UNIVERSITY INSTITUTE of COMPUTING
Asia's Fastest Growing University

NAAC GRADE A+ ACCREDITED UNIVERSITY

Student Name: Sweta Kumari
UID: 24MCC20017
Branch: MCA(CCD)
Section/Group: 1-A
Semester: 2nd
Subject Code: 24CAP-684
Subject Name: Big Data Lab

Library Management System

Aim of the project:

The aim of this mini project is to create a Library Management System that allows users to add and view books. The system will utilize Hadoop for scalable and distributed data storage, running on an Ubuntu operating system.

Objective :

- Core Functionality: Add and view book records easily.
- User Input: Command-line interface for book details (title and author).
- Data Storage: Use Hadoop's HDFS to save book information.
- Input Validation: Ensure valid book details are provided.
- Exit Option: Allow users to exit the program.
- Simplicity: Maintain clarity for easy understanding.

Task to be done:

1. Set Up the Environment:
    - Install Ubuntu.
    - Install and configure Hadoop on Ubuntu.
    - Use Python for programming.
2. Design the Program Structure:
    - Outline program flow (menu: add book, view books, exit).
    - Decide on data format for HDFS storage.
3. Implement Functionality:
    - Add Book: Collect and validate user input, write to HDFS.
    - View Books: Read and display data from HDFS.
4. Create Main Menu:
    - Develop a loop for menu options and user selection.
5. Exit Functionality:
    - Ensure graceful program exit.

Code for experiment:

```
mkdir ~/library_management
cd ~/library_management
touch main.py database.py
```

- **Main.py file:**

```
from database import Database

def main():
    db = Database()
    while True:
        print("Library Management System")
        print("1. Add Book")
        print("2. View Books")
        print("3. Exit")
        choice = input("Enter your choice: ")

        if choice == '1':
            title = input("Enter book title: ")
            author = input("Enter book author: ")
            if title and author:
                db.add_book(title, author)
            else:
                print("Both title and author must be provided.")
        elif choice == '2':
            books = db.view_books()
            if books:
                for book in books:
                    print(f"Title: {book[0]}, Author: {book[1]}")
            else:
                print("No books found.")
        elif choice == '3':
            break
        else:
            print("Invalid choice! Please try again.")

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

- **For database.py**

```python
from hdfs import InsecureClient

class Database:
    def __init__(self):
        self.client = InsecureClient('http://localhost:9870', user='hadoop')
        self.file_path = '/user/hadoop/library_books.txt'

    def add_book(self, title, author):
        book_entry = f"{title},{author}\n"
        with self.client.write(self.file_path, append=True) as writer:
            writer.write(book_entry)

    def view_books(self):
        try:
            with self.client.read(self.file_path) as reader:
                data = reader.read().decode('utf-8').strip().split('\n')
                return [line.split(',') for line in data if line]
        except Exception as e:
            print(f"Error reading from HDFS: {e}")
            return []
```

Output:

```
Library Management System
1. Add Book
2. View Books
3. Exit
Enter your choice: 1
Enter book title: To Kill a Mockingbird
Enter book author: Harper Lee

Library Management System
1. Add Book
2. View Books
3. Exit
Enter your choice: 2
Title: To Kill a Mockingbird, Author: Harper Lee
```

Learning outcomes :

- Gained experience in installing and using Hadoop and HDFS for distributed data storage.
- Improved Python coding skills, focusing on file handling and user input validation.
- Learned to manage and manipulate data in a distributed environment with HDFS.
- Developed a simple command-line interface for better user interaction.
- Gained insights into planning, coding, testing, and debugging while ensuring code clarity.