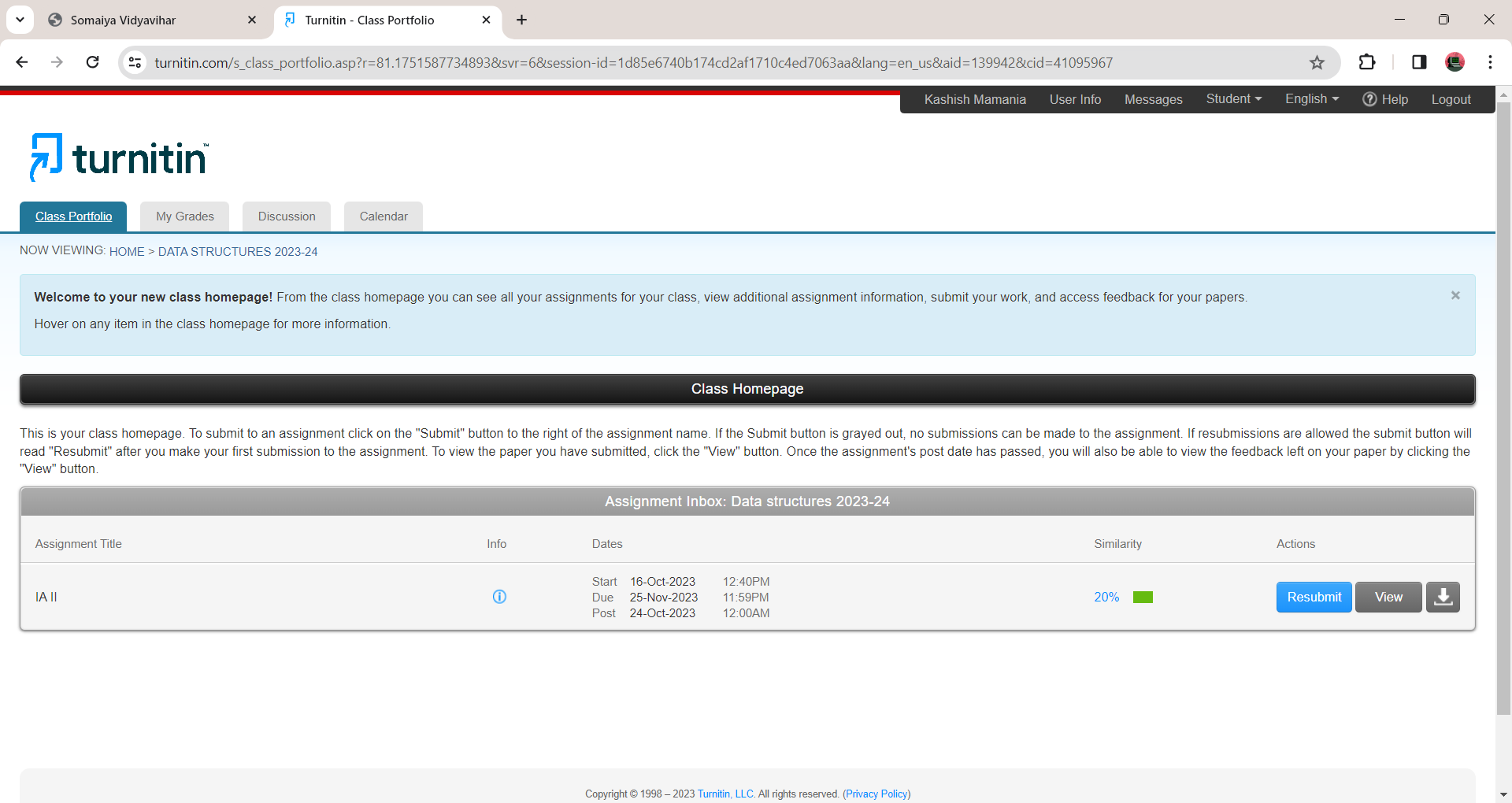
PPT Video link:-

<https://drive.google.com/file/d/1xw7Yqw_g_JEKqoj533gboGPTYwkKNIRH/view?usp=sharing>

PPT link:-

[DS-Library Management System.pdf](https://drive.google.com/file/d/1OE_Wac1uZVJ6Lh4wKi4lPeCci5xZWtI4/view?usp=sharing)



#include <string.h>

#include <stdlib.h>

#include <stdio.h>

#define TABLE\_SIZE 10

struct Book {

char author[50];

char BKNO[13];

char title[100];

struct Book\* n;

};

struct User {

char username[50];

};

struct QN {

struct Book\* book;

struct User user;

struct QN\* n;

};

struct Book\* bookTable[TABLE\_SIZE] = { NULL };

struct QN\* queue = NULL;

int hash(char BKNO[13]) {

int indx = 0;

for (int i = 0; i < 13; i++) {

indx += BKNO[i];

}

return indx % TABLE\_SIZE;

}

void addBook(char title[100], char author[50], char BKNO[13]) {

int indx = hash(BKNO);

struct Book\* newBook = (struct Book\*)malloc(sizeof(struct Book));

strcpy(newBook->title, title);

strcpy(newBook->author, author);

strcpy(newBook->BKNO, BKNO);

newBook->n = NULL;

if (bookTable[indx] == NULL) {

bookTable[indx] = newBook;

}

else {

struct Book\* c = bookTable[indx];

while (c->n) {

c = c->n;

}

c->n = newBook;

}

}

int removeBook(char BKNO[13]) {

int indx = hash(BKNO);

struct Book\* c = bookTable[indx];

struct Book\* prev = NULL;

while (c != NULL) {

if (strcmp(c->BKNO, BKNO) == 0) {

if (prev == NULL) {

bookTable[indx] = c->n;

}

else {

prev->n = c->n;

}

free(c);

return 1;

}

prev = c;

c = c->n;

}

return 0;

}

void borrowBook(char BKNO[13], char username[50]) {

int indx = hash(BKNO);

struct Book\* c = bookTable[indx];

struct Book\* prev = NULL;

while (c != NULL) {

if (strcmp(c->BKNO, BKNO) == 0) {

struct QN\* newNode = (struct QN\*)malloc(sizeof(struct QN));

newNode->book = c;

strcpy(newNode->user.username, username);

newNode->n = NULL;

if (queue == NULL) {

queue = newNode;

}

else {

struct QN\* tmp = queue;

while (tmp->n) {

tmp = tmp->n;

}

tmp->n = newNode;

}

if (prev == NULL) {

bookTable[indx] = c->n;

}

else {

prev->n = c->n;

}

return;

}

prev = c;

c = c->n;

}

}

int returnBook(char BKNO[13]) {

struct QN\* c = queue;

struct QN\* prev = NULL;

while (c != NULL) {

if (strcmp(c->book->BKNO, BKNO) == 0) {

if (prev == NULL) {

queue = c->n;

}

else {

prev->n = c->n;

}

free(c);

return 1;

}

prev = c;

c = c->n;

}

return 0;

}

void displayLib() {

for (int i = 0; i < TABLE\_SIZE; i++) {

printf("Slot %d:\n", i);

struct Book\* c = bookTable[i];

while (c != NULL) {

printf(" Title: %s\n", c->title);

printf(" Author: %s\n", c->author);

printf(" BKNO: %s\n", c->BKNO);

c = c->n;

}

}

}

void disborrowedbk() {

struct QN\* c = queue;

if (c == NULL) {

printf("No books are currently borrowed.\n");

return;

}

printf("Borrowed Books:\n");

while (c != NULL) {

printf(" Title: %s\n", c->book->title);

printf(" Author: %s\n", c->book->author);

printf(" BKNO: %s\n", c->book->BKNO);

printf(" Borrowed by: %s\n", c->user.username);

c = c->n;

}

}

int main() {

int CHOICE;

char title[100], author[50], BKNO[13], username[50];

while (1) {

printf("\nLibrary Management System Menu:\n");

printf("1. Add a Book\n");

printf("2. Remove a Book\n");

printf("3. Display Library\n");

printf("4. Display Borrowed Books\n");

printf("5. Return a Book\n");

printf("6. Borrow a Book\n");

printf("7. Exit\n");

printf("Enter your choice: ");

scanf("%d", &CHOICE);

switch (CHOICE) {

case 1:

printf("Enter book title: ");

scanf(" %[^\n]", title);

printf("Enter author: ");

scanf(" %[^\n]", author);

printf("Enter BKNO: ");

scanf(" %[^\n]", BKNO);

addBook(title, author, BKNO);

break;

case 2:

printf("Enter BKNO of the book to remove: ");

scanf(" %[^\n]", BKNO);

if (removeBook(BKNO))

printf("Book removed successfully.\n");

else

printf("Book not found.\n");

break;

case 3:

displayLib();

break;

case 4:

disborrowedbk();

break;

case 5:

printf("Enter BKNO of the book to return: ");

scanf(" %[^\n]", BKNO);

if (returnBook(BKNO))

printf("Book returned successfully.\n");

else

printf("Book not found or not borrowed.\n");

break;

case 6:

printf("Enter BKNO of the book to borrow: ");

scanf(" %[^\n]", BKNO);

printf("Enter your username: ");

scanf(" %[^\n]", username);

borrowBook(BKNO, username);

break;

case 7:

exit(0);

default:

printf("Invalid choice. Please re-enter again.\n");

}

}

return 0;

}