```
#include <iostream>
#include <string>
using namespace std;
class BookManagement{
   private:
   struct Node{
      int id;
      string title,author,category;
      Node*next;
   };
   public:
   Node*head=NULL;
   void insertBook();
   void menu();
   void updateBook();
   void searchBook();
   void deleteBook();
   void showBooks();
   void searchByCategory();

};
void BookManagement::menu(){
   int choice;
   while (true){
      cout<<"\n\t_Book Management System_";
      cout<<"\n\nS.No     Functions                  Description"<<endl;
      cout<<"\n1\tAdd Book\t\t\tInsert New Book";
      cout<<"\n2\tSearch Book\t\t\tSearch Book by ID";
      cout<<"\n3\tUpdate Book\t\t\tUpdate Book Record";
      cout<<"\n4\tDelete Book\t\t\tDelete Book by ID";
      cout<<"\n5\tShow Books\t\t\tShow All Books";
```

```
cout<<"\n6\tSearch by Category\t\t\tSearch Books by Category";
cout<<"\n7\tExit"<<endl;


cout<<"Enter Your Choice:";
cin>>choice;
switch(choice){
   case 1:
   insertBook();
   break;
   case 2:
   searchBook();
   break;
   case 3:
   updateBook();
   break;
   case 4:
   deleteBook();
   break;
   case 5:
   showBooks();
   break;
   case 6:
   searchByCategory();
   break;
   case7:
   return;
   default:
   cout<<"Invalid Choice!"<<endl;



}
```

```
    }
}

void BookManagement::insertBook(){
    Node*newBook=new Node;
    cout<<"\nEnter Book ID:";
    cin>>newBook->id;

    cout<<"Enter Book Title:";
    cin.ignore();
    getline(cin,newBook->title);

    cout<<"Enter Book Author:";
    getline(cin,newBook->author);

    newBook->next = NULL;

    if(head==NULL){
        head=newBook;
    }else{
        Node*ptr=head;
        while(ptr->next!=NULL){
            ptr=ptr->next;

        }
        ptr->next = newBook;

    }
    cout<<"\nNew Book Inserted Successfully!"<<endl;
}
```

```
void BookManagement::searchBook(){
   int bookId;
   cout<<"\nEnter Book ID:";
   cin>>bookId;

   Node*ptr=head;
   while(ptr!=NULL){
      if(bookId==ptr->id){
         cout<<"\nBook ID:"<<ptr->id<<endl;
         cout<<"Book Title:"<<ptr->title<<endl;
         cout<<"Book Author:"<<ptr->author<<endl;
         return;
      }
      ptr=ptr->next;
   }
   cout<<"\nBook Not Found!"<<endl;
}

void BookManagement::updateBook(){
   int bookId;
   cout<<"\nEnter Book ID:";
   cin>>bookId;

   Node*ptr=head;
   while(ptr!=NULL){
      if(bookId==ptr->id){
         cout<<"\nBook ID:"<<ptr->id<<endl;
         cout<<"Enter Updated Book Title:";
         cin.ignore();
         getline(cin,ptr->title);
```

```
            cout<<"Enter Updated Book Author:";
            getline(cin,ptr->author);
            cout<<"\nBook Record Updated Successfully!"<<endl;
            return;
        }
        ptr=ptr->next;
    }
    cout<<"\nBook Not Found!"<<endl;
}

void BookManagement::deleteBook(){
    int bookId;
    cout<<"\nEnter Book ID:";
    cin>>bookId;

    if(head==NULL){
        cout<<"\nBook List is Empty!"<<endl;
        return;
    }
    if (bookId==head->id){
        Node*ptr=head;
        head=head->next;
        delete ptr;
        cout<<"\nBook Record Deleted Successfully!"<<endl;
        return;
    }
    Node*prev=head;
    Node*curr=head->next;
    while(curr!=NULL){
        if (bookId==curr->id){
            prev->next=curr->next;
            delete curr;
            cout<<"\nBook Record Deleted Successfully!"<<endl;
            return;
        }
        prev=curr;
        curr=curr->next;
    }
    cout<<"\nBook Not Found!"<<endl;
}

void BookManagement::showBooks(){
    Node*ptr=head;
    while(ptr!=NULL){
        cout<<"\nBook ID:"<<ptr->id<<endl;
        cout<<"Book Title:"<<ptr->title<<endl;
        cout<<"Book Author:"<<ptr->author<<endl;
        ptr=ptr->next;
    }
}

void BookManagement::searchByCategory(){
    string category;
    cout<<"\nEnter Book Category:";
    cin.ignore();
    getline(cin,category);

    Node*ptr=head;
    bool found=false;
    while(ptr!=NULL){
        if(category==ptr->category){
            if(!found){
                cout<<"\nBooks in Category:"<<category<<endl;
                found=true;
            }
            cout<<"\nBook ID:"<<ptr->id<<endl;
            cout<<"Book Title:"<<ptr->title<<endl;
            cout<<"Book Author:"<<ptr->author<<endl;
        }
        ptr=ptr->next;
    }
    if(!found){
        cout<<"\nNo books found in the category:"<<category<<endl;

    }
}

int main(){
    BookManagement bookManagement;
    bookManagement.menu();
    return 0;
}
```