

PRESIDENCY UNIVERSITY

PYTHON PROJECT

ON

BLOOD BANK MANAGEMENT SYSTEM



PROJECT

BY

CHEKRAPANI-2016CSE085

JOYCE MARY B-2016CSE095

KRISHNA VYAS-2016CSE099

YASHASWINI-2016CSE107

CONTENTS

ABSTRACT

INTRODUCTION

DESCRIPTION

DATA FLOW DIAGRAM

SYSTEM DIAGRAM

APPLICATION

DRAWBACKS

FUTURE ENHANCEMENT

CONCLUSION

ABSTRACT

As we being living in the contemporary world should keenly give approach to the Health issues typically. The number of persons who are in need of blood are increasing in large number day by day. In order to help people who are in need of blood, Online Blood Bank Management can be used effectively for getting the details of blood donors having the same blood group.

INTRODUCTION

The process of building systems has always been complex with system becoming larger, the costs and complexities get multiplied. So the need for better methods for developing systems is widely recognized to be effective and the applied model should meet a few basic requirements. The model should be structured and cover the entire system development process from feasibility study to programming, testing and implementation.

DESCRIPTION

The Donor and the Recipient plays the major role.

There are types of blood that we have to notice when the recipient indulges in it. The emphasis is given more on the Blood Group.

TYPES OF BLOOD GROUP

- Abpositive
- Abnegative
- Opositive
- Onegative
- Apositive
- Anegative
- Bpositive
- Bnegative

WORKING ENVIRONMENT

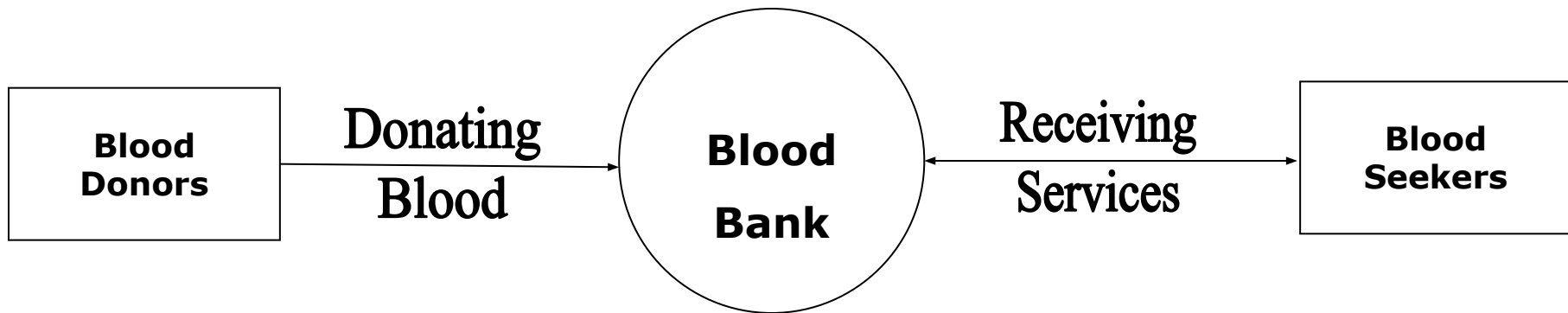
Software Configuration:

Operating System	: Windows 10
Web server	: Personal Web Server
Web Browser	: Internet Explorer
Designing Tool	: Python
Server Side Scripting	: Python 3.6
Backend	: Spyder

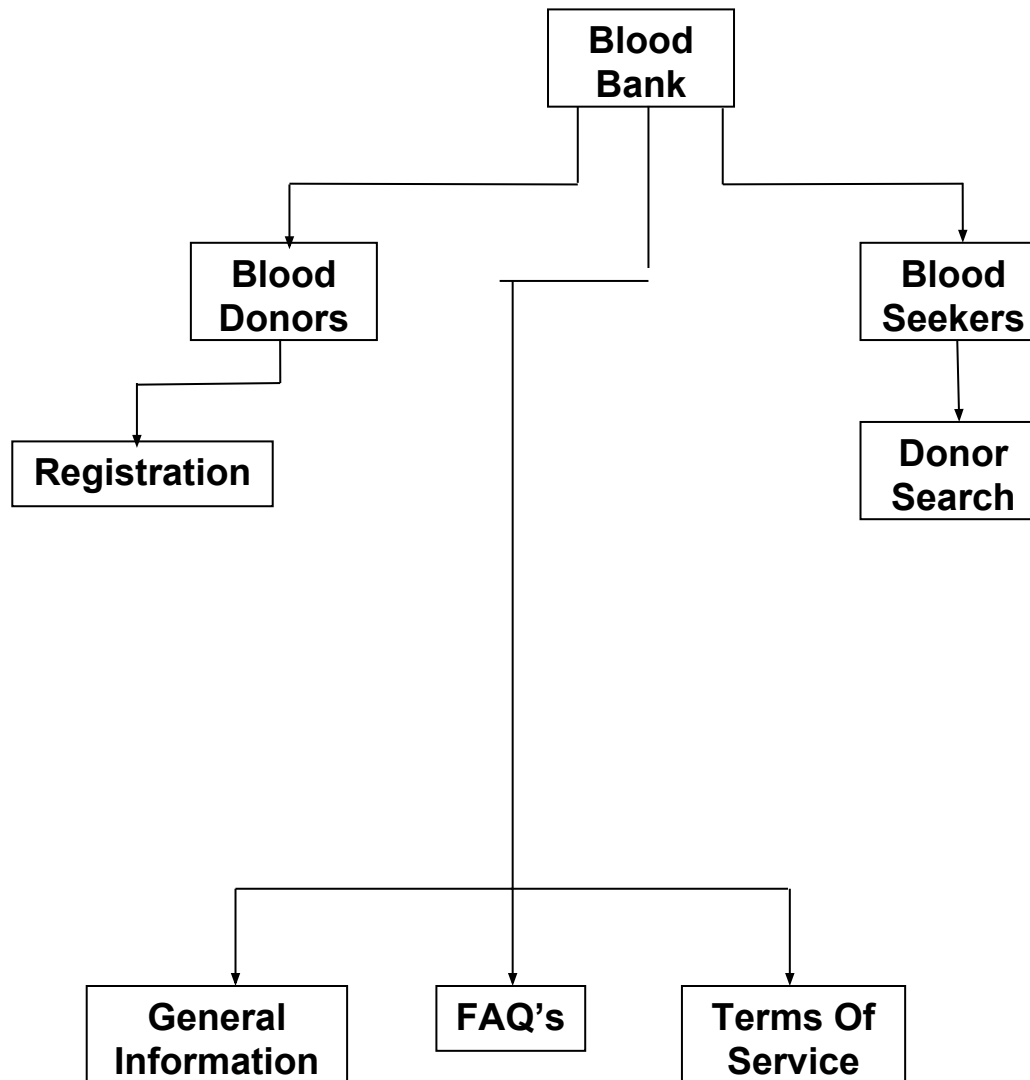
PACKAGES USED

1. xlrd
2. openpyxl
3. datetime



















DATA FLOW DIAGRAM



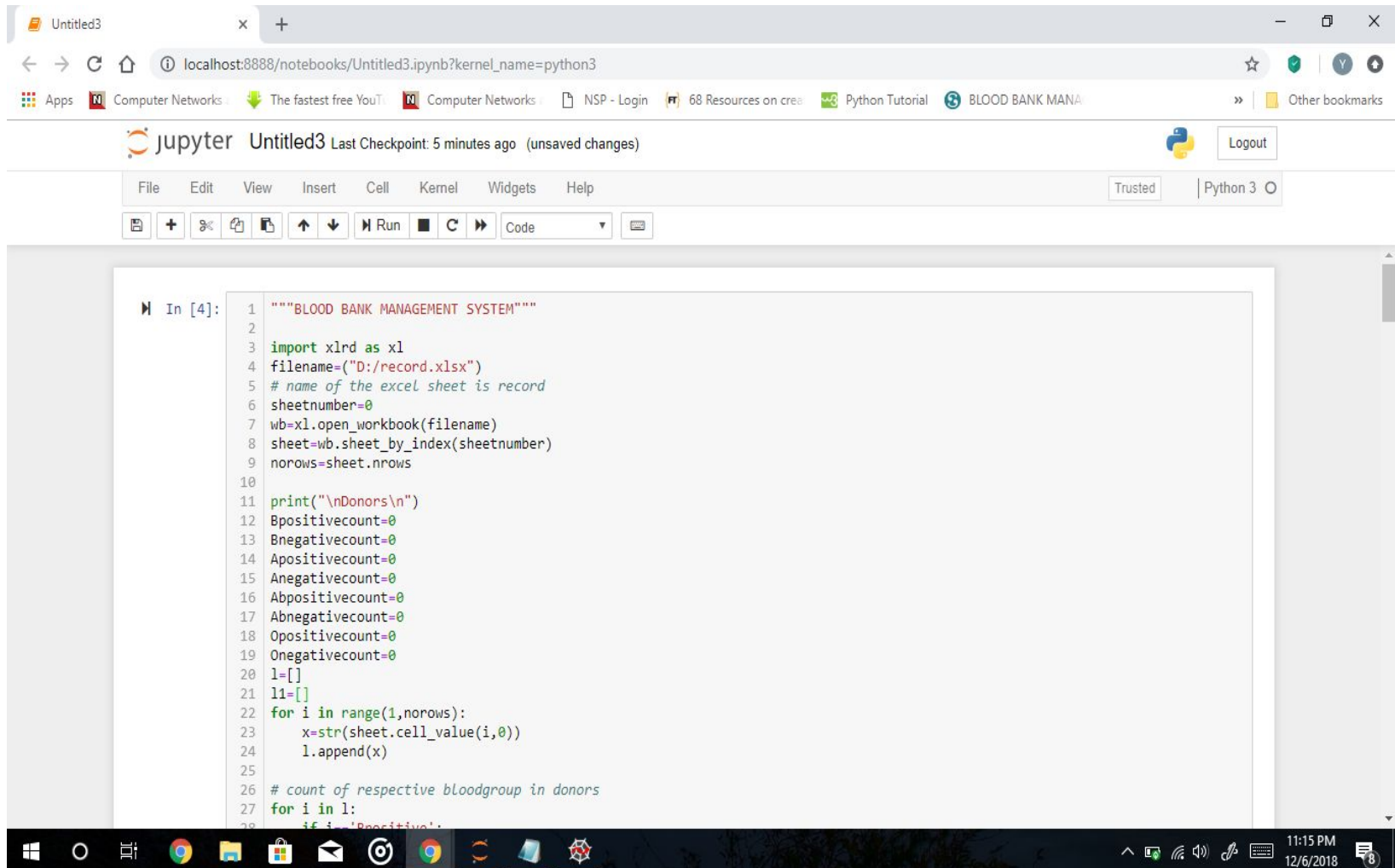
SYSTEM FLOW MODEL



DATABASE

Search the menu (Alt+/)		   		100%	\$ % .0 .00 123	Calibri		11		B <i>I</i> <u>A</u>	  	   	   	 	...		
fx																	
	A	B	C	D	E	F	G	H	I	J							
1	BLOOD GROUP	DONORS NAME	DONORS AGE	DATE		R_BLOOD GROUP	RECIPIENT NAME	RECIPIENT	DATE								
2	Bpositive	Jeny	31	2016		Oposite	Suman	21	2016								
3	Bnegative	Akshay	42	2016		Bpositive	Kumari	32	2016								
4	Bpositive	Satish	50	2016		Anegative	Rajashwari	24	2016								
5	Oposite	Mahesh	26	2016		Bnegative	Likitha	34	2016								
6	Onegative	Ramesh	23	2016		Abpositive	Karunya	26	2016								
7	Abpositive	Kamlesh	29	2016		Oposite	Nawaz	74	2016								
8	Oposite	Nagesh	32	2016		Apositive	Jishnu	45	2016								
9	Onegative	Ramyra	36	2016		Bpositive	Mownesh	65	2016								
10	Bpositive	Rohit	30	2016		Oposite	Divya	45	2016								
11	Abnegative	Rakesh	25	2016		Abnegative	Varshini	35	2016								
12	Apositive	Hari	27	2016		Onegative	Keerthana	35	2016								
13	Anegative	Sankalp	24	2016		Onegative	Shrjana	65	2016								
14	Oposite	Shaila	25	2016		Anegative	Daksh	24	2017								
15	Bnegative	Manjula	33	2017		Bpositive	Bhavik	68	2017								
16	Apositive	Raju	32	2017		Apositive	Vignesh	45	2017								
17	Bpositive	kousalya	24	2017		Oposite	Sheetal	65	2017								
18	Onegative	Gangalakshmi	23	2017		Abnegative	Reena	66	2017								
19	Anegative	Sangeetha	19	2017		Bpositive	Shawn	73	2017								
20	Abnegative	Maharaj	32	2017		Onegative	Rinz	25	2017								
21	Bpositive	Luvina	37	2017		Bnegative	Damodaran	34	2017								
22	Oposite	Michael	26	2017		Bnegative	Salman	55	2017								
23	Bnegative	Sony	33	2017		Abpositive	Joanah	64	2017								
24	Abpositive	Shiny	31	2017		Apositive	Jotham	69	2018								
25	Bnegative	Pushpa	23	2017		Oposite	Julia	70	2018								
26	Apositive	Nikhil	22	2018		Bpositive	Akila	77	2018								
27	Abpositive	Sushanth	19	2018		Bnegative	Neha	34	2018								
28	Bnegative	Antony	43	2018		Abpositive	Alsenan	55	2018								
29	Oposite	Divesh	23	2018		Anegative	Sarah	43	2018								
30	Bpositive	Mohammad	34	2018		Onegative	Joel	18	2018								

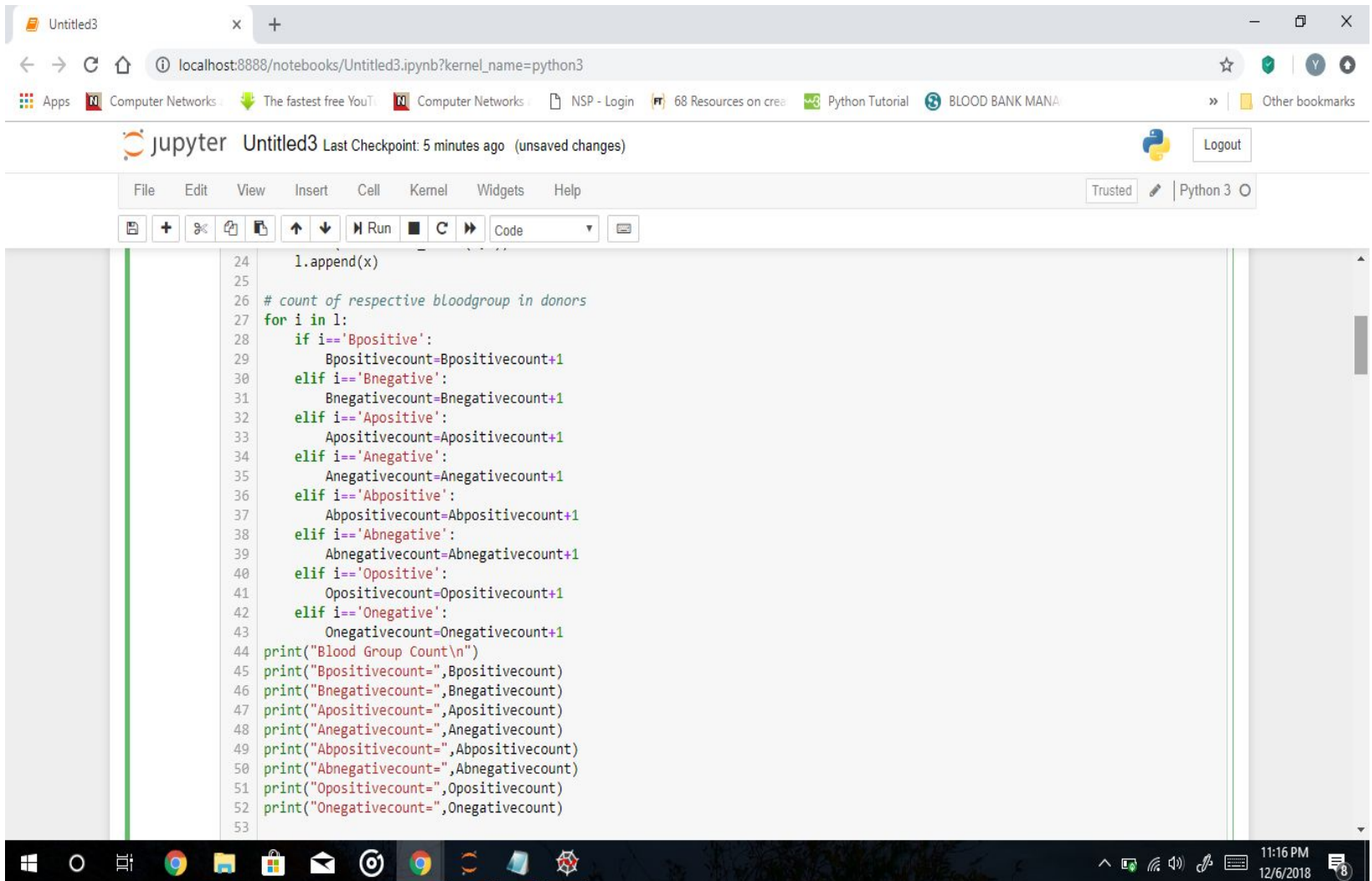
SCREENS OF THE WORK(CODE)



The screenshot displays a Jupyter Notebook titled 'Untitled3' running on a local server at localhost:8888. The interface includes a standard web browser address bar, a Jupyter logo, and a menu bar with options like File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. The code is written in Python and is part of a 'BLOOD BANK MANAGEMENT SYSTEM'. It imports the 'xlrd' module to read an Excel file named 'record.xlsx'. The script initializes several counters for blood types (positive and negative for B, A, and O) and a list 'l' to store donor information. It then iterates through the rows of the Excel sheet, extracting the donor's name and blood group, and appends the name to the list 'l'. The code is partially visible, showing lines 1 through 29.

```
In [4]: 1 """BLOOD BANK MANAGEMENT SYSTEM"""
2
3 import xlrd as xl
4 filename=("D:/record.xlsx")
5 # name of the excel sheet is record
6 sheetnumber=0
7 wb=xl.open_workbook(filename)
8 sheet=wb.sheet_by_index(sheetnumber)
9 norows=sheet.nrows
10
11 print("\nDonors\n")
12 Bpositivecount=0
13 Bnegativecount=0
14 Apositivecount=0
15 Anegativecount=0
16 Abpositivecount=0
17 Abnegativecount=0
18 Opositivecount=0
19 Onegativecount=0
20 l=[]
21 l1=[]
22 for i in range(1,norows):
23     x=str(sheet.cell_value(i,0))
24     l.append(x)
25
26 # count of respective bloodgroup in donors
27 for i in l:
28     if i=='Positive':
```

SCREENS OF THE WORK(CODE)

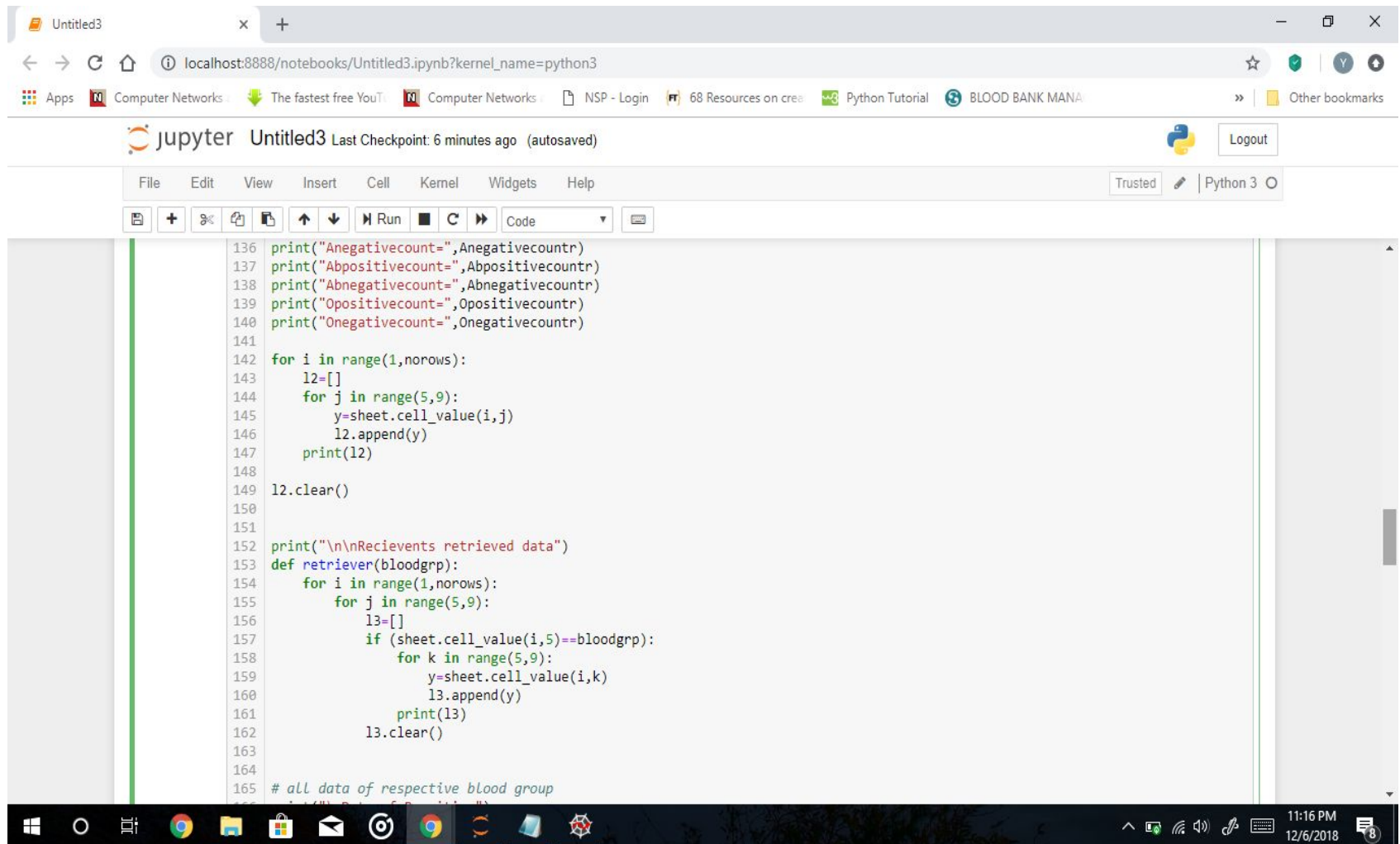


The screenshot displays a Jupyter Notebook titled 'Untitled3' in a web browser. The address bar shows the URL 'localhost:8888/notebooks/Untitled3.ipynb?kernel_name=python3'. The notebook interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations and execution. The code is written in a Python cell and is as follows:

```
24     l.append(x)
25
26     # count of respective bloodgroup in donors
27     for i in l:
28         if i=='Bpositive':
29             Bpositivecount=Bpositivecount+1
30         elif i=='Bnegative':
31             Bnegativecount=Bnegativecount+1
32         elif i=='Apositive':
33             Apositivecount=Apositivecount+1
34         elif i=='Anegative':
35             Anegativecount=Anegativecount+1
36         elif i=='Abpositive':
37             Abpositivecount=Abpositivecount+1
38         elif i=='Abnegative':
39             Abnegativecount=Abnegativecount+1
40         elif i=='Opositive':
41             Opositivecount=Opositivecount+1
42         elif i=='Onegative':
43             Onegativecount=Onegativecount+1
44     print("Blood Group Count\n")
45     print("Bpositivecount=",Bpositivecount)
46     print("Bnegativecount=",Bnegativecount)
47     print("Apositivecount=",Apositivecount)
48     print("Anegativecount=",Anegativecount)
49     print("Abpositivecount=",Abpositivecount)
50     print("Abnegativecount=",Abnegativecount)
51     print("Opositivecount=",Opositivecount)
52     print("Onegativecount=",Onegativecount)
53
```

The Windows taskbar at the bottom shows the time as 11:16 PM on 12/6/2018.

SCREENS OF THE WORK(CODE)

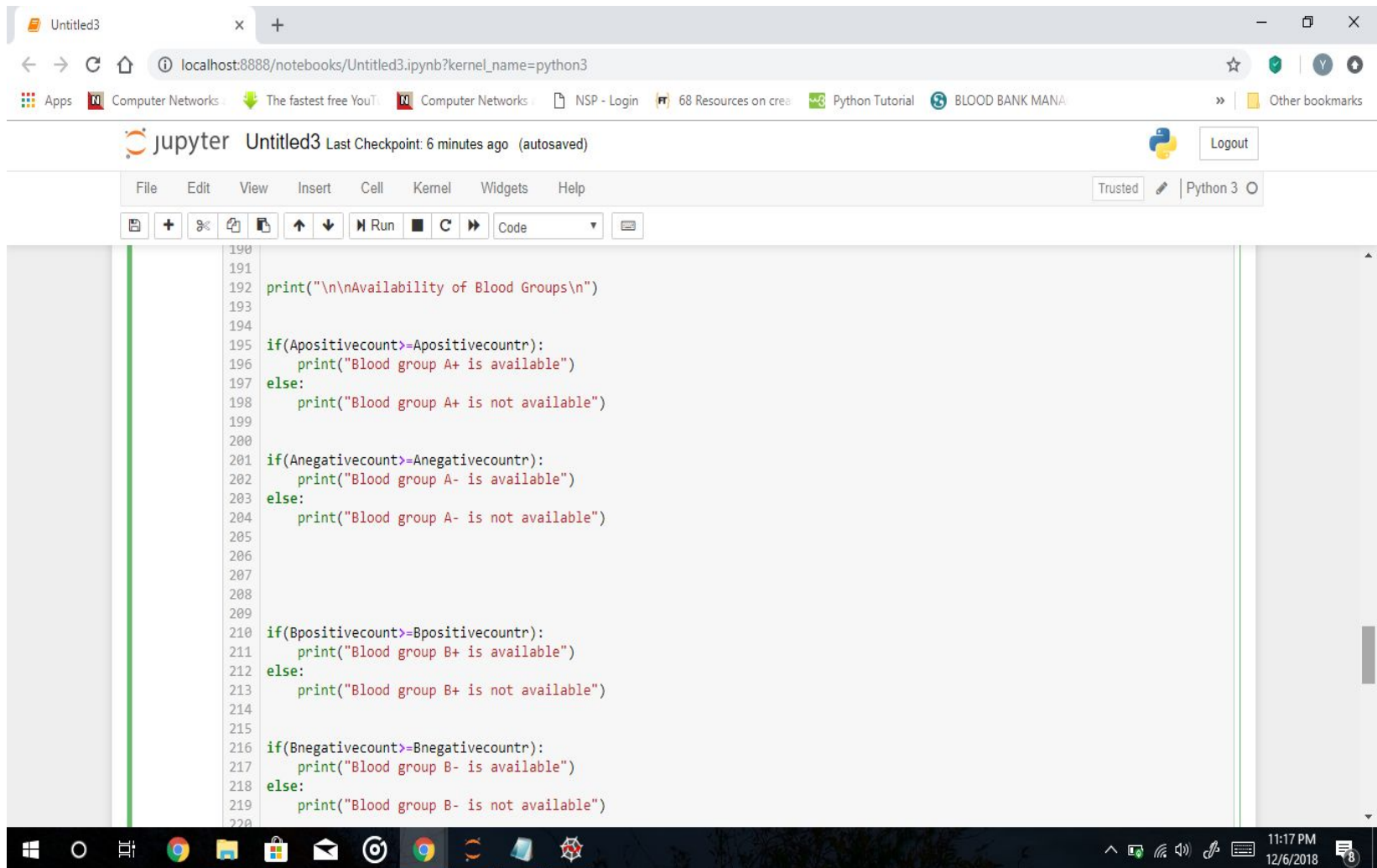


The screenshot displays a Jupyter Notebook titled 'Untitled3' running on a local server at `localhost:8888/notebooks/Untitled3.ipynb?kernel_name=python3`. The interface includes a standard web browser address bar, a Jupyter logo, and a 'Logout' button. Below the header is a menu bar with options: File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. A toolbar contains icons for file operations, running the cell, and a dropdown menu currently set to 'Code'. The main area shows a Python script with the following code:

```
136 print("Anegativecount=",Anegativecount)
137 print("Abpositivecount=",Abpositivecount)
138 print("Abnegativecount=",Abnegativecount)
139 print("Opositivecount=",Opositivecount)
140 print("Onegativecount=",Onegativecount)
141
142 for i in range(1,norows):
143     l2=[]
144     for j in range(5,9):
145         y=sheet.cell_value(i,j)
146         l2.append(y)
147     print(l2)
148
149 l2.clear()
150
151
152 print("\n\nRecievents retrieved data")
153 def retriever(bloodgrp):
154     for i in range(1,norows):
155         for j in range(5,9):
156             l3=[]
157             if (sheet.cell_value(i,5)==bloodgrp):
158                 for k in range(5,9):
159                     y=sheet.cell_value(i,k)
160                     l3.append(y)
161                 print(l3)
162             l3.clear()
163
164
165 # all data of respective blood group
```

The Windows taskbar at the bottom shows the system time as 11:16 PM on 12/6/2018, along with various background application icons.

SCREENS OF THE WORK(CODE)

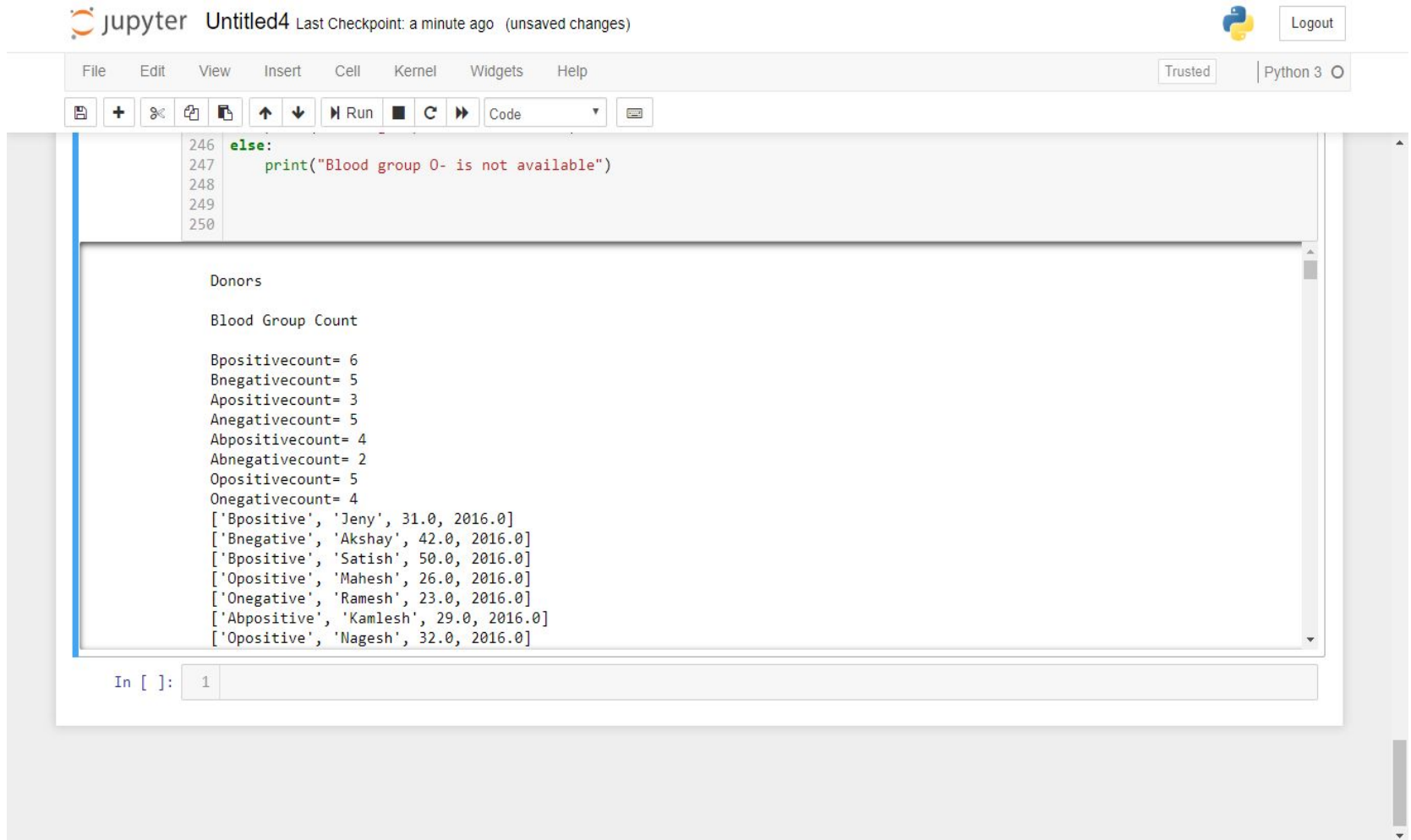


The screenshot displays a Jupyter Notebook titled 'Untitled3' running on a local server at localhost:8888. The interface includes a standard web browser address bar, a Jupyter logo, and a menu bar with options like File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. The code is written in Python and checks the availability of blood groups A+, A-, B+, and B- based on positive and negative counts. The code is as follows:

```
190
191
192 print("\n\nAvailability of Blood Groups\n")
193
194
195 if(Apositivecount>=Apositivecount):
196     print("Blood group A+ is available")
197 else:
198     print("Blood group A+ is not available")
199
200
201 if(Anegativecount>=Anegativecount):
202     print("Blood group A- is available")
203 else:
204     print("Blood group A- is not available")
205
206
207
208
209
210 if(Bpositivecount>=Bpositivecount):
211     print("Blood group B+ is available")
212 else:
213     print("Blood group B+ is not available")
214
215
216 if(Bnegativecount>=Bnegativecount):
217     print("Blood group B- is available")
218 else:
219     print("Blood group B- is not available")
220
```

The Windows taskbar at the bottom shows the system time as 11:17 PM on 12/6/2018.

SCREENS OF THE WORK(OUTPUT)



A screenshot of a Jupyter Notebook interface. The top bar shows the Jupyter logo, the file name 'Untitled4', and the status 'Last Checkpoint: a minute ago (unsaved changes)'. On the right, there is a Python logo and a 'Logout' button. Below the top bar is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. To the right of the menu bar are 'Trusted' and 'Python 3' indicators. Below the menu bar is a toolbar with icons for saving, adding cells, undo, redo, running, and other functions. The main area shows a code cell with the following code:

```
246 else:
247     print("Blood group O- is not available")
248
249
250
```

Below the code cell is an output area displaying the following text:

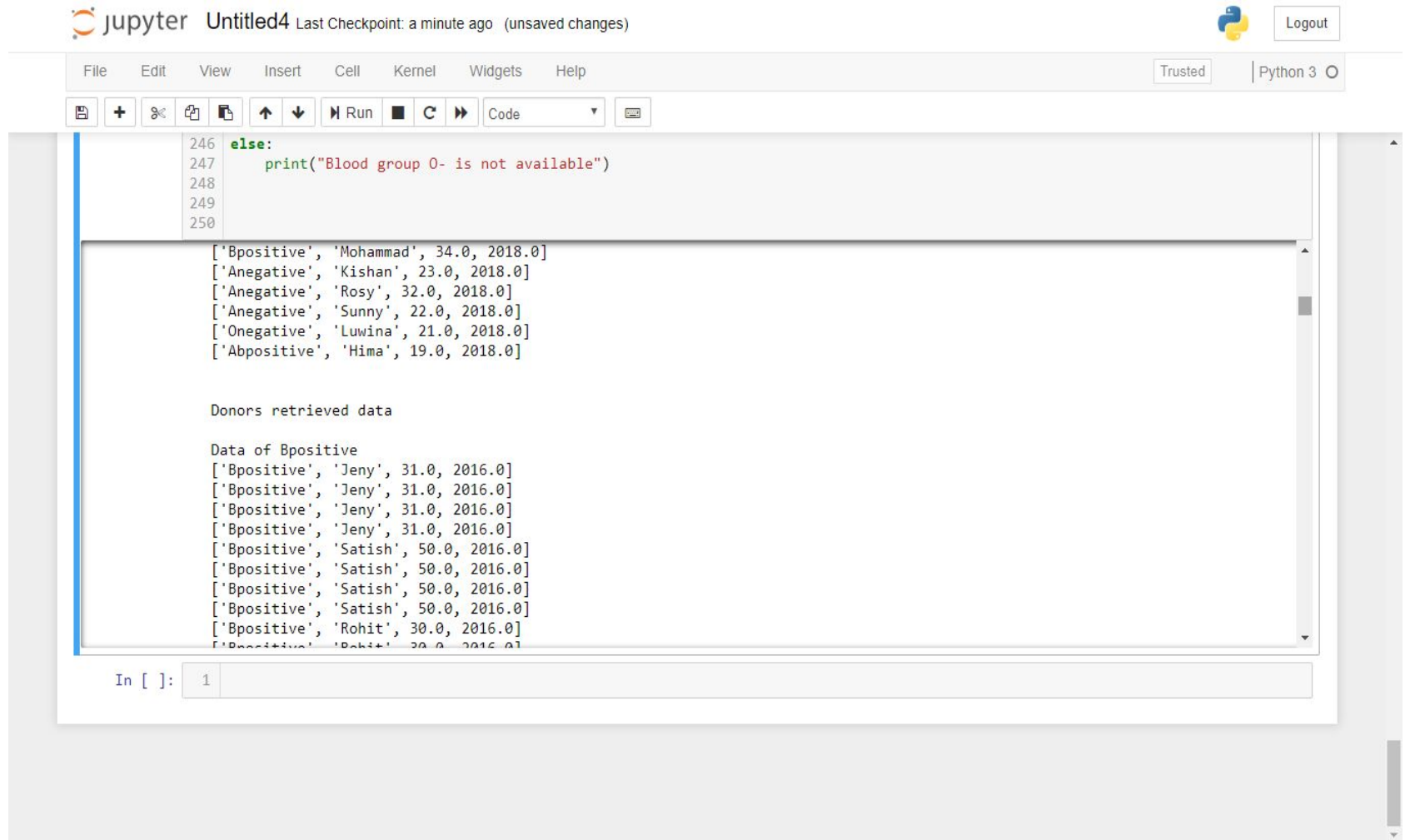
```
Donors

Blood Group Count

Bpositivecount= 6
Bnegativecount= 5
Apositivecount= 3
Anegativecount= 5
Abpositivecount= 4
Abnegativecount= 2
Opositivecount= 5
Onegativecount= 4
['Bpositive', 'Jeny', 31.0, 2016.0]
['Bnegative', 'Akshay', 42.0, 2016.0]
['Bpositive', 'Satish', 50.0, 2016.0]
['Opositive', 'Mahesh', 26.0, 2016.0]
['Onegative', 'Ramesh', 23.0, 2016.0]
['Abpositive', 'Kamlesh', 29.0, 2016.0]
['Opositive', 'Nagesh', 32.0, 2016.0]
```

At the bottom, there is an input prompt 'In []:' followed by the number '1'.

SCREENS OF THE WORK(OUTPUT)



A screenshot of a Jupyter Notebook interface. The top bar shows the Jupyter logo, the file name 'Untitled4', and a status message 'Last Checkpoint: a minute ago (unsaved changes)'. On the right, there is a Python logo and a 'Logout' button. Below the top bar is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. To the right of the menu bar are 'Trusted' and 'Python 3' indicators. Below the menu bar is a toolbar with icons for saving, adding cells, running, and other functions. The main area of the notebook shows a code cell with the following code:

```
246 else:
247     print("Blood group O- is not available")
248
249
250
```

The output of the code cell is displayed below the code:

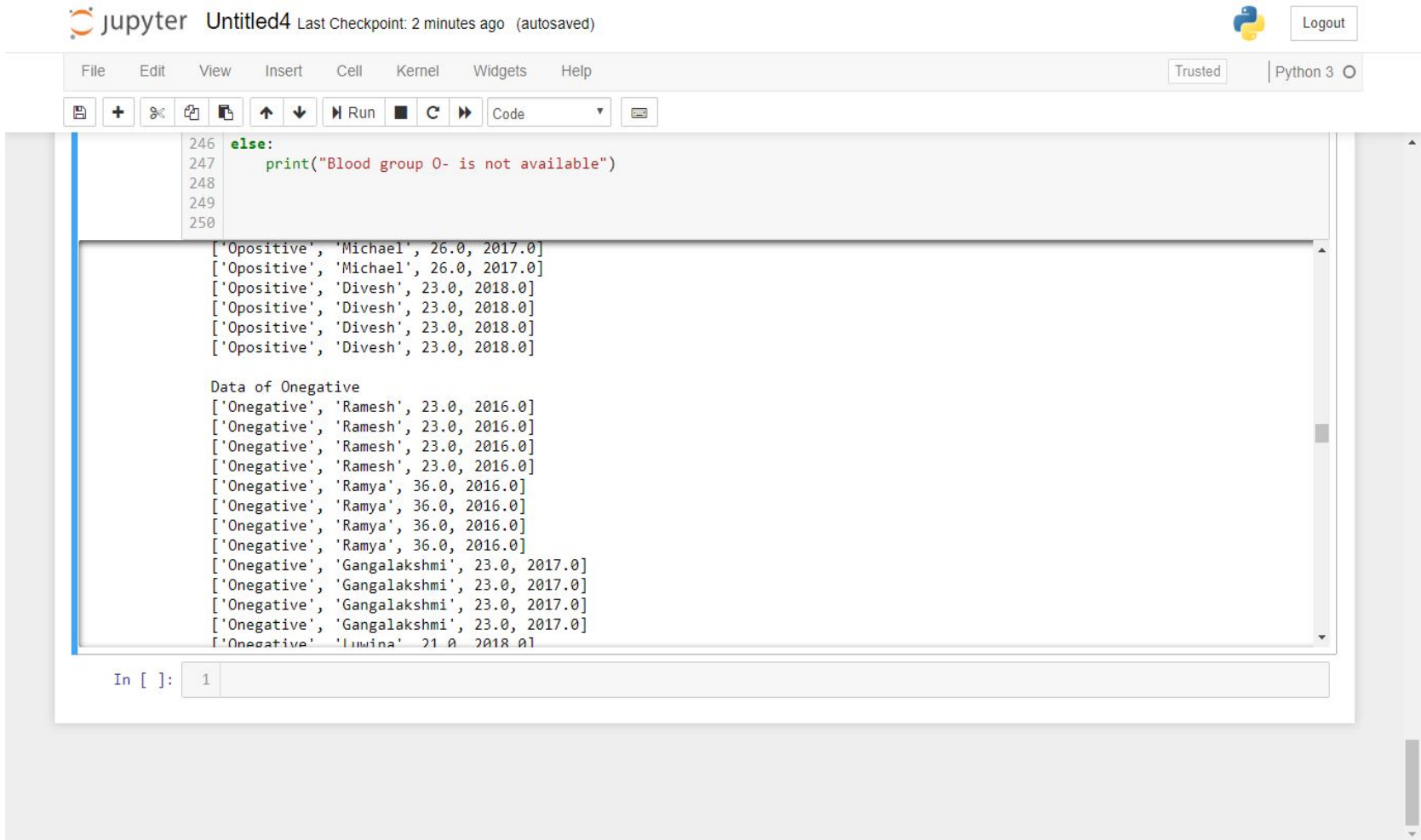
```
['Bpositive', 'Mohammad', 34.0, 2018.0]
['Anegative', 'Kishan', 23.0, 2018.0]
['Anegative', 'Rosy', 32.0, 2018.0]
['Anegative', 'Sunny', 22.0, 2018.0]
['Onegative', 'Luwina', 21.0, 2018.0]
['Abpositive', 'Hima', 19.0, 2018.0]

Donors retrieved data

Data of Bpositive
['Bpositive', 'Jeny', 31.0, 2016.0]
['Bpositive', 'Jeny', 31.0, 2016.0]
['Bpositive', 'Jeny', 31.0, 2016.0]
['Bpositive', 'Jeny', 31.0, 2016.0]
['Bpositive', 'Satish', 50.0, 2016.0]
['Bpositive', 'Satish', 50.0, 2016.0]
['Bpositive', 'Satish', 50.0, 2016.0]
['Bpositive', 'Satish', 50.0, 2016.0]
['Bpositive', 'Rohit', 30.0, 2016.0]
['Bpositive', 'Rohit', 30.0, 2016.0]
```

At the bottom of the notebook, there is an input prompt 'In []:' followed by a text box containing the number '1'.

SCREENS OF THE WORK(OUTPUT)



A screenshot of a Jupyter Notebook interface. The top bar shows the Jupyter logo, the file name 'Untitled4', and a status message 'Last Checkpoint: 2 minutes ago (autosaved)'. On the right, there is a Python logo and a 'Logout' button. Below the top bar is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. A 'Trusted' status indicator and 'Python 3' are also visible. The toolbar contains icons for saving, adding cells, deleting, copying, pasting, undo, redo, running, and a code editor dropdown. The code editor shows a Python snippet with an 'else' block that prints a message. The output area displays the execution results, including a list of data points for 'Opositive' and 'Onegative' categories, each with a name, age, and year. The input prompt 'In []:' is followed by the number '1'.

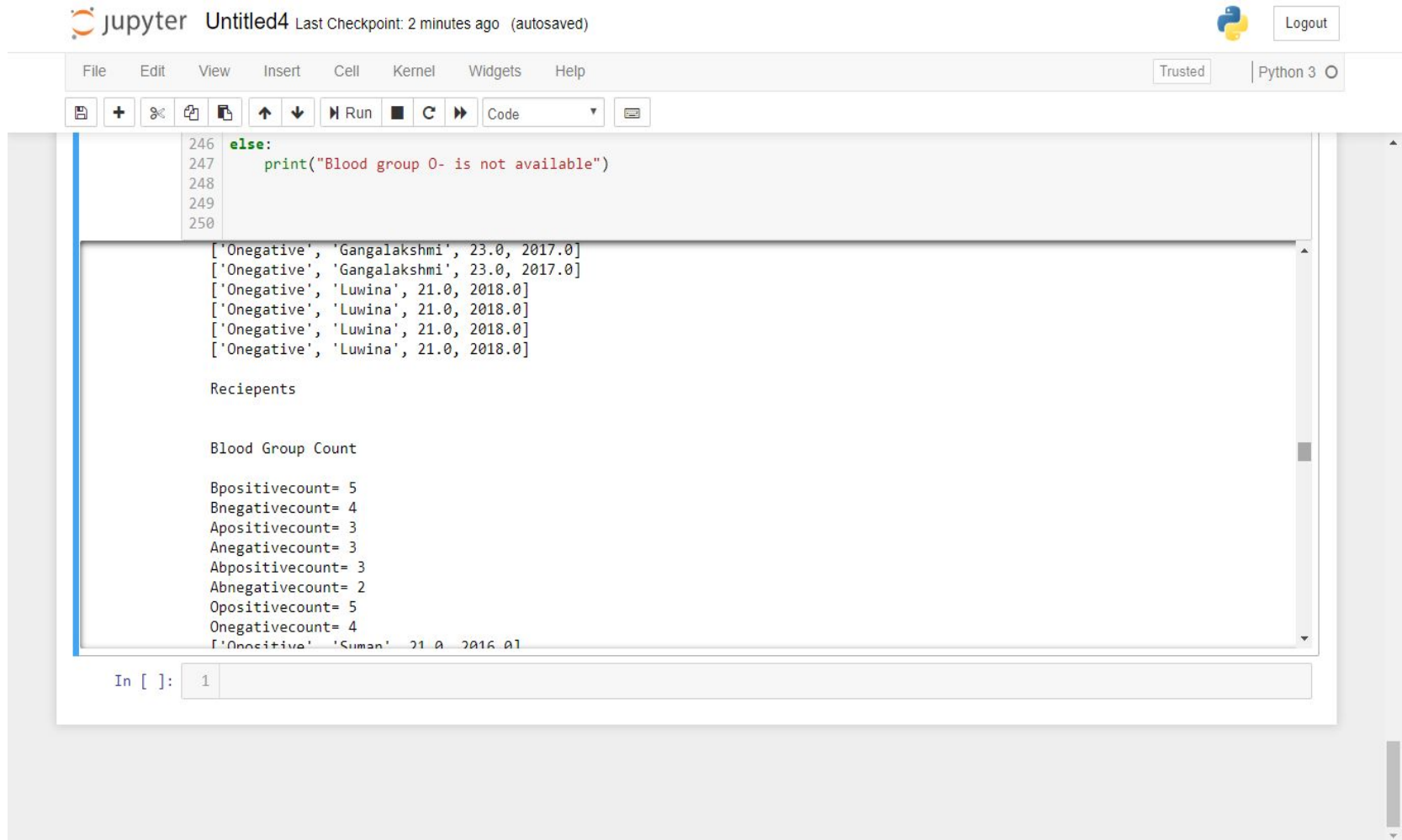
```
246 else:
247     print("Blood group O- is not available")
248
249
250
```

```
['Opositive', 'Michael', 26.0, 2017.0]
['Opositive', 'Michael', 26.0, 2017.0]
['Opositive', 'Divesh', 23.0, 2018.0]
['Opositive', 'Divesh', 23.0, 2018.0]
['Opositive', 'Divesh', 23.0, 2018.0]
['Opositive', 'Divesh', 23.0, 2018.0]

Data of Onegative
['Onegative', 'Ramesh', 23.0, 2016.0]
['Onegative', 'Ramesh', 23.0, 2016.0]
['Onegative', 'Ramesh', 23.0, 2016.0]
['Onegative', 'Ramesh', 23.0, 2016.0]
['Onegative', 'Ramya', 36.0, 2016.0]
['Onegative', 'Ramya', 36.0, 2016.0]
['Onegative', 'Ramya', 36.0, 2016.0]
['Onegative', 'Ramya', 36.0, 2016.0]
['Onegative', 'Gangalakshmi', 23.0, 2017.0]
['Onegative', 'Gangalakshmi', 23.0, 2017.0]
['Onegative', 'Gangalakshmi', 23.0, 2017.0]
['Onegative', 'Gangalakshmi', 23.0, 2017.0]
['Onegative', 'Luvina', 21.0, 2018.0]
```

In []: 1

SCREENS OF THE WORK(OUTPUT)



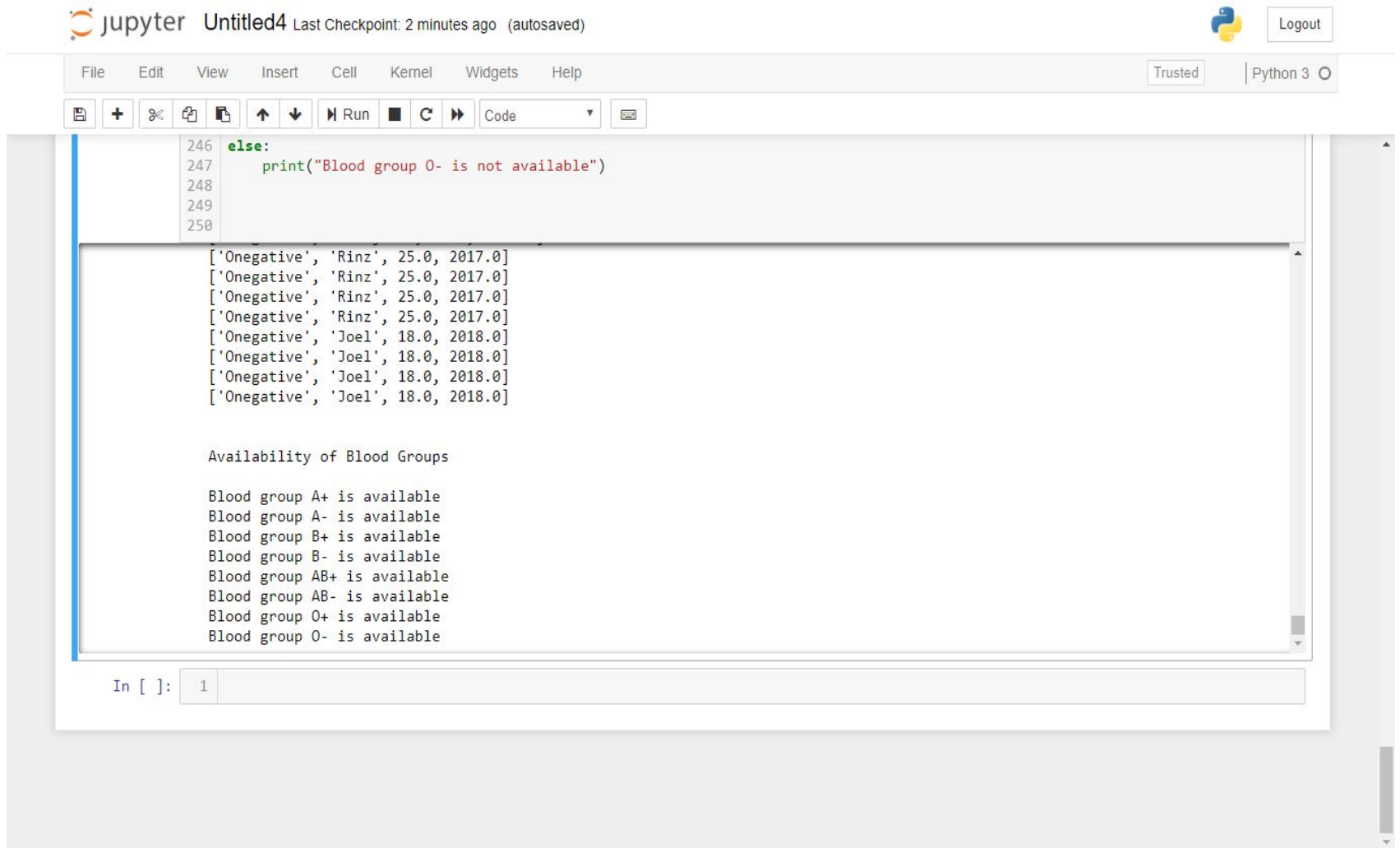
A screenshot of a Jupyter Notebook interface. The top bar shows the Jupyter logo, the file name 'Untitled4', and the status 'Last Checkpoint: 2 minutes ago (autosaved)'. On the right, there is a Python logo and a 'Logout' button. Below the top bar is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. To the right of the menu bar are 'Trusted' and 'Python 3' indicators. Below the menu bar is a toolbar with icons for saving, adding cells, running, and other functions. The main area of the notebook shows a code cell with the following code:

```
246 else:
247     print("Blood group O- is not available")
248
249
250
```

The output of the code cell is displayed below the code. It shows a list of six tuples, each representing a blood donor with their blood group, name, age, and year of birth. The tuples are: ['Onegative', 'Gangalakshmi', 23.0, 2017.0], ['Onegative', 'Gangalakshmi', 23.0, 2017.0], ['Onegative', 'Luwina', 21.0, 2018.0], ['Onegative', 'Luwina', 21.0, 2018.0], ['Onegative', 'Luwina', 21.0, 2018.0], and ['Onegative', 'Luwina', 21.0, 2018.0]. Below the list, the text 'Reciepts' is displayed. Then, the text 'Blood Group Count' is displayed. Finally, the counts for each blood group are displayed: Bpositivecount= 5, Bnegativecount= 4, Apositivecount= 3, Anegativecount= 3, Abpositivecount= 3, Abnegativecount= 2, Opositivecount= 5, Onegativecount= 4, and a final line showing ['Onegative', 'Suman', 21.0, 2016.0].

In []: 1

SCREENS OF THE WORK(OUTPUT)



The screenshot displays a Jupyter Notebook interface. At the top, the Jupyter logo is followed by the text "Jupyter" and "Untitled4". To the right of "Untitled4" is the text "Last Checkpoint: 2 minutes ago (autosaved)". Further right is a Python logo and a "Logout" button. Below this is a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". To the right of the menu bar are buttons for "Trusted" and "Python 3". Below the menu bar is a toolbar with icons for saving, adding, deleting, and running code, along with a dropdown menu set to "Code". The main area of the notebook shows a code cell with the following Python code:

```
246 else:
247     print("Blood group O- is not available")
248
249
250
```

The output of the code cell is displayed below the code. It shows a list of blood group availability data for two individuals, Rinz and Joel, across different years. The data is presented as a list of lists, where each inner list contains the blood group, the name, the age, and the year.

```
['Onegative', 'Rinz', 25.0, 2017.0]
['Onegative', 'Rinz', 25.0, 2017.0]
['Onegative', 'Rinz', 25.0, 2017.0]
['Onegative', 'Rinz', 25.0, 2017.0]
['Onegative', 'Joel', 18.0, 2018.0]
['Onegative', 'Joel', 18.0, 2018.0]
['Onegative', 'Joel', 18.0, 2018.0]
['Onegative', 'Joel', 18.0, 2018.0]
```

Below the list, the text "Availability of Blood Groups" is displayed. This is followed by a list of blood groups and their availability status:

```
Blood group A+ is available
Blood group A- is available
Blood group B+ is available
Blood group B- is available
Blood group AB+ is available
Blood group AB- is available
Blood group O+ is available
Blood group O- is available
```

At the bottom of the notebook, the input prompt "In []:" is followed by the number "1".

APPLICATIONS AND DRAWBACKS

With the help of Online Blood Bank Management, people who are having the thought of donating blood gets registered in Online Blood Bank giving his details. Online Blood Bank Management site is available to everyone easily. It preserves the documents and updates the value when ever the donor or recipient enhances.

The optimization of the code is slow and to implement this to the large group of people is in fact very difficult.

FUTURE ENHANCEMENT

To optimize the code and use SQL Database and implement in the large scale and synchronize to the very best of the enhancement.

To be able to reach even to the very end of the city. As there was a little number of contact person's information given, some people may face difficulty in getting blood fast. So i like to gather more information regarding the contact persons in other cities as well as villages and will provide much more services for the people and help everyone with humanity.

CONCLUSION

This project has given us an ample opportunity to design, code, test and implements an application. This has helped in putting into practice of various Software Engineering principles and Database Management concepts like maintaining integrity and consistency of data.

THANK YOU