LIBRARY MANAGEMENT SYSTEM WITH SMS AUTOREPLY

Submitted to
The faculty of engineering and technology of
Kakatiya University, Warangal
In partial fulfillment of the requirements
to award
Bachelor of Technology
in
Computer Science & Engineering

BY

K . SAI SREENIVAS REDDY	**[B16CS084]**
E . MANOHAR	**[B16CS062]**
P . SPANDANA	**[B16CS067]**
R . SAI KRISHNA	**[B17CS201L]**

Under the Guidance of

P . VIJAY KUMAR
Assistant Professor, Department of CSE

Department of Computer Science & Engineering,
KAKATIYA INSTITUTE OF TECHNOLOGY & SCIENCE
(An Autonomous Institute under Kakatiya University),
WARANGAL (T.S)
2019-2020

KAKATIYA INSTITUTE OF TECHNOLOGY & SCIENCE

Warangal – 506 015, Telangana, INDIA. *(An AUTONOMOUS INSTITUTE under Kakatiya University, Warangal)*

కాకతీయ సాంకేతిక విజ్ఞాన శాస్త్ర విద్యాలయం, వరంగల్ – ౫౦౬౦౧౫.

Estd-1980

KITSW

CERTIFICATE

This is to certify that **K.SAI SREENIVAS REDDY (B16CS084), E.MANOHAR (B16CS062), P.SPANDANA (B16CS067), R.SAI KRISHNA (B17CS201L)** of the IV/IV B.Tech. Computer Science and Engineering have satisfactorily completed the major project work entitled "**LIBRARY MANAGEMENT SYSTEM WITH SMS AUTOREPLY**" in the partial fulfillment of the requirements of B. Tech degree during the academic year 2019-2020.

Supervisor	**Head of the Department**
P. Vijay Kumar	**Dr. V. Shankar**
Assistant Professor, Dept. of CSE	*Professor, Dept. of CSE*
KITS, Warangal	KITS, Warangal

DECLARATION

We declare that the work presented in this project report is original and has been carried out in the Department of Computer Science and Engineering, Kakatiya Institute of Technology and Science, Warangal, Telangana, and to best of our knowledge it has been not submitted elsewhere for any degree

SAI SREENIVAS REDDY KARRA
Roll No. B16CS084

MANOHAR ERRA
Roll No. B16CS062

SPANDANA PALLE
Roll No. B16CS067

SAI KRISHNA RESU
Roll No. B17CS201L

ACKNOWLEDGEMENT

We extend our sincere and heartfelt thanks to our esteemed guide, **P.VIJAY KUMAR** sir for his exemplary guidance, monitoring and constant encouragement who stood as pillar of strength right through our project, till the preparation of report and helped by boosting our morale.

We are grateful to respected coordinator **S.VENKTRAMULU** sir for permitting us in utilizing all the necessary facilities of the Institute in completing the project.

We would like to extend our thanks to our respected Head of department, **Dr.V.SHANKAR** sir for his whole-hearted support, who has provided all the facilities to conduct our project work with immense cooperation and inspiration.

Our special thanks to each and every faculty members and programmers in the department for their suggestions and taking care of all software needs.

Last but not the least, we would like to thank our friends and family for the constant support and encouragement they have given us during the course of our work.

K. SAI SREENIVAS REDDY	**[B16CS084]**
E. MANOHAR	**[B16CS062]**
P. SPANDANA	**[B16CS067]**
R. SAI KRISHNA	**[B17CS201L]**

ABSTRACT

This Library Management System can be used to maintain records of user details like registering books, checking balance penalty amount etc. This system not only works from admin side but also from user side. User can also check his/her own details and get the notification for balance amount of penalty, books returning date, can also check available books.

This system also includes SMS autoreply function. He/she will get reply via SMS. He will be getting all kinds of information via SMS such as return date warnings booking confirmation etc.

In this application there are two modules namely, Librarian and Student. Librarian can register Student details and provide him/her a login id and password. He/she can add books as well and can view the due dates and penalty amount. Students get login id and password, he/she can view his personal details, check for desire books and get update of due dates, issue date of books and penalty amount to be paid. The student can also book a particular book from anywhere and go to library to collect it directly without any other formalities.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 - INTRODUCTION

1. INTRODUCTION

Library is used for storing books, issuing books and also provides reference books. The existing system or process is time consuming. It is better to take advantage of the developments in technology to support and to improve the experience in library services. To expedite the process of search services and borrowing books from the library in the proposed system, we need a system that allows users of library to access library services online using a web based library application.

This proposed system provides convenience to users in searching of books, viewing their details, viewing the number of books in their account, ordering a book etc. The users are also provided with a facility to view information about book renewal dates and fines. This system also sends warning messages if the returning date for a book is near and also sends confirmation message for orders.

CHAPTER 2 – LITERATURE SURVEY

2. LITERATURE SURVEY

Literature survey is the most important step in software development process. Before developing the tools it is necessary to determine the time factor and economy. Once things are satisfied, then next steps are to determine which operating system and languages can be used for developing the tool or software. Once the programmers start building the software, they need lot of support. This support can be obtained from senior programmers, from books or from websites. Before building a software the above considerations are taken into account for developing the proposed system.

2.1 Purpose of the project

The application introduced in this chapter is an advanced application. This application helps the library staff to manage the activities in library more efficiently. Search process becomes more efficient. Student can order or request for a new book via this application if he/she needs a book. The interesting feature of this application is that student will be informed about return dates and order/request confirmation via SMS. This application is mainly developed to make the activities of library faster.

2.2 Existing system

In existing library system, many transactions are done manually, which consume more time like searching for a book. Student must remember the book renewal dates, fine for a particular book which is sometimes difficult. Student has to come to library, search for a book and if that book is available he can then borrow the book. This takes up a lot of time and manual power.

2.3 Solutions for existing system problems

The development of the new system contains the following activities which try to automate the entire process, keeping in view the existing system.

- User friendly interface should be provided in the application with different options.
- The application must make the overall library management system much easier and flexible.
- Search process must be made faster.
- The prolonged process of borrowing a book from library must b reduced as much as possible.
- The student must be given access to check his/her fines and return dates of different books they possess from anywhere using an internet access.

2.4 Proposed system

Proposed system enables to overcome inabilities of the existing system. It is an automated library management system.

- This application can be used in any smart phone or computer, if they have internet access.
- Students can search for required books using the subject.
- It also has a benefit of viewing the number of books in one’s account.
- This system also sends alert messages for return dates and order confirmation messages via SMS to mobile phones.
- A student can order a book and borrow it from library. This whole process is made easier and simple.
- Students can view information about fine amounts, return dates of various books in their account.

CHAPTER 3 - DESIGN

3. DESIGN

3.1 Software requirements specification

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specification is properly given, it is possible to design a system., which will fill into required environment. This is because the requirements have to be known during the initial stage so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

Hardware Requirements:-

- RAM- 4GB
- ROM- 200GB
- Processor- Intel core i2
- Hard Disk: 10GB

Software Requirements:-

- Operating system: windows XP or higher/ Linux
- Bootstrap
- WAMP Server(Cross-platform, Glassfish(server), MySQL, JSP)
- MySQL
- NetBeans IDE 8.2

Languages used:-

- HTML (Hypertext Markup Language)
- CSS (Cascading Style Sheet)
- JavaScript
- JSP (Java Server Pages)
- XML (eXtensible Markup Language)

Overview of languages used:-

Hypertext Markup Language (HTML):-

Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as cascading style sheets (CSS) and scripting language such as Java script.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by *tags*, written using angle brackets. Tags such as <img/> and <input/> directly introduce content into the page. Other tags such as <p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as Java script which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

Basic HTML Tags:-

<! -- --> specifies comments

<A>……….</A> Creates hypertext links

<B>……….</B> Formats text as bold

<BIG>……….</BIG> Formats text in large font.

<BODY>…</BODY> Contains all tags and text in the HTML document

<CENTER>...</CENTER> Creates text

<DD>…</DD> Definition of a term

<DL>...</DL> Creates definition list

<FONT>…</FONT> Formats text with a particular font

<FORM>...</FORM> Encloses a fill-out form

<FRAME>...</FRAME> Defines a particular frame in a set of frames

<H#>…</H#> Creates headings of different levels (1 – 6)

<HEAD>...</HEAD> Contains tags that specify information about a document

<HR>...</HR> Creates a horizontal rule

<HTML>…</HTML> Contains all other HTML tags

<META>...</META> Provides meta-information about a document

<SCRIPT>…</SCRIPT> Contains client-side or server-side script

<TABLE>…</TABLE> Creates a table

<TD>…</TD> Indicates table data in a table

<TR>…</TR> Designates a table row

<TH>…</TH> Creates a heading in a table

CSS:-

- **CSS** stands for **C**ascading **S**tyle **S**heets.
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media.
- CSS saves a lot of work. It can control the layout of multiple web pages all at once.
- External stylesheets are stored in CSS files.

CSS Syntax:-

```
Selector { Declaration ; Declaration }
              ↓
        Property : value
```

JavaScript:-

JavaScript was initially created to "make web pages alive". The programs in this language are called *scripts*. They can be written right in a web page's HTML and run automatically as the page loads. Scripts are provided and executed as plain text. They don't need special preparation or compilation to run. Today, JavaScript can execute not only in the browser, but also on the server, or actually on any device that has a special program called the JavaScript engine.

JSP:-

- It stands for **Java Server Pages**.
- It is a server side technology.
- It is used for creating web application.
- It is used to create dynamic web content.
- In this JSP tags are used to insert JAVA code into HTML pages.
- It is an advanced version of Servlet Technology.
- It is a Web based technology helps us to create dynamic and platform independent web pages.
- In this, Java code can be inserted in HTML/ XML pages or both.
- JSP is first converted into servlet by JSP container before processing the client's request.

Features of JSP:-

- **Coding in JSP is easy** :- As it is just adding JAVA code to HTML/XML.
- **Reduction in the length of Code** :- In JSP we use action tags, custom tags etc.
- **Connection to Database is easier** :-It is easier to connect website to database and allows to read or write data easily to the database.
- **Make Interactive websites** :- In this we can create dynamic web pages which helps user to interact in real time environment.
- **Portable, Powerful, flexible and easy to maintain** :- as these are browser and server independent.
- **No Redeployment and No Re-Compilation** :- It is dynamic, secure and platform independent so no need to re-compilation.
- **Extension to Servlet** :- as it has all features of servlets, implicit objects and custom tags.

Tags in JSP:-

- **Declaration Tag** :-It is used to declare variables.
  Syntax:- <%! JSP declaration %>

- **Java Scriptlets** :- It allows us to add any number of JAVA code, variables and expressions.
  Syntax:- <% Java code %>

- **JSP Expression** :- It evaluates and convert the expression to a string.
  Syntax:- <%= Java statement %>

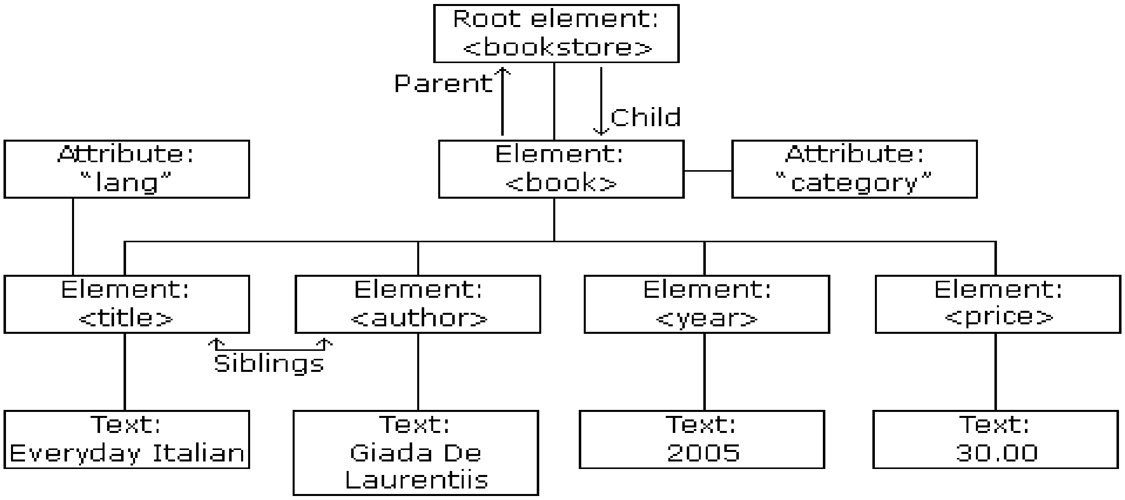
- **JAVA Comments** :- It contains the text that is added for information which has to be ignored.
  Syntax:- <%-- Comments --%>

- **JSP Directive tag:-** It is mainly used to import libraries and to include other pages in the current one.
  Syntax:- <%@ type attribute %>

XML:-

- XML stands for eXtensible Markup Language.
- XML is a markup language much like HTML.
- XML was designed to store and transport data.
- XML was designed to be self-descriptive.
- XML is a W3C Recommendation.

XML Tree Structure example:-



3.2 Database Tables

Student table (Table name : stud)

Column Name	Data Type	Constraint
Uid	Varchar(9)	Primary Key
Name	Varchar(40)	Not null
Pass	Varchar(16)	Not null
Email	Varchar(25)	Not null
phone	Varchar(10)	Not null

Staff table (Table name : staff)

Column Name	Data Type	Constraint
Uid	Varchar(9)	Primary Key
Name	Varchar(40)	Not null
Pass	Varchar(16)	Not null
Email	Varchar(25)	Not null
phone	Varchar(10)	Not null

Library Books table (Table name : books)

Column Name	Data Type	Constraint
Bid	Int(6)	Primary Key
Title	Varchar(70)	Not null
Author	Varchar(50)	Not null
Subject	Varchar(50)	Not null
Isbn	Varchar(20)	Not null
Av_quant	Int(2)	Not null

Students Books table (Table name : stud_books)

Column Name	**Data Type**	**Constraint**
Uid	Varchar(9)	Primary Key
Bid	Int(6)	Not null
Issued_on	Date	Not null
Return_on	Date	Not null
Fine	Int(4)	Not null

Library Books Issue List table (Table name : issue_list)

Column Name	**Data Type**	**Constraint**
Uid	Varchar(9)	Primary Key
Bid	Int(6)	Not null
Title	Varchar(70)	Not null
Author	Varchar(50)	Not null
Book_date	Date	Not null

3.3 Data flow diagrams (DFD's)

DFD is a simple graphical formalism that can be used to represent systematic terms of the input data to a system, various processes are carried out on those data, and the output data is generated by the system. It is also called as bubble chart.

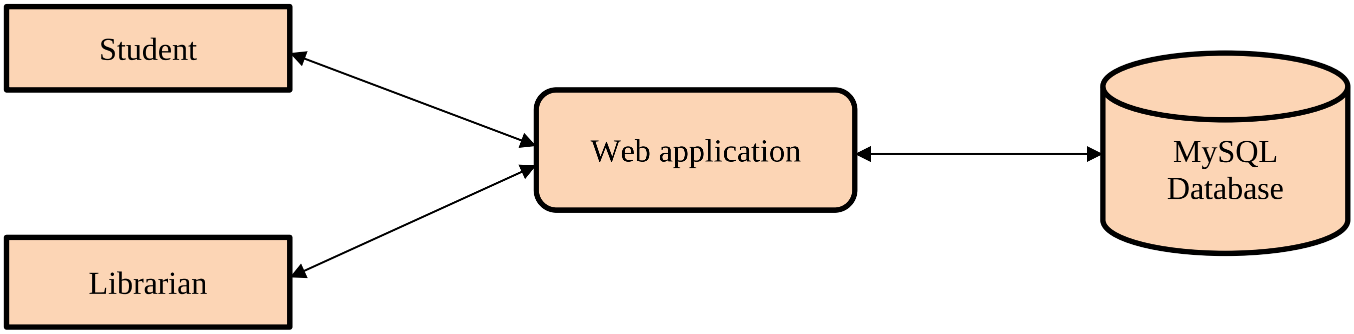
The DFD is one of the most important modelling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information that flows in the system.

The DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts about information flow and the transformations that are applied as data moves from input to output.

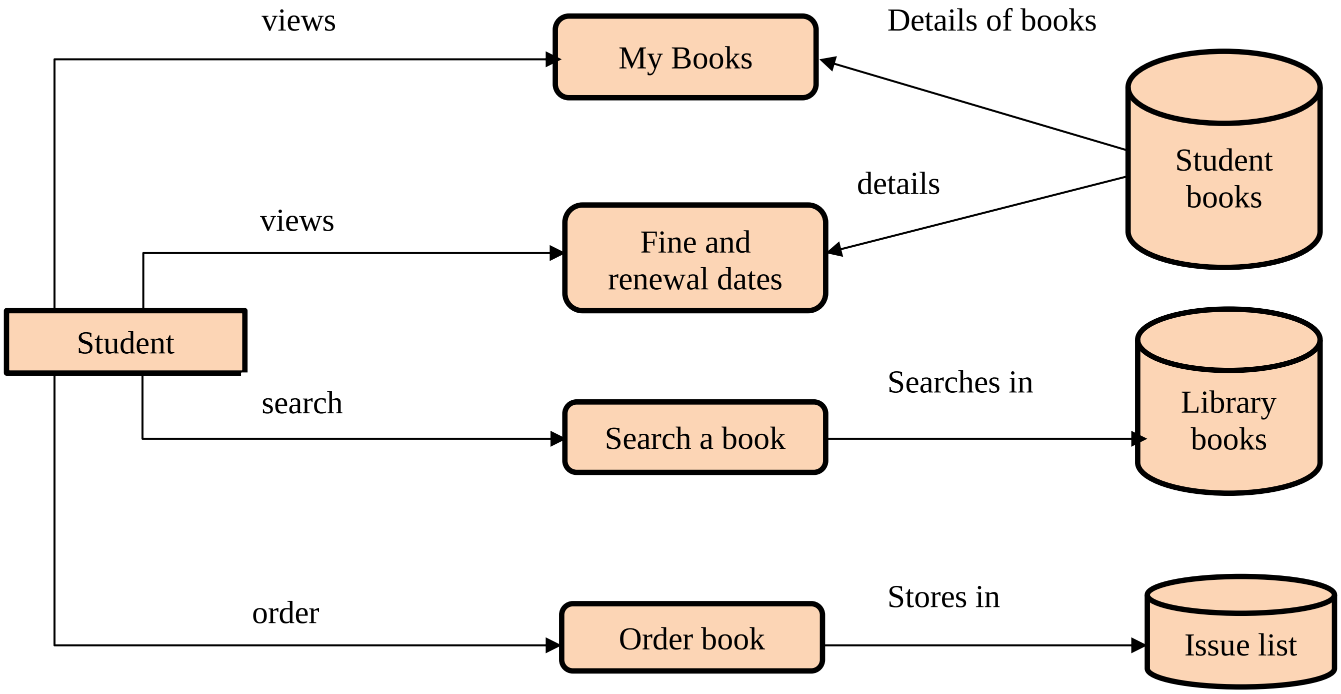
DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into the levels that represent increasing information flow and functional details. DFD is often used as a preliminary step to create an overview of the system. DFD is a designing tool used in top-down approach of a system design.

Context level DFD is most basic form of DFD. It aims to show how the entire system works at a glance. Level 1 DFD's aim to give an overview of the full system. They look at the system in more detail.

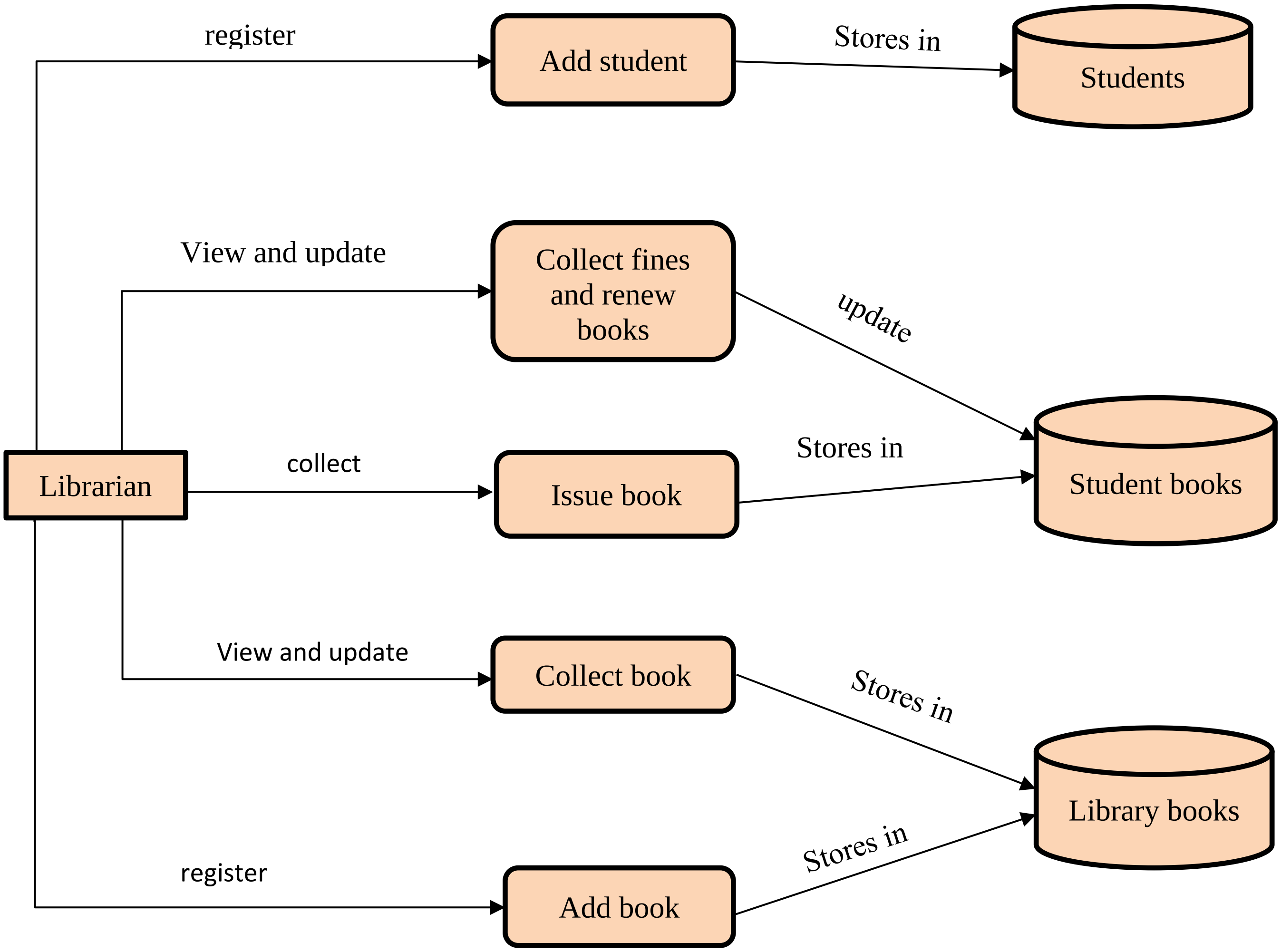
Context level DFD:-



Level 1 DFD for Student:-



Level 1 DFD for Librarian:-



CHAPTER 4 - IMPLEMENTATION

4. IMPLEMENTATION

In this chapter the implementation details of the software or application are explained. The implementation details include modules and some important code. This is the phase where the software is built using the design from previous phase.

4.1 Modules

This application is meant to optimize the process for library management. This application is web based i.e., we can only access the application through a web browser and only if we have an internet connection. This particular application consists of two different modules namely Librarian module and Student module. In turn these two modules contain sub modules or sub parts.

Librarian module:-

Librarian module is developed for use by library staff to maintain the library. This particular module consists of following buttons on the left of window which perform unique operations and display result on the right side:-

1. Add Student - opens a window where we can register new student.
2. Edit Student – opens a window where we can edit the details of existing student.
3. Add Book – opens a window where we can add a new book or add quantity to existing book to a library.
4. Edit Book – opens a window where we can edit details of a book in a library.
5. View Available Books – opens a window with a table consisting of all available books in a library.
6. View Issue List – opens a window with a table consisting of student roll no's along with the books they have ordered. Using this window and buttons provided in it the librarian cn issue books to students based on orders.
7. Collect Book – opens a window which is used to collect the book returned by a student.
8. Collect Fine – opens a window which is used to collect fines from the students.
9. Renew Book – opens a window which is used to renew a book possessed by a student.
10. Change Password – opens a window which is used to change the password of the account.

Student module:-

Student module is developed for use by the students to search and order the required books, view information such as fines, return dates, book details in their account etc. This particular module consists of following buttons on the left of window which perform unique operations and display result on the right side:-

1. View Available Books – opens a window with a table consisting of all the available books in a library.
2. Order Book – opens a window which can be used to search and order the required books.

3. View My Orders – opens a window with a table consisting of your orders along with cancel order button used for cancelling an order.
4. View My Books – opens a window with a table consisting of details about the books you have taken from the library.
5. View My Fines/Return Dates – opens a window with the information of fines and return dates on different books that he/she has taken from the library.
6. View My Details – opens a window with your details in it.
7. Change Password – opens a window which can be used to change the password of the account.

4.2 Sample codes

Home page:

```
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>HOME</title>
        <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css"
>
        <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></scrip
t>
        <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></
script>
    </head>
    <body>
      <style>
      body, html {
  height: 100%;
  margin: 0;
}

.bg {
  /* The image used */
  background-image: url("book.jpg");

  /* Full height */
  height: 100%;
  /*-webkit-filter: blur(12px);*/ /* Safari 6.0 - 9.0 */
  /*filter: blur(12px);*/
  /* Center and scale the image nicely */
  background-position: center;
  background-repeat: no-repeat;
  background-size: cover;
}
          .button {
  background-color: #4CAF50;
  border: none;
  border-radius: 12px;
  box-shadow: 0 8px 16px 0 rgba(0,0,0,0.2), 0 6px 20px 0 rgba(0,0,0,0.19);
  color: white;
  padding: 15px 64px;
  top: 50%;
  left: 50%;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 20px;
  margin: 4px 2px;
  cursor: pointer;
}
.texter{
    color:greenyellow;
    font-family: "Times New Roman", Times, serif;
}

        </style>

<div class="bg">
      <div class="container">

    <center><h1 class="texter">Kakatiya Institute Of Technology And
Science</h1>
    <h4 class="texter">Warangal - 506 015, Telangana, India(An
AUTONOMOUS INSTITUTE under Kakatiya Institute, Warangal)</h4>
    <br>
    <br>
    <h1 class="texter">LIBRARY SYSTEM</h1>
    <br><br>
    <br><br><br>
    <a href="Slogin.jsp" data-target="#Modal" data-toggle="modal"><button
class="button">Student Login</button></a><br><br>
    <br><br><br>
    <a href="Llogin.jsp" data-target="#Modal2" data-toggle="modal"><button
class="button">Librarian Login</button></a>
    <div class="modal fade text-center" id="Modal">
        <div class="modal-dialog">
            <div class="modal-content">
      </div>
    </div>
  </div>
  <div class="modal fade text-center" id="Modal2">
    <div class="modal-dialog">
      <div class="modal-content">

      </div>
    </div>
  </div>
  </center>
  </div>
   </div>
  </body>
</html>
```

Librarian login function JSP code:

```
    <%
  try{
    String username = request.getParameter("uid");
    String password = request.getParameter("pwd");
    String message="The User-Id or password you have entered is incorrect.
Please type the correct credentials and try again.";
    Connection conn=MyDatabase.DatabaseTask.getConnection();
    PreparedStatement pst = conn.prepareStatement("Select * from staff
where uid=? and pass=?");
    pst.setString(1, username);
    pst.setString(2, password);
    ResultSet rs = pst.executeQuery();
    if(rs.next())
    {
      request.setAttribute("uid", username);
      RequestDispatcher dispatcher =
request.getRequestDispatcher("/Lmodule.jsp");
      dispatcher.forward( request, response );
    }
    else
    {
      request.setAttribute("msg", message);
      RequestDispatcher dispatcher =
request.getRequestDispatcher("/libwrnglog.jsp");
      dispatcher.forward( request, response );
    }
  }
  catch(Exception e){
     out.println("Something went wrong !! Please try again : "+e);
  }

finally
{
MyDatabase.DatabaseTask.closeConnection();
}
%>
```

Student login function JSP code:

```
<%
   try{
       String username = request.getParameter("uid");
       String password = request.getParameter("pwd");
               String message="The User-Id or password you have entered is
          incorrect. Please type the correct credentials and try again.";
       Connection conn=MyDatabase.DatabaseTask.getConnection();
       PreparedStatement pst = conn.prepareStatement("Select * from stud
where uid=? and pass=?");
       pst.setString(1, username);
       pst.setString(2, password);
       ResultSet rs = pst.executeQuery();
       if(rs.next())
       {
         request.setAttribute("uid", username);
         RequestDispatcher dispatcher =
request.getRequestDispatcher("/smodule.jsp");
         dispatcher.forward( request, response );
       }
       else
       {
         request.setAttribute("msg", message);
         RequestDispatcher dispatcher =
request.getRequestDispatcher("/studwrnglog.jsp");
         dispatcher.forward( request, response );
       }
  }
  catch(Exception e){
     out.println("Something went wrong !! Please try again : "+e);
  }

finally
```

```
{
MyDatabase.DatabaseTask.closeConnection();
}
%>
```

Librarian module Add student database code:

```
<%
        String roll=request.getParameter("roll");
        String name=request.getParameter("name");
        String pass=request.getParameter("pass");
        String mail=request.getParameter("mail");
        String phone=request.getParameter("ph");
        try
        {
           Connection con=MyDatabase.DatabaseTask.getConnection();
           PreparedStatement pstmt=con.prepareStatement("select * from stud
where uid=?");
           pstmt.setString(1, roll);
           ResultSet rs=pstmt.executeQuery();
           if(rs.next())
           {
              %>
                   <jsp:include page="/addstud.jsp"/>
                   <%
              out.println("<center><h3 class=\"alerttexters\">Student bearing
roll.no "+roll+" already exists</h3></center>");

           }
           else
           {
           PreparedStatement stmt=con.prepareStatement("insert into
stud(uid,name,pass,email,phone) values(?,?,?,?,?)");
           stmt.setString(1, roll);
           stmt.setString(2, name);
           stmt.setString(3, pass);
           stmt.setString(4, mail);
           stmt.setString(5, phone);
           int res=stmt.executeUpdate();
           if(res>0)
           {
              %>
                   <jsp:include page="/addstud.jsp"/>
                   <%
              out.println("<center><h3 class=\"succtexters2\">Student bearing
roll.no "+roll+" registered successfully</h3></center>");
```

```
        }
        else
        {
          %>
                <jsp:include page="/addstud.jsp"/>
                <%
          out.println("<center><h3 class=\"alerttexters\">Student
registration unsuccessful</h3></center>");
        }
        stmt.close();
        }
        pstmt.close();
      }
      catch(Exception e)
      {
          out.println("<center><h3>Registration
unsuccessful"+e+"</h3></center>");
          out.println("<center><h3>Please provide another different
number as id.</h3></center>");
          out.println("</br></br><center>To try again<a
href=\"Index.jsp\">CLICK HERE</a></center>");
      }
      finally
      {
        MyDatabase.DatabaseTask.closeConnection();
      }
      %>
```

Librarian module Edit student database code:

```
<%

       try
      {
        String uid=request.getParameter("uid");
        String name=request.getParameter("name");
        String pass=request.getParameter("pass");
        String mail=request.getParameter("mail");
        String ph=request.getParameter("ph");
        Connection con=MyDatabase.DatabaseTask.getConnection();
        PreparedStatement stmt=con.prepareStatement("update stud set
name=?,pass=?,email=?,phone=? where uid=?");
        stmt.setString(1, name);
        stmt.setString(2, pass);
        stmt.setString(3, mail);
```

```
          stmt.setString(4, ph);
          stmt.setString(5, uid);
          int res=stmt.executeUpdate();
          if(res>0)
          {
            %>
                <jsp:include page="/editst.jsp"/>
                <%
            out.println("<center><h3 class=\"succtexters2\">Details of the
student bearing Roll.No "+uid+" updated successfully.</h3></center>");
          }
          else
          {
%>
                <jsp:include page="/editst.jsp"/>
                <%
            out.println("<center><h3 class=\"alerttexters\">There's been an
error in updating the details of student bearing Roll.No "+uid+", Please try
again.</h3></center>");

          }
          stmt.close();
        }
        catch(Exception e)
        {
            out.println("<center><h3>Registration
unsuccessful"+e+"</h3></center>");
            out.println("<center><h3>Please provide another different
number as id.</h3></center>");
            out.println("</br></br><center>To try again<a
href=\"Index.jsp\">CLICK HERE</a></center>");
        }
        finally
        {
          MyDatabase.DatabaseTask.closeConnection();
        }
        %>
```

Librarian module Add Book database code:

```
<%
        int bid=Integer.parseInt(request.getParameter("bid"));
        String title=request.getParameter("title");
        String author=request.getParameter("author");
        String sub=request.getParameter("sub");
        String isbn=request.getParameter("isbn");
        int quantity=Integer.parseInt(request.getParameter("quantity"));
        try
        {
           Connection con=MyDatabase.DatabaseTask.getConnection();
           PreparedStatement stmt=con.prepareStatement("insert into
books(bid,Title,author,subject,isbn,av_quant) values(?,?,?,?,?,?)");
           stmt.setInt(1, bid);
           stmt.setString(2, title);
           stmt.setString(3, author);
           stmt.setString(4, sub);
           stmt.setString(5, isbn);
           stmt.setInt(6, quantity);
           int res=stmt.executeUpdate();
           if(res>0)
           {
              %>
                   <jsp:include page="/addbook.jsp"/>
                   <%

              out.println("<center><h3 class=\"succtexters2\">Book stock
bearing id "+bid+" added successfully</h3></center>");

           }
           else
           {
              %>
                   <jsp:include page="/addbook.jsp"/>
                   <%
              out.println("<center><h3 class=\"alerttexters\">Book stock
updation of book with id "+bid+" unsuccessful</h3></center>");
           }
           stmt.close();
           }
        catch(Exception e)
        {
              out.println("<center><h3>Registration
unsuccessful"+e+"</h3></center>");
              out.println("<center><h3>Please provide another different
number as id.</h3></center>");
                    out.println("</br></br><center>To try again<a
href=\"Index.jsp\">CLICK HERE</a></center>");
            }
            finally
            {
                MyDatabase.DatabaseTask.closeConnection();
            }
            %>
```

Librarian module view available books code:

```
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.PreparedStatement"%>
<%@page import="java.sql.Connection"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
        <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css"
>
        <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></scrip
t>
        <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></
script>
    </head>
    <body>
        <style>
            .texter{
    color:darkgreen;
    font-size: 20px;
    font-family: "Times New Roman", Times, serif;
}
.texters{
    color:darkgreen;
    font-family: "Times New Roman", Times, serif;
    font-weight: bold;
}
        </style>
        <div class="container">
```

```
  <center>
    <h2 class="texters">BOOKS AVAILABLE IN LIBRARY</h2>
    <table class="table">
      <thead>
      <tr>
        <th class="texter">BOOK ID</th>
        <th class="texter">TITLE</th>
        <th class="texter">AUTHOR</th>
        <th class="texter">SUBJECT</th>
        <th class="texter">ISBN NO.</th>
        <th class="texter">QUANTITY AVAILABLE</th>
      </tr>
      </thead>
    <%
      try
      {
        Connection con=MyDatabase.DatabaseTask.getConnection();
        PreparedStatement pstmt=con.prepareStatement("select * from
books where av_quant>0");
        ResultSet rs=pstmt.executeQuery();
        while(rs.next())
        {
          %>
          <tbody>
          <tr>
            <td><%=rs.getInt(1)%></td>
            <td><%=rs.getString(2)%></td>
            <td><%=rs.getString(3)%></td>
            <td><%=rs.getString(4)%></td>
            <td><%=rs.getString(5)%></td>
            <td><%=rs.getInt(6)%></td>
          </tr>
          <%
        }
      }
      catch(Exception e)
      {
        System.out.println("Exception"+e);
      }
      finally
      {

      }
    %>
          </tbody>
    </table>
  </center>
    </div>
  </body>
</html>
```

Librarian module issue book database code:

```
<%
      String uid=request.getParameter("uid");
      int bid=Integer.parseInt(request.getParameter("bid"));
      try
      {
        String isbn;
        Connection con=MyDatabase.DatabaseTask.getConnection();
        PreparedStatement pst1=con.prepareStatement("select isbn from
books where bid=?");
        pst1.setInt(1, bid);
        ResultSet rs1=pst1.executeQuery();
        if(rs1.next())
        {
          isbn=rs1.getString(1);
        }
        java.sql.Date sqlDate = new java.sql.Date(new
java.util.Date().getTime());
        Calendar c=new GregorianCalendar();
        c.add(Calendar.DATE, 30);
        Date d=c.getTime();
        java.sql.Date sqlDate2 = new java.sql.Date(d.getTime());
          PreparedStatement pst3=con.prepareStatement("insert into
stud_books(uid,bid,issued_on,return_on,fine) values(?,?,?,?,?) ");
          pst3.setString(1, uid);
          pst3.setInt(2, bid);
          pst3.setDate(3, sqlDate);
          pst3.setDate(4, sqlDate2);
          pst3.setInt(5,0);
          int res2=pst3.executeUpdate();
          if(res2>0)
          {
            PreparedStatement pst4=con.prepareStatement("delete from
issue_list where uid=? and bid=?");
            pst4.setString(1, uid);
            pst4.setInt(2, bid);
            int res3=pst4.executeUpdate();
            if(res3>0)
            {
               out.println("<center><h2 class=\"texter\">Book with id 
"+bid+" issued to student bearing roll.no "+uid+" 
successfully</h2></center>");
               %>
               <jsp:include page="/issuebk.jsp"/>
               <%
            }
            else
            {
               out.println("<center><h2>FAILED</h2></center>");
            }
          }
          else
             out.println("<center><h2>FAILED</h2></center>");

      }
      catch(Exception e)
      {
        out.println("<center><h2>EXCEPTION  "+e+"</h2></center>");
      }
      %>
```

Librarian module collect book database code:

```
<%
         String uid=request.getParameter("uid");
         int bid=Integer.parseInt(request.getParameter("bid"));
         try
         {
            Connection con=MyDatabase.DatabaseTask.getConnection();
            PreparedStatement pst=con.prepareStatement("delete from 
stud_books where uid=? and bid=? and fine=0");
            pst.setString(1,uid);
            pst.setInt(2,bid);
            int res=pst.executeUpdate();
            if(res>0)
            {
               PreparedStatement pst2=con.prepareStatement("update books set 
av_quant=av_quant+1 where bid=?");
            pst2.setInt(1,bid);
            int res2=pst2.executeUpdate();
            if(res2>0)
            {
               out.println("<center><h3 class=\"succtexters2\">Book collected 
and added back to library successfully</h3></center>");
```

```
                    %>
                    <jsp:include page="collectbk_func.jsp">
        <jsp:param name="uid" value="<%=uid%>" />
        </jsp:include>
                    <%
                }
else
out.println("<center><h2>FAILED</h2></center>");
                }
else
{
out.println("<center><h3 class=\"alerttexters\">Book cannot be collected
because of fine imposed on it</h3></center>");
%>
<jsp:include page="collectbk_func.jsp">
        <jsp:param name="uid" value="<%=uid%>" />
        </jsp:include>
<%
}
}
            catch(Exception e)
            {
                out.println("<center><h2>Exception " +e+"</h2></center>");
            }
            %>
```

Librarian module collect fine database code:

```
<%
            String uid=request.getParameter("uid");
            int bid=Integer.parseInt(request.getParameter("bid"));
            int amt=Integer.parseInt(request.getParameter("amt"));
            try
            {
                Connection con=MyDatabase.DatabaseTask.getConnection();
                PreparedStatement pst=con.prepareStatement("update stud_books
set fine=fine-? where uid=? and bid=?");
                pst.setInt(1, amt);
                pst.setString(2, uid);
                pst.setInt(3, bid);
                int res=pst.executeUpdate();
                if(res>0)
                {
                   out.println("<center><h2 class=\"texter\">Fine collected for book
id "+bid+" from "+uid+" successfully</h2></center>");
```

```
                %>
                <jsp:include page="finebk_func.jsp">
        <jsp:param name="uid" value="<%=uid%>" />
        </jsp:include>
                <%
             }
else
out.println("<center><h2>FAILED</h2></center>");
          }
          catch(Exception e)
          {
             out.println("<center><h2>Exception " +e+"</h2></center>");
          }
          %>
```

Librarian module Renew book database code:

```
<%
          String uid=request.getParameter("uid");
          int bid=Integer.parseInt(request.getParameter("bid"));
          try
          {
             Connection con=MyDatabase.DatabaseTask.getConnection();
             PreparedStatement st=con.prepareStatement("select fine from
stud_books where uid=? and bid=?");
             st.setString(1, uid);
             st.setInt(2, bid);
             ResultSet rs=st.executeQuery();
             if(rs.next())
             {
                int fine=rs.getInt("fine");
                if(fine==0)
                {
             java.sql.Date sqlDate = new java.sql.Date(new
java.util.Date().getTime());
             Calendar c=new GregorianCalendar();
             c.add(Calendar.DATE, 30);
             Date d=c.getTime();
             java.sql.Date sqlDate2 = new java.sql.Date(d.getTime());
             PreparedStatement pst=con.prepareStatement("update stud_books
set return_on=? where uid=? and bid=?");
             pst.setDate(1, sqlDate2);
             pst.setString(2,uid);
             pst.setInt(3,bid);
             int res=pst.executeUpdate();
```

```
            if(res>0)
            {
               %>
               <jsp:include page="renewbk_func.jsp">
      <jsp:param name="uid" value="<%=uid%>" />
      </jsp:include>
               <%
                  out.println("<center><h3 class=\"texters2\">Book with book id
"+bid+" renewed successfully</h3></center>");
}
            else
                out.println("<center><h2>FAILED</h2></center>");
         }
else
{
          %>
               <jsp:include page="renewbk_func.jsp">
      <jsp:param name="uid" value="<%=uid%>" />
      </jsp:include>
               <%
                  out.println("<center><h3 class=\"alerttext\">Book with book
id "+bid+" cannot be renewed until a fine of Rs."+fine+" imposed on the book
is paid</h3></center>");
}
}
}
         catch(Exception e)
         {
            out.println("<center><h2>Exception " +e+"</h2></center>");
         }
         %>
```

Change password database code (Librarian):

```
<%
         String id=request.getParameter("uid");
         String old_pass=request.getParameter("old_pass");
         String new_pass=request.getParameter("new_pass");
         try
         {
            Connection con=MyDatabase.DatabaseTask.getConnection();
            PreparedStatement pst=con.prepareStatement("select * from staff
where uid=? and pass=?");
            pst.setString(1, id);
            pst.setString(2, old_pass);
            ResultSet rs=pst.executeQuery();
```

```
        if(rs.next())
        {
          PreparedStatement pst2=con.prepareStatement("update staff set
pass=? where uid=?");
          pst2.setString(1, new_pass);
          pst2.setString(2, id);
          int res=pst2.executeUpdate();
          if(res>0)
          {
            %>
          <jsp:include page="/Lib_changepass.jsp?uid=<%=id%>"/>
          <%
            out.println("<center><h3 class=\"texters2\">Password changed
successfully.</h3></center>");
          }
          else
          {
            %>
          <jsp:include page="/Lib_changepass.jsp?uid=<%=id%>"/>
          <%
            out.println("<center><h3 class=\"texters\">Password change
unsuccessful. Please try again.</h3></center>");
          }
          pst2.close();
        }
        else
        {
          %>
          <jsp:include page="/Lib_changepass.jsp?uid=<%=id%>"/>
          <%
          out.println("<center><h3 class=\"texters\">The old password
entered is not matching with the account password, please type the correct
password and try again.</h3></center>");
        }
        pst.close();
        rs.close();
      }
      catch(Exception e)
      {
        out.println("Error : "+e);
      }
      finally
      {
        MyDatabase.DatabaseTask.closeConnection();
      }
      %>
```

Student module order book database code:

```
    <%
      String author=request.getParameter("author");
      String title=request.getParameter("title");
      String uid=request.getParameter("uid");
      int bid=Integer.parseInt(request.getParameter("bid"));
      int count=0;
      %>
      <jsp:include page="/Stud_orderbk.jsp"/>
      <%
      try
      {
       Connection con=MyDatabase.DatabaseTask.getConnection();
       PreparedStatement pst=con.prepareStatement("select count(*) from
issue_list where uid=?");
       pst.setString(1, uid);
       PreparedStatement pst2=con.prepareStatement("select count(*) from
stud_books where uid=?");
       pst2.setString(1, uid);
       ResultSet rsc1=pst.executeQuery();
       ResultSet rsc2=pst2.executeQuery();
       if(rsc1.next() && rsc2.next())
       {
         count=rsc1.getInt(1)+rsc2.getInt(1);
       }
       if(count<4)
       {
       PreparedStatement st1=con.prepareStatement("select * from
issue_list where uid=? and bid=?");
       st1.setString(1, uid);
       st1.setInt(2, bid);
       ResultSet rs1=st1.executeQuery();
       if(!rs1.next())
       {
         PreparedStatement st2=con.prepareStatement("select * from
stud_books where uid=? and bid=?");
         st2.setString(1, uid);
         st2.setInt(2, bid);
         ResultSet rs2=st2.executeQuery();
         if(!rs2.next())
         {
           java.sql.Date sqlDate = new java.sql.Date(new
java.util.Date().getTime());
           PreparedStatement st3=con.prepareStatement("insert into
issue_list(uid,bid,title,author,book_date) values(?,?,?,?,?)");
```

```
                    st3.setString(1, uid);
                    st3.setInt(2, bid);
                    st3.setString(3, title);
                    st3.setString(4, author);
                    st3.setDate(5, sqlDate);
                    int res1=st3.executeUpdate();
                    if(res1>0)
                    {
                        PreparedStatement st4=con.prepareStatement("update books 
set av_quant=av_quant-1 where bid=?");
                        st4.setInt(1, bid);
                        int res2=st4.executeUpdate();
                        if(res2>0)
                        {
                            out.println("<center><h3 class=\"succtexters2\">Order 
placed successfully for book with title \""+title+"\", Please collect the book 
from library within 2 days or else the order gets expired.</h3></center>");
                        }
                        else
                        {
                            out.println("<center><h3 class=\"alerttexters\">An error 
occured,Please try again.</h3></center>");
                        }
                    }
                    else
                    {
                        out.println("<center><h3 class=\"alerttexters\">An error 
occured,Please try again.</h3></center>");
                    }
                }
                else
                {
                    out.println("<center><h3 class=\"alerttexters\">You already 
possess the book with title \""+title+"\" with you.</h3></center>");
                }
            }
            else
            {
               out.println("<center><h3 class=\"alerttexters\">You have already 
ordered the book with title \""+title+"\", please collect the book from library 
within 2 days or else the order gets expired.</h3></center>");
            }
            }
            else
            {
              out.println("<center><h3 class=\"alerttexters\">The count of books
you have ordered and couunt of books with you combined have exceeded the
limit.</h3></center>");
             }
            }
            catch(Exception e)
            {
               out.println("Exception : " +e);
            }
            finally
            {
               MyDatabase.DatabaseTask.closeConnection();
            }
        %>
```

Student module view my orders code:

```
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.PreparedStatement"%>
<%@page import="java.sql.Connection"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
        <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css"
>
        <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></scrip
t>
        <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></
script>
        <style>
            .texter{
    color:darkgreen;
    font-size: 20px;
    font-family: "Times New Roman", Times, serif;
}
.texters{
    color:darkgreen;
    font-family: "Times New Roman", Times, serif;
    font-weight: bold;
}
```

```
.button {
  background-color: #4CAF50;
  border: none;
  border-radius: 12px;
  /*box-shadow: 0 8px 16px 0 rgba(0,0,0,0.2), 0 6px 20px 0
rgba(0,0,0,0.19);*/
  color: white;
  padding: 8px 15px;
  top: 50%;
  left: 50%;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 13px;
  font-family: "Times New Roman", Times, serif;
  margin: 4px 2px;
  cursor: pointer;
}
.alerttexters{
    color: red;
    font-family: "Times New Roman", Times, serif;
    font-weight: bold;
}
        </style>
    </head>
    <body>
        <%
            String uid=request.getParameter("uid");
            try
            {
                Connection con=MyDatabase.DatabaseTask.getConnection();
               PreparedStatement pstmt=con.prepareStatement("select count(*)
from stud_books where uid=?");
               pstmt.setString(1, uid);
               ResultSet rs=pstmt.executeQuery();
               if(rs.next() && rs.getInt(1)>0)
               {
                   PreparedStatement pstmt2=con.prepareStatement("select bid from
stud_books where uid=?");
               pstmt2.setString(1, uid);
               ResultSet rs2=pstmt2.executeQuery();
               %>
               <div class="container">
    <center>
        <h2 class="texters">MY  BOOKS</h2>
        <table class="table">
        <thead>
        <tr>
           <th class="texter">BOOK ID</th>
           <th class="texter">TITLE</th>
           <th class="texter">AUTHOR</th>
           <th class="texter">SUBJECT</th>
           <th class="texter">ISBN NUMBER</th>
        </tr>
        </thead>
          <%
          while(rs2.next())
          {
             PreparedStatement pstmt3=con.prepareStatement("select
Title,author,subject,isbn from books where bid=?");
             pstmt3.setInt(1, rs2.getInt(1));
             ResultSet rs3=pstmt3.executeQuery();
             if(rs3.next())
             {
              %>
              <tbody>
             <tr>
              <td><%=rs2.getInt(1)%></td>
                <td><%=rs3.getString(1)%></td>
                <td><%=rs3.getString(2)%></td>
                <td><%=rs3.getString(3)%></td>
                <td><%=rs3.getString(4)%></td>
             </tr>
              <%
             }
          }
          }
else
{
out.println("<center><h3 class=\"alerttexters\">There are no books taken by
you from the library</h3></center>");

}
}
        catch(Exception e)
        {
           out.println("<center><h2>Exception"+e+"</h2></center>");
        }
finally
{
MyDatabase.DatabaseTask.closeConnection();
}
```

```
          %>
                    </tbody>
        </table>
    </center>
                </div>
    </body>
</html>
```

Student module view fine/return dates code:

```
<%@page import="java.sql.*"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
        <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css"
>
        <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></scrip
t>
        <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></
script>
        <style>
            .texter{
    color:darkgreen;
    font-size: 20px;
    font-family: "Times New Roman", Times, serif;
}
.texters{
    color:darkgreen;
    font-family: "Times New Roman", Times, serif;
    font-weight: bold;
}
.button {
  background-color: #4CAF50;
  border: none;
  border-radius: 12px;
  /*box-shadow: 0 8px 16px 0 rgba(0,0,0,0.2), 0 6px 20px 0
rgba(0,0,0,0.19);*/
  color: white;
  padding: 8px 15px;
  top: 50%;
  left: 50%;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 13px;
  font-family: "Times New Roman", Times, serif;
  margin: 4px 2px;
  cursor: pointer;
}
.alerttexters{
    color: red;
    font-family: "Times New Roman", Times, serif;
    font-weight: bold;
}
        </style>
    </head>
    <body>
        <%
            String uid=request.getParameter("uid");
            try
            {
                Connection con=MyDatabase.DatabaseTask.getConnection();
               PreparedStatement pstmt=con.prepareStatement("select count(*)
from stud_books where uid=?");
               pstmt.setString(1, uid);
               ResultSet rs=pstmt.executeQuery();
               if(rs.next() && rs.getInt(1)>0)
               {
                   PreparedStatement pstmt2=con.prepareStatement("select
bid,return_on,fine from stud_books where uid=?");
               pstmt2.setString(1, uid);
               ResultSet rs2=pstmt2.executeQuery();
               %>
               <div class="container">
    <center>
        <h2 class="texters">FINES  AND  RETURN  DATES</h2>
        <table class="table">
            <thead>
            <tr>
                <th class="texter">BOOK ID</th>
                <th class="texter">TITLE</th>
                <th class="texter">AUTHOR</th>
                <th class="texter">SUBJECT</th>
                <th class="texter">RETURN DATE</th>
                <th class="texter">FINE IMPOSED</th>
```

```
      </tr>
      </thead>
        <%
        while(rs2.next())
        {
          PreparedStatement pstmt3=con.prepareStatement("select
Title,author,subject,isbn from books where bid=?");
          pstmt3.setInt(1, rs2.getInt(1));
          ResultSet rs3=pstmt3.executeQuery();
          if(rs3.next())
          {
           %>
           <tbody>
          <tr>
           <td><%=rs2.getInt(1)%></td>
            <td><%=rs3.getString(1)%></td>
            <td><%=rs3.getString(2)%></td>
            <td><%=rs3.getString(3)%></td>
            <td><%=rs2.getDate(2)%></td>
            <td><%=rs2.getInt(3)%></td>
          </tr>
           <%
          }
        }
        }
else
{
out.println("<center><h3 class=\"alerttexters\">There are no books taken by
you from the library</h3></center>");
}
}
      catch(Exception e)
      {
        out.println("<center><h2>Exception"+e+"</h2></center>");
      }
finally
{
MyDatabase.DatabaseTask.closeConnection();
}
      %>
           </tbody>
    </table>
  </center>
        </div>
  </body>
</html>
```

Student module view his/her details code:

```
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.PreparedStatement"%>
<%@page import="java.sql.Connection"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
       <style>
.texterhead{
    color:darkgreen;
    font-family: "Times New Roman", Times, serif;
}
            .texter{
    color:darkgreen;
    font-size:18px;
    font-family: "Times New Roman", Times, serif;
}
        </style>
    <center><div>
            <%
                String uid=request.getParameter("uid");
                try
                {
                     Connection con=MyDatabase.DatabaseTask.getConnection();
                PreparedStatement pstmt=con.prepareStatement("select * from stud
where uid=?");
                pstmt.setString(1, uid);
                ResultSet rs=pstmt.executeQuery();
                if(rs.next())
                {
                %>
            <h1 class="texterhead">MY  DETAILS</h1>

            <table frame="none" cellspacing="25">
                <tr>
                    <td class="texter">Roll No</td>
                    <td class="texter">:<input type="text"
value="<%=rs.getString(1)%>" disabled></td>
                </tr>
                <tr>
                    <td class="texter">Name</td>
                    <td class="texter">:<input type="text" name="name"
value="<%=rs.getString(2)%>" disabled></td>
                </tr>

                <tr>
                    <td class="texter">Email</td>
                    <td class="texter">:<input type="text" name="mail"
value="<%=rs.getString(4)%>" disabled></td>
                </tr>
                <tr>
                    <td class="texter">Phone</td>
                    <td class="texter">:<input type="text" name="ph"
value="<%=rs.getString(5)%>" disabled></td>
                </tr>

            </table>

           <%
              }
              else
{
out.println("<center><h2>Student with the particular Roll No does not
exists</h2></center>");
                    out.println("</br></br><center><h2>To add this Student to the
library database please click \"Add Student\" button on your
left</h2></center>");
}
                }
                catch(Exception e)
                {
                    out.println(“Error : ”+e);
                }
                %>
        </div></center>
    </body>
</html>
```

Automatic fine updating code (Every 24 hours):

```
<%
            Connection con=null;
            response.setIntHeader("Refresh", 86400 );
            try
           {
               con=MyDatabase.DatabaseTask.getConnection();
               PreparedStatement pstmt=con.prepareStatement("update
stud_books set fine=fine+2 where DATEDIFF(return_on,CURDATE())<0;");
               int res=pstmt.executeUpdate();
               if(res>0)
               {
                   out.println("Success");
               }
               else
               {
                   out.println("Failed");
               }
               pstmt.close();
           }
           catch(Exception e)
           {
               out.println(e);
           }
            finally
            {
                MyDatabase.DatabaseTask.closeConnection();
            }
        %>
```

Automatic return book warning SMS send code (Every 24 hours):

```
<%@page import="java.sql.*"%>
<%@page import="java.net.* "%>
<%@page import="java.io.* "%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <%
        Connection con=null;
        BufferedReader rd=null;
        response.setIntHeader("Refresh", 86400 );
        try {
                con = MyDatabase.DatabaseTask.getConnection();
                String apiKey = "apikey=" +
URLEncoder.encode("sg+bnN8tWLE-
pYMjnX09vLp9ImQGwoDzvUYJXZLWKn", "UTF-8");
                String sender = "&sender=" +
URLEncoder.encode("TXTLCL", "UTF-8");
                String send_url = "https://api.textlocal.in/send/?";
                PreparedStatement pst1=con.prepareStatement("select
uid,bid,DATEDIFF(return_on,CURDATE()) from stud_books where
DATEDIFF(return_on,CURDATE())>=0 and
DATEDIFF(return_on,CURDATE())<4;");
                ResultSet rs1=pst1.executeQuery();
                while(rs1.next())
                {
                   String msg="";
                   String number="";
                   if(rs1.getInt(3)==0)
                   {
                      msg+="You have to return the book with below details by
today or else fine will be imposed.\n";
                   }
                   else
                   {
                      msg+="You have to return the book with below details
within "+rs1.getInt(3)+" days or else fine will be imposed.\n";
                   }
                   PreparedStatement pst2=con.prepareStatement("select phone
from stud where uid=?");
                   pst2.setString(1,rs1.getString("uid"));
                   ResultSet rs2=pst2.executeQuery();
                   if(rs2.next())
                   {
                      number += rs2.getString("phone");
                   }
                   else
                   {
                      out.println("Error retrieving phone number.");
                   }
                   PreparedStatement pst3=con.prepareStatement("select
Title,author,isbn from books where bid=?");
                   pst3.setInt(1, rs1.getInt("bid"));
                   ResultSet rs3=pst3.executeQuery();
```

```
                        if(rs3.next())
                        {
                           msg+="BOOK DETAILS:\nBook Id :
"+rs1.getInt("bid")+"\nTitle : "+rs3.getString("Title")+"\nAuthor :
"+rs3.getString("author");
                           msg+="\nISBN No. : "+rs3.getString("isbn");
                        }
                        else
                        {
                           out.println("Error retrieving book details.");
                        }
                        msg+="\nNOTE:-Library closing time is 4:30pm.";
                        //out.println(msg);
                        String message = "&message=" + URLEncoder.encode(msg,
"UTF-8");
                              String numbers = "&numbers=" +
URLEncoder.encode("91"+number, "UTF-8");

                              // Send data
                              String data = "" + send_url + apiKey + numbers +
message + sender;
                              URL url = new URL(data);
                              URLConnection conn = url.openConnection();
                              conn.setDoOutput(true);
                     rd = new BufferedReader(new
InputStreamReader(conn.getInputStream()));
                              String line;
                              String sResult="";
                              while ((line = rd.readLine()) != null) {
                              // Process line...
                                       sResult=sResult+line+" ";
}
                              rd.close();
                              pst2.close();
                     pst3.close();
                              out.println(sResult);
                     }
                     pst1.close();
                     } catch (Exception e) {
                              out.println("Error SMS "+e);}
           finally{
       MyDatabase.DatabaseTask.closeConnection();}
%>
   </body>
</html>
```

CHAPTER 5: TESTING

5. TESTING

Testing is the process of detecting errors. Testing performs a very critical role for quality assurance and for ensuring the reliability of software. The results of testing are used later on during maintenance also.

Psychology of Testing:

The aim of testing is often to demonstrate that a program works by showing that it has no errors. The basic purpose of testing phase is to detect the errors that may be present in the program. Hence one should not start testing with the intent of showing that a program works, but the intent should be to show that a program doesn't work. Testing is the process of executing a program with the intent of finding errors.

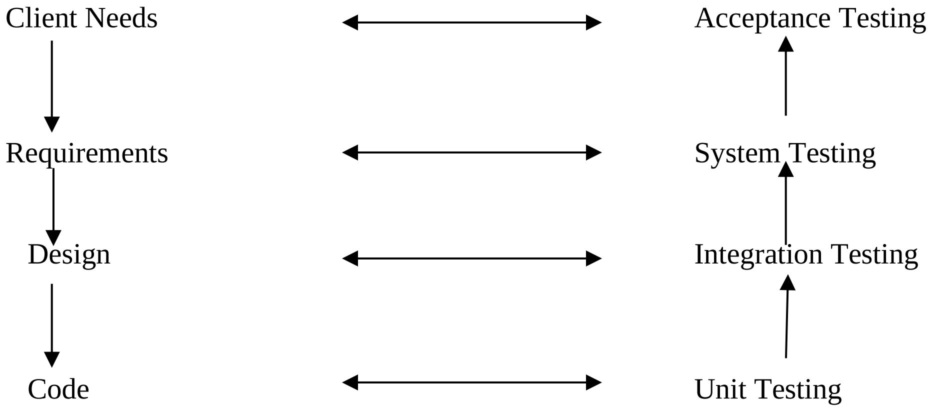
Testing Objectives:

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally, we can say, Testing is a process of executing a program with the intent of finding an error.

- A successful test is one that uncovers an as yet undiscovered error.
- A good test case is one that has a high probability of finding error, if it exists.
- The tests are inadequate to detect possibly present errors.
- The software more or less confirms to the quality and reliable standards.

LEVELS OF TESTING:

In order to uncover the errors present in different phases we have the concept of levels of testing. The basic levels of testing are:-



Unit testing:

Unit testing is essentially for the verification of the code produced during the coding phase and the goal is test. The internal logic of the module/program. In the Generic code project, the unit testing is done during coding phase of data entry forms whether the functions are working properly or not. In this phase all the drivers are tested they are rightly connected or not.

Integration Testing:

After the unit testing we have to perform integration testing. The goal here is to see if modules can be Integrated properly, the emphasis being on testing interfaces between modules. This testing activity can be considered as testing the design and hence the emphasis on testing module interactions.

- **Top-down Integration**

  This is an incremental approach for construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main control module (main program).

- **Bottom-up Integration**

  This, as its name implies, begins construction and testing with components at the lowest levels in the program structure.

Regression Testing:

This is the re-execution of some subset of tests that have already been conducted to ensure that changes have not propagated unintended side effects.

System Testing:

Here the entire software system is tested. The reference document for this process is the requirements document, and the goal as to see if software meets its requirements.

Acceptance Testing:

Acceptance Test is performed with realistic data of the client to demonstrate that the software is working satisfactorily. Testing here is focused on external behavior of the system, the internal logic of program is not emphasized.

White Box Testing:

White box testing is a testing case design method that uses the control structure of the procedure design to derive test cases. All independents path in a module are exercised at least once, all logical decisions are exercised at once, execute all loops at boundaries and within their operational bounds exercise internal data structure to ensure their validity. Here the customer is given three chances to enter a valid choice out of the given menu. After which the control exits the current menu.

Black Box Testing:

This testing method considers a module as a single unit and checks the unit at interface and communication with other modules rather than getting into details at statement level. Here the module will be treated as a block box that will take some input and generate output. Output for a given set of input combinations are forwarded to other modules.

Validation Testing:

At the culmination of integration testing, software is completely assembled as a package, interfacing errors have been uncovered and corrected, and a final series of software tests validation testing may begin. Validation can be defined in many ways, but a simple definition is that validation succeeds when software functions in a manner that can be reasonably expected by the customer. Reasonable expectations are defined in a document that describes all user-visible attributes of the software. This is called as “software requirements specification”. This contains a validation criteria section. Information contained in that section forms the basis for a validation testing approach Alpha and Beta Testing.

Alpha and Beta Testing:

It is virtually impossible for a software developer to foresee how the customer will really use a program. Instructions for use may be misinterpreted; strange combinations of data may be regularly used; output that seemed clear to the tester may be unintelligible to a user in the field. When custom software is built for one customer, a collection of acceptance tests are conducted to enable the customer to validate all requirements. Conducted by the end use rather than software engineers, an acceptance test can range from an informal “test drive" to a planned and systematically executed series of tests. In fact, acceptance testing can be conducted over a period of weeks or months, thereby uncovering cumulative errors that might degrade the system over time. If software is developed as a product to be used by many customers, it is impractical toper form formal acceptance tests with each one. Most software product builders use a process called alpha and beta testing to uncover errors that only the end-user seems able to find.

This Alpha testing is conducted at the developer's site by a customer. The software is used in a natural setting with the developer "looking over the shoulder" of the user and recording errors and usage problems. Alpha tests are conducted in a controlled environment.

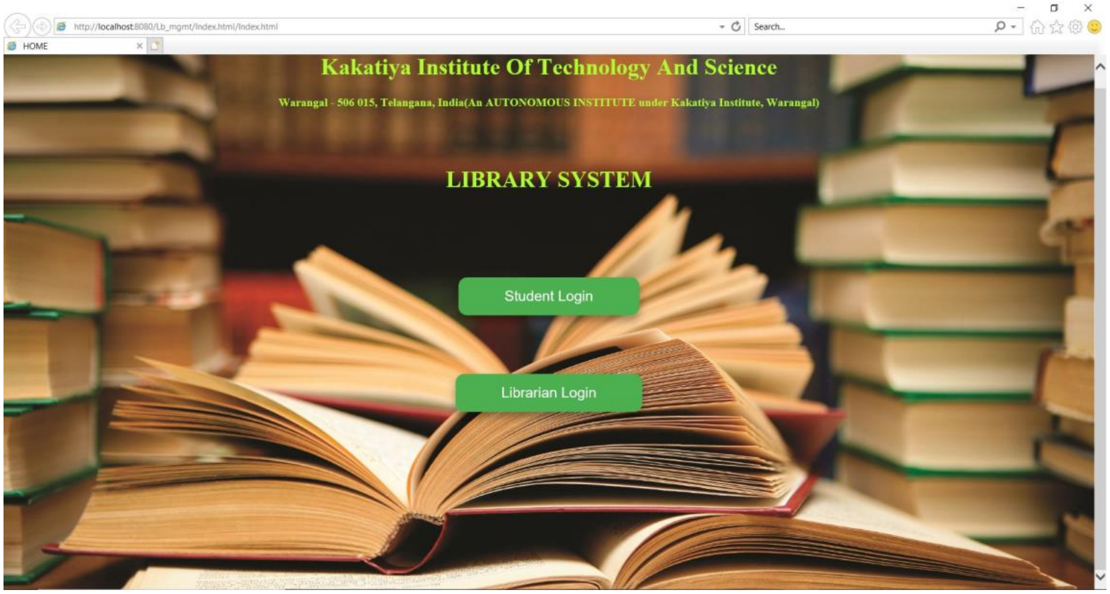
This Beta testing is conducted at one or more customer sites by the end-user of the software. Unlike Alpha testing, the developer is generally not present. Therefore, the beta test is a "live" application of the software in an environment that cannot be controlled by the developer. The customer records all problems that are encountered during beta testing and reports these to the developer at regular intervals. As a result of problems reported during beta tests, software engineers make modifications and then prepare for release of the software product to the entire customer base.

CHAPTER 6 - RESULTS

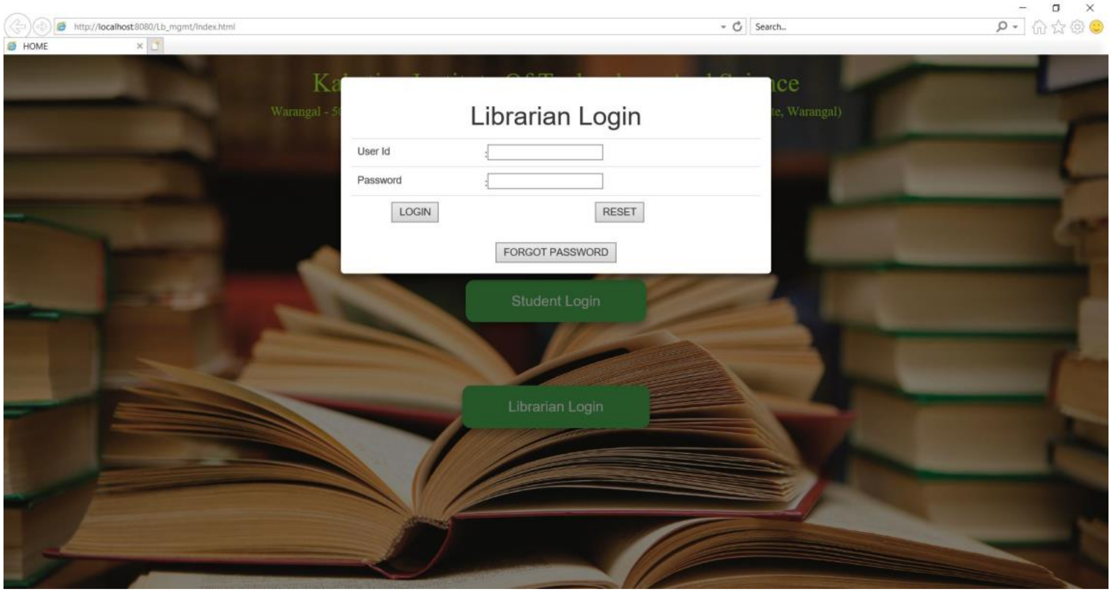
6. RESULTS

This chapter presents some screenshots taken while using the developed application. They can be called as the results or outputs of executing the application.

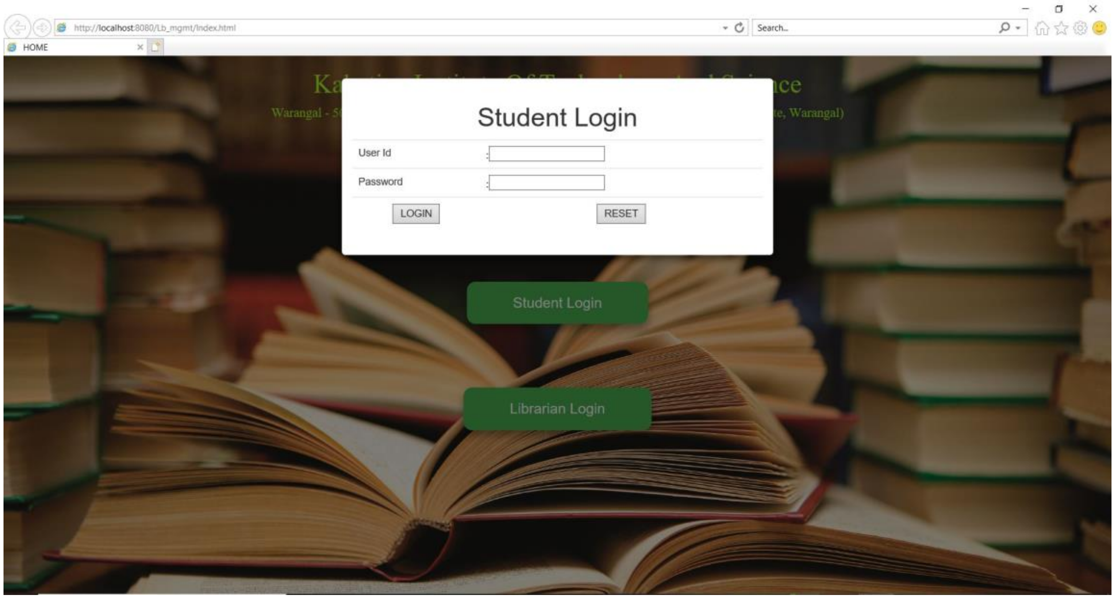
Home page:


http://localhost:8080/Lb_mgmt/Index.html/Index.html
Search...
HOME
Kakatiya Institute Of Technology And Science
Warangal - 506 015, Telangana, India(An AUTONOMOUS INSTITUTE under Kakatiya Institute, Warangal)
LIBRARY SYSTEM
Student Login
Librarian Login

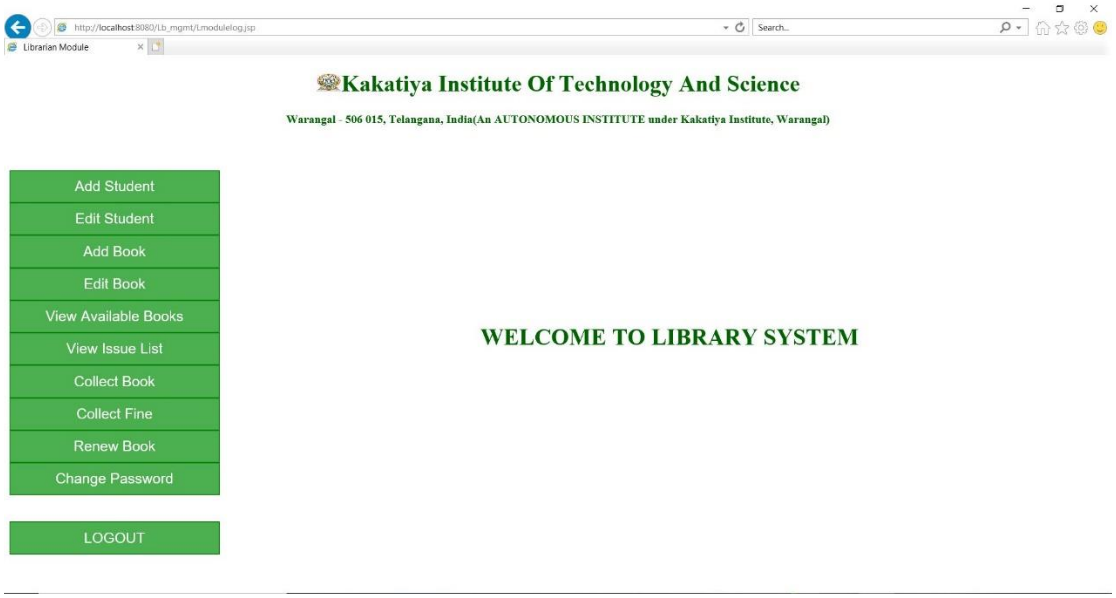
Librarian login page:



Student login page:



Librarian home page:

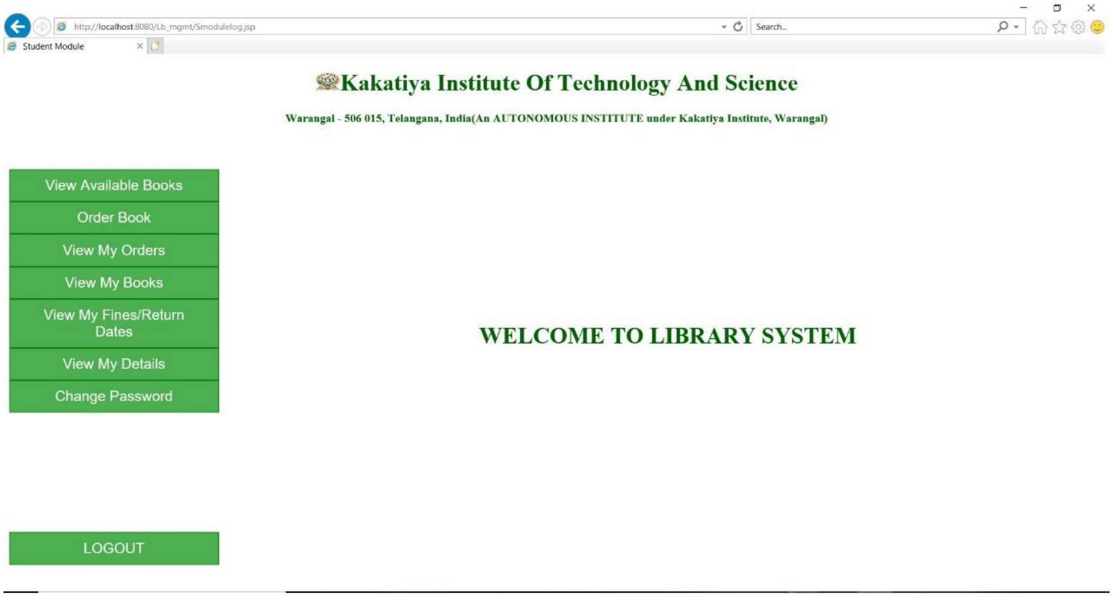


Kakatiya Institute Of Technology And Science

Warangal - 506 015, Telangana, India(An AUTONOMOUS INSTITUTE under Kakatiya Institute, Warangal)

WELCOME TO LIBRARY SYSTEM

Student home page:

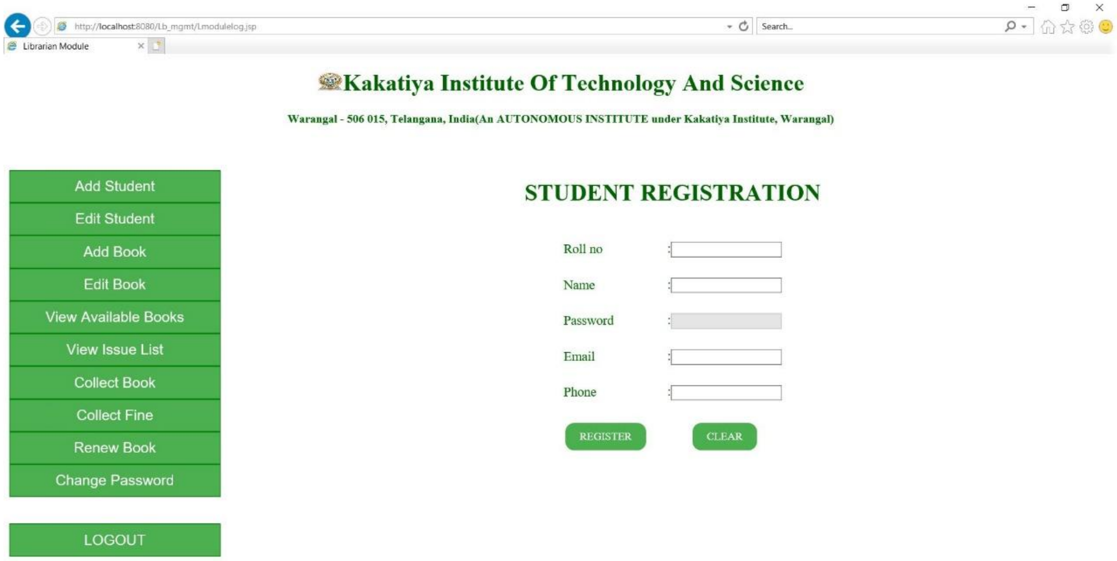


Kakatiya Institute Of Technology And Science

Warangal - 506 015, Telangana, India(An AUTONOMOUS INSTITUTE under Kakatiya Institute, Warangal)

WELCOME TO LIBRARY SYSTEM

Librarian module add student page:



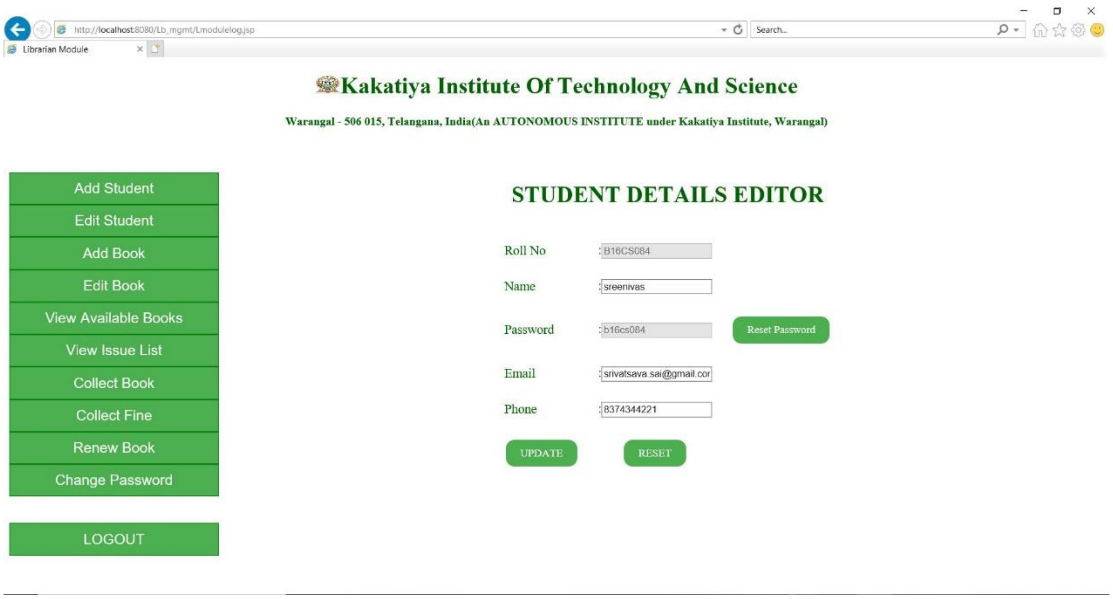
Kakatiya Institute Of Technology And Science

Warangal - 506 015, Telangana, India(An AUTONOMOUS INSTITUTE under Kakatiya Institute, Warangal)

STUDENT REGISTRATION

Roll no	:
Name	:
Password	:
Email	:
Phone	:

Librarian module edit student page:



http://localhost:8080/Lb_mgmt/Lmodulelog.jsp

Librarian Module

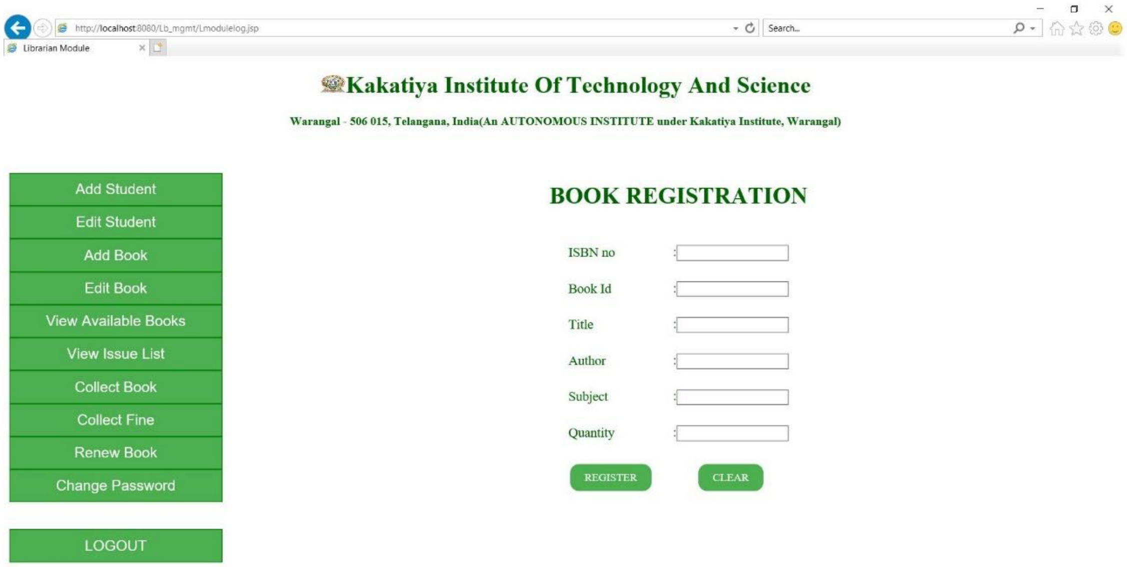
Kakatiya Institute Of Technology And Science

Warangal - 506 015, Telangana, India(An AUTONOMOUS INSTITUTE under Kakatiya Institute, Warangal)

STUDENT DETAILS EDITOR

Roll No	: B16CS084
Name	: sreenivas
Password	: b16cs084
Email	: srivatsava.sai@gmail.cor
Phone	: 8374344221

Librarian module add book page:



http://localhost:8080/Lb_mgmt/Lmodulelog.jsp

Librarian Module

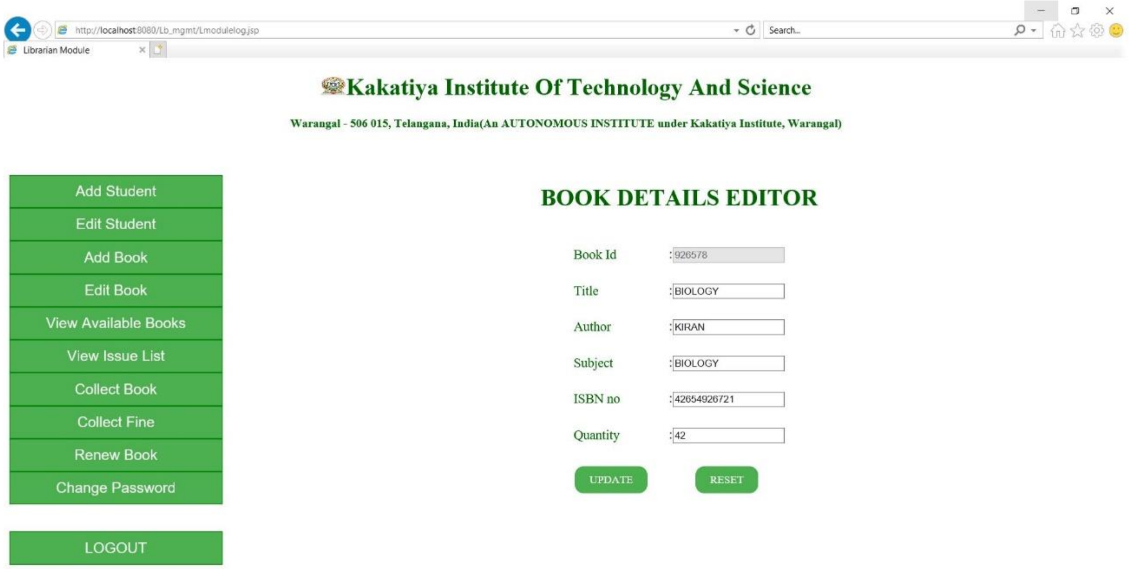
Kakatiya Institute Of Technology And Science

Warangal - 506 015, Telangana, India(An AUTONOMOUS INSTITUTE under Kakatiya Institute, Warangal)

BOOK REGISTRATION

ISBN no	:
Book Id	:
Title	:
Author	:
Subject	:
Quantity	:

Librarian module edit book page:



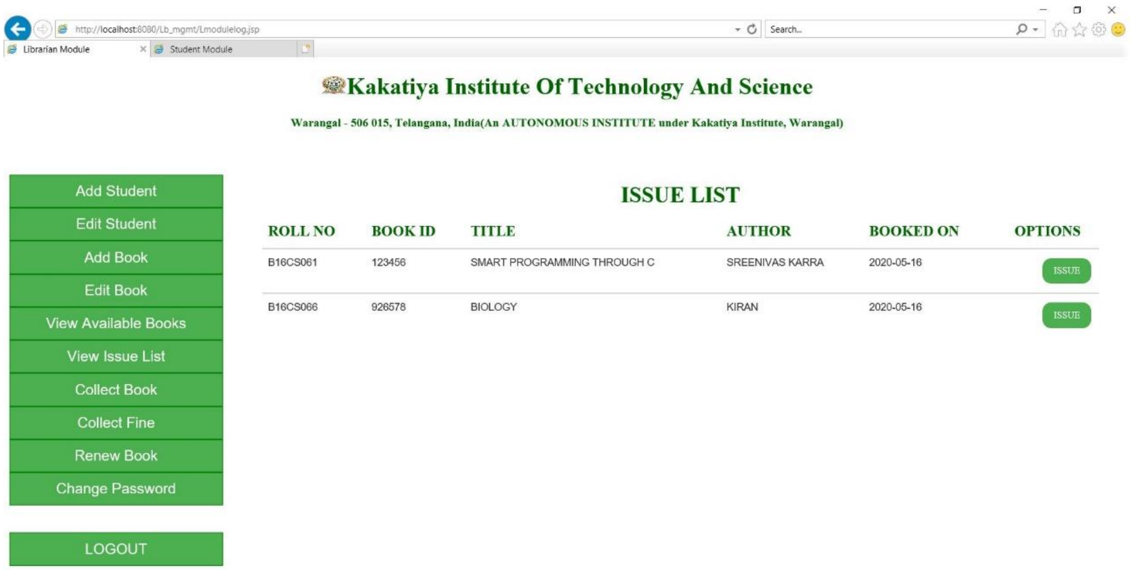
Kakatiya Institute Of Technology And Science

Warangal - 506 015, Telangana, India(An AUTONOMOUS INSTITUTE under Kakatiya Institute, Warangal)

BOOK DETAILS EDITOR

Book Id	:926578
Title	:BIOLOGY
Author	:KIRAN
Subject	:BIOLOGY
ISBN no	:42654926721
Quantity	:42

Librarian module issue list page:



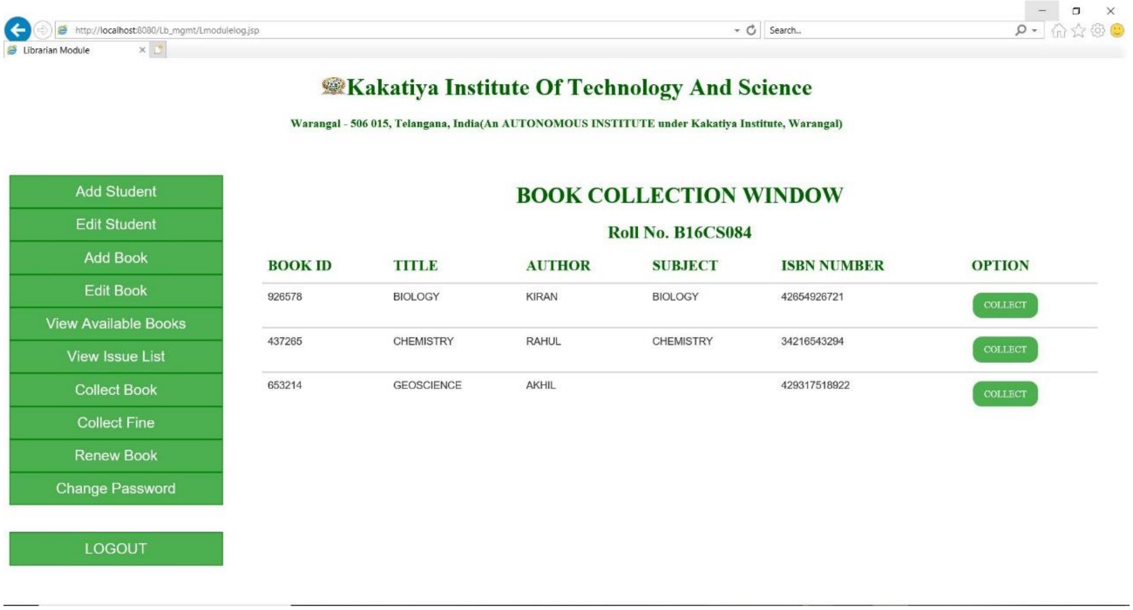
Kakatiya Institute Of Technology And Science

Warangal - 506 015, Telangana, India(An AUTONOMOUS INSTITUTE under Kakatiya Institute, Warangal)

ISSUE LIST

ROLL NO	BOOK ID	TITLE	AUTHOR	BOOKED ON	OPTIONS
B16CS061	123456	SMART PROGRAMMING THROUGH C	SREENIVAS KARRA	2020-05-16	ISSUE
B16CS066	926578	BIOLOGY	KIRAN	2020-05-16	ISSUE

Librarian module collect book page:



Kakatiya Institute Of Technology And Science

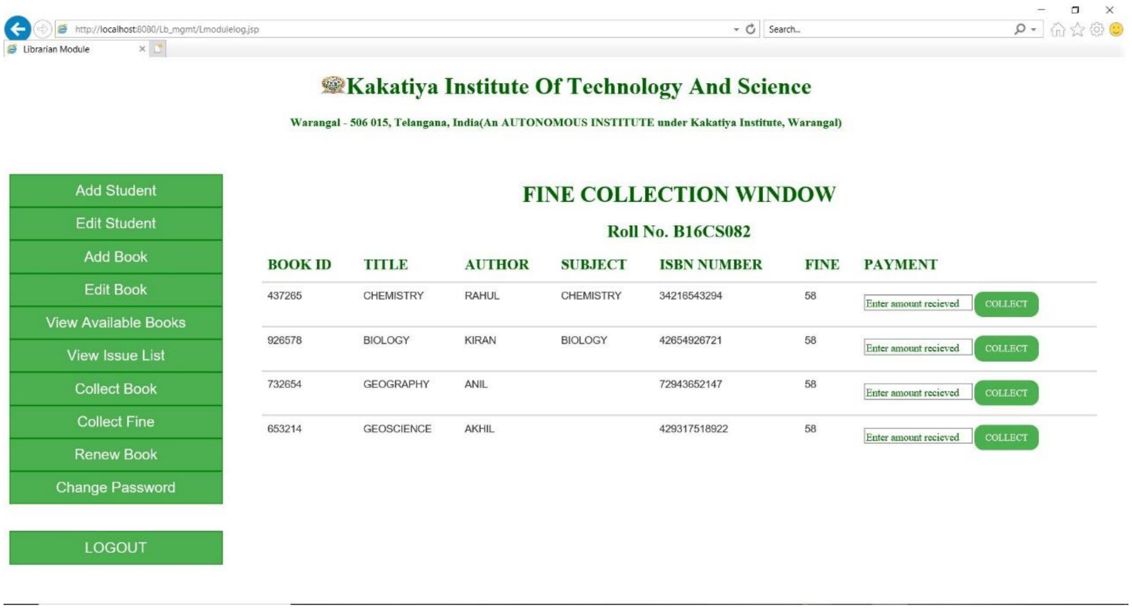
Warangal - 506 015, Telangana, India(An AUTONOMOUS INSTITUTE under Kakatiya Institute, Warangal)

BOOK COLLECTION WINDOW

Roll No. B16CS084

BOOK ID	TITLE	AUTHOR	SUBJECT	ISBN NUMBER	OPTION
926578	BIOLOGY	KIRAN	BIOLOGY	42654926721	COLLECT
437265	CHEMISTRY	RAHUL	CHEMISTRY	34216543294	COLLECT
653214	GEOSCIENCE	AKHIL		429317518922	COLLECT

Librarian module collect fine page:



Kakatiya Institute Of Technology And Science

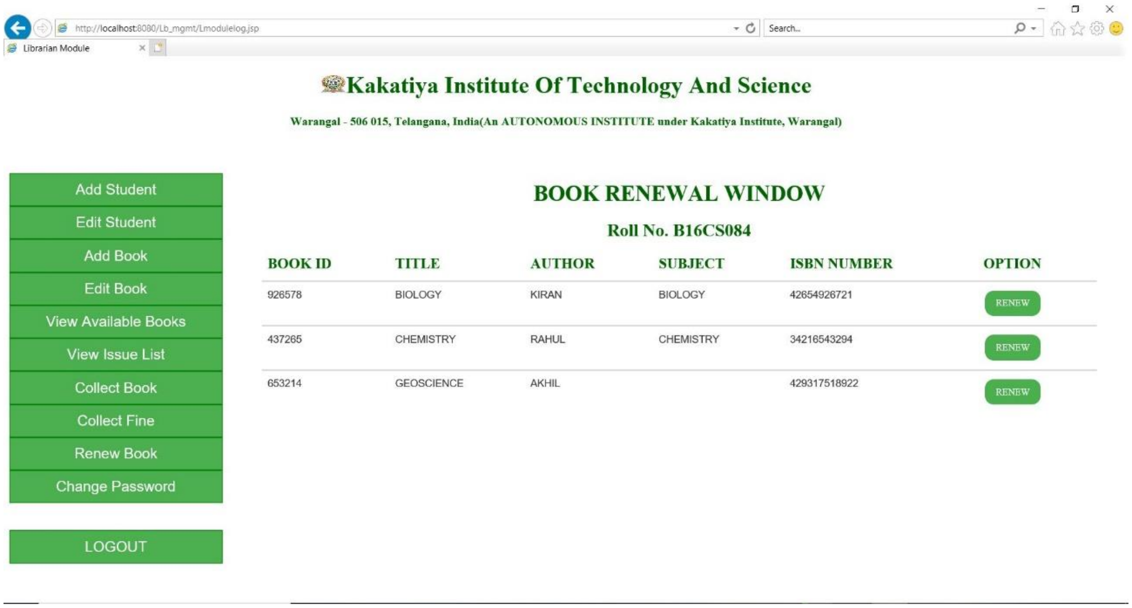
Warangal - 506 015, Telangana, India(An AUTONOMOUS INSTITUTE under Kakatiya Institute, Warangal)

FINE COLLECTION WINDOW

Roll No. B16CS082

BOOK ID	TITLE	AUTHOR	SUBJECT	ISBN NUMBER	FINE	PAYMENT
437265	CHEMISTRY	RAHUL	CHEMISTRY	34216543294	58	Enter amount recieved COLLECT
926578	BIOLOGY	KIRAN	BIOLOGY	42654926721	58	Enter amount recieved COLLECT
732654	GEOGRAPHY	ANIL		72943652147	58	Enter amount recieved COLLECT
653214	GEOSCIENCE	AKHIL		429317518922	58	Enter amount recieved COLLECT

Librarian module renew book page:



Kakatiya Institute Of Technology And Science

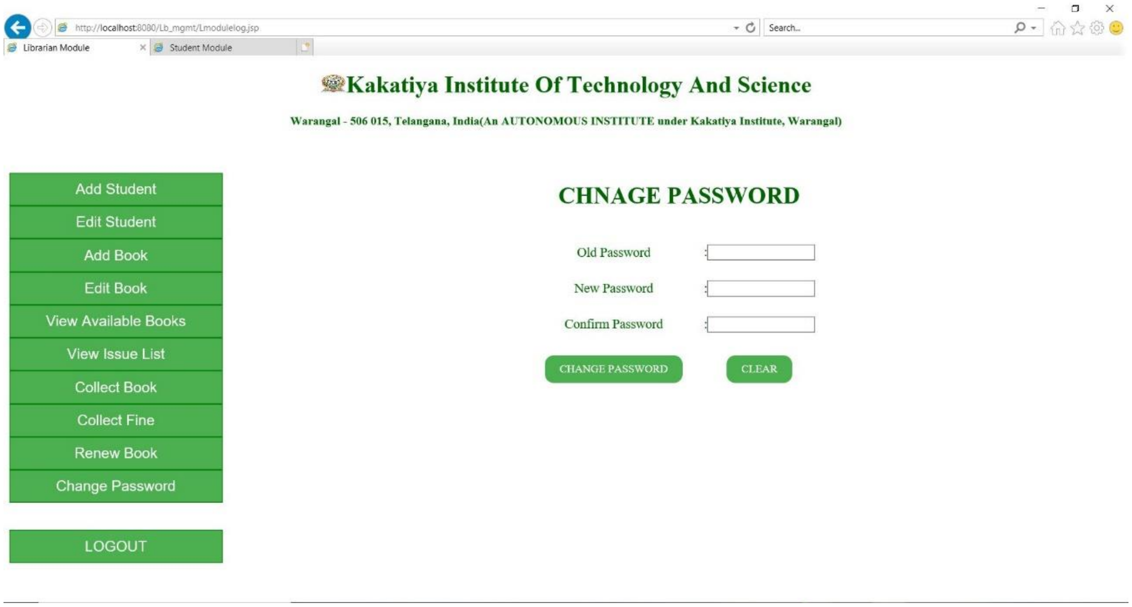
Warangal - 506 015, Telangana, India(An AUTONOMOUS INSTITUTE under Kakatiya Institute, Warangal)

BOOK RENEWAL WINDOW

Roll No. B16CS084

BOOK ID	TITLE	AUTHOR	SUBJECT	ISBN NUMBER	OPTION
926578	BIOLOGY	KIRAN	BIOLOGY	42654926721	RENEW
437265	CHEMISTRY	RAHUL	CHEMISTRY	34216543294	RENEW
653214	GEOSCIENCE	AKHIL		429317518922	RENEW

Librarian module change password page:



Kakatiya Institute Of Technology And Science

Warangal - 506 015, Telangana, India(An AUTONOMOUS INSTITUTE under Kakatiya Institute, Warangal)

CHNAGE PASSWORD

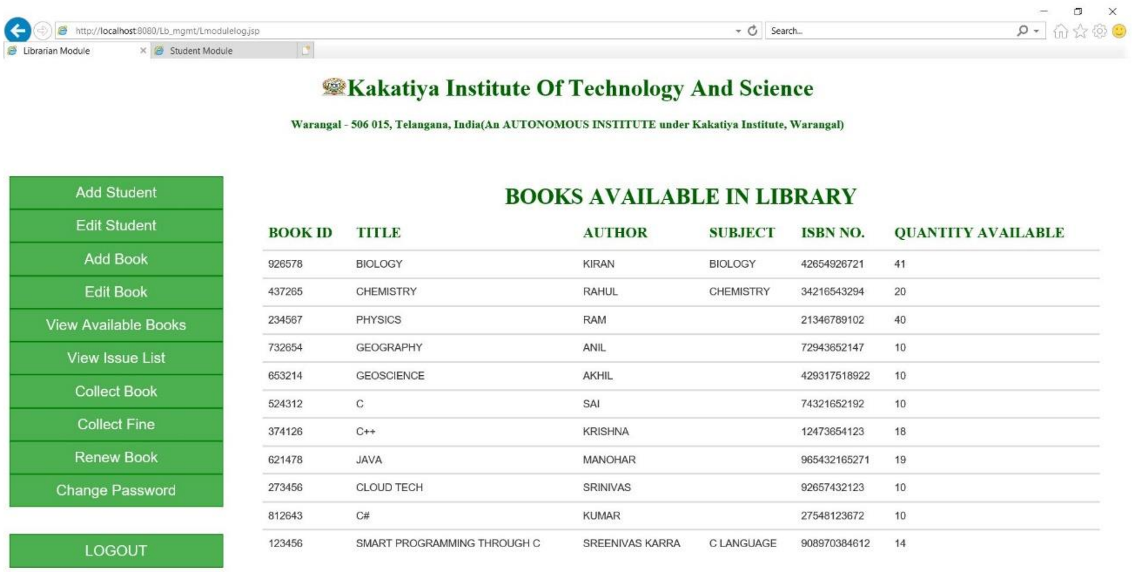
Old Password :

New Password :

Confirm Password :

The above change password page for both the modules is same.

Librarian module view all available books page:



Kakatiya Institute Of Technology And Science

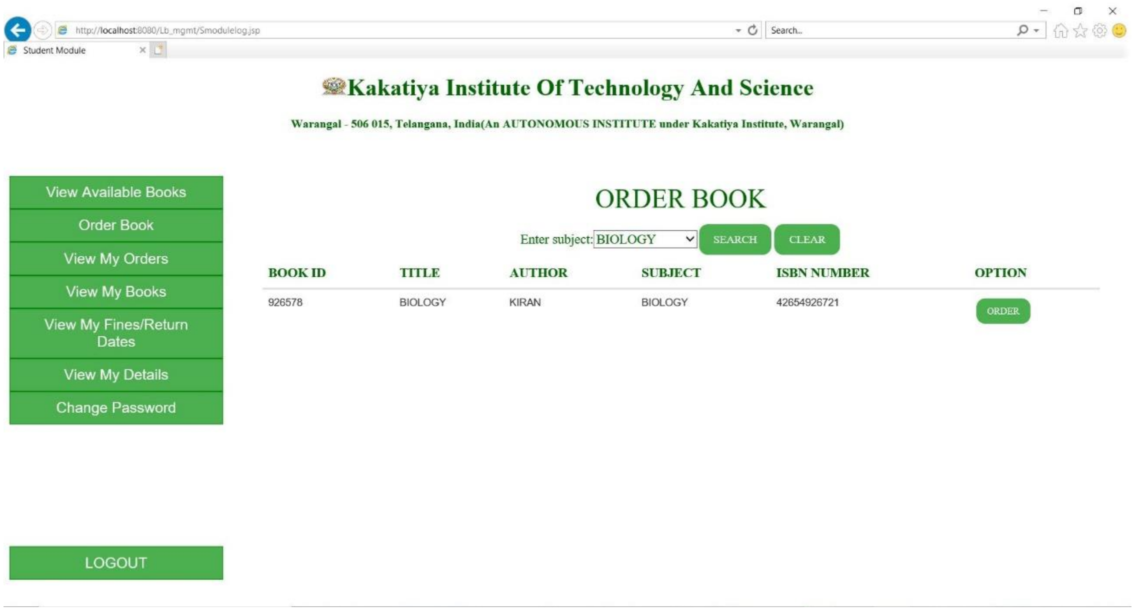
Warangal - 506 015, Telangana, India(An AUTONOMOUS INSTITUTE under Kakatiya Institute, Warangal)

BOOKS AVAILABLE IN LIBRARY

BOOK ID	TITLE	AUTHOR	SUBJECT	ISBN NO.	QUANTITY AVAILABLE
926578	BIOLOGY	KIRAN	BIOLOGY	42654926721	41
437265	CHEMISTRY	RAHUL	CHEMISTRY	34216543294	20
234567	PHYSICS	RAM		21346789102	40
732654	GEOGRAPHY	ANIL		72943652147	10
653214	GEOSCIENCE	AKHIL		429317518922	10
524312	C	SAI		74321652192	10
374126	C++	KRISHNA		12473654123	18
621478	JAVA	MANOHAR		965432165271	19
273456	CLOUD TECH	SRINIVAS		92657432123	10
812643	C#	KUMAR		27548123672	10
123456	SMART PROGRAMMING THROUGH C	SREENIVAS KARRA	C LANGUAGE	908970384612	14

The above page is same in both librarian and student modules.

Student module order book page:



Kakatiya Institute Of Technology And Science

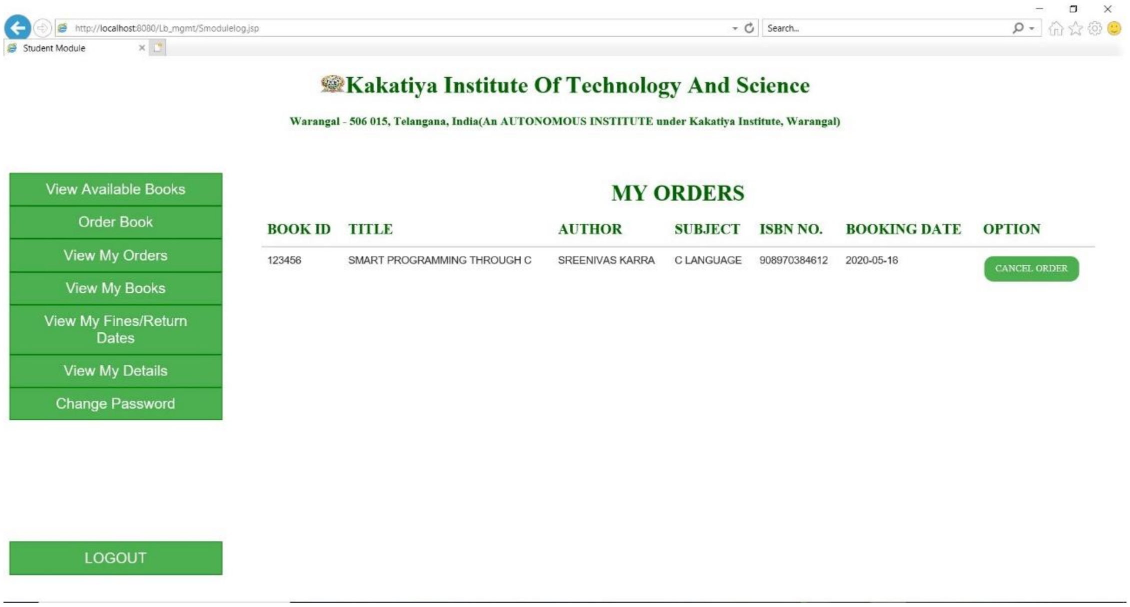
Warangal - 506 015, Telangana, India(An AUTONOMOUS INSTITUTE under Kakatiya Institute, Warangal)

ORDER BOOK

Enter subject: BIOLOGY

BOOK ID	TITLE	AUTHOR	SUBJECT	ISBN NUMBER	OPTION
926578	BIOLOGY	KIRAN	BIOLOGY	42654926721	ORDER

Student module view my orders page:



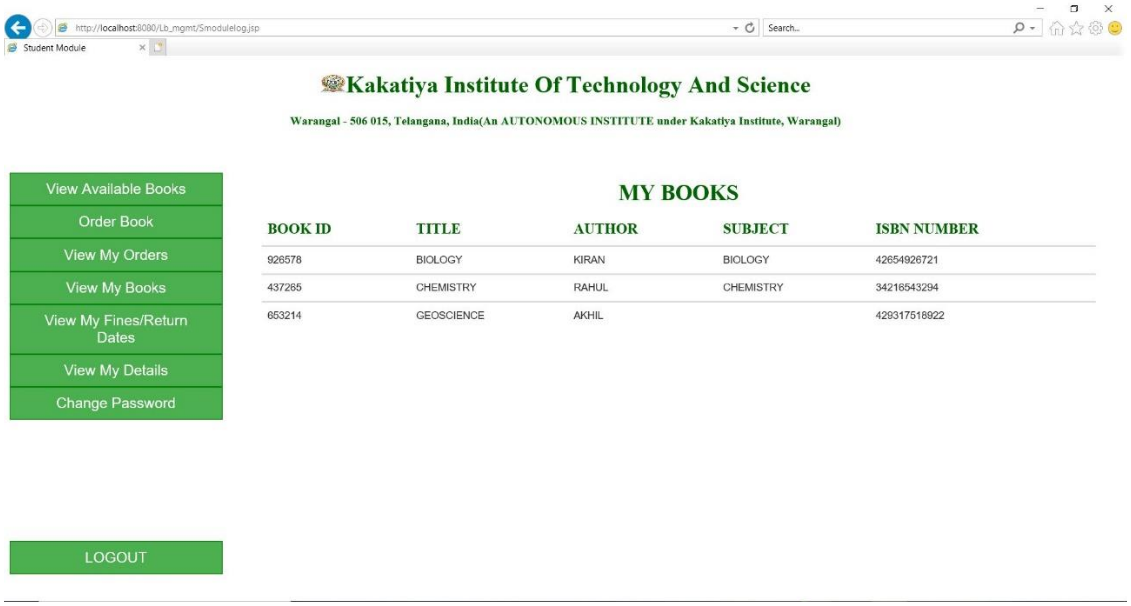
Kakatiya Institute Of Technology And Science

Warangal - 506 015, Telangana, India(An AUTONOMOUS INSTITUTE under Kakatiya Institute, Warangal)

MY ORDERS

BOOK ID	TITLE	AUTHOR	SUBJECT	ISBN NO.	BOOKING DATE	OPTION
123456	SMART PROGRAMMING THROUGH C	SREENIVAS KARRA	C LANGUAGE	908970384612	2020-05-16	CANCEL ORDER

Student module view my books page:



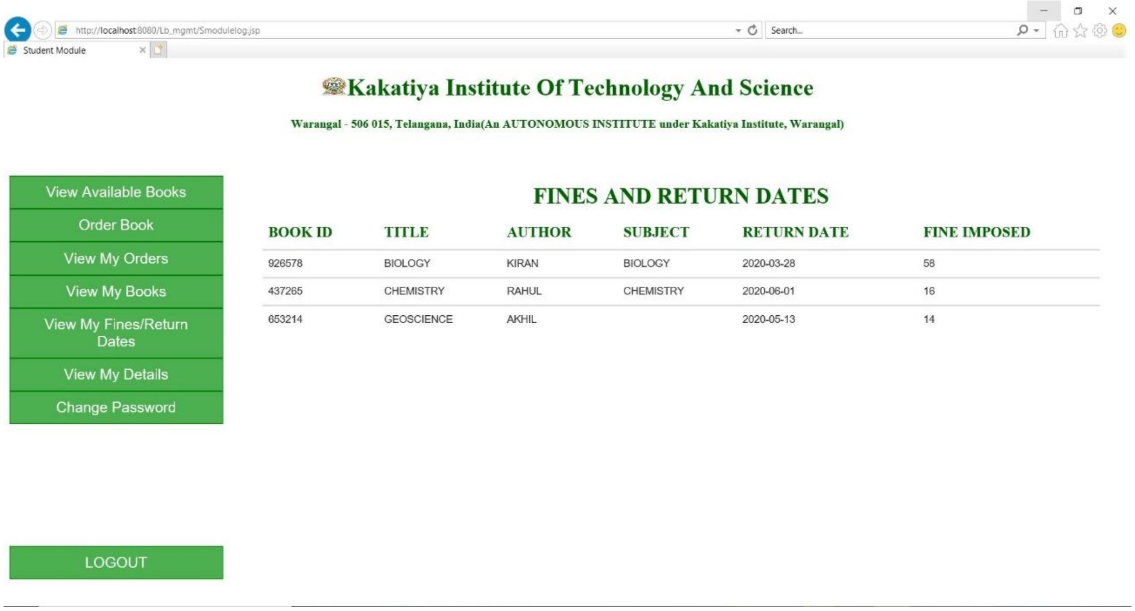
Kakatiya Institute Of Technology And Science

Warangal - 506 015, Telangana, India(An AUTONOMOUS INSTITUTE under Kakatiya Institute, Warangal)

MY BOOKS

BOOK ID	TITLE	AUTHOR	SUBJECT	ISBN NUMBER
926578	BIOLOGY	KIRAN	BIOLOGY	42654926721
437265	CHEMISTRY	RAHUL	CHEMISTRY	34216543294
653214	GEOSCIENCE	AKHIL		429317518922

Student module view my fines/return dates page:



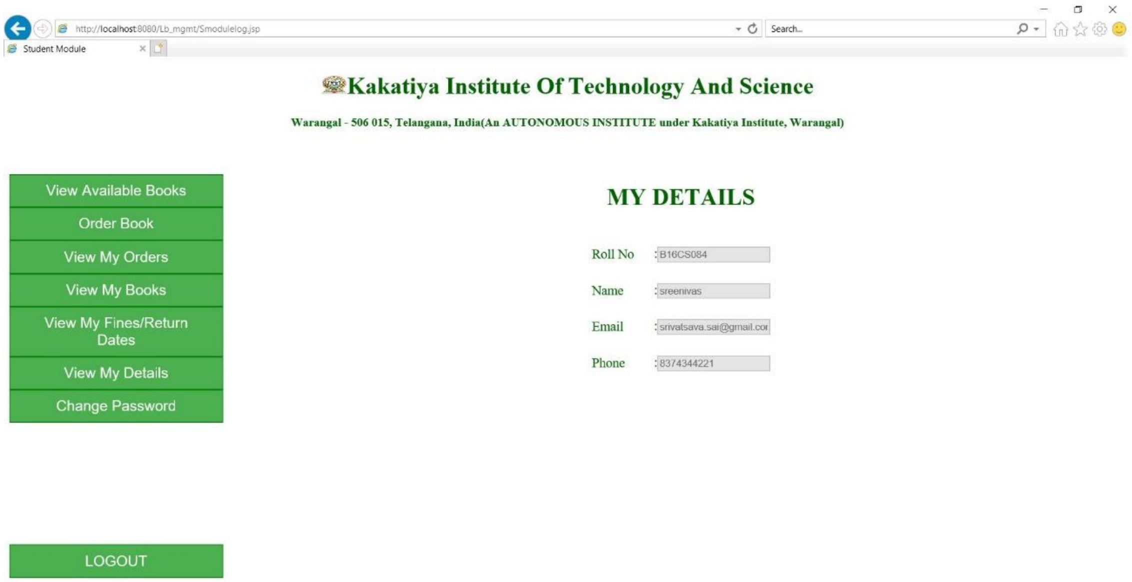
Kakatiya Institute Of Technology And Science

Warangal - 506 015, Telangana, India(An AUTONOMOUS INSTITUTE under Kakatiya Institute, Warangal)

FINES AND RETURN DATES

BOOK ID	TITLE	AUTHOR	SUBJECT	RETURN DATE	FINE IMPOSED
926578	BIOLOGY	KIRAN	BIOLOGY	2020-03-28	58
437265	CHEMISTRY	RAHUL	CHEMISTRY	2020-06-01	16
653214	GEOSCIENCE	AKHIL		2020-05-13	14

Student module view my details page:



Kakatiya Institute Of Technology And Science

Warangal - 506 015, Telangana, India(An AUTONOMOUS INSTITUTE under Kakatiya Institute, Warangal)

MY DETAILS

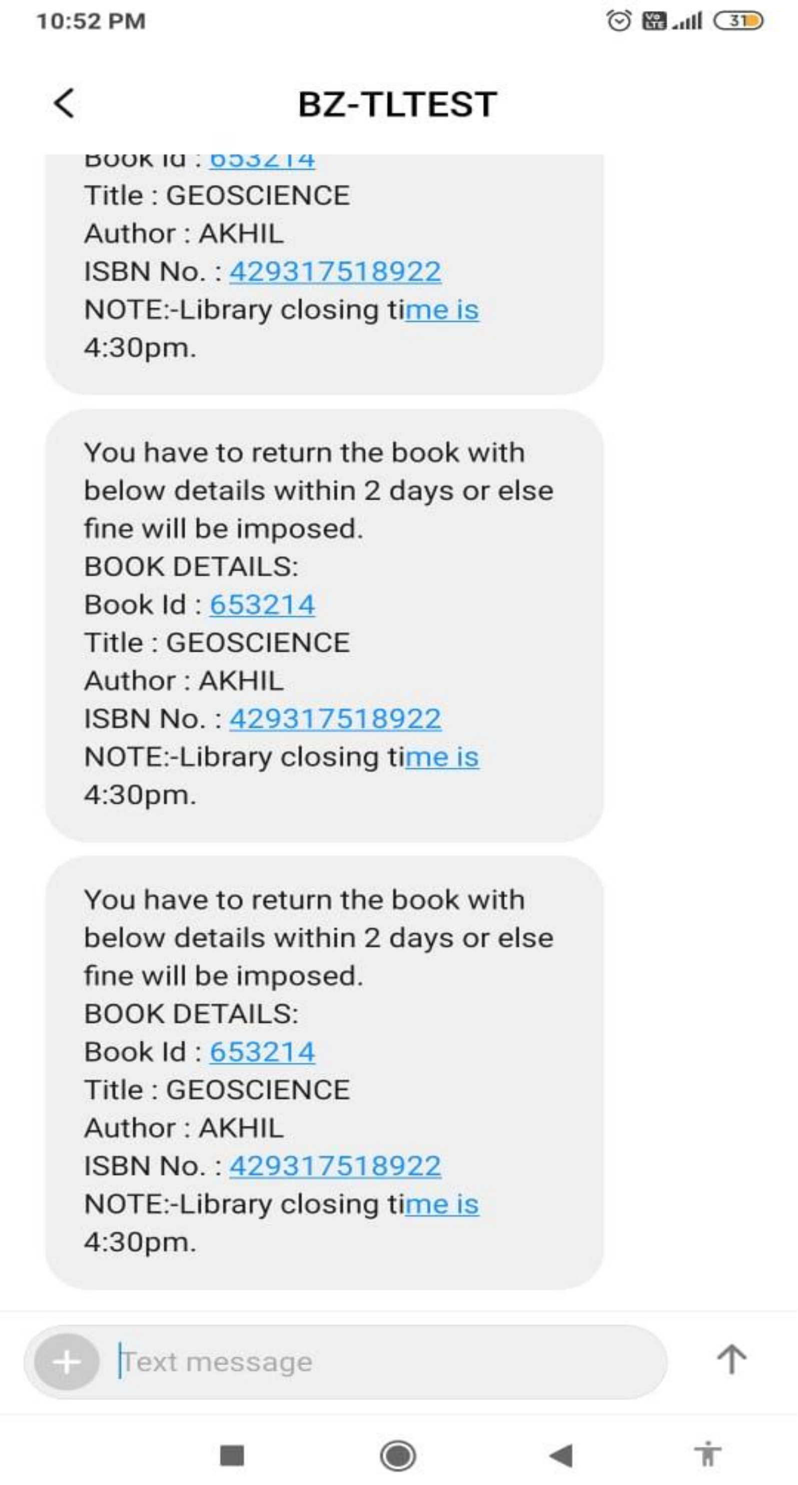
Roll No : B16CS084

Name : sreenivas

Email : srivatsava.sai@gmail.cor

Phone : 8374344221

Screenshot of warning SMS received by student:



Book Id : 653214
Title : GEOSCIENCE
Author : AKHIL
ISBN No. : 429317518922
NOTE:-Library closing time is
4:30pm.

You have to return the book with below details within 2 days or else fine will be imposed.
BOOK DETAILS:
Book Id : 653214
Title : GEOSCIENCE
Author : AKHIL
ISBN No. : 429317518922
NOTE:-Library closing time is
4:30pm.

You have to return the book with below details within 2 days or else fine will be imposed.
BOOK DETAILS:
Book Id : 653214
Title : GEOSCIENCE
Author : AKHIL
ISBN No. : 429317518922
NOTE:-Library closing time is
4:30pm.

CHAPTER 7 - CONCLUSION

7. CONCLUSION

It has been a great pleasure for our team to work on this exciting project. This project is helpful to learn about the practical knowledge of web applications. It also provides knowledge about the latest technology used in developing web applications and client-server technology that is having a great demand at present and will have the same demand in future. This will provide better opportunities and guidance in future for developing projects independently.

Benefits:

The project is identified by the merits of the system offered to the user. Some of the merits are:-

- It is a web application which can be accessed from anywhere using internet.
- The application offers user to enter data through simple and interactive interface.
- This application guides the user by providing various messages.
- From every part of the project the user is provided with buttons through framing so that he can move from one option to another as per his/her requirement. We can say that the project is user friendly, which is one of the primary concerns of a good project.
- Data storage and retrieval will become faster and easier because the data is stored in a systematic manner and in a single database.
- Decision making process would be greatly enhanced because of faster processing of information.
- This project also has an additional facility i.e., It sends alert messages to users mobile phones via SMS.
- Through these features it will increase the efficiency, accuracy and transparency of the application.

CHAPTER 8 – FUTURE SCOPE

8. FUTURE SCOPE

Library management system with SMS autoreply is a complete automated system, though it has few limitations but it has a lot of future scope that can be added to make it widely acceptable.

One of the future scopes is to add the mechanism of sending emails to users so that the user can get various information using his email. Another functionality that can be added is that the facility of getting information by sending SMS to the application.

Reviews, ratings and comments on different books can also be incorporated. This would help the user to go through reviews on different books and select the best possible book available. We can also add the feature of suggestions of books based on previous searches.

CHAPTER 9 - BIBLIOGRAPHY

9. BIBLIOGRAPHY

Websites:

- www.w3schools.com/html
- www.w3schools.com/css
- www.w3schools.com/js
- www.tutorialspoint.com/jsp
- www.textlocal.in

Books:

- HTML & CSS by Jon Duckett
- JavaScript and jQuery by Jon Duckett
- Eloquent JavaScript, 3rd Edition by Marijn Haverbeke