ONLINE LIBRARY MANAGEMENT SYSTEM

A mini project report submitted
for partial fulfillment for the award of the degree

B-Tech in computer science & engineering

SUBMITTED BY:

Tirri Gowtham (Team Lead)	- 319129510062
Vurukuti Sai Goutham	- 319129510049
Jampana Sai Ganesh Varma	- 319129510021
Tekkepati Navya Sree	- 319129510047

Under the Esteemed Guidance of

Mrs. T. ANUSHA

Asst. Prof. Department of **CSE , WISTM.**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
WELLFARE INSTITUTE OF SCIENCE TECHNOLOGY AND MANAGEMENT**

**(Approved AICTE ,DELHI&affiliated to Andhra University) Pinagadi
Pendurthi(M),Visakhapatnam-531173**

WELLFARE INSTITUTE OF SCIENCE TECHNOLOGY AND MANAGEMENT

Approved AICTE ,DELHI&affiliated to Andhra University(V),
Pendurthi(M),Visakhapatnam-531173

CERTIFICATE

This is to certify that the mini project report entitled
”online library management system”

SUBMITTED BY:

Tirri Gowtham (Team Lead)	- 319129510062
Vurukuti Sai Goutham	- 319129510049
Jampana Sai Ganesh Varma	- 319129510021
Tekkepati Navya Sree	- 319129510047

For partial fulfillments for the award of the degree Bachelor of Technology in computer science and Engineering is a bonafide project work carried out by him .This work is not submitted any University for the award of any degree.

INTERNAL GUIDE

Mrs. T. ANUSHA
Assistant Professor**,**
Department of CSE **,**
WISTM , ENGG.COLLEGE

HEAD OF THE DEPARTMENT

Mrs.K.V.LAKSHMI,M.TECH
Assistant Professor,
Department of CSE **,**
WISTM , ENGG.COLLEGE

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people who made it possible and whose constant guidance and encouragement crown all the effort with success. I like to thank **prof. M. James Stephen**, Principal, of Wellfare Technology and Management , for this kind cooperation that gives me boundless pleasure to avail this opportunity to express my deep sense of gratitude and while hearted thanks to Head of Department **Mrs.K.V.Lakshmi, Assistant Professor,** department of computer science and Engineering and my project guide **Mrs.T.Anusha, Assistant Proffesor**, Department fo Computer Science and Engineering, and all the faculty of computer science department for their valuable suggestions and co-operaiton during the development of this project. I am really grateful to my parents and all my friends for their constructive suggestions to do this project in effective manner

SUBMITTED BY:

• Tirri Gowtham (Team Lead)	- 319129510062
• Vurukuti Sai Goutham	- 319129510049
• Jampana Sai Ganesh Varma	- 319129510021
• Tekkepati Navya Sree	- 319129510047

DECLARATION

I here declare that the project entitled "online library management system" has been done under the guidance of **Mrs. T.ANUSHSA Asst. Prof. Department of CSE** and is dissertation of my own work except where specifically ask to the contrary and is submitted to the department of computer science and Engineering, wellfare Insistute of science technology and management for the partial fulfilment of the requirement for the award of B.Tech degree.

SUBMITTED BY:

Tirri Gowtham (Team Lead)	- 319129510062
Vurukuti Sai Goutham	- 319129510049
Jampana Sai Ganesh Varma	- 319129510021
Tekkepati Navya Sree	- 319129510047

TABLE OF CONTENTS

ABSTRACT

Librarian usually performs their work manually. It takes a more time and the human effort is wasted in retrieving information, handling of huge data, generation of reports, updating of data, maintaining records etc.., For these problems we maintaining a library management system. It saves a lot of time to the librarian and the user.

The librarian can a keep track of all transaction of students and faculty information, and also librarian can add books information, student and faculty information, updating and delete the information, and as well as the librarian can keep track of book returning and issue the book.

Functionalities:

- Student/ faculty information are maintained.
- Add/ delete/ update books information.
- Add / delete/ update user’s information.
- Issue/ return books information.
- Fine calculation.
- Search book availability.

INTRODUCTION

Library Management System is an application which refers to library systems which are generally small or medium in size. It is used by librarian to manage the library record various transactions like issue of books, return of books, addition of new books, addition of new students etc. Books and student maintenance modules are also included in this system which would keep track of the students using the library and also a detailed description about the books a library contains. In addition, report module is also included in Library Management System. If user's position is admin, the user is able to generate different kinds of reports like lists of students registered, list of books, issue and return reports. All these modules are able to help librarian to manage the library.

PURPOSE:

- The purpose of this project is to provide a friendly environment to maintain the details of books and library members.
- The main purpose of this project is to maintain easy circulation system using computers and to provide different reports.
- Improved user service through greater access to accurate information.
- Due to computerized information it's reduced the risk of paper work such as file lost, file damaged and time consuming. It can help user to manage the transaction or record more effectively and time saving.

SYSTEM ANALYSIS

EXISTING SYSTEM:

System Analysis is a detailed study of the various operations performed by a system and their relationships within and outside of the system. Here the key question is- what all problems exist in the present system what must be done to solve the problem Analysis begins when a user or manager begins a study of the program using existing system. In our existing system all the transaction of books are done manually, So taking more time for a transaction like borrowing a book or returning a book and also for searching of members and books. Another major disadvantage is that to preparing the list of books borrowed and the available books in the library will take more time, currently it is doing as a one day process for verifying all records. So after conducting the feasibility study we decided to make the manual Library management system to be computerized.

PROPOSED SYSTEM:

Proposed system is an automated Library Management System. Through our software user can add members, add books, search members, search books, update information, edit information, borrow and return books in quick time. Our proposed system has the following advantages.

- User friendly interface
- Fast access to database
- Less error
- More Storage Capacity
- Search facility
- Look and Feel Environment

All the manual difficulties in managing the Library have been rectified by implementing computerization.

SYSTEM REQUIREMENTS

SOFTWARE

Front End:

- HTML
- CSS
- JAVA SCRIPT
- BOOTSTRAP
- JQUERY

Back End:

- PYTHON
- DJANGO

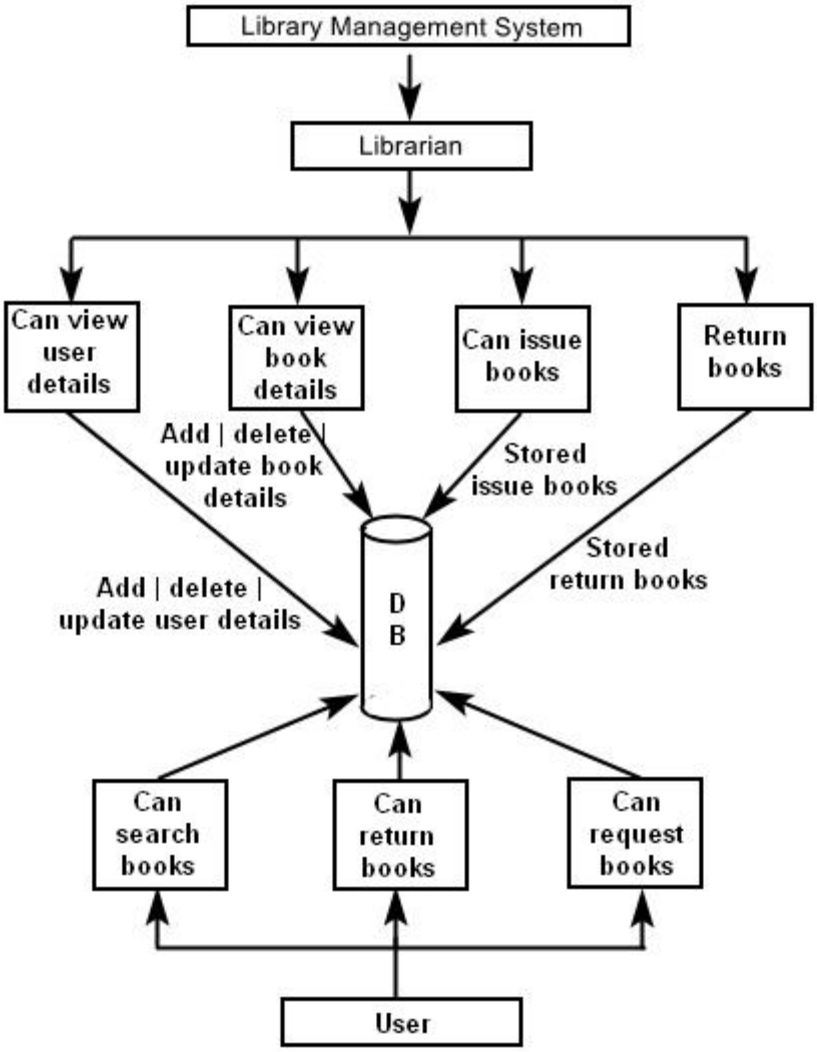
Server:

- DJANGO Server

HARDWARE

- Windows 10
- Intel i3 processor
- RAM 4GB
- External 500GB

SYSTEM ARCHITECTURE



TECHNOLOGIES STACK

- HTML
- CSS
- JAVA SCRIPT
- BOOTSTRAP
- JQUERY
- PYTHON
- DJANGO

HTML [Hyper Text Markup Language]:

- HTML stands for Hyper Text Markup Language.
- HTML describes the structure of web pages using markup.
- HTML elements are the building blocks of HTML pages.
- HTML elements are represented by tags.
- HTML tags label pieces of content such as heading, paragraph, table and so on.
- Browsers do not display the HTML tags, but use them to render the content of the page.

<!doctype html>: The <!doctype> is not an HTML tag. It is an instruction to the web browser about what version of html the pages is written it. This page is written in HTML5 as opposed to say HTML 4.01.

<html>: The <html> element is the root element of an HTML page.

<head>: The <head> element is a container for meta information about the document and it is placed between the <html> and <body> tags.

<title>: The <title> tag is required in all HTML documents and it specifies a title for the document.

<body>: The <body> element contains the visible page content.

<h1>: The <h1> element defines a large heading.

<h6>: The <h6> element defines a small heading.

<p>: The <p> element defines a paragraph.

<a>: The <a> element defines the HTML links. The link's destination is specified in the href attribute.

<img>: The <img> element defines the HTML images. The source file (src), alternative text (alt), width, and height are provided as attributes.

<button>: The <button> element defines a clickable button, inside a button you can put content.

<br>: The <br> element defines the line break. This tag is an empty tag, which means that it has no end tag.

<!-- -->: The <!-- --> element defines the comment tag.

<table>: The <table> element defines the HTML tables.

<tr>: The <tr> element defines the table row.

<th>: The <th> element defines the table header. By default, table headings are bold and centered.

<td>: The <td> element defines the table data/cell.

<div>: The <div> element defines the block level element. A block-level element always starts on a new line and takes up the full width available.

<form>: It defines a form that is used to collect user input.

<label>: The <label> element represents a caption for an item in a user interface.

<input>: The <input> element is the most important form element. The <input> element can be displayed in several ways, depending on the type attribute.

<ul>: The <ul> element to define an unordered list.

<ol>: The <ol> element to define an ordered list.

<li>: The <li> element to define a list item.

CSS [Cascading Style Sheet]:

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS handles the look and feel part of a web page. Using CSS, you can control the colour of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colours are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.

CSS can be added to HTML elements in 3 ways:

- **Inline** - by using the style attribute in HTML elements.
- **Internal** - by using a <style> element in the <head> section.
- **External** - by using an external CSS file.

Css Colors: Colors are specified using predefined color names, or RGB, HEX, HSL, RGBA, HSLA values.

- **Css Names:** In HTML, a color can be specified by using a color name.
- **Background Color**: You can set the background color for HTML elements.
- **Text Color:** You can set the color of text.
- **Border Color:** You can set the color of borders.
- **Color Values:** In HTML, colors can also be specified using RGB values, HEX values, HSL values, RGBA values, and HSLA values.

Css Backgrounds: The CSS background properties are used to define the background effects for elements.

- **Background color:** The background-color property specifies the background color of an element.
- **Background Image:** The background-image property specifies an image to use as the background of an element. By default, the image is repeated so it covers the entire element.
- **Background Repeat:** The background-repeat property specifies a background image will be repeated.
- **Background Attachment:** The background-attachment property specifies a background image is fixed or scrolls with the rest of the page.
- **Background Position:** The background-position property specifies a starting position of a background image.

Css Borders:

- **Css Border Properties:** The CSS border properties allow you to specify the style, width, and color of an element's border.
- **Border Style:** The Border-style property specifies what kind of border to display.

- **Border Width:** The border-width property specifies the width of the four borders. The width can be set as a specific size (in px, pt, cm, em, etc) or by using one of the three pre-defined values: thin, medium, or thick.
- **Border Color:** The border-color property is used to set the color of the four borders.

Css Margins: The CSS margin properties are used to create space around elements, outside of any defined borders.

- **Margin-Individual Sides:** The CSS margin properties are used to create space around elements, outside of any defined borders.
- **Margin-Shorthand Property:** To shorten the code, it is possible to specify all the margin properties in one property.
- **Margin Collapse:** Top and bottom margins of elements are sometimes collapsed into a single margin that is equal to the largest of the two margins. This does not happen on left and right margins! Only top and bottom margins.

Css Padding: The CSS padding properties are used to generate space around an element's content, inside of any defined borders.

- **Padding-Individual sides:** CSS has properties for specifying the padding for each side of an elements are padding-top, padding-botom, padding left, padding-right.
- **Padding - Shorthand Property:** To shorten the code, it is possible to specify all the padding properties in one property.

CSS Height and Width:

- **Setting max-width:** The max-width property is used to set the maximum width of an element. The max-width can be specified in length values, like px, cm, etc., or in percent (%) of the containing block, or set to none (this is default. Means that there is no maximum width).

CSS Box Model: All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout.

The CSS box model is essentially a box that wraps around every HTML element. It consists of margins, borders, padding, and the actual content.

- **Content -** The content of the box, where text and images appear
- **Padding -** Clears an area around the content. The padding is transparent
- **Border -** A border that goes around the padding and content
- **Margin -** Clears an area outside the border. The margin is transparent

The box model allows us to add a border around elements, and to define space between elements.

CSS Outline: An outline is a line that is drawn around elements, OUTSIDE the borders, to make the element "stand out".

- **Outline Style:** The outline-style property specifies the style of the outline, and can have one of the following values like dotted, dashed, solid, double etc.
- **Outline Color:** The outline-color property is used to set the color of the outline.
- **Outline Width:** The outline-width property specifies the width of the outline, and can have one of the following values:
    - thin (typically 1px)
    - medium (typically 3px)
    - thick (typically 5px)
    - A specific size (in px, pt, cm, em, etc)
- **Outline Offset:** The outline-offset property adds space between an outline and the edge/border of an element. The space between an element and its outline is transparent.

CSS Text:

- **Text Color:** The color property is used to set the color of the text.
- **Text Alignment:** The text-align property is used to set the horizontal alignment of a text. A text can be left or right aligned, centered, or justified.
- **Text Decoration:** The text-decoration property is used to set or remove decorations from text. The value text-decoration: none; is often used to remove underlines from link.
- **Text Transformation:** The text-transform property is used to specify uppercase and lowercase letters in a text. It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word.
- **Text Indentation:** The text-indent property is used to specify the indentation of the first line of a text.

- **Letter Spacing:** The letter-spacing property is used to specify the space between the characters in a text.
- **Line Height:** The line-height property is used to specify the space between lines.
- **Text Direction:** The direction property is used to change the text direction of an element.
- **Word Spacing:** The word-spacing property is used to specify the space between the words in a text.

CSS Font Families:

In CSS, there are two types of font family names:

- **generic family** - a group of font families with a similar look (like "Serif" or "Monospace")
- **font family** - a specific font family (like "Times New Roman" or "Arial"). The font family of a text is set with the font-family property. The font-family property should hold several font names as a "fallback" system. If the browser does not support the first font, it tries the next font, and so on.
- **Font Style:** The font-style property is mostly used to specify italic text. This property has three values:
- **normal** - The text is shown normally.
- **italic** - The text is shown in italics.
- **oblique** - The text is "leaning" (oblique is very similar to italic, but less supported)
- **Font Size:** The font-size property sets the size of the text. Being able to manage the text size is important in web design. However, you should not use font size adjustments to make paragraphs look like headings, or headings look like paragraphs.

Css Links: Links can be styled differently depending on what **state** they are in.

The four links states are:

- a:link - a normal, unvisited link
- a:visited - a link the user has visited
- a:hover - a link when the user mouses over it
- a:active - a link the moment it is clicked

JAVA SCRIPT:

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

- JavaScript is a lightweight, interpreted programming language.
- Designed for creating network-centric applications.
- Complementary to and integrated with Java.
- Complementary to and integrated with HTML.
- Open and cross-platform.

JavaScript Can Change HTML Content: One of many JavaScript HTML methods is getElementById().

<script>Tag: In HTML, JavaScript code must be inserted
between <script> and </script> tags.

JavaScript in <head> or <body>: You can place any number of scripts in an HTML document. Scripts can be placed in the <body>, or in the <head> section of an HTML page, or in both.

External JavaScript:

External scripts are practical when the same code is used in many different web pages.

JavaScript files have the file extension **.js**.

To use an external script, put the name of the script file in the src (source) attribute of a <script> tag:

JavaScript Display Possibilities:

JavaScript can "display" data in different ways:

- Writing into an HTML element, using innerHTML.
- Writing into the HTML output using document.write().
- Writing into an alert box, using window.alert().
- Writing into the browser console, using console.log().

Using innerHTML:

To access an HTML element, JavaScript can use the document.getElementById(id) method. The id attribute defines the HTML element. The innerHTML property defines the HTML content.

Using window.alert():: You can use an alert box to display data.

Using console.log(): For debugging purposes, you can use the console.log() method to display data.

JavaScript White Space: JavaScript ignores multiple spaces. You can add white space to your script to make it more readable.

JavaScript Code Blocks: JavaScript statements can be grouped together in code blocks, inside curly brackets {...}. The purpose of code blocks is to define statements to be executed together.

JavaScript Values: The JavaScript syntax defines two types of values: Fixed values and variable values. Fixed values are called **literals**. Variable values are called **variables**.

JavaScript Variables: In a programming language, **variables** are used to **store** data values. JavaScript uses the var keyword to **declare** variables. An **equal sign** is used to **assign values** to variables.

JavaScript Identifiers:

All JavaScript **variables** must be **identified** with **unique names**.

These unique names are called **identifiers**.

Identifiers can be short names (like x and y) or more descriptive names (age, sum, totalVolume).

The general rules for constructing names for variables (unique identifiers) are:

- Names can contain letters, digits, underscores, and dollar signs.
- Names must begin with a letter
- Names can also begin with $ and _ (but we will not use it in this tutorial)
- Names are case sensitive (y and Y are different variables)
- Reserved words (like JavaScript keywords) cannot be used as names

JavaScript Functions:

A JavaScript function is a block of code designed to perform a particular task.

A JavaScript function is executed when "something" invokes it.

Function Invocation:

The code inside the function will execute when "something" **invokes** (calls) the function:

- When an event occurs (when a user clicks a button)
- When it is invoked (called) from JavaScript code
- Automatically (self invoked)

Function Return:

When JavaScript reaches a return statement, the function will stop executing.

If the function was invoked from a statement, JavaScript will "return" to execute the code after the invoking statement.

JavaScript Events:

HTML events are **"things"** that happen to HTML elements.

When JavaScript is used in HTML pages, JavaScript can **"react"** on these events.

HTML Events:

An HTML event can be something the browser does, or something a user does.

Here are some examples of HTML events:

- An HTML web page has finished loading
- An HTML input field was changed
- An HTML button was clicked

Often, when events happen, you may want to do something. JavaScript lets you execute code when events are detected.

HTML allows event handler attributes, **with JavaScript code**, to be added to HTML elements.

- **Onchange-** An HTML element has been changed.
- **Onclick-** The user clicks an HTML element.
- **Onmouseover-** The user moves the mouse over an HTML element.
- **Onmouseout-** The user moves the mouse away from an HTML element.
- **Onleydown-** The user pushes a keyboard key.
- **Onload-** The browser has finished loading the page.

BOOTSTRAP:

Bootstrap is a free and open source front end development framework for the creation of websites and web apps. The Bootstrap framework is built on HTML, CSS and JavaScript (JS) to facilitate the development of responsive.

.align : A set of utility classes that are equivalent to writing the css property. You can use this on inline and table cell elements.

.align-content : Added to the parent flexbox container to determining how the elements are aligned horizontally.

.align-items : Class added to flexbox child items to specify if it should align towards the top or bottom of the container (start, end).

.bg : Background color utility classes.

.border : A versatile border utility class that lets you add/remove borders on a side or change a border color.

.btn-outline: A button variation to have outlined buttons instead of a solid background.

.carousel-control : When you have an image carousel with pagination you will use this class on the previous and next anchor links.

.carousel-fade : Animates the slide transition with a crossfade instead of a slide.

.form-control-plaintext : Use the class to remove the default form field styling and preserve the correct margin and padding.

.form-row : Works similar to a grid. but is more compact to make the form look more uniform.

.h-* : Height utility class that makes the element a percentage height of its parent element.

.justify-content-*-* : Class specifies where the flex items will be positioned inside the container.

.**nav-fill :** Makes all nav items use all available horizontal space. Nav items are different widths based on their content.

.nav-justified : Makes all nav items equal width and use all available horizontal space.

.navbar-collapse : The nav links that are collapsed and shown when toggled on mobile widths.

.navbar-expand-* : Since the navbar is displayed collapse on mobile first, this class specifies what breakpoint you want the navbar to not be collapsed.

.navbar-text : Vertically centres text inside a navbar.

.navbar-toggler-icon : The cheeseburger navigation icon is set using an svg background image of three horizontal lines.

.btn-group-lg : Increases the default button group size.

.btn-group-sm : Decreases the default button group size.

.btn-group-toggle : This class replaces an input checkbox with a custom style that is toggable on click.

.btn-outline-* : Transparent background with colored text and boarder.In bootstrap there is some button styles.There are

.btn,.btn-default,.btn-primary,.btn-success,.btn-info,.btn-warning,.btn-danger,.btn-link.

.carousel-item : The wrapper class applied to each individual carousel item.

.col-form-label : Class added to form labels to apply consistent padding and margins

.display-* : This set of classes increases the font size of headings in 4 stages. These classes are used for headings outside of the main content of the page like jumbotrons and page headers. Append (1-4) to the end to adjust size.

.dropdown-item : This class is added to each link item shown in a dropdown menu.

.dropdown-toggle-split : Removes the interactivity from a dropdown so it does not appear clickable.

.font-* : italic, weight-bold, weight-light, weight-normal, monospace.

.form-inline : Use this class to have a series of labels and form elements on a single horizontal row.

.form-text : This class is used for help text alongside form elements. You can add .text-muted to make the text lighter in color.

.input-group-text: This class adds the background color and text styles to the text inside an input group.

.nav-item : If your nav uses a list add this class to each list item for the proper spacing.

.nav-link : Each anchor link inside your nav is given this class in order to have the proper styling.

.nav-pills : Use this class along with .nav to make each nav link into a button.

.navbar-brand : Most navbars contain a logo or brand. This class is added to the anchor.

.navbar : Navigation header class.

.navbar-nav : The wrapper class of the navigation elements excluding the brand.

.navbar-light : Add this class to your navbar if you would like it to have a light background and dark text.

.navbar-toggler : The infamous cheeseburger icon to signify a navigation menu on mobile.

.position-* : Not responsive, but a group of utility classes to add common position values.Position may be absolute, fixed, relative, static, sticky.

.container : Fixed width container with widths determined by screen sites. Equal margin on the left and right

.container-fluid : Spans the full width of the screen.

.media : Media components are image heading and description text items. Blog comments, portfolio projects, album covers, etc.

.modal : The parent wrapper class of modal content.

.modal-body : The modal body content classes : Header - Body – Footer.

.modal-content : modal-content contains modal-body, modal-header, and modal-footer.

.model-dialog : The secondary wrapper class of the entire modal content.

.modal-header : The header section of the modal that contains the title and close button.

.modal-footer : The footer of the modal that contains action buttons or help text.

.tooltip: The Tooltip plugin is small pop-up box that appears when the user moves the mouse pointer over an element.

.popover: It is a pop-up box that appears when the user clicks on an element. The difference is that the popover can contain much more content.

.panel: A panel in bootstrap is a bordered box with some padding around its content. To color the panel, use contextual classes .panel-default, .panel-primary, .panel-success, .panel-info, .panel-warning, or .panel-danger.

. pagination: If you have a web site with lots of pages, you may wish to add some sort of pagination to each page.

. Jumbotron: A jumbotron indicates a big box for calling extra attention to some special content or information.

A jumbotron is displayed as a grey box with rounded corners. It also enlarges the font sizes of the text inside it.

.pager: Pager is also a form of pagination,Pager provides previous and next buttons (links).

. Glyphicons: Glyphicons can be used in text, buttons, toolbars, navigation, forms.Examples of glyphicons are Envelope glyphicon,Print glyphicon, Search glyphicon ,Download glyphicon.

JQUERY:

jQuery simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development. jQuery is a JavaScript toolkit designed to simplify various tasks by writing less code.

There are two versions of jQuery available for downloading:

- Production version - this is for your live website because it has been minified and compressed
- Development version - this is for testing and development (uncompressed and readable code)

Both versions can be downloaded from jQuery.com.

The jQuery library is a single JavaScript file, and you reference it with the HTML <script> tag (notice that the <script> tag should be inside the <head> section).

The jQuery syntax is tailor-made for **selecting** HTML elements and performing some **action** on the element(s).

Basic syntax is: **$(selector).action()**

- A $ sign to define/access jQuery
- A (*selector*) to "query (or find)" HTML elements
- A jQuery action() to be performed on the element(s)
- $(this).hide() - hides the current element.
- $("p").hide() - hides all <p> elements.
- $(".test").hide() - hides all elements with class="test".
- $("#test").hide() - hides the element with id="test".

jQuery Selectors:

jQuery selectors allow you to select and manipulate HTML element(s).

jQuery selectors are used to "find" (or select) HTML elements based on their name, id, classes, types, attributes, values of attributes and much more. It's based on the existing CSS Selectors, and in addition, it has some own custom selectors.

All selectors in jQuery start with the dollar sign and parentheses: $().

The element Selector:

The jQuery element selector selects elements based on the element name.

The #id Selector:

The jQuery **#id** selector uses the id attribute of an HTML tag to find the specific element.

An id should be unique within a page, so you should use the #id selector when you want to find a single, unique element.

The .class Selector:

The jQuery .class selector finds elements with a specific class.

$(document).ready():

The $(document).ready() method allows us to execute a function when the document is fully loaded.

click():

The click() method attaches an event handler function to an HTML element.

The function is executed when the user clicks on the HTML element.

dblclick():

The dblclick() method attaches an event handler function to an HTML element.

mouseenter():

The mouseenter() method attaches an event handler function to an HTML element.

mouseleave():

The mouseleave() method attaches an event handler function to an HTML element.

mousedown():

The mousedown() method attaches an event handler function to an HTML element.

mouseup():

The mouseup() method attaches an event handler function to an HTML element.

hover():

The hover() method takes two functions and is a combination of the mouseenter() and mouseleave() methods.

PYTHON :

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

IN BUILT FUNCTIONS IN PYTHON:

abs(x): Return the absolute value of a number. The argument may be an integer, a floating point number, or an object implementing __abs__(). If the argument is a complex number, its magnitude is returned.

Aiter(async_iterable**) :** Return an asynchronous iterator for an asynchronous iterable. Equivalent to calling x.__aiter__().

all(*iterable*) **:** Return True if all elements of the *iterable* are true (or if the iterable is empty).

any(*iterable*) **:** Return True if any element of the *iterable* is true. If the iterable is empty, return False.

bin(*x*) : Convert an integer number to a binary string prefixed with “0b”. The result is a valid Python expression. If *x* is not a Python int object, it has to define an __index__() method that returns an integer.

breakpoint(**args*, ***kws*) : This function drops you into the debugger at the call site.

callable(*object*) : Return True if the *object* argument appears callable, False if not. If this returns True, it is still possible that a call fails, but if it is False, calling *object* will never succeed.

chr(*i*) : Return the string representing a character whose Unicode code point is the integer *i*.

dir([*object*]) : Without arguments, return the list of names in the current local scope. With an argument, attempt to return a list of valid attributes for that object.

divmod(*a*, *b*) : Take two (non-complex) numbers as arguments and return a pair of numbers consisting of their quotient and remainder when using integer division.

enumerate(*iterable*, *start=0*) : Return an enumerate object. *iterable* must be a sequence, an iterator, or some other object which supports iteration.

eval(*expression*[, *globals*[, *locals*]]) : The arguments are a string and optional globals and locals. If provided, *globals* must be a dictionary. If provided, *locals* can be any mapping object.

exec(*object*[, *globals*[, *locals*]]) : This function supports dynamic execution of Python code. *object* must be either a string or a code object.

filter(*function, iterable*) : Construct an iterator from those elements of *iterable* for which *function* returns true.

float([*x*]) : Return a floating point number constructed from a number or string *x*.

format(*value*[, *format_spec*]) : Convert a *value* to a "formatted" representation, as controlled by *format_spec*. The interpretation of *format_spec* will depend on the type of the *value* argument.

getattr(*object*, *name*[, *default*]) : Return the value of the named attribute of *object*. *name* must be a string. If the string is the name of one of the object's attributes, the result is the value of that attribute.

globals() : Return the dictionary implementing the current module namespace. For code within functions, this is set when the function is defined and remains the same regardless of where the function is called.

hasattr(*object*, *name*) : The arguments are an object and a string. The result is True if the string is the name of one of the object's attributes, False if not.

hash(*object*) : Return the hash value of the object (if it has one). Hash values are integers.

hex(*x*) : Convert an integer number to a lowercase hexadecimal string prefixed with "0x".

id(*object*) : Return the "identity" of an object. This is an integer which is guaranteed to be unique and constant for this object during its lifetime.

input([*prompt*]) : If the *prompt* argument is present, it is written to standard output without a trailing newline.

isinstance(*object*, *classinfo*) : Return True if the *object* argument is an instance of the *classinfo* argument.

issubclass(*class*, *classinfo*) : Return True if *class* is a subclass (direct, indirect, or virtual) of *classinfo*. A class is considered a subclass of itself.

iter(*object*[, *sentinel*]) : Return an iterator object. The first argument is interpreted very differently depending on the presence of the second argument.

len(*s*) : Return the length.

locals() : Update and return a dictionary representing the current local symbol table.

map(*function*, *iterable*, *...*) : Return an iterator that applies *function* to every item of *iterable*, yielding the results.

max(*iterable*, *[, *key*, *default*]) : Return the largest item in an iterable or the largest of two or more arguments.

min(*iterable*, *[, *key*, *default*]) : Return the smallest item in an iterable or the smallest of two or more arguments.

next(*iterator*[, *default*]) : Retrieve the next item from the iterator by calling its __next__() method.

oct(*x*) : Convert an integer number to an octal string prefixed with “0o”.

open(*file*, *mode='r'*, *buffering=1*, *encoding=None*, *errors=None*, *newline=None*, *closefd=True*, *opener=None*) : Open *file* and return a corresponding file object.

ord(*c*) : Given a string representing one Unicode character, return an integer representing the Unicode code point of that character.

pow(*base*, *exp*[, *mod*]) : Return *base* to the power *exp.*

print(**objects, sep=' ', end='\n', file=sys.stdout, flush=False*) **:** Print *objects* to the text stream *file*, separated by *sep* and followed by *end*. *sep*, *end*, *file*, and *flush*, if present, must be given as keyword arguments.

range(*start*, *stop*[, *step*]) : Rather than being a function, range is actually an immutable sequence type.

repr(*object*) : Return a string containing a printable representation of an object.

reversed(*seq*) : Return a reverse iterator.

round(*number*[, *ndigits*]) : Return *number* rounded to *ndigits* precision after the decimal point.

setattr(*object*, *name*, *value*) : This is the counterpart of getattr().

sorted(*iterable*, */*, ***, *key=None*, *reverse=False*) : Return a new sorted list from the items in *iterable*.

str(*object=''*) : Return a str version of *object*.

sum(*iterable*, */*, *start=0*) : Sums *start* and the items of an *iterable* from left to right and returns the total. The *iterable*'s items are normally numbers, and the start value is not allowed to be a string.

super([*type*[, *object-or-type*]]) : Return a proxy object that delegates method calls to a parent or sibling class of *type*.

tuple([*iterable*]) : Rather than being a function, tuple is actually an immutable sequence type, as documented in Tuples .

type(*object*) : With one argument, return the type of an *object*. The return value is a type object and generally the same object.

vars([*object*]) : Return the **__**dict**__** attribute for a module.

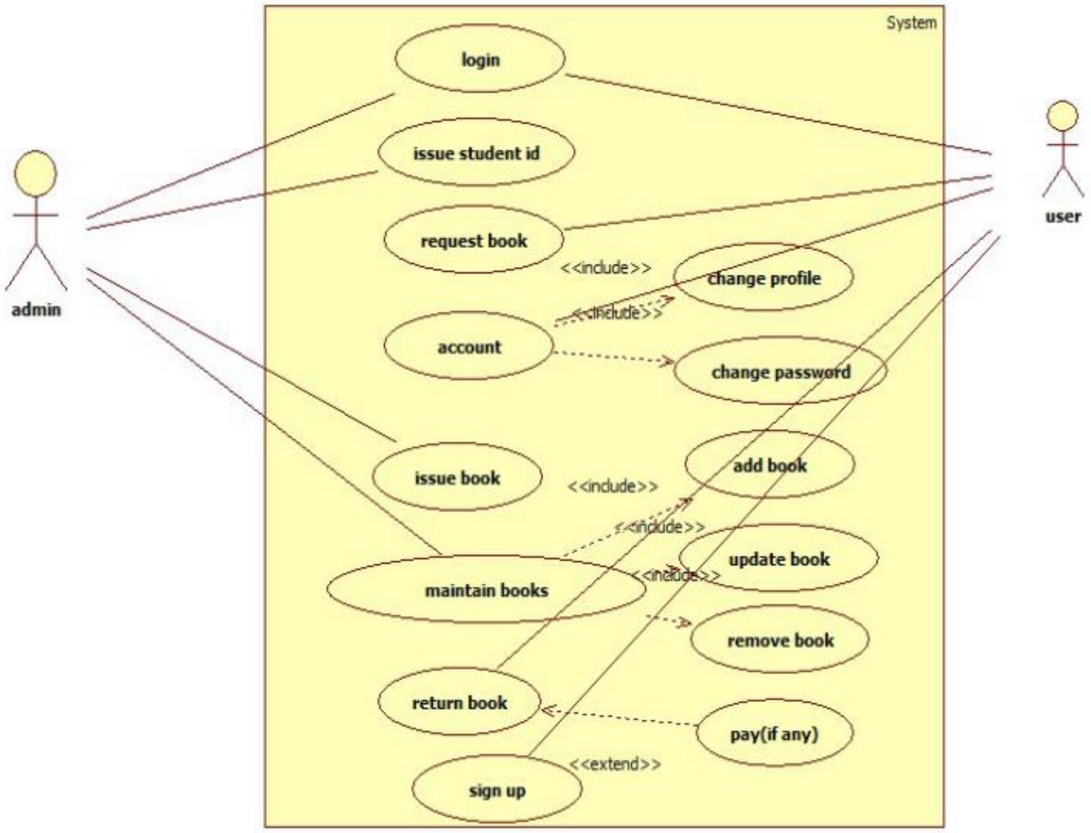
zip(**iterables, strict=False*) : Iterate over several iterables in parallel, producing tuples with an item from each one.

DJANGO :

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

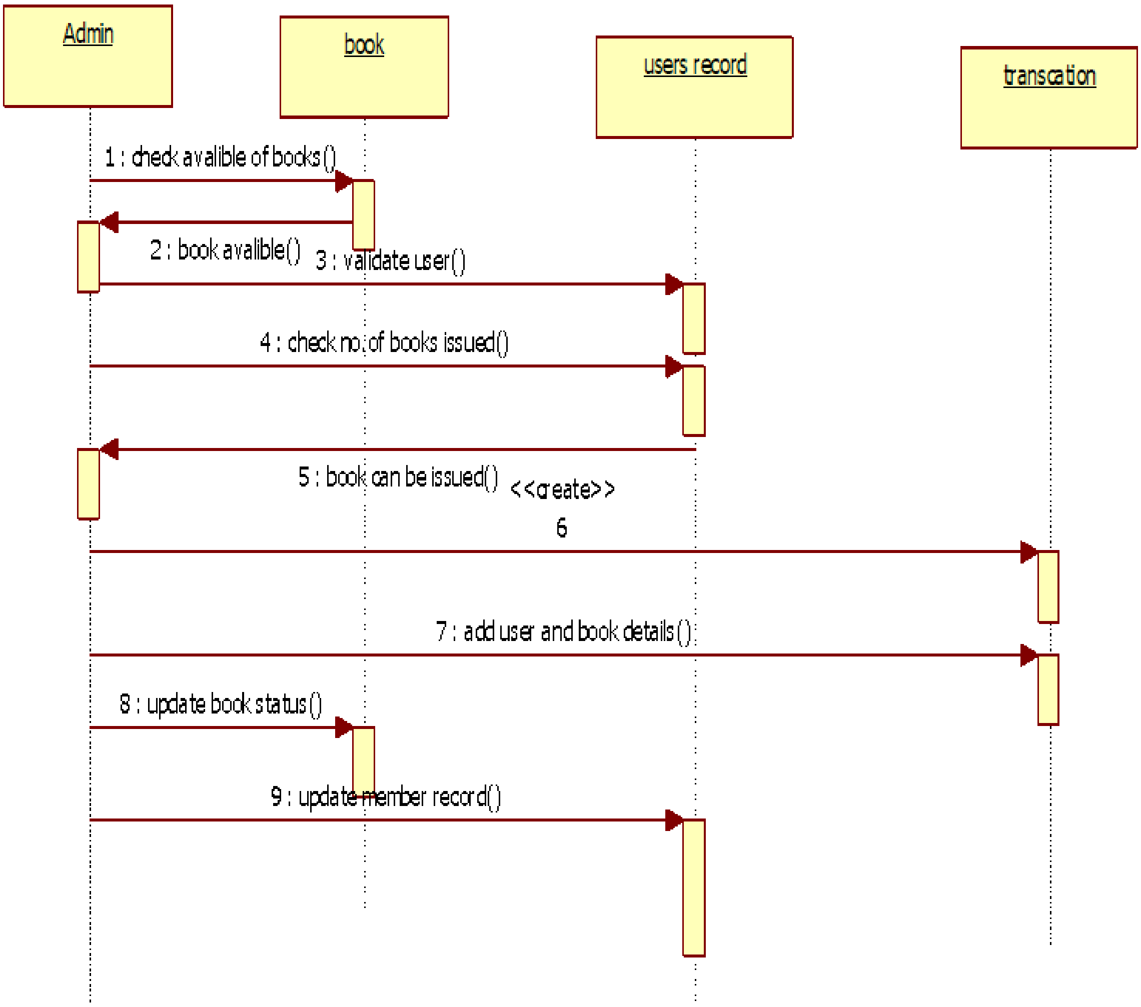
UML DIAGRAMS

Use Case Diagram:



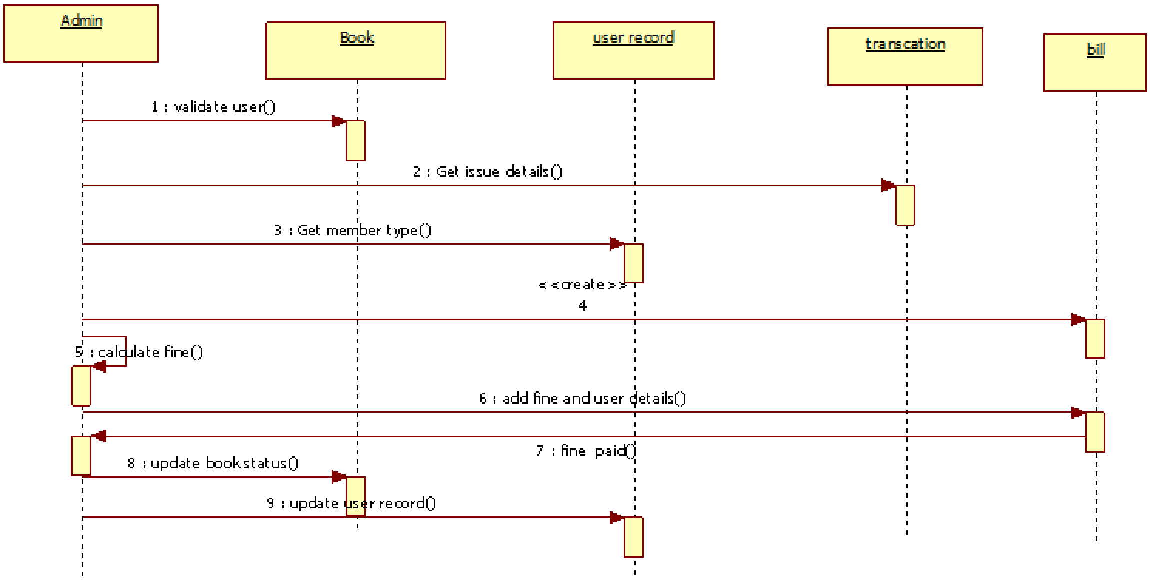
Use Case diagrams give a graphic overview of the actors involved in a system, different functions needed by those actors and how different functions interact. In that have two actors that is admin and user. Admin can manage the users account details, and manage the books information also. But the user can manage your account, request book and return book.

Sequence Diagram for issue book:



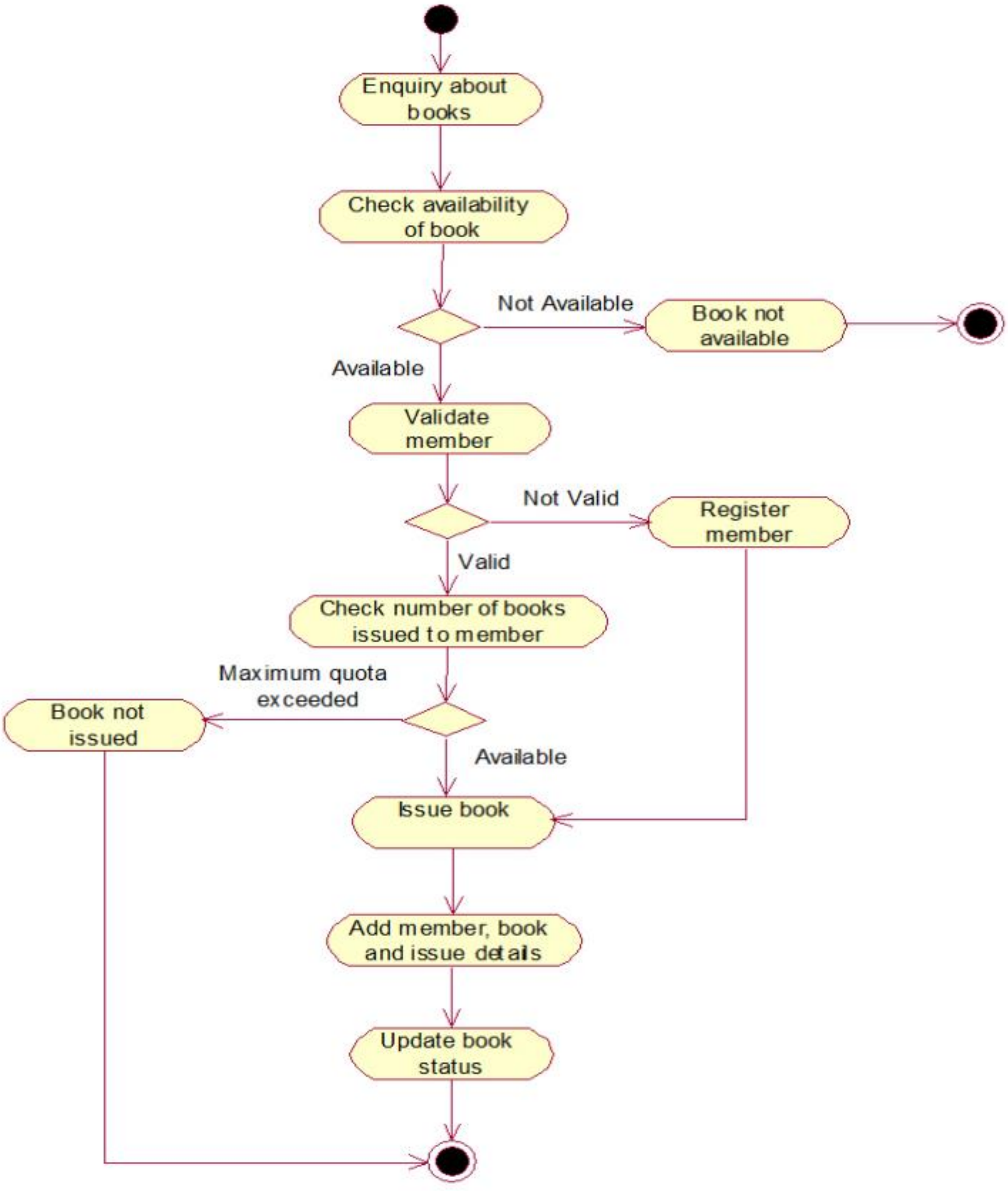
The sequence diagram can show the librarian can availability of books based on the users request, validate member, and then check no of books issued to the user in the member record, and transaction is need to create, update book status and member record table.

Sequence Diagram for Return Book:



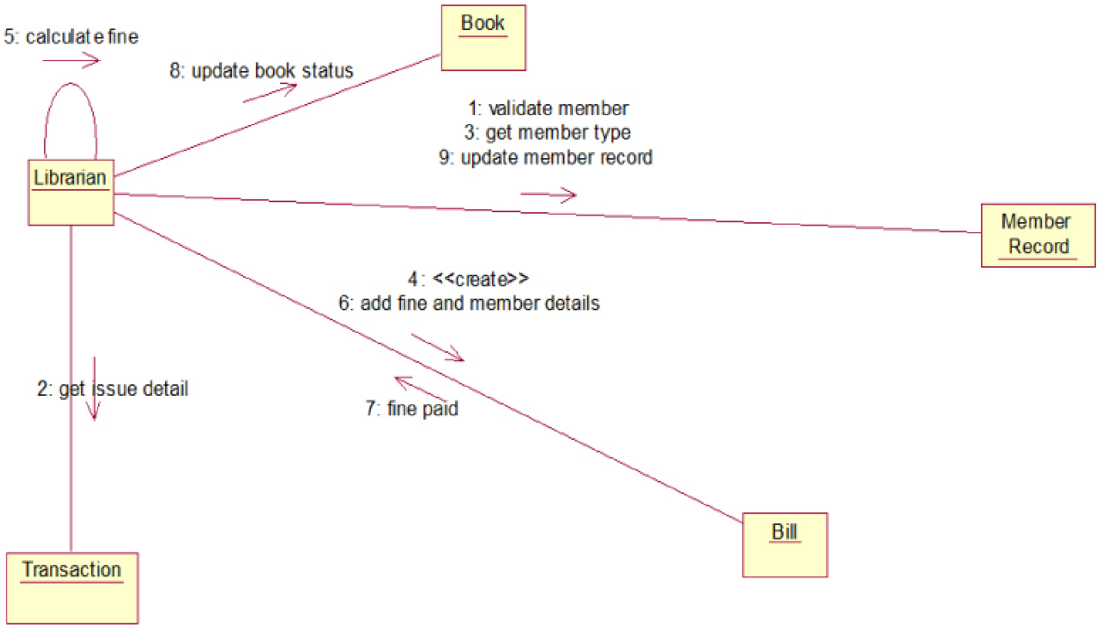
Sequence diagram for return book the member (user) cannot return the book in time then the admin can calculate the fine depends upon the duration. And that transaction is generated by librarian and update their status.

Activity Diagram:



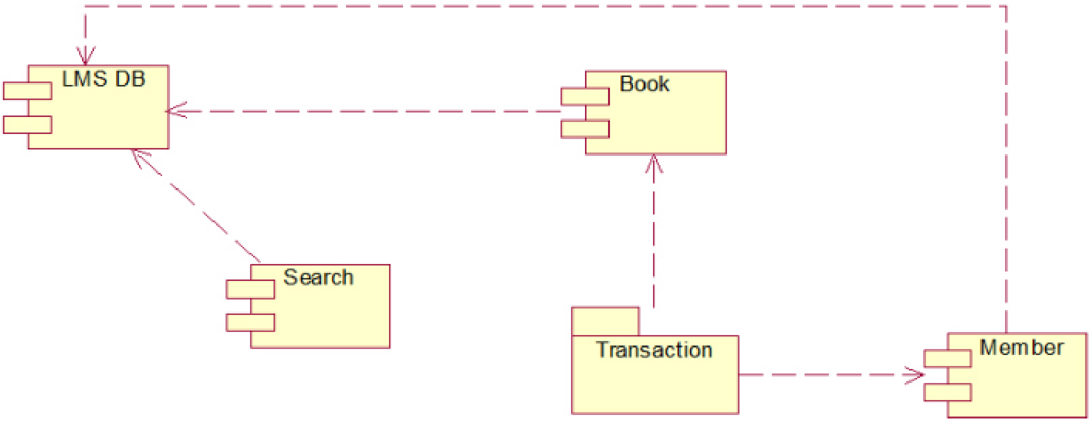
Activity Diagram represent workflows in a graphical way in library management system. In that have mainly describes the book issued to the users. The librarian can check the validity the user. If the user can validate the user can admin give gook to the user otherwise not issued book to the user.

Collaboration Diagram:



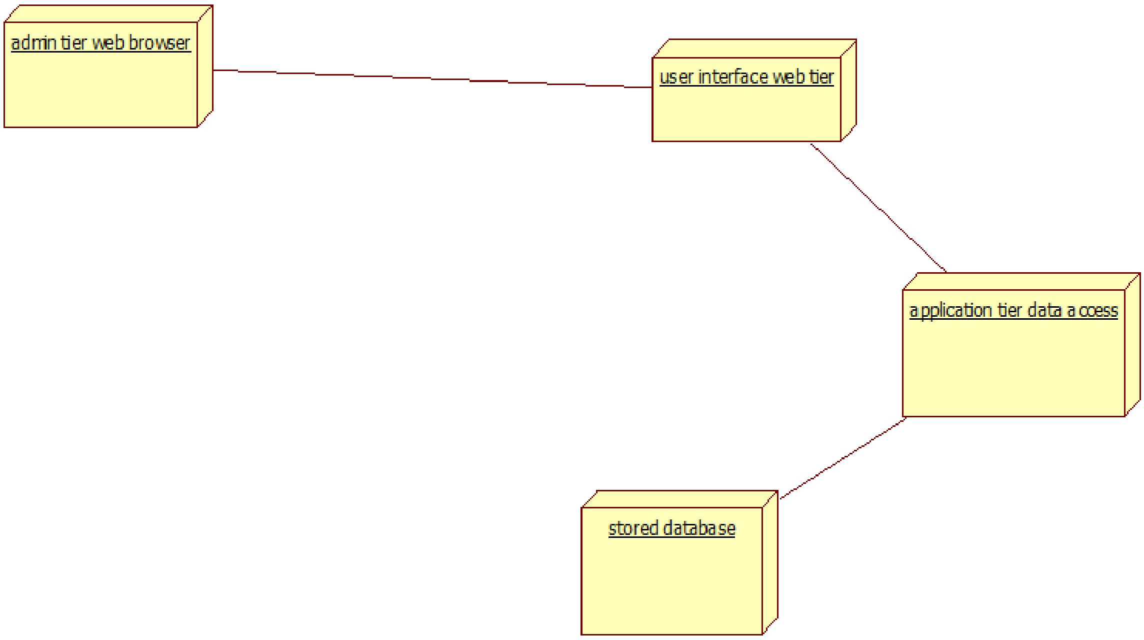
The collaboration diagram show how the users registers and authorities maintains the details of the registered users in the information system. Here the sequence is numbered according to the flow of execution and access the information of the users in librarian database.

Component Diagram:



Component diagram displays the structural relationship of components of a software system. Components communicate with each other using interfaces. The interfaces are linked using connectors.

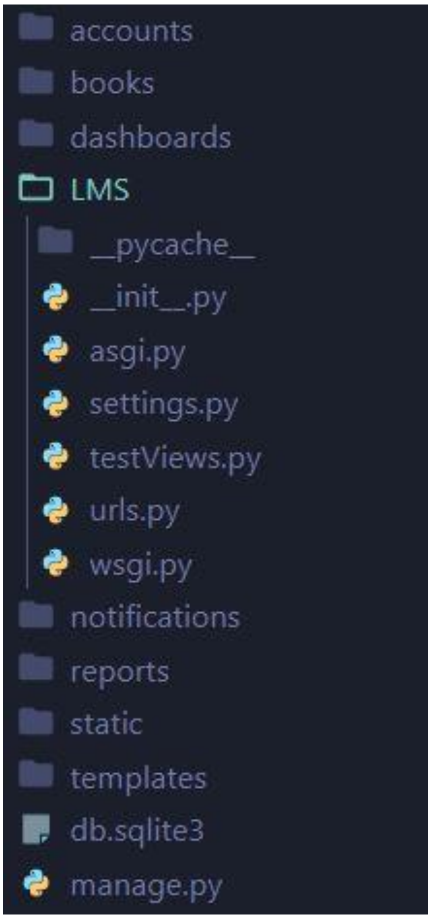
Deployment Diagram:



A Deployment diagram shows the hardware and software in that system. It can describe the main roles in that software system. The deployment diagram captures the configuration of the run time element of the application. We use deployment diagram diagram to model the static and dynamic view of an system.

FILE STRUCTURE

Base :



```
accounts
books
dashboards
LMS
    __pycache__
    __init__.py
    asgi.py
    settings.py
    testViews.py
    urls.py
    wsgi.py
notifications
reports
static
templates
db.sqlite3
manage.py
```

manage.py :

This file is the command line utility of the project and we will be using this file only to deploy, debug and test with the project.

The file contains the code for starting the server, migrating and controlling the project through command-line.

This file provides all the functionality as with the django-admin and it also provides some project specific functionalities. During this tutorial, we will frequently use some of the commands that are runserver, makemigrations, migrate etc. We will be using these commands more frequently than others.

runserver is a command to start the test server provided by Django framework and that is also one of the advantages of Django over other frameworks.

makemigrations is the command for integrating your project with files or apps you have added in it. This command will actually check for any new additions in your project and then add that to the same.
migrate, the last command is to actually add those migrations you made in the last command with the whole project. You can get the idea as the former command is used for saving the changes in the file and later one to actually apply that change to the whole project then the single file.

LMS :

LMS is the python package of the project. It has the configuration files of project settings.

i. __init__.py

The __init__.py file is empty and it exists in the project for the sole purpose of telling the python interpreter that this directory is a package. That's one of the standard rules of **python packages**.
Although we won't be doing anything on this file.

ii. settings.py

The settings.py is the main file where we will be adding all our applications and middleware applications. As the name suggests this is the main settings file of the Django project. This file contains the installed applications and middleware information which are installed on this Django project.

iii. urls.py

urls.py file contains the project level URL information. URL is universal resource locator and it provides you with the address of the resource (images, webpages, web-applications) and other resources for your website.

iv. wsgi.py

Django is based on python which uses WSGI server for web development. This file is mainly concerned with that and we will not be using this file much. wsgi is still important though if you want to deploy the applications on Apache servers or any other server because Django is still backend and you will need its support with different servers.

CODING

Settings.py

```
"""
Django settings for LMS project.
Generated by 'django-admin startproject' using Django 4.0.3.
For more information on this file, see
https://docs.djangoproject.com/en/4.0/topics/settings/
For the full list of settings and their values, see
https://docs.djangoproject.com/en/4.0/ref/settings/
"""
from pathlib import Path
import os
from decouple import config


# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent


# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/4.0/howto/deployment/checklist/


# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure-^(gw-@z&&^0kdyo3rs4vbw-_j%83&1&3_m3ijr4e400=w=lanl'


# SECURITY WARNING: don't run with debug turned on in production!
```

```
DEBUG = True
ALLOWED_HOSTS = []

# Application definition
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'accounts',
    'dashboards',
    'reports',
    'notifications',
    'books',
]
MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]
```

```
ROOT_URLCONF = 'LMS.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': ['templates'],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'LMS.wsgi.application'


# Database
# https://docs.djangoproject.com/en/4.0/ref/settings/#databases


DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}

# Password validation
# https://docs.djangoproject.com/en/4.0/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]
# Internationalization
# https://docs.djangoproject.com/en/4.0/topics/i18n/

LANGUAGE_CODE = 'en-us'
```

```
TIME_ZONE = 'UTC'
USE_I18N = True
USE_TZ = True
# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/4.0/howto/static-files/

STATIC_URL = 'static/'
STATICFILES_DIRS = ['static']

from django.contrib.messages import constants as messages
MESSAGE_TAGS = {
    messages.ERROR: 'danger',
}
# Default primary key field type
# https://docs.djangoproject.com/en/4.0/ref/settings/#default-auto-field
DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
# SMTP CONFIGURATION
EMAIL_HOST = config('EMAIL_HOST')
EMAIL_PORT = config('EMAIL_PORT', cast=int)
EMAIL_HOST_USER = config('EMAIL_HOST_USER')
EMAIL_HOST_PASSWORD = config('EMAIL_HOST_PASSWORD')
EMAIL_USE_TLS = config('EMAIL_USE_TLS', cast=bool)
```

manage.py :

```
#!/usr/bin/env python
"""Django's command-line utility for administrative tasks."""
```

```
import os
import sys

def main():

    """Run administrative tasks."""

    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'LMS.settings')

    try:

        from django.core.management import execute_from_command_line

    except ImportError as exc:

        raise ImportError(

            "Couldn't import Django. Are you sure it's installed and "

            "available on your PYTHONPATH environment variable? Did you "

            "forget to activate a virtual environment?"

        ) from exc

    execute_from_command_line(sys.argv)

if __name__ == '__main__':

    main()
```

urls.py (LMS) :

```
from django.contrib import admin

from django.urls import path

from . import testViews

urlpatterns = [

    path('admin/', admin.site.urls),
    path('', testViews.adminHome, name='adminHome'),

    path('students', testViews.students, name='students'),

    path('bookRegister', testViews.bookRegister, name='bookRegister'),

]
```

Views.py (accounts) :

```
from accounts.models import Account

from django.shortcuts import get_object_or_404, redirect, render, HttpResponse

from .forms import RegistrationForm

from .models import Account, UserProfile

from django.contrib import messages, auth

from django.contrib.auth.decorators import login_required


# email verification

from django.contrib.sites.shortcuts import get_current_site

from django.template.loader import render_to_string

from django.utils.http import urlsafe_base64_decode, urlsafe_base64_encode

from django.contrib.auth.tokens import default_token_generator

from django.core.mail import EmailMessage

from django.utils.encoding import force_bytes

from books.models import book, bookProduct

# forms to edit profile

from .forms import UserProfileForm, UserForm

# change password

# from django.contrib import ch
```

```
def register(request):
    if request.method == 'POST':
        form = RegistrationForm(request.POST)
        if form.is_valid():
            first_name = form.cleaned_data['first_name']
            phone_number = form.cleaned_data['phone_number']
            last_name = form.cleaned_data['last_name']
            email = form.cleaned_data['email']
            password = form.cleaned_data['password']
            # confirm_password = form.cleaned_data['confirm_password']
            username = email.split('@')[0]

            # if password == confirm_password:
            user = Account.objects.create_user(first_name=first_name, last_name=last_name, email=email, username=username,password=password)
            user.phone_number = phone_number
            user.save()

            # user activation
            current_site = get_current_site(request)
            email_subject = 'Account Activation'
            message = render_to_string('accounts/verification_email.html',{
                'user':user,
                'domain':current_site,
                'uid': urlsafe_base64_encode(force_bytes(user.pk)),
                'token': default_token_generator.make_token(user),
            })
            to_email = email
            send_mail = EmailMessage(email_subject,message, to=[to_email])
            send_mail.send()

            # messages.success(request, 'Registration successful, We have sent an verification
email')
            return redirect('/accounts/login?command=verification&email='+email)
    else:
        form = RegistrationForm()
    context = {'form':form}
    return render(request, 'accounts/register.html', context)

def activate(request, uidb64, token):
    try:
        uid = urlsafe_base64_decode(uidb64).decode()
        user = Account._default_manager.get(pk=uid)
    except(ValueError, Account.DoesNotExist,TypeError,OverflowError):
        user = None

    if user is not None and default_token_generator.check_token(user,token):
        user.is_active = True
        user.save()
        messages.success(request, "Your account has been activated")
        return redirect('login')
    else:
        messages.error(request, 'invalid acitvation link')
        return redirect('register')


def login(request):
    if request.method == 'POST':
        email = request.POST['email']
        password = request.POST['password']


        user = auth.authenticate(request, email=email, password=password)
        if user:
            try:
              cart = Cart.objects.get(cart_id=_cart_id(request))
            except Cart.DoesNotExist:
                cart = None
            is_cart_item_exist = CartItem.objects.filter(cart=cart).exists()
            if is_cart_item_exist:
                cart_items = CartItem.objects.filter(cart = cart)
                # Get product variation by cart_id
                product_variation = []
                for item in cart_items:
                    variation = item.variation.all()
                    product_variation.append(list(variation))
                # get cart item of user and store exisiting variations list
                cart_items = CartItem.objects.filter(user=user)
                ex_var_list = []
                id = []
                for item in cart_items:
                    existing_variation = item.variation.all()
                    ex_var_list.append(list(existing_variation))
                    id.append(item.id)

                for pr in product_variation:
                    if pr in ex_var_list:
                        index = ex_var_list.index(pr)
                        item_id = id[index]

                        item = CartItem.objects.get(id=item_id)
                        print('item :', item)
                        item.quantity +=1
                        item.user = user
                        print('item.user :',user)
                        item.save()
                else:
                    cart_items = CartItem.objects.all().filter(cart=cart)
                    for item in cart_items:
                        item.user = user
                        item.save()


        else:
            messages.error(request, 'Invalid credentials')
```

```
            return redirect('login')


        auth.login(request, user)
        messages.success(request,'Now You logged in')
        return redirect('dashboard')
    return render(request, 'accounts/login.html')
@login_required(login_url='login')
def logout(request):
    auth.logout(request)
    messages.success(request,'You was logged out')
    return redirect('login')




def forgotPassword(request):
    if request.method == 'POST':
        email = request.POST['email']

        if Account.objects.filter(email=email).exists():
            user = Account.objects.get(email=email)

            # Reset password email
            current_site = get_current_site(request)
            email_subject = 'Account Activation'
            message = render_to_string('accounts/reset_password_email.html',{
                'user':user,
                'domain':current_site,
                'uid': urlsafe_base64_encode(force_bytes(user.pk)),
                'token': default_token_generator.make_token(user),
            })
            to_email = email
            send_mail = EmailMessage(email_subject,message, to=[to_email])
            send_mail.send()

            messages.success(request, 'Reset password link has been sent to your email')
            return redirect('login')
        else:
            messages.error(request, 'Email does not exist')
            return redirect('forgotPassword')
    return render(request, 'accounts/forgot_password.html')


def resetpassword_validation(request, uidb64,  token):
    try:
        uid = urlsafe_base64_decode(uidb64).decode()
        print(uid)
        user = Account._default_manager.get(pk=uid)

    except(ValueError, Account.DoesNotExist,TypeError,OverflowError):
        user = None
    if user is not None and default_token_generator.check_token(user,token):
            request.session['uid'] = uid
            messages.success(request, 'Reset your password')
            return redirect('resetPassword')
    else:
        messages.error(request, 'This link has been expired.')
        return redirect('login')



def resetPassword(request):
    if request.method == 'POST':
        password = request.POST['password']
        confirm_password = request.POST['confirm_password']

        if password == confirm_password:
            uid = request.session.get('uid')
            user = Account.objects.get(pk=uid)
            user.set_password(password)
            user.save()
            messages.success(request, 'Password reset was succesful')
            return redirect('login')
        else:
            messages.error(request, 'Password in not matched')
            return redirect('resetPassword')
    else:
        return render(request, 'accounts/reset_password.html')



@login_required(login_url='login')
def dashboard(request):
    books = book.objects.filter(user_id=request.user.id,is_issued=True).book_by('-created_at')
    book_count = books.count()
    userprofile = UserProfile.objects.get(user_id=request.user.id)

    context = {
        'book_count' : book_count,
        'userprofile':userprofile
    }
    return render(request, 'accounts/dashboard.html',context)



@login_required(login_url='login')
def mybooks(request):
    books = Book.objects.filter(user=request.user,is_issued=True).book_by('-created_at')
    context={
        'books':books
    }
    return render(request, 'accounts/my_books.html',context)
```

```
@login_required(login_url='login')
def edit_profile(request):
    userprofile = get_object_or_404(UserProfile, user=request.user)
    if request.method == 'POST':
        user_form = UserForm(request.POST, instance=request.user)
        profile_form = UserProfileForm(request.POST, request.FILES, instance=userprofile)

        if user_form.is_valid() and profile_form.is_valid():
            user_form.save()
            profile_form.save()
    else:
        user_form = UserForm(instance=request.user)
        profile_form = UserProfileForm(instance=userprofile)


    context = {
        'user_form':user_form,
        'profile_form':profile_form,
        'userprofile':userprofile
    }
    return render(request, 'accounts/edit_profile.html', context)


@login_required(login_url='login')
```

```
def change_password(request):
    if request.method == 'POST':
        current_password = request.POST['current_password']
        new_password = request.POST['confirm_password']
        confirm_password = request.POST['confirm_password']

        user = Account.objects.get(username__exact=request.user.username)
        if new_password == confirm_password:
            success = user.check_password(current_password)
            if success:
                user.set_password(new_password)
                user.save()
                messages.success(request, 'Your password has been changed')
                return redirect('change_password')
            else:
                messages.error(request, 'Invalid Current Password')
                return redirect('change_password')
        else:
            messages.error(request, 'Password does not matched')
            return redirect('change_password')
    context = {}
    return render(request, 'accounts/change_password.html', context)
```

models.py (accounts) :

```
from django.db import models
from django.contrib.auth.models import AbstractBaseUser, BaseUserManager
```

```
# Create your models here.
class MyAccountManager(BaseUserManager):
    def create_user(self, first_name, last_name, username, email, password=None):
        if not email:
            raise ValueError('User must have an email address')

        if not username:
            raise ValueError('User must have an username')

        user = self.model(
            email = self.normalize_email(email),
            username = username,
            first_name = first_name,
            last_name = last_name,
        )

        user.set_password(password)
        user.save(using=self._db)
        return user

    def create_superuser(self, first_name, last_name, email, username, password):
        user = self.create_user(
            email = self.normalize_email(email),
            username = username,
            password = password,
            first_name = first_name,
            last_name = last_name,
        )
        user.is_admin = True
        user.is_active = True
        user.is_staff = True
        user.is_superadmin = True
        user.save(using=self._db)
        return user



class Account(AbstractBaseUser):
    first_name      = models.CharField(max_length=50)
    last_name       = models.CharField(max_length=50)
    username        = models.CharField(max_length=50, unique=True)
    email           = models.EmailField(max_length=100, unique=True)
    phone_number    = models.CharField(max_length=50)

    # required
    date_joined     = models.DateTimeField(auto_now_add=True)
    last_login      = models.DateTimeField(auto_now_add=True)
    is_admin        = models.BooleanField(default=False)
    is_staff        = models.BooleanField(default=False)
    is_active       = models.BooleanField(default=False)
    is_superadmin   = models.BooleanField(default=False)

    USERNAME_FIELD = 'email'
    REQUIRED_FIELDS = ['username', 'first_name', 'last_name']

    objects = MyAccountManager()

    def full_name(self):
        return f'{self.first_name} {self.last_name}'

    def __str__(self):
        return self.email

    def has_perm(self, perm, obj=None):
        return self.is_admin

    def has_module_perms(self, add_label):
        return True


class UserProfile(models.Model):
    user = models.OneToOneField(Account, on_delete=models.CASCADE)
    stream = models.CharField(max_length=100, blank=True)
    roll_no = models.CharField(max_length=100, blank=True)
    year = models.CharField(max_length=20, blank=True)
    profile_pic = models.ImageField(upload_to='user_profile/', blank=True)


    def __str__(self):
        return self.user.first_name


    def full_address(self):
        return f'{self.address_line1} {self.address_line2}'
```

forms.py (accounts) :

```
from django.contrib.auth.models import User
from .models import Account, UserProfile
from django import  forms


class RegistrationForm(forms.ModelForm):
    password = forms.CharField(widget=forms.PasswordInput(attrs={
        'placeholder':'Enter password',
        'class':'form-control'
    }))
    confirm_password = forms.CharField(widget=forms.PasswordInput(attrs={
        'placeholder':'Confirm password',
        'class':'form-control'
    }))
    class Meta:
        model = Account
        fields = ['first_name','last_name','email','password','phone_number']
```

```
def __init__(self, *args, **kwargs):
    super(RegistrationForm, self).__init__(*args, **kwargs)
    self.fields['first_name'].widget.attrs['placeholder'] = 'First Name'
    self.fields['last_name'].widget.attrs['placeholder'] = 'Last Name'
    self.fields['email'].widget.attrs['placeholder'] = 'Enter Email'
    self.fields['phone_number'].widget.attrs['placeholder'] = 'Enter Phone Number'
    self.fields['password'].widget.attrs['placeholder'] = 'Enter Password'
    self.fields['confirm_password'].widget.attrs['placeholder'] = 'Confirm Password'

    for field in self.fields:
        self.fields[field].widget.attrs['class'] = 'form-control'
        # self.fields[field].widget.attrs['placeholder'] = field

# Confirm password in the form itself
def clean(self):
    cleaned_data = super(RegistrationForm, self).clean()
    password = cleaned_data.get('password')
    conf_password = cleaned_data.get('confirm_password')

    if password != conf_password:
        raise forms.ValidationError(
            'Password Does Not Match'
        )
```

```
class UserForm(forms.ModelForm):
    class Meta :
        model = Account
        fields = ['first_name','last_name', 'phone_number']

    def __init__(self,*args,**kwars):
        super(UserForm,self).__init__(*args,**kwars)
        for field in self.fields:
            self.fields[field].widget.attrs['class']='form-control'

class UserProfileForm(forms.ModelForm):
    profile_pic = forms.ImageField(required=False,error_messages={'Invalid':('image files only')}, widget=forms.FileInput)
    class Meta :
        model = UserProfile
        fields = ['address_line1','address_line2', 'city','state','country', 'profile_pic']

    def __init__(self,*args,**kwars):
        super(UserProfileForm,self).__init__(*args,**kwars)
        for field in self.fields:
            self.fields[field].widget.attrs['class']='form-control'
```

baseAdminDashboard.html :

```
{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">



    <!-- Font Awesome CDN  -->

    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">



    <!--=========== Bootstrap cdn ===========-->

      <!-- Bootstrap CSS -->

      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-KyZXEAg3QhqLMpG8r+8fhAXLRk2vvoC2f3B09zVXn8CA5QIVfZOJ3BCsw2P0p/We" crossorigin="anonymous">



    <!-- Aos Cdn -->

    <link href="https://unpkg.com/aos@2.3.1/dist/aos.css" rel="stylesheet">



    <!-- Chart Js CDN -->

    <script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/3.0.1/chart.min.js" integrity="sha512-2uu1jrAmW1A+SMwih5DAPqzFS2PI+OPw79OVLS4NJ6jGHQ/GmIVDDlWwz4KLO8DnoUmYdU8hTtFcp8je6zxbCg==" crossorigin="anonymous" referrerpolicy="no-referrer"></script>
    <!-- Custom CSS -->

    <link rel="stylesheet" href="{% static 'css/main.css' %}">

    <link rel="stylesheet" href="{% static 'css/baseDashboard.css' %}">

    <link rel="stylesheet" href="{% static 'css/dashboard.css' %}">


    {% block title %}

    {% endblock title %}

</head>

<body>


    <div class="page-wrapper chiller-theme toggled shadow">

      <a id="show-sidebar" class="btn btn-sm btn-dark" href="#">

        <i class="fa fa-bars"></i>

      </a>

      <nav id="sidebar" class="sidebar-wrapper">

        <div class="sidebar-content">

          <div class="sidebar-brand">

            <img src="{% static 'img/logo.png' %}" class="bg-white rounded" width="90%"
alt="">

            <div id="close-sidebar" >

              <i class="fa fa-times"></i>

            </div>

          </div>

          <!-- sidebar-header  -->
```

```
<div class="sidebar-search">
  <div>
    <div class="input-group">
      <input type="text" class="form-control search-menu" placeholder="Search...">
      <div class="input-group-append">
        <span class="input-group-text">
          <i class="fa fa-search" ></i>
        </span>
      </div>
    </div>
  </div>
</div>
<!-- sidebar-search  -->
<div class="sidebar-menu">
  <ul>
    <li class="header-menu">
      <span>General</span>
    </li>
    <li class="nav-item active">
      <a href="{% url 'adminHome' %}"> <i class="fa fa-tachometer"></i>
        <span>Dashboard</span>
      </a>
    </li>

    <li class="nav-item">
```

```
 <a href="{% url 'students' %}"> <i class="fa fa-users"></i>
   <span>Students</span>
   <!-- <span class="badge badge-pill badge-warning"></span> -->
 </a>
</li>

<li class="nav-item">
 <a href="{% url 'bookRegister' %}"> <i class="fa fa-book"></i>
   <span>Book Register</span>
 </a>
</li>

<li class="nav-item">
 <a href=""> <i class="fa fa-bar-chart"></i>
   <span>Reports</span>
 </a>
</li>

<li class="nav-item">
 <a href=""> <i class="fa fa-bell"></i>
   <span>Notifications</span>
 </a>
</li>
<li class="header-menu">
 <span>Account</span> </li>
```

```
        <li class="nav-item">
          <a href=""> <i class="fa fa-user"></i>
            <span>Profile</span>
          </a>
        </li>

        <li class="nav-item">
          <a href=""> <i class="fa fa-lock"></i>
            <span>Privacy</span>
          </a>
        </li>
        <li class="nav-item">
          <a href="#"><i class="fa fa-sign-out"></i>
            <span>Logout</span>
          </a>
        </li>
      </ul>
    </div>
    <!-- sidebar-menu  -->
  </div>
</nav>
<!-- sidebar-wrapper  -->
<main class="page-content">
 {% block content %}
 {% endblock content %}
```

```
     </main>

     <!-- page-content" -->

   </div>



   <!-- Bootstrap -->

   <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js" integrity="sha384-
J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n"
crossorigin="anonymous"></script>

   <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
integrity="sha384Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvo
xMfooAo" crossorigin="anonymous"></script>

   <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"
integrity="sha384wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3
Og8ifwB6" crossorigin="anonymous"></script>



   <!-- Custom Js -->

   <script src="{% static 'js/dashboard.js' %}"></script>

   <script src="{% static 'js/charts.js' %}"></script>

   </body>

</html>
```

adminDashboard.html :

```
{% extends 'dashboards/admin/baseAdminDashboard.html' %}




{% load static %}

{% block title %}

  <title>dashboard</title>

{% endblock title %}
{% block content %}


<section id="admin-dashboard">
 <div class="title">
    <h2 class="text-center my-2">ENGINEERING Statistics</h2>
    <div class="bar"></div>
 </div>
 <div class="row container stats">
  <div class="col col-md-2">
    <i class="fa fa-male"></i>
    <h5>Students</h5>
    <h5 class="mt-2">1489</h5>

    <div class="on-hover">
      <a href="#"><i class="fa fa-eye">View Students</i></a>
    </div>
  </div>
  <div class="col col-md-2 ">
    <i class="fa fa-book"></i>
    <h5>Total Books</h5>
    <h5 class="mt-2">12976</h5>

    <div class="on-hover">
```

```
      <a href="#"><i class="fa fa-eye">View Students</i></a>
    </div>
  </div>
  <div class="col col-md-2 ">
    <i class="fa fa-clone"></i>
    <h5>Available Books</h5>
    <h5 class="mt-2">11248</h5>
  </div>
  <div class="col col-md-2 ">
    <i class="fa fa-list-alt"></i>
    <h5>Books Issued</h5>
    <h5 class="mt-2">3492</h5>
  </div>
  <div class="col col-md-2 ">
    <i class="fa fa-inr"></i>
    <h5>Books price</h5>
    <h5 class="mt-2">89,850</h5>
  </div>
</div>

  <div class="title mt-5">
    <h2 class="text-center my-2">DIPLOMA Statistics</h2>
    <div class="bar"></div>
 </div>
  <div class="row container stats">
```

```
<div class="col col-md-2">
  <i class="fa fa-male"></i>
  <h5>Students</h5>
  <h5 class="mt-2">879</h5>
</div>
<div class="col col-md-2 ">
  <i class="fa fa-book"></i>
  <h5>Total Books</h5>
  <h5 class="mt-2">4976</h5>
</div>
<div class="col col-md-2 ">
  <i class="fa fa-clone"></i>
  <h5>Available Books</h5>
  <h5 class="mt-2">3248</h5>
</div>
<div class="col col-md-2 ">
  <i class="fa fa-list-alt"></i>
  <h5>Books Issued</h5>
  <h5 class="mt-2">1092</h5>
</div>
<div class="col col-md-2 ">
  <i class="fa fa-inr"></i>
  <h5>Books price</h5>
  <h5 class="mt-2">49,850</h5>
</div>
```

```
  </div>
</section>
{% endblock content %}
```

bookRegister.html :

```
{% extends 'dashboards/admin/baseAdminDashboard.html' %}
{% load static %}
{% block title %}
  <title>Students</title>
{% endblock title %}

{% block content %}
<section id="admin-students" class="container">
    <div class="title">
        <h2 class="text-center my-2">Book Register</h2>
        <div class="bar"></div>
     </div>

     <div class="input-group mb-3 mt-5 bg-white" id="search-form">
        <input type="text" class="form-control" placeholder="Search Student" >
        <div class="input-group-append text-center">
          <i class="fa fa-search text-lg p-2"></i>
        </div>
      </div>

     <form action="" class="student-filter">
        <div class="row">
            <div class="col-sm-3">
                <select name="" id="stream">
                    <option value="">-- Select Stream --</option>
                    <option value="">Engineering</option>
            <option value="">Diploma</option>
            <option value="">Faculty</option>
        </select>
    </div>

    <div class="col-sm-3">
        <select name="" id="course" >
            <option value="">-- Select Course --</option>
            <option value="">CIVIL</option>
            <option value="">CSE</option>
            <option value="">ECE</option>
            <option value="">EEE</option>
            <option value="">MECH</option>
        </select>
    </div>

    <div class="col-sm-3">
        <select name="" id="year" >
            <option value="">-- Select Year --</option>
            <option value="">1st</option>
            <option value="">2nd</option>
            <option value="">3rd</option>
            <option value="">4th</option>
        </select>
    </div>

    <div class="col-sm-3">
        <button class="btn btn-primary">Filter</button>
    </div>
</div>
```

```
    </form>


    <div class="ctns d-flex justify-content-end mt-5 ">
      <button id="export-student-btn" class="btn-success mx-2"><i class="fa fa-plus"></i> Export</button>
      <button id="add-student-btn" class="btn-primary"><i class="fa fa-plus"></i> Add Student</button>
    </div>
    <table class="table mt-2 table-striped">
      <thead class="text-white">
        <tr>
          <th scope="col">Roll No</th>
          <th scope="col">Name</th>
          <th scope="col">Branch</th>
          <th scope="col">Stream</th>
          <th scope="col">Year</th>
          <th scope="col">DOB</th>
          <th scope="col">Mobile</th>
          <th scope="col">Edit/View</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <th scope="row">1</th>
          <td>Mark</td>
          <td>CSE</td>
          <td>Engineering</td>
          <td>3rd</td>
          <td>01/01/2001</td>
          <td>996784567</td>
          <td>
```

```
  <button class="btn-sm btn-success"><i class="fa fa-pencil"></i> Edit</button>
  <button class="btn-sm btn-primary"> <i class="fa fa-eye"></i> View</button>
  </td>
</tr>

<tr>
  <th scope="row">2</th>
  <td>Mark</td>
  <td>CSE</td>
  <td>Dipoma</td>
  <td>3rd</td>
  <td>01/01/2001</td>
  <td>996784567</td>
  <td>
  <button class="btn-sm btn-success"><i class="fa fa-pencil"></i> Edit</button>
  <button class="btn-sm btn-primary"> <i class="fa fa-eye"></i> View</button>
  </td>
</tr>

<tr>
  <th scope="row">3</th>
  <td>Mark</td>
  <td>CSE</td>
  <td>Faculty</td>
  <td>3rd</td>
  <td>01/01/2001</td>
  <td>996784567</td>
  <td>
  <button class="btn-sm btn-success"><i class="fa fa-pencil"></i> Edit</button>
  <button class="btn-sm btn-primary"> <i class="fa fa-eye"></i> View</button>
  </td>
</tr>

<tr>
  <th scope="row">4</th>
  <td>Mark</td>
  <td>CSE</td>
  <td>Dipoma</td>
  <td>3rd</td>
  <td>01/01/2001</td>
  <td>996784567</td>
  <td>
  <button class="btn-sm btn-success"><i class="fa fa-pencil"></i> Edit</button>
  <button class="btn-sm btn-primary"> <i class="fa fa-eye"></i> View</button>
  </td>
</tr>

<tr>
  <th scope="row">5</th>
  <td>Mark</td>
  <td>CSE</td>
  <td>Dipoma</td>
  <td>3rd</td>
  <td>01/01/2001</td>
  <td>996784567</td>
  <td>
  <button class="btn-sm btn-success"><i class="fa fa-pencil"></i> Edit</button>
  <button class="btn-sm btn-primary"> <i class="fa fa-eye"></i> View</button>
  </td>
</tr>
```

```
    <tr>
      <th scope="row">6</th>
      <td>Mark</td>
      <td>CSE</td>
      <td>Dipoma</td>
      <td>3rd</td>
      <td>01/01/2001</td>
      <td>996784567</td>
      <td>
      <button class="btn-sm btn-success"><i class="fa fa-pencil"></i> Edit</button>
      <button class="btn-sm btn-primary"> <i class="fa fa-eye"></i> View</button>
      </td>
    </tr>

    <tr>
      <th scope="row">7</th>
      <td>Mark</td>
      <td>CSE</td>
      <td>Dipoma</td>
      <td>3rd</td>
      <td>01/01/2001</td>
      <td>996784567</td>
      <td>
      <button class="btn-sm btn-success"><i class="fa fa-pencil"></i> Edit</button>
      <button class="btn-sm btn-primary"> <i class="fa fa-eye"></i> View</button>
      </td>
    </tr>
     </tbody>
</table>
```

```
</section>
{% endblock content %}
```

Students.hmtl :

```
{% extends 'dashboards/admin/baseAdminDashboard.html' %}



{% load static %}


{% block title %}
  <title>Students</title>
{% endblock title %}



{% block content %}


<section id="admin-students" class="container">
    <div class="title">
        <h2 class="text-center my-2">STUDENT'S DATA</h2>
        <div class="bar"></div>
     </div>


     <div class="input-group mb-3 mt-5 bg-white" id="search-form">
        <input type="text" class="form-control" placeholder="Search Student" >
        <div class="input-group-append text-center">
          <i class="fa fa-search text-lg p-2"></i>
        </div>
      </div>


     <form action="" class="student-filter">
```

```
<div class="row">
    <div class="col-sm-3">
        <select name="" id="stream">
            <option value="">-- Select Stream --</option>
            <option value="">Engineering</option>
            <option value="">Diploma</option>
            <option value="">Faculty</option>
        </select>
    </div>

    <div class="col-sm-3">
        <select name="" id="course" >
            <option value="">-- Select Course --</option>
            <option value="">CIVIL</option>
            <option value="">CSE</option>
            <option value="">ECE</option>
            <option value="">EEE</option>
            <option value="">MECH</option>
        </select>
    </div>

    <div class="col-sm-3">
        <select name="" id="year" >
            <option value="">-- Select Year --</option>
            <option value="">1st</option>
            <option value="">2nd</option>
            <option value="">3rd</option>
            <option value="">4th</option>
        </select>
    </div>

    <div class="col-sm-3">
```

```
            <button class="btn btn-primary">Filter</button>
          </div>
        </div>
      </form>


      <div class="ctns d-flex justify-content-end mt-5 ">
        <button id="export-student-btn" class="btn-success mx-2"><i class="fa fa-plus"></i>
Export</button>
        <button id="add-student-btn" class="btn-primary"><i class="fa fa-plus"></i> Add
Student</button>
      </div>
      <table class="table mt-2 table-striped">
        <thead class="text-white">
          <tr>
            <th scope="col">Roll No</th>
            <th scope="col">Name</th>
            <th scope="col">Branch</th>
            <th scope="col">Stream</th>
            <th scope="col">Year</th>
            <th scope="col">DOB</th>
            <th scope="col">Mobile</th>
            <th scope="col">Edit/View</th>
          </tr>
        </thead>
        <tbody>
          <tr>
            <th scope="row">1</th>
            <td>Mark</td>
            <td>CSE</td>
            <td>Engineering</td>
            <td>3rd</td>
            <td>01/01/2001</td>
            <td>996784567</td>
```

```
  <td>
  <button class="btn-sm btn-success"><i class="fa fa-pencil"></i> Edit</button>
  <button class="btn-sm btn-primary"> <i class="fa fa-eye"></i> View</button>
  </td>
</tr>

<tr>
  <th scope="row">2</th>
  <td>Mark</td>
  <td>CSE</td>
  <td>Dipoma</td>
  <td>3rd</td>
  <td>01/01/2001</td>
  <td>996784567</td>
  <td>
  <button class="btn-sm btn-success"><i class="fa fa-pencil"></i> Edit</button>
  <button class="btn-sm btn-primary"> <i class="fa fa-eye"></i> View</button>
  </td>
</tr>

<tr>
  <th scope="row">3</th>
  <td>Mark</td>
  <td>CSE</td>
  <td>Faculty</td>
  <td>3rd</td>
  <td>01/01/2001</td>
  <td>996784567</td>
  <td>
  <button class="btn-sm btn-success"><i class="fa fa-pencil"></i> Edit</button>
  <button class="btn-sm btn-primary"> <i class="fa fa-eye"></i> View</button>
  </td>
```

```
</tr>

<tr>
  <th scope="row">4</th>
  <td>Mark</td>
  <td>CSE</td>
  <td>Dipoma</td>
  <td>3rd</td>
  <td>01/01/2001</td>
  <td>996784567</td>
  <td>
  <button class="btn-sm btn-success"><i class="fa fa-pencil"></i> Edit</button>
  <button class="btn-sm btn-primary"> <i class="fa fa-eye"></i> View</button>
  </td>
</tr>

<tr>
  <th scope="row">5</th>
  <td>Mark</td>
  <td>CSE</td>
  <td>Dipoma</td>
  <td>3rd</td>
  <td>01/01/2001</td>
  <td>996784567</td>
  <td>
  <button class="btn-sm btn-success"><i class="fa fa-pencil"></i> Edit</button>
  <button class="btn-sm btn-primary"> <i class="fa fa-eye"></i> View</button>
  </td>
</tr>

<tr>
  <th scope="row">6</th>
```

```
          <td>Mark</td>
          <td>CSE</td>
          <td>Dipoma</td>
          <td>3rd</td>
          <td>01/01/2001</td>
          <td>996784567</td>
          <td>
          <button class="btn-sm btn-success"><i class="fa fa-pencil"></i> Edit</button>
          <button class="btn-sm btn-primary"> <i class="fa fa-eye"></i> View</button>
          </td>
        </tr>
      <tr>
          <th scope="row">7</th>
          <td>Mark</td>
          <td>CSE</td>
          <td>Dipoma</td>
          <td>3rd</td>
          <td>01/01/2001</td>
          <td>996784567</td>
          <td>
          <button class="btn-sm btn-success"><i class="fa fa-pencil"></i> Edit</button>
          <button class="btn-sm btn-primary"> <i class="fa fa-eye"></i> View</button>
          </td>
        </tr>
      </tbody>
    </table>
  </section>
{% endblock content %}
```

baseDashboard.css :

```
@keyframes swing {
   0% {
    transform: rotate(0deg);
  }
  10% {
    transform: rotate(10deg);
  }
  30% {
    transform: rotate(0deg);
  }
  40% {
    transform: rotate(-10deg);
  }
  50% {
    transform: rotate(0deg);
  }
  60% {
    transform: rotate(5deg);
  }
  70% {
    transform: rotate(0deg);
  }
  80% {
    transform: rotate(-5deg);
  }
  100% {
    transform: rotate(0deg);
  }
}

@keyframes sonar {
  0% {
    transform: scale(0.9);
    opacity: 1;
  }
  100% {
    transform: scale(2);
    opacity: 0;
  }
}
body {
  font-size: 0.9rem;
}
.page-wrapper .sidebar-wrapper,
.sidebar-wrapper .sidebar-brand > a,
.sidebar-wrapper .sidebar-dropdown > a:after,
.sidebar-wrapper .sidebar-menu .sidebar-dropdown .sidebar-submenu li a:before,
.sidebar-wrapper ul li a i,
.page-wrapper .page-content,
.sidebar-wrapper .sidebar-search input.search-menu,
.sidebar-wrapper .sidebar-search .input-group-text,
.sidebar-wrapper .sidebar-menu ul li a,
#show-sidebar,
#close-sidebar {
  -webkit-transition: all 0.3s ease;
  -moz-transition: all 0.3s ease;
  -ms-transition: all 0.3s ease;
  -o-transition: all 0.3s ease;
  transition: all 0.3s ease;
}

#close-sidebar {
  margin-left: 10px;
}

/*----------------page-wrapper----------------*/
```

```
.page-wrapper {
  height: 100vh;
}

.page-wrapper .theme {
  width: 40px;
  height: 40px;
  display: inline-block;
  border-radius: 4px;
  margin: 2px;
}

.page-wrapper .theme.chiller-theme {
  background: #1e2229;
}

/*----------------toggeled sidebar----------------*/

.page-wrapper.toggled .sidebar-wrapper {
  left: 0px;
}

@media screen and (min-width: 768px) {
  .page-wrapper.toggled .page-content {
    padding-left: 300px;
  }
}
/*----------------show sidebar button----------------*/
#show-sidebar {
  position: fixed;
  left: 0;
  top: 10px;
  border-radius: 0 4px 4px 0px;
  width: 35px;
  transition-delay: 0.3s;
}
.page-wrapper.toggled #show-sidebar {
  left: -40px;
}
/*----------------sidebar-wrapper----------------*/

.sidebar-wrapper {
  width: 260px;
  height: 100%;
  max-height: 100%;
  position: fixed;
  top: 0;
  left: -300px;
  z-index: 999;
}

.sidebar-wrapper ul {
  list-style-type: none;
  padding: 0;
  margin: 0;
}

.sidebar-wrapper a {
  text-decoration: none;
}

/*----------------sidebar-content----------------*/
```

```
.sidebar-content {
  max-height: calc(100% - 30px);
  height: calc(100% - 30px);
  overflow-y: auto;
  position: relative;
}

.sidebar-content.desktop {
  overflow-y: hidden;
}

/*--------------------sidebar-brand----------------------*/

.sidebar-wrapper .sidebar-brand {
  padding: 10px 20px;
  display: flex;
  align-items: center;
}

.sidebar-wrapper .sidebar-brand > a {
  text-transform: uppercase;
  font-weight: bold;
  flex-grow: 1;
}

.sidebar-wrapper .sidebar-brand #close-sidebar {
  cursor: pointer;
  font-size: 20px;
}
/*--------------------sidebar-header----------------------*/

.sidebar-wrapper .sidebar-header {
  padding: 20px;
  overflow: hidden;
}

.sidebar-wrapper .sidebar-header .user-pic {
  float: left;
  width: 60px;
  padding: 2px;
  border-radius: 12px;
  margin-right: 15px;
  overflow: hidden;
}

.sidebar-wrapper .sidebar-header .user-pic img {
  object-fit: cover;
  height: 100%;
  width: 100%;
}

.sidebar-wrapper .sidebar-header .user-info {
  float: left;
}

.sidebar-wrapper .sidebar-header .user-info > span {
  display: block;
}

.sidebar-wrapper .sidebar-header .user-info .user-role {
  font-size: 12px;
}

.sidebar-wrapper .sidebar-header .user-info .user-status {
  font-size: 11px;
  margin-top: 4px;
}

.sidebar-wrapper .sidebar-header .user-info .user-status i {
  font-size: 8px;
  margin-right: 4px;
  color: #5cb85c;
}

/*----------------------sidebar-search-----------------------*/

.sidebar-wrapper .sidebar-search > div {
  padding: 10px 20px;
}

/*---------------------sidebar-menu------------------------*/

.sidebar-wrapper .sidebar-menu {
  padding-bottom: 10px;
}

.sidebar-wrapper .sidebar-menu .header-menu span {
  font-weight: bold;
  font-size: 14px;
  padding: 15px 20px 5px 20px;
  display: inline-block;
}

.sidebar-wrapper .sidebar-menu ul li a {
  display: inline-block;
  width: 100%;
  text-decoration: none;
  position: relative;
  padding: 8px 30px 8px 20px;
}

.sidebar-wrapper .sidebar-menu ul li a i {
  margin-right: 10px;
  font-size: 12px;
  width: 30px;
  height: 30px;
  line-height: 30px;
  text-align: center;
  border-radius: 4px;
}


.sidebar-wrapper .sidebar-menu .sidebar-dropdown .sidebar-submenu ul {
  padding: 5px 0;
}

.sidebar-wrapper .sidebar-menu .sidebar-dropdown .sidebar-submenu li {
  padding-left: 25px;
  font-size: 13px;
}

.sidebar-wrapper .sidebar-menu .sidebar-dropdown .sidebar-submenu li a:before {
  content: "\f111";
  font-family: "Font Awesome 5 Free";
  font-weight: 400;
  font-style: normal;
  display: inline-block;
  text-align: center;
  text-decoration: none;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  margin-right: 10px;
  font-size: 8px;
}

.sidebar-wrapper .sidebar-menu ul li a span.label,
.sidebar-wrapper .sidebar-menu ul li a span.badge {
  float: right;
  margin-top: 8px;
  margin-left: 5px;
}

.sidebar-wrapper .sidebar-menu .sidebar-dropdown .sidebar-submenu li a .badge,
.sidebar-wrapper .sidebar-menu .sidebar-dropdown .sidebar-submenu li a .label {
  float: right;
  margin-top: 0px;
}

.sidebar-wrapper .sidebar-menu .sidebar-submenu {
  display: none;
}

.sidebar-wrapper .sidebar-menu .sidebar-dropdown.active > a:after {
  transform: rotate(90deg);
  right: 17px;
}

/*--------------------------side-footer------------------------------*/

.sidebar-footer {
  position: absolute;
  width: 100%;
  bottom: 0;
  display: flex;
}

.sidebar-footer > a {
  flex-grow: 1;
  text-align: center;
  height: 30px;
  line-height: 30px;
  position: relative;
}

.sidebar-footer > a .notification {
  position: absolute;
  top: 0;
}

.badge-sonar {
  display: inline-block;
  background: #980303;
  border-radius: 50%;
  height: 8px;
  width: 8px;
  position: absolute;
  top: 0;
}

.badge-sonar:after {
  content: "";
  position: absolute;
  top: 0;
  left: 0;
  border: 2px solid #980303;
  opacity: 0;
  border-radius: 50%;
  width: 100%;
  height: 100%;
  animation: sonar 1.5s infinite;
}

/*-------------------------page-content----------------------------*/

.page-wrapper .page-content {
  display: inline-block;
  width: 100%;
  padding-left: 0px;
  padding-top: 20px;
}

.page-wrapper .page-content > div {
  padding: 20px 40px;
}

.page-wrapper .page-content {
  overflow-x: hidden;
}

/*----------------------------chiller-theme------------------------------------------------*/

.chiller-theme .sidebar-wrapper {
    background: #3F4D67;
}
```

```
}

.chiller-theme .sidebar-wrapper .sidebar-header,
.chiller-theme .sidebar-wrapper .sidebar-search,
.chiller-theme .sidebar-wrapper .sidebar-menu {
   border-top: 1px solid #3F4D67;
}

.chiller-theme .sidebar-wrapper .sidebar-search input.search-menu,
.chiller-theme .sidebar-wrapper .sidebar-search .input-group-text {
   border-color: transparent;
   box-shadow: none;
}

.chiller-theme .sidebar-wrapper .sidebar-header .user-info .user-role,
.chiller-theme .sidebar-wrapper .sidebar-header .user-info .user-status,
.chiller-theme .sidebar-wrapper .sidebar-search input.search-menu,
.chiller-theme .sidebar-wrapper .sidebar-search .input-group-text,
.chiller-theme .sidebar-wrapper .sidebar-brand>a,
.chiller-theme .sidebar-wrapper .sidebar-menu ul li a,
.chiller-theme .sidebar-footer>a {
   color: #bcc5cf;
}

.chiller-theme .sidebar-wrapper .sidebar-menu ul li:hover>a,
.chiller-theme .sidebar-wrapper .sidebar-menu .sidebar-dropdown.active>a,
.chiller-theme .sidebar-wrapper .sidebar-header .user-info,
.chiller-theme .sidebar-wrapper .sidebar-brand>a:hover,
.chiller-theme .sidebar-footer>a:hover i {
   color: #b8bfce;
}

.page-wrapper.chiller-theme.toggled #close-sidebar {
   color: #bdbdbd;
}

.page-wrapper.chiller-theme.toggled #close-sidebar:hover {
   color: #ffffff;
}

.chiller-theme .sidebar-wrapper ul li:hover a i,
.chiller-theme .sidebar-wrapper .sidebar-dropdown .sidebar-submenu li a:hover:before,
.chiller-theme .sidebar-wrapper .sidebar-search input.search-menu:focus+span,
.chiller-theme .sidebar-wrapper .sidebar-menu .sidebar-dropdown.active a i {
   color: #16c7ff;
   text-shadow:0px 0px 10px rgba(22, 199, 255, 0.5);
}

.chiller-theme .sidebar-wrapper .sidebar-menu ul li a i,
.chiller-theme .sidebar-wrapper .sidebar-menu .sidebar-dropdown div,
.chiller-theme .sidebar-wrapper .sidebar-search input.search-menu,
.chiller-theme .sidebar-wrapper .sidebar-search .input-group-text {
   background: #333F54;
}

.chiller-theme .sidebar-wrapper .sidebar-menu .header-menu span {
   color: #6c7b88;
}

.chiller-theme .sidebar-footer {
   background: #333F54;
   box-shadow: 0px -1px 5px #282c33;
   border-top: 1px solid #464a52;
}
```

```
  .chiller-theme .sidebar-footer>a:first-child {
      border-left: none;
  }

  .chiller-theme .sidebar-footer>a:last-child {
      border-right: none;
  }

  .input-group {
    background: #333F54;
  }

  .sidebar-menu .active {
    border-left: 4px solid #1DC4E9;
    background: #333F54;
  }

#university-dashboard .input-group, #university-dashboard  .input-group-text {
    background-color: white;
}
```

dashboard.css :

```
button {
    border: none;
}
.stats .col{
    display: flex;
    align-items: center;
    flex-direction: column;
    justify-content: center;
    color: white ;
    margin: 10px;
    padding: 15px;
    border-radius: 10px;
    text-align: center;
    justify-content: space-around;
    position: relative;
    overflow: hidden;
}
.stats h2{
    font-weight: 550;
}
.stats .col i {
    margin-bottom: 5px;
    font-size: 25px;
}
.stats .col:nth-child(1) {
    background: linear-gradient(to bottom right, #895FF9, #21D0E1);
}
.stats .col:nth-child(4) {
    background: linear-gradient(to bottom right, #56448a, #6992f2);
}
.stats .col:nth-child(2) {
    background: linear-gradient(to bottom right, #9E35F0, #E92AE4);
}
.stats .col:nth-child(3) {
    background:  linear-gradient(to bottom right, #f7af73, #b88645);
}
.stats .col:nth-child(5) {
    background: linear-gradient(to bottom left, #30CDD9, #37D5A4 );
}

/* Stats col on hover */
.on-hover {
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background: #3F4D67;
    display: flex;
    align-items: center;
    justify-content: center;
    border-radius: 10px;
    transform: scaleX(0);
    transition: all 0.3s ease-in-out;
    color: white;
}
.stats .col:hover .on-hover {
    transform: scaleX(1);
}



/* Admin students table and filter */

#admin-students select {
    background: white;
    border: none;
    box-shadow: 2px 2px 2px 2px  #00000013;
}
.student-filter .row{
    padding: 10px 0px;
    justify-content: center;
    align-items: center;
}
```

```
}

.student-filter select {
   margin-right: 20px;
   margin: 10px auto;
   width: 100%;
   padding: 5px 10px;
   border: none;
   border-radius: 5px;
}

#search-form {
   max-width: 350px;
   border: none;
   box-shadow: 2px 2px 2px 2px  #00000013;
   border-radius: 5px;
}
#search-form input {
   border: none;
}

.ctns button {
   border: none;
   padding: 5px 10px;
}
thead {
   background: #23395f;
}

.btn-success {
   background: #41b67f;
}
.btn-primary {
   background: #6f6ff2;
```

```
}
```

main.css :

```
@import url('https://fonts.googleapis.com/css2?family=Poppins');

* {
   margin: 0;
   padding: 0;
   box-sizing: border-box;
}
body {
   background: #F4F3F6;
   font-family: 'Poppins', sans-serif;
}
/* ========= Defaults ========= */

h1,h2,h3 {
   color: #344767;
}
img {
   max-width: 100%;
}

/* Commomn */
.bar {
   width: 200px;
   height: 4px;
   background: linear-gradient(to right, #F49DA8, #FB548F );
   margin: 10px auto;
   border-radius: 10px; }
```

dashboard.js :

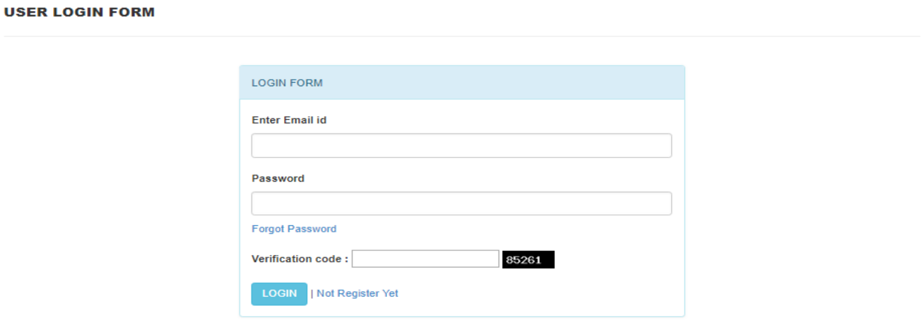
```
jQuery(function ($) {
   $(".sidebar-dropdown > a").click(function() {
  $(".sidebar-submenu").slideUp(200);
  if (
   $(this)
     .parent()
     .hasClass("active")
  ) {
   $(".sidebar-dropdown").removeClass("active");
   $(this)
     .parent()
     .removeClass("active");
  } else {
   $(".sidebar-dropdown").removeClass("active");
   $(this)
     .next(".sidebar-submenu")
     .slideDown(200);
   $(this)
     .parent()
     .addClass("active");
  }
});

$("#close-sidebar").click(function() {
  $(".page-wrapper").removeClass("toggled");
});
$("#show-sidebar").click(function() {
  $(".page-wrapper").addClass("toggled");
});
});
```

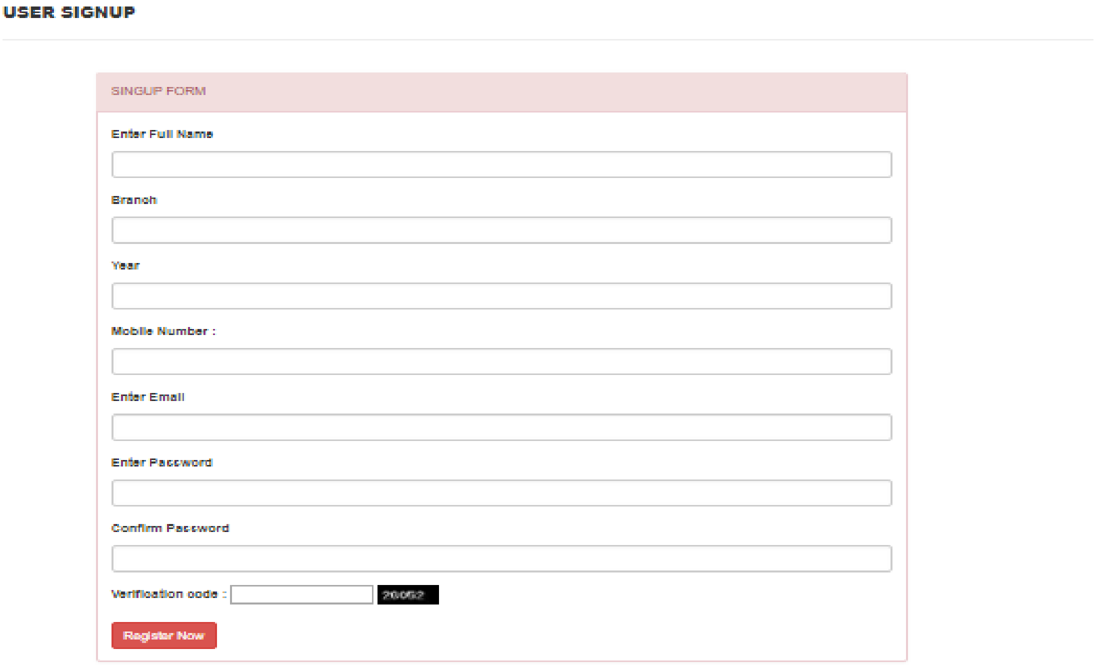
SCREENS

USER SCREENS:

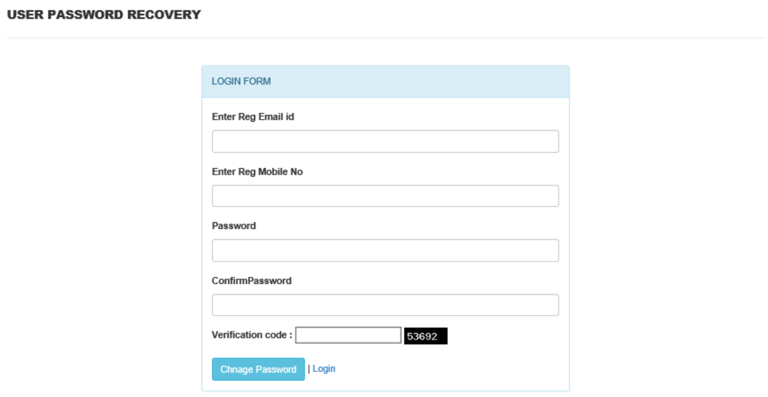
User Login:



User Signup:



User Forgot Password:



USER PASSWORD RECOVERY

LOGIN FORM

Enter Reg Email id

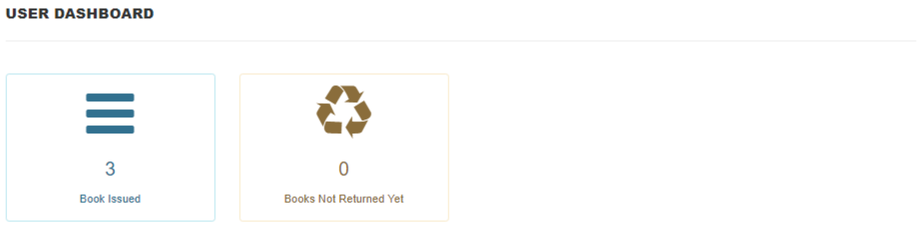
Enter Reg Mobile No

Password

ConfirmPassword

Verification code : 53692

User Dashboard:

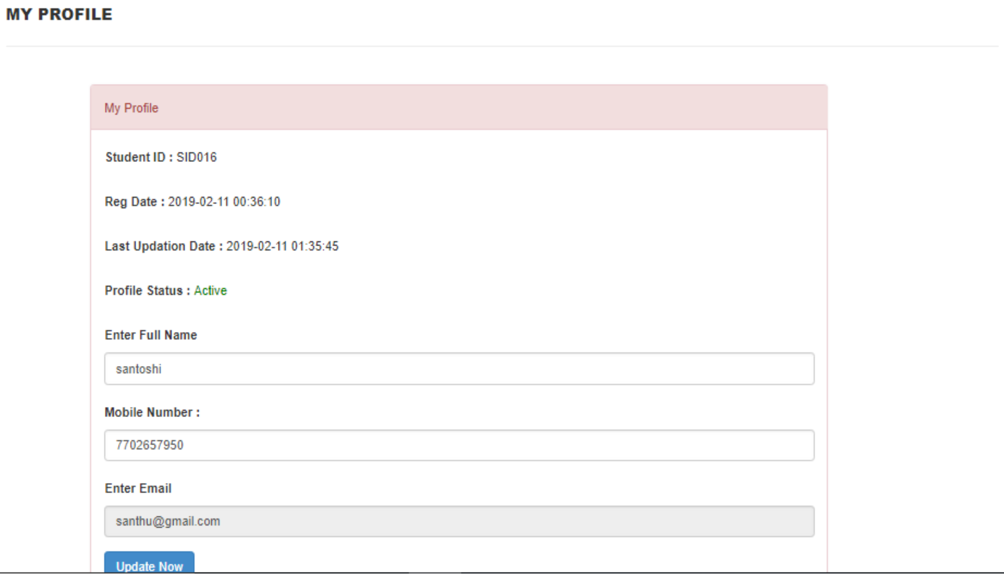


USER DASHBOARD

3
Book Issued

0
Books Not Returned Yet

User My Profile:



MY PROFILE

My Profile

Student ID : SID016

Reg Date : 2019-02-11 00:36:10

Last Updation Date : 2019-02-11 01:35:45

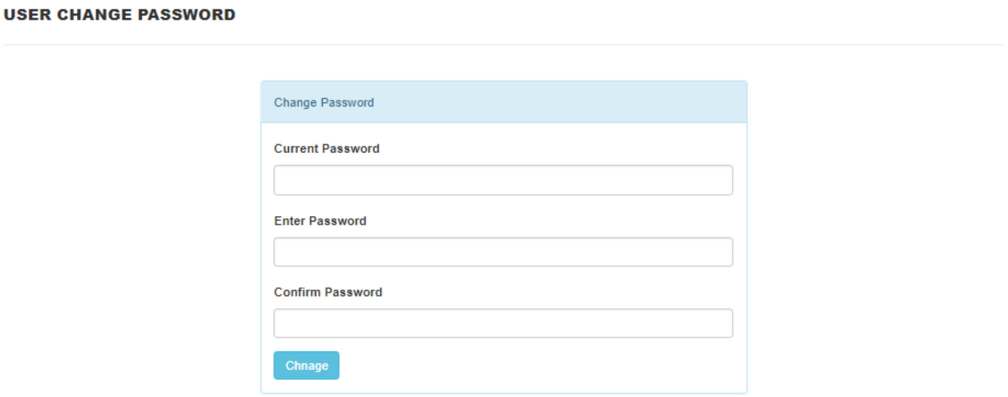
Profile Status : Active

Enter Full Name: santoshi

Mobile Number : 7702657950

Enter Email: santhu@gmail.com

User Change Password:



USER CHANGE PASSWORD

Change Password

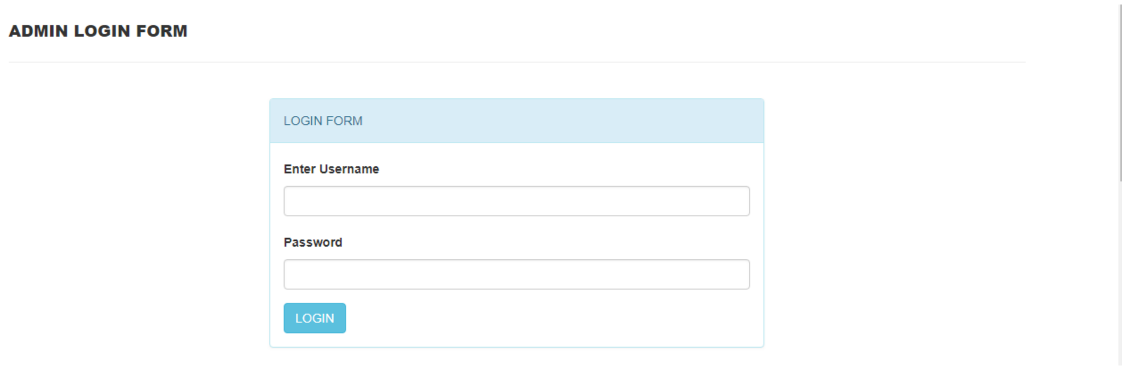
Current Password

Enter Password

Confirm Password

ADMIN SCREENS:

Login:



ADMIN LOGIN FORM

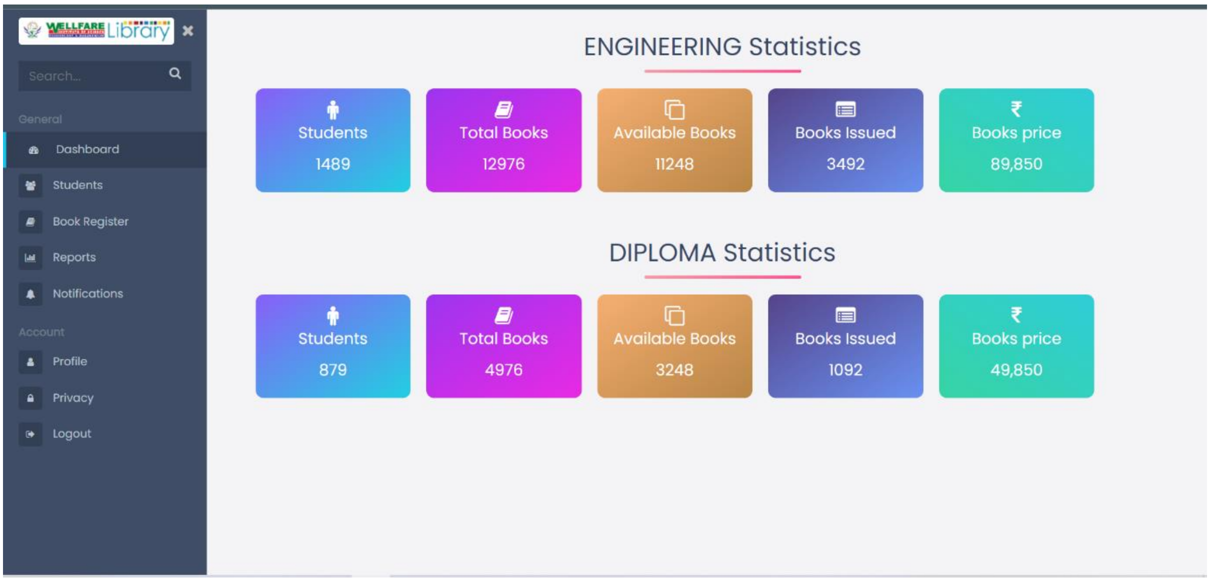
LOGIN FORM

Enter Username

Password

LOGIN

Dashboard:



ENGINEERING Statistics

Students 1489

Total Books 12976

Available Books 11248

Books Issued 3492

₹ Books price 89,850

DIPLOMA Statistics

Students 879

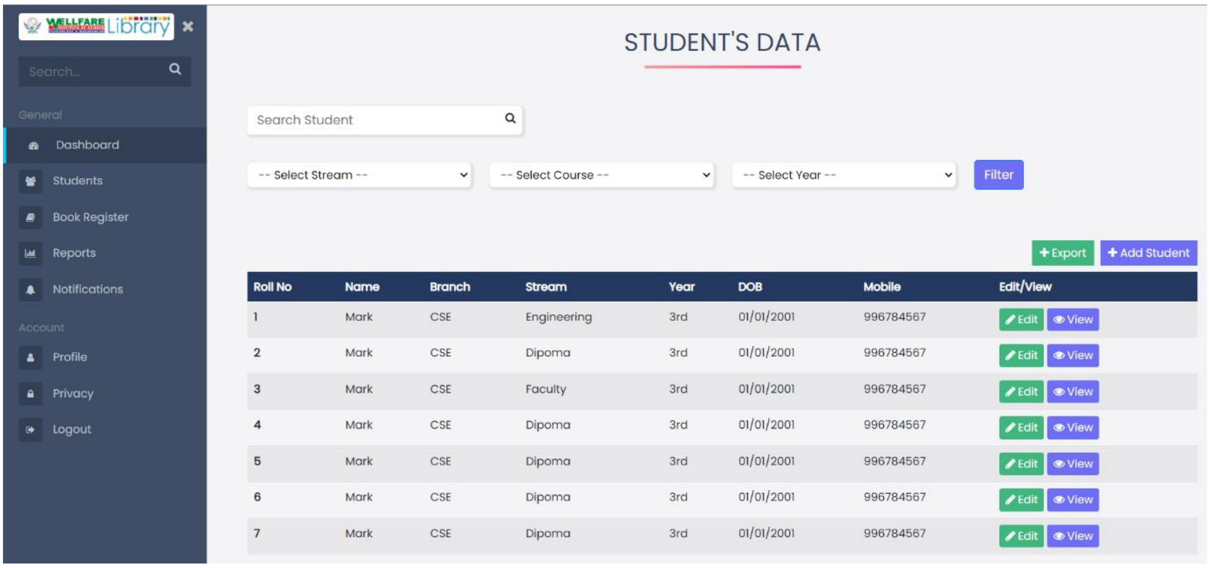
Total Books 4976

Available Books 3248

Books Issued 1092

₹ Books price 49,850

Dashboard Students :

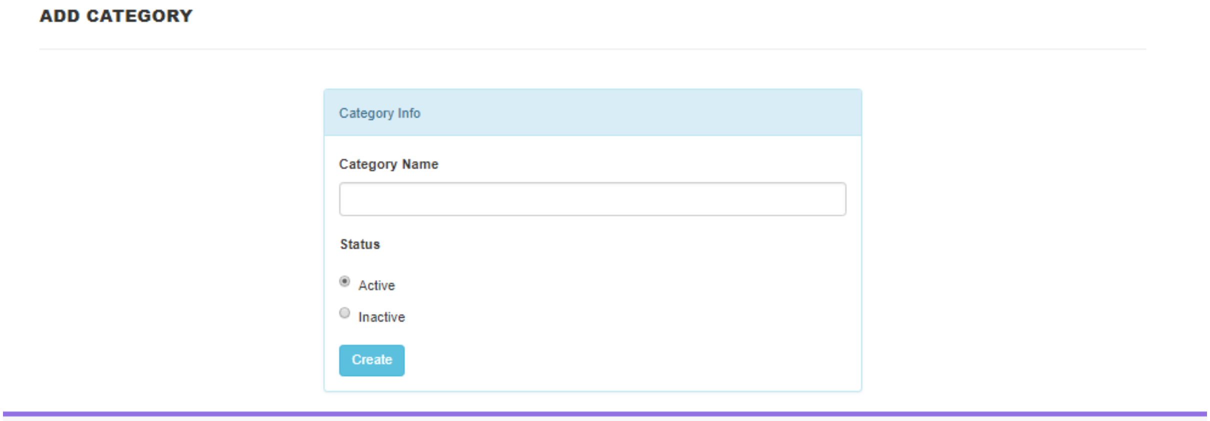


STUDENT'S DATA

Search Student

Roll No	Name	Branch	Stream	Year	DOB	Mobile	Edit/View
1	Mark	CSE	Engineering	3rd	01/01/2001	996784567	Edit View
2	Mark	CSE	Dipoma	3rd	01/01/2001	996784567	Edit View
3	Mark	CSE	Faculty	3rd	01/01/2001	996784567	Edit View
4	Mark	CSE	Dipoma	3rd	01/01/2001	996784567	Edit View
5	Mark	CSE	Dipoma	3rd	01/01/2001	996784567	Edit View
6	Mark	CSE	Dipoma	3rd	01/01/2001	996784567	Edit View
7	Mark	CSE	Dipoma	3rd	01/01/2001	996784567	Edit View

Add Category:


ADD CATEGORY

Category Info

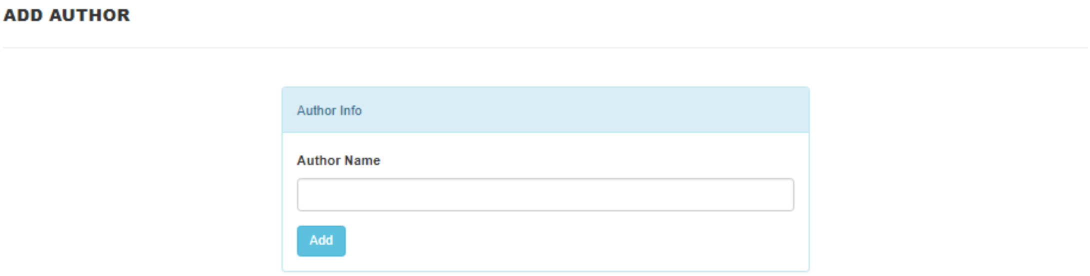
Category Name

Status

● Active

○ Inactive

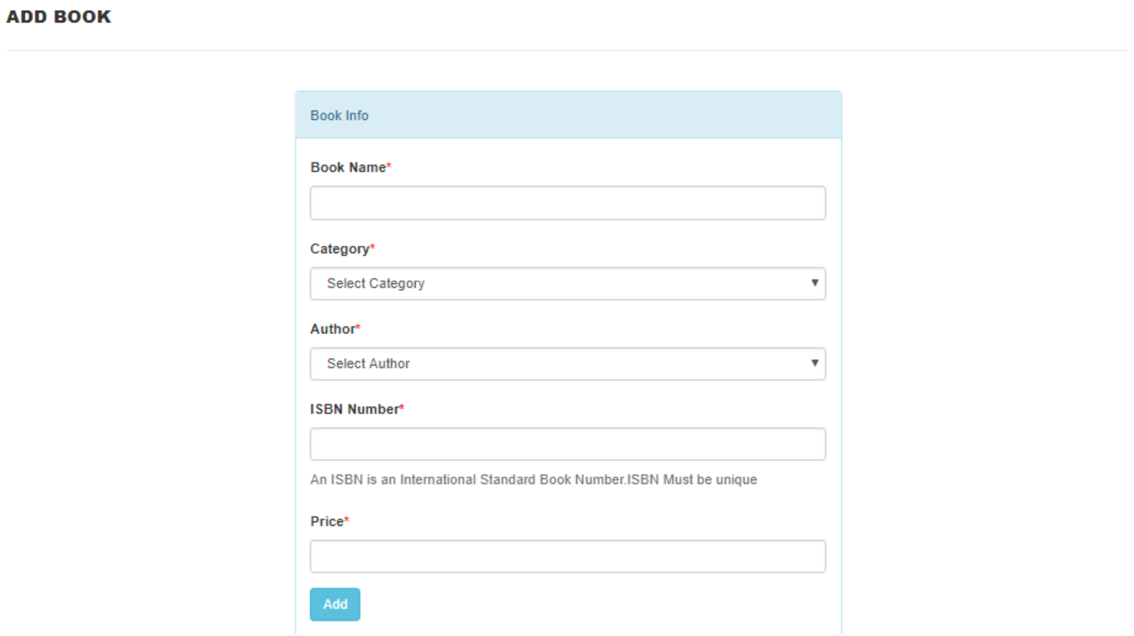
Add Author:


ADD AUTHOR

Author Info

Author Name

Add Book:



ADD BOOK

Book Info

Book Name*

Category*

Select Category ▾

Author*

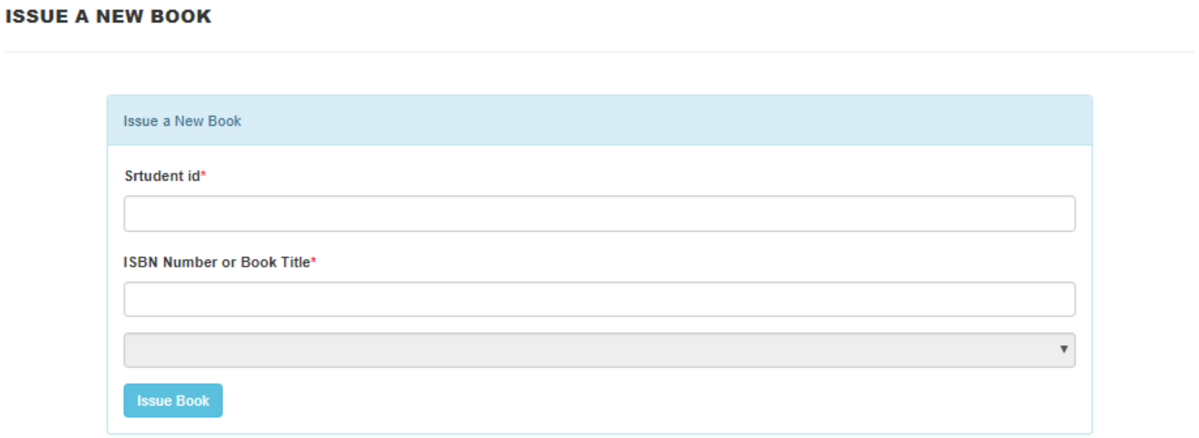
Select Author ▾

ISBN Number*

An ISBN is an International Standard Book Number.ISBN Must be unique

Price*

Issue a Book:



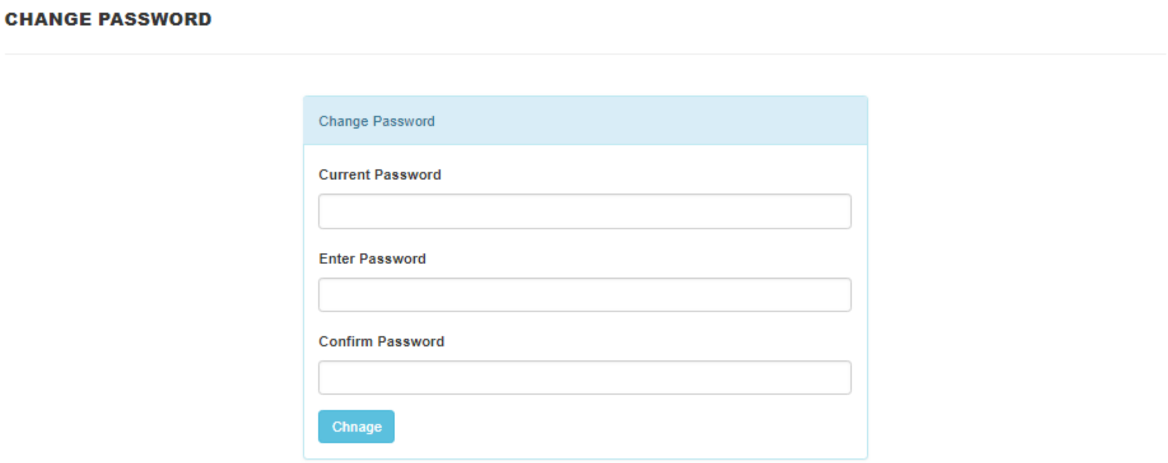
ISSUE A NEW BOOK

Issue a New Book

Srtudent id*

ISBN Number or Book Title*

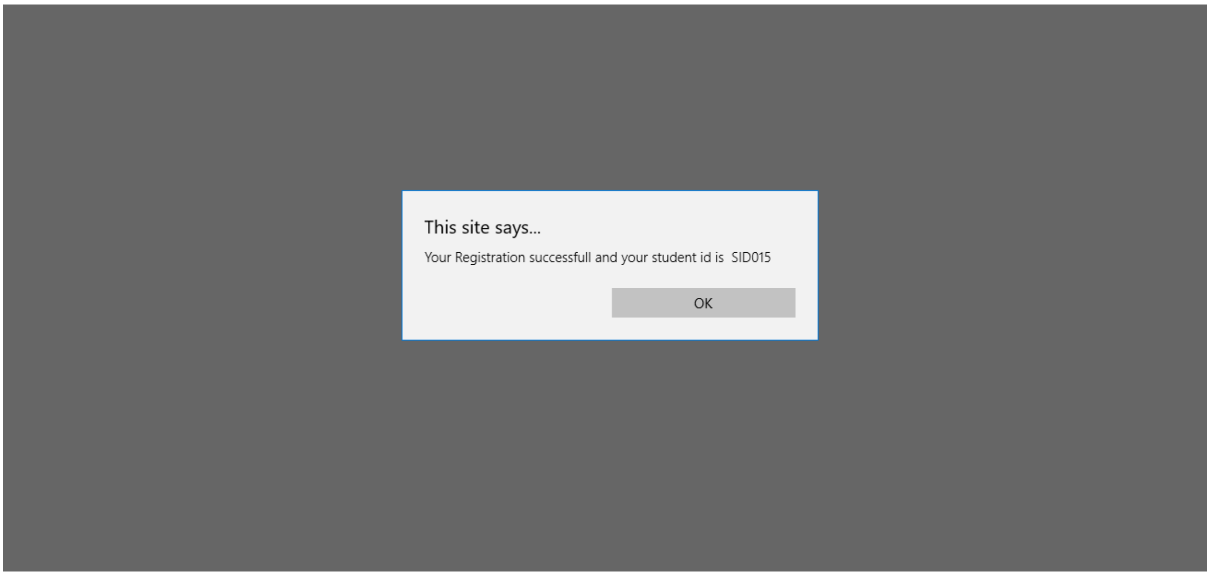
Change Password:



OUTPUT SCREENS

USER SCREENS:

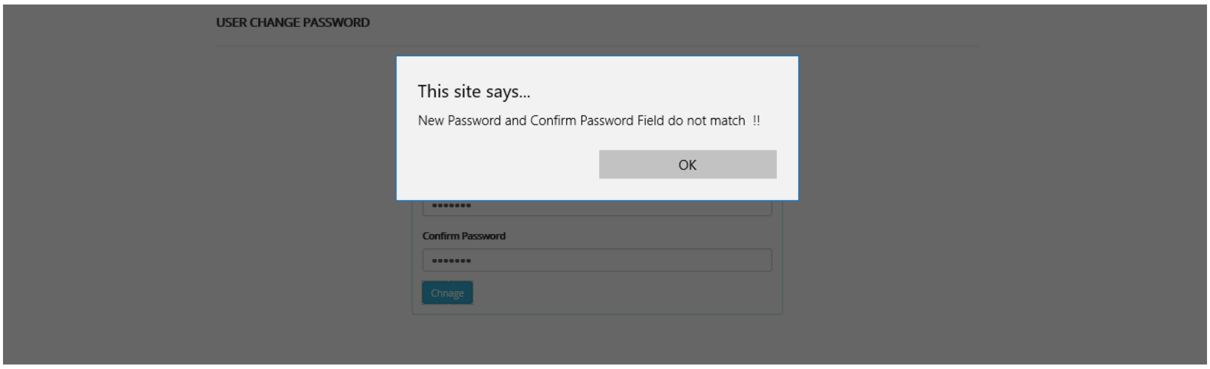
Sign up:



This site says...

Your Registration successfull and your student id is SID015

Confirm Password:



USER CHANGE PASSWORD

This site says...

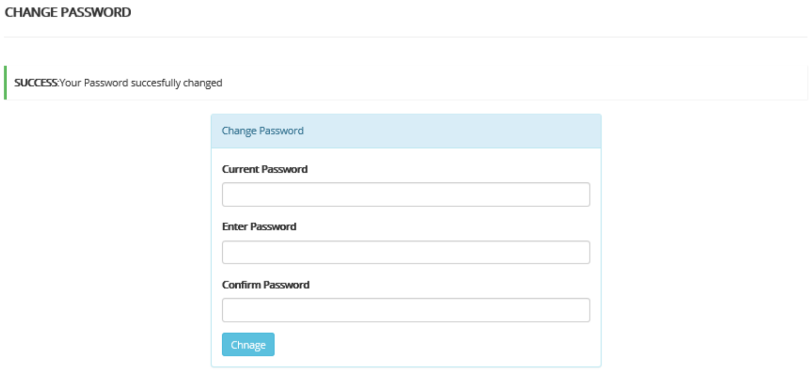
New Password and Confirm Password Field do not match !!

•••••••

Confirm Password

•••••••

Change Password:



CHANGE PASSWORD

SUCCESS:Your Password succesfully changed

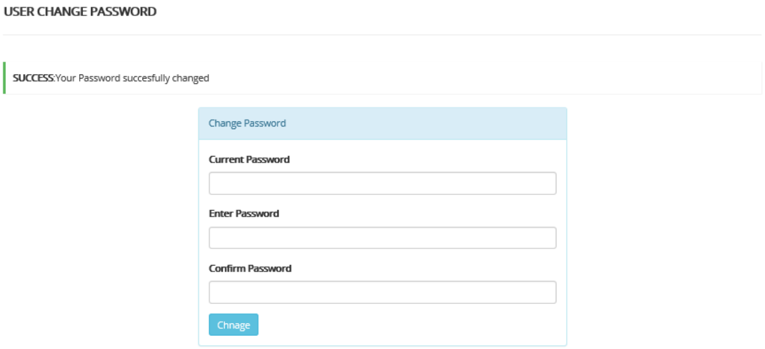
Change Password

Current Password

Enter Password

Confirm Password

Manage Issue Book:



USER CHANGE PASSWORD

SUCCESS:Your Password succesfully changed

Change Password

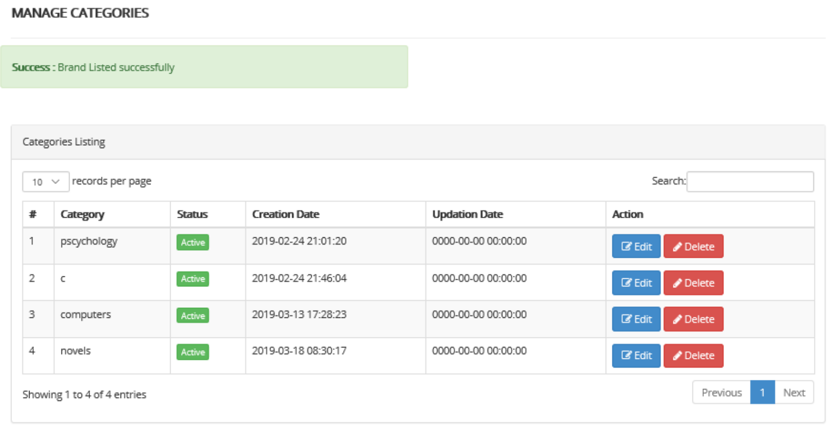
Current Password

Enter Password

Confirm Password

ADMIN SCREENS:

Manage Category:



MANAGE CATEGORIES

Success : Brand Listed successfully

Categories Listing

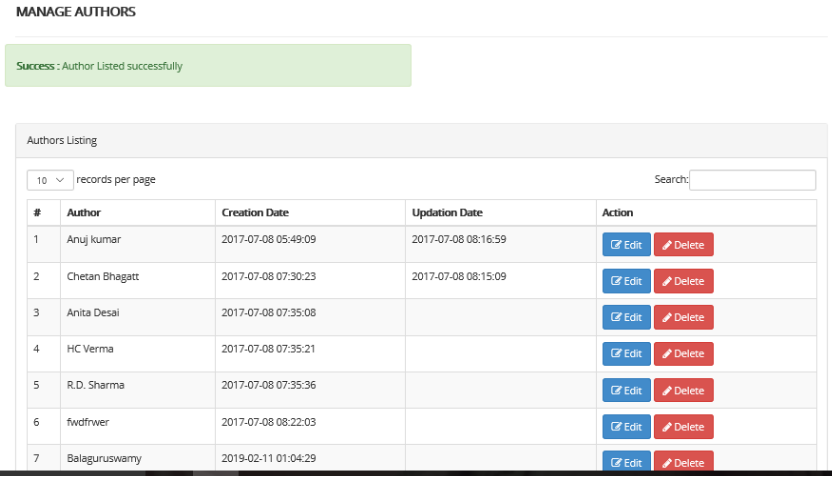
10 records per page

Search:

#	Category	Status	Creation Date	Updation Date	Action
1	pscychology	Active	2019-02-24 21:01:20	0000-00-00 00:00:00	Edit Delete
2	c	Active	2019-02-24 21:46:04	0000-00-00 00:00:00	Edit Delete
3	computers	Active	2019-03-13 17:28:23	0000-00-00 00:00:00	Edit Delete
4	novels	Active	2019-03-18 08:30:17	0000-00-00 00:00:00	Edit Delete

Showing 1 to 4 of 4 entries

Manage Author:



MANAGE AUTHORS

Success : Author Listed successfully

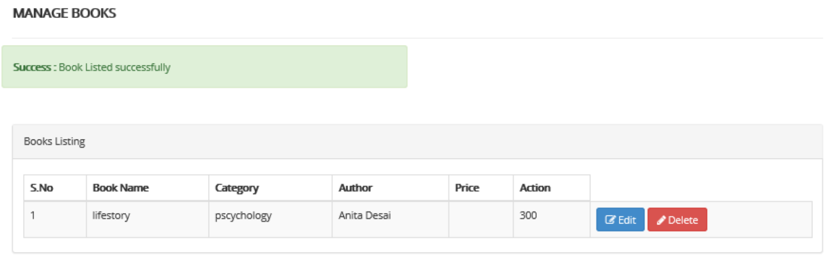
Authors Listing

10 records per page

Search:

#	Author	Creation Date	Updation Date	Action
1	Anuj kumar	2017-07-08 05:49:09	2017-07-08 08:16:59	Edit Delete
2	Chetan Bhagatt	2017-07-08 07:30:23	2017-07-08 08:15:09	Edit Delete
3	Anita Desai	2017-07-08 07:35:08		Edit Delete
4	HC Verma	2017-07-08 07:35:21		Edit Delete
5	R.D. Sharma	2017-07-08 07:35:36		Edit Delete
6	fwdfrwer	2017-07-08 08:22:03		Edit Delete
7	Balaguruswamy	2019-02-11 01:04:29		Edit Delete

Manage Book:



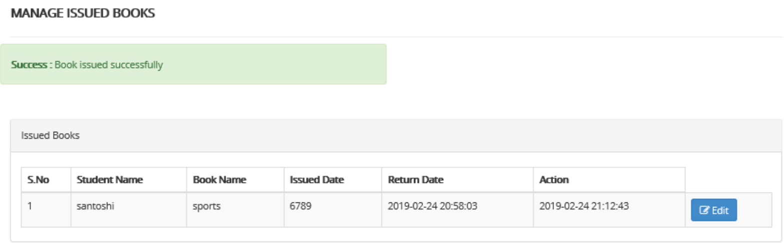
MANAGE BOOKS

Success : Book Listed successfully

Books Listing

S.No	Book Name	Category	Author	Price	Action	
1	lifestory	pscychology	Anita Desai		300	Edit Delete

Manage Issue Book:



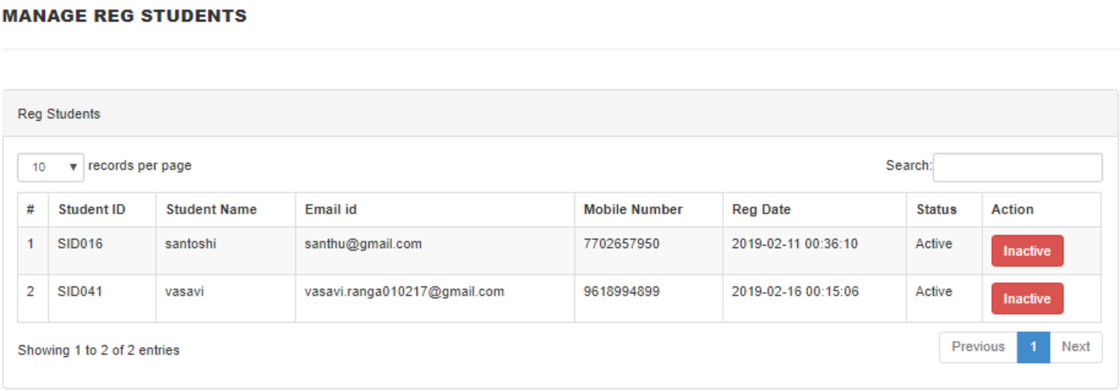
MANAGE ISSUED BOOKS

Success : Book issued successfully

Issued Books

S.No	Student Name	Book Name	Issued Date	Return Date	Action	
1	santoshi	sports	6789	2019-02-24 20:58:03	2019-02-24 21:12:43	Edit

Manage Registerd Students:



MANAGE REG STUDENTS

Reg Students

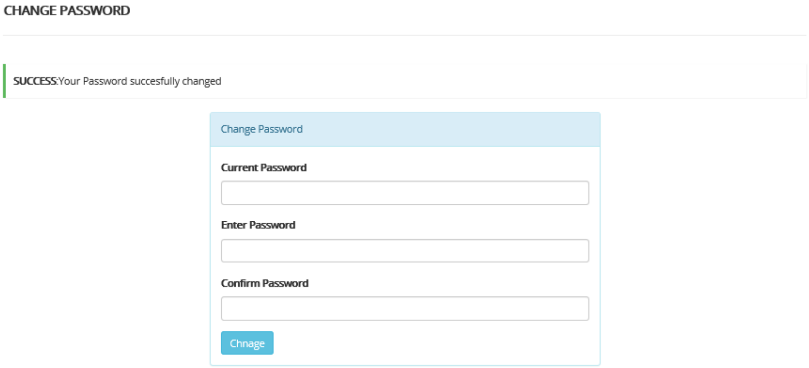
10 records per page

Search:

#	Student ID	Student Name	Email id	Mobile Number	Reg Date	Status	Action
1	SID016	santoshi	santhu@gmail.com	7702657950	2019-02-11 00:36:10	Active	Inactive
2	SID041	vasavi	vasavi.ranga010217@gmail.com	9618994899	2019-02-16 00:15:06	Active	Inactive

Showing 1 to 2 of 2 entries

Change Password:



CHANGE PASSWORD

SUCCESS: Your Password succesfully changed

Change Password

Current Password

Enter Password

Confirm Password

CONCLUSION:

The Library Management System allows the user to store the book details and the persons details. This software allows storing the details of all the data related to library. The implementation of the system will reduce data entry time and provide readily calculated reports.