**Library Management System**

**Project Prerequisites:**

You have to build a library management system using Django where an admin can

perform CRUD ( create, read, update and delete) operations like

● Add a Book

● Update an entry of a book

● Delete a book

● Get all books

1. Admin Signup:

a) Insert admin details in tables.

b) An admin records should be unique based on email, which means

duplicates records should not be able to enter in DB.

2. Admin Login:

a) The Admin can log in based on email and password.

3. Create an entry for Books.

a) The admin can create a new entry of a book.

4. Retrieve all the books.

a) Retrieve all the books.

5. Update a book.

a) The book records should be able to update.

6. Delete a book.

a) The book record should be deleted

Student View:

1. View all the records of book

**Frontend Technology Used:**

* HTML
* CSS
* Bootstrap

**Backend Technology Used:**

* Django

### Project Features:

#### For admin

* Admin can add new books
* View the whole list of books and delete any book
* View the whole list of students and delete any student
* Issue a book to a student
* View all the issued books

#### For students

* A student can see his/her profile
* Edit their profile
* Can change password
* View the issued book by them

**Project File Structure:**

Django follows the MVT design pattern (Model View Template).

**Model :**

The data you want to present, usually data from a database.The models created are written inside the **Model.py** file

**Models.py**

**Book model:**

It saves all the data of each book added to the system.

**Student model:**

It saves all the details of each student registered to the system.

**IssuedBook model:**

It saves all the details of any book issued by a student.

After creating models in models.py file you need to run the migration command

**python manage.py makemigrations**

then the next command is to migrate

**python manage.py migrate**

after execution of this command run the server

**python manage.py runserver**

**View :**

A request handler that returns the relevant template and content - based on the request from the user.

**Urls.py**

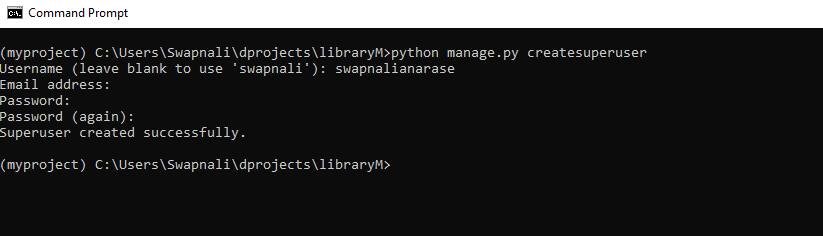
It saves all the details of any book issued by a student.

Here in this file all urls of this project are present.

**Admin:**

To login as admin you will first need to create a superuser. If your server is running off your server by ctrl+c and then run below command to create the superuser

**Python manage.py createsuperuser**

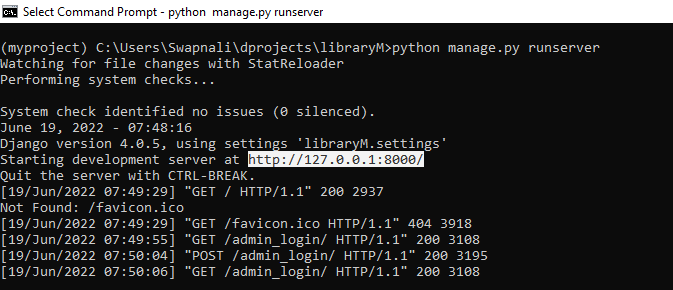
****

here you can create the username as per your choice and then enter the password. Password will not be visible but it is written.

Then after creating the user again run the server by command

**python manage.py runserver**

Now you are all ready to see the management system on your browser. Just need to copy the http link from the command prompt (cmd) and paste in the browser.



**Template Folder:**

In templates folder all the html files are written. A text file (like an HTML file) containing the layout of the web page, with logic on how to display the data.

1.Home Page (index.html):

On the homepage of the library management project, there are various options like student registration, student login and admin login. Students can register through register as student option. Then they can also login by clicking on student login. Admin can login through admin login.

2.Add Book(add\_book.html):

After the admin login, the admin can add new books or delete a book. Admin just have to give the details of the book in a form and add that book.

3.View All Books(view\_books.html):

As we have saved all the details of the book inside the book model, then from the book model we fetch all the data of the books and display it in the form of a table.

4.View All Students(view\_students.html):

All the details of the registered students are displayed in the form of a table. The admin is able to see the students list.

5.Issue A Book(issue\_book.html):

Students can ask the admin whichever book they want to issue. Then the admin can issue a book to students by selecting the book and the student.

6.View All Issued Books(view\_issued\_book.html):

After issuing the books, the admin can view all the issued books by students. If the date of issue crosses 14 days then a fine of Rs 5 each day will also be shown. After returning the book the admin can delete the issued book details of the students.

7.Student Profile(profile.html):

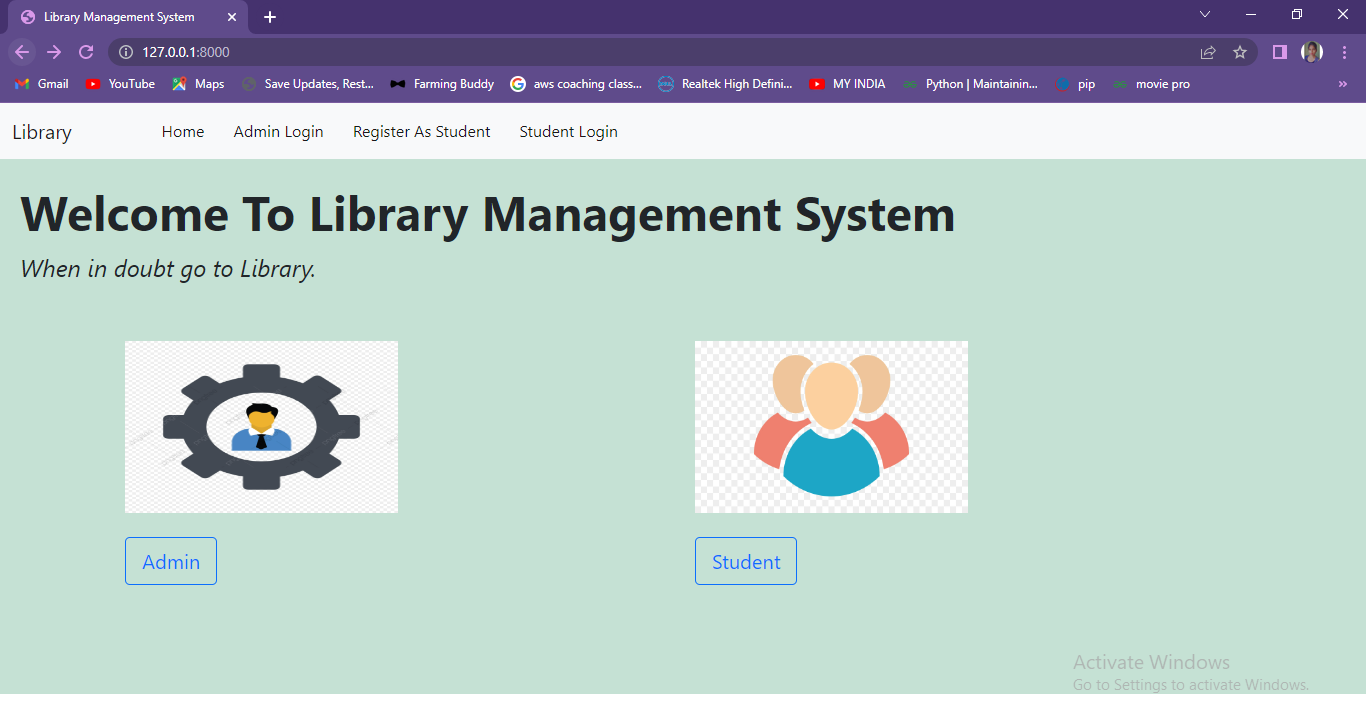
After login, students can see their profile which contains all the details that they had entered while registration.

8.Edit Profile(edit\_profile.html):

On clicking on the edit profile button on the profile page, students can edit some of the details from their profile.

**Output:**

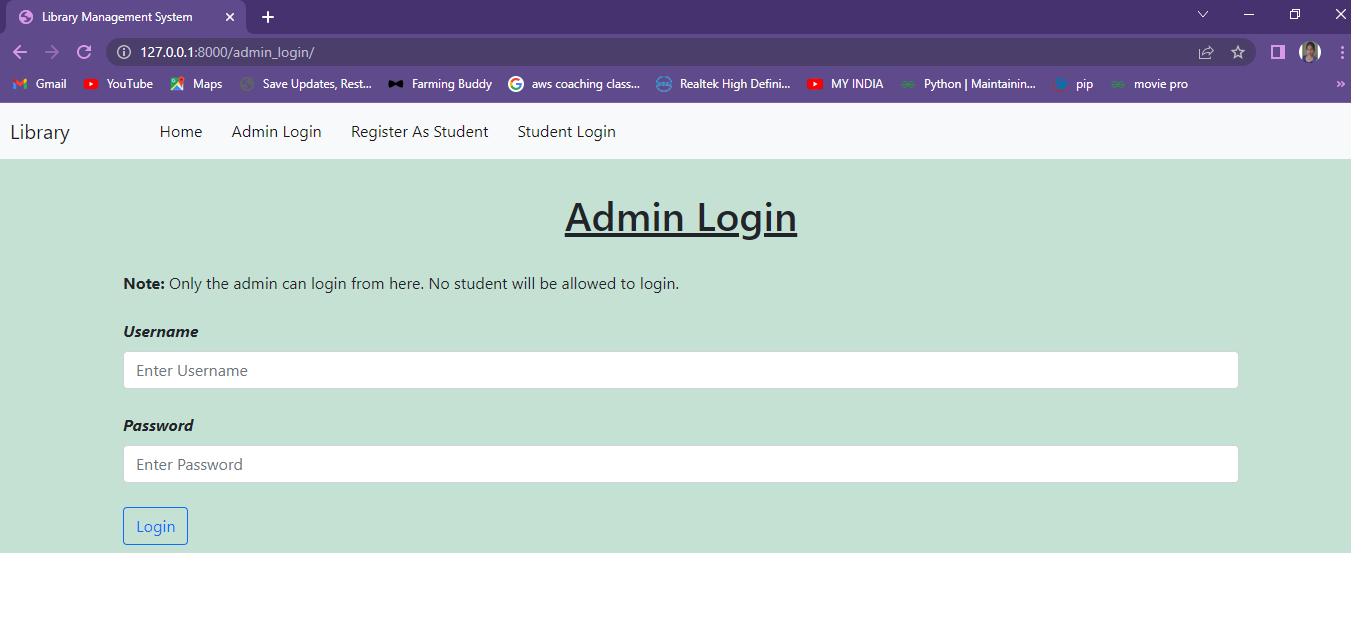
**Home Page**

****

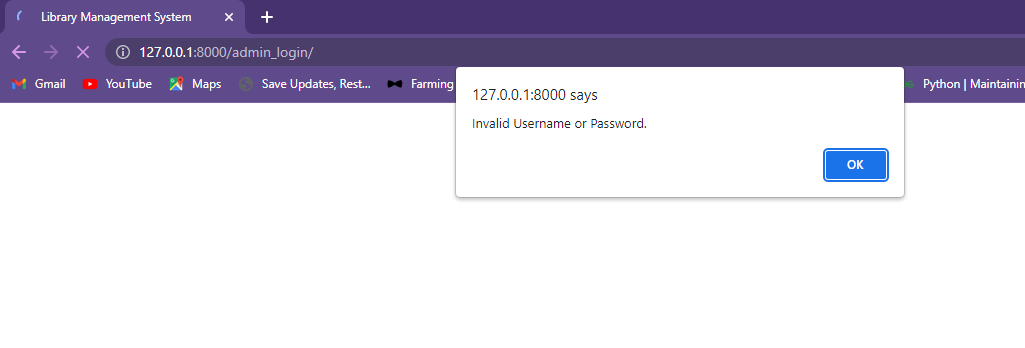
Here in home page you will get the options to login. If you Admin login on clicking on **Admin Login** tab. And if you are a user then login on clicking on **Student Login.** If you are new the first you will have to register yourself and then you can login into the system.

First will see the admin side features.

**Admin Login Page**



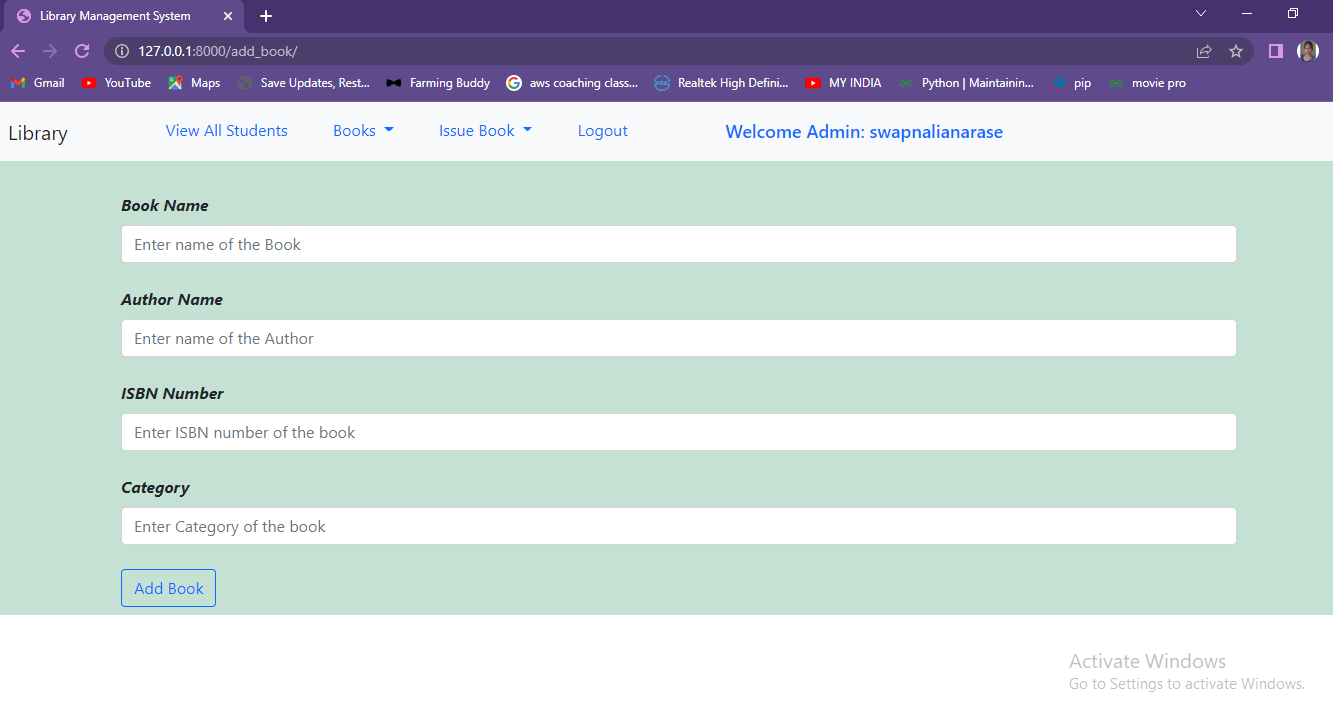
Here admin has to enter valid username and password. If admin enters invalid username or password message will pop-up.



Enter valid username and password to login as admin.

After login in successfully you will be redirected to **Add\_book** page.

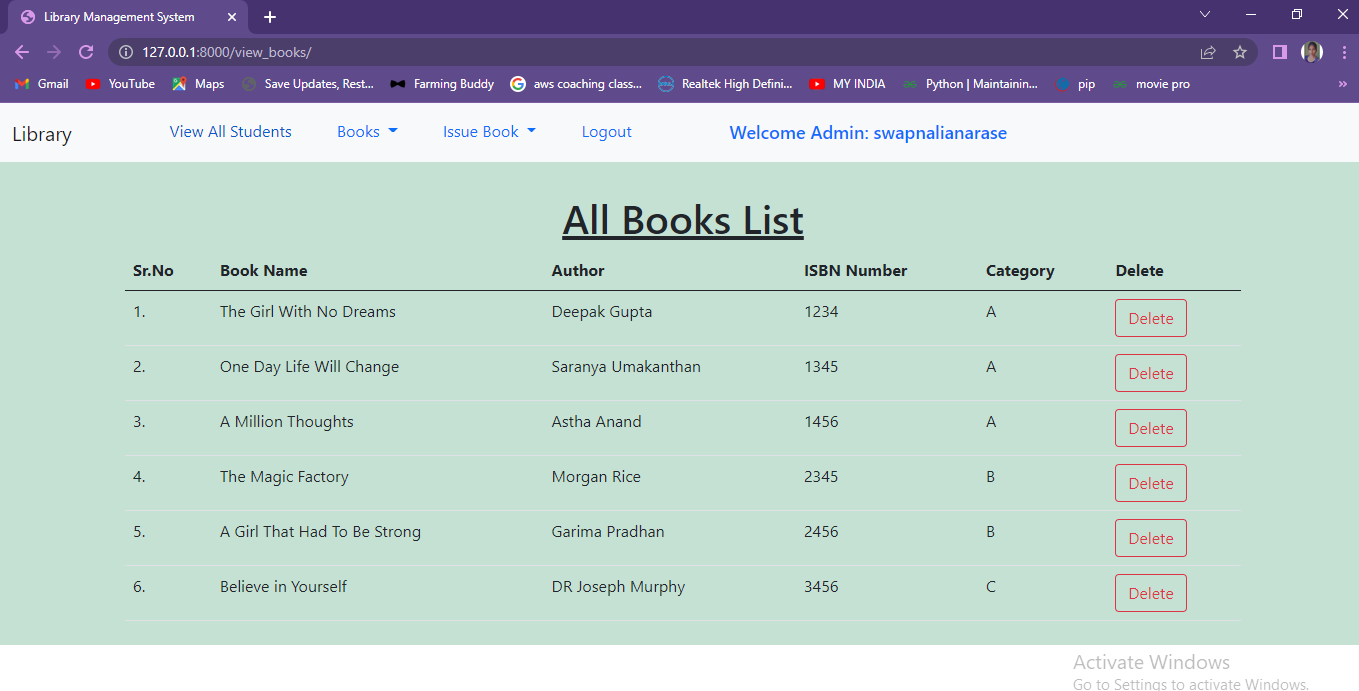
**Add Book Page**



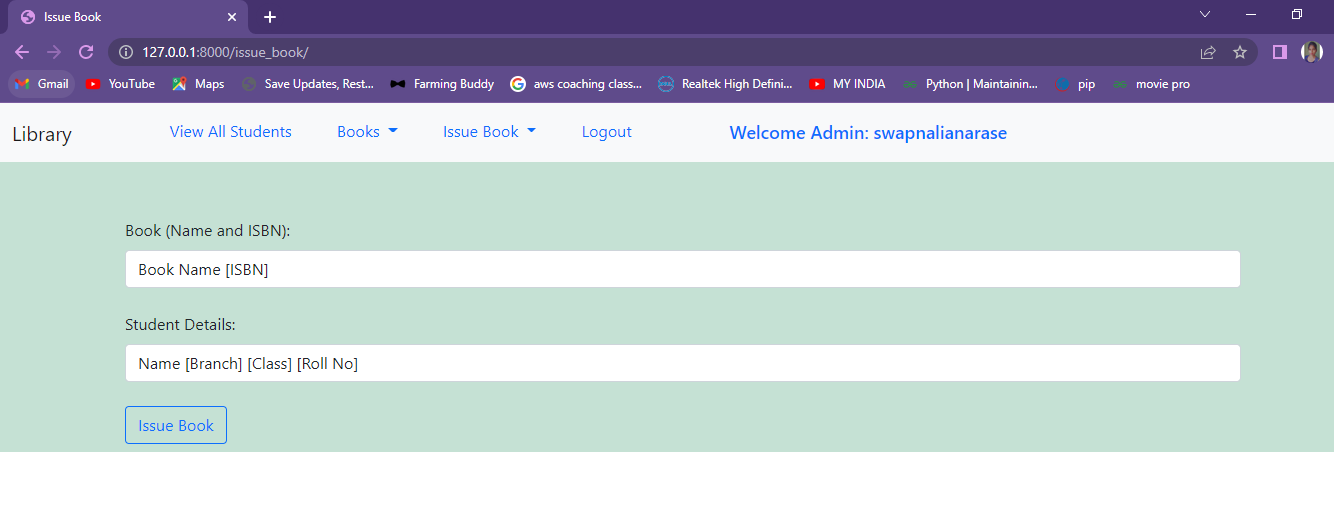
Here in add\_book page the admin can add new books to the list where he will add book name and other details.

On this page admin can view the list of all students those who has registered. All add books can also be viewed. Which book is issued to student list is also available. Then after the work is done admin can logout.

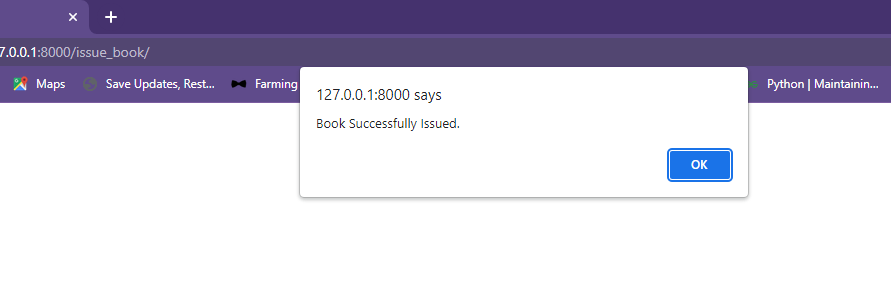
**View All Books**



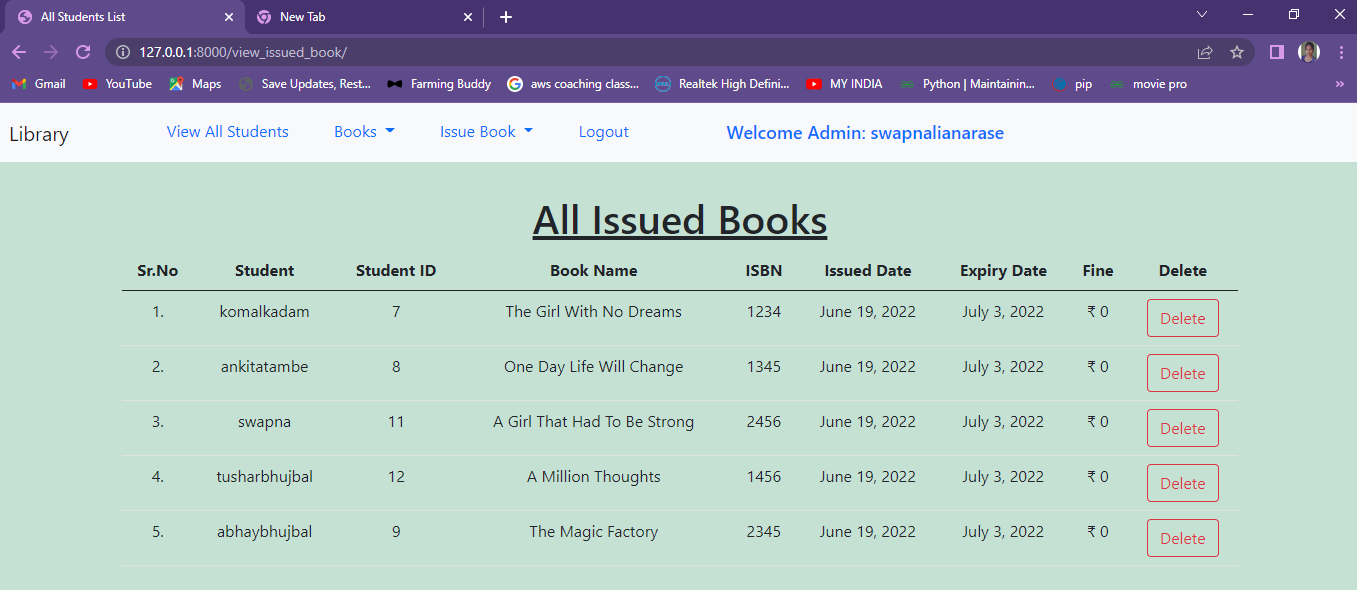
**Issue Book**



Here admin will issue a book to the student as per the requirement.

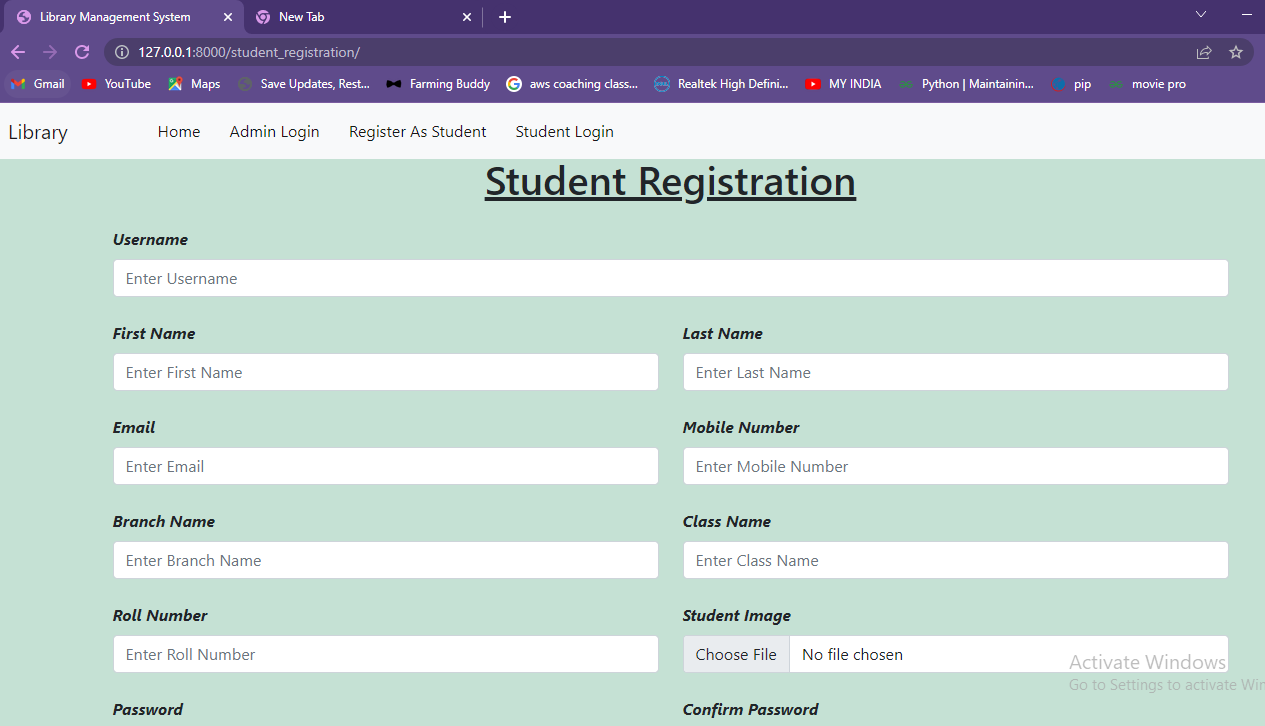


**View Issued Book Page**



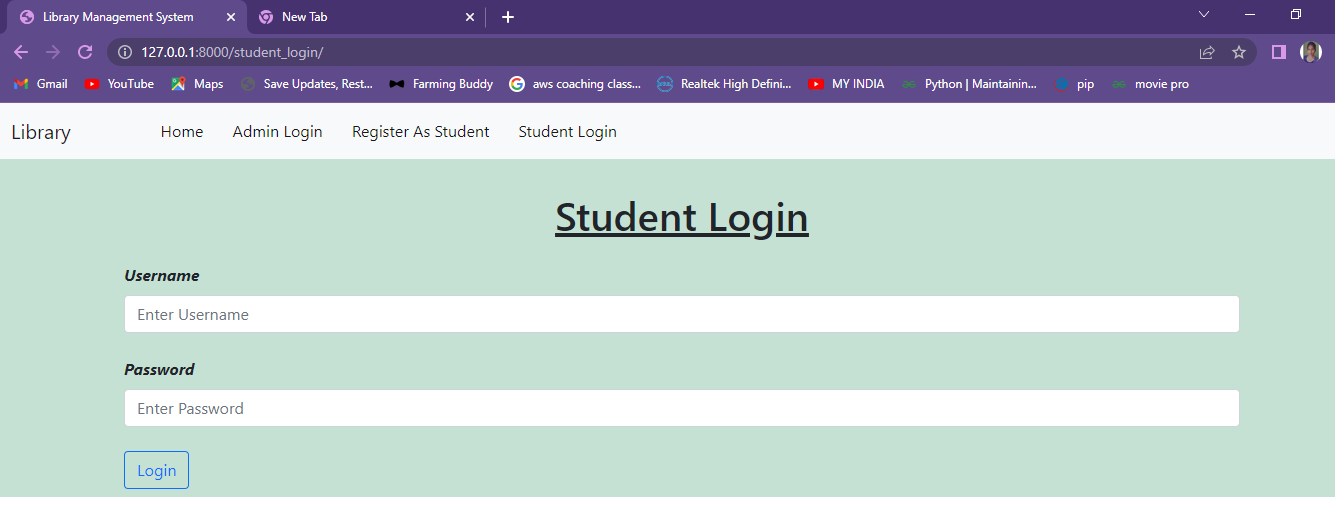
Here admin will check which book is issued at what date and to which student.

**Student Registration Page**

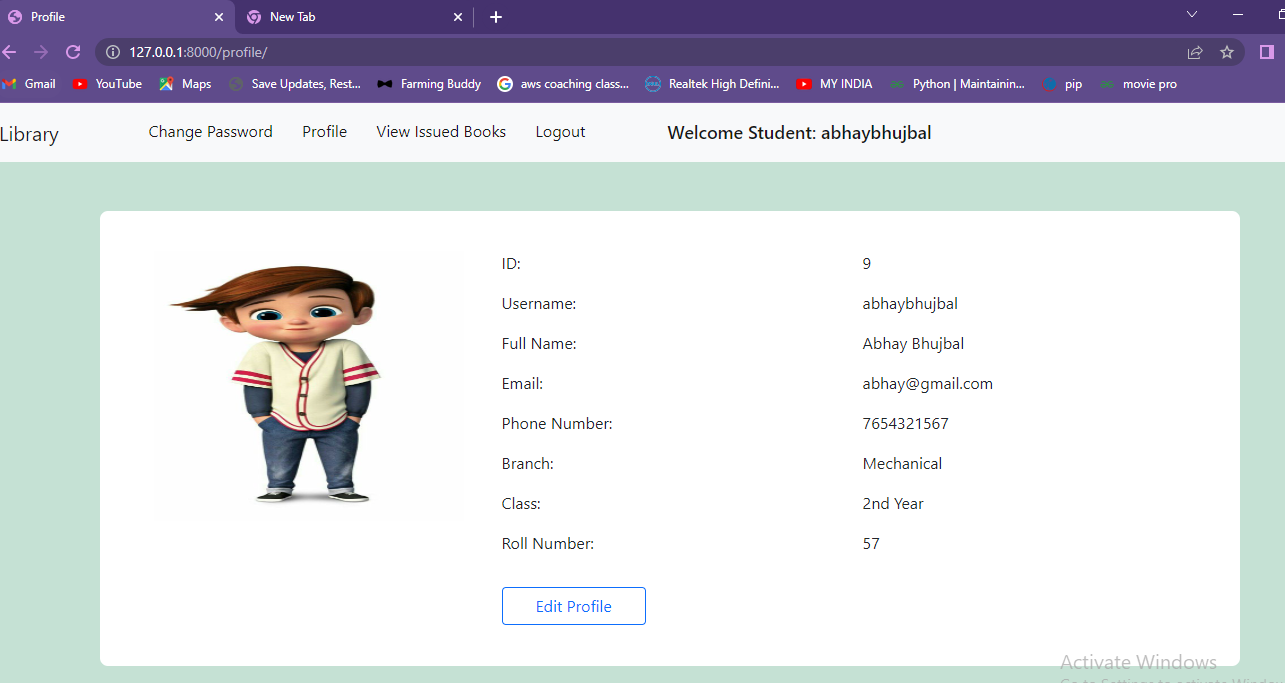
****

If user is new then he can register and get into to issue book. After registration the student will be redirected to the student login page so that he can login using username and password. Below is the student login page.

**Student Login Page**

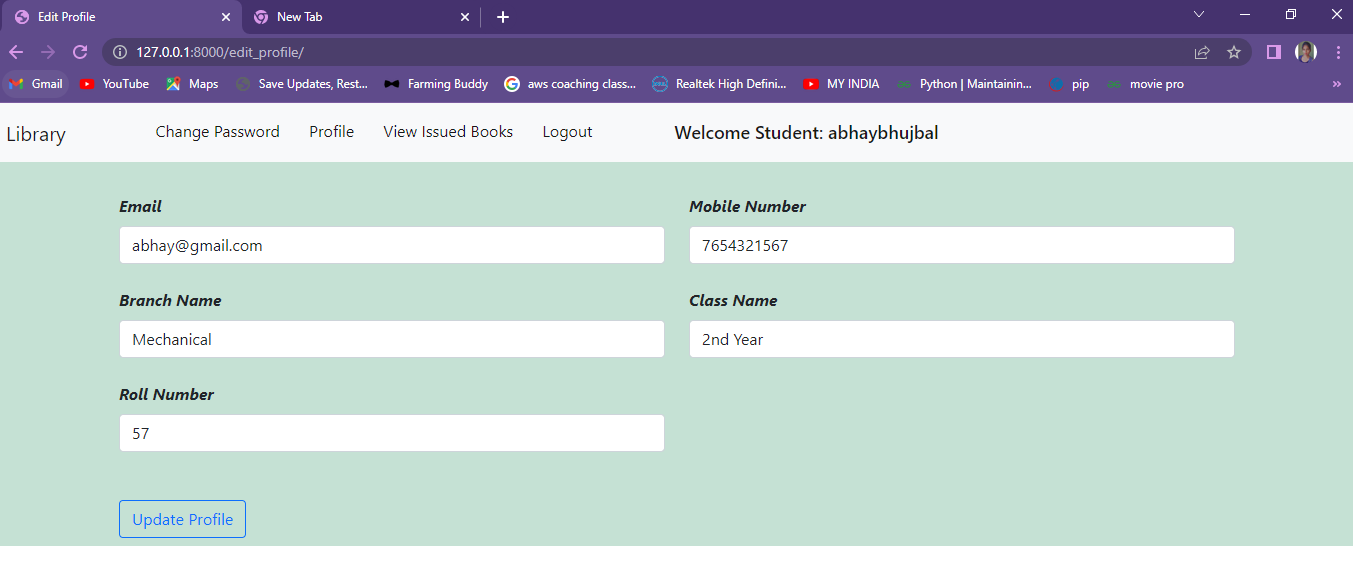
****

**Student Profile Page**

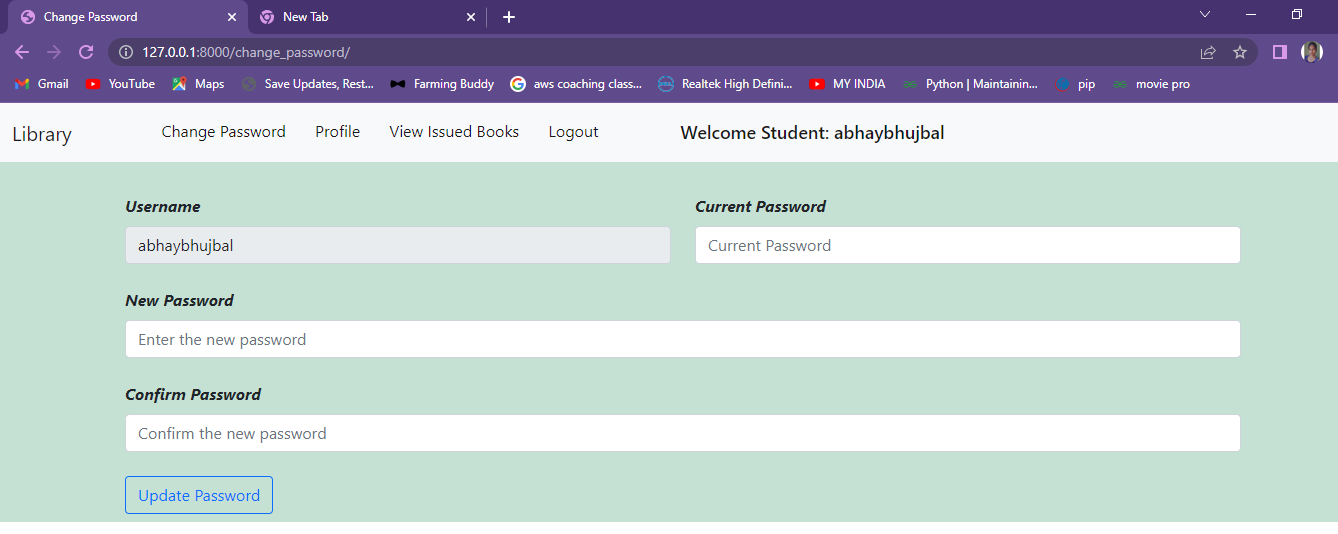


This is a student profile page where all details of student can be seen. Student can edit this profile also, password can also be changed, books issued by a student can also be viewed.

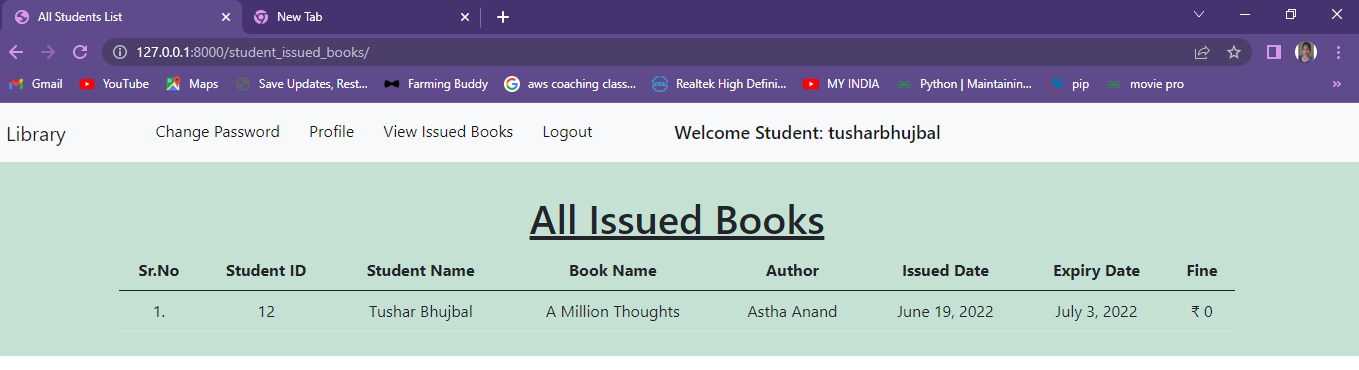
**Edit Profile Page**

****

**Change Password Page**

****

**View Issued Book**

****

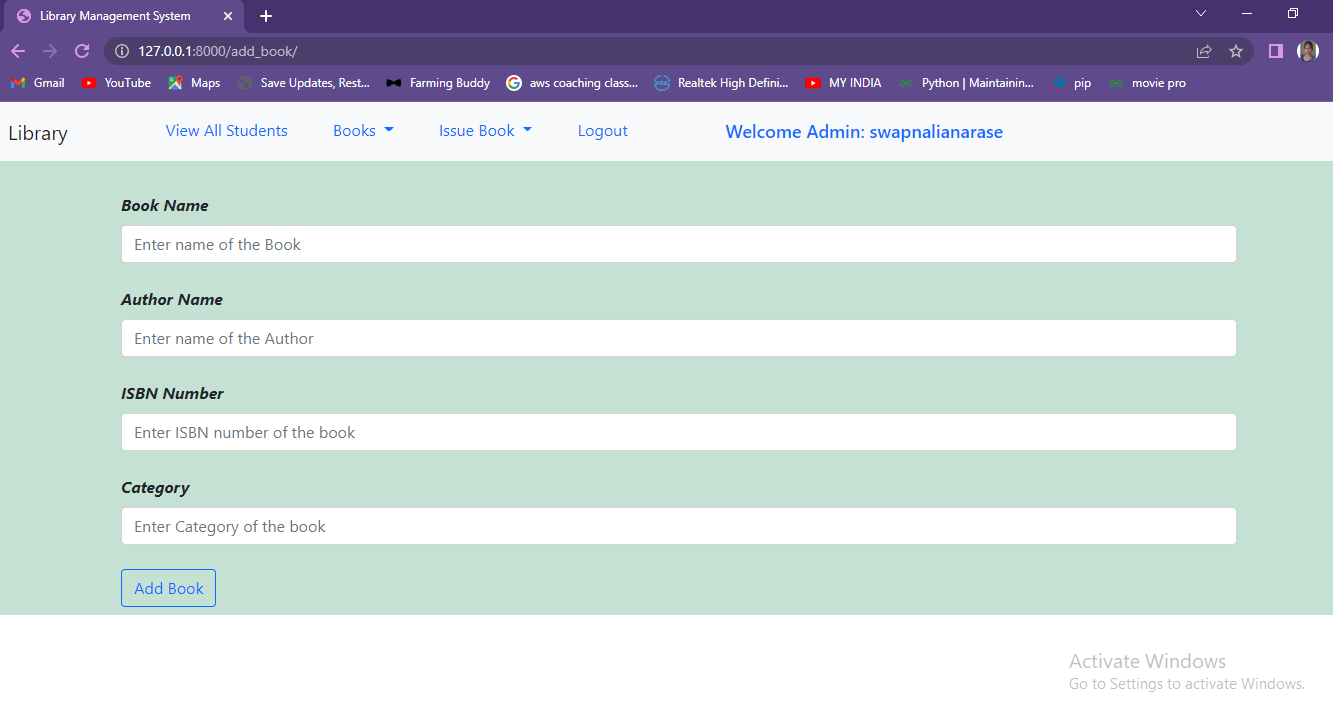
This page will show the student which book is issued by them and when he has to return the book.

All the operations are done.

Admin need to perform the **CRUD** operation.

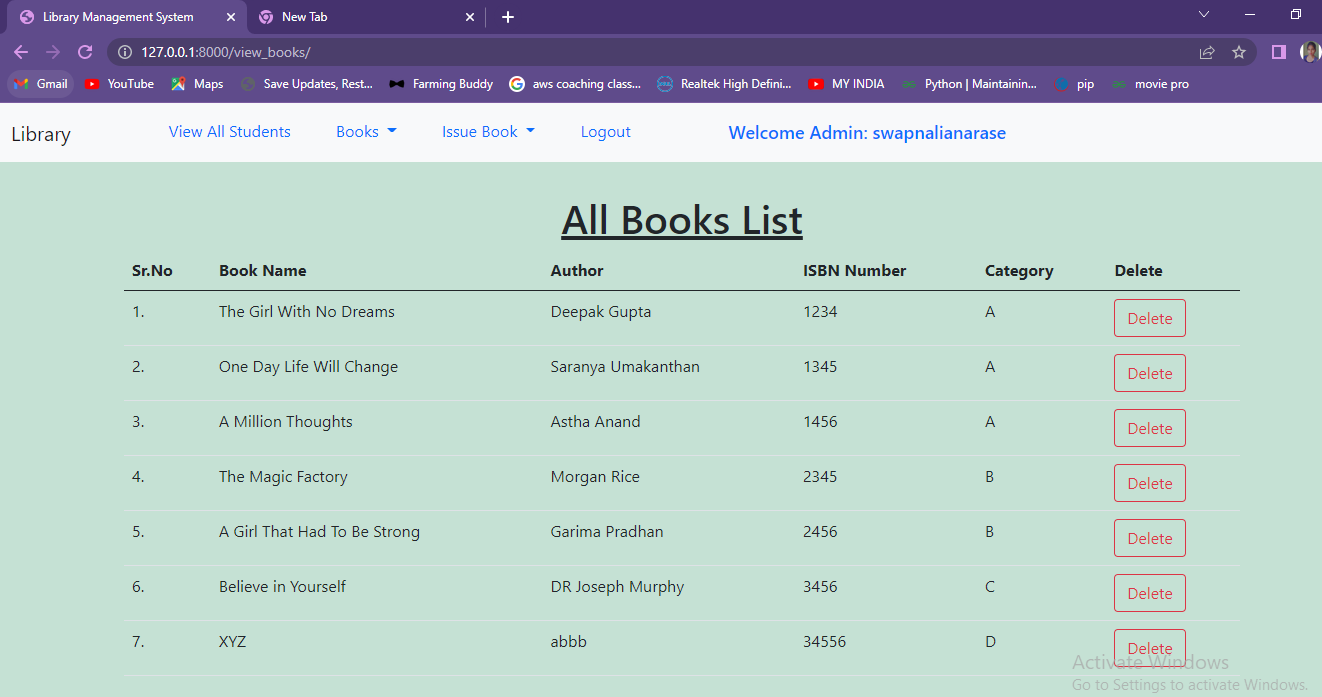
**Create (Add Book):**

Here admin is adding books to the system.

****

**Retrieve (View All Books):**

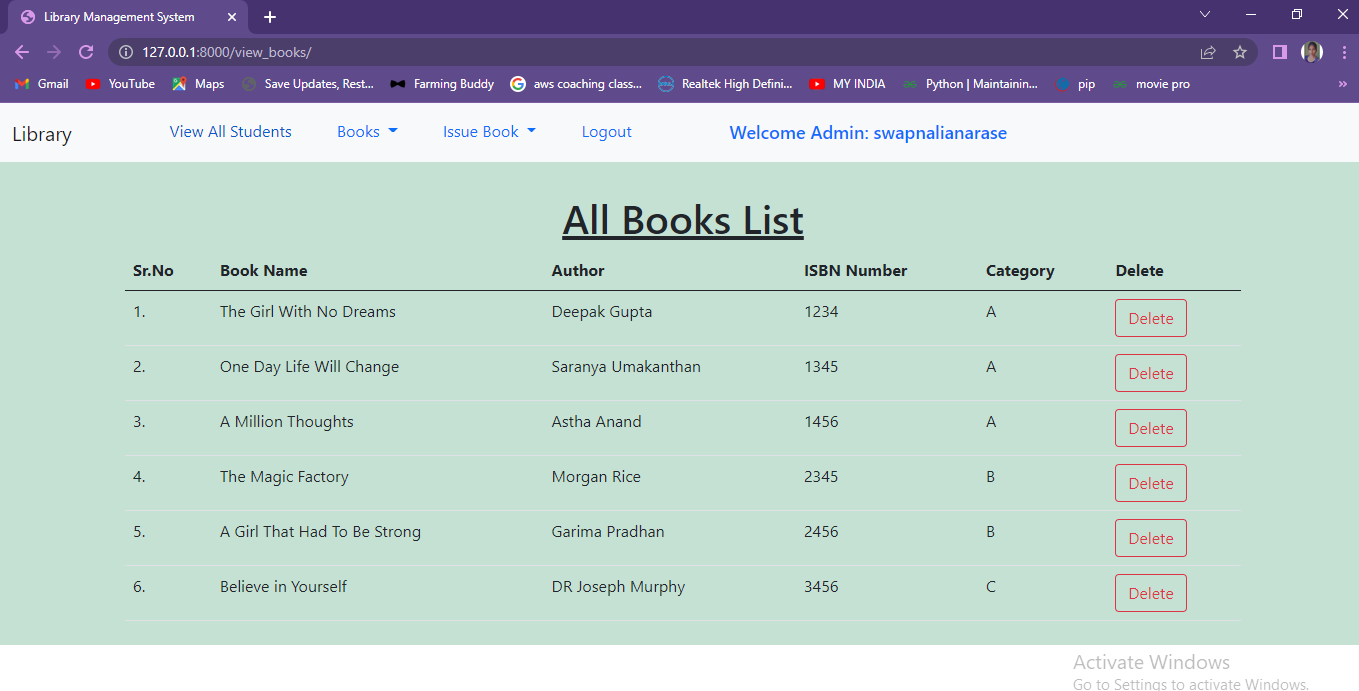
Here admin can view all the added books to the system.



**Update (Update):**

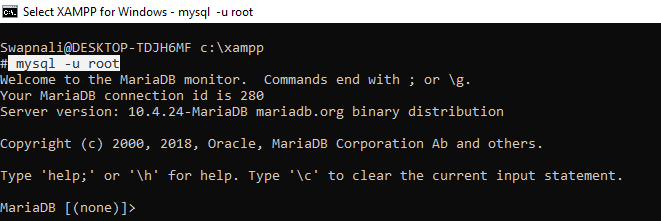
**Delete (Delete Book or Student):**

After deleting the book only 6 are remaining.



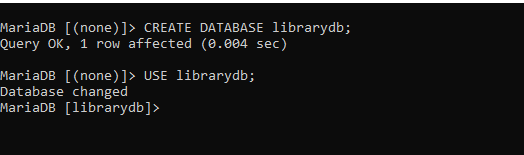
Here the data entered is stored in MySQL database.

While performing the operations you have to keep your **Xampp server** on. So that all the data will be saved inside the database.

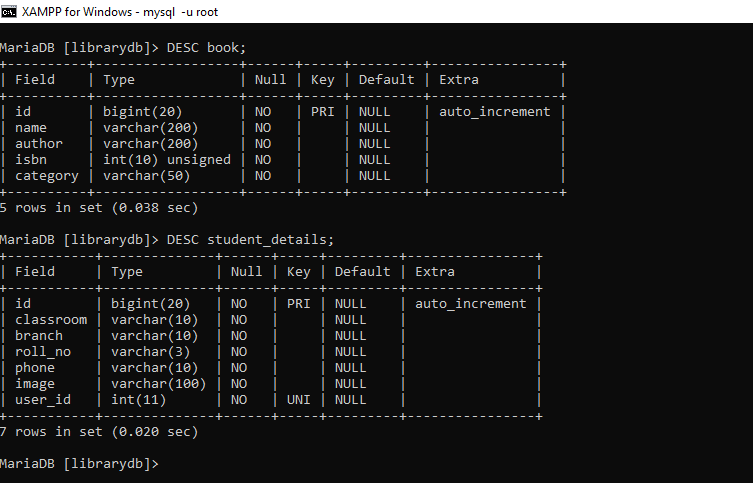


After your xampp is on enter this above command , it will connect to the MariaDB and you are ready to start your work.

Database named **librarydb** was created and linked to the system so that the entered data will be stored in the database created.

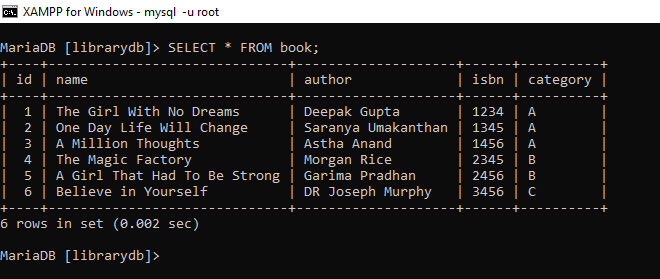


There are two tables created for **books** and **student details** in the database.

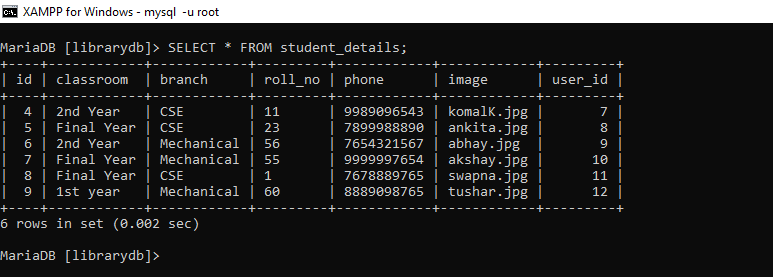


All data inserted in the system is stored in these tables.

**Books Table:**

****

**Student details Table:**

****