

**CMPE 226**

**Fall 2014**

**DEPARTMENT OF COMPUTER ENGINEERING**

**Library Management System Database Project**

**Submitted to**

Dr. Hungwen Li

**Submitted by**

Geetha Anne - 008741057

Layla Reza - 008716058

**TABLE OF CONTENTS:**

1. **Executive Summary………………………………………………………….. 3**
2. **Introduction……………………………………..……………………………… 4**
3. **Background Work……………………………………………….……………. .5**
4. **Objectives………………………………………………………………………… 6**
5. **Approach and Methodology………………………………………………….6**
6. **Major Findings………………………………………………….……………… 19**
7. **Analysis…………………………….………………………………………………19**
8. **Conclusion…………………………………………………………………………20**
9. **References………………………………………………………………………..20**

**Executive Summary**

Conventionally the readers have to go to library for searching books & magazines and they may have to stand in long reservation lines for reserving/issuing the material. This kind of functionality leads in wastage of time and extra efforts. Here we have provided an online facility through which anyone, anywhere in campus, having a library account can search for and reserve books, find out who is having a particular book and put in a request to library. This particular automated online solution will eliminate the problems related to the manual errors and time consuming tasks. PHP and MYSQL are utilized for the creation of the beautiful front end for user interactions and for data storage respectively.

**Introduction:**

In the present computerized world, Library management system plays a vital role in all the educational institutions. Generally, every institution be it a high school, college or University, their libraries maintain huge collections of journals, text books, articles and various publications. Traditionally library book management was done manually and required more man power.

**Purpose:**

LIBRARY MANAGEMENT SYSTEM is a software application to maintain the records related to Book borrowing, Stock Maintenance, Book Search, Catalog, Book Issue, Book Returns, Fine Collection, and all necessary requirements for the Library to manage day to day operations.

**Scope:**

The scope of the project is to create a web based library management system. A web based application can create the mobility to operate even when the admin is out of office. All categories of information like status of books, article, journals and publications can be updated or viewed using this website. The returns and fine details of the books borrowed by students will be maintained. All the web pages are created using PHP and MYSQL is used as the required database. Initially all the required web pages are created along with the database tables. The required business logic will be added to improve the scope of the application and the database connections are established.

**Background work:**

Library management system has a wider scope towards the educational institutions these days. In general each and every educational institution right from a high school to Universities, maintain ample number of books, journals, articles and different publications with them. Initially library management is done manually and the required records are saved across books. The common details that were maintained across libraries include the overall information of all kinds of books against different publications and also the details of the issues and refunds of the corresponding books.

The manual approach followed initially is fine if the case is limited to few books and as the scope of library is improving these days, a perfect automated system is required in this context. There were many automated software’s developed in this context and most of them are successful in maintain the library information as per the required standards of schools and universities as well. The main scope of the library management system mainly depends on few aspects like the number of publications and books maintained and also the nature of the client like either it is targeted for universities, colleges or schools.

**Objectives:**

**Aim**: To develop a web based library management system using PHP and MYSQL database**.**

Following are the key objectives

1. To review different types of library management systems what were popular and identify the key advantages and limitations
2. To design a web based library management system using the relational database concepts and Entity Relational Models.
3. To create the required web pages and add the PHP code
4. To develop the required database using MYSQL and establish the database connections required.
5. To test the application against few test cases and run the application to generalize a web based library management that suits for schools, colleges and universities as well.

**Approach and Methodology**

As discussed in the background section, scope of the project is to create a web based library management system. The main reason behind to create a web based application is that to create the mobility to operate even when the admin is out of office.

All the status of books, article, journals and publications can be updated or viewed using this website and also all the issues and returns of the books can be maintained against the student details. All the web pages are created using PHP and MYSQL is used as the required database. All the books issued to teachers and students is separately maintained with duration and fine details. Books with same author and book name are stored with different IDs.

Entity Relational Diagrammatic modeling is used to create the basic design and flow of the application and once it is done the actual coding is done. Initially all the required web pages are created along with the database tables. The required business logic is added to improve the scope of the application and the database connections are established. Few test cases are created and manual testing is used in this project. Test cases are created in a manner to check all the required field level validations and also database validations. Once the application is created the corresponding test cases are executed against couple of runs and the bugs are resolved if any identified. Once the testing is done, the final product is delivered.

**Timeline and Scheduling:**

**Schedule**

We have divided our project work in eight parts. In the following figure we have represent the time period of each part.

**Activity one:** Provide the Entity- Relationship diagram of our database.

**Activity two:** Create the relational model of our database.

**Figure: Gantt chart for project schedule.**

**Activity three:** Create a database in MySQL.

**Activity four:** Implement the relational model create for our project.

**Activity five**: Add data to the database table.

**Activity six:** Provide a list of simple queries to show that our database is consistent with our E/R and relational model.

**Activity seven:** Create and use views, stored procedures and triggers to ensure the consistency of our Database.

**Activity eight:** Use PHP to create our GUI and access the database.

**Hardware and Software requirements**

**Hardware Requirements**

Processor : Pentium IV 2GHz and Above

RAM : 2GB RAM

Monitor : 15” Color Monitor

**Software Requirements**

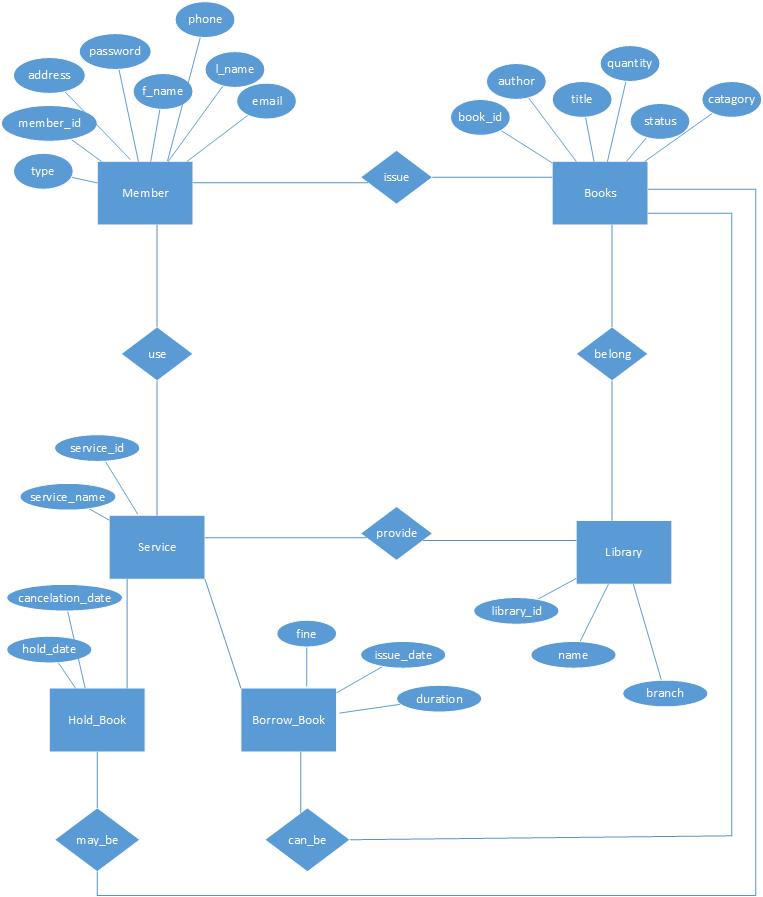
Operating System. : Windows, Linux or Mac

Developing Tool : PHP, HTML, XAMPP

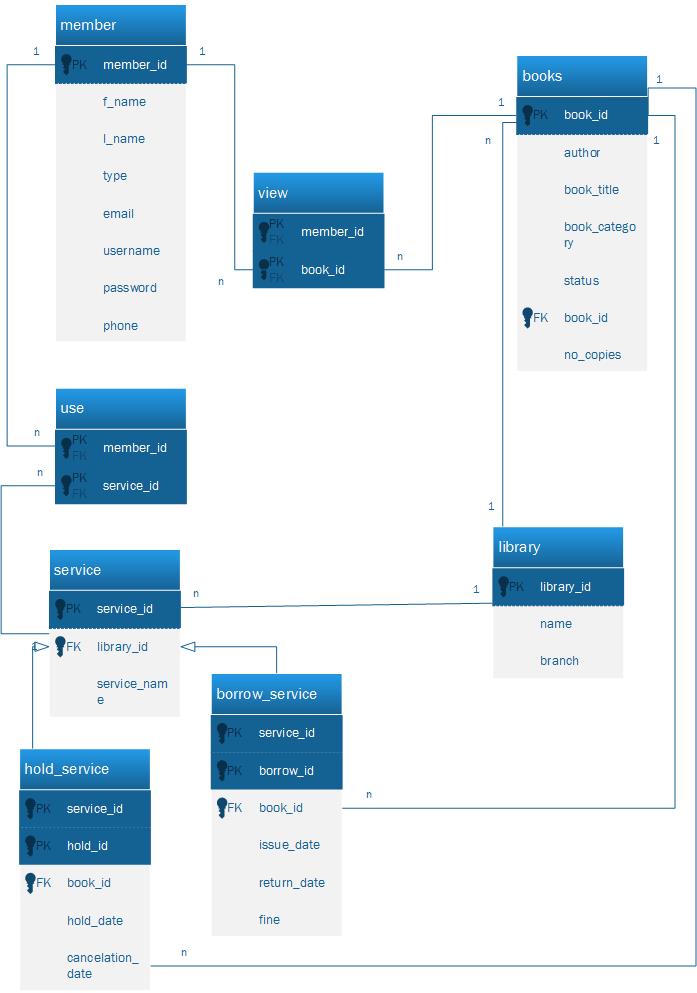
Database : MYSQL

Server : Apache

**Entity-Relationship Diagram for the project:**

****

**Figure: Entity-Relationship Diagram in generalized form.**

****

**Figure: More Detailed Entity/Relationship Diagram with attributes.**

**Sample queries used in the project:**

**Entity sets:**

·         Books: book\_id,book\_title,book\_category,no\_copies,status,author

·         Member: member\_id,email,password,username,type,f\_name,l\_name,phone

·         Service: service\_id,service\_name

·         Hold\_service: cancelation\_date,hold\_date,hold\_id

·         Borrow\_service: issue\_date,fine,return\_date,borrow\_id

·         Library: library\_id, name,branch.

**Relationships:**

•          view: (Books, Member)  (n:m)

•          use: (Member, Service)  (n:m)

•          has (Library, Service)(1:m)

•          belongs\_to(Library: Books) (1:m)

•          get\_involved\_in (Books, Service)(0:m)

•          participate\_in(Books, Service) (0:m)

----------------------------------------------

Query for borrow a book.

----------------------------------------------

Insert into Service (service\_id,service\_name,library\_id) values (1,'borrow',1)

Insert into borrow\_service(service\_id,book\_id,issue\_date,return\_date)

Values(1,‘00001’,’11-18-20140’,’11-27-2014’);

      Insert into use ( member\_id,service\_id)( 1,1);

----------------------------------------------

 Query for hold a book.

----------------------------------------------

 Insert into Service (service\_id,service\_name,library\_id) values (2,'hold',1)

Insert into hold\_service(service\_id,book\_id,hold\_date,cancellation\_date)

Values(1,‘00002’,’11-18-20140’,’11-27-2014);

      Insert into used\_service ( member\_id,service\_id)( 1,2);

----------------------------------------------

Query for returning a book.

----------------------------------------------

Update borrow\_service set return\_date = ’11-28-2014’

----------------------------------------------

Query for cancelling a hold.

----------------------------------------------

Delete from service where service\_id = 2;

Delete from used\_service where service\_id = 2

Delete from hold\_service where service\_id = 2

----------------------------------------------

 Query to see user information.

----------------------------------------------

Select \* from member where member\_id = 1;

----------------------------------------------

Query to see member with id 1 is holding any book.

----------------------------------------------

Select distinct from member m,used\_service u,hold\_service h where

m.member\_id = 1 and

m.member\_id = u.member\_id and

u.service\_id = h.service\_id;

----------------------------------------------

 Query to see member with id 1 is borrowing any book.

----------------------------------------------

Select distinct from member m,used\_service u,borrow\_service b where

m.member\_id = 1 and

m.member\_id = u.member\_id and

u.service\_id = b.service\_id;

----------------------------------------------

**Use of stored procedure**

----------------------------------------------

Stored procedure to see all the books in database.

CREATE PROCEDURE ‘library\_system’.’GetAllBooks’()

BEGIN

    Select \* from books;

END

We called the procedure with this statement.

Call GetAllBooks()

----------------------------------------------

**Use of trigger**

----------------------------------------------

Create a trigger to audit the update of book table.

First we created a book\_audit table to save the data.

CREATE TABLE ‘book\_audit’(

  ‘id’ int(11)NOT NULL AUTO INCREMENT,

  `book\_id` VARCHAR(50) NOT NULL,

  `book\_title` VARCHAR(50) NOT NULL,

  `changed\_on` DATETIME NOT NULL,

  `action` VARCHAR(50) NOT NULL,

  PRIMARY KEY (`book\_id`)

)

Here we created the trigger:

DELIMITER$$

CREATE TRIGGER before\_book\_update

  BEFORE UPDATE ON books

  FOR EACH ROW BEGIN

  INSERT INTO book\_audit

  SET action= ‘update’,

  book\_id = OLD.book\_id,

  book\_title = OLD.book\_title,

  changed\_on = NOW();

END$$

DELIMITER;

----------------------------------------------

-- Inserting book data for table `book`

----------------------------------------------

INSERT INTO `books` (book\_id,book\_title,book\_category, no\_copies,status,author,library\_id) VALUES

('00001', 'The tail of Lam-ang', 'Filipino', '10','available','danbrown',1),

('00003', 'vb6 for dummies', 'programming', '10', 'available','danbrown',1),

('00002', 'php for dummies', 'programming', '10', 'available','danbrown',1),

('00004', 'reading fundamentals', 'english', '10', 'available','danbrown',1),

('00005', 'css design', 'programming', '10', 'available','danbrown',1),

('00006', 'geometry', 'math', '10', 'available','danbrown',1),

('00007', 'statistics', 'math', '10', 'available','danbrown',1),

('00008', 'Lam-ang', 'english', 'sample', 'available','danbrown',1),

('00009', 'css design', 'sample', '20', 'available','danbrown',1);

------------------------------------------------------

-- Inserting data for table `member`

------------------------------------------------------

INSERT INTO `member` (f\_name,l\_name,email,phone,type, username,password) VALUES

('layla','reza','layla.reza@yahoo.com','408999989','student','laylareza', 'password1'),

('geetha','anne','geetha.anne@yahoo.com','408978769','student','geethanne', 'password2'),

('tom','jerry','tom.jerry@yahoo.com','412998749','staff','tomjerry', 'password3'),

('admin','admin','admin@yahoo.com','41834749','staff','administrator', 'password4');

-----------------------------------------------------------

Inserting data for table `library`

------------------------------------------------------------

INSERT INTO `library` (library\_id,name,branch) VALUES

(1, 'King Library', 'San Jose');

INSERT INTO `library` (library\_id,name,branch) VALUES

(2, 'MLK1', 'San Jose');

INSERT INTO `library` (library\_id,name,branch) VALUES

(3, 'MLK2, 'Santa Clara');

-----------------------------------------------------------

Deleting book data for table `books`

-------------------------------------------------------

DELETE from ‘books’ where book\_id= ‘0001’;

DELETE from ‘books’ where book\_id= ‘00023’ and author =’danbrown’;

-----------------------------------------------------------

Deleting member data for table `members`

-------------------------------------------------------

DELETE from ‘member’ where member.lname= ‘anne’;

DELETE from ‘member’ where member\_id= ‘00001’;

-----------------------------------------------------------

Deleting data for table `library`

-------------------------------------------------------

DELETE from ‘library’ where library.name= ‘MLK’;

DELETE from ‘library’ where branch= “Santa Clara”;

-----------------------------------------------------------

Updating data for table `library`

-------------------------------------------------------

UPDATE ‘library’ SET library.name= ‘MLK’ where library\_id=3;

UPDATE ‘library’ SET library.branch= ‘Sunnyvale’ where library\_id=2;

-----------------------------------------------------------

Updating data for table `books`

-------------------------------------------------------

UPDATE ‘books’ SET boooks.author= ‘John gReesham’ where library\_id=0001;

UPDATE ‘books’ SET book\_title = ‘Beauty Quest’ where book\_id=’00023’;

-----------------------------------------------------------

Updating data for table `member`

-------------------------------------------------------

UPDATE ‘member’ SET f\_name= ‘Geetharani’ where member\_id=0001;

UPDATE ‘member’ SET email = ‘kaka.keku@gmail.com’ where f\_name=’anne’;

-----------------------------------------------------------

Creating a VIEW for the books table

-------------------------------------------------------

CREATE VIEW [book\_by\_brown] AS  
SELECT book\_id,Book\_title from books where author= ‘danbrown’;

CREATE VIEW [book name contains letter T] AS  
SELECT book\_id,Book\_title from books where book\_title is %W%;

-----------------------------------------------------------

Creating a VIEW for the member table

-------------------------------------------------------

CREATE VIEW [member lname] AS  
SELECT \* from member where member.l\_name is %ar% group by member\_id;

CREATE VIEW [member email] AS  
SELECT \* from member where member.email is %yahoo% group by member\_id;

1. SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='TRADITIONAL,ALLOW\_INVALID\_DATES';

2. DROP SCHEMA IF EXISTS `library\_system` ;

3. CREATE SCHEMA IF NOT EXISTS `library\_system` DEFAULT CHARACTER SET utf8 ;

USE `library\_system` ;

4. -- -----------------------------------------------------

-- Table `library\_system`.`books`

-- ---------------------------------------------------------

CREATE TABLE IF NOT EXISTS `library\_system`.`books` (

`book\_id` VARCHAR(50) NOT NULL,

`book\_title` VARCHAR(50) NOT NULL,

`book\_category` VARCHAR(50) NOT NULL,

`no\_copies` VARCHAR(50) NOT NULL,

`status` VARCHAR(50) NOT NULL,

`author` VARCHAR(60) NOT NULL,

`library\_id` INT(11) NOT NULL,

PRIMARY KEY (`book\_id`),

CONSTRAINT `fk\_books\_library\_id`

FOREIGN KEY (`library\_id`)

REFERENCES `library\_system`.`library` (`library\_id`)

ON DELETE CASCADE

ON UPDATE NO ACTION

)

ENGINE = InnoDB;

5. -- -----------------------------------------------------

-- Table `library\_system`.`view`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `library\_system`.`view` (

`book\_id` VARCHAR(50) NOT NULL,

`member\_id` INT(11) NOT NULL,

PRIMARY KEY (`book\_id`,`member\_id` ),

CONSTRAINT `fk\_view\_book\_id`

FOREIGN KEY (`book\_id`)

REFERENCES `library\_system`.`books` (`book\_id`)

ON DELETE CASCADE

ON UPDATE NO ACTION,

CONSTRAINT `fk\_view\_member\_id`

FOREIGN KEY (`member\_id`)

REFERENCES `library\_system`.`member` (`member\_id`)

ON DELETE CASCADE

ON UPDATE NO ACTION

)

ENGINE = InnoDB;

6. -- -----------------------------------------------------

-- Table `library\_system`.`member`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `library\_system`.`member` (

`member\_id` INT(11) NOT NULL AUTO\_INCREMENT,

`email` VARCHAR(50) NOT NULL,

`password` VARCHAR(50) NOT NULL,

`username` varchar(50) NOT NULL,

`type` VARCHAR(45) NULL,

`f\_name` VARCHAR(45) NULL,

`l\_name` VARCHAR(45) NULL,

`phone` VARCHAR(45) NULL,

PRIMARY KEY (`member\_id`))

ENGINE = InnoDB

AUTO\_INCREMENT = 3;

7. -- -----------------------------------------------------

-- Table `library\_system`.`use`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `library\_system`.`use` (

`member\_id` INT(11) NOT NULL,

`service\_id` INT(11) NOT NULL,

PRIMARY KEY (`service\_id`,`member\_id` ),

CONSTRAINT `fk\_use\_service\_id`

FOREIGN KEY (`service\_id`)

REFERENCES `library\_system`.`service` (`service\_id`)

ON DELETE CASCADE

ON UPDATE NO ACTION,

CONSTRAINT `fk\_use\_member\_id`

FOREIGN KEY (`member\_id`)

REFERENCES `library\_system`.`member` (`member\_id`)

ON DELETE CASCADE

ON UPDATE NO ACTION

)

ENGINE = InnoDB;

8. -- -----------------------------------------------------

-- Table `library\_system`.`service`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `library\_system`.`service` (

`service\_id` INT(11) NOT NULL AUTO\_INCREMENT,

`service\_name` VARCHAR(50) NOT NULL,

`library\_id` INT(11) NOT NULL,

PRIMARY KEY (`service\_id`),

CONSTRAINT `fk\_service\_library\_id`

FOREIGN KEY (`library\_id`)

REFERENCES `library\_system`.`library` (`library\_id`)

ON DELETE CASCADE

ON UPDATE NO ACTION

)

ENGINE = InnoDB

AUTO\_INCREMENT = 4;

9. -- -----------------------------------------------------

-- Table `library\_system`.`borrow\_service`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `library\_system`.`borrow\_service` (

`issue\_date` DATETIME NULL,

`fine` INT NULL,

`return\_date` DATETIME,

`service\_id` INT(11) NOT NULL,

`borrow\_id` INT(11) NOT NULL,

`book\_id` VARCHAR(50) NOT NULL,

PRIMARY KEY (`service\_id`, `borrow\_id`),

CONSTRAINT `fk\_borrow\_service\_book\_id`

FOREIGN KEY (`book\_id`)

REFERENCES `library\_system`.`books` (`book\_id`)

ON DELETE CASCADE

ON UPDATE NO ACTION,

CONSTRAINT `fk\_borrow\_service\_service\_id`

FOREIGN KEY (`service\_id`)

REFERENCES `library\_system`.`service` (`service\_id`)

ON DELETE CASCADE

ON UPDATE NO ACTION)

ENGINE = InnoDB;

10. -- -----------------------------------------------------

-- Table `library\_system`.`hold\_service`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `library\_system`.`hold\_service` (

`cancelation\_date` DATETIME NULL,

`hold\_date` DATETIME NULL,

`service\_id` INT(11) NOT NULL,

`hold\_id` INT(11) NOT NULL,

`book\_id` VARCHAR(50) NOT NULL,

PRIMARY KEY (`service\_id`, `hold\_id`),

CONSTRAINT `fk\_hold\_service\_book\_id`

FOREIGN KEY (`book\_id`)

REFERENCES `library\_system`.`books` (`book\_id`)

ON DELETE CASCADE

ON UPDATE NO ACTION,

CONSTRAINT `fk\_hold\_service\_service\_id`

FOREIGN KEY (`service\_id`)

REFERENCES `library\_system`.`service` (`service\_id`)

ON DELETE CASCADE

ON UPDATE NO ACTION)

ENGINE = InnoDB;

11. -- -----------------------------------------------------

-- Table `library\_system`.`library`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `library\_system`.`library` (

`library\_id` INT(11) NOT NULL,

`name` VARCHAR(50) NULL,

`branch` VARCHAR(50) NULL,

PRIMARY KEY (`library\_id`))

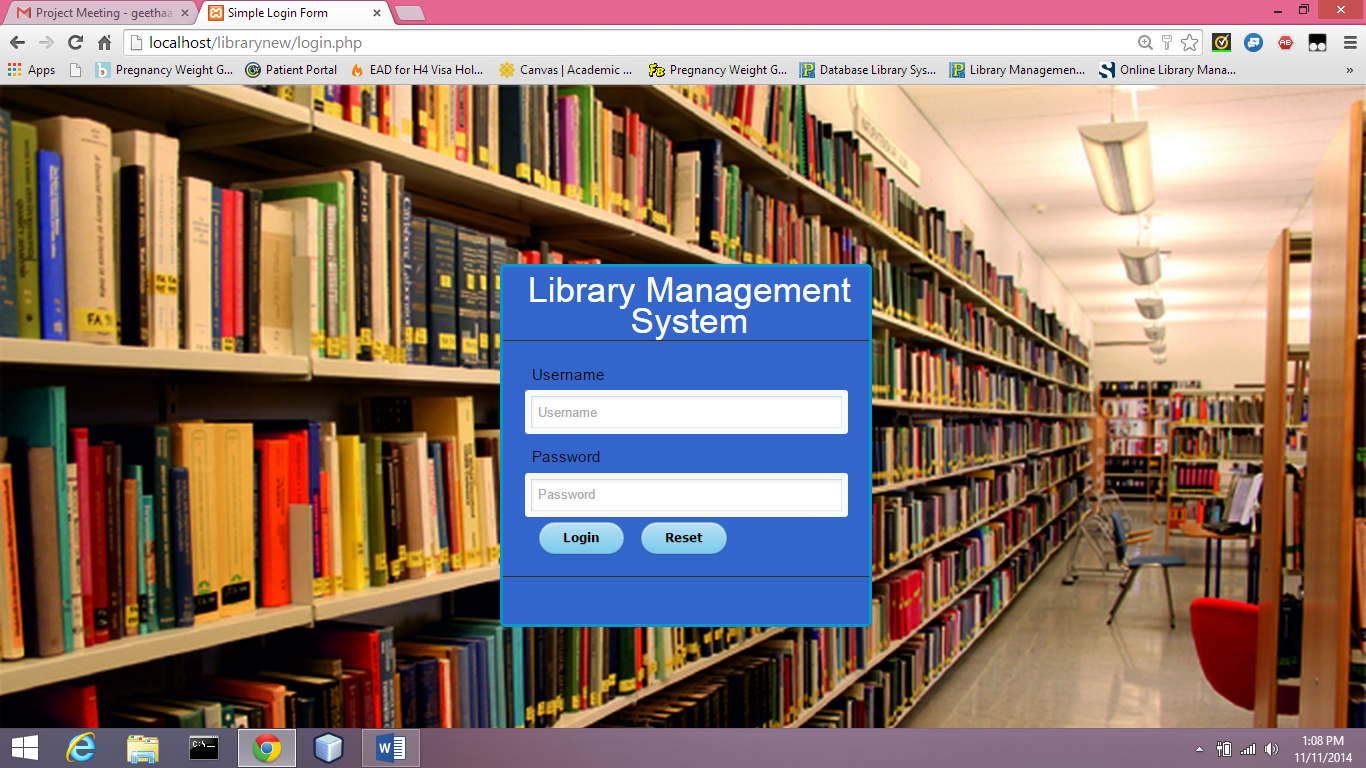
ENGINE = InnoDB

AUTO\_INCREMENT = 8;

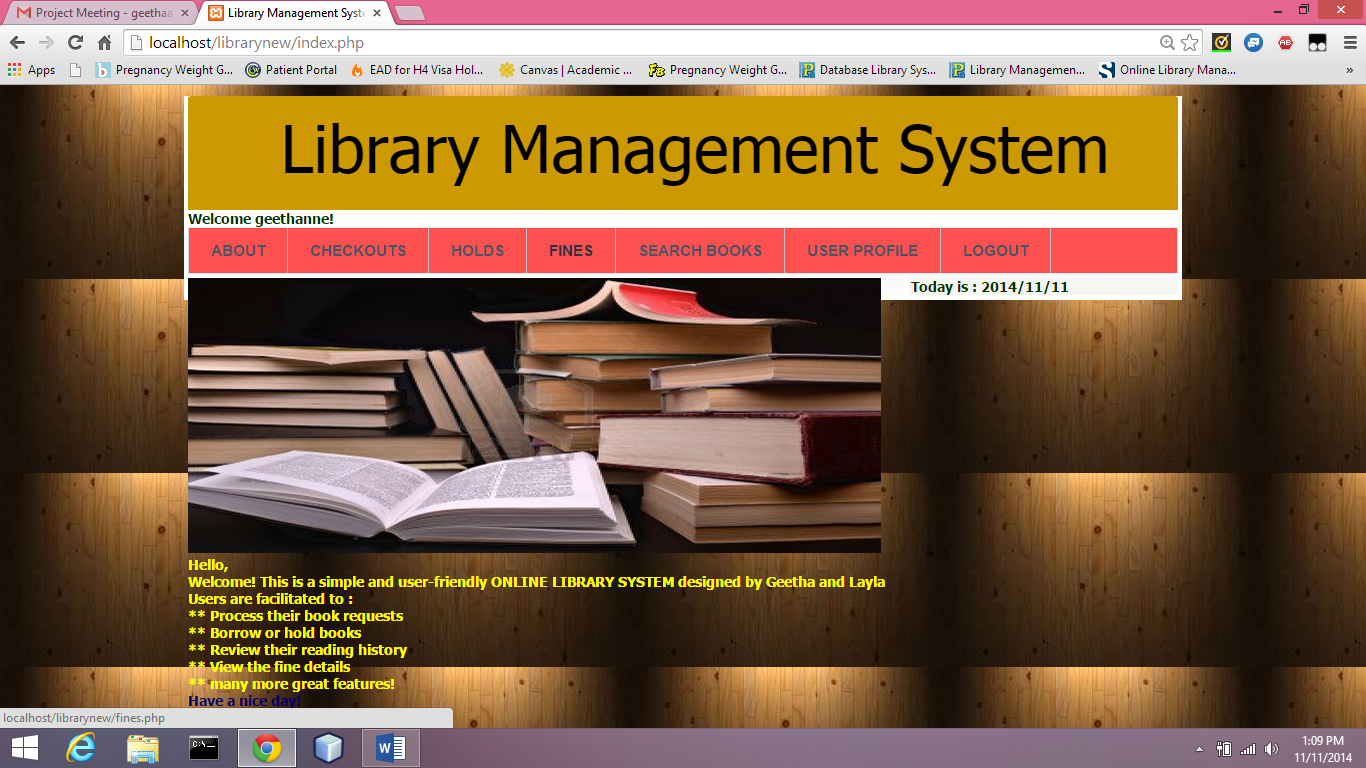
12. Select DISTINCT book\_category from books ORDER BY book\_category ASC.

13. Select \* from books where book\_category like '%$categ%'

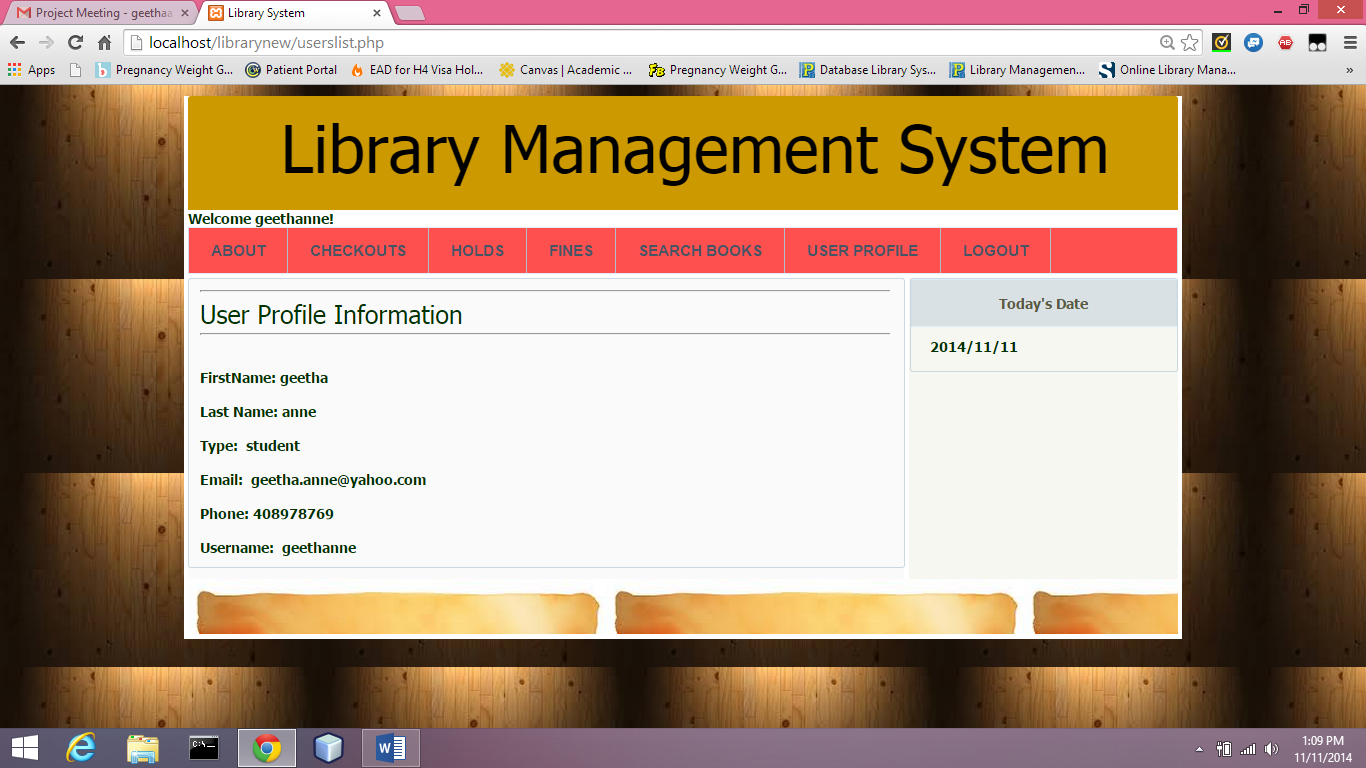
**Screen Shots of Web layout of project:**



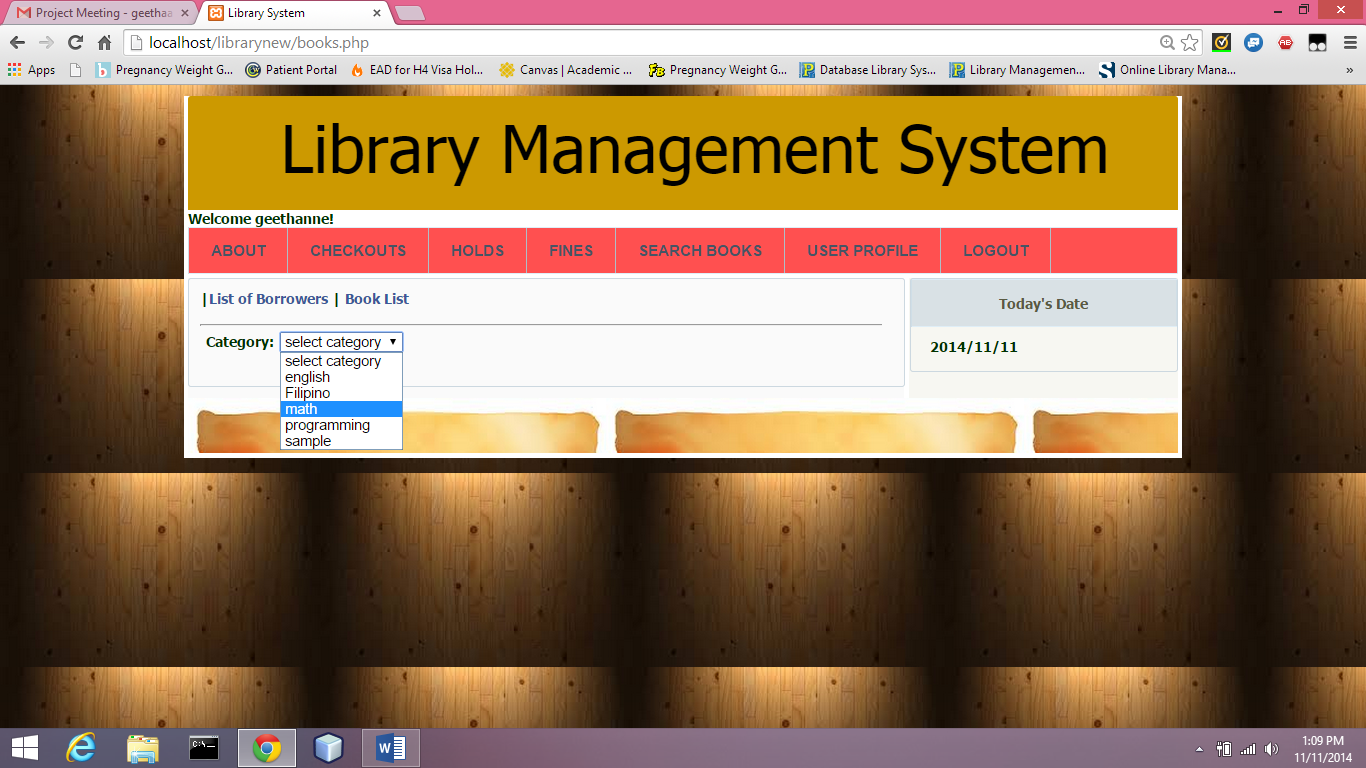
**Figure: Login page of the library management system website**



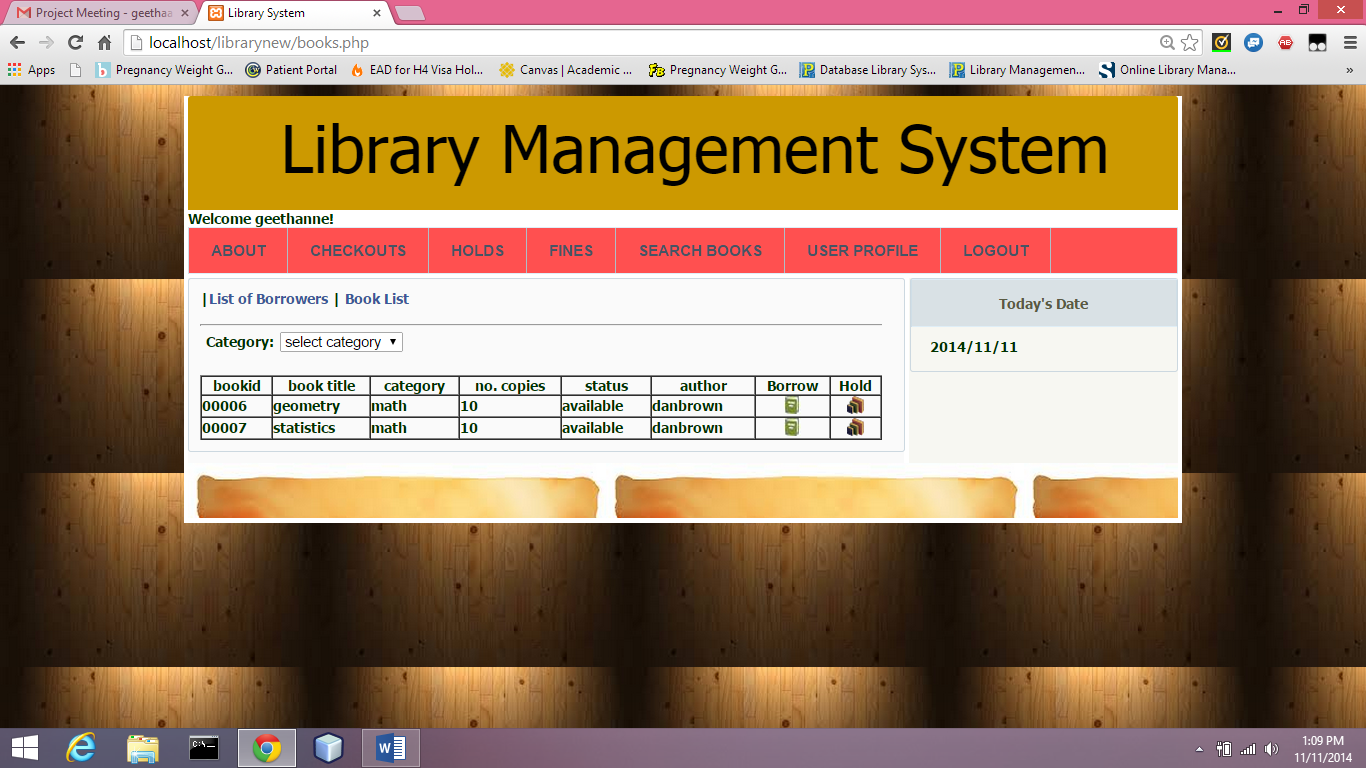
**Figure: About/ home page of the library website.**



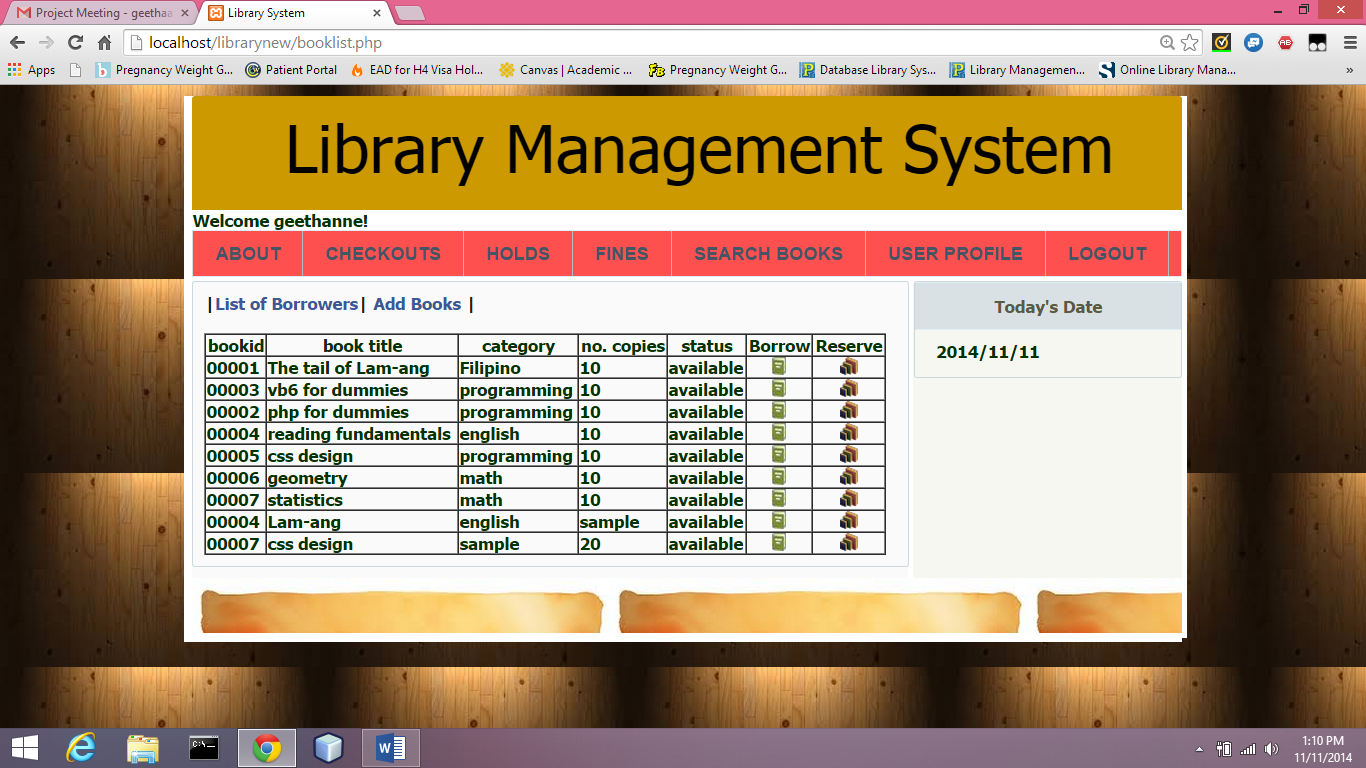
**Figure: User Profile Information page**

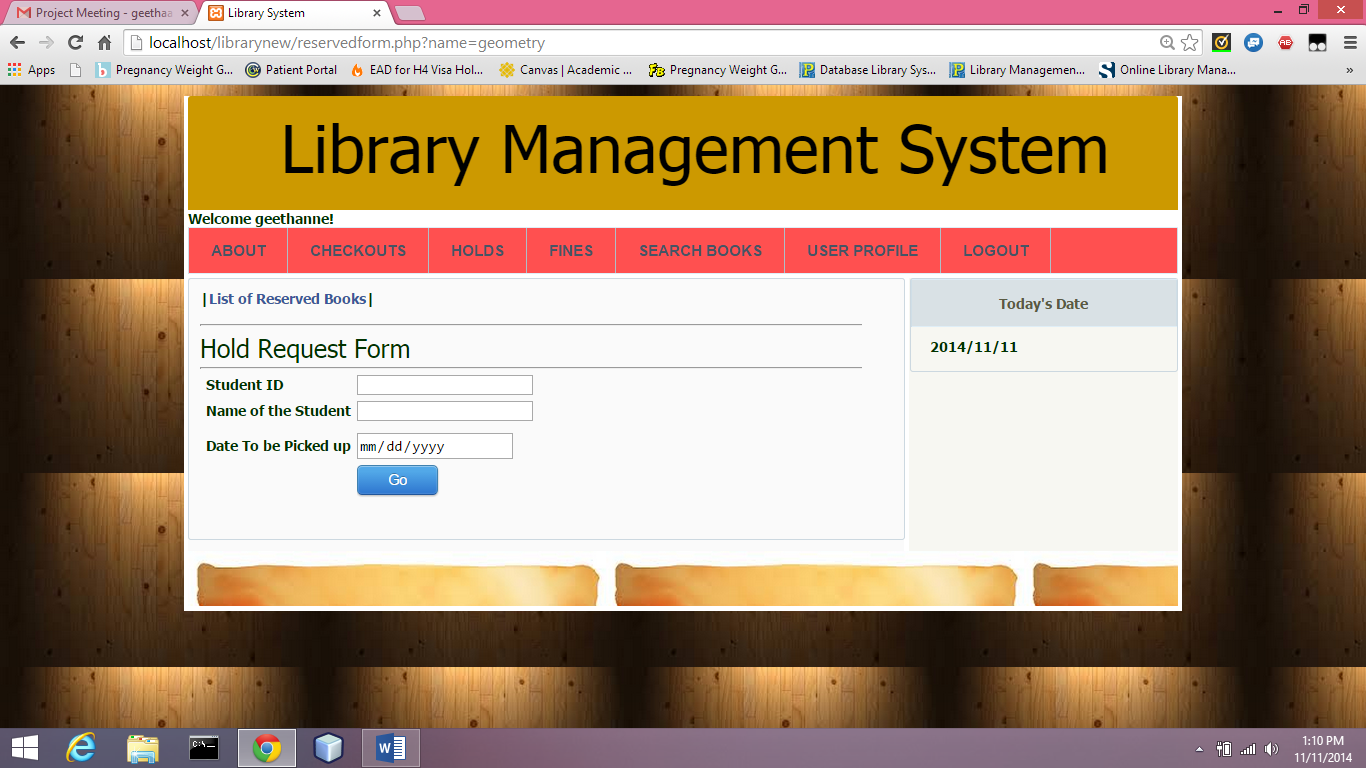


**Figure: Search book page based on category**

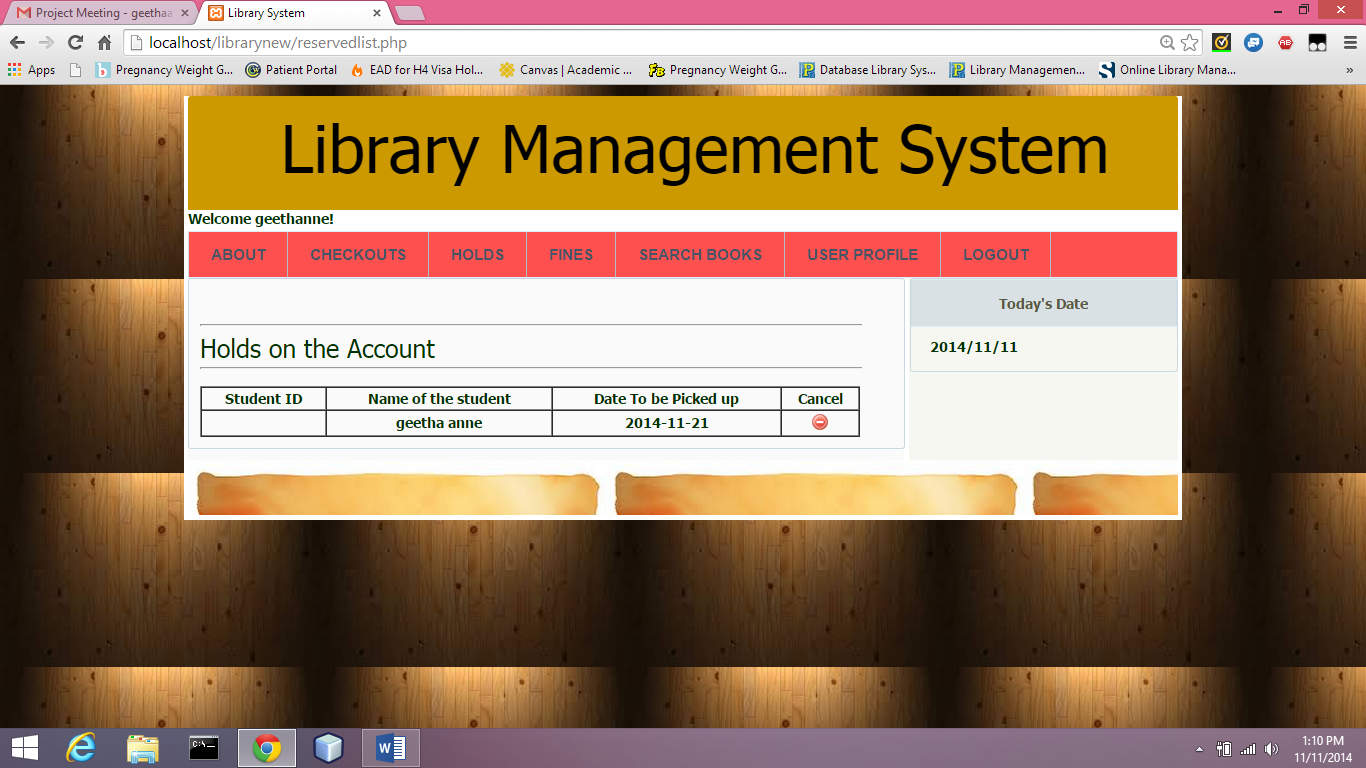


**Figure: Search result as per the required category of book.**

**Figure: List of books that are available being displayed**



**Figure: Hold request Form**



**Figure: Holds page showing the books that are on hold by the user.**

**Major findings:**

1. MYSQL database has similar query processing as of a SQL server.
2. MYSQL server can be run along with the Apache server and PHP from a control panel called XAMPP.
3. CSS creation and modification in PHP is way easier and simpler compared to most other web technologies.
4. In the library-system database that we created, it is made sure that, each book with same name and same author will have different IDs, fine details, issue records are clearly maintained.

**Analysis:**

**Lessons Learnt and Future recommendations:**

During the course of designing this database project we have learnt the following important points:

1. Creating a perfect and appropriate E/R Diagram will make it easy for the developers to dive into database creation directly.
2. PHP can used as a powerful web tool for creating interactive front end.
3. We have learnt the concepts like views, trigger, stored procedures etc.
4. We have also learnt the complete step-by-step process of SQL server installation.

As a future enhancement, this project can include large volumes of data in the tables with many more users and make it live and usable for an organization.

**Conclusion:**

The proposed **“LIBRARY MANAGEMENT SYSTEM”** designed by us, is a computerized process to reduce human errors and to increase the efficiency. The main focus of this project is to lessen human efforts. The maintenance of the records is made efficient, as all the records are stored in the MYSQL database, through which data can be retrieved easily. The navigation control is provided in all the forms to navigate through the large amount of records. PHP is a great web development platform that helped us develop beautiful and interactive user interface.

**References:**

Ullman D. J., Widom J., Molina G. H. (June 15, 2008). Database systems: The complete book. New York: Prentice Hall.

Naramore E., Gerner J., Scouarnec L. Y., Skoltz J., Glass K. M. (January 1, 2004). Beginning PHP5, Apache and MYSQL Web development. Wiley Publishing.

Nixon R. (June, 2014). Learning PHP, MYSQL, JavaScript, CSS and HTML: A step by step guide to creating dynamic websites. New York: Oreilly.