In [1]:
```
pip install seaborn --break-system-packages
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: seaborn in ./.local/lib/python3.12/site-package
s (0.13.2)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in /usr/lib/python3/dist-p
ackages (from seaborn) (1.26.4)
Requirement already satisfied: pandas>=1.2 in ./.local/lib/python3.12/site-pac
kages (from seaborn) (2.3.3)
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in /usr/lib/python3/dis
t-packages (from seaborn) (3.6.3)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/lib/python3/dist
-packages (from pandas>=1.2->seaborn) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/lib/python3/dist-packages
(from pandas>=1.2->seaborn) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in ./.local/lib/python3.12/site-
packages (from pandas>=1.2->seaborn) (2025.3)
Note: you may need to restart the kernel to use updated packages.
```

DATA WRANGLING 2

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [3]:
```
df=pd.read_csv("/home/hardwarelab00/Downloads/tecdiv.csv")
```

In [4]:
```
df
```

Out[4]:

	Timestamp	Email Address	Name	Ema
0	1/17/2022 12:45:09	sejal.zambare19@pccoepune.org	Sejal Zambare	sejal.zambare19@gmail.cc
1	1/17/2022 12:45:44	rushikesh.thorat19@pccoepune.org	Rushikesh Vilas Thorat	rushikesh.thorat19@pccoepune.c
2	1/17/2022 12:46:10	atharv.sontakke19@pccoepune.org	Atharv Sontakke	atharv123sontakke@gmail.cc
3	1/17/2022 12:46:21	amisha.sherekar19@pccoepune.org	Amisha Sunil Sherekar	amisha.sherekar19@pccoepune.c
4	1/17/2022 12:46:31	saurabh.sawardekar19@pccoepune.org	Saurabh Raju Sawardekar	saurabh.sawardekar19@pccoepune.c
...	...	...	...	
59	1/20/2022 9:24:40	pratik.meshram20@pccoepune.org	Pratik Amrut Meshram	pratik.meshram20@pccoepune.c
60	1/20/2022 9:36:14	prasad.zore19@pccoepune.org	Prasad Zore	prasad.zore@outlook.cc
61	1/20/2022 9:42:34	sudhir.varu19@pccoepune.org	SUDHIR VARU	sudhirvaru01@gmail.cc
62	1/20/2022 10:22:05	bhagyashree.takale19@pccoepune.org	Bhagyashree Gorakh	bbhagyashree002@gmail.cc

	Timestamp	Email Address	Name	Ema
			Takale	
63	1/20/2022 10:38:06	sarvesh.waghmare19@pccoepune.org	Waghmare Sarvesh Jitendra	sarvesh.waghmare19@pccoepune.c

64 rows × 11 columns

In [5]:
```
print("starting 5 rows are as follows")
df.head()
```

```
starting 5 rows are as follows
```

Out[5]:

	Timestamp	Email Address	Name	Email
0	1/17/2022 12:45:09	sejal.zambare19@pccoepune.org	Sejal Zambare	sejal.zambare19@gmail.com
1	1/17/2022 12:45:44	rushikesh.thorat19@pccoepune.org	Rushikesh Vilas Thorat	rushikesh.thorat19@pccoepune.org
2	1/17/2022 12:46:10	atharv.sontakke19@pccoepune.org	Atharv Sontakke	atharv123sontakke@gmail.com
3	1/17/2022 12:46:21	amisha.sherekar19@pccoepune.org	Amisha Sunil Sherekar	amisha.sherekar19@pccoepune.org
4	1/17/2022 12:46:31	saurabh.sawardekar19@pccoepune.org	Saurabh Raju Sawardekar	saurabh.sawardekar19@pccoepune.org

In [6]:
```
print("last 5 rows are ")
df.tail()
```

```
last 5 rows are
```

Out[6]:

	Timestamp	Email Address	Name	Email
59	1/20/2022 9:24:40	pratik.meshram20@pccoepune.org	Pratik Amrut Meshram	pratik.meshram20@pccoepune.org
60	1/20/2022 9:36:14	prasad.zore19@pccoepune.org	Prasad Zore	prasad.zore@outlook.com
61	1/20/2022 9:42:34	sudhir.varu19@pccoepune.org	SUDHIR VARU	sudhirvaru01@gmail.com
62	1/20/2022 10:22:05	bhagyashree.takale19@pccoepune.org	Bhagyashree Gorakh Takale	bbhagyashree002@gmail.com
63	1/20/2022 10:38:06	sarvesh.waghmare19@pccoepune.org	Waghmare Sarvesh Jitendra	sarvesh.waghmare19@pccoepune.org

In [8]:
```
df.describe() # description about the dataset
```

Out[8]:

	Mobile No.	First year: Sem 1	First year: Sem 2	Second year: Sem 1	Second year: Sem 2
count	6.400000e+01	64.000000	64.000000	64.000000	64.000000
mean	8.623097e+09	8.834219	9.095469	9.292031	9.377187
std	9.132070e+08	11.187839	11.171986	0.528523	0.495185
min	7.028870e+09	0.000000	0.000000	6.900000	7.200000
25%	7.766559e+09	7.237500	7.655000	9.050000	9.140000
50%	8.805720e+09	8.260000	8.400000	9.445000	9.450000
75%	9.335094e+09	8.802500	9.115000	9.645000	9.725000
max	9.975810e+09	95.000000	95.000000	9.910000	9.950000

In [10]:
```
df.info() # gives the information about the dataset
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 64 entries, 0 to 63
Data columns (total 11 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Timestamp             64 non-null     object 
 1   Email Address         64 non-null     object 
 2   Name                  64 non-null     object 
 3   Email                 64 non-null     object 
 4   Roll no               64 non-null     object 
 5   PRN No.               64 non-null     object 
 6   Mobile No.            64 non-null     int64  
 7   First year:   Sem 1   64 non-null     float64
 8   First year:   Sem 2   64 non-null     float64
 9   Second year:   Sem 1  64 non-null     float64
 10  Second year:   Sem 2  64 non-null     float64
dtypes: float64(4), int64(1), object(6)
memory usage: 5.6+ KB
```

In [11]:
```
print("columns of the dataset are ")
df.columns
```

Out[11]:
```
columns of the dataset are 
Index(['Timestamp', 'Email Address', 'Name', 'Email', 'Roll no ', 'PRN No.',
       'Mobile No.', 'First year:   Sem 1', 'First year:   Sem 2',
       'Second year:   Sem 1', 'Second year:   Sem 2'],
      dtype='object')
```

In [12]:
```
df.isnull().sum()
```

Out[12]:
```
Timestamp              0
Email Address          0
Name                   0
Email                  0
Roll no                0
PRN No.                0
Mobile No.             0
First year:   Sem 1    0
First year:   Sem 2    0
Second year:   Sem 1   0
```

```
Second year:   Sem 2    0
dtype: int64
```

HERE WE CAN SEE THAT THERE ARE NO NULL VALUES. HENCE THERE IS NO NEED OF DATA CLEANING OR REPLACING NULL VALUES

In [15]:
```
#converting roll no from TEC0C342 -> 342
for i in df['Roll no '].iteritems():
    data['Roll no '][i[0]]=data['Roll no '][i[0]][-3:]
df.head()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
/tmp/ipykernel_7728/340051193.py in ?()
      1 #converting roll no from TEC0C342 -> 342
----> 2 for i in df['Roll no '].iteritems():
      3     data['Roll no '][i[0]]=data['Roll no '][i[0]][-3:]
      4 df.head()

~/.local/lib/python3.12/site-packages/pandas/core/generic.py in ?(self, name)
   6317             and name not in self._accessors
   6318             and self._info_axis._can_hold_identifiers_and_holds_name(n
ame)
   6319         ):
   6320             return self[name]
-> 6321         return object.__getattribute__(self, name)

AttributeError: 'Series' object has no attribute 'iteritems'
```

In [16]:
```
python __version__
```

```
  Cell In[16], line 1
    python __version__
           ^
SyntaxError: invalid syntax
```

In [17]:
```
import sys
print(sys.version)
```

```
3.12.3 (main, Nov  6 2025, 13:44:16) [GCC 13.3.0]
```

In [18]:
```
# Converting the roll numbers from TECOC342 --> 342
for i in df.index:
    df.loc[i, 'Roll no '] = df.loc[i, 'Roll no '][-3:]
df.head()
'''df.loc[i, 'Roll no '] = df.loc[i, 'Roll no '][-3:]
df.loc[i, 'Roll no ']: This uses .loc[] to access a specific cell in the DataF
.loc[] is used to select specific rows and columns from a DataFrame. Here, it'


'''df.loc[i, 'Roll no '][-3:]:
df.loc[i, 'Roll no '] retrieves the value in the 'Roll no ' column for row i (
[-3:]: This is Python's slice notation applied to the string. The [-3:] slice
For example, "TECOC342"[-3:] would return "342" '''
```

Out[18]:

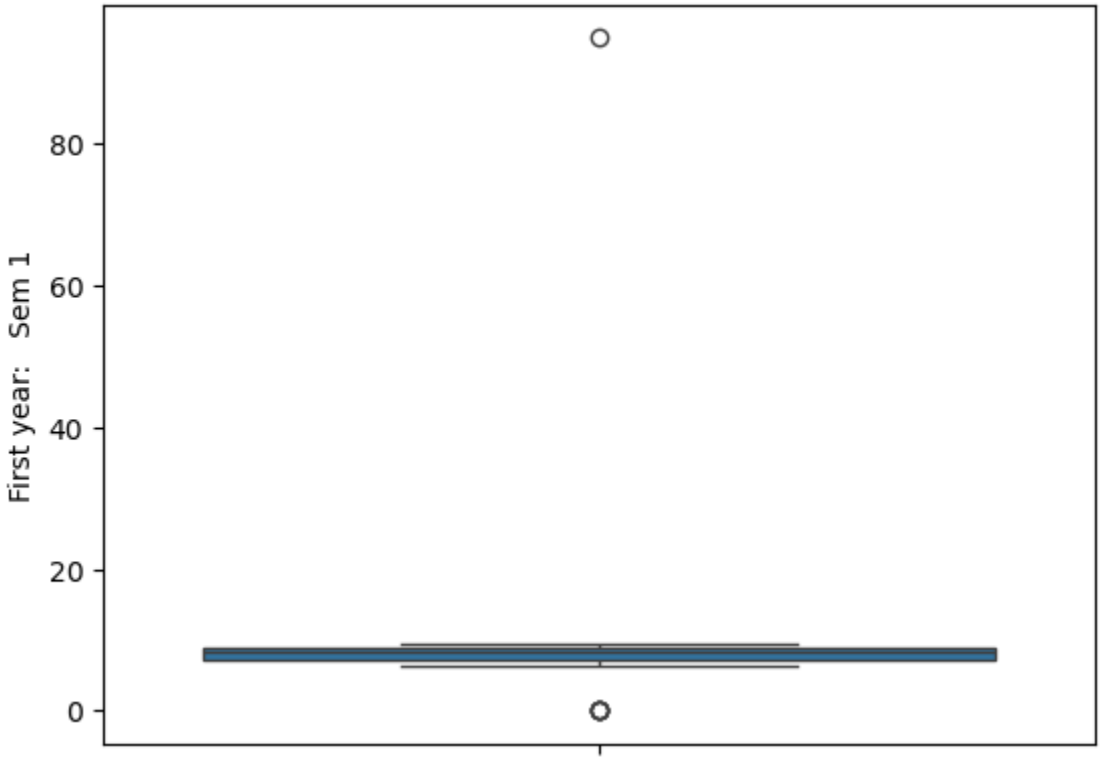
	Timestamp	Email Address	Name	Email
0	1/17/2022 12:45:09	sejal.zambare19@pccoepune.org	Sejal Zambare	sejal.zambare19@gmail.com

	Timestamp	Email Address	Name	Email
1	1/17/2022 12:45:44	rushikesh.thorat19@pccoepune.org	Rushikesh Vilas Thorat	rushikesh.thorat19@pccoepune.org
2	1/17/2022 12:46:10	atharv.sontakke19@pccoepune.org	Atharv Sontakke	atharv123sontakke@gmail.com
3	1/17/2022 12:46:21	amisha.sherekar19@pccoepune.org	Amisha Sunil Sherekar	amisha.sherekar19@pccoepune.org
4	1/17/2022 12:46:31	saurabh.sawardekar19@pccoepune.org	Saurabh Raju Sawardekar	saurabh.sawardekar19@pccoepune.org

OUTLINERS : Outliers are data points that significantly differ from the other observations in a dataset. They are extreme values that can be much higher or lower than most of the other data

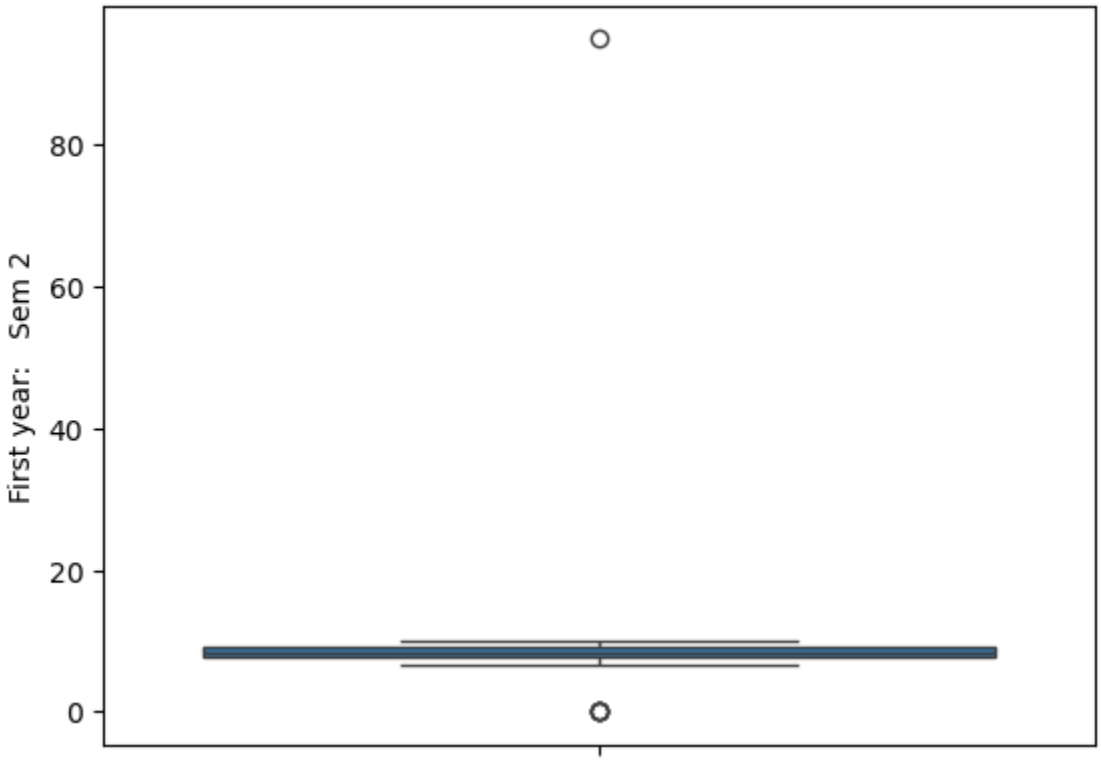
In [23]:
```
sns.boxplot(y=df['First year:   Sem 1'])
'''sns.boxplot():
sns refers to Seaborn, a Python visualization library based on Matplotlib. It
boxplot() is a type of plot that displays the distribution of a dataset based

'''y=df['First year: Sem 1']:
y= specifies the data to be plotted on the y-axis. In this case, the data come
df['First year: Sem 1'] refers to a column in your DataFrame df, which presuma
```

Out[23]:
```
"y=df['First year: Sem 1']:\ny= specifies the data to be plotted on the y-axi
s. In this case, the data comes from the column 'First year: Sem 1' of the Dat
aFrame df.\ndf['First year: Sem 1'] refers to a column in your DataFrame df, w
hich presumably contains the data for the first-semester marks or scores for a
group of students in their first year."
```



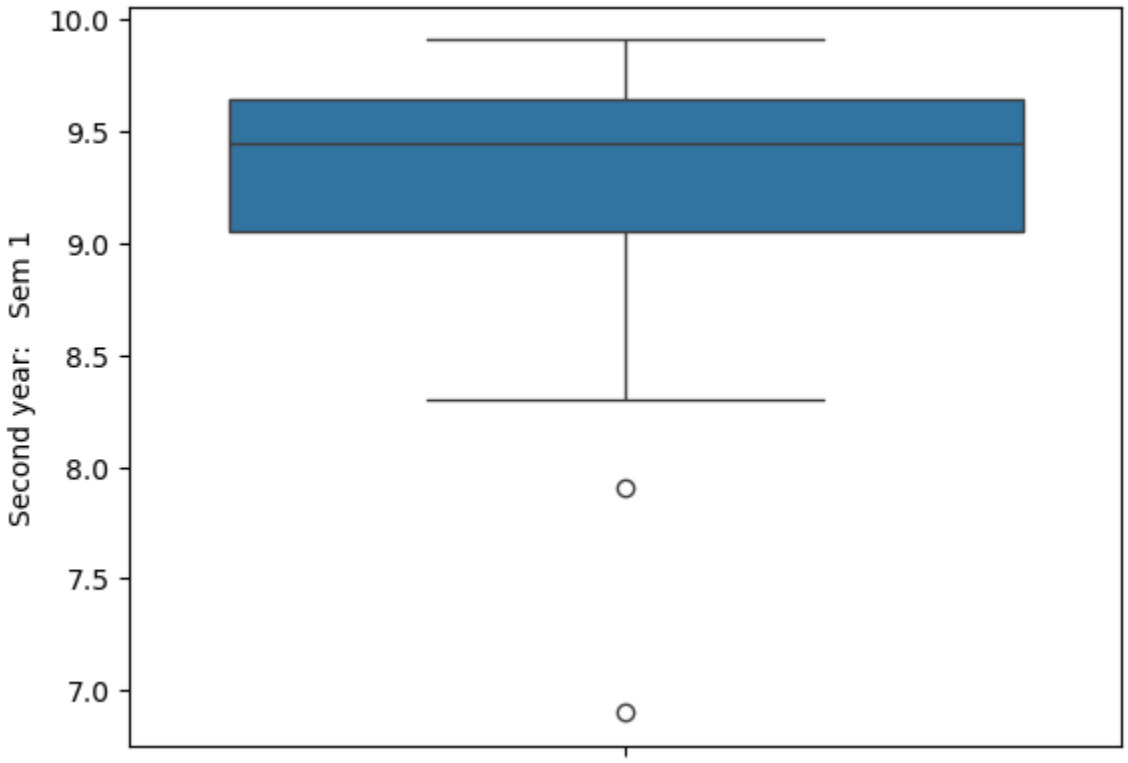
In [20]:
```
sns.boxplot(y=df['First year:   Sem 2'])
```

Out[20]:
```
<AxesSubplot: ylabel='First year:   Sem 2'>
```



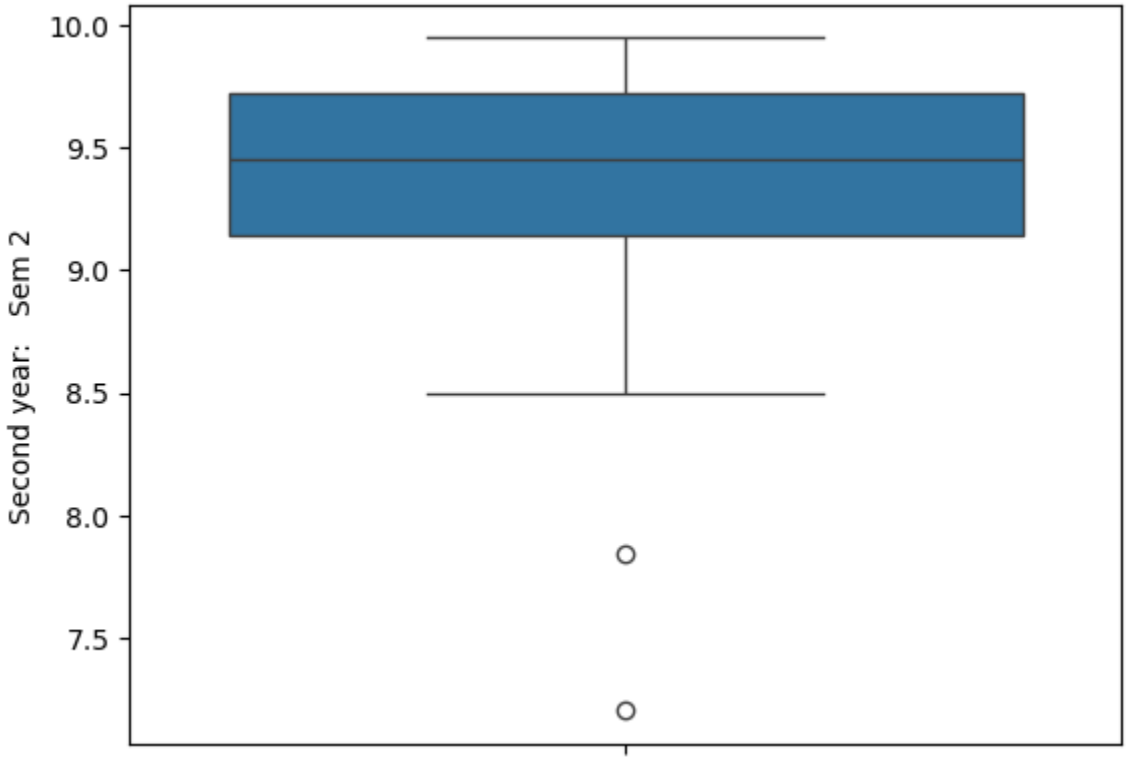
In [24]:
```
sns.boxplot(y=df["Second year:   Sem 1"])
'''sns.boxplot():
sns refers to Seaborn, a Python visualization library based on Matplotlib. It
boxplot() is a type of plot that displays the distribution of a dataset based

'''y=df['First year: Sem 1']:
y= specifies the data to be plotted on the y-axis. In this case, the data come
df['First year: Sem 1'] refers to a column in your DataFrame df, which presuma
```

Out[24]:
```
"y=df['First year: Sem 1']:\ny= specifies the data to be plotted on the y-axi
s. In this case, the data comes from the column 'First year: Sem 1' of the Dat
aFrame df.\ndf['First year: Sem 1'] refers to a column in your DataFrame df, w
hich presumably contains the data for the first-semester marks or scores for a
group of students in their first year."
```



In [22]:
```
sns.boxplot(y=df["Second year:   Sem 2"])
```

Out[22]:
```
<AxesSubplot: ylabel='Second year:   Sem 2'>
```



In [ ]: