|  |
| --- |
| CSCI760 Project Fall 2015 |
| NYC LIBRARY |
| Library Management System |
|  |
| **Jay Lahoti** |
| **1030858** |

|  |
| --- |
|  |

INDEX

1. Introduction

1.1 Application Description

1.2 Database system Description

2. ER Data design Model

2.1 Entities, Attributes , Domain and their property

2.2 Relations and their Attributes

2.3 ER Diagram

3. Logical Design of Database

4. Relational Database Design

4.1. Perform normalization

5. Implementation of Database and SQL Query

5.1 Implementation of Database NYC Library has a database named elib.

5.2 SQL statements that query the database

6. Screens-shots

1. introduction

1.1. Application Description

The city library has different branches. Each branch has a number of copies of a particular book. The library stores information about books and authors of these books. Readers can borrow or read books from any of these branches.

Readers information is also stored by application. If reader want to borrow a book , he/she has to be register in the system. Readers can reserve upto 10 books from catalog , but book has to be picked up before 6 pm , otherwise, it will be cancelled. Borrower can keep a copy of book till 20 days , then have to pay 20 cent per day. and then have to return from the same branch from where it is borrowed.

1.2. Database system Description

NYC Library allow student to login by his/her card number. If card Number is not in a database, then it will ask reader to login again or register. Reader can register to system online. Each reader will provide a unique reader ID for future use. Reader can go to reader's functions. Reader Functions Menu will open. Where reader can search a book by ID, title, or publisher name. He can borrow a book and can do book checkout. Reader can Book reserve. NYC Library system will compute fine for a book copy borrowed by a reader based on the current date. NYC Library print the list of book reserved by a reader and their status and print the book id and titles of books published by a publisher. At last, reader can Quit the system by log out.

NYC Library can allow admin to login and redirect him to access administrative Functions Menu, in which admin can add a book copy, Search book copy and check its status. He can add new reader. He can Print branch information (name and location). Print top 10 most frequent borrowers in a branch and the number of books each has borrowed. Print top 10 most borrowed books in a branch. Find the average fine paid per reader.

1. ER Data design Model

2.1. Entities , Attributes , Domain and their property

According to the database of the system, NYC Library has a elib database.

1. Book

Book has 7 attributes.

ISBN which is a INT. It is single valued strong entity type.

TITLE is a VARCHAR, multi valued attribute.

Publisher Name is a VARCHAR , single valued.

Publisher Date is a VARCHAR, can be multi valued for different versions.

Publisher address is a VARCHAR, can be multivalued.

Branch ID is a INT and it is a foreign key to refer branch table.

Author ID is a INT and it is a foreign key to refer author table.

Book ID is a PRIMARY KEY and it is a INT and auto increment.

No of Copy is a INT.

1. Branch

Branch ID is a PRIMARY KEY and it is an INT, Auto Increment.

Location is a VARCHAR , multi valued.

Branch Name is a VARCHAR , single valued.

1. Reader

Reader ID is a PRIMARY KEY , INT and auto increment.

Reader Name is a VARCHAR.

Address is a VARCHAR and multi valued.

Phone number is a INT , single valued.

FINE is a INT , single valued.

1. Author

Author ID is INT , Primary key and auto increment.

Author Name is VARCHAR , single valued.

1. Admin

ID ,Password

2.2. Relations and their Attributes

1. reader can issue/reserve book

List of Attributes

* Reader ID
* Book ID
* return date
* issue date
* due date

Cardinality of relation is 1 to 10 as reader can issue upto 10 books.

reader[0,1]

book[0,10]

1. author writes book

List of Attributes

* Author ID
* Book ID

cardinality of relation is 1 to many as author can write many books.

author[1,1]

book[1,n]

1. branch has books

List of Attributes

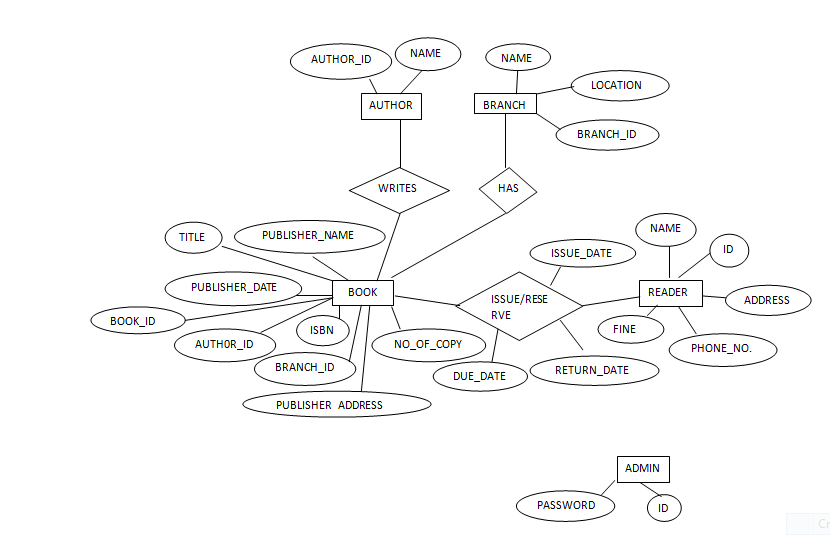
* branch ID
* book ID
* no of copy

Cardinality of relation is many to many as branch has multiple copies and copies can be hold at different branches.

branch[1,n]

book[0,n]

2.3. ER Diagram



1. Logical Design of Database
2. book(auth0r\_id,book\_id,branch\_id,isbn,no\_of\_copy,publisher\_address,publisher\_date,publisher\_name,title)

primary key: book\_id

foreign keys: auth0r\_id, branch\_id

1. branch(branch\_id, location, name)

primary key: branch\_id

1. reader(contact\_no, fine, reader\_address , reader\_id , reader\_name)

primary key : reader\_id

1. author(author\_id,name)

primary key : author\_id

1. issue(book\_id, due\_date, issue-date, issue\_id,reader\_id,return\_date)

primary key : reader\_id, book\_id

1. writes(author\_id, book\_id)

primary key : author\_id, book\_id

1. has(branch\_id,book\_id)

primary key : branch\_id, book\_id

1. Relational Database Design

4.1. Perform normalization

Relation is not in a normal form. There are redundacy of writes(author\_id, book\_id) and has(branch\_id,book\_id). we can remove those two relations from the database.

1. book(auth0r\_id,book\_id,branch\_id,isbn,no\_of\_copy,publisher\_address,publisher\_date,publisher\_name,title)
2. branch(branch\_id, location, name)
3. reader(contact\_no, fine, reader\_address , reader\_id , reader\_name)
4. author(author\_id,name)
5. issue(book\_id, due\_date, issue-date, issue\_id,reader\_id,return\_date)
6. Implementation of Database and SQL Query

5.1. Implementation of Database

NYC Library has a database named elib.

1. CREATE DATABASE *elib;*
2. CREATE TABLE book  
   (  
   book\_id int(10) ,  
   title varchar(30),  
   publisher\_name varchar(30),

publisher\_date varchar(10),

publisher\_address varchar(30),

no\_of\_copy int(10),

isbn varchar(30),  
auth0r\_id int(10),

branch\_id int(30),

CONSTRAINT book\_id PRIMARY KEY (book\_id)

CONSTRAINT FOREIGN KEY (branch\_id)  
REFERENCES branch(branch\_Id),

CONSTRAINT FOREIGN KEY (author\_id)  
REFERENCES author(author\_id)  
);

1. CREATE TABLE branch  
   (  
   branch\_id int(10) ,  
   location varchar(30),  
   name varchar(30),

CONSTRAINT branch\_id PRIMARY KEY (branch\_id)  
);

1. CREATE TABLE issue  
   (  
   book\_id int(10) ,  
   issue\_date date,

due\_date date,

return\_date date,

book\_id int(10),

reader\_id int(10),

CONSTRAINT issue\_id PRIMARY KEY (issue\_id),

CONSTRAINT FOREIGN KEY (reader\_id,book\_id)  
REFERENCES reader(reader\_id),book(book\_id)  
);

1. CREATE TABLE reader  
   (  
   reader\_id int(10),

reader\_name varchar(30),

reader\_address varchar(30),

contact\_no varchar(30),

fine int(100),  
CONSTRAINT reader\_id PRIMARY KEY (reader\_id),

);

1. insert into book

(

title,

publisher\_name,

publisher\_date,

publisher\_address,

no\_of\_copy,

isbn,

auth0r\_id,

branch\_id

)

values

(

'$title',

'$publisher\_name',

'$publisher\_date',

'$publisher\_address',

'$copy',

'$isbn',

'$auther\_id',

'$branch\_id'

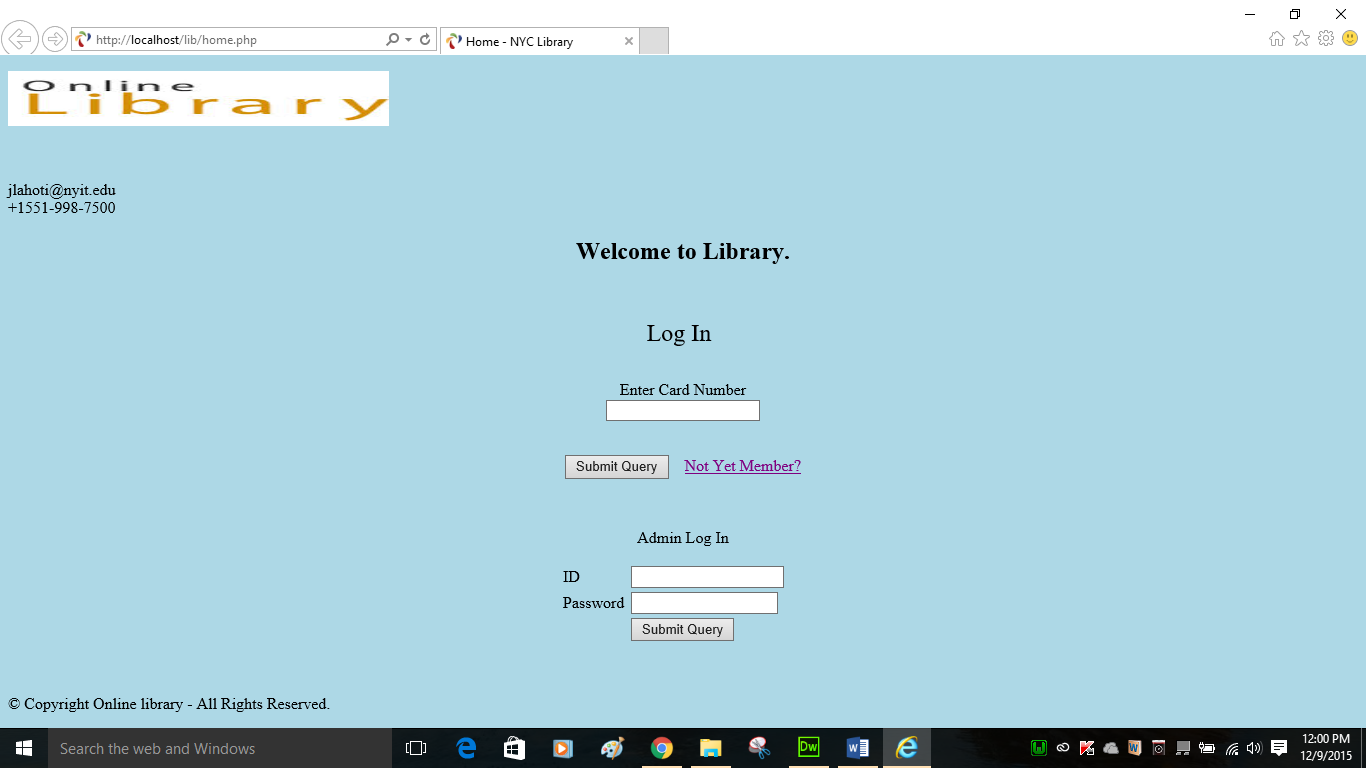
);

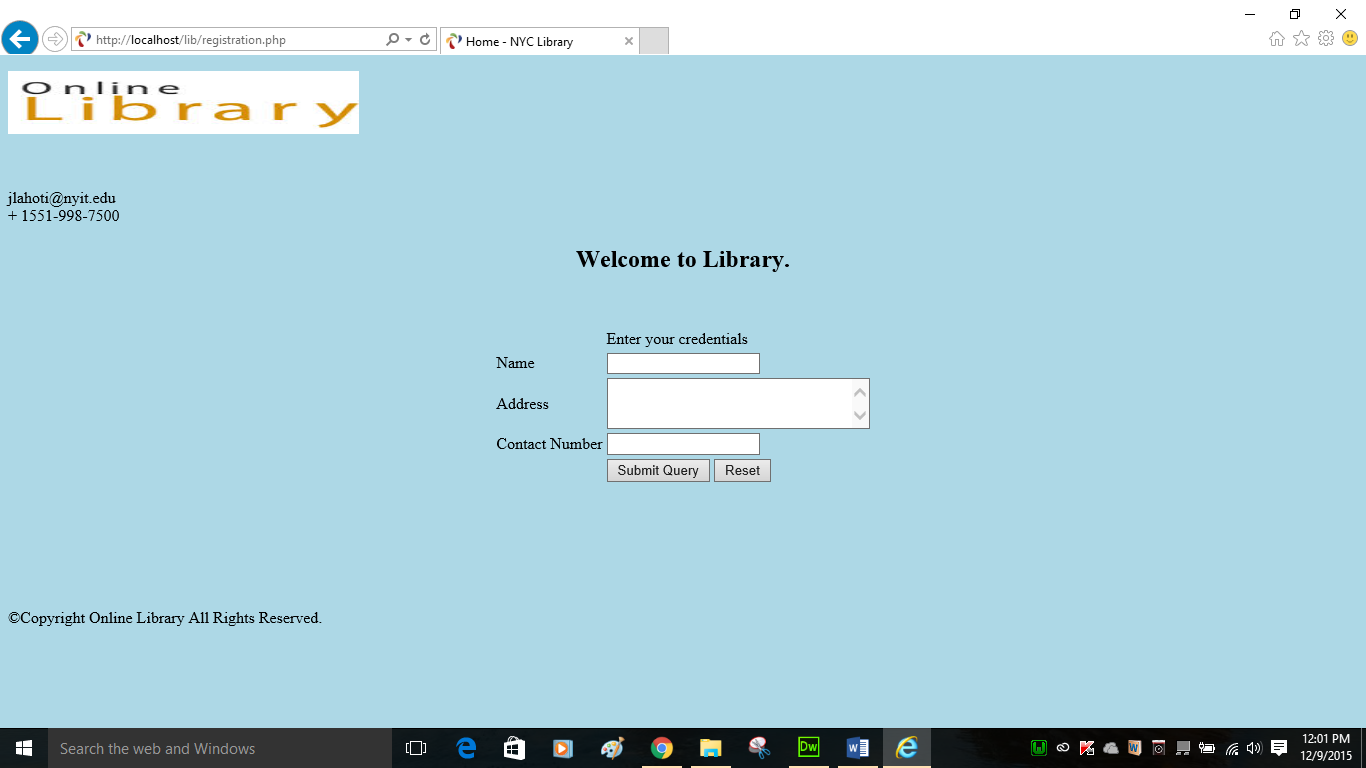
1. "insert into reader(reader\_name , reader\_address, contact\_no) values('$rname', '$address', '$contactno')";
2. "insert into issue(issue\_date, due\_date, return\_date, book\_id, reader\_id) values('$issue\_date', '$due\_date1', 'null', '$book\_id', '$reader\_id')";

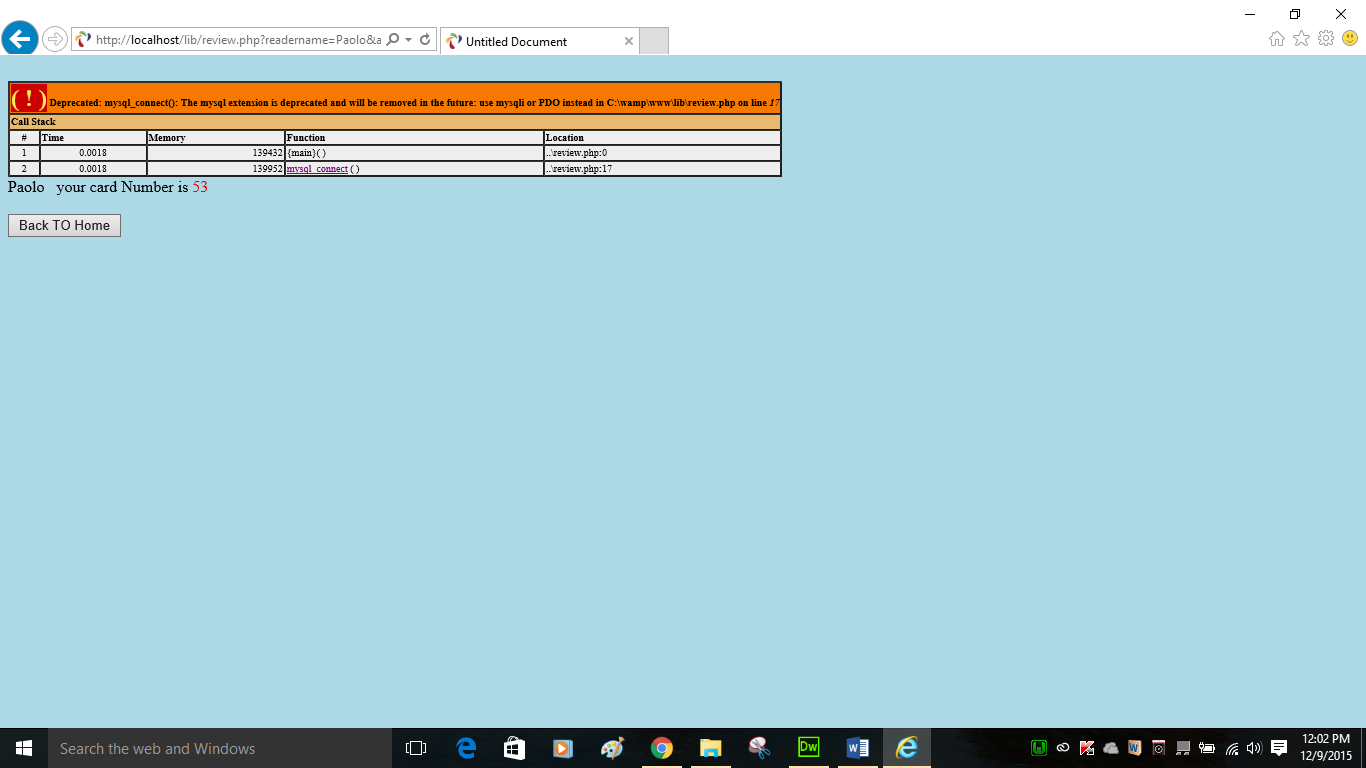
5.2. SQL statements that query the database

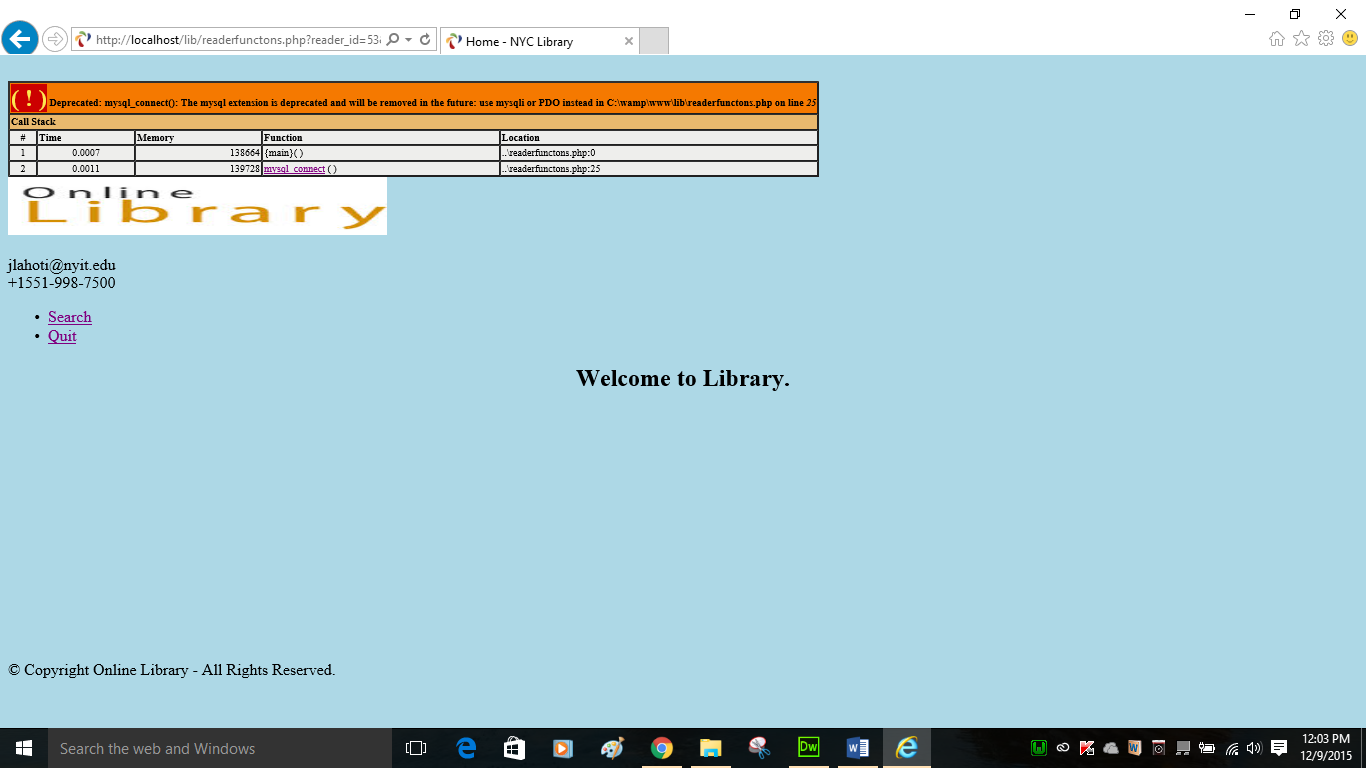
1. "UPDATE book SET no\_of\_copy=(no\_of\_copy-1) where book\_id='".$book\_id."'";
2. "select issue.\*,book.\* from issue,book where issue.book\_id = book.book\_id";
3. "update issue set return\_date='".$return\_date."' where book\_id='".$book\_id."'";
4. "UPDATE book SET no\_of\_copy=(no\_of\_copy+1) where book\_id='".$book\_id."'";
5. "select \* from book where book\_id=".$id;
6. "select \* from book where no\_of\_copy >=1 ";
7. "insert into branch(branch\_id,location,name) values('$branch\_id','$location','$name')";

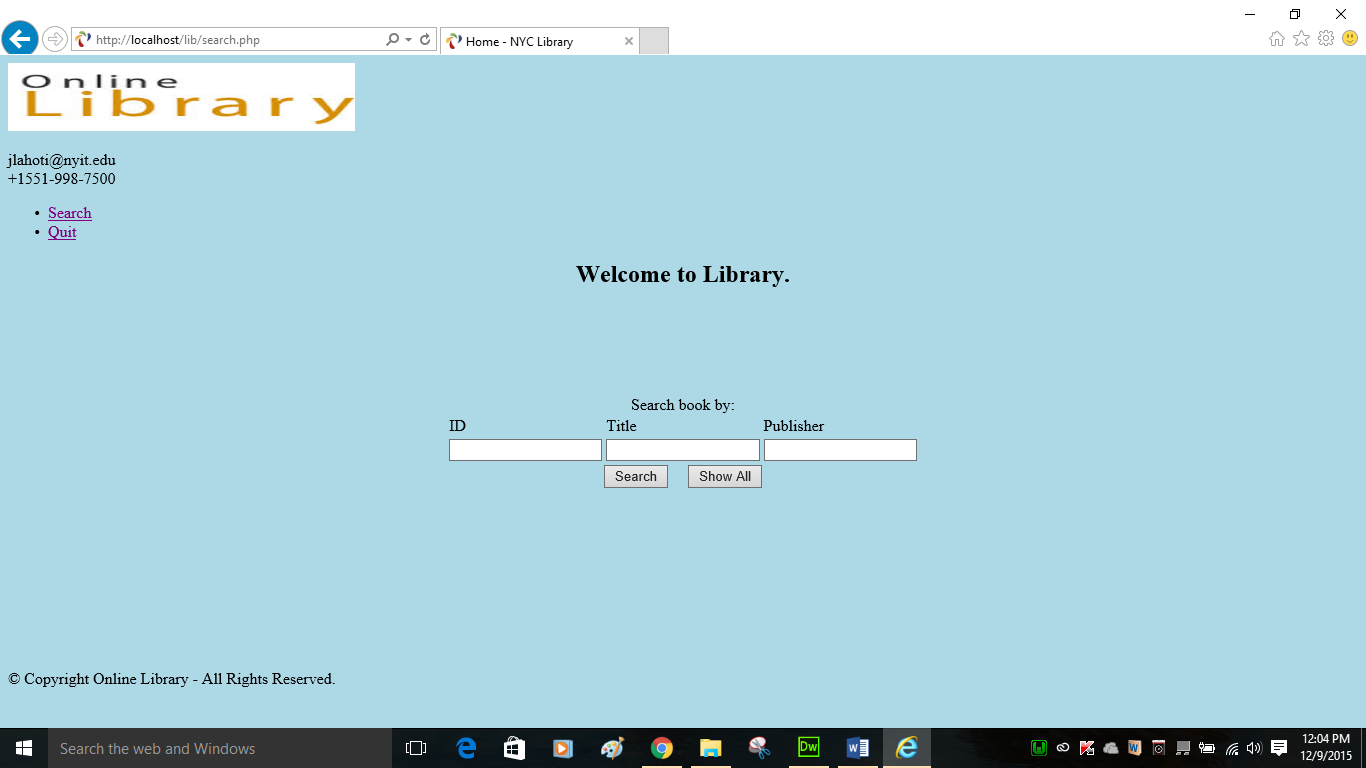
1. Screen-shots



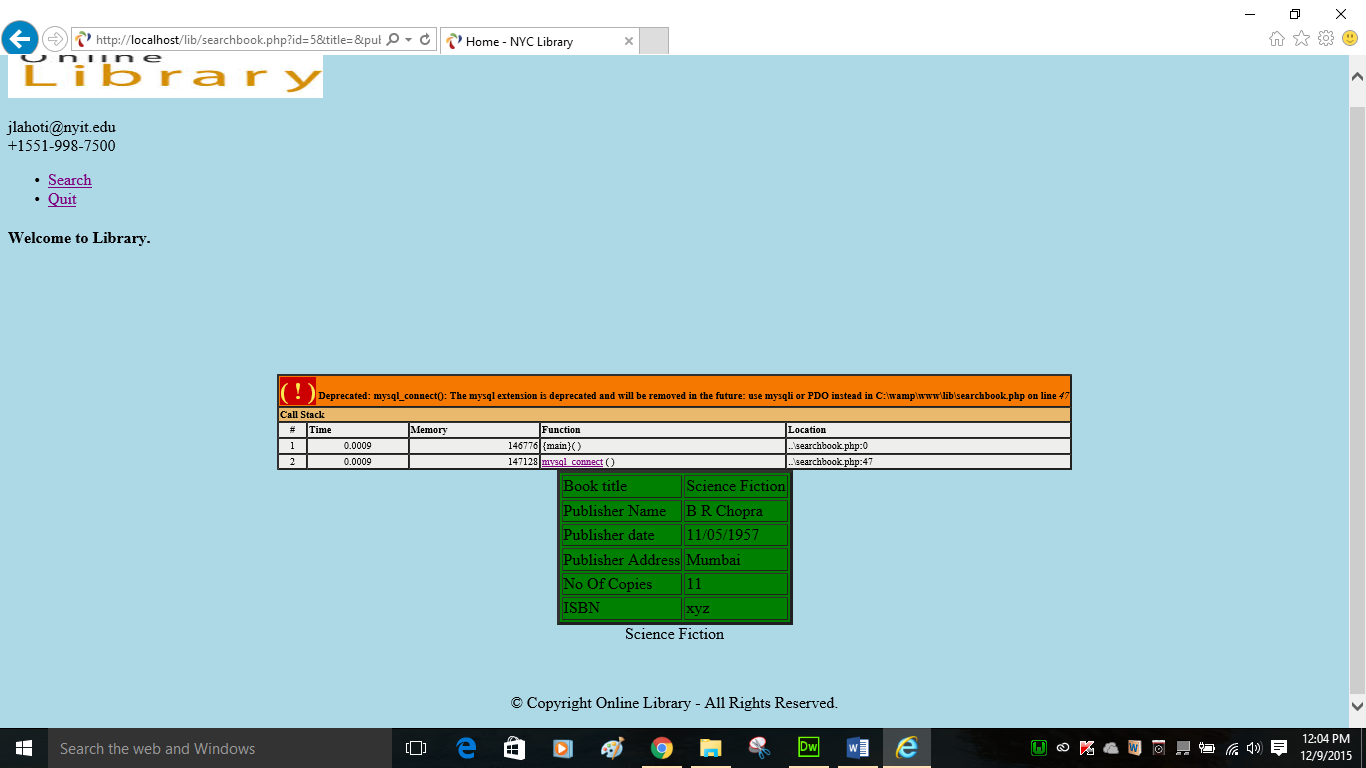








Search Book By ID/Title/Publisher



Show all book search



Admin Page

