PROJECT DOCUMENTATION

LIBRARY MANAGEMENT SYSTEM

INTRODUCTION

Library Management System is a comprehensive library management solution that is suitable for both large and small libraries. Its flexible design enables Library Management System to be installed in a range of Library organizations, ranging from public libraries, through to academic, joint use and special libraries. This Library Management System Software is capable of handling Books with equal ease and efficiency. This Library Management System, utilizing the latest advancements in the Information Technology to provide and improve Library Services.

Purpose

The purpose of this project is to develop an application that will automate the whole procedure of a library. The software that would be developed should have facilities like Add / Delete Members, Add / Delete Books, Issue & Return. The application should be secured, as well as with limited access. The main requirement of the project will be the ease of use, besides being the most efficient and effective tool for the purpose. The application should be user friendly. It should be robust and scalable. An automated solution would be very beneficial to the organization, as it would bring structure to the whole process so that it can be traced for any kind of query. Also, an automated solution will lead to optimal utilization of the available resources, reducing duplication of effort, increasing efficiency and minimizing time-delays. Following are the main purpose of computerization:

- To provide services to all the employees for issue, return & search etc. at one place.
- To improve co-ordination in staff.
- To reduce paper filling work.
- To reduce chances of information leaking.

Scope

For Members: -

- Facility for search of Books based on Access id, Title, Author, Subject, Keyword.
- Facility for ISSUE / RETURN Books.
- Facility to calculate Book fine.

Overview

The rest of the document deals with all the main features of this software. It not only describes various functions but also gives details about how these functions are related to each other.

PROJECT DESCRIPTION

Product Perspective

The manual library management system includes following drawbacks:

- The existing system involves a lot of paper work and manual calculation. This has led to inconsistency and inaccuracy in the maintenance of data.
- The data, which is stored on the paper only, may be lost, stolen or destroyed due to natural calamity like fire and water.
- The existing system is sluggish and time-consuming causing inconvenience to library staff.
- Since there are many books related to different scopes thus it would be very difficult to find a specific book, or edit the data of some book.

Hence the library management system is proposed with the following Product perspective:

- The computerization of the library system will reduce a lot of paperwork and hence the load on the library staff.
- It would be easy to search a specific book.
- The machine performs all calculations for fines and all. Hence chances of error are nil.
- The system provides for user-ID validation, hence unauthorized access is prevented.

System Interfaces

Software will work on Windows OS. The database used will be an open source database like SQL Server and the system will work on Visual Studio Code 2022 and VS code

Software requirements

Operating System: we have chosen windows operating system for its best support and user friendliness.

LMS uses standard .Net core Web API and databases. The database should have backup capabilities.

Angular : version 17

.net core: 8.0

Nodejs : 18.10.0

Database: to save the records of the applicants and their details, SQL Server database is used.

.

Minimum Hardware requirements to develop this software:

Processor: i5 10Gen

RAM :16 GB

Operating system: windows 10

Operations

The basic operations of the 'Library Management System' are described as follows:

- The staff member first has to register him/herself before using the system.
- The staff member can then login into the system with his/her username and password.

- The staff member can then add, delete or update the book records within the system.
- The staff member can issue books when requesting by the members and then update the book record too.
- The system will generate the fine slip according to the returned date when book is returned by the member.
- The staff member will update the book issue and return records into the book management system.
- The member can search for a specific book by entering book information.
- The staff member can update member records within the system.

Site Adaption Requirements

The system will require an application server for the runtime components and a database for storage. The system will run on select popular application servers and use select popular database for data storage.

User characteristics

User1: Staff- Staff will add, delete or update book records within the system. He/she will issue the books as per requested by the member and will calculate the fine according to returned date of the book. Staff needs to have complete understanding of functionalities and internal processing of the system.

User2: Member- Member will request for book issue and then will return the book. They can search for a specific book. The user does not need to have complete understanding of complex functionalities and internal processing of the system.

Assumptions and Dependencies

The assumptions are:

- The coding should be error free.

- The system should be user-friendly so that it is easy to use for the users.
- The information of all users, books and libraries must be stored in a database.
- The system should have more storage capacity and provide fast access to the database.

The dependencies are:

- On the basis of listing requirements and specifications the project will be developed.
- The end users should have proper understanding of the product.
- The information of all users must be stored in a database that is accessible by the LMS.
- Any update regarding the book from the library is to be recorded to the database and the data entered should be correct.

Functional Requirements

- **Login:**

  Description: Staff member will login to the system. The user must be registered in the system before login.
  Input: Enter the username and password.
  Output: Staff will be able to use the features of software.
  Processing: Username and password will be checked by the system. If they are incorrect a message will be displayed.
- **Add/Remove books:**

  Description: The staff can add or remove book by entering details.
  Input: Enter the book detail you want to remove or add within the stock.
  Output: Confirmation of addition or deletion and update list of books available.
  Processing: The details of books must be right in order to add them else there will be problems in future.
- **Search:**

  Description: The users can search a book by entering book details such as author's name, book name etc.
  Input: Enter the details you know about the book.
  Output: The list of available books is displayed.

Processing: A message is displayed if the book related to the entered details is not available.

- **Issues book:**

  Description: The staff member checks the availability of book which the member want to get issued.
  Input: Enter book code.
  Output: Confirmation for book issue or apology for failure in issue.
  Processing: If selected book is available then the book will be issued and the record is updated else error will be displayed.
- **Return book:**

  Description: The member wants to return the book.
  Input: return the book to the library.
  Output: The record will be updated.
  Processing: If book is not returned on the time then fine is calculated.

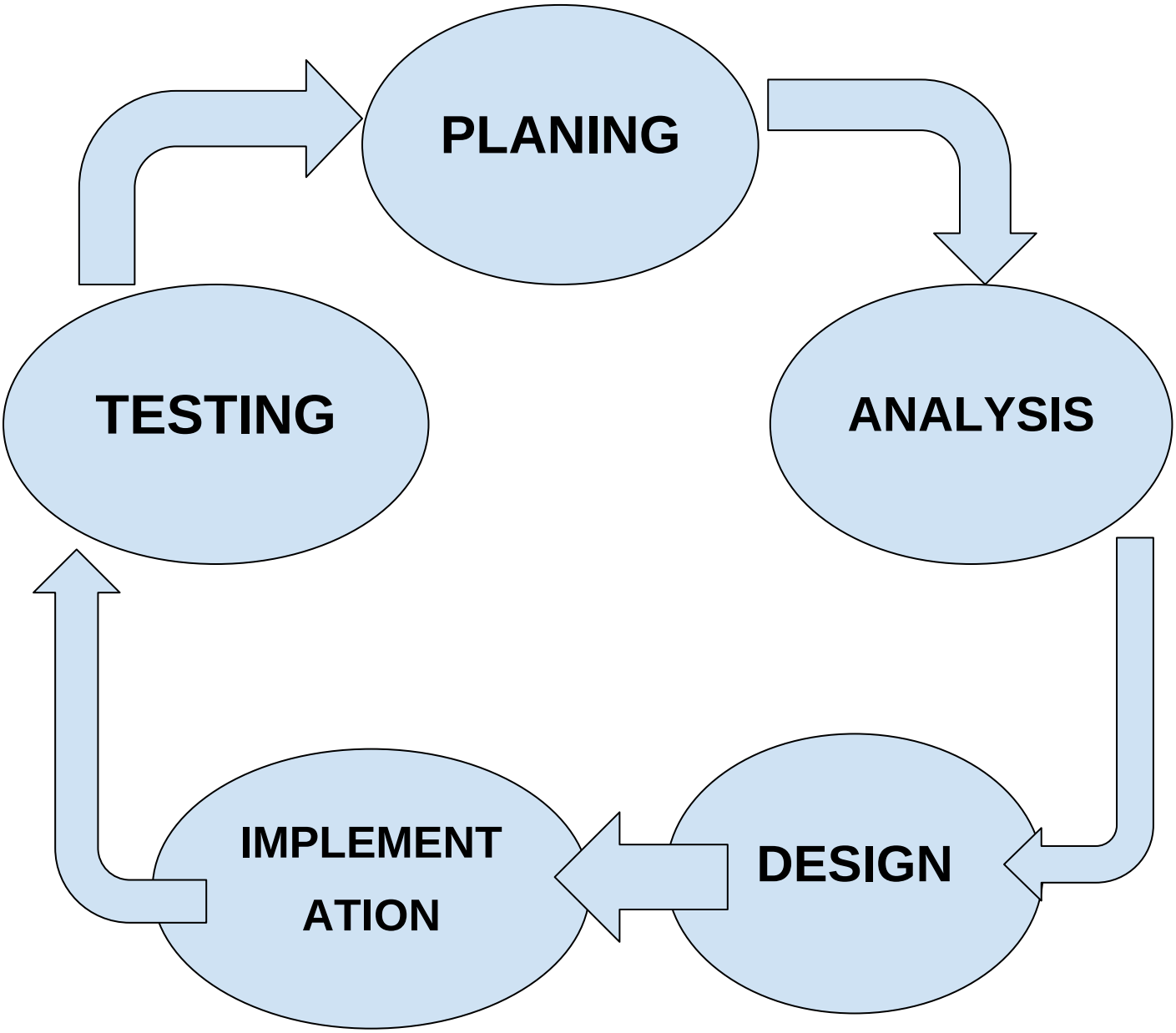
- **Fine:**

  Description: If book is not returned on the time by member then fine is charged on per day basis.
  Input: check for the fines.
  Output: Details about fines on the book issued by the staff.
  Processing: The fine will be calculated, if it crossed the date of return.

Email confirmation:

- The user receives an confirmation email after registration on LMS portal.

PROJECT WORK FLOW



Software Development Life Cycle (SDLC)

the Software Development Life Cycle (SDLC) is a step-by-step process that software developers follow to design, create, test, deploy, and maintain computer programs or applications. It's like a roadmap that helps them organize and manage the entire journey of building software from the idea stage to its ongoing use and improvement.

PLANING:

The planning process is a process used to develop objectives, develop tasks to meet objectives, determine needed resources, create a timeline, determine tracking and assessment, finalize the plan, and distribute the plan to the team.

ANALYSIS:

SDLC analysis is the part of the software development process where developers carefully examine and understand the requirements and goals of the project. It's like figuring out what needs to be done before actually starting to build the software. This analysis phase helps set the direction for the entire development process.

DESIGN:

the SDLC design phase is where the developers create a clear and detailed plan or blueprint for how the software will look and function. It's like designing the layout and features of a building before construction starts. The design phase in SDLC ensures that everyone involved understands how the software will be structured and how users will interact with it.

IMPLEMENTATION :

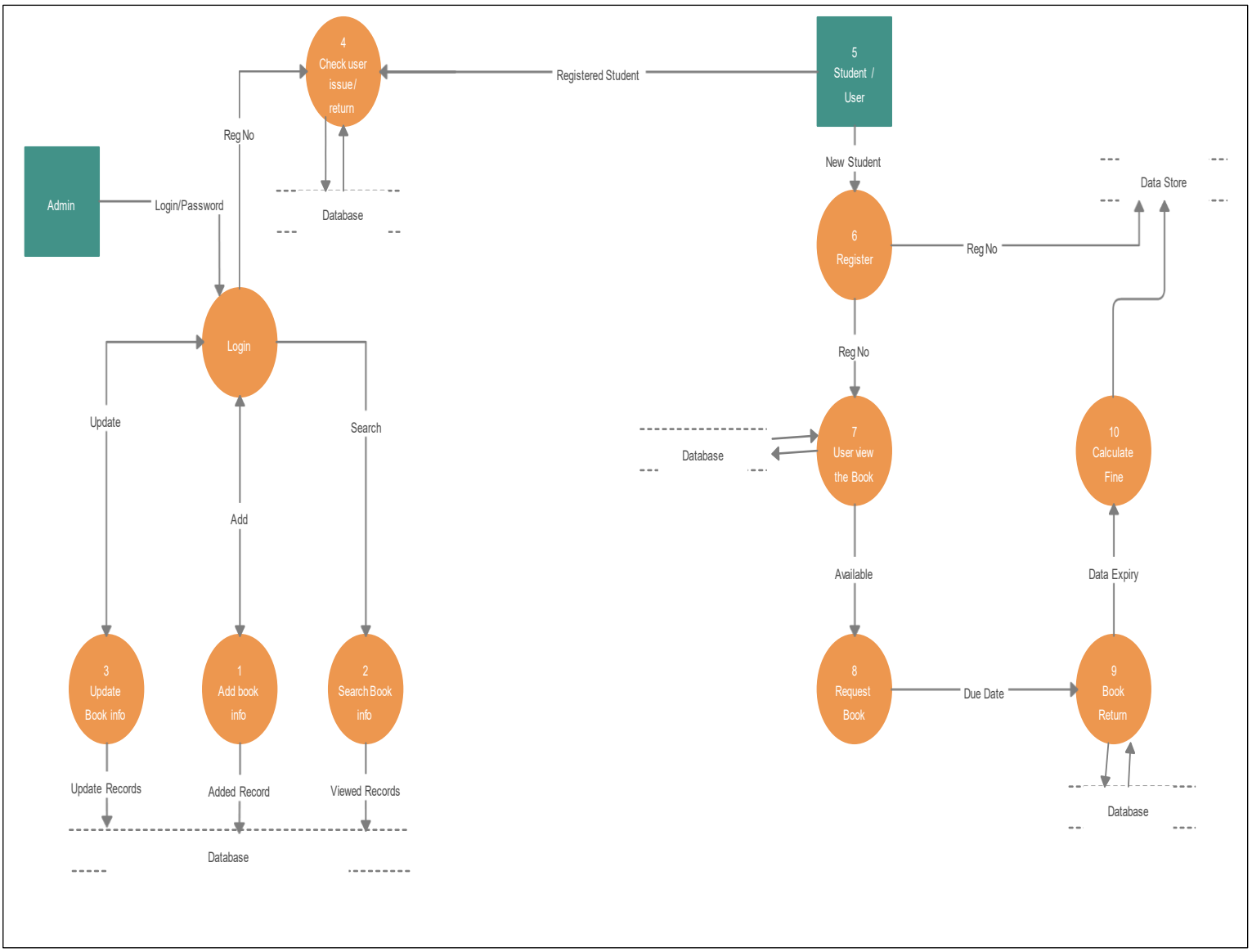
the SDLC implementation phase is where the actual coding and building of the software take place. It's like the construction phase of a building

project. Developers write the code based on the design created in earlier phases. This is when the software starts to come to life, and the planned features are turned into a functional product.

TESTING:

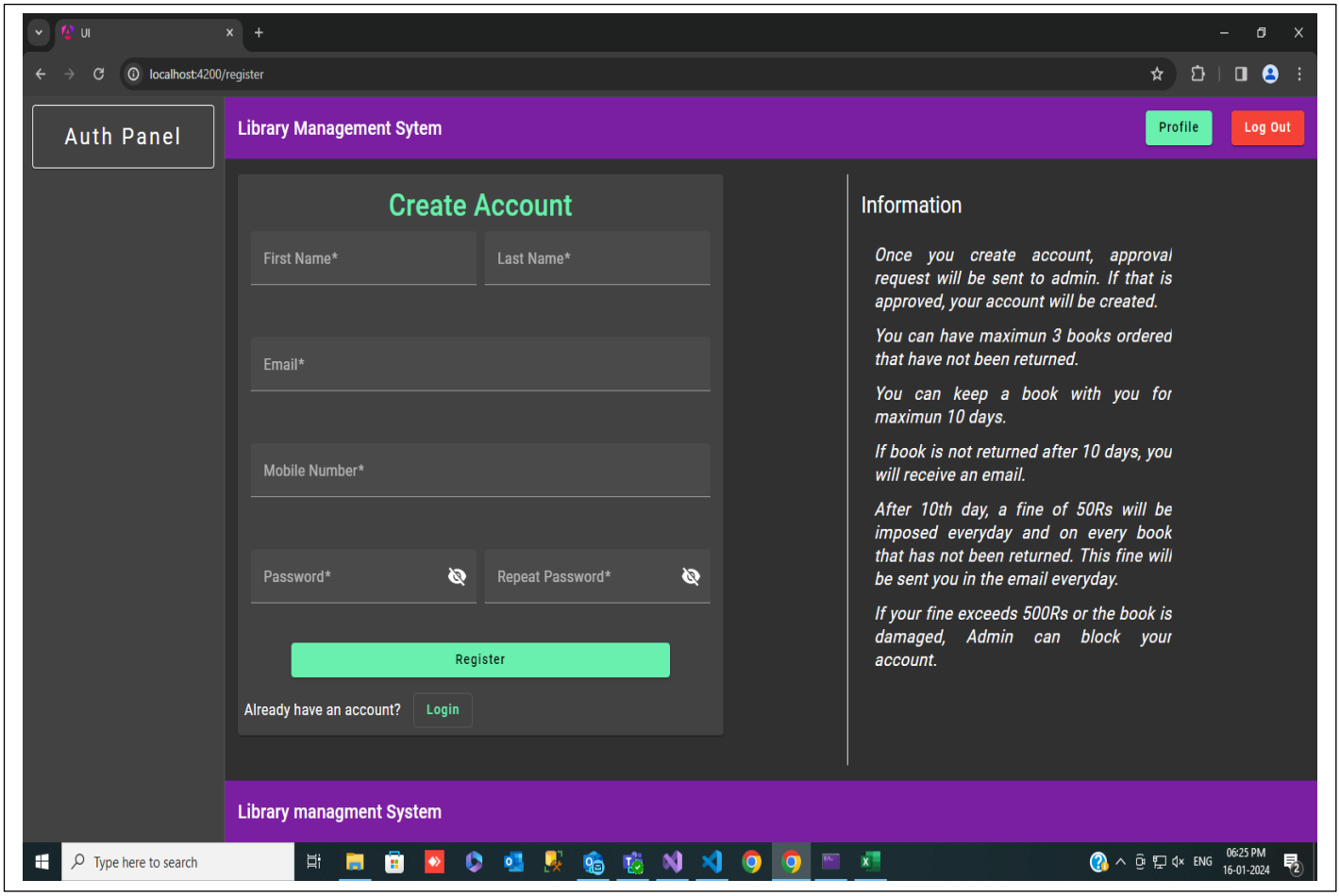
the SDLC testing phase is like putting the software through a series of checks to make sure it works as intended. It's similar to quality control in manufacturing. Testers examine the software, try different scenarios, and identify any bugs or issues. The goal is to ensure that the software is reliable and does what it's supposed to do before it's released to users. Testing helps catch and fix any problems to deliver a better-quality product.

Dataflow Diagram:

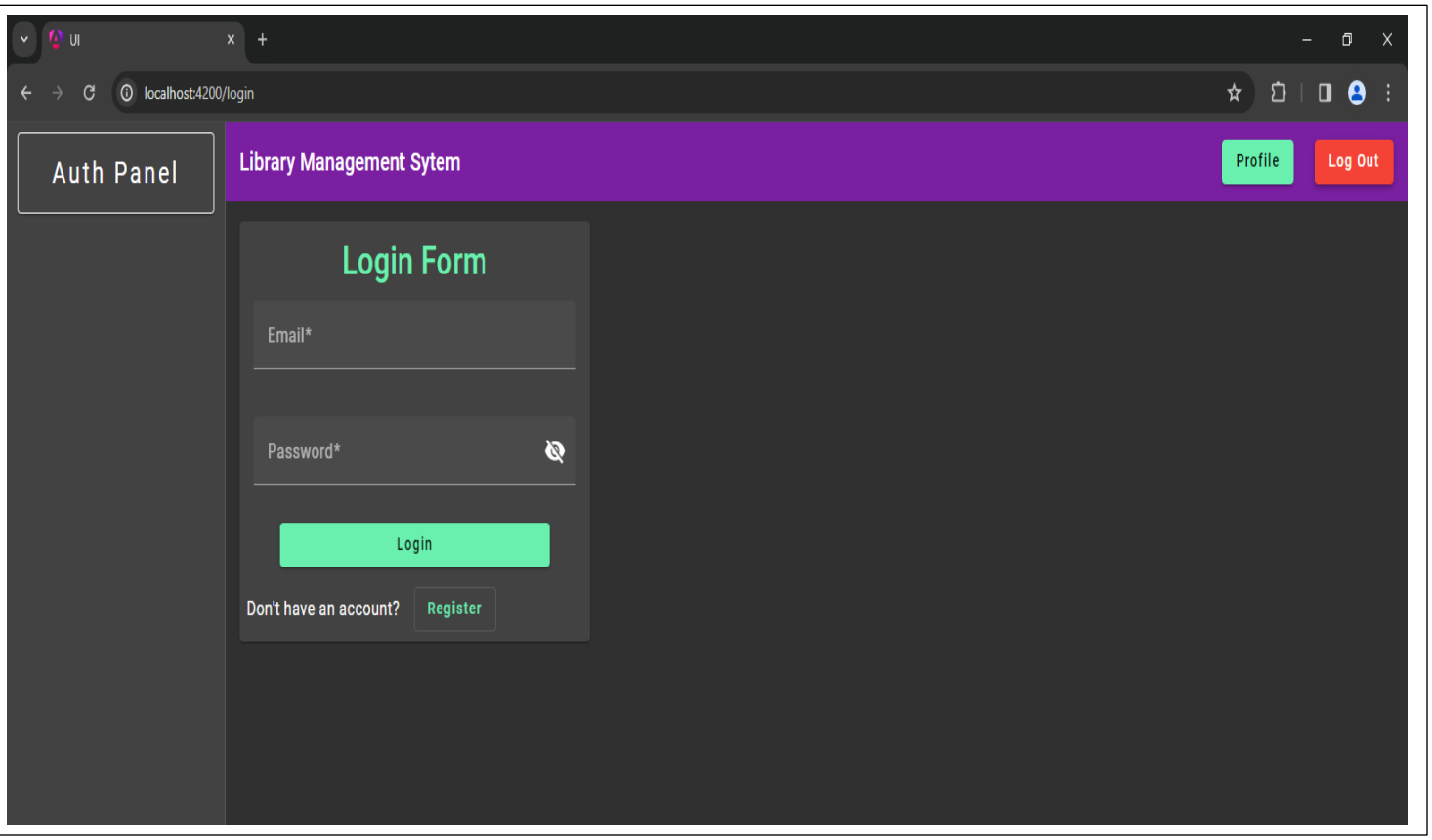


SCREENS

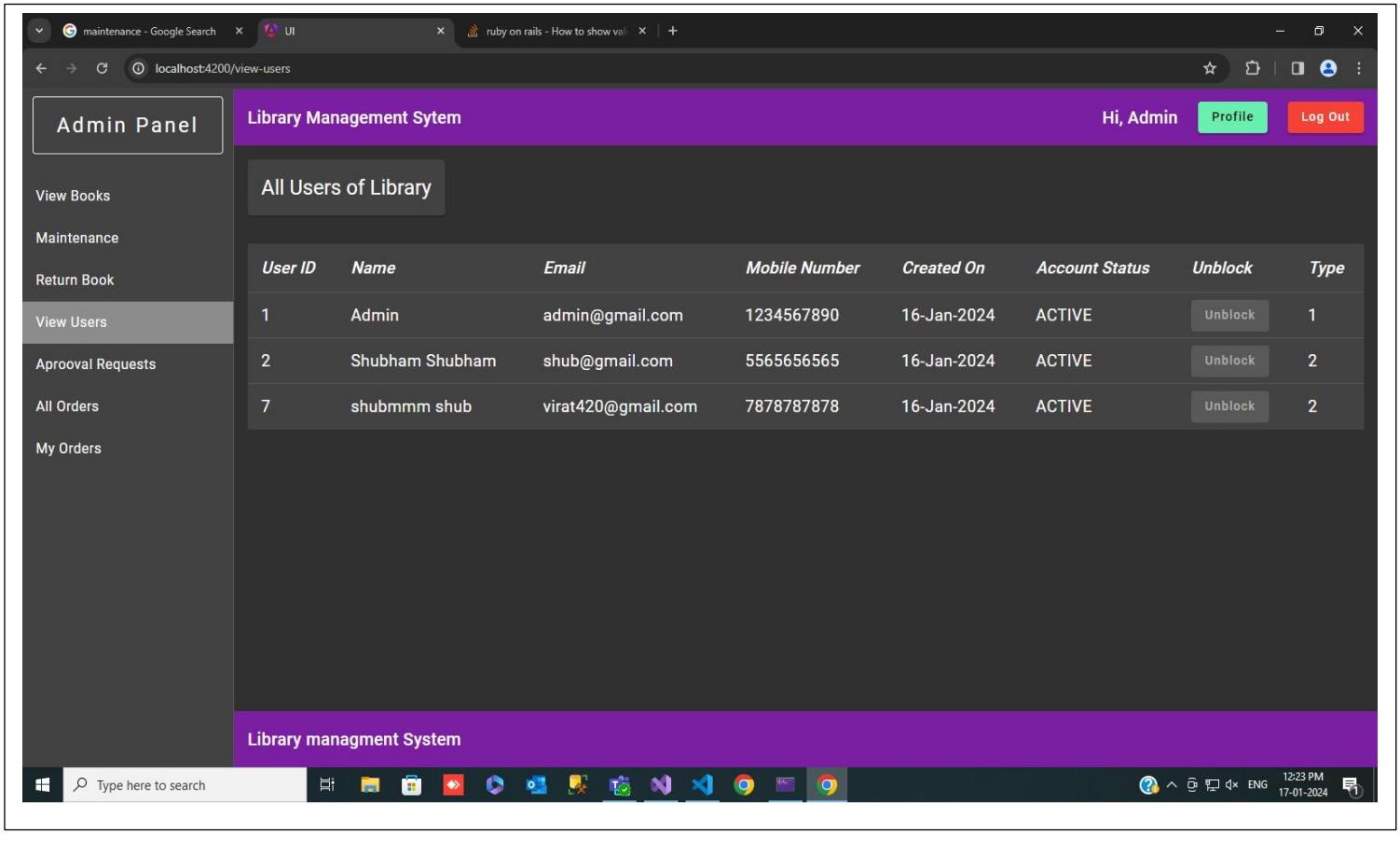
RESISTRATION:



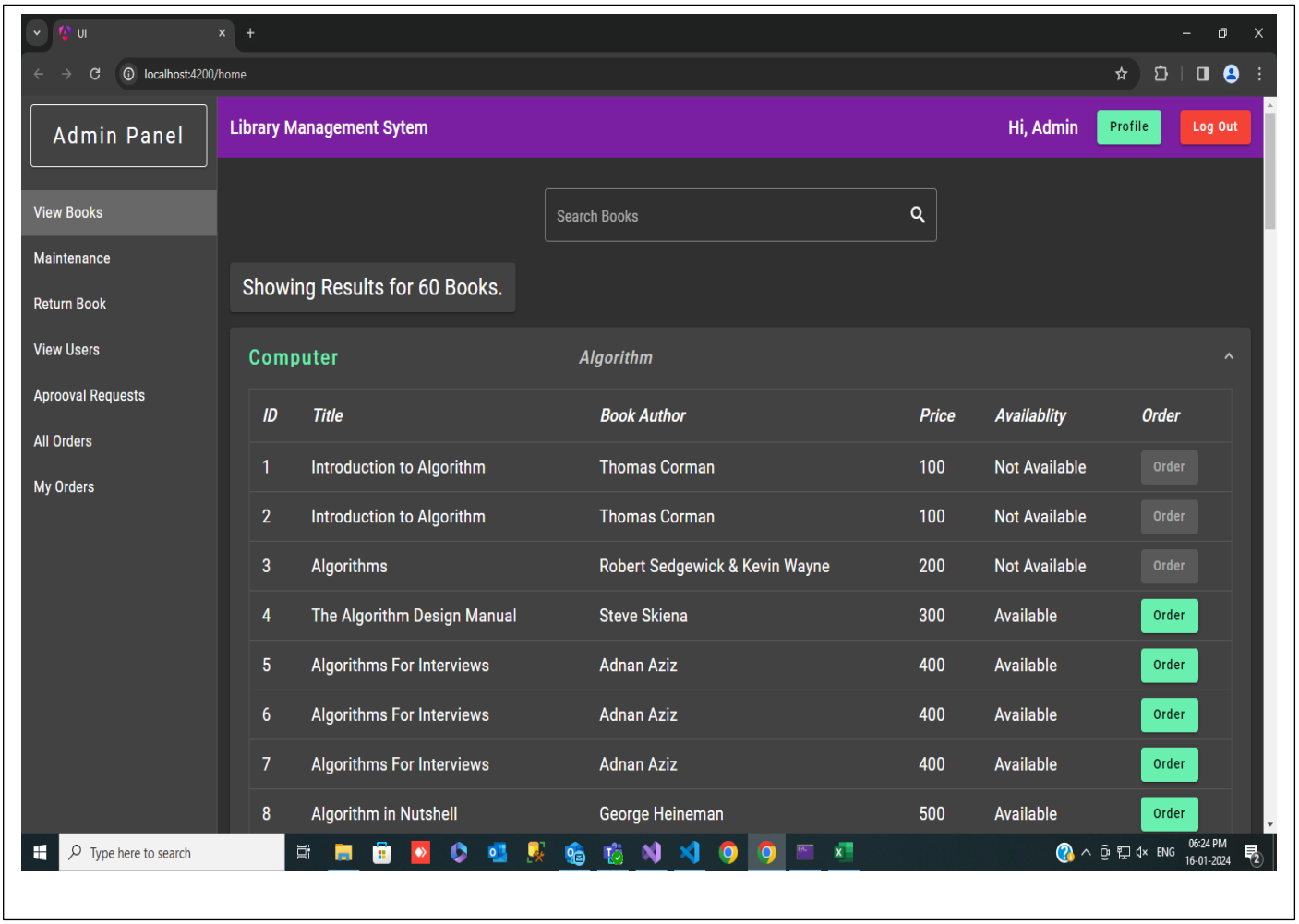
LOGIN:



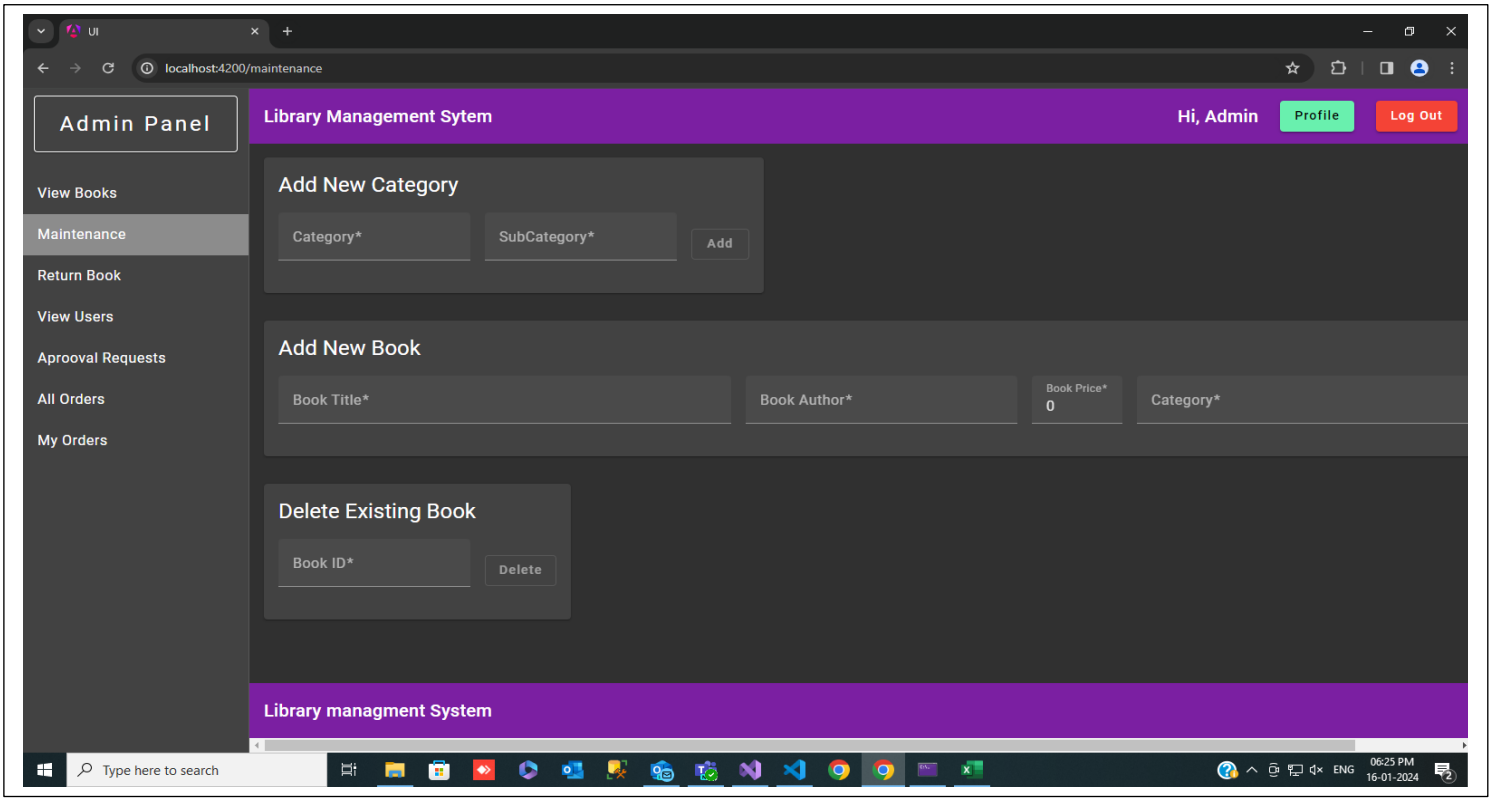
VIEW USERS:



VIEW BOOK:



INSERT & DELETE BOOK:



ORDERS AND RETURN BOOK: