Programming in C++ Lab
Project Report

Under the guidance of

Prof. Suseta Dutta

Prof. Archisman Ghosh

Prof. Pallavi Saha

by

Shalini Guha (Roll-03)

Saptarshi De (Roll-02)

6thSemester

CSE-3A

Department of Computer Science & Engineering

University of Engineering & Management, Kolkata

Year - 2018

Acknowledgement

This is to certify that the project report for Programming in C++ lab is being submitted by Shalini Guha(Roll-03) and Saptarshi De (Roll-02) in B.Tech -semester VI is a record bonafide work carried out by them. The results embodied in this report have not been submitted to any other University for the award of any degree.

Prof. Suseta Dutta	Prof. Archisman Ghosh	Prof. Pallavi Saha

Bookshop Management System

Introduction

This Bookshop Management System aims at providing an effective interface for the employees of Bookstores to conveniently maintain and access a database of books and resources and also allows admins to monitor stocks, annual sales and to provide access to other employees and access existing employee's records.

It is a web based application that is capable of accessing

and storing the personal and assessment data of the employees of a company. The application is

created using a three tier architecture, consisting of the

- Presentation
- Business Logic and
- Data Access Layers

The goal of this application was to design a secure, interactive and easy to use application that can provide the details in the directory to people that need it, easily.

Requirement of the exercise:

- A Login page to verify the user that wishes to check the information
- A User Registration page to register a new user
- A directory View page that displays the results of the stored directory
- A Record Updation page that allows a valid user to update the details of an employee
- A Record Insertion page that allows for new records to be entered into the database
- A Record Deletion option that allows for deletion of obsolete records

Software Used:

IDE :

- NetBeans IDE 8.2

Frontend Technologies:

- JavaScript
- Bootstrap
- jQuery
- HTML
- CSS

Server Management/Technology:

- Apache Tomcat 8.5.24
- Java Server Pages

Drivers:

- Google Gson-2.4
- JDBC-api-1.4

Database Server:

- MySQL

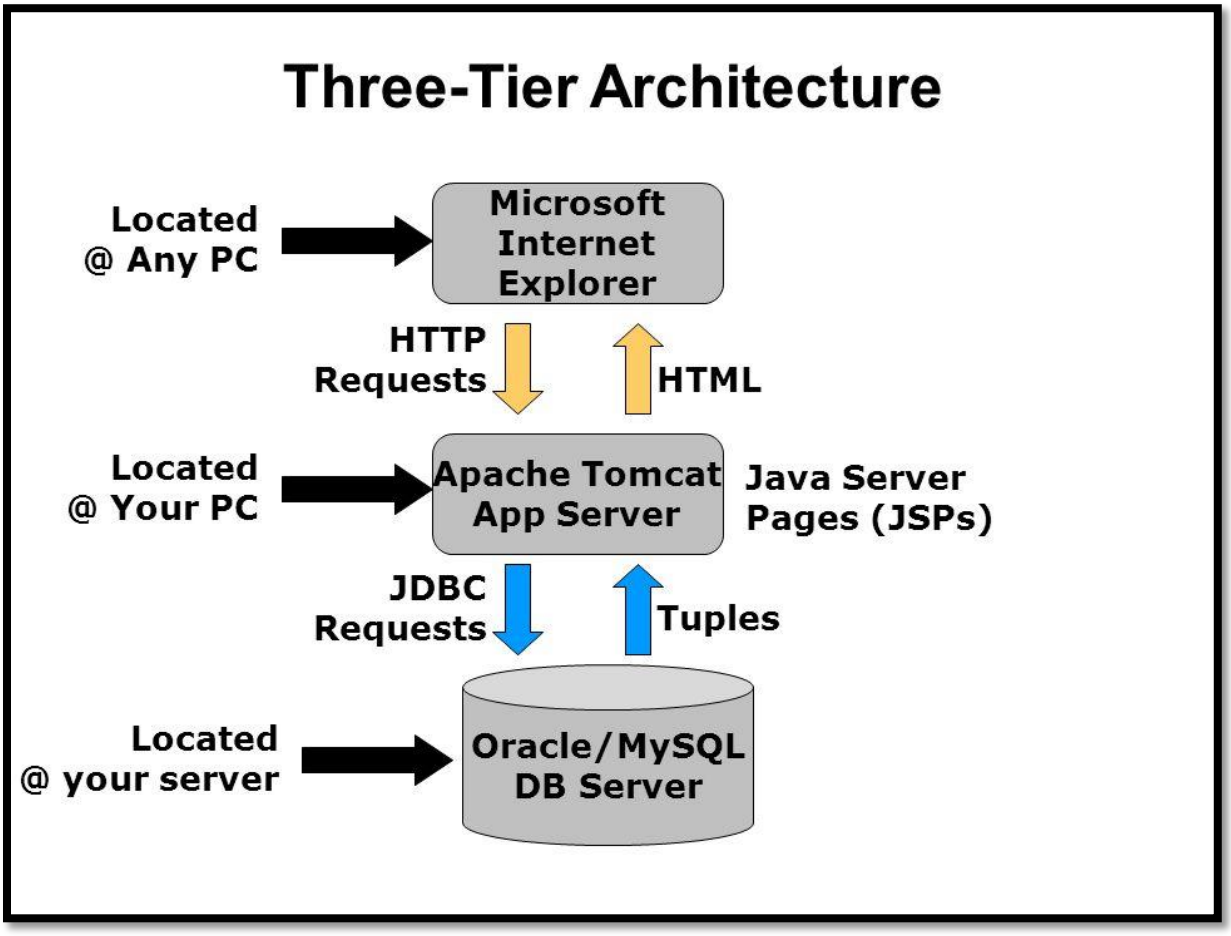
Backend Technology:

- Java

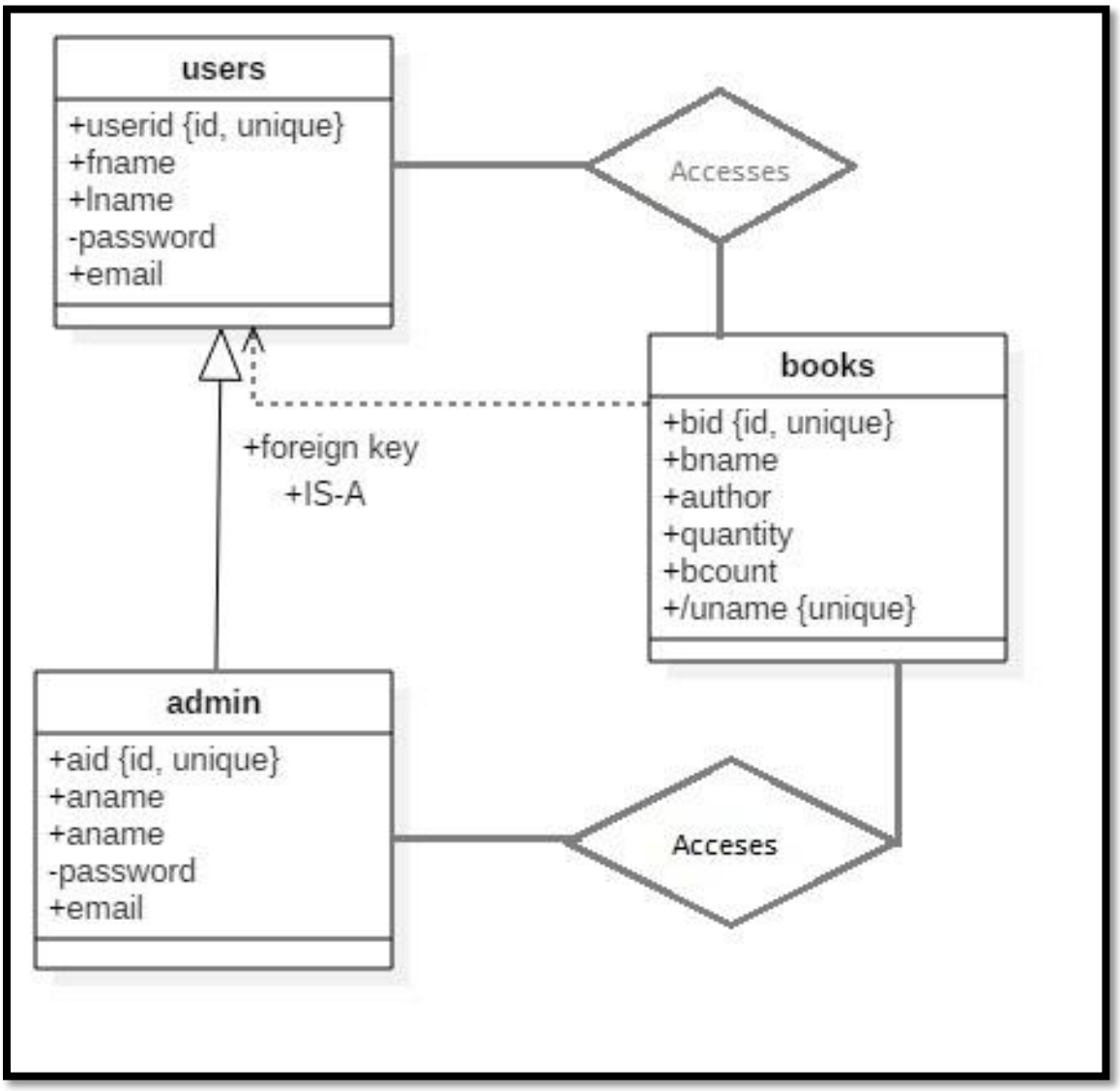
Technical Architecture

The aforementioned Application is a Three Tiered Architecture, I.e. separation of the total application into Presentation, Business Logic/Business Access and Data Access Layers.

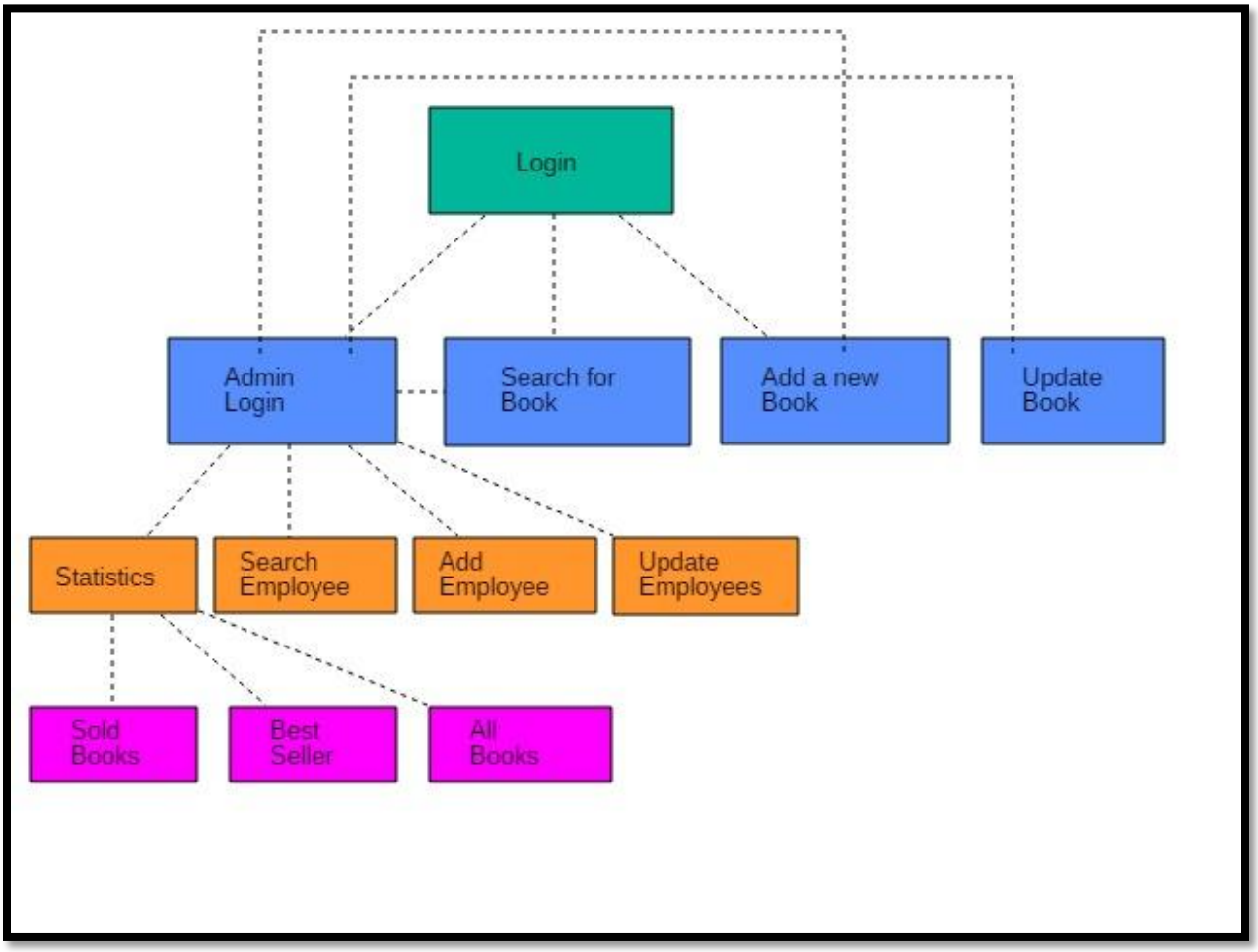
- The Presentation Layer comprises the front end of the webpages that are being viewed by the user.
- The Business Logic Layer contains the server side functions that need to be performed on the given input or choices picked by the user.
- The Data Access Layer has all the SQL queries and the connection to the database on which the related queries must be performed.



ER Diagram

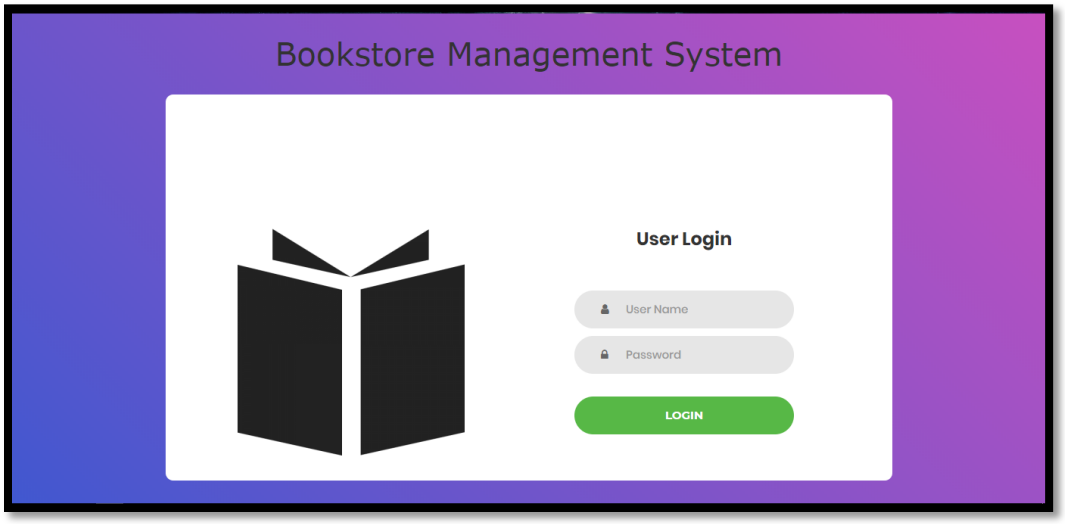

users
+userid {id, unique}
+fname
+lname
-password
+email
Accesses
books
+bid {id, unique}
+bname
+author
+quantity
+bcount
+/uname {unique}
+foreign key
+IS-A
admin
+aid {id, unique}
+aname
+aname
-password
+email
Acceses

Flow Chart

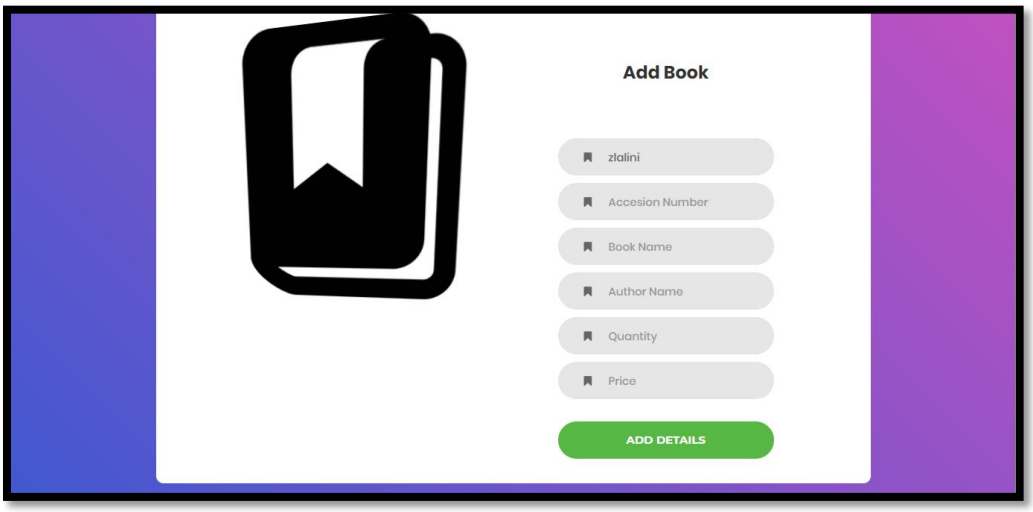

Login
Admin Login
Search for Book
Add a new Book
Update Book
Statistics
Search Employee
Add Employee
Update Employees
Sold Books
Best Seller
All Books

Output

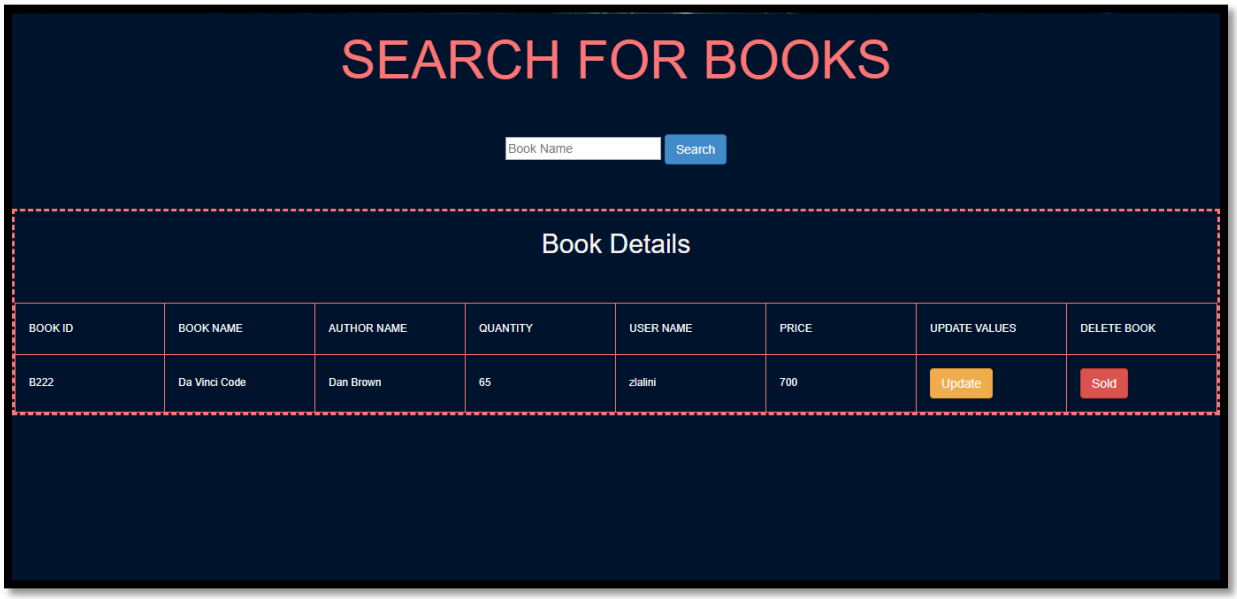
Login Page



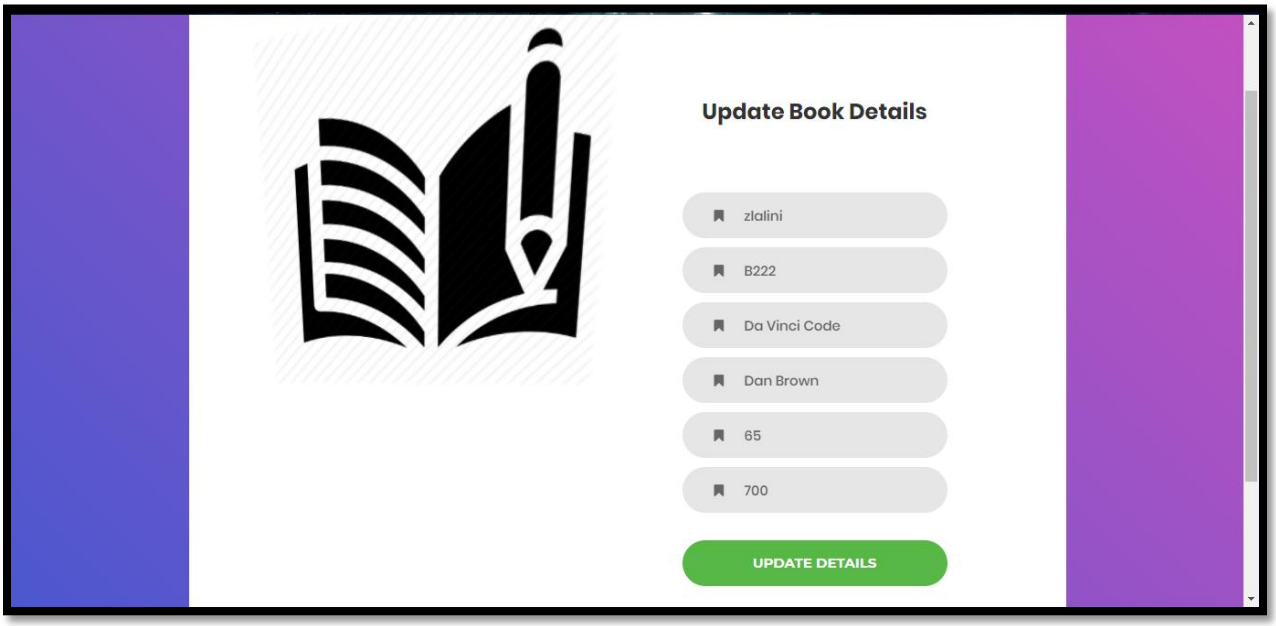
Add a New Book



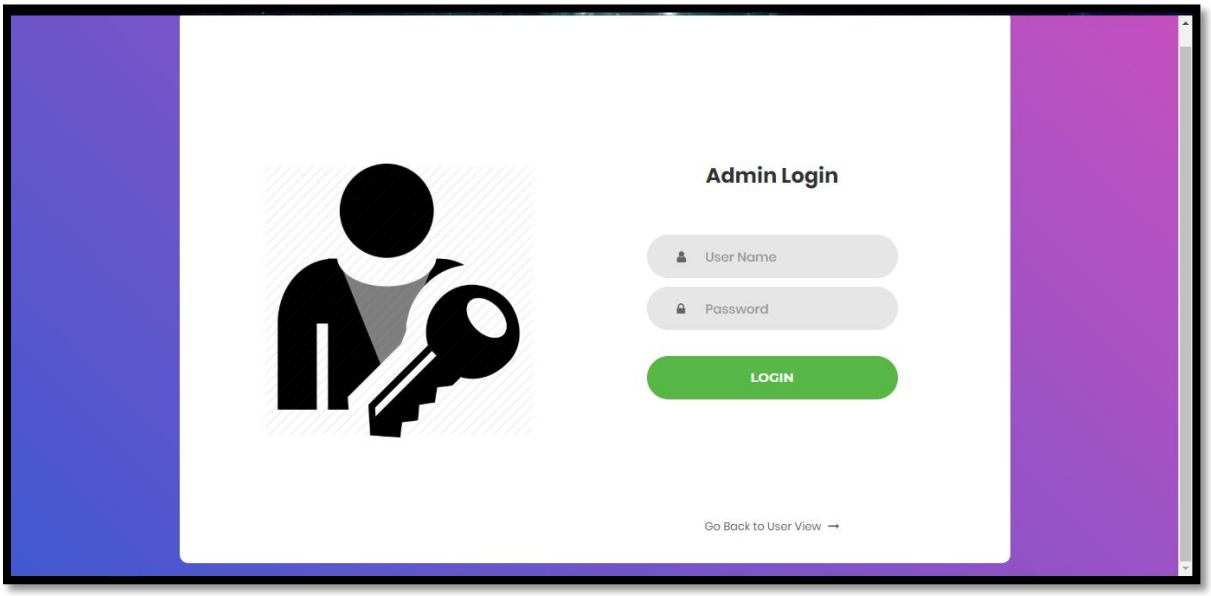
Search for a Book



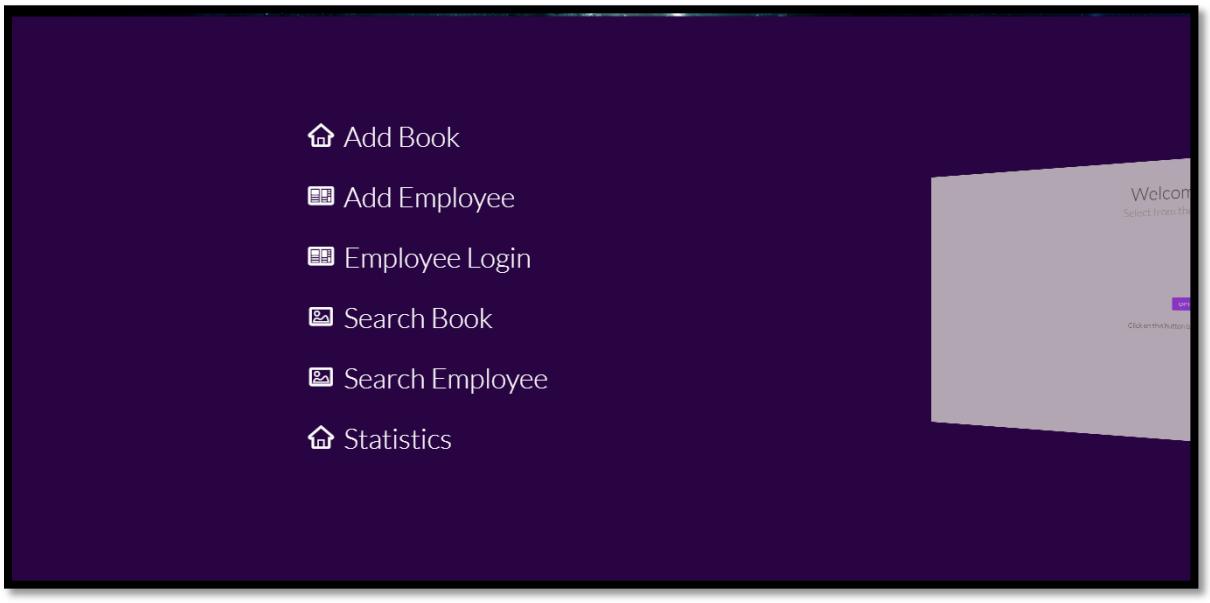
Update a Book



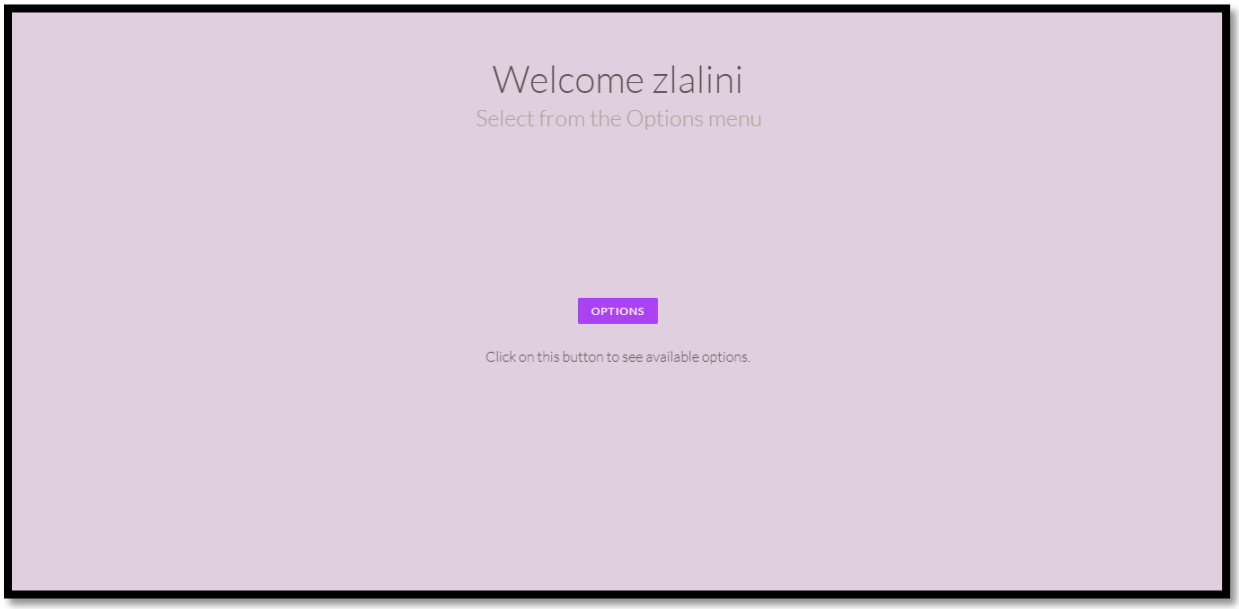
Admin Login



Admin Options

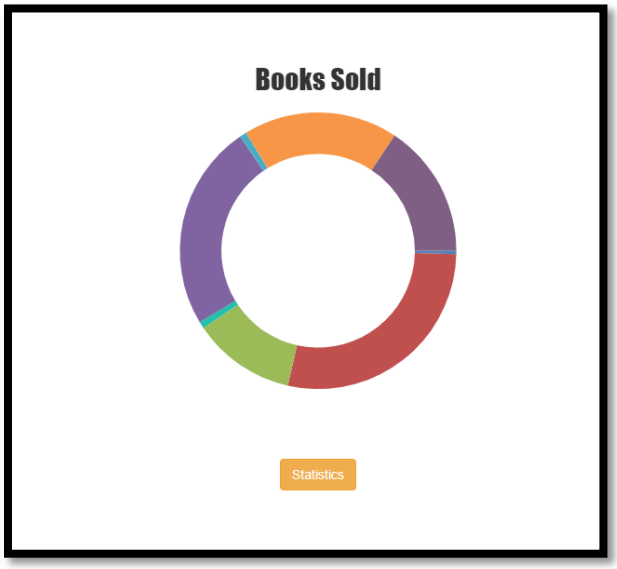


Admin /User Page



User Options

Statistics

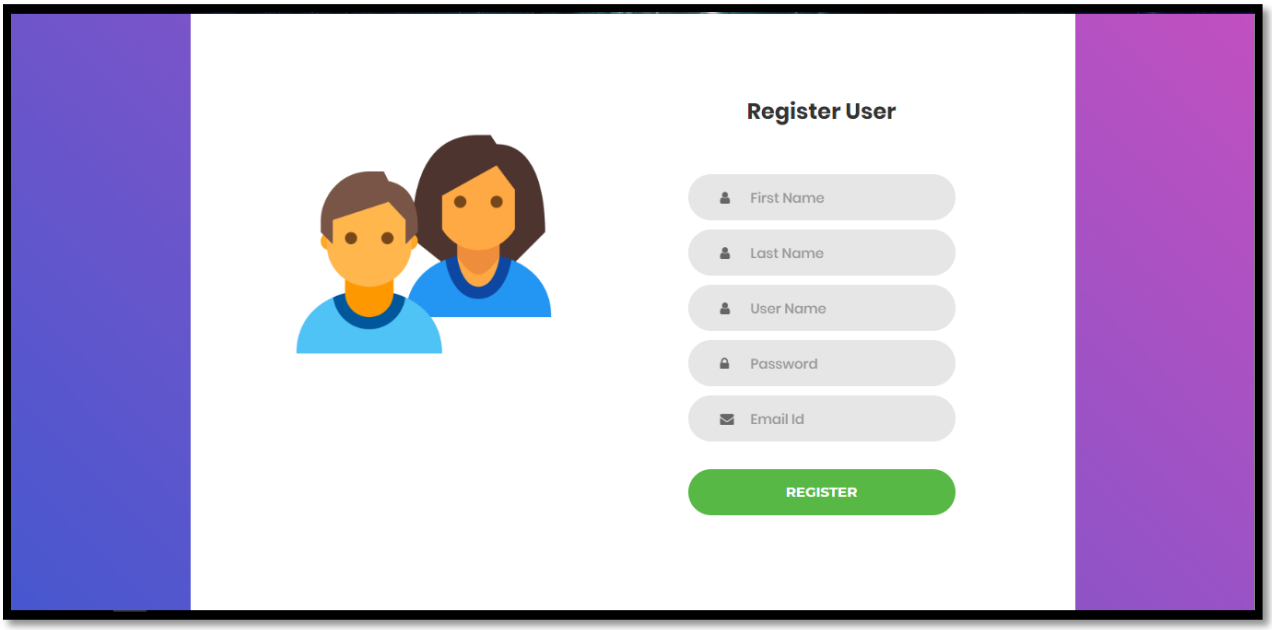


LIST OF BOOKS

Book Details

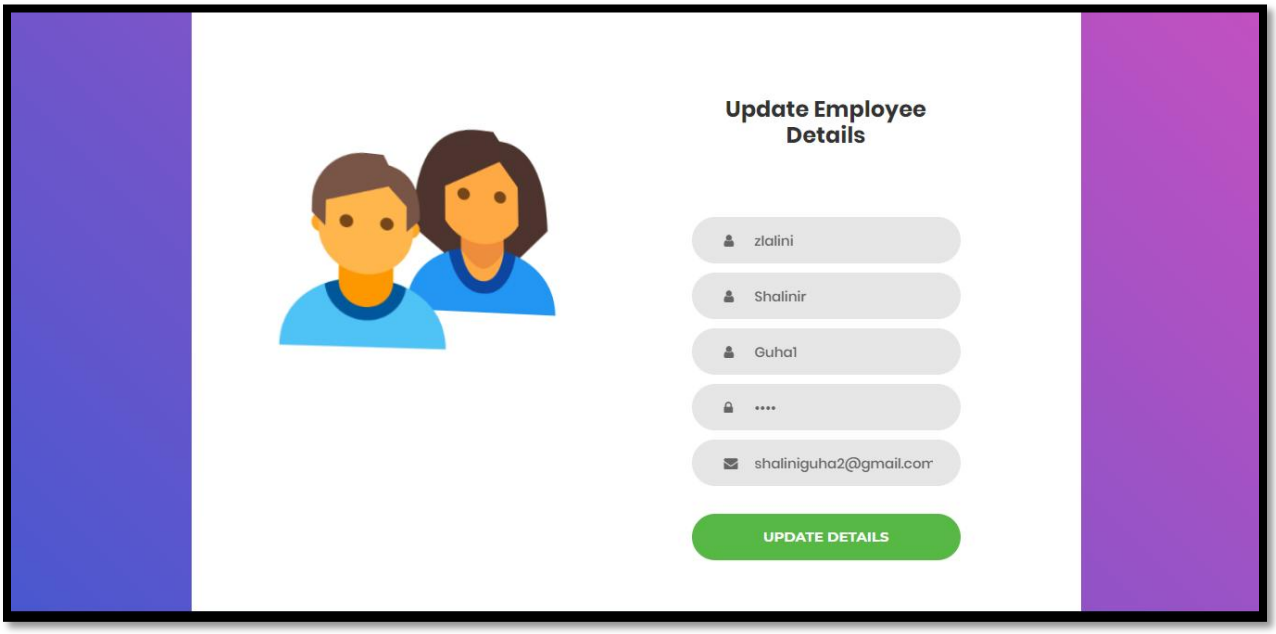
BOOK ID	BOOK NAME	AUTHOR NAME	QUANTITY	USER NAME	PRICE	UPDATE VALUES	DELETE BOOK
B111	Panchatantra	Someone	0	zlalini	1200	Update	Sold
B222	Da Vinci Code	Dan Brown	65	zlalini	700	Update	Sold
B333	Game of thrones	George RR Martin	30	zlalini	600	Update	Sold
B4568	History	SS Kumar	0	zlalini	678	Update	Sold
B555	Harry Potter	JK Rowling	57	zlalini	1500	Update	Sold

Add a New Employee



Search for a Book

Update a Book



Future Scope

- The system could be modified suitably to work on a large network. This involves, among other, resolving used conflicts, protecting database integrity and ensuring consistency of data if it is distributed across multiple locations.
- Maintain attendance details of employee
- Maintain Mobile details.

Conclusion

- An attempt is made in all its earnest towards the successful completion of the project. This system was verified with valid as well as with invalid data.
- This system is user friendly since it has been developed a successful GUI environment. Since the connection can be extended to any database. The control will be more powerful.
- Connecting it to any type of database extends the development control. Any suggestions for future development of the system are welcome.
- Upgrading the system if may can be done without affecting the proper functioning of system.

References

- https://www.roseindia.net/jsp/jsp-login-form-with-mysql-database-connection-and-back-end-validation.shtml
- https://canvasjs.com/
- stackoverflow.com
- www.codeproject.com
- www.webdeveloper.com
- www.w3schools.com

1. Write a program in C++ to search for a word in a sentence using friend function.

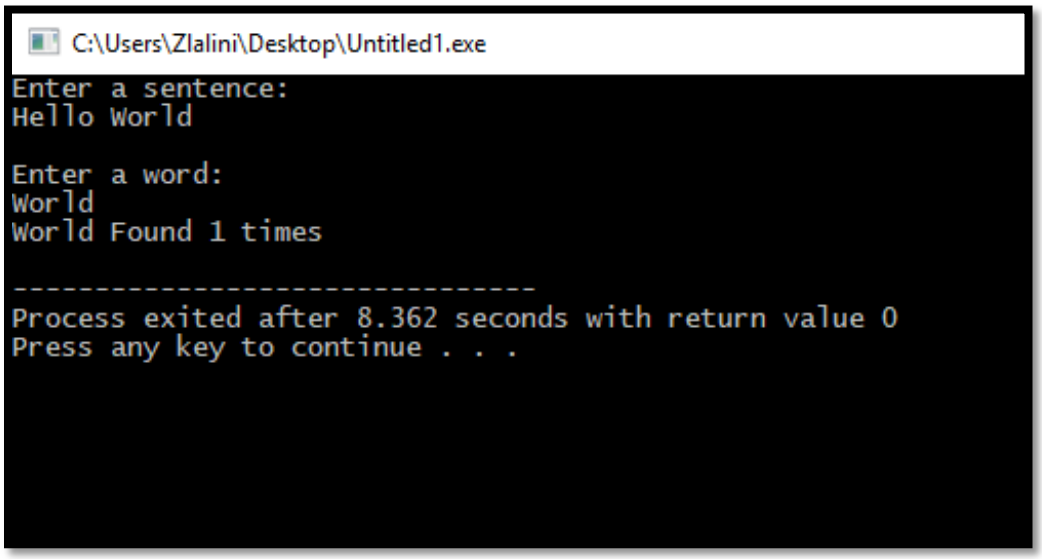
Code:

```
/*  Find a word in a given word in a sentence
Author : Shalini Guha
Saptarshi De
Date : 04.04.2018
Version : 1.0 */

#include<iostream>

#include<string>

using namespace std;

class Search{

     private:

          string word,sentence;

          public:

               Search(string a,string b)   //Parameterized
Constructor to initialise object

               {

                    word=a;

                    sentence=b;

               }

     friend void search(Search); //Friend Function to search
for the word
```

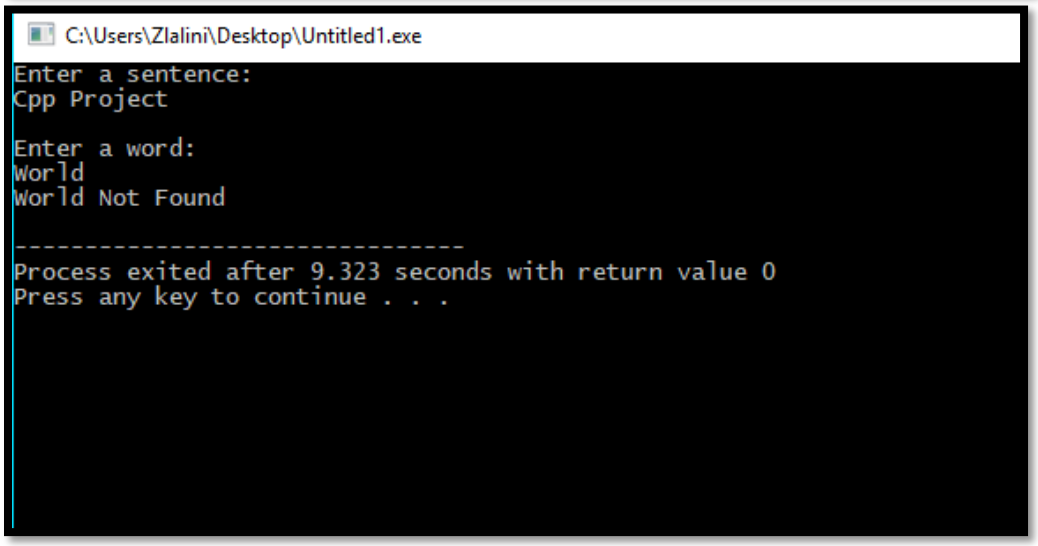
```
};
void search(Search obj1)
{
     int i=0;  //Counter to count the number of occurences of
the word
     int pos;
     string sentence=obj1.sentence;
     string word=obj1.word;
     pos=sentence.find(word,0);  //Variable to iterate
through string
          while(pos!=std::string::npos){
               i++;
               pos=sentence.find(word,pos+1);
          }
          if(i>0)
          cout<<word<<" Found "<<i<<" times"<<endl;
          else
          cout<<word<<" Not Found"<<endl;

}
int main()
{
     string sentence,word;
```

```
    cout<<"Enter a sentence:"<<endl; //Sentence to be
searched
    getline(cin,sentence);
    cout<<"\nEnter a word:"<<endl;        //Word to be found
    getline(cin,word);
    Search obj(word,sentence);
    search(obj);
}
```

Output:


C:\Users\Zlalini\Desktop\Untitled1.exe
Enter a sentence:
Hello World
Enter a word:
World
World Found 1 times

Process exited after 8.362 seconds with return value 0
Press any key to continue . . .


C:\Users\Zlalini\Desktop\Untitled1.exe
Enter a sentence:
Cpp Project
Enter a word:
World
World Not Found

Process exited after 9.323 seconds with return value 0
Press any key to continue . . .