|  |  |  |
| --- | --- | --- |
| **Practical 1** | **Data Storage and Retrieval** | **Date: 04/08/2025** |
| **Aim:** | Storage and Retrieval | |
| **Requirement:** | Python IDE | |
| **Code:** | Code:  my\_list=[40,50,60,'bye',False]  another\_list=[]  another\_list.append('mango')  another\_list.append('watermelon')  print(my\_list[0])  print(my\_list[3])  for item in my\_list:  print(item)  my\_dict = {  'name':'Alex',  'age': 33,  'city':'Melbourne'  }  another\_dict={}  another\_dict['product']='Computer'  another\_dict['price']=1600  print(my\_dict['name'])  print(my\_dict.get('country','Norway'))  for key, value in my\_dict.items():  print(f"{key}:{value}")  my\_set={1,2,3,4}another\_set=set()  another\_set.add('yellow')  another\_set.add('orange')  print(my\_set)  if 'orange' in another\_set:  print("Orange is in this set.") | |
| **Output:** | 40  bye  40  50  60  bye  False  Alex  Norway  name:Alex  age:33  city:Melbourne  {1, 2, 3, 4}  Orange is in this set. | |
| **Result:** | Storing and retrieving data using dictionaries and set, list. | |
| **Date of Submission:** | | **Sign:** |

|  |  |  |
| --- | --- | --- |
| **Practical 2** | **Hadoop Installation** | **Date: 18/09/2025** |
| **Aim:** | To install and configure **Hadoop** on a **Linux environment (Ubuntu/WSL)**, including setting up Java, Hadoop environment variables, and verifying installation using HDFS commands. | |
| **Requirement:** | Python IDE , PowerShell (must be run as Administrator) | |
| **Command:** | 1. wsl --install -d Ubuntu 2. Enter new UNIX username: fintech 3. New password: password 4. cd /home 5. ls 6. cd fintech 7. sudo su >password >root 8. apt update && apt upgrade -y 9. apt install openjdk-11-jdk wget ssh tar -y 10. java --version 11. wget https://dlcdn.apache.org/hadoop/common/hadoop-3.3.6/hadoop-3.3.6.tar.gz 12. tar -xzvf hadoop-3.3.6.tar.gz 13. mv hadoop-3.3.6/usr/local/hadoop 14. cd /home 15. cd fintech 16. nano .bashrc or nano ~/.bashrc 17. Path setup : 18. Ctrl O <enter> Ctrl x (save and exit) 19. Source ~/.bashrc 20. hadoop version | |
| **Output:** |  | |
| **Result:** | Installation of Hadoop 3.3.6 | |
| **Date of Submission:** | | **Sign:** |

|  |  |  |
| --- | --- | --- |
| **Practical 3** | **Word Counting using Mapreduce in Hadoop** | **Date:**  25/09/2025 |
| **Aim:** | To perform **Word Count using Hadoop Streaming** by writing a Python mapper and reducer, executing them on input data through HDFS, and displaying the final word frequency output. | |
| **Requirement:** | Python IDE , PowerShell (must be run as Administrator) | |
| **Code:**  **mapper.py** |  | |
| **Code:**  **reducer.py** |  | |
| **Command:** | # Step 1: Start Ubuntu WSL  wsl -d Ubuntu  # Step 2: Enter password  Password: password  # Step 3: Switch to root user  wsl -d Ubuntu -u root  # Step 4: Check files and navigate  ls  cd Onedrive  cd Desktop  mv mapper.py /  mv reducer.py /  # Step 6: Go to root directory  cd /  # Step 7: Install dos2unix to fix line endings  apt install dos2unix  # Step 8: Convert Python files to Unix format  dos2unix mapper.py reducer.py  # Step 9: Make Python files executable  chmod +x mapper.py reducer.py  # Step 10: Create input file  echo "hello world nano hello map reduce hello hello" > input.txt  # Step 11: Test mapper and reducer locally  cat input.txt | ./mapper.py | sort | ./reducer.py  # Step 12: Create input directory in HDFS  hdfs dfs -mkdir -p /input  # Step 13: Put input file into HDFS  hdfs dfs -put -f input.txt /input  # Step 14: Run Hadoop streaming job  hadoop jar $HADOOP\_HOME/share/hadoop/tools/lib/hadoop-streaming-3.3.6.jar \  -files mapper.py,reducer.py \  -mapper mapper.py \  -reducer reducer.py \  -input /input \  -output /output  # Step 15: View output from HDFS  hdfs dfs -cat /output/part-00000 | |
| **Output:** |  | |
| **Date of Submission:** | | **Sign:** |

|  |  |  |
| --- | --- | --- |
| **Practical 4** | **Sorting Numbers using Mapreduce in Hadoop** | **Date:**  26/09/2025 |
| **Aim:** | To implement a **Hadoop Streaming MapReduce program in Python** to **sort numbers** from an input file stored in HDFS and display the sorted output. | |
| **Requirement:** | Python IDE , PowerShell (must be run as Administrator) | |
| **Code:**  **Mapper\_sort.py** |  | |
| **Code:**  **Reducer\_sort.py** |  | |
| **Command:** | # Step 1: Start Ubuntu WSL  wsl -d Ubuntu  # Step 2: Go to root directory  cd /  # Step 3: Get root access  sudo su  Password: password  # Step 4: Move mapper and reducer files to root  mv /home/fintech/mapper\_sort.py /  mv /home/fintech/reducer\_sort.py /  # Step 5: List files  ls  # Step 6: Convert files to Unix format  dos2unix mapper\_sort.py reducer\_sort.py  # Step 7: Give execution rights (green = executable)  chmod +x mapper\_sort.py reducer\_sort.py  # Step 8: Create input file with numbers  nano number.txt  # (Inside nano → type numbers → Ctrl+O, Enter, Ctrl+X to save & exit)  # Step 9: Create input directory in HDFS  hdfs dfs -mkdir /numbers  # Step 10: Put input file into HDFS  hdfs dfs -put number.txt /numbers/input.text  # Step 11: Run Hadoop streaming job  hadoop jar $HADOOP\_HOME/share/hadoop/tools/lib/hadoop-streaming-3.3.6.jar \  -files mapper\_sort.py,reducer\_sort.py \  -input /numbers \  -output /numbers\_out \  -mapper mapper\_sort.py \  -reducer reducer\_sort.py  # Step 12: View sorted output  hdfs dfs -cat /numbers\_out/part-00000 | |
| **Output:** |  | |
| **Date of Submission:** | | **Sign:** |