A
PROJECT REPORT ON

"INSTANT LIBRARY"

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE OF

BACHELOR OF SCIENCE (M.S.Cs.)

by

PENUBAKULA ALEKHYA - 1100-21-467-001

PITTELA INDHU - 1100-21-467-002

MEENAKSHI HEEREKAR - 1100-21-467-003

AEPURI NITHYA - 1100-21-467-004

Under the guidance of

K.Prasanth Kumar

Head Department of Computer Science

Faculty of Computer Science

OSMANIA UNIVERSITY
(2021-2024)

Indian Institute of Management and Commerce

6-1-91, Adj.Telephone Bhavan, Khairatabad, Hyderabad – 500004
website: www.iimchyderabad.com

DECLARATION

We hereby declare that the project titled "**INSTANT LIBRARY**" submitted to the Indian Institute of Management and Commerce, affiliated to Osmania University, Hyderabad for the award of the degree of Bachelor of Science (M.S.Cs.) is a result of original project work carried out in this project record. It is further declared that the project report or any part thereof has not been previously submitted to any University or Institute for the award of a degree or diploma.

Name of the Student(Roll No.) **Signature**

PENUBAKULA ALEKHYA
1100-21-467-001

PITTELA INDHU
110-21-467-002

MEENAKSHI HEEREKAR
1100-21-467-003

AEPURI NITHYA
1100-21-467-004

CERTIFICATE

This is to certify that this is the bonafide record of the project entitled "**INSTANT LIBRARY**", submitted by PENUBAKULA ALEKHYA (1100-21-467-001), PITTELA INDHU (1100-21-467-002), MEENAKSHI HEEREKAR (1100-21467003), and AEPURI NITHYA (1100-21-467-004) of B.Sc. (M.S.Cs.) in the partial fulfillment of the requirements for the degree of Bachelor of Science (M.S.Cs.). This project report has not been submitted to any other university or institute for the award of any degree or diploma.

Internal Guide **Signature of the Principal**

External Examiner

ACKNOWLEDGEMENT

We take this opportunity to convey our deepest sense of gratitude and indebtedness to **Sri. K. Raghuveer,** Principal of the Indian Institute of Management and Commerce. We would also like to express our sincere thanks to **Sri. K.Prasanth Kumar,** Head Department of Computer Science and our project guide, and all the faculties of the Department of Computer Science for their valuable suggestions and guidance in helping us to complete our project successfully.

PENUBAKULA ALEKHYA
(1100-21-467-001)

PITTELA INDHU
(1100-21-467-002)

MEENAKSHI HEEREKAR
(1100-21-467-003)

AEPURI NITHYA
(1100-21-467-004)

ABSTRACT

Title of the project: Instant Library - A Library at Your Fingertips

Creating a website with different levels of access using HTML, CSS, and JavaScript:

1. Design the website: First, you need to design the layout of your website using

   HTML and CSS. You can use a CSS framework like Bootstrap or Materialize to make the design process easier. You should create separate pages for the Enter page, E-Book page, Request book page, and any other pages you need.
2. Create the Enter page:

- **HTML Structure:** Define the basic HTML structure with DOCTYPE declaration, meta tags for charset and viewport, and a title.
- **Styling:** Use CSS to style the body, buttons, and titles. Incorporate responsive design practices to ensure the site looks good on all devices.
- **JavaScript:** Implement JavaScript functions to handle button actions, such as redirection to other pages and Counting the number of entries.

3. Create the Button page:

   - **HTML and CSS:** Similar to the entry page, but tailored to the specific functions and content of the button page. Each button should be linked to relevant actions or pages.

Overall, building an INSTANT LIBRARY website by using CSS, JavaScript, HTML can be a complex process, but breaking it down into smaller steps can

make it more manageable. It is also important to test the website thoroughly to ensure that it works as intended.

References:

1. **HTML and CSS:** HTML and CSS are the building blocks of web development. You will need to have a good understanding of these two languages to create a website. W3Schools is a great resource to learn HTML and CSS.
2. **JavaScript**: JavaScript is a programming language that is used to add interactivity to web pages. You will need to know JavaScript to create a website with student credentials. Mozilla Developer Network has a great tutorial on JavaScript.
3. **Codecademy:** Codecademy is an online learning platform that offers courses in web development. You can use Codecademy to learn HTML, CSS.
4. **W3Schools:** This website provides tutorials and examples for web development, including HTML, CSS, and JavaScript. These languages are commonly used in creating websites with user authentication, which are essential for a college website with student credentials.
5. **Udemy:** Udemy offers a variety of online courses on web development and programming. including courses on creating websites with user authentication and databases. These courses can provide a comprehensive understanding of web development concepts and techniques.
6. **Stack Overflow:** Stack Overflow is a popular Q&A platform for developers. You can search for questions related to college websites with student credentials, and see how other developers have solved similar problems.
7. **WordPress:** If you are looking for a quick and easy way to create a college website with student credentials, you can consider using

WordPress. WordPress is a popular content management system that provides user authentication and database integration features out- of the box, and there are many templates and plugins available that can help you customize your website.

Software Requirements:

- **Text Editors:** Tools like Visual Studio Code.
- **Design Tools:** Adobe Photoshop, Illustrator, or online tools like Canva for creating visual assets.
- **Browsers:** For testing, use multiple browsers (Chrome, Firefox, Safari) to ensure compatibility.
- **Local Server:** Tools like XAMPP or MAMP or live server can be used to set up a local development environment for testing.
- Back-end, HTML, CSS, JavaScript, node.js, npm packages, server, database(sqlite3)

TABLE OF CONTENTS

INTRODUCTION

Title: INSTANT LIBRARY: Project Report

1. Introduction:

- **Overview:**

  The "Instant Library" project is a web-based application designed to
  provide users with a seamless experience to access and request e-books. The application offers an efficient registration and login system, user-friendly navigation to select and read e-books by subject, and a request feature for books not available in the library. Developed using modern web technologies, this project aims to enhance digital learning and resource accessibility.

- **Background:**

  With the growing shift towards digital education, there is an increasing need for accessible and organized online libraries. Traditional libraries are often limited by physical space and availability of resources. The "Instant Library" project addresses these limitations by offering a digital platform where users can easily access a wide range of e-books and request additional resources**.**

- **Purpose:**

  The primary purpose of this project is to provide a comprehensive, user-friendly digital library platform that facilitates easy access to educational resources. It aims to simplify the process of finding and requesting books, making educational materials more accessible to students and educators.

2. Problem Statement:

Traditional libraries face numerous challenges, including limited physical space, book availability, and operational hours. These constraints often hinder the accessibility of educational resources. The "Instant Library" project aims to overcome these challenges by providing an online platform where users can access e-books and request additional resources at any time and from anywhere.

3. Key Features:

- **User-friendly Interface**: The home page serves as the entry point where student can enter into the web and can choose an e-book, request a book, and exit.

- **Search and Access**: From the main screen, students have the capability to explore their desired books by selecting the e-book option and select the course and subject of interest. Upon proceeding to the subsequent screen, they can access details such as the book title, author, table of contents/index page, PDFs, and in plan of adding reviews associated with each book through dashboards.

- **Request Functionality**: If a desired book is unavailable, users can request the book using the request box specifying the book credentials, their year/semester, and a space for commenting. This information is automatically sent via email to a librarian for further action.

- **Automated Notifications**: Upon receiving a request, the library system generates automated notifications confirming the receipt of the request. Once the requested book becomes available the updates will be sent to the website.

- **In the plan of adding Quality Assessment Dashboard**: The system features dashboards created using Power BI, allowing students to track the efficiency of books based on the ratings from learners.

- **Exit Button**: Users can easily exit from the information interface and return to the home page with a single click on the exit button, ensuring smooth navigation.

❖ **Benefits**:

- **Timesaving**: Students can efficiently find and retrieve books using their mobile phones, eliminating the necessity for prolonged manual searches.

- **Enhanced User Experience**: The user-friendly interface and intuitive functionalities improve overall satisfaction and convenience for users.
- **Efficient Resource Management**: Automated notifications and request tracking streamline the process of book acquisition and management for library staff.

- **Data-driven Decision Making**: The quality assessment dashboard provides insights into book usage patterns, enabling informed decisions on resource allocation and procurement.

❖ **Technology**

- Programming Language: HTML, JavaScript
- User Interface: HTML, CSS.
- Back-end, browser, live server, database(sqlite3)

Objective

Main Objective

The main objective of the "Instant Library" project is to develop an online platform that allows users to register, log in, and access a wide range of e-books categorized by subject. Additionally, users can request books that are not currently available in the library.

Sub-Objectives

- To create a user-friendly registration and login system.
- To develop a navigation system that allows users to select e-books by subject.
- To implement a request feature for users to request additional books.
- To ensure secure and efficient management of user data and book requests.

Scope of Work

Functional Scope

- **Administrators**: Manage the overall functionality and content of the library.
- **Employees:** Assist in managing book requests and updating the library database.
- **IT Department:** Maintain the technical aspects of the application, ensuring smooth operation.

Technical Scope

- Use of HTML, CSS, and JavaScript for front-end development.
- Use of Node.js and Express for back-end development.
- Use of SQLite3 for database management.

SYSTEM ANALYSIS

4. System Analysis and Requirements

4.1 Stakeholders

Identifying the stakeholders is crucial for understanding their roles and responsibilities within the memo management system. The following stakeholders have

- **Administrators:** Responsible for managing user roles, permissions, and overall system configuration.
- **Employee**: Users who need access to books pdf and request books based on their roles and responsibilities within the organization.
- **IT Department:** Provides technical support and ensures the system's smooth operation.

4.2 Functional Requirements

Based on the stakeholders' inputs and needs, the following functional requirements have been identified:

- **User Registration and Authentication:**
  - Users can register with personal details.
  - Secure login system for registered users.
- **User Dashboard:**
  - Options to select e-books, request books, or exit.
- **E-Book Selection:**
  - Subject-specific pages with available e-books.
- **Book Request Form:**
  - Form to submit book requests with user details and book information.
- **Data Management:**
  - Storage of user registration details and book requests in the database.

4.3 Non-Functional Requirements

In addition to the functional requirements, the following non-functional requirements have been identified:

- **Security:**
  - Secure handling of user data.
  - Protection against unauthorized access.
- **User-Friendly Interface:**
  - Intuitive navigation and easy-to-use forms.
- **Performance:**
  - Quick response times and efficient data retrieval.
- **Scalability:**
  - Ability to handle an increasing number of users and book requests.

4.4 Constraints

The following constraints are considered during the system analysis:

- **Technology Limitations:**
  - Limitations of SQLite3 in handling large datasets.
- **Time Constraints:**
  - Limited timeframe for project completion.
- **Compliance:**
  - Adherence to data protection regulations**.**

By analyzing the requirements and constraints, the system design and implementation will be tailored to meet the organization's specific needs while adhering to the limitations imposed by the chosen technologies and project timeline.

TECHNOLOGIES USED

5. System Design and Architecture

5.1 Technology Stack

The memo management system will be developed using the following technologies:

Front End:

- **HTML(Hypertext Markup Language):** HTML is the standard markup language the structure and organizing the different elements of the system's user interface.

- **CSS(Cascading Style Sheets):** CSS is used for styling and formatting the visual presentation of HTML elements. It enables the system to have a consistent and visually appealing design by specifying fonts, colors, layouts, and other visual properties.

- **JavaScript:** JavaScript is a versatile programming language that runs on the client side of web applications. It adds interactivity and dynamic functionality to web pages. JavaScript will be used to handle user interactions, perform form validations, implement search and filtering functionalities, and facilitate data retrieval and manipulation within the library management system.

Back-End: Node.js, Express, body-parser

Database: SQLite3, DB Browser

5.2 User Interface Design

The user interface of the instant library will be designed using HTML, and CSS. The layout and visual elements will be structured to provide a user-friendly experience, allowing users to navigate through the system easily.

The user interface will incorporate components such as entry, e-books, request books, and editing forms, book lists, and search/filtering options. These components will be styled using CSS and customized according to the organization's branding guidelines and design preferences.

- Simple and intuitive interface with easy navigation.

- Responsive design for accessibility on various devices.

5.3 System Architecture

The instant library will follow a client-side architecture, where the system logic and data handling will primarily reside in the user's web browser. The HTML. CSS, and JavaScript files will be delivered to the user's browser, and the user interactions will be processed locally without requiring server-side processing**.**

- Three-tier architecture: Presentation layer (UI), Application layer (serverside logic), and Data layer (database).

The system's architecture will consist of:

- **HTML files**: These files will define the structure and layout of the web pages, incorporating various components for user interactions and displaying memo information.

- **CSS files:** These files will provide the visual styling and formatting rules for the HTML elements, ensuring a consistent and visually appealing user interface

- **JavaScript files:** These files will handle user interactions, perform form validations, enable search and filtering functionalities, and facilitate data retrieval and manipulation within the system.

By utilizing HTML, CSS and JavaScript, the instant library will be developed with an intuitive and visually engaging user interface, providing an efficient and interactive experience for accessing and managing data.

SYSTEM DESIGN

6. System Design and Architecture

6.1 System Design Approach

The system design of the instant library will follow a modular and component-based approach, ensuring a clear separation of concerns and promoting reusability. The design will focus on the user interface components, data management, authentication.

6.2 User Interface Design

The user interface design will be structured and intuitive, allowing users to navigate the system easily and perform required actions efficiently. The layout will be responsive, adapting to different screen sizes and devices. The user interface will consist of the following key components:

- **Login Forms**: Users will be provided with a login form to authenticate themselves into the system.
- **Data Creation and Editing Forms:** Administrators will have access to forms for creating new field subjects and editing existing ones. These forms will include fields for subject contents, title, author , semester, index page and complete book pdfs .
- **Login Page**: User registration and login forms.
- **User Dashboard**: Buttons for e-book selection, book requests, and exit.
- **Subject Pages**: E-books categorized by subjects.
- **Request Book Page**: Form for submitting book requests.

6.3 Data Management

As the project do involves a database, a simplified data management approach will be employed. The Request book information will be stored within DB Browser sqlite3. These files will serve as the data source for the system, allowing for data retrieval and manipulation.

Upon user authentication, the system will load the necessary book data and store it in the database for subsequent access. Data manipulation and updates will be handled within the client-side JavaScript, ensuring the integrity and consistency of the data information.

- User data and book requests stored in SQLite3 database.
- Efficient data retrieval and management.

6.4 Authentication and User Management

The system will implement a user authentication mechanism using JavaScript to validate user credentials. User registration and login forms will be provided for account creation and authentication. Upon successful login, the user's access

privileges and roles will be retrieved from the data source to determine the level of data access.

Administrators will have additional privileges for creating, editing, and deleting subjects data. User roles and permissions will be managed within the JavaScript logic, allowing administrators to assign access levels to users based on their roles and responsibilities.

- Secure user authentication system.
- Role-based access control for different user types (admin, user).

6.5 System Flow

The system flow will follow a typical login-based workflow. Users will access the system by entering their credentials in the login form. Upon successful authentication, they will be redirected to the main dashboard, where they can view and search for subjects based on their access permissions, request books. Administrators will have additional options to create, edit, and delete data.

By employing a modular and component-based approach, the system design will promote maintainability, scalability, and ease of future enhancements. The user interface will be designed to provide an intuitive and efficient user experience, while data management and authentication mechanisms will ensure the integrity and security of data access and management.

- **User Registration/Login:** Users register and log in.
- **Dashboard Navigation:** Users select e-books or request books.
- **E-Book Selection:** Users navigate to subject-specific pages.
- **Book Request Submission:** Users submit book requests via a form

TESTING

7. Testing and Validation

7.1 Testing Approach

Testing is a critical phase of the development process to ensure that the instant library functions as intended and meets the specified requirements. The following testing approaches will be employed:

- **Unit Testing:** Test individual components for functionality. Individual components and functions within the system, such as authentication, data creation will be tested independently to verify their correctness and expected behavior.
- **Integration Testing:** Ensure components work together seamlessly. Once the individual components are tested, integration testing will be performed to assess the interaction and compatibility between different modules of the system. This ensures that the system functions as a cohesive unit.
- **System Testing:** Validate the entire system against requirements.
  User Acceptance Testing(UAT): Obtain feedback from end-users. Users, particularly administrators and employees, will be involved in the UAT process. They will perform real-world scenarios and provide feedback on the usability, functionality, and performance of the system.

7.2 Test Cases

Test cases will be designed to cover various scenarios and functionalities of the instant library. Some example test cases include:

- Registration and login functionality.
- Navigation to e-book and request pages.
- Submission of book request forms.

➢ **Test Case:** User Authentication
Description: Verify that users can successfully authenticate with valid credentials.
- Steps:

1. Enter a valid username and password.
2. Click on the login button.

Expected Result: The user should be logged into the system and redirected to the main dashboard.

7.3 Validation

In addition to testing, validation will be performed to ensure that the instant library meets the requirements and expectations of the stakeholders. This includes:

- Verify correct data storage and retrieval.
- Ensure secure data handling.
- **User Feedback:** Gather feedback from users during the UAT phase to identify any issues or areas for improvement in terms of usability, functionality, and overall user experience.
- **Requirement Validation:** Validate that the implemented system fulfills the identified functional and non-functional requirements outlined in the project scope. Any deviations or gaps should be addressed.
- **Performance Evaluation:** Assess the system's performance, responsiveness, and scalability by simulating a realistic workload and measuring response times, resource utilization, and system stability.

7.4 Bug Tracking and Resolution

- Use of bug tracking tools to document and resolve issues.
- Throughout the testing and validation process, any identified bugs or issues will be logged and tracked systematically. Each bug will be assigned a priority level based on its impact on system functionality and usability. The development team will address these issues promptly and provide appropriate bug fixes or system enhancements.
- By conducting comprehensive testing, gathering user feedback, and ensuring validation of requirements, instant library will be refined and optimized for stability, functionality, and user satisfaction.

8. IMPLEMENTATION

8.1 Development Environment

The implementation of the instant library will require a suitable development environment. The following tools and technologies will be utilized:

- **Integrated Development Environment (IDE):** Visual Studio Code A IDE such as Visual Studio Code will be used for writing and editing the HTML, CSS, and JavaScript code.

- **Web Browser:** Google Chrome, Mozilla Firefox
  A modern web browser like Google Chrome, Mozilla Firefox, or Microsoft Edge will be used to test and run the system.

- **Web Server:** Node.js with Express
  A simple web server, such as the built-in development server provided by the IDE node.js with express or third-party solutions like XAMPP or WAMP, may be used to serve the web pages locally during development and testing**.**

8.2 Development Process

The implementation of the instant library will involve the following steps:

- **Phase 1:** Requirements gathering and planning.
- **Phase 2:** Design and prototyping.
- **Phase 3:** Development and coding.
- **Phase 4:** Testing and validation.
- **Phase 5:** Deployment and maintenance.

- **Project Setup**: Create a project directory structure and set up the necessary files, including HTML, CSS, and JavaScript files.

- **User Interface Development**: Design and implement the user interface using HTML, CSS. Create the necessary forms, layout components, and styling to achieve the desired visual representation of the system.

- **JavaScript Implementation:** Write the JavaScript code to handle user interactions, implement authentication, library creation and editing functionalities, book retrieval and filtering, and other system features.

- **Integration and Testing:** Integrate the different components and functionalities of the system. Perform unit testing to ensure the individual components work as expected. Conduct integration testing to validate the interaction between various modules and functionalities.

- **User Acceptance Testing:** Involve stakeholders, including administrators and employees, in the testing process to validate the system against real-world scenarios. Gather feedback and make necessary improvements based on the user feedback.

- **Bug Fixing and Refinement**: Address any issues or bugs identified during testing and make necessary refinements to enhance system stability, performance, and user experience.

- **Documentation**: Prepare documentation, including user manuals, installation guides, and code documentation, to provide information on system usage, setup, and maintenance.

8.3 Deployment

The instant library, being a web-based application, can be deployed on a web server or hosted on a cloud platform. The deployment process involves:

- Deployment on a cloud server for accessibility.
- Regular updates and maintenance**.**
    - Testing in Production: Perform thorough testing in the production environment to ensure that the deployed system is working correctly and that all features and functionalities are operational.
    - Monitoring and Maintenance: Implement monitoring tools and practices to track system performance, address any issues, and apply necessary updates or patches as required.

      By following a systematic implementation approach and ensuring thorough testing, the instant library can be successfully developed and deployed for use, providing an efficient and userfriendly solution for accessing and managing books within the organization.

SAMPLE SOURCE CODE

9. SOURCE CODE

<!-- login and registration page -->

```
const express = require('express'); const
bodyParser = require('body-parser'); const
sqlite3 = require('sqlite3').verbose();
const path = require('path'); const bcrypt =
require('bcrypt');
 const app = express(); const db = new
sqlite3.Database('database.db');
 app.use(bodyParser.urlencoded({ extended: true }));
app.use(express.static(path.join(__dirname,
'public')));

// Create users table if it doesn't exist
db.run(`CREATE TABLE IF NOT EXISTS users
(     id INTEGER PRIMARY KEY
AUTOINCREMENT,     name TEXT,     email
TEXT UNIQUE,     password TEXT,
    login_count INTEGER DEFAULT 0
)`);

// Register endpoint app.post('/register', async (req,
res) => {     const { name, email, password, repassword }
= req.body;
     if (password !== repassword)
{
        return res.send('Passwords do not match');
    }      const hashedPassword = await
bcrypt.hash(password, 10);

    db.run(`INSERT INTO users (name, email, password) VALUES (?, ?, ?)`,
[name, email, hashedPassword], function(err) {
if (err) {
            return res.send('Email already registered');
        }
res.redirect('/login.html');
    });
});

// Login endpoint app.post('/login',
(req, res) => {     const { email,
password } = req.body;    db.get(`SELECT
* FROM users WHERE email = ?`, [email],
async (err, user)
```

```
=> {         if (err || !user) {
return res.send('User not found');
        }          const isMatch = await bcrypt.compare(password,
user.password);
         if (!isMatch) {             return
res.send('Incorrect password');
        }         db.run(`UPDATE users SET login_count = login_count + 1 WHERE
id = ?`, [user.id]);

res.redirect('/buttons.html');
    });
});  app.get('/abcd.html', (req,
res) => {
    res.sendFile(path.join(__dirname, 'public', 'C:\Users\P
Alekhya\Desktop\instant_library\e book buttons.html'));
});  const PORT = process.env.PORT || 3000;
app.listen(PORT, () => {     console.log(`Server is
running on port ${PORT}`);
});
```

<!—main dashboard -->

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>INSTANT LIBRARY

</title>
<style>
  /* Your CSS code here */
/* Style for buttons */
button {   padding: 10px
20px;   font-size: 16px;
border: none;   cursor:
pointer;   margin: 10px;
background-color: #007bff;
color: #fff;  border-
radius: 5px;
}  button:hover {
background-color: #0056b3;
}

/* Body style */ body {   font-family: Arial, sans-serif;   background-color:
#f4f4f4;   margin: 0;   padding: 0;   display: flex;   flex-direction:
column;   align-items: center;   justify-content: center;   height: 100vh;
font-family: Arial, sans-serif;       background-image: url('b.jpg'); /*
Replace 'path_to_your_image.jpg' with the path to your image file */
background-size: cover; /* This makes sure the image covers the whole
background */
```

```
}

/* Title style */
h1 {
  color: #13fbe8;
}

/* Center the buttons */
.buttons {   display:
flex;   flex-direction:
column;   align-items:
center;
}

</style>
<script>
  function redirectToUserCredentials() {
window.location.href = "subject section page.html";

}
  function redirectToRequestBooks() {
window.location.href = "http://localhost:3000/";   }
</script>
</head>
<body>
  <h1>Instant Library</h1>
  <i>_A library on your finger tips!</i>

<button onclick="redirectToUserCredentials()">E-Books</button>
<button onclick="redirectToRequestBooks()">Request Books</button>
<button id="exitButton">Exit</button>
 <script> document.getElementById("exitButton").addEventListener("click",
function() {   window.open('', '_self', '');     window.close();
});
</script>

</body>
</html>
```

OUTPUT SCREENS (pictures)

10. OUTPUT SCREEN

- **Login Screen:** User registration and login functionality.

Upon accessing the system, users will be presented with a login screen. It typically consists of input fields for username and password, along with a login button. Users enter their credentials and click the login button to access the system. For new users, there is a separate page for registration and then login.

- **User Dashboard Screen:** Options for e-book selection, book requests, and exit.

After successful login, users will be redirected to the dashboard screen, which serves as the main hub for the instant library. The dashboard may display relevant information and provide quick access to key functionalities such as e-books, request books, and exit buttons. It could include sections like one at a time.

- **Subject pages:** Display of e-books categorized by subject.

After choosing of e-book button from the main dashboard, the user is directed to a page where he can choose his course and subject and when he clicks on the enter button then he is directed to the related/choose course subject in another dashboard, there he will find complete information about the subject and pdfs of the books and user may also find the review presentation of subject books in the same page.

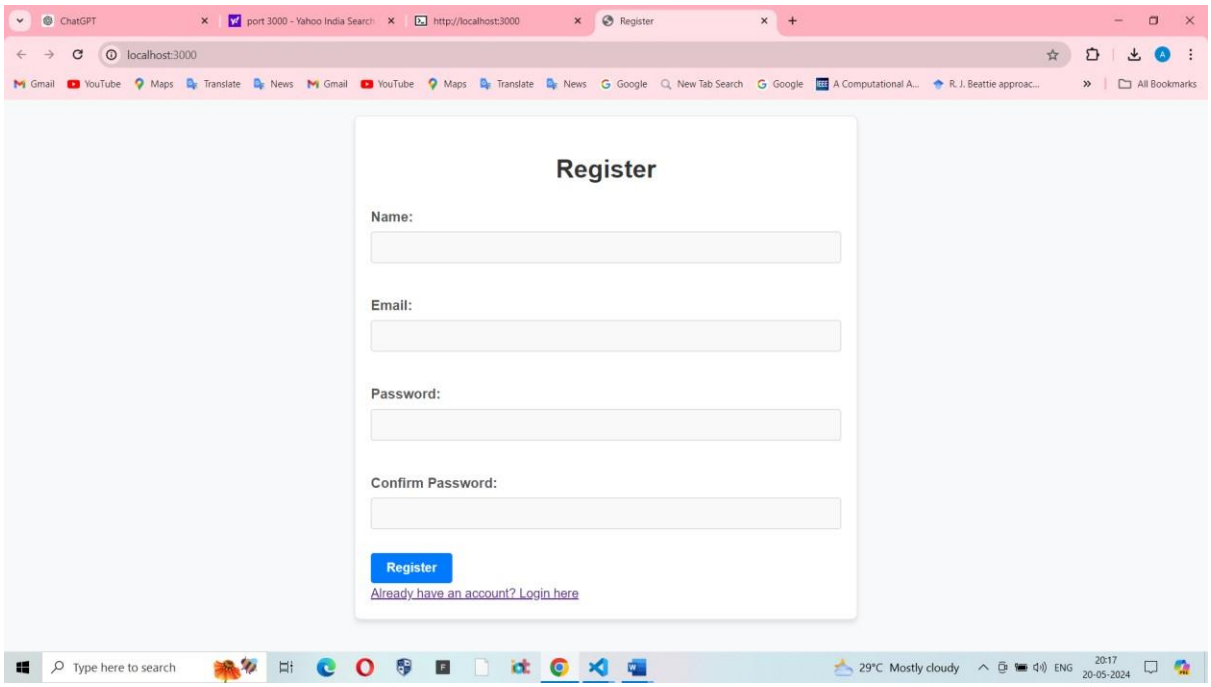
- **Request book page:** Form for book requests with data storage.

After choosing of Request button from the main dashboard, the user is directed to a page where he can give his credentials such as name, roll no., course, semester, subject, etc., and selects the enter button then he will get an automated request confirmation.

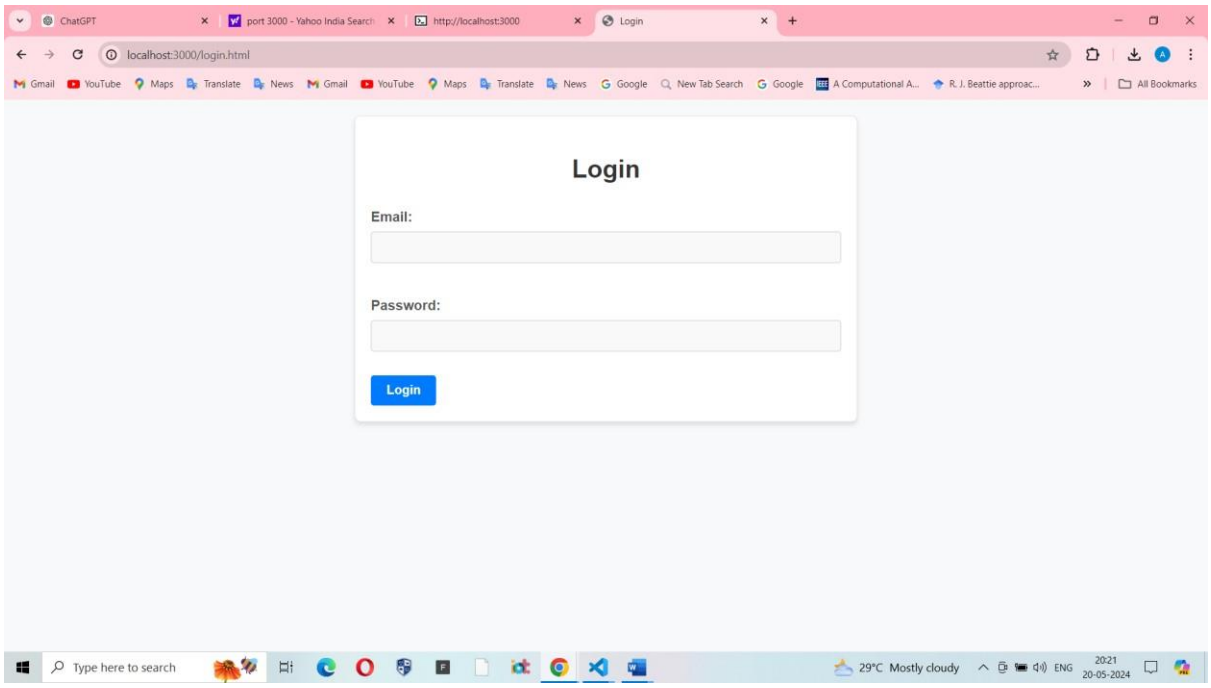
- **Exit button**: The user uses the exit button, when there is no work left in the browser and wants to go back.

Remember that these are just examples, and you can customize the screens based on your specific requirements and design preferences. UI/UX design plays a crucial role in creating visually appealing and user-friendly screens, so it's recommended to involve a designer to create a professional and cohesive visual representation of your instant library project.

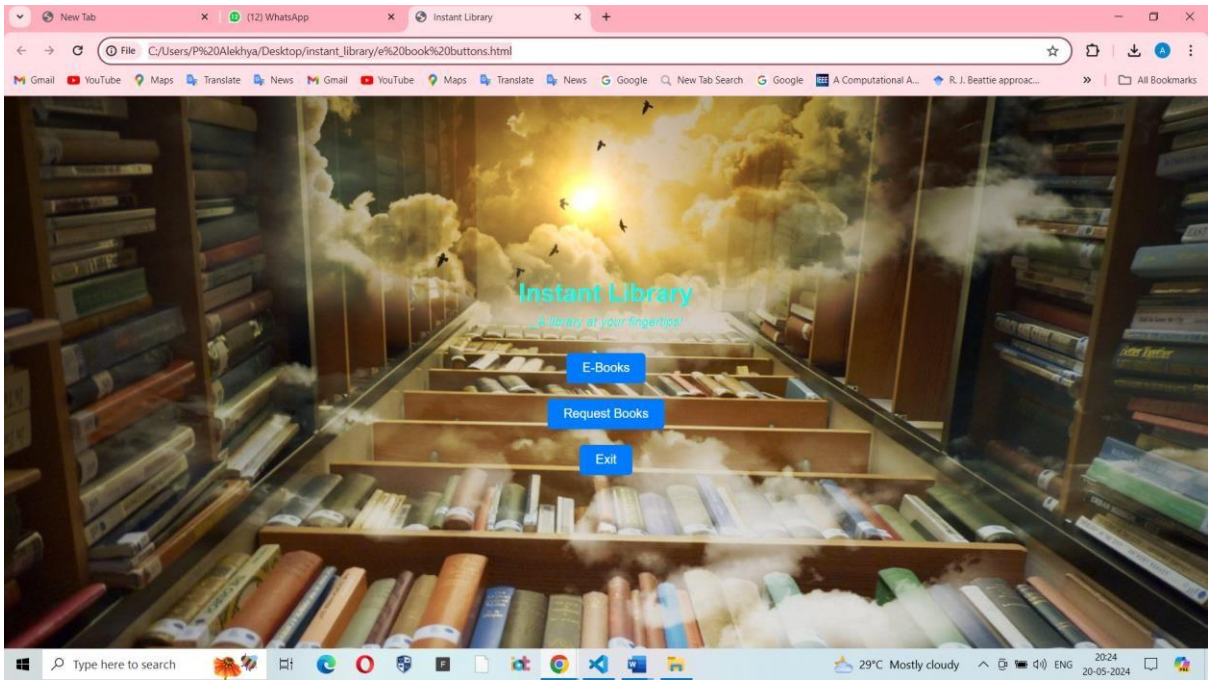
Register screen:



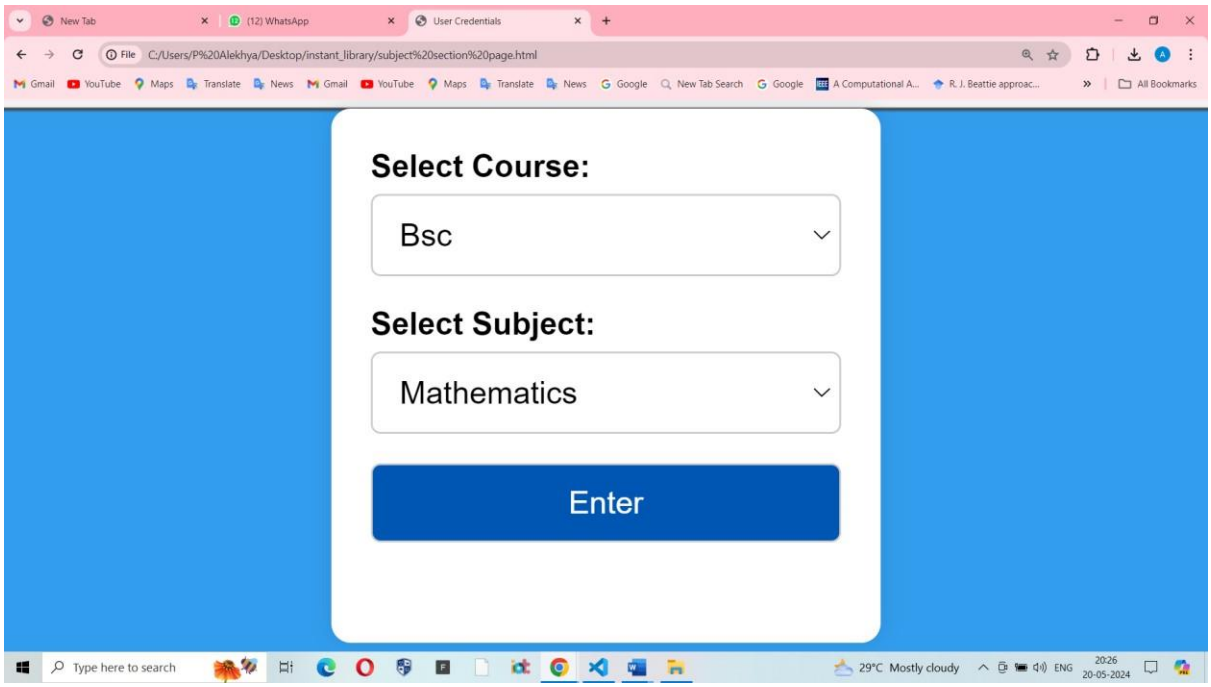
Login screen:



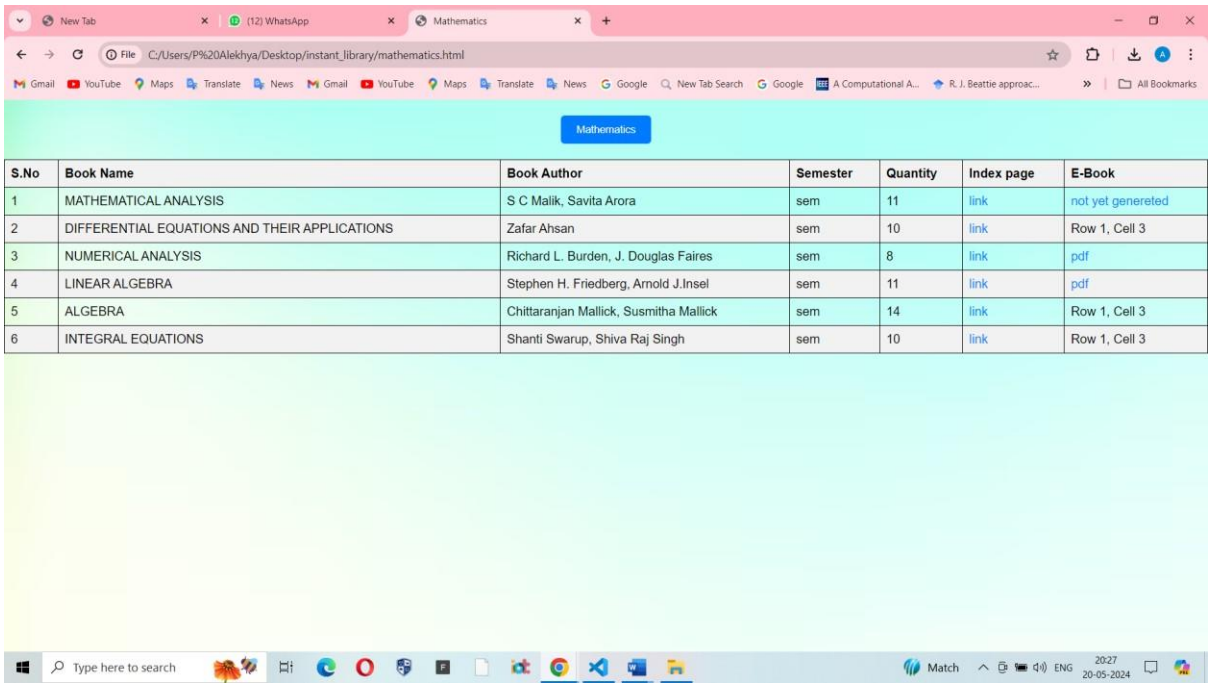
Main page:



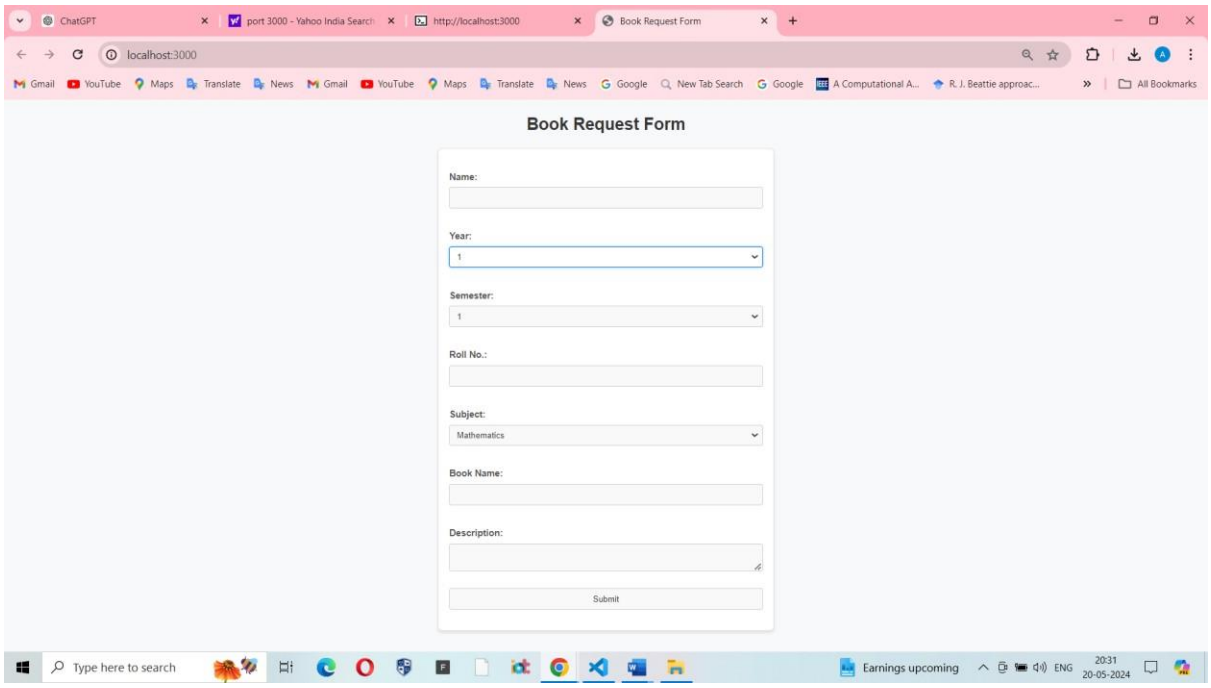
E-book page:



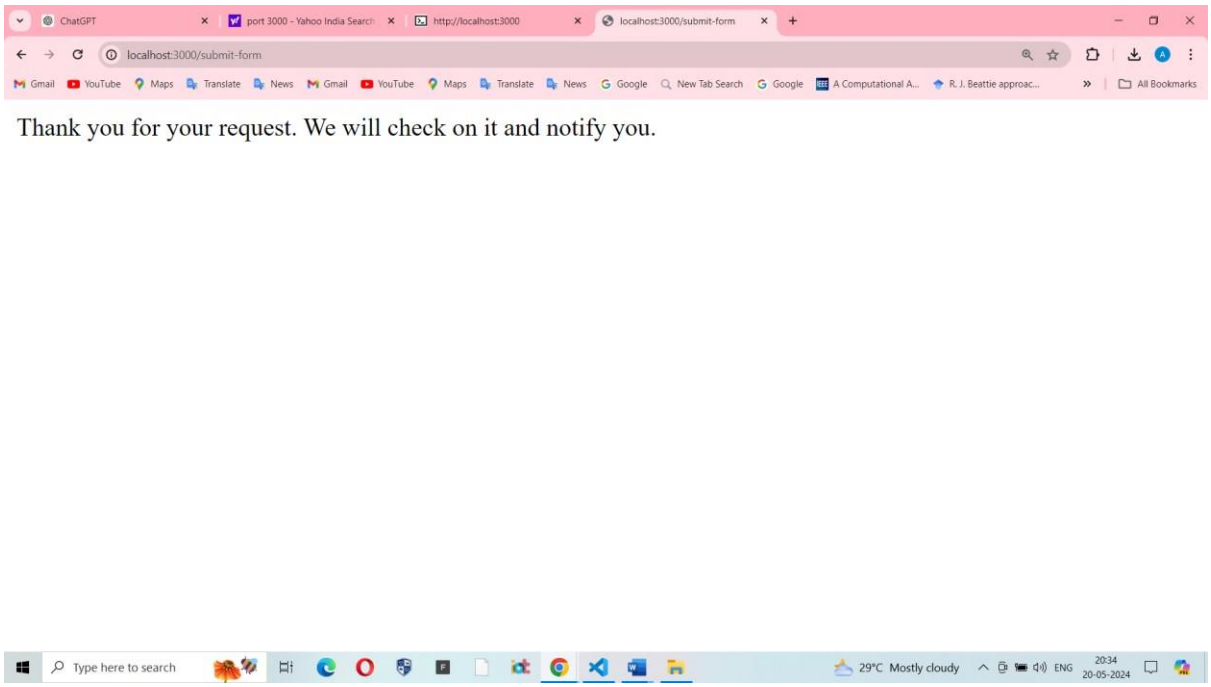
Subject page:



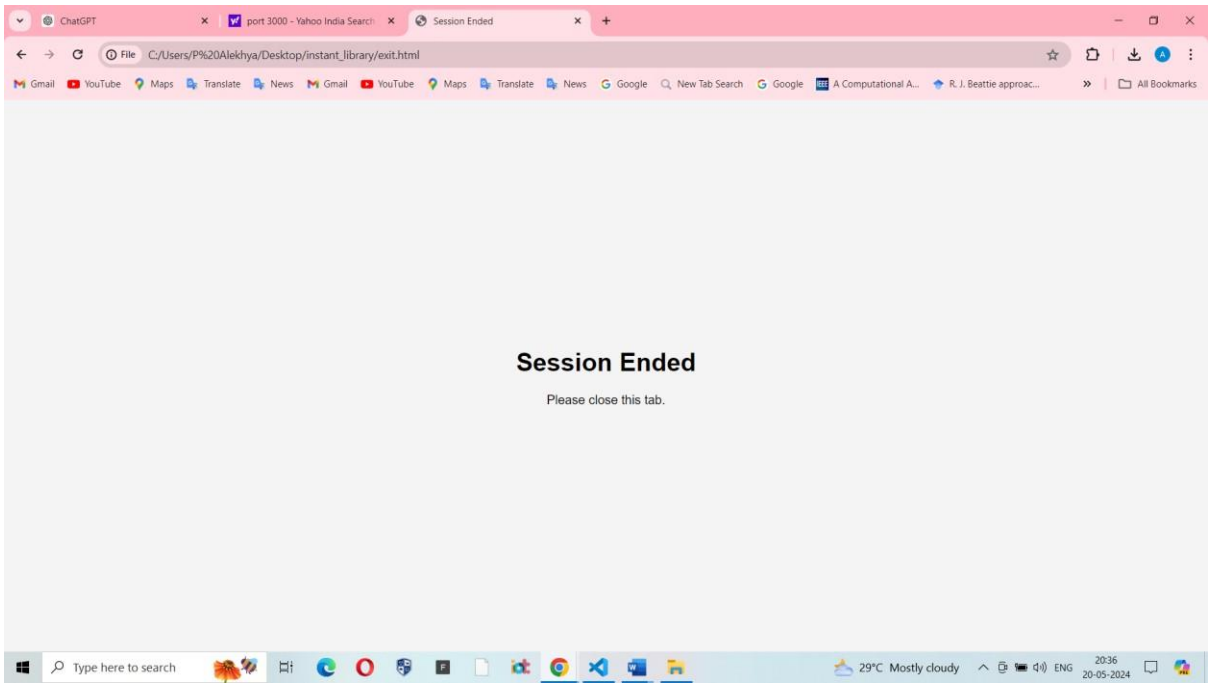
Request book form:



Request confirmation page:

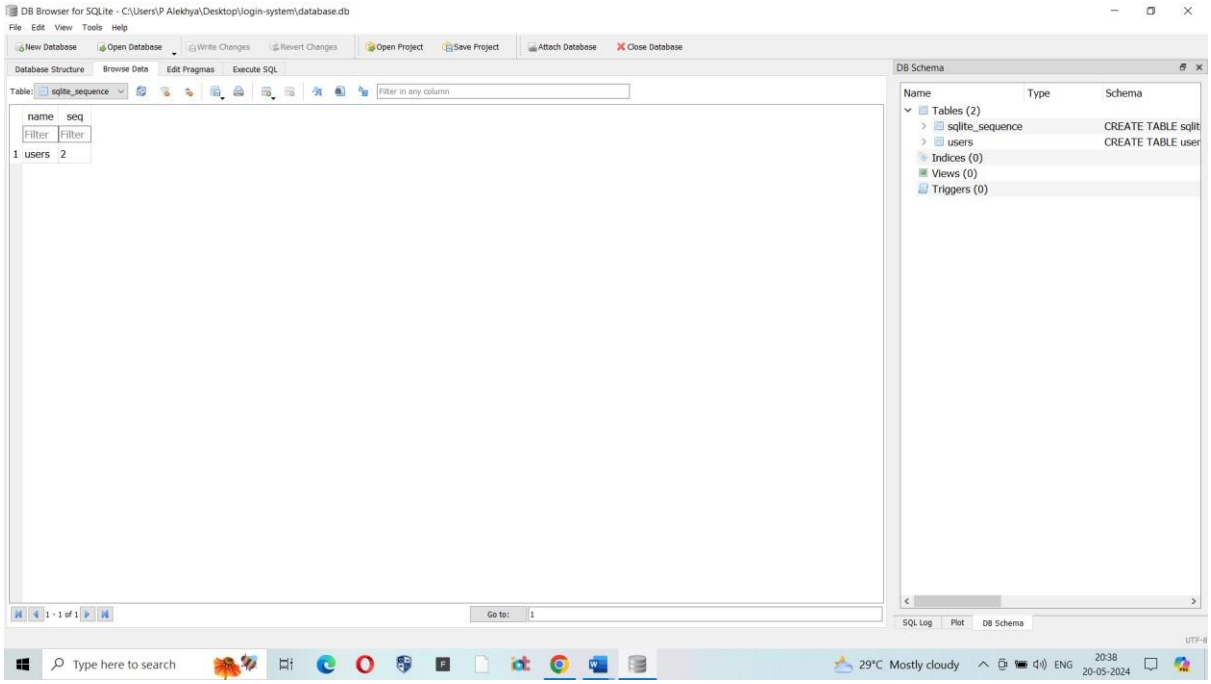


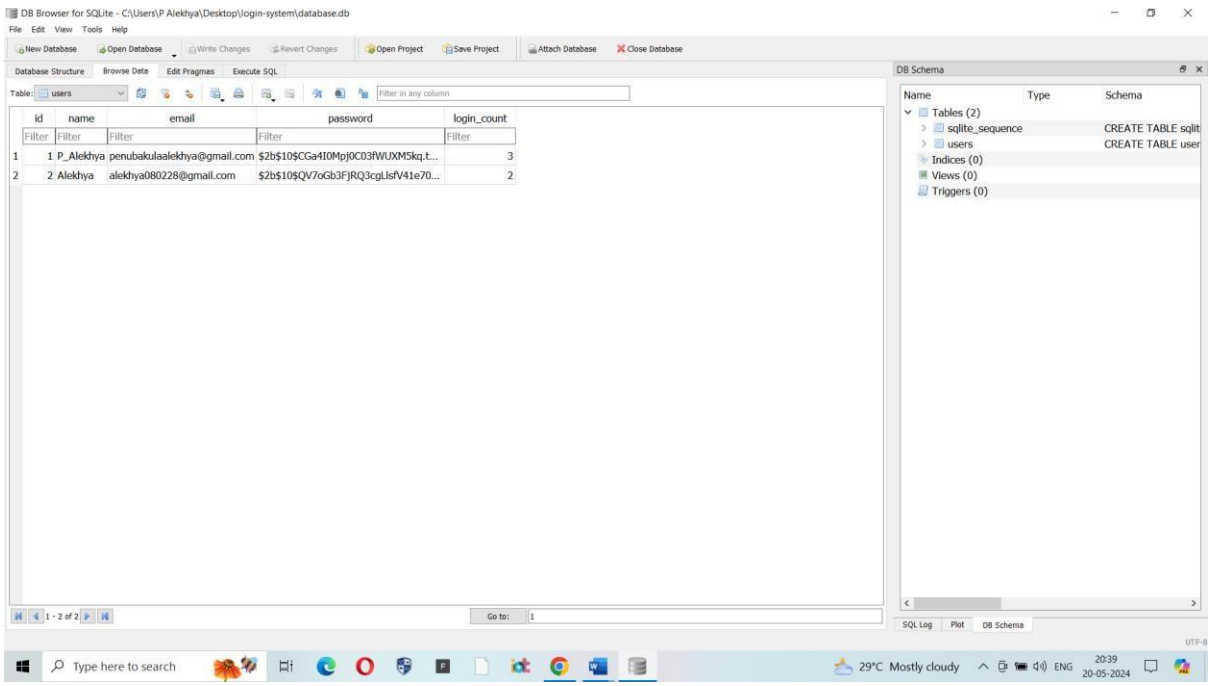
Exit page:



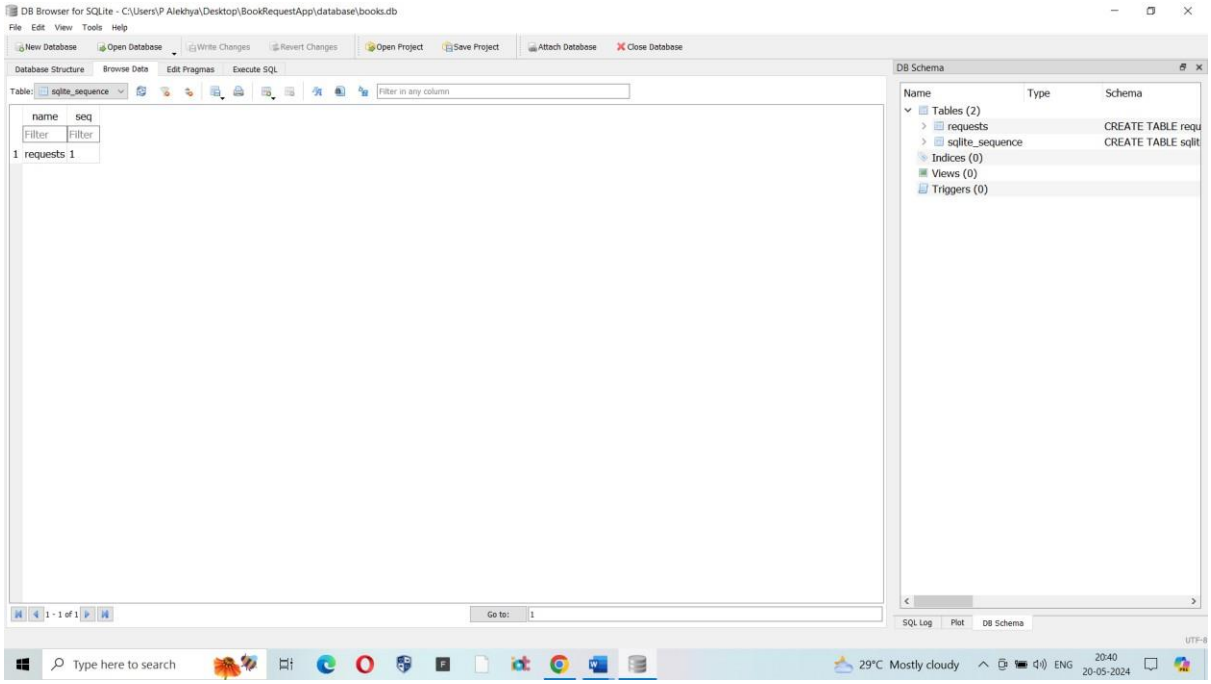
Data storage:

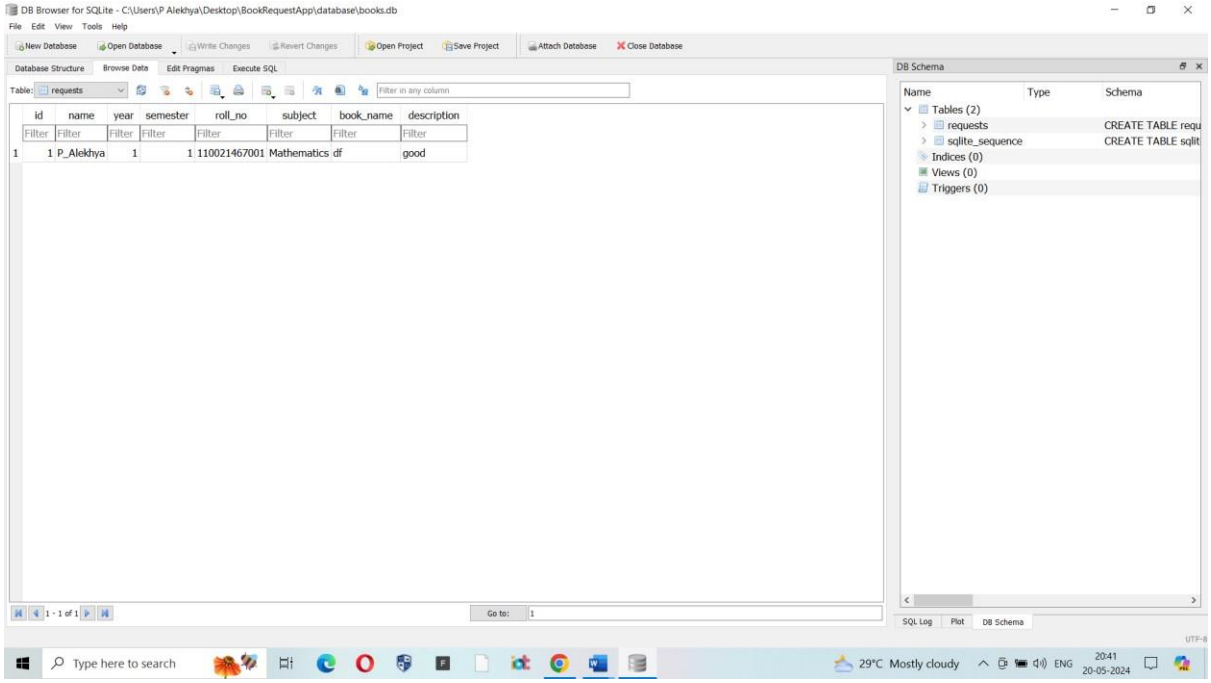
Login information:





Book Request information:


DB Browser for SQLite - C:\Users\P Alekhya\Desktop\BookRequestApp\database\books.db
File Edit View Tools Help
New Database
Open Database
Write Changes
Revert Changes
Open Project
Save Project
Attach Database
Close Database
Database Structure
Browse Data
Edit Pragmas
Execute SQL
Table: sqlite_sequence
Filter in any column
name seq
1 requests 1
DB Schema
Name Type Schema
Tables (2)
requests CREATE TABLE requ
sqlite_sequence CREATE TABLE sqlit
Indices (0)
Views (0)
Triggers (0)
SQL Log
Plot
DB Schema
Type here to search
29°C Mostly cloudy
ENG
20:40
20-05-2024


DB Browser for SQLite - C:\Users\P Alekhya\Desktop\BookRequestApp\database\books.db
File Edit View Tools Help
New Database
Open Database
Write Changes
Revert Changes
Open Project
Save Project
Attach Database
Close Database
Database Structure
Browse Data
Edit Pragmas
Execute SQL
Table: requests
Filter in any column
id name year semester roll_no subject book_name description
1 1 P_Alekhya 1 1 110021467001 Mathematics df good
DB Schema
Name Type Schema
Tables (2)
requests CREATE TABLE requ
sqlite_sequence CREATE TABLE sqlit
Indices (0)
Views (0)
Triggers (0)
SQL Log
Plot
DB Schema
Type here to search
29°C Mostly cloudy
ENG
20:41
20-05-2024

CONCLUSION AND FUTURE SCOPE

11. Conclusion and Future Scope

Summary:

The "Instant Library" project successfully addresses the challenges of traditional libraries by providing an efficient online system for accessing and requesting books. The project leverages modern web technologies and ensures a secure, user-friendly experience. With a clear focus on user registration, authentication, subject-specific content access, and book request functionalities, the system streamlines library interactions and enhances accessibility.

Future Enhancements:

- **User Roles and Permissions:** In the future, the system can be enhanced by implementing different user roles and permissions. This would allow for a more tailored user experience and ensure better management of the library resources. For example, roles such as administrators, librarians, and regular users can be defined, each with specific permissions. Administrators can have full access to manage users and resources, librarians can handle book requests and maintain the catalog, while regular users can access e-books and submit book requests.

- **Integration with External Systems:** To further improve the functionality and reach of the Instant Library, integration with external systems can be considered. This could include:

  - **Integration with School/University Systems:** Connecting with existing academic management systems for seamless user registration and data synchronization.

  - **Third-Party E-book Providers**: Partnering with third-party ebook providers to expand the collection of available resources.

  - **Library Networks**: Integrating with national and international library networks to enable resource sharing and inter-library loans.

- **Mobile Application:** Developing a mobile application would significantly enhance the accessibility and convenience of the Instant Library. The mobile app can offer features such as:

- **Push Notifications:** Keeping users informed about new book arrivals, due dates for borrowed e-books, and other important updates.

- **Offline Access**: Allowing users to download e-books and access them offline.

- **Enhanced User Interface:** Optimizing the user interface for mobile devices to ensure a smooth and intuitive experience.

Future Development Goals

- **Advanced Search Functionality**: Implementing a robust search engine to help users find books quickly and efficiently.

- **User Reviews and Ratings:** Allowing users to leave reviews and ratings for books to guide others in their reading choices.

- **Enhanced Security Features:** Continuously improving security measures to protect user data and ensure safe transactions.

BIBLIOGRAPHY

Reference:

- Documentation for HTML, CSS, JavaScript.
- Node.js and Express framework guides.
- SQLite3 database management resources.
- Best practices for web development and security.

Additional Reference:

- User feedback and testing reports.
- Technical articles and tutorials on web application development.

1. Duckett, J.(2018) HTML & CSS: Design and Build Websites. John Wiley & Sons.
   -This book provides a comprehensive introduction to HTML and CSS, covering the basics of web design and development.

2. McFarland, D. (2019). JavaScript & jQuery: The Missing Manual. O'Reilly Media.
   -This book offers a detailed exploration of JavaScript and jQuery, focusing on their application in web development.

3. Gauchat, A. (2014). CSS Secrets: Better Solutions to Everyday Web Design Problems. O'Reilly Media.
   -A collection of CSS techniques and best practices to enhance the design and styling of web pages.

4. W3Schools. (n.d.). HTML Tutorial. Retrieved from https://www.w3schools.com/html/HTML.
   - An online resource that provides comprehensive tutorials, examples, and references for

5. W3Schools. (n.d.). CSS Tutorial. Retrieved from https://www.w3schools.com/css/
   - This website offers detailed tutorials, examples, and references for CSS, covering various styling techniques.

This detailed project report outlines the comprehensive approach taken to develop the "Instant Library" system, ensuring a robust and user-friendly solution for accessing and managing library resources online. The future enhancements proposed aim to expand the system's functionality, improve user experience, and integrate with broader networks, ensuring the Instant Library continues to evolve to meet the needs of its users.