Experiment No: 7	
Name	Vaibhav Sharma
PRN	22070126125
Date of Performance	16th October 2024
Title	To split the given URL into the said components
Theory (short)	A URL (Uniform Resource Locator) is the address used to access resources on the internet, such as websites or files. It consists of several components, including the protocol, domain or host, port, and path. The process of URL splitting involves parsing these components from a given URL string. **Components of a URL:** 1. **Protocol (Scheme)**: Defines the method used to access the resource, such as http, https, or ftp. This is essential for communication between the client and server. 2. **Domain/Host**: This is the unique address that identifies the location of the server (e.g., suno.com). The domain is crucial for directing requests to the appropriate server. 3. **Port**: Specifies the network port on which the resource is accessible. For web pages, the default port for http is 80, and for https, it is 443. If no port is provided, the default is assumed. 4. **Path**: Refers to the specific resource within the domain. For example, /about might represent the "About" page of a website.
Procedure	**Import Required Module**: 1) Start by importing the `urlparse` function from the `urllib.parse` module. This function helps in parsing the URL into its individual components. **Python- from urllib.parse import urlparse** **Define the Function**: 2) Create a function `url_splitter()` that takes a URL string as input. This function will parse the URL and extract its parts. **Python- def url_splitter(url):** **Parse the URL**:

3) Use urlparse(url) to parse the input URL string into its components, which include the protocol, host, port, and path. This parsed result is stored in a variable, parsed_url.

   Python- parsed_url = urlparse(url)

Extract Components:

4) Extract each component from parsed_url, checking if the component exists. If a component is missing (empty), set it to "Does not Exist".
   a. **Protocol**: Extracted from the scheme attribute.
   b. **Host/Domain**: Extracted from the hostname attribute.
   c. **Port**: Extracted from the port attribute.
   d. **Path**: Extracted from the path attribute.

Python-
protocol = parsed_url.scheme if parsed_url.scheme else "Does not Exist"
host = parsed_url.hostname if parsed_url.hostname else "Does not Exist"
port = parsed_url.port if parsed_url.port else "Does not Exist" path = parsed_url.path if parsed_url.path else "Does not Exist"

Print the Results:

5) After extracting each component, print them with descriptive labels: "Protocol", "Host/Domain", "Port", and "Path". If any component is missing, the function will print "Does not Exist" for that component.

   Python- print(f"Protocol: {protocol}")
   print(f"Host/Domain: {host}")
   print(f"Port: {port}")
   print(f"Path: {path}")

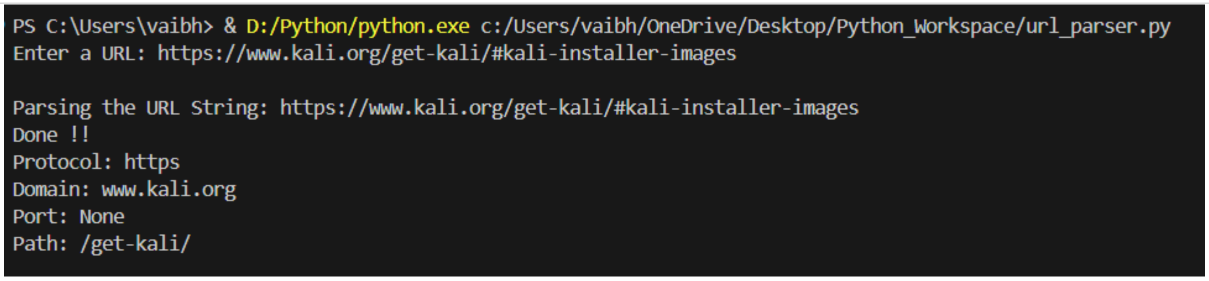
Test the Function:

6) Call the url_splitter() function with the URL https://suno.com to check the output. This URL doesn't have a port or path, so it should print "Does not Exist" for these components.

   Python-
   url = "https://suno.com" url_splitter(url)

Full python code:

```python
from urllib.parse import urlparse

def parse_url(url):


  parsed_url = urlparse(url)
  return {
    "protocol": parsed_url.scheme,
    "domain": parsed_url.netloc,
    "port": parsed_url.port,
    "path": parsed_url.path
  }

def main():

  url = input("Enter a URL: ")

  parsed_url = parse_url(url)

  print("""
Parsing the URL String: {}
Done !!
Protocol: {}
Domain: {}
Port: {}
Path: {}
""".format(url, parsed_url["protocol"], parsed_url["domain"],
parsed_url["port"], parsed_url["path"]))

if __name__ == "__main__":
  main()
```

Output Screenshots	 ```PS C:\Users\vaibh> & D:/Python/python.exe c:/Users/vaibh/OneDrive/Desktop/Python_Workspace/url_parser.py``` ```Enter a URL: https://www.kali.org/get-kali/#kali-installer-images``` ```Parsing the URL String: https://www.kali.org/get-kali/#kali-installer-images``` ```Done !!``` ```Protocol: https``` ```Domain: www.kali.org``` ```Port: None``` ```Path: /get-kali/``` Fig 1- All the components of the link dfe
Observation	As the URL is very simple, we do not have the Port number and the path of the component in the URL. However, if you would like to experiment it by yourself, you can always change the URL variable and add a URL which you commonly use or any other URL works.
Self-assessment Q&A	Q: What is the purpose of the protocol component in a URL? Ans: The protocol (e.g., http, https) defines the method used for communication between the client and the server. Q: What does the domain or host component represent in a URL? Ans: The domain or host is the unique address that identifies the server where the resource is located, directing requests appropriately. Q: Why is the port important in a URL, and what are the default ports for HTTP and HTTPS? Ans: The port specifies where the resource is accessible; the default port for HTTP is 80, and for HTTPS, it is 443.
Conclusion	In this program, we demonstrated how to effectively split a URL into its key components: protocol, host (or domain), port, and path using Python's urllib.parse module. By employing the urlparse function, we can easily extract each part of the URL, enabling us to work with and analyze URLs in a structured manner. The program also accounts for missing components by providing a default response of "Does not Exist" when certain parts, like the port or path, are absent. This approach is useful in various applications such as web scraping, URL validation, and network programming, where understanding the structure of URLs is essential for proper functionality and communication with web resources.