declare lists for different units with their conversion factors and units in them.
- Say meter, $list1 = ["1000000", "micrometer", "100", "cm", "0.001", "km" \dots]$.
- Ask the user what magnitude which unit he wants to convert and store it in a variable, say a.
- Traverse through the list with a for loop and perform calculations with magnitude and conversion factors.
- Display the conversions along with the proper units.

Step 4: if $ch==2$, direct him to the weight converter

- Declare lists for different units with their conversion factors and units in them.
- Say gram, $list1 = ["1000", "mg", "0.001", "kg" \dots]$.
- Ask the user what magnitude which unit he wants to convert and store it in a variable, say a.
- Traverse through the list with a for loop and perform calculations with magnitude and conversion factors.
- Display the conversions along with the proper units.

Step 5: if $ch==3$, direct him to the volume converter

- Declare lists for different units with their conversion factors and units in them.
- Say m^3 , $list1 = [\dots]$.
- Ask the user what magnitude which unit he wants to convert and store it in a variable, say a.
- Traverse through the list with a for loop and perform calculations with magnitude and conversion factors.
- Display the conversions along with the proper units.

Step 6: if $ch==4$, direct him to the area converter

- Declare lists for different units with their conversion factors and units in them.
- Say m^2 , $list1 = []$.
- Ask the user what magnitude which unit he wants to convert and store it in a variable, say a.
- Traverse through the list with a for loop and perform calculations with magnitude and conversion factors.

- Display the conversions along with the proper units.

Step 7: if `ch==5`, direct him to the currency converter

- Declare lists for different units with their conversion factors and units in them.
- Say INR, `list1= [...]`.
- Ask the user what magnitude which unit he wants to convert and store it in a variable, say `a`.
- Traverse through the list with a for loop and perform calculations with magnitude and conversion factors.
- Display the conversions along with the proper units.

Step 8: if `ch==6`, direct him to the time converter

- Declare lists for different units with their conversion factors and units in them.
- Say minute, `list1= [...]`.
- Ask the user what magnitude which unit he wants to convert and store it in a variable, say `a`.
- Traverse through the list with a for loop and perform calculations with magnitude and conversion factors.
- Display the conversions along with the proper units.

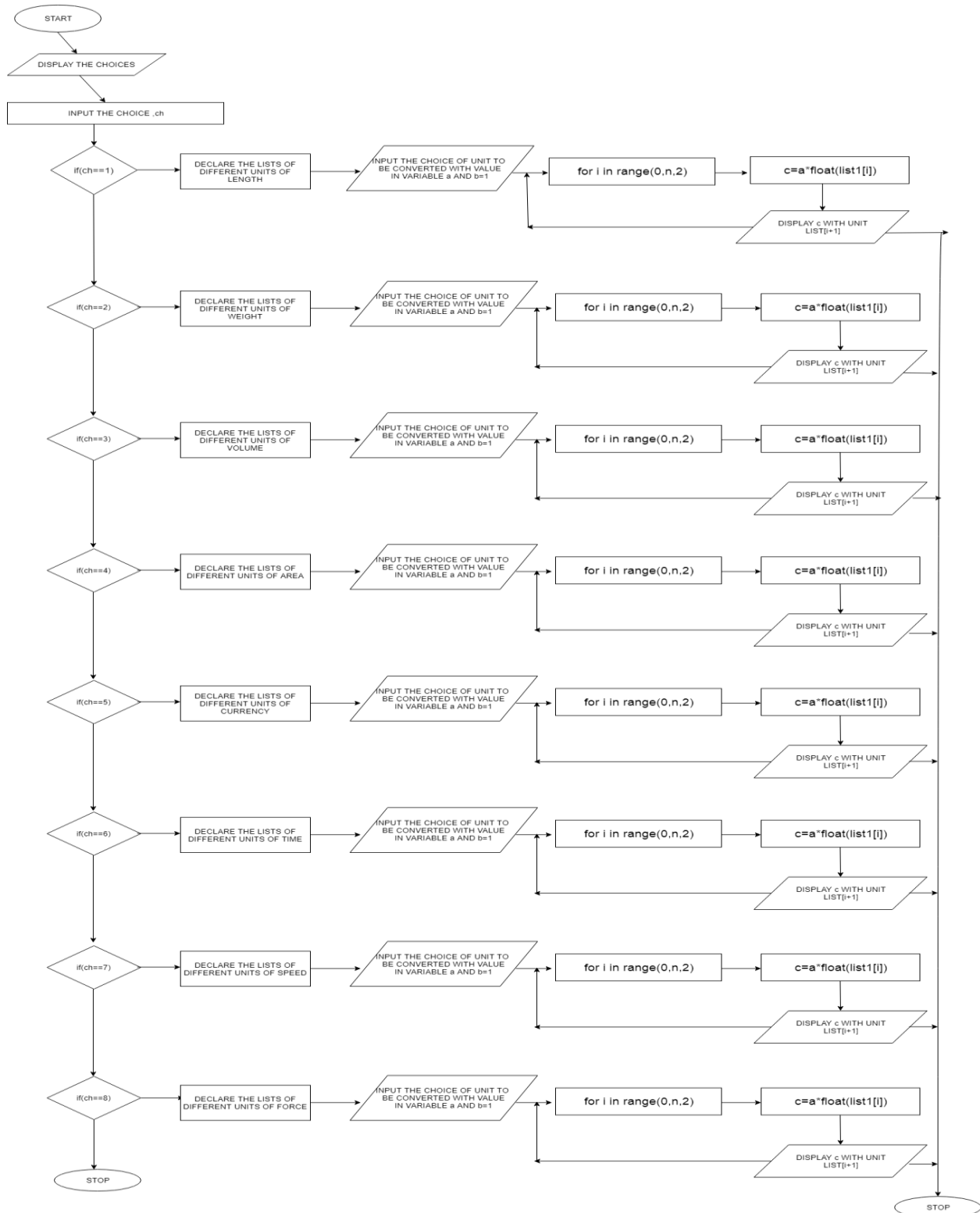
Step 9: if `ch==7`, direct him to the speed converter

- Declare lists for different units with their conversion factors and units in them.
- Say m/s, `list1= [...]`.
- Ask the user what magnitude which unit he wants to convert and store it in a variable, say `a`.
- Traverse through the list with a for loop and perform calculations with magnitude and conversion factors.
- Display the conversions along with the proper units.

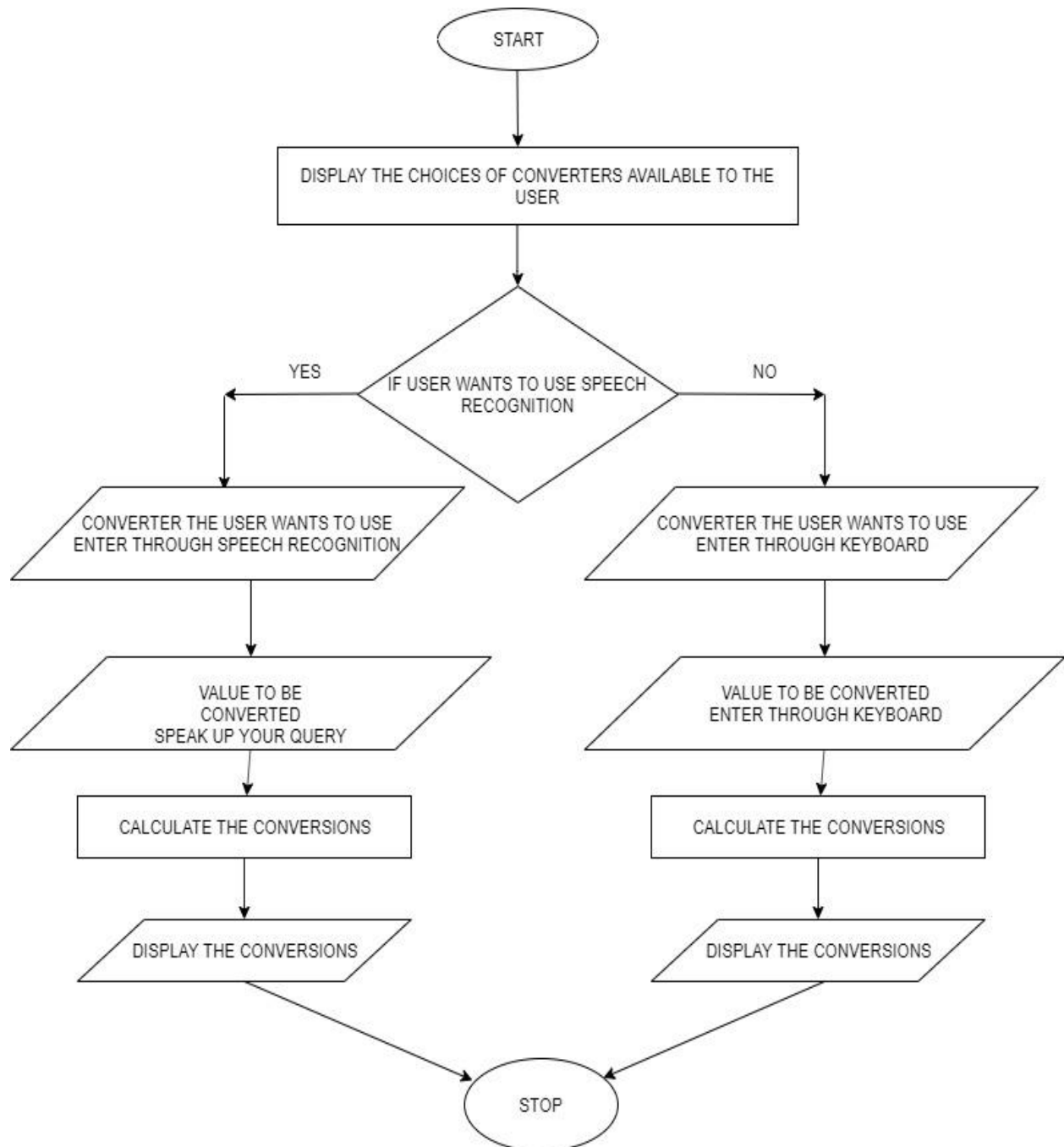
Step 10: if `ch==8`, direct him to the force converter

- Declare lists for different units with their conversion factors and units in them.
- Say N, `list1= [...]`.
- Ask the user what magnitude which unit he wants to convert and store it in a variable, say `a`.
- Traverse through the list with a for loop and perform calculations with magnitude and conversion factors.
- Display the conversions along with the proper units.

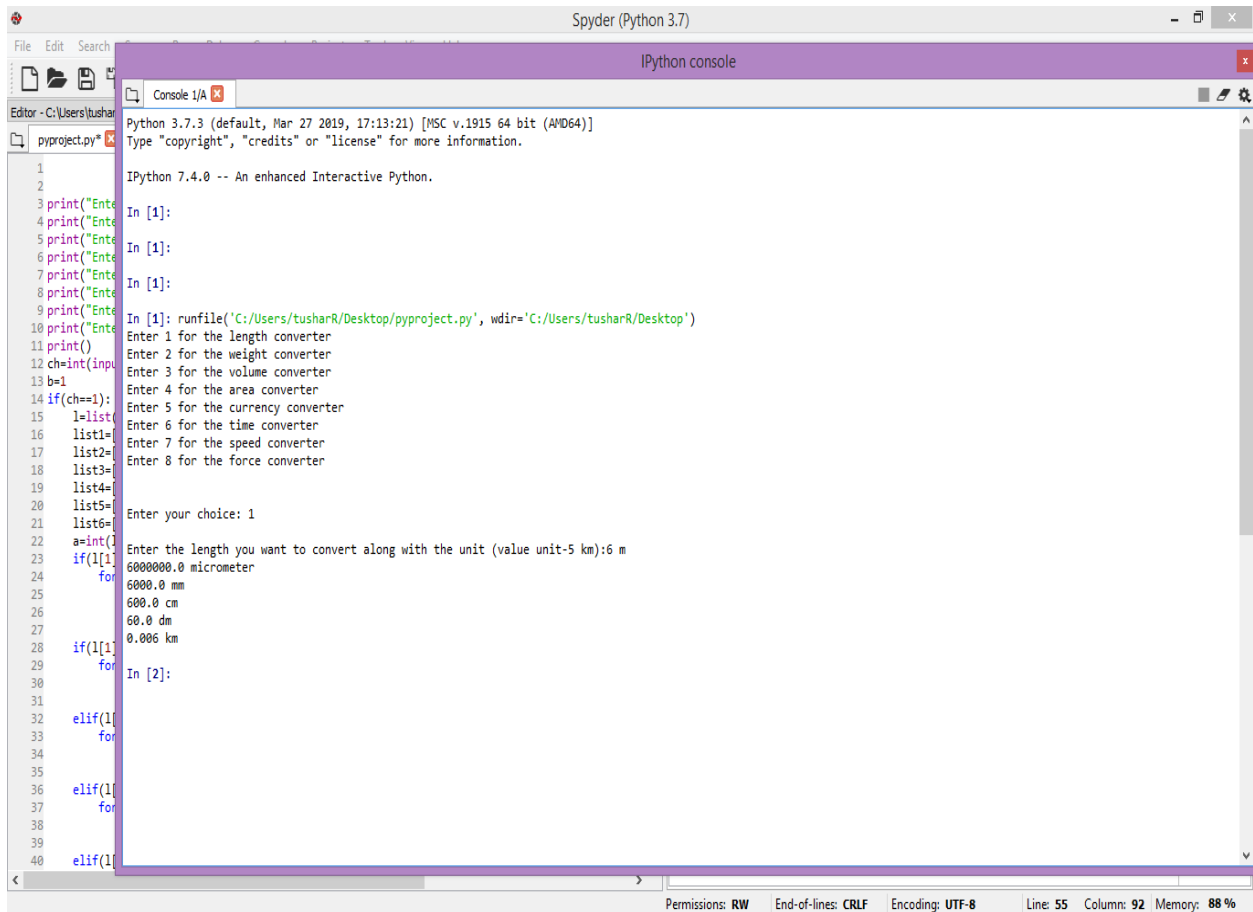
FLOW CHART



WORK FLOW DIAGRAM



DELIVERABLES:



The screenshot displays the Spyder Python IDE interface. The editor window on the left contains a Python script named 'pyproject.py' which implements a length converter. The script prompts the user to enter a choice (1-8) and a length value, then displays the equivalent measurements in various units. The IPython console on the right shows the execution of the script, including the user's input '1' for the choice and '6' for the length, resulting in the output: '6000000.0 micrometer', '6000.0 mm', '600.0 cm', '60.0 dm', and '0.006 km'.

```
Python 3.7.3 (default, Mar 27 2019, 17:13:21) [MSC v.1915 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.4.0 -- An enhanced Interactive Python.

In [1]:
In [1]:
In [1]:
In [1]:
In [1]: runfile('C:/Users/tushar/Desktop/pyproject.py', wdir='C:/Users/tushar/Desktop')
Enter 1 for the length converter
Enter 2 for the weight converter
Enter 3 for the volume converter
Enter 4 for the area converter
Enter 5 for the currency converter
Enter 6 for the time converter
Enter 7 for the speed converter
Enter 8 for the force converter

Enter your choice: 1
Enter the length you want to convert along with the unit (value unit-5 km):6 m
6000000.0 micrometer
6000.0 mm
600.0 cm
60.0 dm
0.006 km

In [2]:
elif(1)
for
elif(1)
for
elif(1)
for
elif(1)
for
```

Fig 1. Length Converter

The screenshot shows the Spyder Python IDE interface. The editor window on the left contains a Python script for a weight converter. The script prompts the user to choose a conversion type (1-8) and then asks for the weight to convert. It uses a list to store conversion factors and a dictionary to map choices to units. The IPython console on the right shows the execution of the script, including the user's input '2' for the weight converter and the resulting output for converting 8.2 g to microgram, mg, kg, lb, and oz.

```
File Edit Search View Help
pyproject.py
239 elif(l[1] == 1):
240     for i in range(1, len(list1)):
241         Enter 1 for the length converter
242         Enter 2 for the weight converter
243         Enter 3 for the volume converter
244         Enter 4 for the area converter
245     elif(l[1] == 2):
246         for i in range(1, len(list2)):
247             Enter 5 for the currency converter
248             Enter 6 for the time converter
249             Enter 7 for the speed converter
250             Enter 8 for the force converter
251         Enter your choice: 2
252     elif(l[1] == 3):
253         for i in range(1, len(list3)):
254             Enter the weight you want to convert along with the unit (5 kg):8.2 g
255             8199999.999999999 microgram
256             8200.0 mg
257             0.008199999999999999 kg
258             0.018080999999999996 lb
259             0.28924679999999997 oz
260         In [4]:
261         a=float(input("Enter the weight you want to convert: "))
262         if(l[1] == 1):
263             for i in range(1, len(list1)):
264                 if(l[1] == 1):
265                     for i in range(1, len(list1)):
266                         elif(l[1] == 2):
267                             for i in range(1, len(list2)):
268                                 elif(l[1] == 3):
269                                     for i in range(1, len(list3)):
270                                         elif(l[1] == 4):
271                                             for i in range(1, len(list4)):
272                                                 elif(l[1] == 5):
273                                                     for i in range(1, len(list5)):
274                                                         elif(l[1] == 6):
275                                                             for i in range(1, len(list6)):
276                                                                 elif(l[1] == 7):
277                                                                     for i in range(1, len(list7)):
278                                                                         elif(l[1] == 8):
279                                                                             for i in range(1, len(list8)):
```

Fig 2. Weight Converter

The screenshot shows the Spyder Python IDE interface. The editor on the left contains a Python script named 'pyproject.py' with the following visible code:

```
239 elif(l1==1):
240     for
241         Enter 1 for the length converter
242         Enter 2 for the weight converter
243         Enter 3 for the volume converter
244         Enter 4 for the area converter
245         Enter 5 for the currency converter
246         Enter 6 for the time converter
247         Enter 7 for the speed converter
248         Enter 8 for the force converter
249
250 Enter your choice: 3
251
252 Enter the volume you want to convert along with the unit (5 L):9 ml
253 0.009000000000000001 L
254 9000.0 mm^3
255 9.0 cm^3
256 0.009000000000000001 dm^3
257 9e-06 m^3
258
259 if(ch==8):
260     l=list
261     list1=
262     list2=
263     list3=
264     list4=
265     list5=
266
267 a=float
268 if(l1==1):
269     for
270
271 elif(l1==2):
272     for
273
274 elif(l1==3):
275     for
276
277 elif(l1==4):
278     for
```

The IPython console on the right shows the execution of the script:

```
In [4]: runfile('C:/Users/tusharR/Desktop/pyproject.py', wdir='C:/Users/tusharR/Desktop')
Enter 1 for the length converter
Enter 2 for the weight converter
Enter 3 for the volume converter
Enter 4 for the area converter
Enter 5 for the currency converter
Enter 6 for the time converter
Enter 7 for the speed converter
Enter 8 for the force converter
Enter your choice: 3
Enter the volume you want to convert along with the unit (5 L):9 ml
0.009000000000000001 L
9000.0 mm^3
9.0 cm^3
0.009000000000000001 dm^3
9e-06 m^3
In [5]: |
```

The status bar at the bottom indicates: Permissions: RW, End-of-lines: CRLF, Encoding: UTF-8, Line: 263, Column: 12, Memory: 87 %.

Fig 3. Volume Converter

```
File Edit Search
Spyder (Python 3.7)
IPython console
Console 2/A
pyproject.py
In [5]: runfile('C:/Users/tusharR/Desktop/pyproject.py', wdir='C:/Users/tusharR/Desktop')
Enter 1 for the length converter
Enter 2 for the weight converter
Enter 3 for the volume converter
Enter 4 for the area converter
Enter 5 for the currency converter
Enter 6 for the time converter
Enter 7 for the speed converter
Enter 8 for the force converter
Enter your choice: 4
Enter the area you want to convert along with the unit (5 m^2):9.5 m^2
950000.0 mm^2
95000.0 cm^2
950.0 dm^2
9.499999999999999e-06 km^2
In [6]: |
239 elif(l[1]
240 for
241 Enter 2 for the weight converter
242 Enter 3 for the volume converter
243 elif(l[1]
244 for
245 Enter 7 for the speed converter
246 Enter 8 for the force converter
247 elif(l[1]
248 for
249 Enter your choice: 4
250 Enter the area you want to convert along with the unit (5 m^2):9.5 m^2
251 elif(l[1]
252 for
253 95000.0 cm^2
254 950.0 dm^2
255 9.499999999999999e-06 km^2
256 if(ch==8):
257 l=list
258 list1=
259 list2=
260 list3=
261 list4=
262 list5=
263 a=float
264 if(l[1]
265 for
266
267
268 elif(l[1]
269 for
270
271
272 elif(l[1]
273 for
274
275
276 elif(l[1]
277 for
278
Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 250 Column: 18 Memory: 87 %
```

Fig 4. Area Converter

The screenshot shows the Spyder Python IDE interface. The left pane displays a Python script for a currency converter. The script includes a menu for selecting conversion types (length, weight, volume, area, currency, time, speed, force) and a specific section for currency conversion. The right pane shows the IPython console with the script's execution output, including prompts for user input and the resulting conversion rates for INR, USD, EUR, JPY, and AUD.

```
142 In [7]: runfile('C:/Users/tushar/Desktop/pyproject.py', wdir='C:/Users/tushar/Desktop')
143 Enter 1 for the length converter
144 Enter 2 for the weight converter
145 Enter 3 for the volume converter
146 Enter 4 for the area converter
147 Enter 5 for the currency converter
148 Enter 6 for the time converter
149 Enter 7 for the speed converter
150 Enter 8 for the force converter
151
152 Enter your choice: 5
153
154 Enter the Currency(INR,USD,EUR,JPY,AUD) you want to convert along with the unit ($ INR):6 INR
155 0.06 USD
156 0.06 EUR
157 9.42 JPY
158 0.12 AUD
159
160 In [8]: |
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177 if(ch==6):
178     l=list()
179     list1=
180     list2=
181     list3=
```

Fig 5. Currency converter

```
ipython (python 3.7)
File Edit Search
Console 2/k
Editor - C:\Users\tushar\pyproject.py
pyproject.py
172 In [8]: runfile('C:/Users/tushar/Desktop/pyproject.py', wdir='C:/Users/tushar/Desktop')
173 Enter 1 for the length converter
174 Enter 2 for the weight converter
175 elif(1) Enter 3 for the volume converter
176 for Enter 4 for the area converter
177 Enter 5 for the currency converter
178 Enter 6 for the time converter
179 if(ch==6): Enter 7 for the speed converter
180 Enter 8 for the force converter
181 l=list()
182 list1=
183 list2=
184 list3= Enter your choice: 6
185 list4=
186 list5= Enter the time you want to convert along with the unit (5 sec):2 day
187 list6= 172800000.0 ms
188 list7= 172800.0 sec
189 list8= 2880.0 min
190 a=float 48.0 hour
191 if(1[1] 0.285714 week
192 for 0.005754 month
193 0.00548 year
194 In [9]:
195 elif(1)
196 for
197 elif(1)
198 for
199 elif(1)
200 for
201 elif(1)
202 for
203 elif(1)
204 for
205 elif(1)
206 for
207 elif(1)
208 for
209 elif(1)
210 for
211
Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 250 Column: 18 Memory: 85 %
```

Fig 6.Time Converter

The screenshot shows the Spyder Python IDE interface. The left pane displays a file explorer with 'pyproject.py' selected. The main editor shows the code for a speed converter. The right pane, titled 'IPython console', shows the execution output. The program prompts the user to enter a choice (1-8) and then the speed value. The user has entered '7' for the speed converter and '5' for the speed value. The program then displays the conversion results for 5 m/s to km/h.

```
File Edit Search
pyproject.py
In [9]: runfile('C:/Users/tushar/Desktop/pyproject.py', wdir='C:/Users/tushar/Desktop')
Enter 1 for the length converter
Enter 2 for the weight converter
Enter 3 for the volume converter
Enter 4 for the area converter
Enter 5 for the currency converter
Enter 6 for the time converter
Enter 7 for the speed converter
Enter 8 for the force converter

Enter your choice: 7

Enter the speed you want to convert along with the unit (5 m/s): 5 km/h
list6= 2.2216 m/s
list7= 0.002224 km/s
list8= 133.3336 m/min
a=float(0.13336 km/min)
if(l[1]):
    for
In [10]: |

elif(l[1])
for

elif(l[1])
for

elif(l[1])
for

elif(l[1])
for

elif(l[1])
for

Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 250 Column: 18 Memory: 86 %
```

Fig 7.Speed Converter

The screenshot shows the Spyder Python IDE interface. The left pane displays the editor with a file named 'pyproject.py'. The right pane shows the IPython console output. The program prompts the user to enter a choice for a converter (1-8). The user enters '8' for the force converter. The program then prompts for the force value in dynes, and the user enters '900000.0'. The program calculates and displays the equivalent values in kilonewtons (kN), kilograms-force (kgf), and pounds-force (lbf).

```
172 In [11]: runfile('C:/Users/tushar/Desktop/pyproject.py', wdir='C:/Users/tushar/Desktop')
173 Enter 1 for the length converter
174 Enter 2 for the weight converter
175 Enter 3 for the volume converter
176 Enter 4 for the area converter
177 Enter 5 for the currency converter
178 Enter 6 for the time converter
179 Enter 7 for the speed converter
180 Enter 8 for the force converter
181 Enter your choice: 8
182 list3=
183 list4=
184 Enter the force you want to convert along with the unit (5 dyn):9 N
185 list6= 900000.0 dyn
186 list7= 0.009000000000000001 kN
187 list8= 0.9177479599999999 kgf
188 newf=2.0232 lbf
189 if(1[1]
190 for
191
192 elif(1
193 for
194
195 elif(1
196 for
197
198 elif(1
199 for
200
201 elif(1
202 for
203
204 elif(1
205 for
206
207 elif(1
208 for
209
210 elif(1
211 for
```

IPython console output:

```
In [11]: runfile('C:/Users/tushar/Desktop/pyproject.py', wdir='C:/Users/tushar/Desktop')
Enter 1 for the length converter
Enter 2 for the weight converter
Enter 3 for the volume converter
Enter 4 for the area converter
Enter 5 for the currency converter
Enter 6 for the time converter
Enter 7 for the speed converter
Enter 8 for the force converter
Enter your choice: 8
Enter the force you want to convert along with the unit (5 dyn):9 N
900000.0 dyn
0.009000000000000001 kN
0.9177479599999999 kgf
2.0232 lbf
In [12]: |
```

Permissions: RW End-of-line: CRLF Encoding: UTF-8 Line: 250 Column: 18 Memory: 86 %

Fig 8. Force Converter

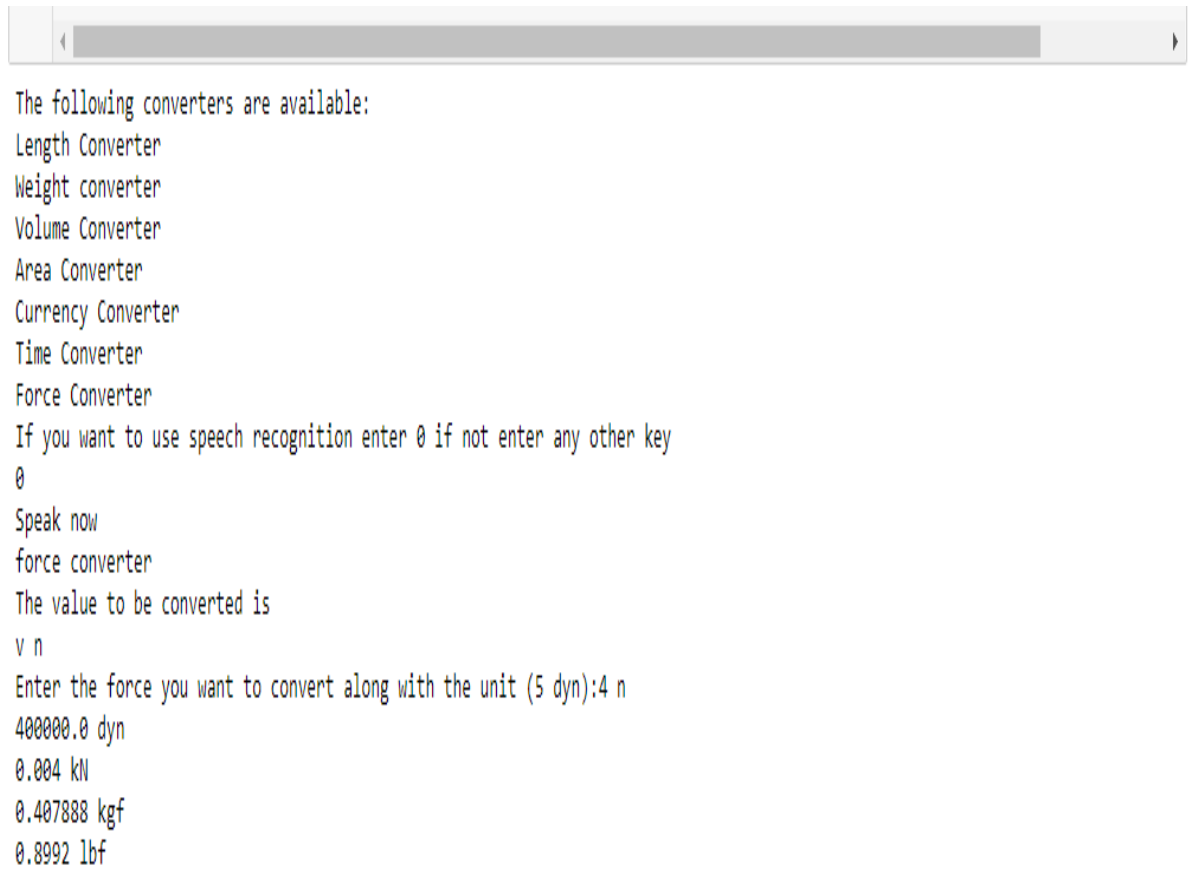
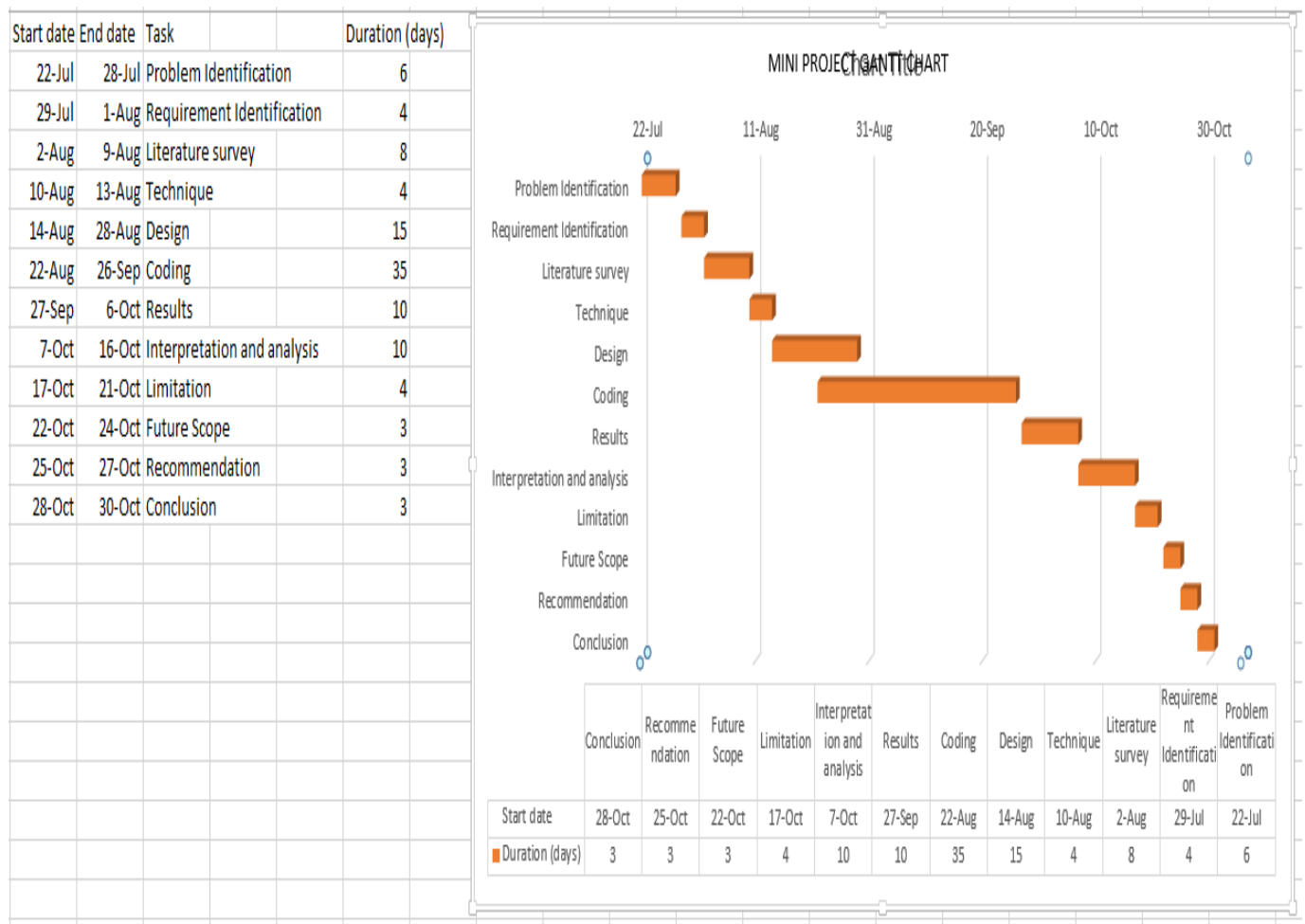


Fig 9. Force Converter with Speech Recognition

STAKEHOLDERS:

- Students
- Scientists
- Mathematicians
- Shopkeeper
- Data Analysts
- Physicists
- Foreign currency Exchange
- Investors

GANTT CHART



REFERENCES:

- <https://www.w3schools.com/>
- <https://www.tutorialspoint.com/>
- <https://stackoverflow.com/>
- <https://www.quora.com/>
- <https://www.geeksforgeeks.org/>