# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

# BELAGAVI-590014, KARNATAKA

## A Mini Project Report

**On**

## "ONLINE LIBRARY MANAGEMENT SYSTEM"

**Submitted in Partial Fulfilment of the Requirement for**

**"DBMS Laboratory with Mini Project -V Semester"**

**For the Award of Degree**

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE & ENGINEERING**


**Submitted By:**

**AMIT HEBBI (1SG17CS010)**

**MANAV PRADHAN (1SG17CS049)**

**Under the Guidance of:**

**Mr. Srikantha Gowda R**
Assistant Professor

**Mrs. Kavya N L**
Assistant Professor




**Department of Computer Science and Engineering**

**SAPTHAGIRI COLLEGE OF ENGINEERING**

**14/5, Chikkasandra, Hesarghatta Main Road Bengaluru – 560057**


**2019-2020**

# SAPTHAGIRI COLLEGE OF ENGINEERING
## 14/5, Chikkasandra, Hesaraghatta Main Road, Bengaluru – 560057.
## Department of Computer Science and Engineering


*Certificate*

Certified that the Mini Project Work entitled **"ONLINE LIBRARY MANAGEMENT SYSTEM"** carried out by **AMIT HEBBI (1SG17CS010) & MANAV PRADHAN (1SG17CS049),** bonafide students of **Sapthagiri College of Engineering**, in partial fulfilment for the award of **Bachelor of Engineering** degree in **Computer Science and Engineering** of **Visvesvaraya Technological University, Belagavi** during the academic year 2019-2020. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the department library. The mini project report has been approved as it satisfies the academic requirements in respect of DBMS Laboratory with Mini Project (17CSL58) prescribed for the said Degree.

| **Signature of the Guide** | **Signature of the Guide** | **Signature of the HOD** |
|---|---|---|
| **Mr. Srikantha Gowda R** | **Mrs. Kavya N L** | **Dr. Kamalakshi Naganna** |
| **Assistant Professor** | **Assistant Professor** | **Professor & HOD** |

**EXTERNAL EXAMINATION**

| **Name of the Examiners** | **Signature with Date** |
|---|---|
| 1._____ | _____ |
| 2._____ | _____ |

# ACKNOWLEDGEMENT

Any achievement does not depend solely on the individual efforts but on the guidance, encouragement and co-operation of intellectuals, elders and friends. A number of personalities, in their own capacities have helped us in carrying out this mini project work. We would like to take this opportunity to thank them all.

We would like to express my profound thanks to **Sri. G Dayanand,** Chairman, Sapthagiri College of Engineering Bangalore, for his continuous support in providing amenities to carry out this Mini Project.

Special Thanks to **Dr. N. Srinivasan,** Director, Sapthagiri College of Engineering Bangalore, for his valuable suggestion.

Also, we would like to express our immense gratitude to **Dr. H Ramakrishna,** Principal, Sapthagiri College of Engineering Bangalore, for his help and inspiration during the tenure of the course.

We also extend our sincere thanks to **Dr. Kamalakshi Naganna**, Professor and Head, Department of Computer Science and Engineering, Sapthagiri College of Engineering, for her constant support.

We would like to express our heartful gratitude to **Mr. Srikantha Gowda R,** Assistant professor and **Mrs**. **Kavya N L,** Assistant professor, Department of Computer Science and Engineering, Sapthagiri College of Engineering, for their timely advice on the mini project and regular assistance throughout the work.

We also extend our sincere thanks to all the **Faculty members** and **supporting staff** Department of Computer Science and Engineering, Sapthagiri College of Engineering, for their constant support and encouragement.

Finally, we thank our parents and friends for their moral support.

**Amit Hebbi**

**Manav Pradhan**

# ABSTRACT

Online Library Management System is a project which aims in developing a computerized system to maintain all the daily work of library .This project has many features which are generally not available in normal library management systems like facility of user login and a facility of admin login .It also has a facility of admin login through which the admin can monitor the whole system . It has also a facility where student after logging in their accounts can see list of books issued and its issue date and return date.

Overall this project of ours is being developed to help the students as well as staff of library to maintain the library in the best way possible and also reduce the human efforts.

# TABLE OF CONTENTS

# LIST OF FIGURES AND TABLES

# CHAPTER 1
# INTRODUCTION

This chapter gives an overview about the aim, objectives, background and operation environment of the system.

## 1.1 PROJECT AIMS AND OBJECTIVES

The project aims and objectives that will be achieved after completion of this project are discussed in this subchapter. The aims and objectives are as follows:

- Online book issue
- Request column for librarian for providing new books
- Student login page where student can find books issued by him/her and date of return.
- A search column to search availability of books

## 1.2 BACKGROUND OF PROJECT

Online 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 using a computerized system where he/she can 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. With this computerized system there will be no loss of book record or member record which generally happens when a non-computerized system is used.

## 1.3 OPERATION ENVIRONMENT

| | |
|---|---|
| PROCESSOR | INTEL CORE PROCESSOR OR BETTER PERFORMANCE |
| OPERATING SYSTEM | WINDOWS VISTA, WINDOWS7, UBUNTU |
| MEMORY | 1GB RAM OR MORE |
| HARD DISK SPACE | MINIMUM 3 GB FOR DATABASE USAGE |
| DATABASE | MY SQL |

Figure 1.1: Operation Environment Table

# CHAPTER 2
# SYSTEM ANALYSIS

In this chapter, we will discuss and analyse about the developing process of Online Library Management System including software requirement specification (SRS) and comparison between existing and proposed system. The functional and non-functional requirements are included in SRS part to provide complete description and overview of system requirement before the developing process is carried out. Besides that, existing vs proposed provides a view of how the proposed system will be more efficient than the existing one.

## 2.1 SOFTWARE REQUIREMENT SPECIFICATION

### 2.1.1 GENERAL DESCRIPTION

PRODUCT DESCRIPTION:

Online Library Management System is a computerized system which helps user(librarian) to manage the library daily activity in electronic format. It reduces 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.

PROBLEM STATEMENT:

The purpose of the Online Library Management System is to allow for storing details of a large number of books, magazines, journals, thesis and allow for add, search, borrow, return facilities separately to administrator/librarian and students.

### 2.1.2 SYSTEM OBJECTIVES

- Improvement in control and performance

  The system is developed to cope up with the current issues and problems of library. The system can add user, validate user and is also bug free.
- Save cost

  After computerized system is implemented less human force will be required to maintain the library thus reducing the overall cost.
- Save time

  Librarian is able to search record by using few clicks of mouse and few search keywords thus saving his valuable time.
- Option of online Notice board

  Librarian will be able to provide a detailed description of workshops going in the college as well as in nearby colleges.
- Lecture Notes

  Teacher have a facility to upload lectures notes in a pdf file having size not more than 10mb

### 2.1.3 SYSTEM REQUIREMENTS

#### *2.1.3.1 NON-FUNCTIONAL REQUIREMENTS*

PRODUCT REQUIREMENTS

- Efficiency Requirement

  When a library management system will be implemented librarian and user will easily access library as searching and book transaction will be very faster.

- Reliability Requirement

  The system should accurately perform member registration, member validation, report generation, book transaction and search

- Usability Requirement

  The system is designed for a user-friendly environment so that student and staff of library can perform the various tasks easily and in an effective way.

ORGANIZATIONAL REQUIREMENT

- Implementation Requirement

  In implementing whole system, it uses html in front end with php as server-side scripting language which will be used for database connectivity and the backend i.e. the database part is developed using MySQL.

- Delivery Requirement

  The whole system is expected to be delivered in six months of time with a weekly evaluation by the project guide.

### *2.1.3.2 FUNCTIONAL REQUIREMENTS*

#### 1.1 USER LOGIN

Description of feature

This feature used by the user to login into system. They are required to enter user id and password before they are allowed to enter the system. The user id and password will be verified and if invalid id is there user is allowed to not enter the system.

Functional requirements

- User id is provided when they register
- The system must only allow user with valid id and password to enter the system
- The system performs authorization process which decides what user level can access to.
- The user must be able to logout after they finished using system.

## 1.2 REGISTER NEW USER

Description of feature

This feature can be performed by all users to register new user to create account.

Functional requirements

- System must be able to verify information
- System must be able to delete information if information is wrong

Admin Features

- Admin Dashboard
- Admin can add/update/ delete category
- Admin can add/update/ delete author
- Admin can add/update/ delete books
- Admin can issue a new book to student and also update the details when student return book
- Admin can search student by using their student ID
- Admin can also view student details
- Admin can change own password

Students Features

- Student can register yourself and after registration they will get student id
- After login student can view own dashboard.
- Student can update own profile.
- Student can view issued book and book return date-time.
- Student can also change own password.
- Student can also recover own password.

### 2.1.4 SOFTWARE AND HARDWARE REQUIREMENTS

This section describes the software and hardware requirements of the system

#### *2.1.4.1 SOFTWARE REQUIREMENTS*

- Operating system- Windows 7 is used as the operating system as it is stable and supports more features and is more user friendly
- Database MYSQL-MYSQL is used as database as it easy to maintain and retrieve records by simple queries which are in English language which are easy to understand and easy to write.
- Development tools and Programming language- HTML is used to write the whole code and develop webpages with CSS, java script for styling work and php for sever side scripting.

#### *2.1.4.2 HARDWARE REQUIREMENTS*

- Intel core i5 2nd generation is used as a processor because it is fast than other processors an provide reliable and stable and we can run our pc for long time. By using this processor, we can keep on developing our project without any worries.
- Ram 1 GB is used as it will provide fast reading and writing capabilities and will in turn support in processing

## 2.2 EXISTING VS PROPOSED SYSTEM

i. Existing system does not have any facility of teacher's login or student login whereas proposed system will have a facility of student login as well as teacher's login
ii. Existing system does not have a facility of online reservation of books whereas proposed system has a facility of online reservation of books
iii. Existing system does not have any facility of online notice board where description of workshops happening in our college as well as nearby colleges is being provided.

iv. Existing system does not have any option of lectures notes uploaded by teachers whereas proposed system will have this facility

v. Existing system does not have any facility to generate student reports as well book issue reports whereas proposed system provides librarian with a tool to generate reports

vi. Existing system does not have any facility for book request and suggestions where as in proposed system after logging in to their accounts student can request books as well as provide suggestions to improve library

## 2.3 SOFTWARE TOOLS USED

The whole Project is divided in two parts the front end and the back end.

### *2.3.1 FRONT END*

The front end is designed using of HTTML, Php, CSS, Java script

- HTML- **HTML** or **Hyper Text Mark-up Language** is the main mark-up language for creating web pages and other information that can be displayed in a web browser.HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets (like <*html*>), within the web page content. HTML tags most commonly come in pairs like <*h1*> and <*/h1*>, although some tags represent empty elements and so are unpaired, for example <*img*>. The first tag in a pair is the start tag, and the second tag is the end tag (they are also called opening tags and closing tags). In between these tags' web designers can add text, further tags, comments and other types of text-based content. The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page.HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behaviour of HTML web pages.

- CSS- **Cascading Style Sheets (CSS)** is a style sheet language used for describing the look and formatting of a document written in a mark-up language. While most often used to style web pages and interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. CSS is a cornerstone specification of the web and almost all web pages use CSS style sheets to describe their presentation.CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colours, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content (such as by allowing for table less web design).CSS can also allow the same mark-up page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on Braille-based, tactile devices. It can also be used to allow the web page to display differently depending on the screen size or device on which it is being viewed. While the author of a document typically links that document to a CSS file, readers can use a different style sheet, perhaps one on their own computer, to override the one the author has specified. However, if the author or the reader did not link the document to a specific style sheet the default style of the browser will be applied.CSS specifies a priority scheme to determine which style rules apply if more than one rule matches against a particular element. In this so-called cascade, priorities or weights are calculated and assigned to rules, so that the results are predictable.

- JAVA SCRIPT- **JavaScript (JS)** is a dynamic computer programming language. It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed. It is also being used in server-side programming, game development and the creation of desktop and mobile applications. JavaScript is a prototype-based scripting language with dynamic typing and has first- class functions. Its syntax was influenced by C. JavaScript copies many names and naming conventions from Java, but the two languages are otherwise unrelated and have very different semantics. The key

design principles within JavaScript are taken from the Self and Scheme programming languages. It is a multi- paradigm language, supporting object-oriented, imperative, and functional programming styles. The application of JavaScript to use outside of web pages—for example, in PDF documents, site-specific browsers, and desktop widgets—is also significant. Newer and faster JavaScript VMs and platforms built upon them (notably Node.js) have also increased the popularity of JavaScript for server-side web applications. On the client side, JavaScript was traditionally implemented as an interpreted language but just-in-time compilation is now performed by recent (post-2012) browsers.

- PHP- **PHP** is a server-side scripting language designed for web development but also used as a general-purpose programming language. PHP is now installed on more than 244 million websites and 2.1 million web servers. Originally created by Rasmus Lerdorf in 1995, the reference implementation of PHP is now produced by The PHP Group. While PHP originally stood for Personal Home Page, it now stands for PHP: Hypertext Pre-processor, a recursive backronym. PHP code is interpreted by a web server with a PHP processor module, which generates the resulting web page: PHP commands can be embedded directly into an HTML source document rather than calling an external file to process data. It has also evolved to include a command-line interface capability and can be used in standalone graphical applications. PHP is free software released under the PHP License. PHP can be deployed on most web servers and also as a standalone shell on almost every operating system and platform, free of charge.

### *2.3.2 BACK END*

The back end is designed using MySQL which is databases used to design the database.

- MYSQL- **MySQL** ("My S-Q-L", officially, but also called "My Sequel") is (as of July 2013) the world's second most widely used open-source relational database management system (RDBMS). It is named after co-founder Michael Widenius daughter, My. The SQL phrase stands for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU

General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for- profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation .MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack (and other 'AMP' stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Free-software-open source projects that require a full-featured database management system often use MySQL. For commercial use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: TYPO3, MODx, Joomla, WordPress, phpBB, MyBB, Drupal and other software. MySQL is also used in many high-profile, large-scale websites, including Wikipedia, Google (though not for searches), Facebook, Twitter, Flickr, and YouTube.

# CHAPTER 3

# SYSTEM DESIGN

## 3.1 TABLE DESIGN

Various Tables to Maintain Information:

- **admin** *table*

**admin**

| Column | Type | Null | Default | Links to |
|---|---|---|---|---|
| id *(Primary)* | int(11) | No | | |
| FullName | varchar(100) | Yes | *NULL* | |
| AdminEmail | varchar(120) | Yes | *NULL* | |
| UserName | varchar(100) | No | | |
| Password | varchar(100) | No | | |
| updationDate | timestamp | No | 0000-00-00 00:00:00 | |

**Indexes**

| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
|---|---|---|---|---|---|---|---|---|
| PRIMARY | BTREE | Yes | No | id | 1 | A | No | |

Figure 3.1.1: Admin Table Design

- Table **tblcategory** track the record of category

**tblcategory**

| Column | Type | Null | Default | Links to |
|---|---|---|---|---|
| id *(Primary)* | int(11) | No | | |
| CategoryName | varchar(150) | Yes | *NULL* | |
| Status | int(1) | Yes | *NULL* | |
| CreationDate | timestamp | Yes | CURRENT_TIMESTAMP | |
| UpdationDate | timestamp | No | 0000-00-00 00:00:00 | |

**Indexes**

| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
|---|---|---|---|---|---|---|---|---|
| PRIMARY | BTREE | Yes | No | id | 4 | A | No | |

Figure 3.1.2: Category Table Design

❖ Table **tblauthors** track the details of authors

## tblauthors

| Column | Type | Null | Default | Links to |
|---|---|---|---|---|
| id *(Primary)* | int(11) | No | | |
| AuthorName | varchar(159) | Yes | *NULL* | |
| creationDate | timestamp | Yes | CURRENT_TIMESTAMP | |
| UpdationDate | timestamp | Yes | *NULL* | |

**Indexes**

| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
|---|---|---|---|---|---|---|---|---|
| PRIMARY | BTREE | Yes | No | id | 6 | A | No | |

Figure 3.1.3: Authors Table Design

❖ Table **tblstudents** for students record and student login details

## tblstudents

| Column | Type | Null | Default | Links to |
|---|---|---|---|---|
| id *(Primary)* | int(11) | No | | |
| StudentId | varchar(100) | Yes | *NULL* | |
| FullName | varchar(120) | Yes | *NULL* | |
| EmailId | varchar(120) | Yes | *NULL* | |
| MobileNumber | char(11) | Yes | *NULL* | |
| Password | varchar(120) | Yes | *NULL* | |
| Status | int(1) | Yes | *NULL* | |
| RegDate | timestamp | Yes | CURRENT_TIMESTAMP | |
| UpdationDate | timestamp | Yes | *NULL* | |

**Indexes**

| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
|---|---|---|---|---|---|---|---|---|
| PRIMARY | BTREE | Yes | No | id | 7 | A | No | |
| StudentId | BTREE | Yes | No | StudentId | 7 | A | Yes | |

Figure 3.1.4: Students Table Design

❖ Table **tblissuedbookdetails** for maintain issued book and returned book record

## tblissuedbookdetails

| Column | Type | Null | Default | Links to |
|---|---|---|---|---|
| id *(Primary)* | int(11) | No | | |
| BookId | int(11) | Yes | *NULL* | tblbooks -> id |
| StudentID | varchar(150) | Yes | *NULL* | tblstudents -> StudentId |
| IssuesDate | timestamp | Yes | CURRENT_TIMESTAMP | |
| ReturnDate | timestamp | Yes | *NULL* | |
| RetrunStatus | int(1) | No | | |
| fine | int(11) | Yes | *NULL* | |

**Indexes**

| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
|---|---|---|---|---|---|---|---|---|
| PRIMARY | BTREE | Yes | No | id | 8 | A | No | |

Figure 3.1.5: Book Details Table Design

❖ Table **tbalbooks** for books records

## tblbooks

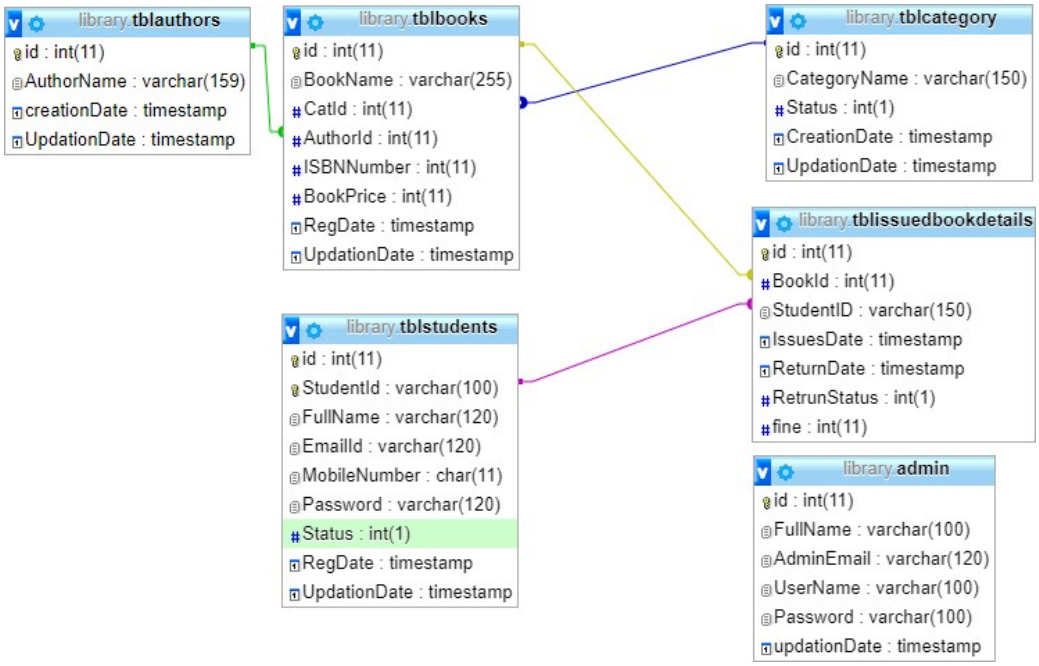
| Column | Type | Null | Default | Links to |
|---|---|---|---|---|
| id *(Primary)* | int(11) | No | | |
| BookName | varchar(255) | Yes | *NULL* | |
| CatId | int(11) | Yes | *NULL* | tblcategory -> id |
| AuthorId | int(11) | Yes | *NULL* | tblauthors -> id |
| ISBNNumber | int(11) | Yes | *NULL* | |
| BookPrice | int(11) | Yes | *NULL* | |
| RegDate | timestamp | Yes | CURRENT_TIMESTAMP | |
| UpdationDate | timestamp | Yes | *NULL* | |

**Indexes**

| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
|---|---|---|---|---|---|---|---|---|
| PRIMARY | BTREE | Yes | No | id | 2 | A | No | |

Figure 3.1.6: Books Table Design

❖ SCHEMA DIAGRAM




***ADMIN***

| Id (primary) | FullName | AdminEmail | Username | password | updationDate |
|---|---|---|---|---|---|

***TBLCATEGORY***

| Id (primary) | CategoryName | Status | CreationDate | UpdationDate |
|---|---|---|---|---|

***TBLAUTHORS***

| Id (primary) | AunthorName | CreationDate | UpdationDate |
|---|---|---|---|

***TBLSTUDENTS***

| Id (primary) | StudentId | FullName | EmailId | MobileNo | Password | Status | RegDate | UpdationDate |
|---|---|---|---|---|---|---|---|---|

***TBLISSUEDBOOKDETAILS***

| Id (primary) | BookId | StudentId | IssueDate | ReturnDate | ReturnStatus | Fine |
|---|---|---|---|---|---|---|

***TBLBOOKS***

| Id (primary) | BookName | CatId | AuthorId | ISBNNumber | BookPrice | RegDate | UpdationDate |
|---|---|---|---|---|---|---|---|

Figure 3.1.7: Schema Diagram

## 3.2 DATA FLOW DIAGRAMS

Data Flow Diagram for Admin Login

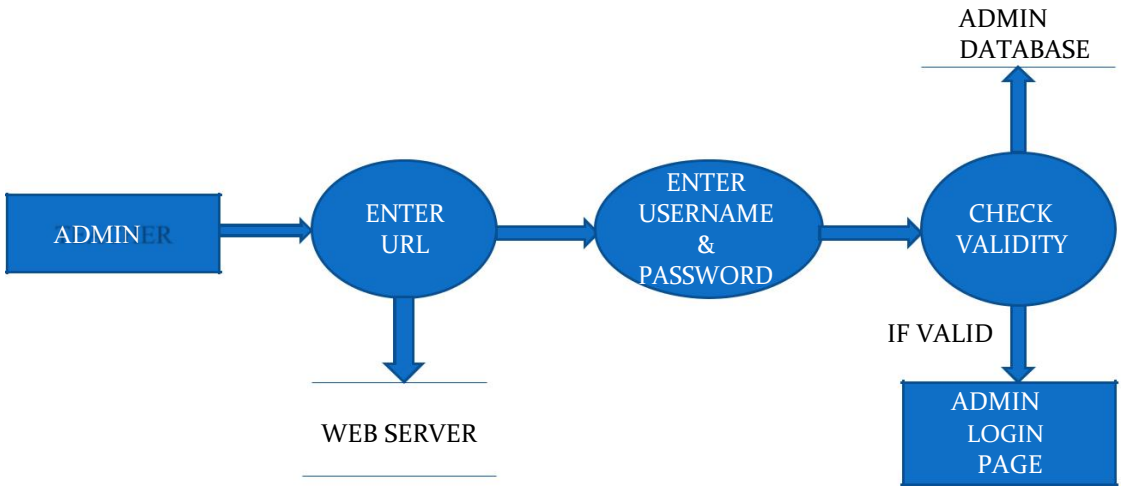



Figure 3.2.1: Data Flow Diagram for Admin Login

After entering to the home page of the website, Admin can choose the ADMIN LOGIN option where they are asked to enter username & password, and if he/she is a valid user then a teacher login page will be displayed.

Admin Dataflow Diagram After Login

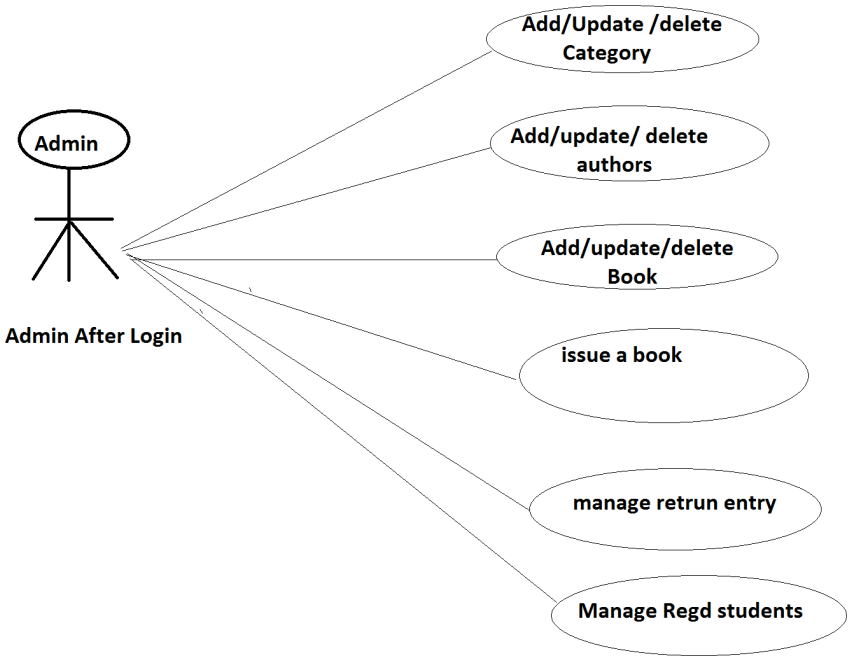



Figure 3.2.2: Admin Dataflow Diagram

Data Flow Diagram for Issuing Book

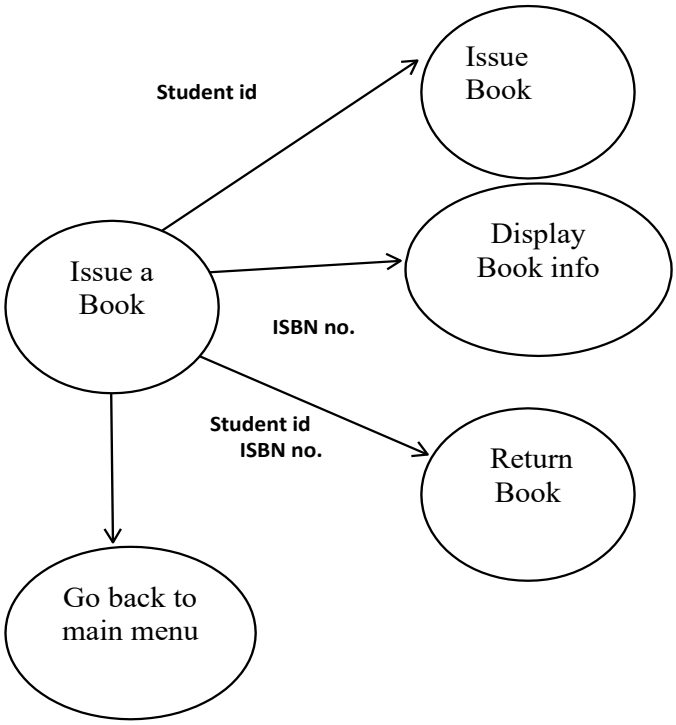



Figure 3.2.3: Data Flow Diagram for Issuing Book

Students After Login

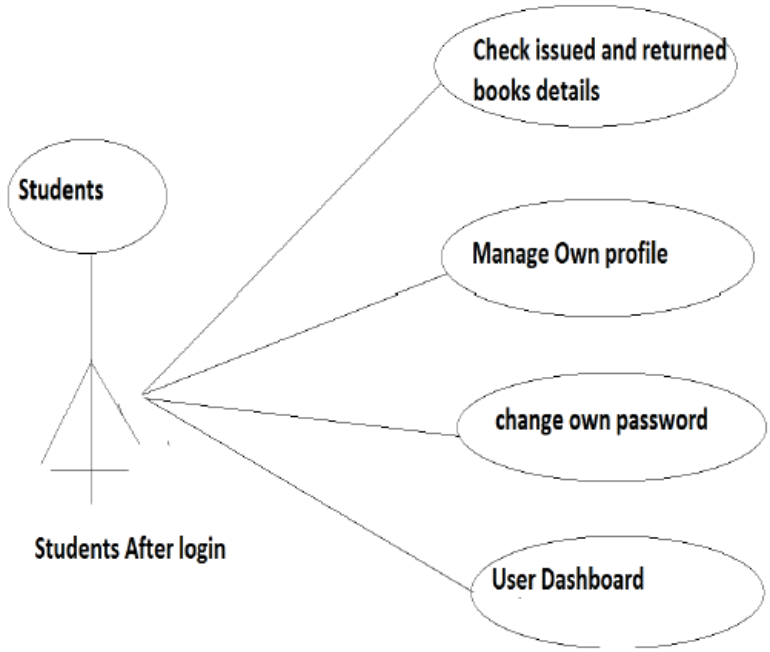



Figure 3.2.4: Data Flow Diagram for Students After Login

## 3.3 ENTITY RELATIONSHIP DIAGRAM

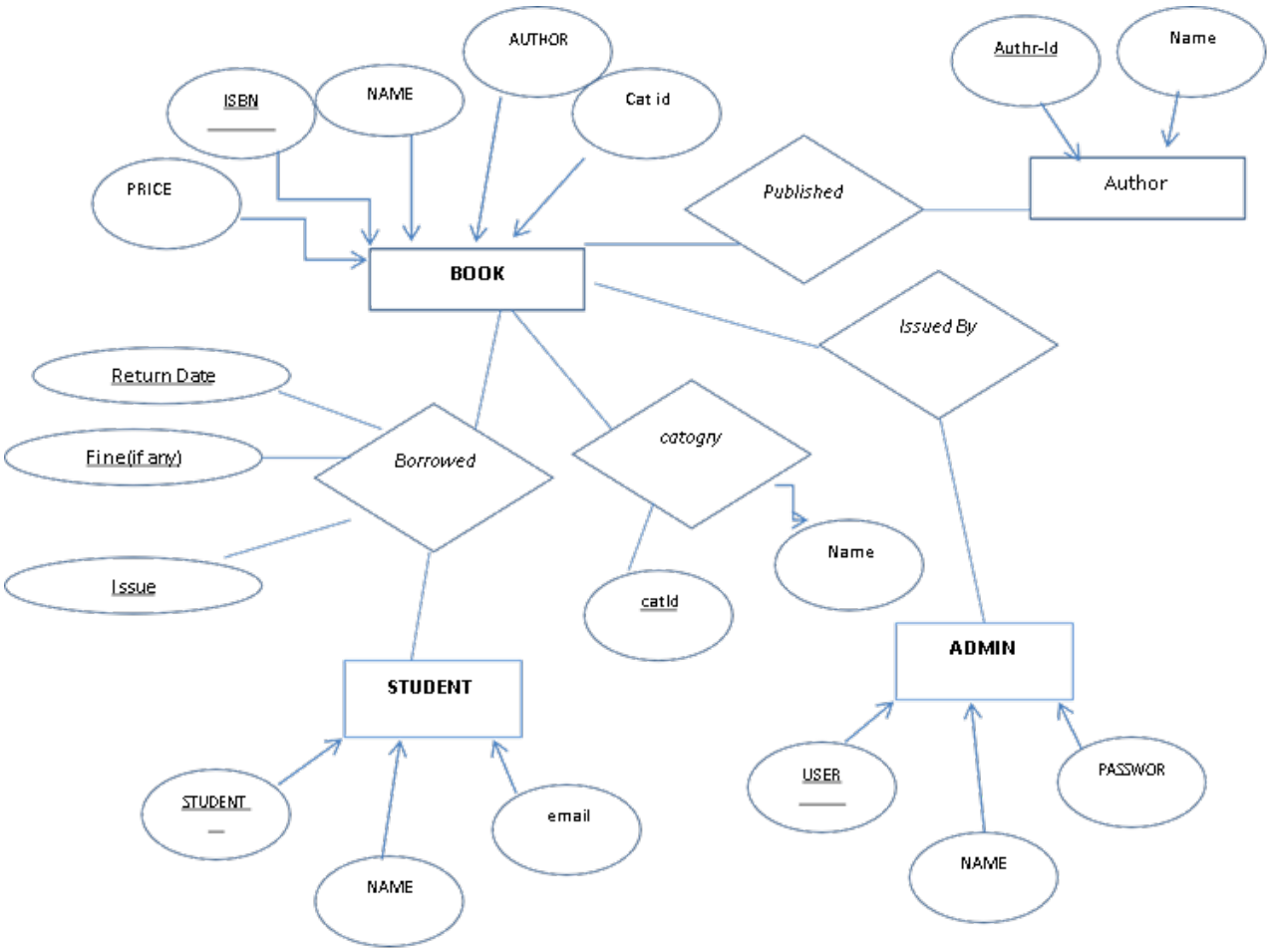



Figure 3.3: Entity Relationship Diagram

## 3.4 TESTING

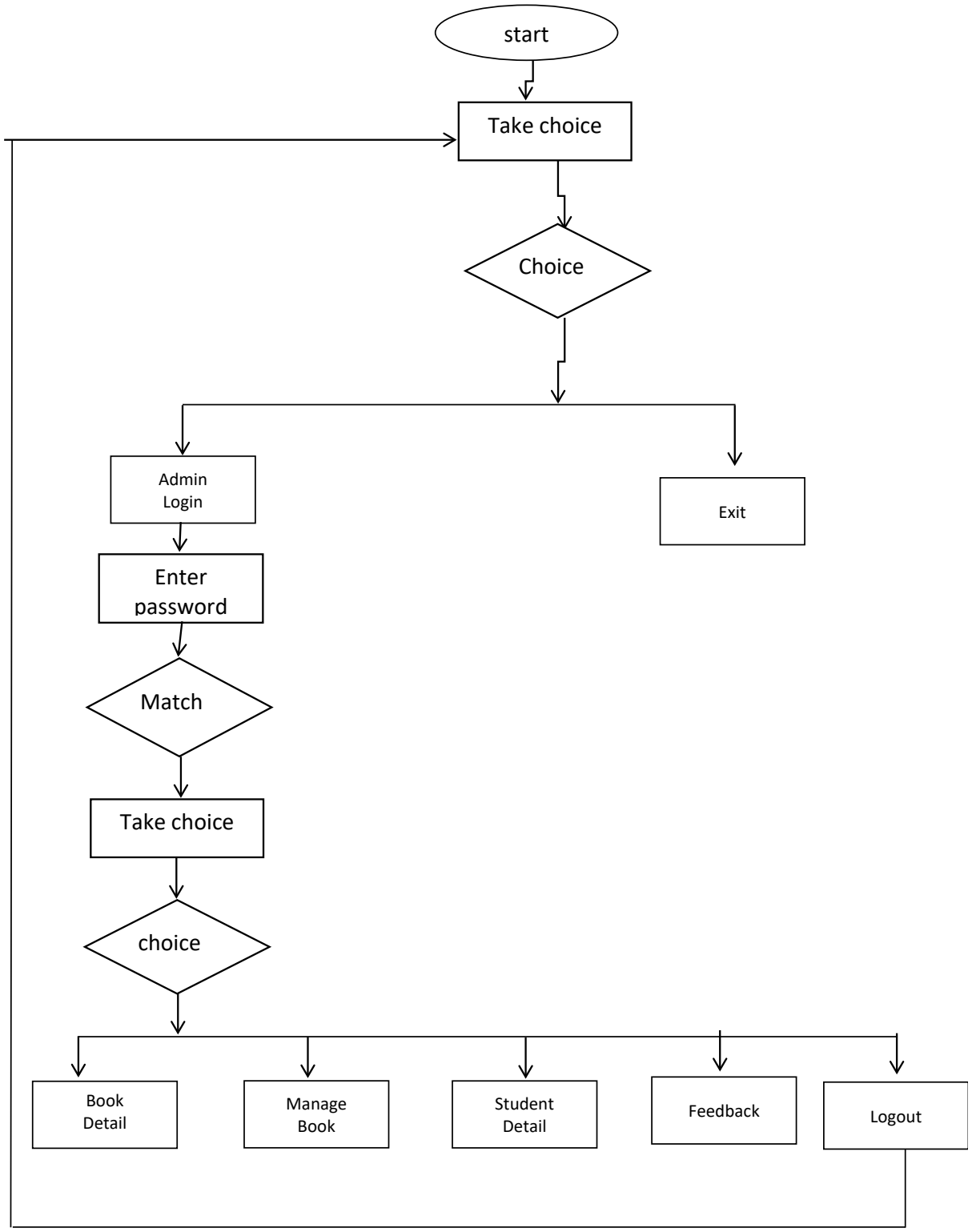



Figure 3.4: Testing Diagram

# CHAPTER 4

# SYSTEM IMPLEMENTATION

## 4.1. SCREEN SHOTS

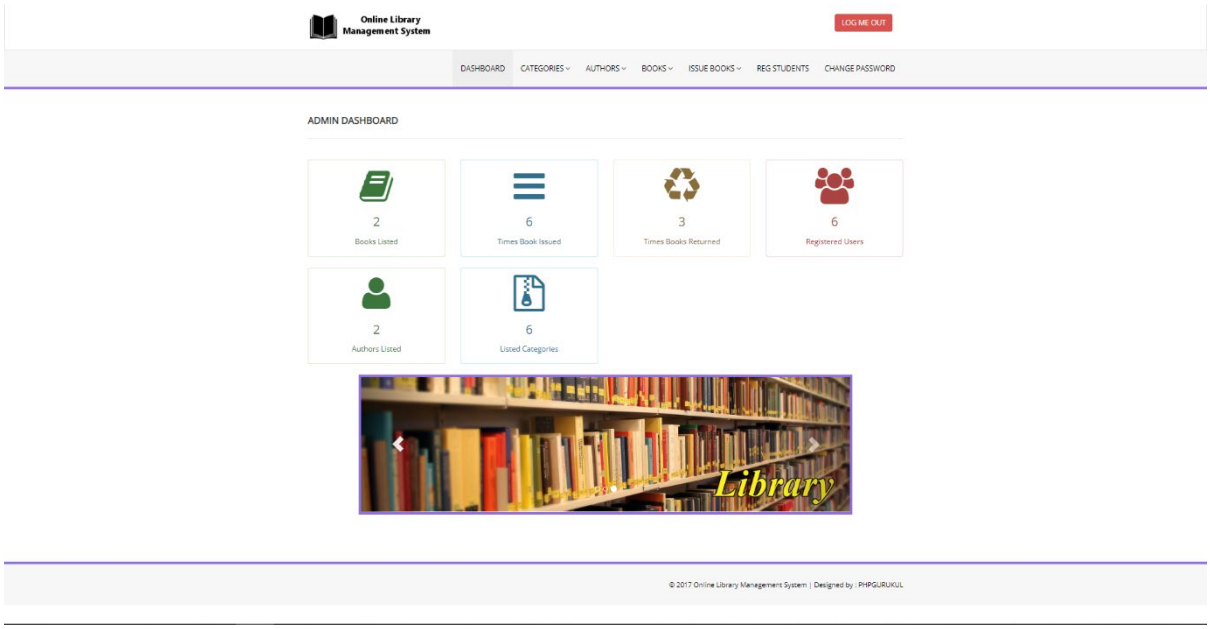



Figure 4.1.1: Admin Dashboard

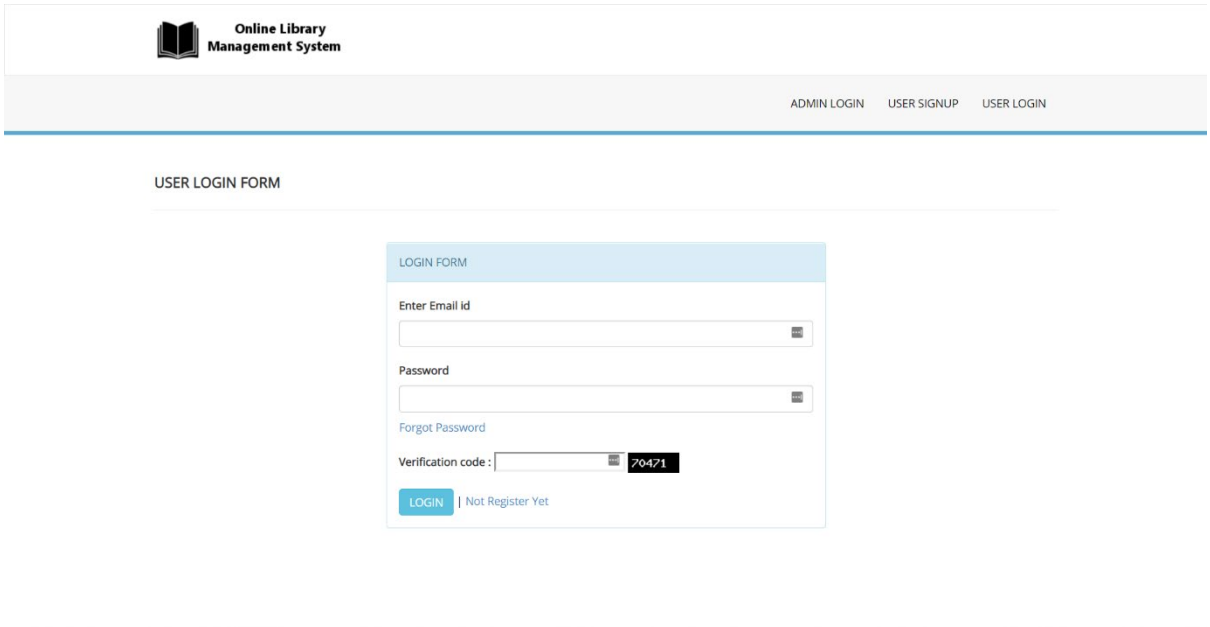



Figure 4.1.2: User Login Page

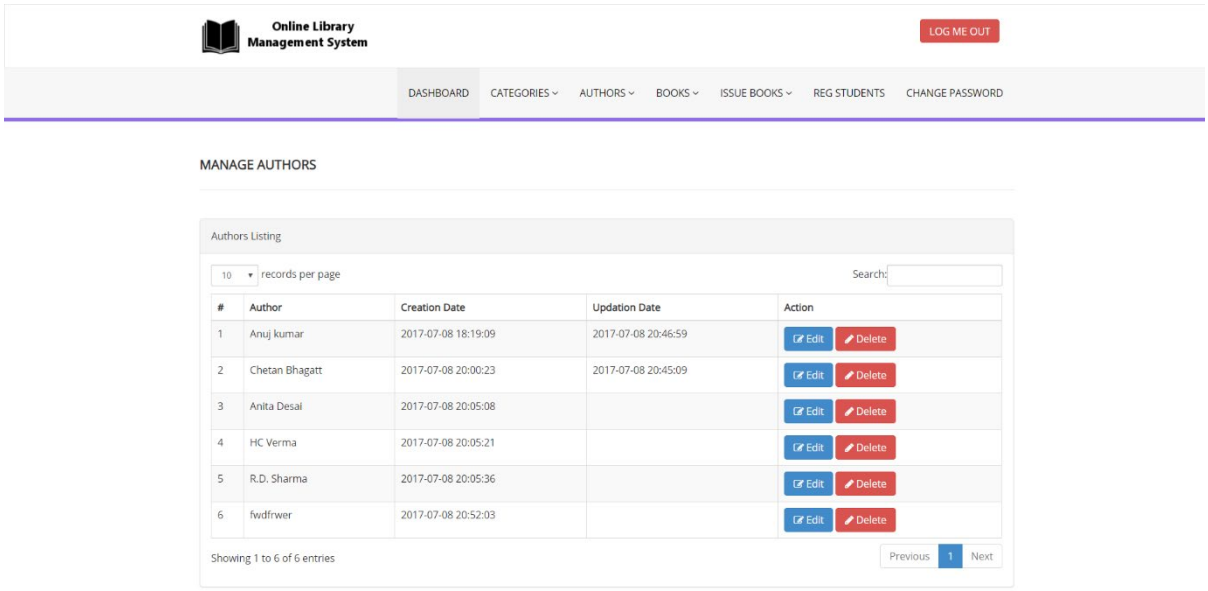



Figure 4.1.3: Book Authors

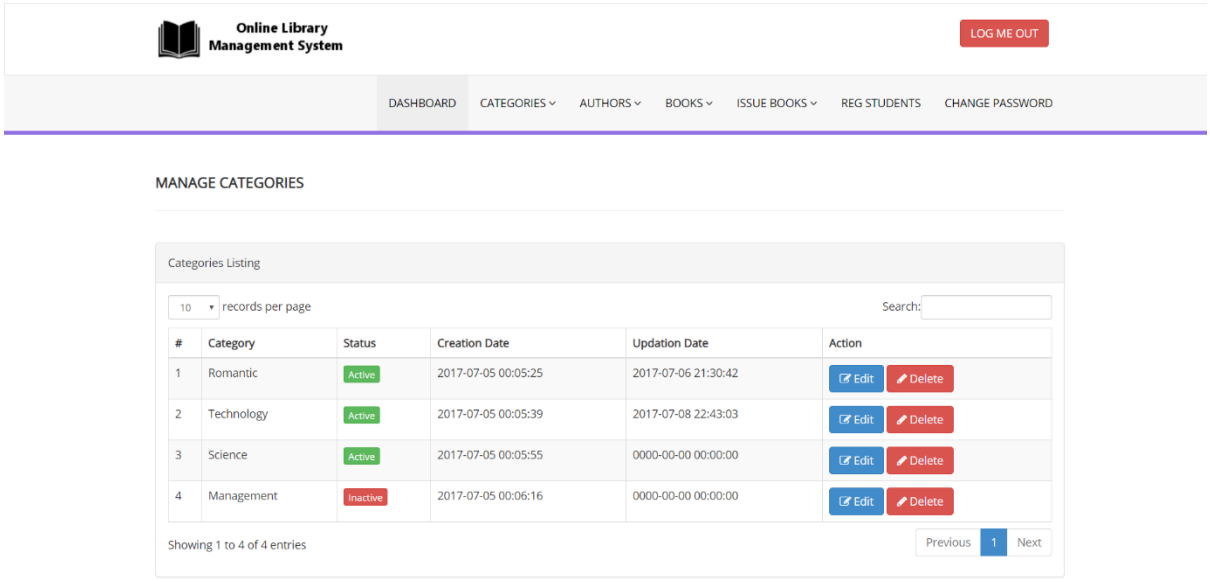



Figure 4.1.4: Book Categories

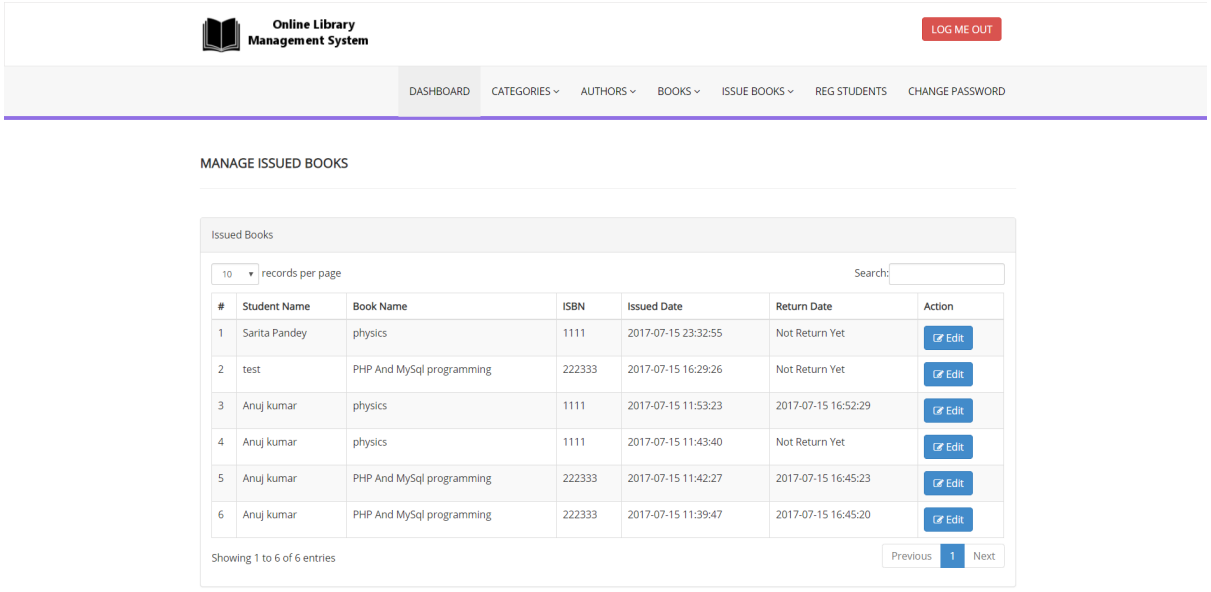



Figure 4.1.5: Issued Books

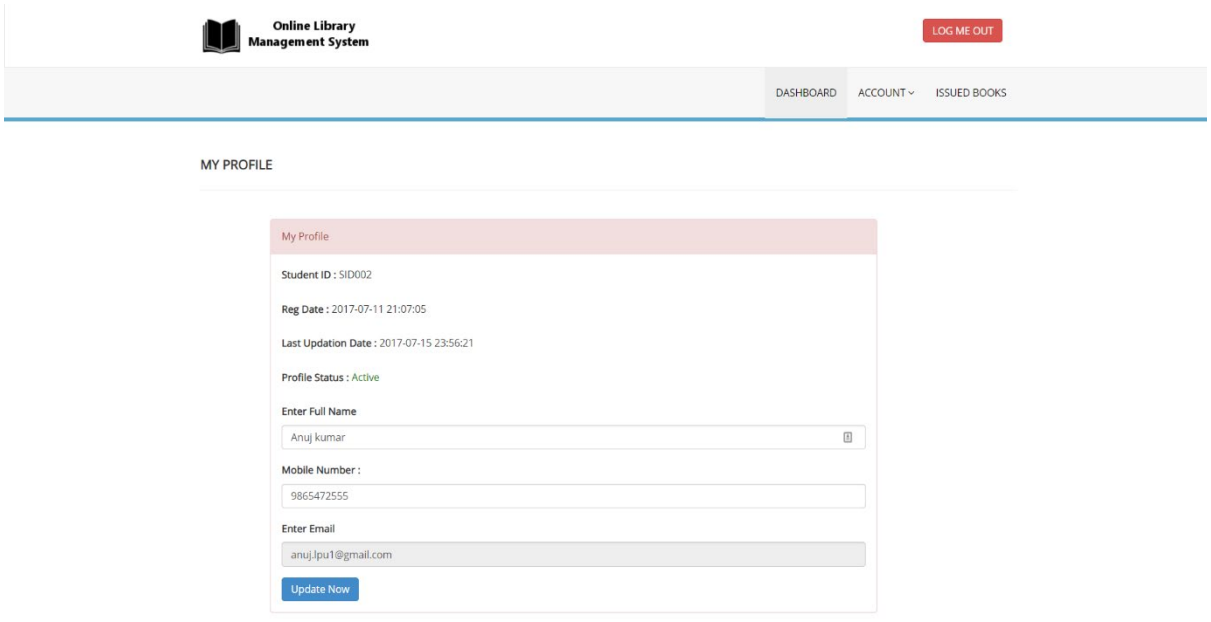



Figure 4.1.6: User Profile

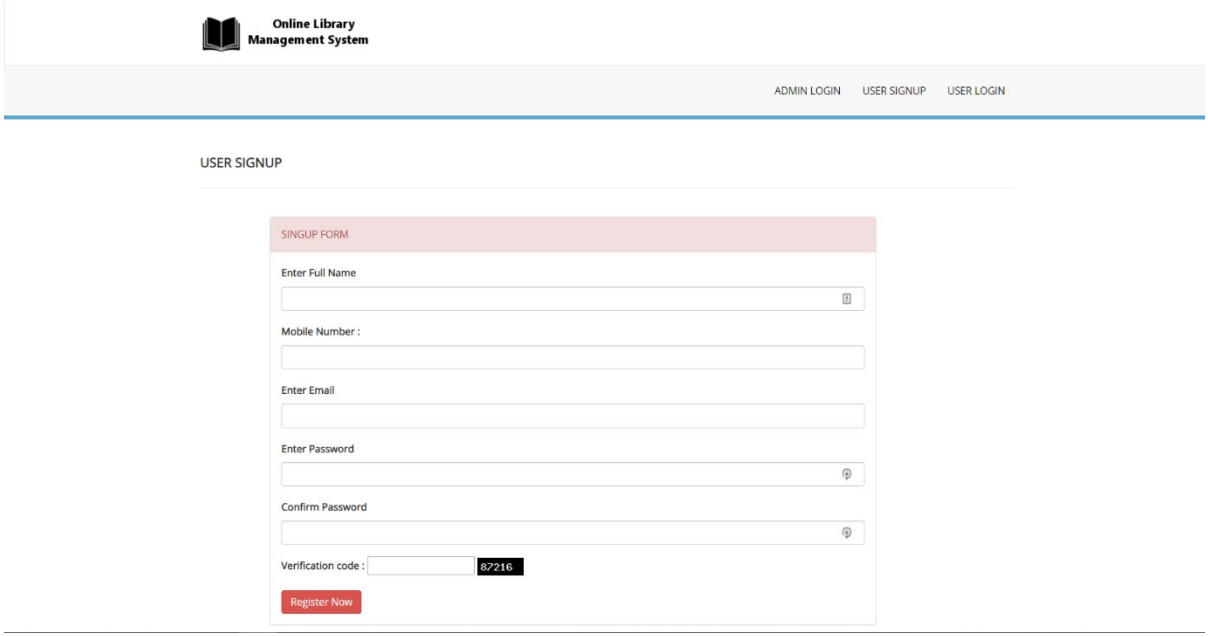



Figure 4.1.7: User Signup

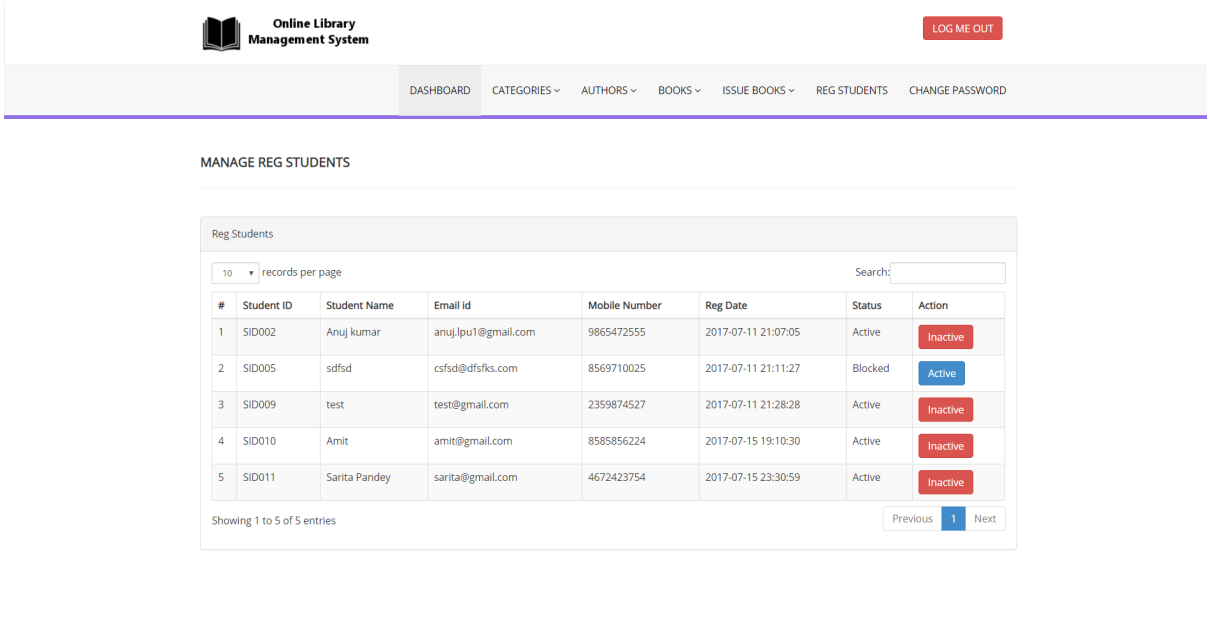



Figure 4.1.8: Registered Students

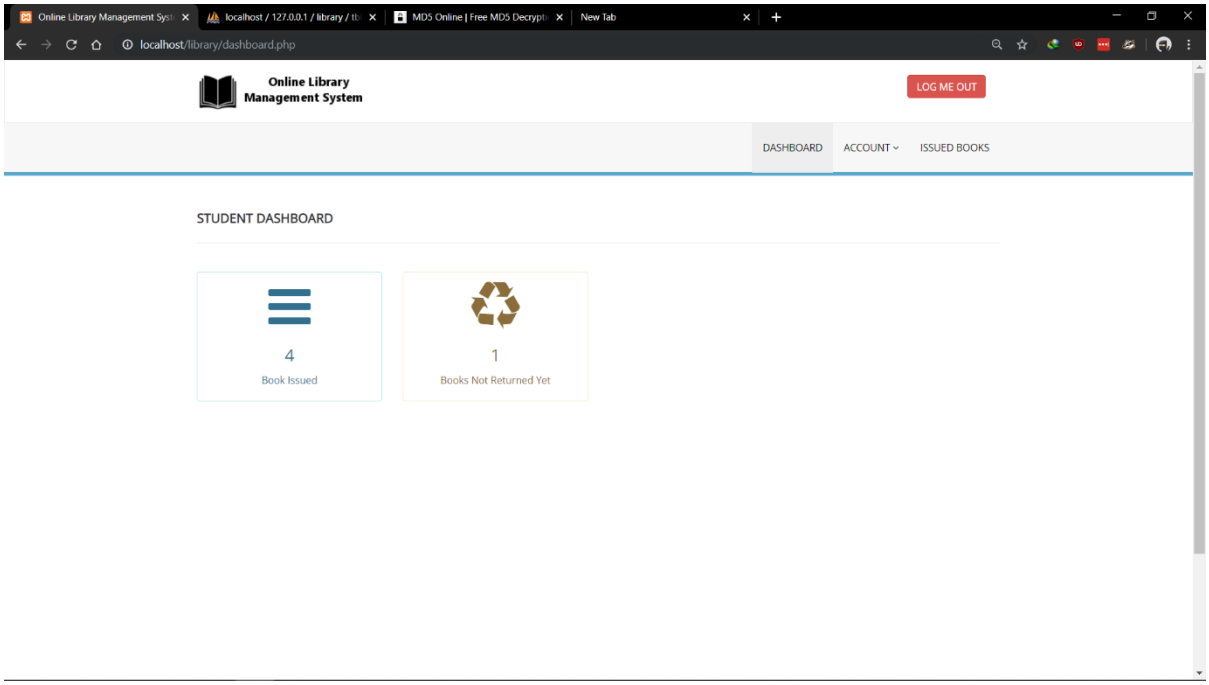



Figure 4.1.9: Student Dashboard

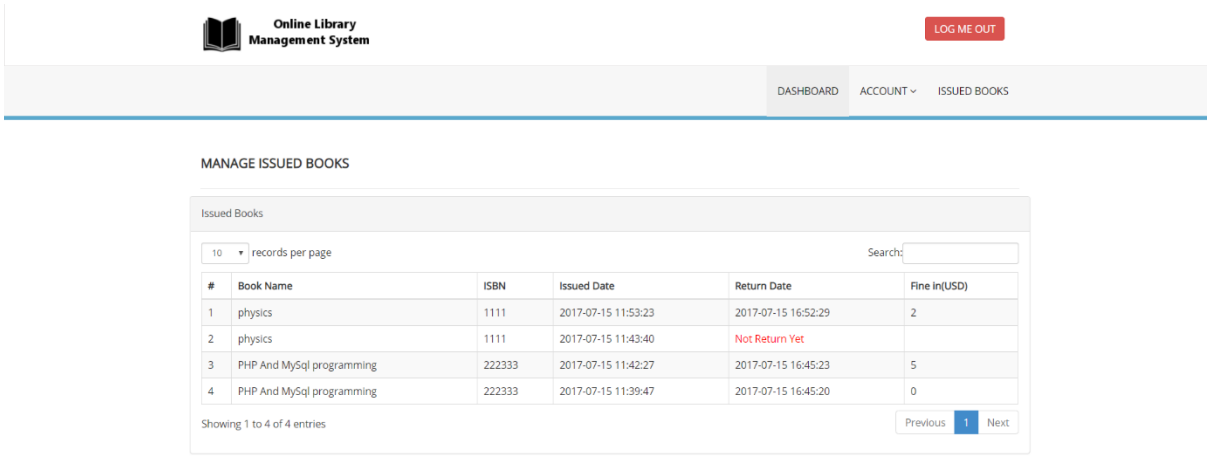



Figure 4.1.10: Issued books

# CHAPTER 5
# SYSTEM TESTING

The aim of the system testing process was to determine all defects in our project. The program was subjected to a set of test inputs and various observations were made and based on these observations it will be decided whether the program behaves as expected or not. Our Project went through two levels of testing

1. Unit testing
2. Integration testing

## 5.1 UNIT TESTING

Unit testing is undertaken when a module has been created and successfully reviewed. In order to test a single module, we need to provide a complete environment is besides the module we would require

- The procedures belonging to other modules that the module under test calls
- Non local data structures that module accesses
- A procedure to call the functions of the module under test with appropriate parameters

1. Test For the admin module
    - Testing admin login form-This form is used for log in of administrator of the system. In this we enter the username and password if both are correct administration page will open otherwise if any of data is wrong it will get redirected back to the login page and again ask for username and password
    - Student account addition- In this section the admin can verify student details from student academic info and then only add student details to main library database it contains add and delete buttons if user click add button data will be added to student database and if he clicks delete button the student data will be deleted

- Book Addition- Admin can enter details of book and can add the details to the main book table also he can view the books requests.

2. Test for Student login module
   - Test for Student login Form-This form is used for log in of Student .In this we enter the library id, username and password if all these are correct student login page will open otherwise if any of data is wrong it will get redirected back to the login page and again ask for library id, username and password.
   - Test for account creation- This form is used for new account creation when student does not fill the form completely it asks again to fill the whole form when he fill the form fully it gets redirected to page which show waiting for conformation message as his data will be only added by administrator after verification.
   - Test for teacher login module-
   - Test for teacher login form- This form is used for logging of teacher. In this we enter the username and password if all these are correct teacher login page will open otherwise if any of data is wrong it will get redirected back to the login page and again ask for username and password.

## 5.2 INTEGRATION TESTING

In this type of testing we test various integration of the project module by providing the input. The primary objective is to test the module interfaces in order to ensure that no errors are occurring when one module invokes the other module.

# CHAPTER 6
# CONCLUSION & FUTURE SCOPE

This website provides a computerized version of library management system which will benefit the students as well as the staff of the library.

It makes entire process online where student can search books, staff can generate reports and do book transactions. It also has a facility for student login where student can login and can see status of books issued as well request for book or give some suggestions. It has a facility of teacher's login where teachers can add lectures notes and also give necessary suggestion to library and also add info about workshops or events happening in our college or nearby college in the online notice board.

There is a future scope of this facility that many more features such as online lectures video tutorials can be added by teachers as well as online assignments submission facility , a feature of group chat where students can discuss various issues of engineering can be added to this project thus making it more interactive more user friendly and project which fulfils each users need in the best way possible

# REFERENCES


- http://www.w3schools.com/html/html_intro.asp
- https://www.w3schools.com/php/default.asp
- https://www.w3schools.com/sql/default.asp
- Fundamentals of software engineering by Rajib mall, PHIlearning
- Web development and application