|  |  |
| --- | --- |
| **Total Marks:** | **07.5** |
| **Obtained Marks:** |  |

**DSA LAB**

**Project Proposal**

**Submitted To: Miss Tehreem Saboor**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Student Name:Muhammad Bin Aftab,Sheheryar Ahmad,Asad Javed**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Reg Number: 2080122,2080131,2180107**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Library Management System Project**

**Introduction:**

This project is a Data Structures Application. The project considers Library Management Project: Where the Data Records are managed by the library management. The library management and the records for adding any book and modifying their data and records. The project finds the Add a book, Delete a book, Search any book, list all of the books, Delete a record, Display a record and store its record temporarily. Note that this BST implementation just serves as a starting point. You may modify it (like adding other records of books like add record and delete record which means that we can modify it at any time) or extend it (for instance, writing the book records to a text file for later retrieval)

**Problem Statement**

This project is proposed to resolve a problem, which the library staff face during their day to day interaction with the students. In addition, it is helpful to the institute administration.

**Proposed System**

This project is a sub category of student management system. This system provides a platform to ask queries and to save his personal data records of the student.

**Benefits of Proposed System**

The project is useful for educational institutes.

**Time Complexity**

This project is not very complex so it is complies in about to 252 line of codes so it has low time complexity.

**Storage Complexity**

As the data is stored temporarily so there isn’t any reason to use file handling only data will be stored in RAM at the time of execution.

**Features of the Project**

* The common features of the projects are:
* This is very easy to use for each user.
* Increase efficiencies and reduce Costs
* Transform IT for higher education
* Easy solution
* Easy process
* Secure all data
* Easy account maintenance
* History
* Easy maintenance process
* The user of the database can see all information and also can edit if necessary.
* Easy implemented routine for student and librarians.

**Objectives**

The main objective of the library management system is to manage the details of Libraries, Students, Registration, Profiles.

Fees. It manages all the information about student.

* To reflect and conserver basic values.
* TO carry out educational futures.
* To manage social change.
* To profit by experience.
* To carry out modernization
* To propagate science.
* To adopt technology.
* To realize National integration.

**Benefits**

* Increases Productivity.
* Best Collaboration
* Saves Natural resources
* Access from Any where
* Increase in Student Enrollment Ratio
* Transparency Increases
* Reduction in the Coast of Communication
* Reduces Workload.

**Requirements**

* Track Student performance and progress
* Keep a check
* Availability of the schedule at regular intervals.
* New room facility feature to boost libraries recognition.

**Conclusion:**

This is a program that implements a binary search tree data structure to store books. It includes functions to insert a book into the tree, delete a book from the tree, search for a book in the tree, list all books in the tree in order by their ISBN, and list all books by a given author.

The structure Book represents a book, with fields for the title, author, and ISBN of the book. The structure BinarySearchTree represents a node in the binary search tree, with fields for the Book structure, a pointer to the left child of the node, and a pointer to the right child of the node.

The root variable is a global pointer to the root node of the binary search tree.

The **CreateNewNode** function creates a new node for a given book and returns a pointer to the new node.

The **InsertBook** function inserts a given book into the binary search tree. It takes as arguments the book to be inserted and a pointer to the root of the subtree in which to insert the book. If the subtree is empty, it creates a new node for the book and returns a pointer to the new node. If the ISBN of the book to be inserted is less than the ISBN of the root of the subtree, it inserts the book into the left subtree of the root. If the ISBN of the book to be inserted is greater than the ISBN of the root of the subtree, it inserts the book into the right subtree of the root.

The **DeleteBook** function deletes a book with a given ISBN from the binary search tree. It takes as arguments the ISBN of the book to be deleted and a pointer to the root of the subtree in which to delete the book. If the subtree is empty, it returns a pointer to the root. If the ISBN of the book to be deleted is less than the ISBN of the root of the subtree, it deletes the book from the left subtree of the root. If the ISBN of the book to be deleted is greater than the ISBN of the root of the subtree, it deletes the book from the right subtree of the root. If the root of the subtree is the book to be deleted, it handles the deletion in one of three ways:

If the root has no children, it simply deletes the root.

If the root has one child, it deletes the root and replaces it with its child.

If the root has two children, it finds the successor of the root (the leftmost node in the right subtree of the root), copies the successor's book to the root, and then deletes the successor.

The **SearchBook** function searches for a book with a given ISBN in the binary search tree. It takes as arguments the ISBN of the book to be searched for and a pointer to the root of the subtree in which to search. If the subtree is empty, it returns false. If the ISBN of the book to be searched for is less than the ISBN of the root of the subtree, it searches the left subtree of the root. If the ISBN of the book to be searched for is greater than the ISBN of the root of the subtree, it searches the right subtree of the root. If the root of the subtree is the book being searched for, it returns true.

The **ListAllBooks** function lists all the books in the binary search tree in order by their ISBN. It takes as argument a pointer to the root of the subtree to list the books from.