# IWT Assignment 2 - LMS

## Prepared by:

- Mahesh Kumar - 34
- Raghu Buchala - 40
- Ram Charan - 41
- Sai Vishal - 48
- Shyam Sundar - 52
- Siddhardha - 53

# ACKNOWLEDGEMENT

We would like to express our deepest appreciation to all those who provided us with the possibility to complete this project. Special gratitude to our project mentor, **Mr. Srikanth Sir** whose contribution in stimulating suggestions and encouragement, helped us to coordinate our project especially in writing this report. Furthermore, we would also like to acknowledge with much appreciation the department of CSE to give us this humble opportunity. A special thanks to our family members and friends for the support and encouragement they have given to us over the course.

# ABSTRACT


## Library Management System – Python Django Project

A College Library management is a project that manages and stores books information electronically according to student needs along with the student information. This helps both students and library managers to keep a constant track of all the books available in the library. It allows both the admin and the student to search for the desired book. It becomes necessary for colleges to continuously check the books issued and returned and even calculate fines. This task if carried out manually includes chances of mistakes. These errors are avoided by allowing the system to keep track of information such as issue date, last date to return the book and even fine information and thus there is no need to keep manual track of this information which avoids the chances of mistakes.

This project aims to create an interactive library management system using python Django as a base programming language, HTML and CSS for graphical user interface and files for data management.

# INTRODUCTION

The library management system is a project written in python programming language using Django. This project is written to simulate the role of a librarian in a library. The features of this project are it can keep track of all the books available in the library. It has a very interactive and simple user interface that both student and admin can use to receive or return books and admin to monitor the process and make sure everything goes correctly.

## Project aims:

The project aims and objectives that will be achieved after completion of this project are:

- A database containing the list of all books in the library.
- Facility to mark book either as available or issued.
- Add various books to the library database.
- To store the information regarding the author's name and book id.
- Facility to issue an available book to students.
- To get the student id while the student borrows a book.
- Facility to remove books from the system.
- Facility to allow students to return the book to the library.

# SYSTEM REQUIREMENTS:

## Software Requirements:

Python 3.9 IDLE or PyCharm or VS Code
Django -Latest Version
Windows (7 or above) or Mac OS (10.11 or higher)

## Hardware Requirements:

Processor: Intel I3 or above
Disk space: 1 GB or more

## Existing system vs Proposed system:

Early days Libraries are managed manually. It required a lot of time to record or retrieve the details. The employees who have to record the details must perform their job very carefully. Even a small mistake would create a lot of problems. Security of information is very less. Report generations of all the information are a very tough task.

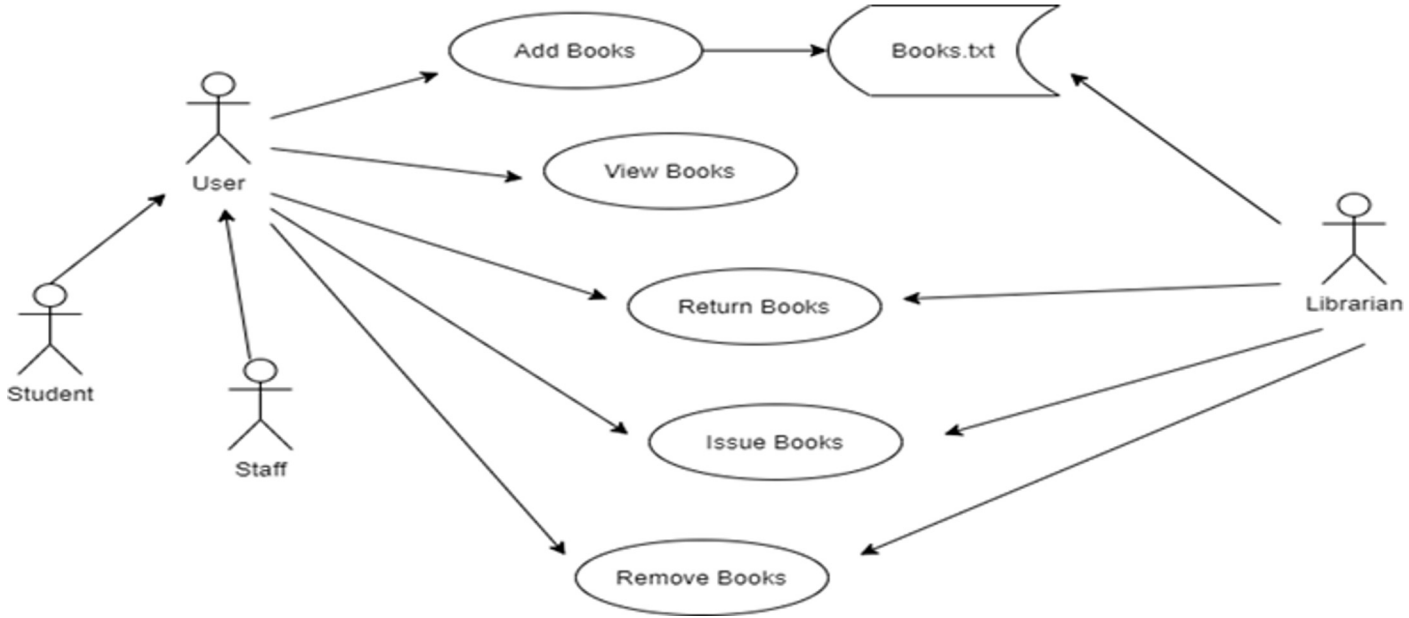
Maintenance of the Library catalogue and arrangement of the books to the catalogue is a very complex task. In addition to its maintenance of member details, issue dates and return dates etc. manually is a complex task.

All the operations must be performed perfectly for the maintenance of the library without any degradation which may finally fail the entire system.

To solve the inconveniences as mentioned in the existing system, a Library Management System is proposed. The proposed system contains the following features:

- A single file-based management system that holds the record of the book id, book name, author and whether the book is available or has been issued.
- A function to add new books to the library database.
- A function to remove books from the database.
- A function that issues the required book to a student depending on its availability.
- A function that manages the return of borrowed books from the students.
- Pop up errors if there is any fallacy in book id or student id

# Methodology:



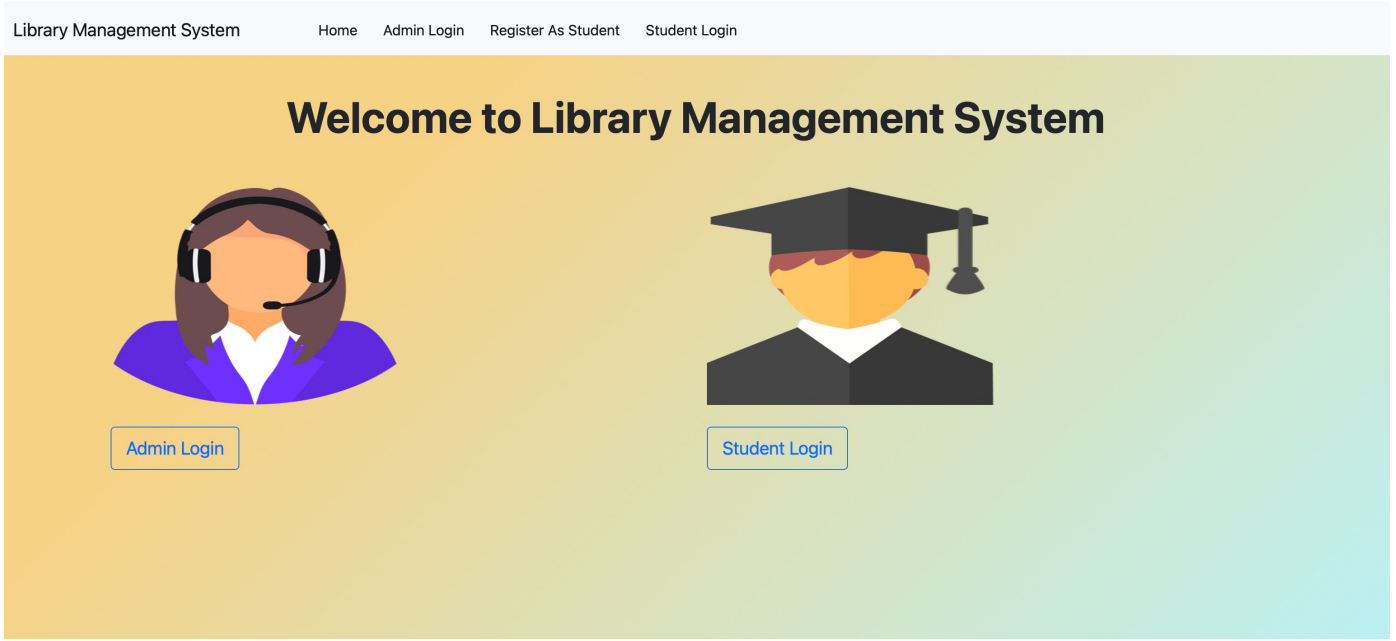
The above level diagram is the functioning of the Library Management System in a very lucid way. The user can either be a staff or a student. The system will provide view functionality to view different books available in the library. Further, the library staff person can add/update the resources. The users of the system can request to issue/return the book for which they have to follow certain criteria.

## Sample Code

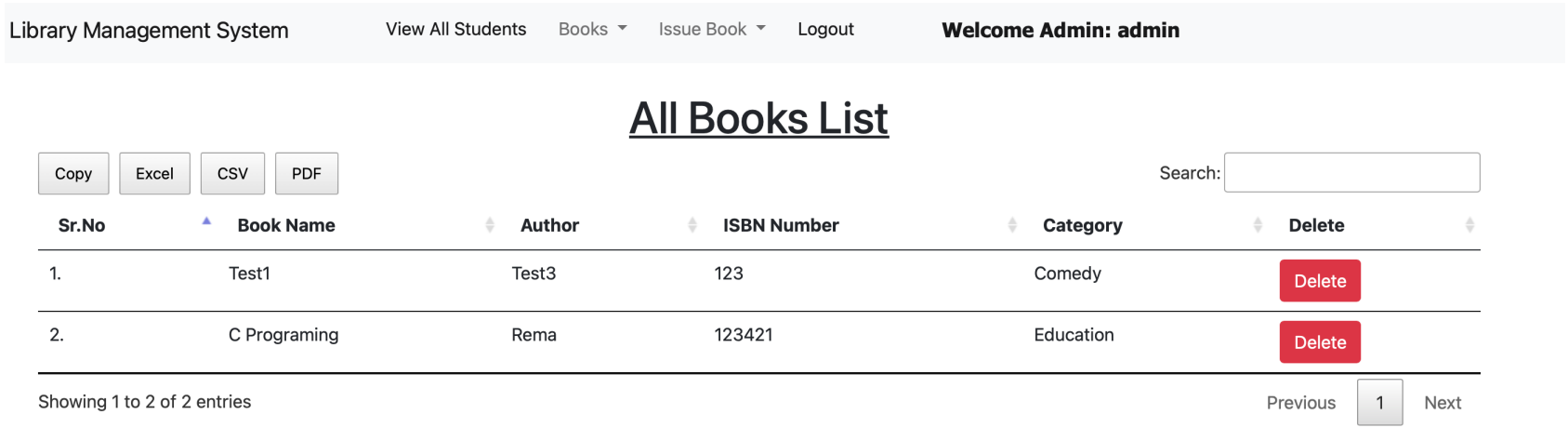
```
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">

<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">

<title>{% block title %} {% endblock %}</title>
{% block css %}
{% endblock %}
</head>
<body>
<nav class="navbar navbar-expand-lg navbar-light bg-light sticky-top">
<div class="container-fluid">
<a class="navbar-brand" href="#">Library Management System</a>
<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarSupportedContent">
                 
<ul class="navbar-nav me-auto mb-2 mb-lg-0">
<li class="nav-item">
<a class="nav-link active" aria-current="page" href="/">Home</a>
</li>
   
<li class="nav-item">
<a class="nav-link active" href="/admin_login/">Admin Login</a>
</li>
   
<li class="nav-item">
<a class="nav-link active" href="/student_registration/">Register As Student</a>
</li>
   
<li class="nav-item">
<a class="nav-link active" href="/student_login/">Student Login</a>
</li>
</ul>
</div>
</div>
</nav>
{% block body %}
{% endblock %}
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/js/bootstrap.bundle.min.js" integrity="sha384-U1DAWAznBHeqEIlVSCgzq+c9gqGAn5c/t99JyeKa9xxaYpSvHU5awsuZVVFIhvj" crossorigin="anonymous"></script>
{% block js %}
{% endblock %}
</body>
</html>
```

# SCREENSHOTS OF THE PROJECT:

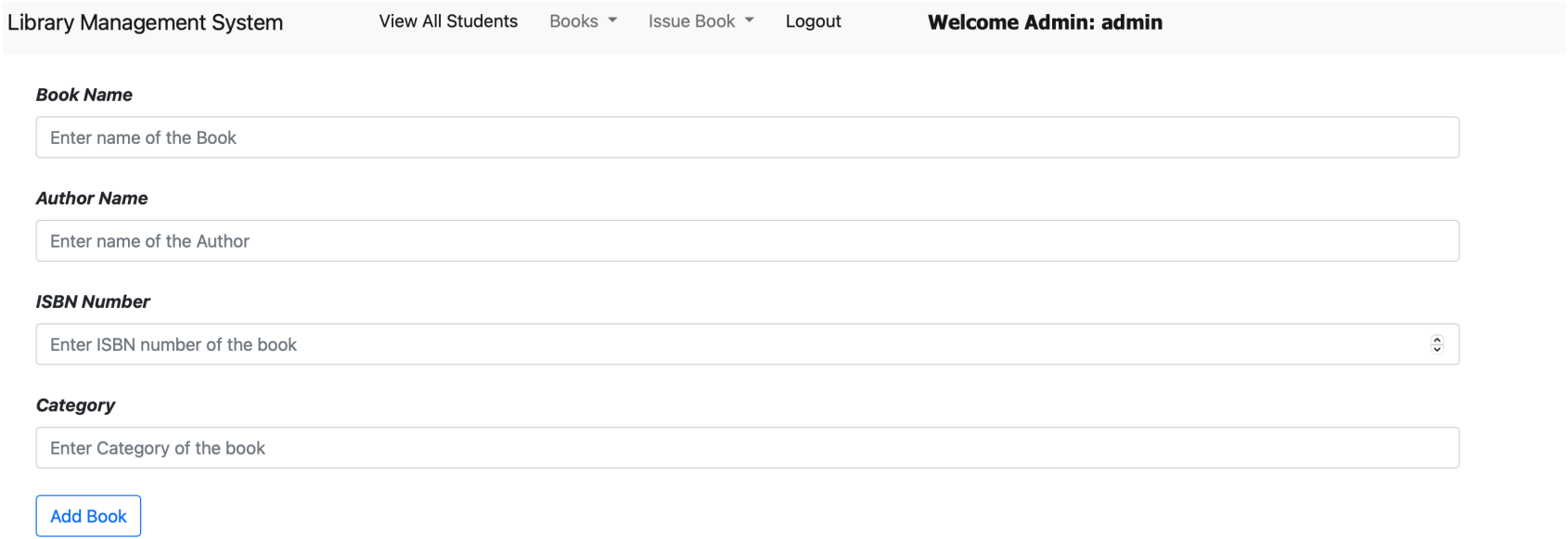
## Main window:



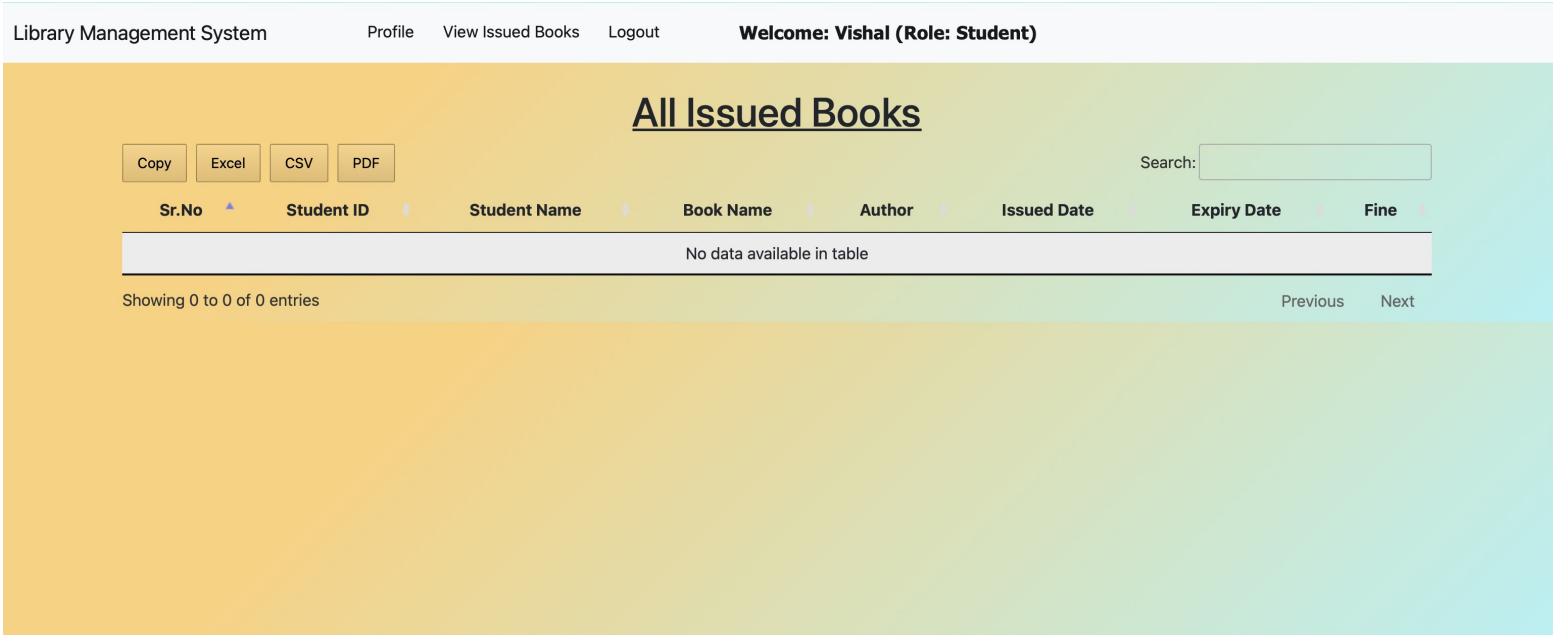
## View books window:



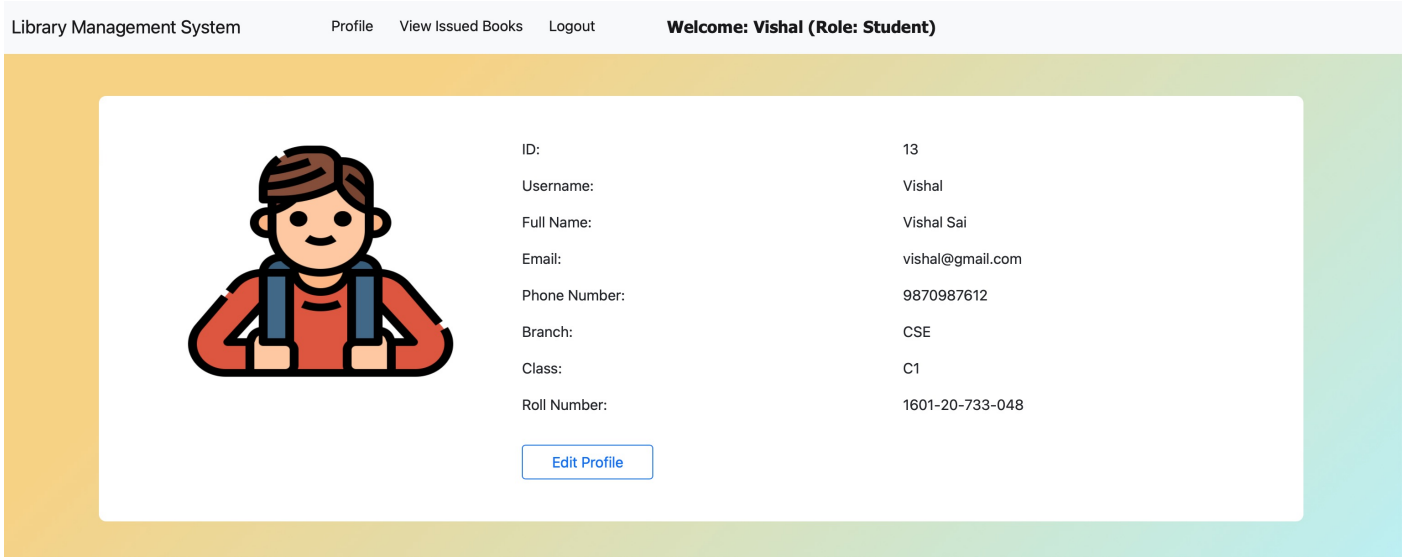
# Add book window:



# Issue book window:



# Student Login :



# FUTURE SCOPE:

- Personal Customization
- User friendly and aesthetic user interface
- Shifting the database to cloud
- E-book borrowing section

# REPORT AND ANALYSIS:

The above model of the Library Management System makes use of various modules to function as a system with integrity. In such programs, the extent of integration of various modules is a parameter defining the susceptibility of the whole system. However, the above system was observed to be versatile and the modules were integrating without any concerns, making the program run smoothly and perform various executions without any glitches. The handover of command from one module to the other when a desired function was called happened to be errorless and smoothly executed. The add books button which opens a window prompt to add the books to the database stores the data in a file in a dictionary format with the book id as the key. The issue book and return book functions throw an error when an incorrect book or student id is entered. The issue book button opens a prompt that asks the book and student id and issues the book only if it is marked as available in the database. The subtle use of message boxes makes the system more user friendly and easy to use. Although major improvements can be implemented to this system there is no hesitancy to conclude that the system is operable, user friendly and fulfils the objectives of this project.

Hereby we conclude that the Library Management System was able to accomplish all of its prerequisites successfully and also has the potential to make the process of library management computerized, relieving the workload from a librarian.