DEEP LEARNING SECTION : 1

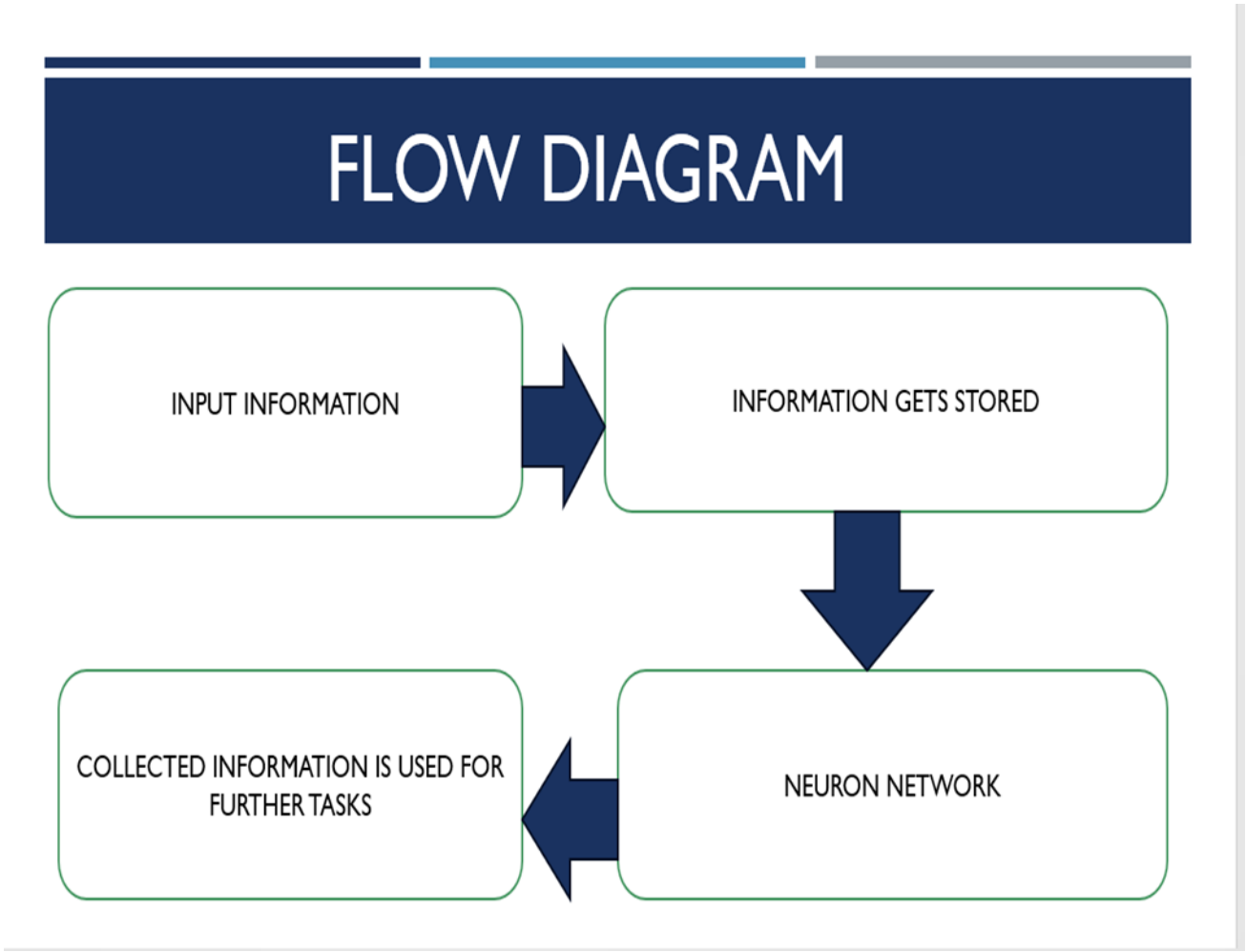
TEAM NAME : INNOVATORS

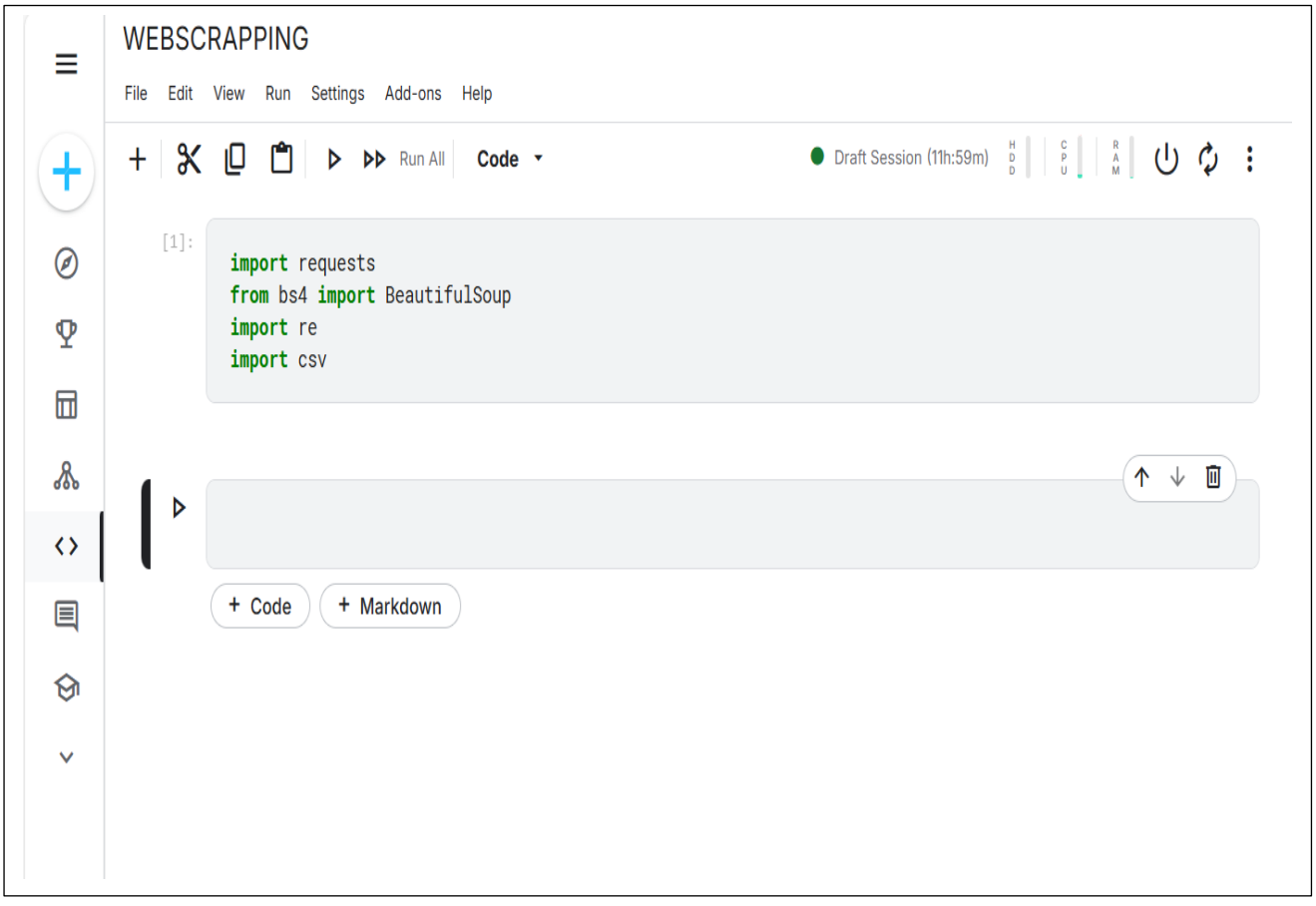
STATEMENT

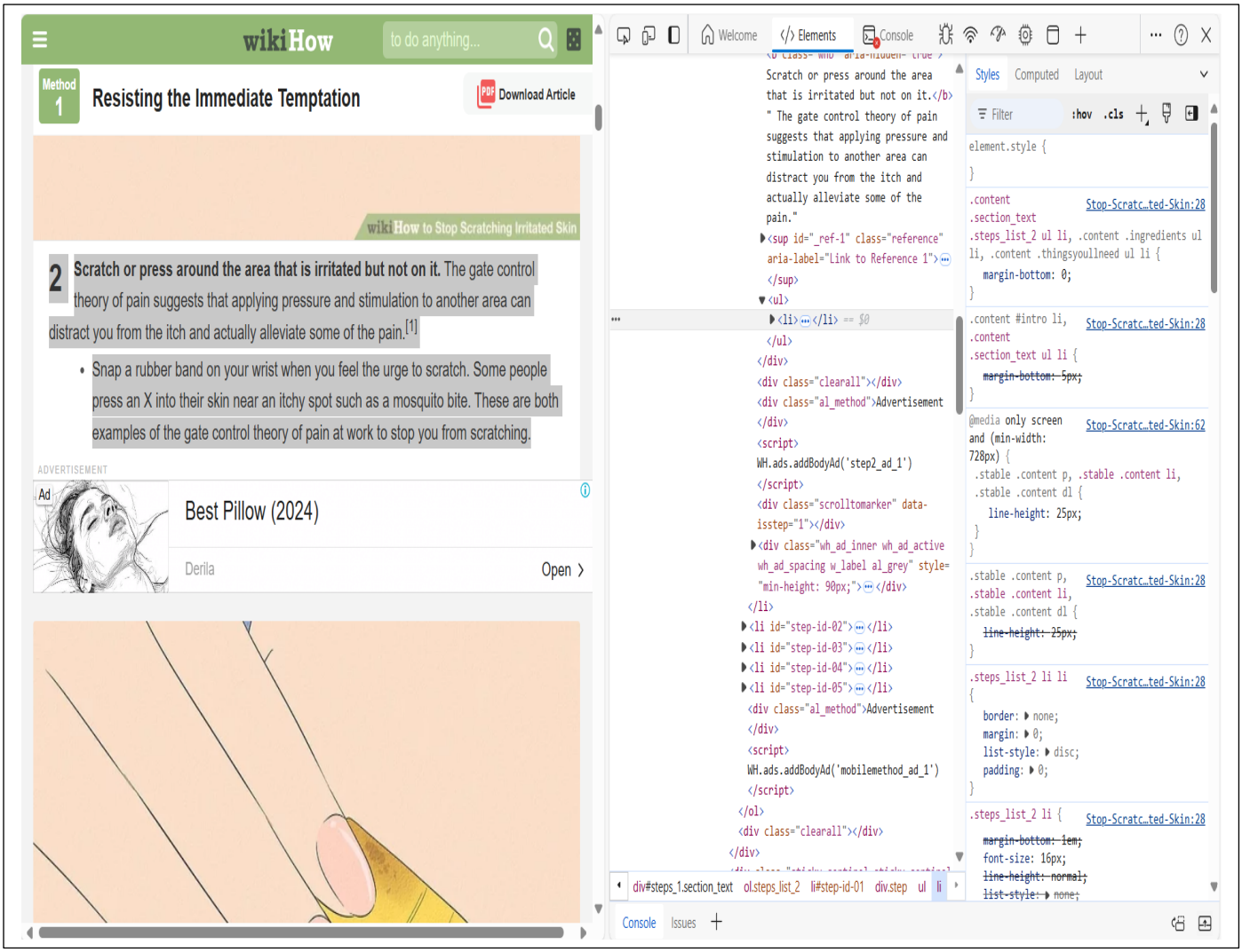
- The statement focuses on to create a fine tuned model in which the information is given to the system ant it accumulates the raw data and it utilises it for further tasks that to be given

OBJECTIVE

- The main objective of the mentioned statement to give information to the database and it utilises that it for final specified tasks
- It's like teaching the child and the child utilises the knowledge in his future endeavours
- The datas were collected and stored in a network called neuron network
- The detailed information of the above mentioned were explained briefly in the upcoming.


FLOW DIAGRAM
INPUT INFORMATION
INFORMATION GETS STORED
NEURON NETWORK
COLLECTED INFORMATION IS USED FOR FURTHER TASKS


WEBSCRAPPING
File Edit View Run Settings Add-ons Help
Run All
Code
Draft Session (11h:59m)
[1]:
import requests
from bs4 import BeautifulSoup
import re
import csv
+ Code
+ Markdown




```
1  if(len(subheadings)):
2      with open('/kaggle/working/wikiHow.csv', mode='a', newline='', encoding='utf-8') as csv_file:
3          writer = csv.writer(csv_file)
4          for i in range (len(subheadings)):
5              writer.writerow([article_title, subheadings [i], paragraphs[i]])
```

```
  Cell In[13], line 1
    1  if(len(subheadings)):
                            ^
SyntaxError: invalid syntax
```

+ Code + Markdown

```
html_content response.content
# Parse the HTML content using BeautifulSoup
soup = BeautifulSoup(html_content, 'html.parser")
article_title soup.find('title').text.strip()
print(article_title+" "+str(count))
# Extract the subheadings and paragraphs using the appropriate HTML tags
subheadings = []
paragraphs = []
steps soup.find_all('div', {'class': 'step'})
for step in steps:
subheading_element = step.find('b')
if(subheading_element is not None):
subheading text subheading_element.text.strip().replace('\n', '') subheading_text subheading_text.encode('ascii", errors-'ignore').decode()
subheading_text = re.sub(",", subheading_text)
subheadings.append(subheading_text)
subheading_element.extract()
for span_tag in step.find_all('span'):
span_tag.extract()
paragraph_text = step.text.strip().replace('\n', '').replace(',') paragraph_text = paragraph_text.encode('ascii", errors='ignore'), decode()
paragraph_text = re.sub(",", paragraph_text)
paragraphs.append(paragraph_text)
if(len(subheadings)):
with open('/kaggle/working/wikiHow.csv", mode="a", newline='", encoding='utf-8') as csv_file:
writer csv.writer(csv_file)
for i in range(len(subheadings)):
writer.writerow([article_title, subheadings [1], paragraphs[i]])
```