**PROJECT REPORT (CSF102)**

*A report submitted in partial fulfilment of the requirement for the course*

**PROGRAMMING FOR PROBLEM SOLVING**

Part of the degree of

BACHELOR OF COMPUTER SCIENCE AND ENGINEERING

A picture containing drawing

Description automatically generated

**Submitted to**

Mr. Sanjay Kumar

Assistant Professor

**Submitted by:**

Rafat Nazia - 1000021572 Btech CSE

**SCHOOL OF COMPUTING**

**DIT UNIVERSITY, DEHRADUN**

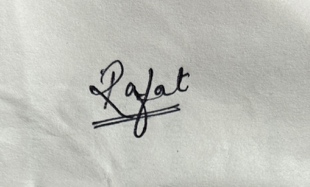
(State Private University through State Legislature Act No. 10 of 2013 of Uttarakhand and approved by UGC)

**Mussoorie Diversion Road, Dehradun, Uttarakhand - 248009, India.**

**November, 2023**

**CANDIDATES DECLARATION**

I hereby certify that the work, which is being presented in the Report, entitled “**To create Online Bookstore ”**, in partial fulfilment of the requirement as part of the course **Programming for Problem Solving** of the Degree of **Bachelor of Computer Science and Engineering** and submitted to the DIT University is an authentic record of my work carried out during the period ***5/04/2024***to **15/04/2024** under the guidance of Mr. Sanjay kumar.



**Date: 15/04/2024**

**Signature of the Candidate**

**INTRODUCTION**

**Nowadays, the network plays an import role in people’s life. In the process of the improvement of the people’s living standard, people’s demands of the life’s quality and efficiency is more higher, the traditional bookstore’s inconvenience gradually emerge, and the online bookstore has gradually be used in public. The online bookstore is a revolution of book industry. The traditional bookstores’ operation time, address and space is limited, so the types of books and books to find received a degree of restriction. But the online bookstore broke the management mode of traditional bookstore, as long as you have a computer, you can buy the book anywhere, saving time and effort, shortening the time of book selection link effectively. The online bookstore system based on the principle of provides convenience and service to people.**

**Which Data Struture is used in this program ?**

**1.Tree**

**2. String Array**

**Features of Online Bookstore:-**

1. **Broad Collection of eBooks**

A great eBook experience begins with a broad collection of available b eBooks, both installed and ready to read, and those that can be sourced.

1. Support Diverse Formats

The popularity of eBooks has led to an increase in the number of online eBook stores. There are various formats available today, including EPUB, MOBI, PDF, TXT files, and Word documents.

1. A Reader-Friendly User Interface

To build a successful online eBook store, a user-friendly design is crucial. The interface has to build a design suiting the reader’s needs enabling a smooth and effortless experience while reading books on mobile devices.

**Implementation :-**

**The aim of this thesis is to design and implement a web application as an online bookstore system, which consists of website interface and database. The system includes two website applications: customer application and administrator application. The stock manager is one type of administrator.**

**Why we use binary Search Tree:-**

 Binary trees are often chosen over more fundamental structures, such as arrays and linked lists, because you can search a binary tree quickly (as opposed to a linked list) and you can quickly insert data and delete data from a binary tree (as opposed to an array).

**Functions Used in this project:-**

1. struct BookNode\* createBookNode():- Function to create a new book node
2. struct BookNode\* insertBook():- Function to insert a book into the BST
3. struct BookNode\* searchBook():- Function to search for a book in the BST
4. void printBookDetails():- Function to print the details of a book
5. void displayBooks():- Function to display all books in inorder traversal

**Algorithm:-**

To implement an Online Bookstore using a Binary Search Tree (BST), you would follow these steps:

1. Define the Book Structure: Each node of the BST represents a book and contains information such as title, author, ISBN, price, etc.
2. Define the Binary Search Tree Node: Each node in the BST will contain a book and pointers to its left and right child nodes.
3. Insertion Operation: When inserting a new book, follow the rules of a binary search tree:

* If the tree is empty, create a new node with the book information.
* Compare the key (e.g., book title) of the new book with the key of the current node.
* If the key is less than the current node's key, move to the left subtree. If it's greater, move to the right subtree.
* Repeat until finding an appropriate empty spot to insert the new node.

1. Search Operation: To search for a book:

* Start from the root node.
* Compare the search key with the key of the current node.
* If the keys match, return the book.
* If the search key is less than the current node's key, move to the left subtree; if it's greater, move to the right subtree.
* Repeat until finding the book or reaching a leaf node (indicating the book is not in the tree).

1. Deletion Operation: Deleting a node from the BST involves three cases:

* Deleting a leaf node: Simply remove the node from the tree.
* Deleting a node with one child: Replace the node with its child.
* Deleting a node with two children: Find the minimum value node in the right subtree (or maximum in the left subtree), replace the node to be deleted with this minimum value node, and recursively delete the minimum value node.

1. Update Operation: To update the information of a book:

* Search for the book using its key.
* Once found, update the information of the book.

1. Traversal: You may implement various tree traversal algorithms like inorder, preorder, and postorder traversal to iterate over all the books in the tree.
2. Additional Operations: You may implement additional operations like finding the minimum or maximum book, finding successor/predecessor of a book, etc.

**Code:-**

// To Create Online Bookstore using Binary Search Tree.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

// Structure for a book node

struct BookNode {

char title[100];

char author[100];

int isbn;

struct BookNode \*left;

struct BookNode \*right;

};

// Function to create a new book node

struct BookNode\* createBookNode(char title[], char author[], int isbn) {

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

strcpy(newNode->title, title);

strcpy(newNode->author, author);

newNode->isbn = isbn;

newNode->left = newNode->right = NULL;

return newNode;

}

// Function to insert a book into the BST

struct BookNode\* insertBook(struct BookNode\* root, char title[], char author[], int isbn) {

// If the tree is empty, return a new node

if (root == NULL)

return createBookNode(title, author, isbn);

// Otherwise, recur down the tree

if (strcmp(title, root->title) < 0)

root->left = insertBook(root->left, title, author, isbn);

else if (strcmp(title, root->title) > 0)

root->right = insertBook(root->right, title, author, isbn);

// Return the (unchanged) node pointer

return root;

}

// Function to search for a book in the BST

struct BookNode\* searchBook(struct BookNode\* root, char title[]) {

// Base Cases: root is null or title is present at root

if (root == NULL || strcmp(root->title, title) == 0)

return root;

// Title is greater than root's title

if (strcmp(root->title, title) < 0)

return searchBook(root->right, title);

// Title is smaller than root's title

return searchBook(root->left, title);

}

// Function to print the details of a book

void printBookDetails(struct BookNode\* book) {

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

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

printf("ISBN: %d\n", book->isbn);

}

// Function to display all books in inorder traversal

void displayBooks(struct BookNode\* root) {

if (root != NULL) {

displayBooks(root->left);

printBookDetails(root);

displayBooks(root->right);

}

}

// Main function

int main() {

struct BookNode\* root = NULL;

// Inserting books

root = insertBook(root, "The Great Gatsby", "F. Scott Fitzgerald", 123456);

root = insertBook(root, "To Kill a Mockingbird", "Harper Lee", 234567);

root = insertBook(root, "1984", "George Orwell", 345678);

// Displaying all books

printf("All Books:\n");

displayBooks(root);

// Searching for a book

char searchTitle[100];

strcpy(searchTitle, "1984");

struct BookNode\* foundBook = searchBook(root, searchTitle);

if (foundBook != NULL) {

printf("\nBook found:\n");

printBookDetails(foundBook);

} else {

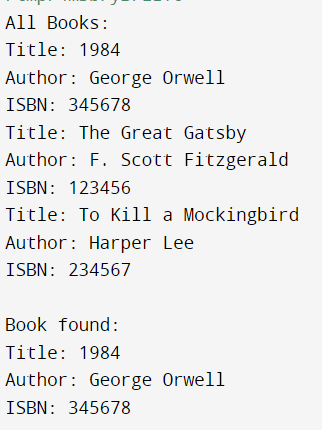
printf("\nBook not found.\n");

}

return 0;

}

**Output:-**



**Conclusion**

**In conclusion, implementing an online bookstore using a binary search tree can offer several benefits such as efficiency, scalability, and flexibility. However, careful consideration of implementation details and regular maintenance are necessary to ensure optimal performance and a positive user experience.**