Date.....

Assignment - 4.

Name- TARUN SHARMA B.Tech CSE(AI/ML) Sec-B.

RNO- 2401720104

```
import java.io.*;
import java.util.*;

Class Book implements Comparable<Book>{
    int bookId;
    String title, author, category;
    boolean isIssued;

    Book (int id, String t, String a, String c){
        bookId = id;
        title = t;
        author = a;
        category = c;
    }

    void displayBookDetails(){
        System.out.println(bookID + ")" + title + ")" +
        author + ")" + category + ") Issued: " + isIssued);
    }

    void markAsIssued(){ isIssued = true; }
    void markAsReturned(){ isIssued = false; }

    public int compareTo(Book b){
        return title.compareToIgnoreCase(b.title);
    }
}
```

Spiral

Date.....

```
class AuthorComparator implements Comparator<Book>{
    public int compare(Book a, Book b)
    {return a.author.compareToIgnoreCase(b.author);}
}

class Member {
    int memberID;
    String name, email;
    List<Integer> issuedBooks = new ArrayList<>();

    Member(int id, String n, String e){
        memberID = id;
        name = n;
        email = e;
    }
    void displayMemberDetails(){
        System.out.println(memberId + "|" + name +
        "|" + email + "|" + issued + "|" issuedBooks);
    }
    Void addIssuedBook(int b)
{ issuedBooks.add(b); }
    void returnIssuedBook(int b){
        issuedBooks.remove(Integer.valueOf(b)); }
}

public class LibraryManager {
    Map<Integer, Book> books = new HashMap<>();
    Map<Integer, Member> members = new HashMap
    <>();

    Set<String> categories = new HashSet<>();
    Scanner sc = new Scanner(System.in);
```

Date.....

int nextBookId = 101, nextMemberId = 1;

void saveToFile() {
try (BufferedWriter bw = new BufferedWriter(new
FileWriter("books.txt"))) {
for (Book b : books.values())
bw.write(b.bookId + "," + b.title
+ "," + b.author + "," + b.category +
"," + b.isIssued + "\n");
} catch (Exception e) {}
}

void loadFromFile() {
try (BufferedReader br = new FileReader
("books.txt")) {
String line;
while ((line = br.readLine()) != null)
String[] p = line.split(",");
Book b = new
Book(Integer.parseInt(p[0]), p[1], p[2],
p[3]);
b.isIssued = Boolean.parseBoolean(p[4]);
books.put(b.bookId, b);
categories.add(p[3]);
nextBookId =
Math.max(nextBookId, b.bookId + 1);
}
} catch (Exception e) {}
}

Date.....

```
Void addBook () {
    sc.nextLine();
    System.out.println()
    String t = sc.nextLine();

    System.out.println("Enter Author:");
    String a = sc.nextLine();

    System.out.print("Enter Category:");
    String c = sc.nextLine();

    Book b = new Book(nextBookId++, t, a, c);
    books.put(b.bookId, b);
    Categories.add(c);
    saveToFile();

    System.out.println("Book added successfully" +
        b.bookId + "\n");
}

Void addMember() {
    SC.nextLine();
    System.out.print("Name");
    String n = sc.nextLine();
    System.out.print("Email:");
    String e = sc.nextLine()

    Member m = new Member(nextMemberId++,
        n, e);
    members.put(m.memberId, m);
    saveToFile();
```

Signal

Teacher's Sign

Date.....

```
System.out.println("Member added with id"+ m.memberId"
                    + "\n");
}

void IssueBook(){
    System.out.print("BookId");
    int b = sc.nextInt();
    System.out.print("memberId:");
    int m = sc.nextInt();

    if (books.containsKey(b) && members.contains
        Key(m)){
        books.get(b).markAsIssued();
members.get(m).addIssueBook(b);
        saveToFile();
        Sout("Book issued"+"\n");
    }
}

void returnBook(){
    System.out.print("Book id");
    int b = sc.nextInt();
    System.out.print("MemberID:");
    int m = sc.nextInt();

    if(books.containsKey(b) && member.contains
        key(m)){
        books.get(b).markAsReturned();

        members.get(m).returnIssuedBook(b);
        saveToFile();
        Sout("Book returned");
    }
}
```

Spiral

Teacher's Sign .....

Date.....

```
void searchBooks(){
    sc.nextLine();
    System.out.println("Search");
    String k = sc.nextLine().toLowerCase();
    for(Book b : books.values()){
        if(String.valueOf(b.bookId).equals(k) ||
        b.title.toLowerCase().contains(k) ||
        b.author.toLowerCase().contains(k) ||
        b.category.toLowerCase().contains(k) ||
        b.category.toLowerCase().contains(k)){
            b.displayBookDetails();
        }
        System.out.println();
}

void sortBooks(){
    List<Book> list = new ArrayList<>(
        books.values());
    System.out.println("1. Sort by Title\n2.
        Sort by Author");
    int ch = sc.nextInt();
    if(ch==1) Collections.sort(list);
    else Collections.sort(list, new
    AuthorComparator());
    for(Book b : list)
        sout(b)
            sout();
}
```

Spiral

Teacher's Sign .....

Date.....

```
public void menu(){
    loadFromFile();
    int c;
    do{
        Sout("1.Add Bank \n 2. Add Member\n 3. Issue \n 4. Return \n 5. Search \n
6. Sort \n 7. Exit");
        c = sc.nextInt();
        Switch(c){
            Case 1: addBook(); break;
            Case 2: addMember(); break;
            Case 3: IssueBook(); break;
            Case 4: returnBook(); break;
            Case 5: searchBooks(); break
            Case 6: SortBooks(); break;
        }
    }while(c != 7);
}
public static void main(String[] args){
    new LibraryManagement().menu();
}
}
```