We have created a management system for Library **(Library Management System)** which is has been implemented using **Oracle Application Express (Apex) and SQL.** We have made 12 pages, those are:

i. Login page.
ii. Homepage (Dashboard).
iii. Admin Report page (With Add Admin form)
iv. Books Report (With Add Books Form)
v. Student Report (With Add Student form)
vi. Faculty Report (With Add Faculty form)
vii. Book Issue :
    - Book Issue Form.
    - Issued Book Table (Faculty).
    - Issued Book Table (Student).
viii. Book Submission ( With Due Fine)

- **Login Page:** It was automatically generated while creating the application in oracle apex. The login and password was same as our oracle apex login details.

- **Admin Report Page:** For this page we have created a report with form. The report type was set to interactive report and the data source is set to the table we created named **LMS_ADMIN**, selected the primary key to **ADMIN_ID**. After that a button named add student will automatically be created which will execute and enter the data into table named **LMS_ADMIN**.

- **Books Report Page:** The process of making the Books report page is exactly same as the Admin report page, but in that case we have to set the data source to **LMS_BOOK_DETAIL** and the primary key will be Book Id.

- **Add Student Report Page**: This page was manually created using the blank page template. We have added a region and set some items Student name, Gender, DOB, Course, Contact number, Department and set a button. The button will execute sql command to add the details into two separate tables as **LMS_STUDENT_DETAILS** & **LMS_BORROWER**.

```
DECLARE
    student_id number;

BEGIN

student_id := LMS_STUDENT_DETAILS_SEQ.nextval;
insert into
 LMS_STUDENT_DETAILS (STUDENT_ID,STUDENT_NAME,GENDER,DATE_OF_BIRTH,COURSE,CONTACT_NUMBER,DEPARTMENT)
values(student_id,:P6_STUDENT_NAME, :P6_GENDER, :P6_DOB, :P6_COURSE, :P6_CONTACT_NO, :P6_DEPARTMENT );

insert into LMS_BORROWER (BORROWER_ID,ENTITY_OWNER_FK,ENTITY_TYPE)
values(LMS_BORROWER_SEQ.nextval, student_id, 'STUDENT');
END;
```

- **Faculty Report Page**: Similarly as add student page, we have to insert the data into two separate tables **LMS_STAFF_DETAIL** & **LMS_BORROWER**.

```
DECLARE
    student_id number;

BEGIN

student_id := LMS_STUDENT_DETAILS_SEQ.nextval;
insert into LMS_STUDENT_DETAILS (STUDENT_ID,STUDENT_NAME,GENDER,DATE_OF_BIRTH,COURSE,CONTACT_NUMBER,DEPARTMENT)
values(student_id,:P6_STUDENT_NAME, :P6_GENDER, :P6_DOB, :P6_COURSE, :P6_CONTACT_NO, :P6_DEPARTMENT );

insert into LMS_BORROWER (BORROWER_ID,ENTITY_OWNER_FK,ENTITY_TYPE)
values(LMS_BORROWER_SEQ.nextval, student_id, 'STUDENT');
END;
```

- **Book Issue**: This page is designed to show the book issue form and two reports to show the issued books with the details of the borrower. To issue the books, we have set a sql query**:**

```
Declare
user_id number(10);

begin
    if(:P10_USER_TYPE = 'FACULTY') then
        user_id := :P10_BORROWER_FACULTY;
          insert into
 LMS_BOOK_ISSUE(BOOK_ISSUE_ID,BOOK_ID,BORROWER_ID,ISSUED_BY,BORROWED_FROM_DATE,BORROWED_TO_DATE,BORROWER_TYP
E,BOOK_RT_STATUS)
    values(LMS_BOOK_ISSUE_SEQ.nextval,:P10_BOOK_TITLE,user_id, 1, :P10_BOROWED_DATE,:P10_RETURN_DATE,
'FACULTY',1 );

    update LMS_BOOK_DETAIL set BOOK_NO_OF_COPIES_CURRENT = BOOK_NO_OF_COPIES_CURRENT-1 where
 BOOK_ID=:P10_BOOK_TITLE;

    else
        user_id := :P10_STUDENT;
        insert into
 LMS_BOOK_ISSUE(BOOK_ISSUE_ID,BOOK_ID,BORROWER_ID,ISSUED_BY,BORROWED_FROM_DATE,BORROWED_TO_DATE,BORROWER_TYP
E,BOOK_RT_STATUS)
    values(LMS_BOOK_ISSUE_SEQ.nextval,:P10_BOOK_TITLE,user_id, 1, :P10_BOROWED_DATE,:P10_RETURN_DATE,
'STUDENT',1 );

    update LMS_BOOK_DETAIL set BOOK_NO_OF_COPIES_CURRENT = BOOK_NO_OF_COPIES_CURRENT-1 where
 BOOK_ID=:P10_BOOK_TITLE;

    end if;



end;
```

This sql query insert the value of borrower into **LMS_BOOK_ISSUE** table and updates the **LMS_BOOK_DETAIL** as the number of copies of the book reduces if a user borrows the book.

Then in order to show the issued book details we use the following queries:

```
select LBI.BOOK_ISSUE_ID,
(select LBD.BOOK_TITLE from LMS_BOOK_detail LBD where LBD.BOOK_ID=LBI.BOOK_ID) as Book_Title,
(select STAFF_NAME from LMS_STAFF_DETAIL
where STAFF_ID= (select LB.ENTITY_OWNER_FK from LMS_BORROWER LB
WHERE LB.BORROWER_ID=LBI.BORROWER_ID)) as Staff_Name,
LBI.ISSUED_BY,LBI.BORROWED_FROM_DATE,LBI.BORROWED_TO_DATE
from LMS_BOOK_ISSUE LBI where LBI.BORROWER_TYPE= 'FACULTY' and BOOK_RT_STATUS =1
```

```
select LBI.BOOK_ISSUE_ID,
(select STUDENT_NAME from LMS_STUDENT_DETAILS where
STUDENT_ID= (select LB.ENTITY_OWNER_FK from LMS_BORROWER LB WHERE LB.BORROWER_ID=LBI.BORROWER_ID)) as
 STUDENT_Name,
(select LBD.BOOK_TITLE from LMS_BOOK_detail LBD where LBD.BOOK_ID=LBI.BOOK_ID) as Book_Title,

LBI.ISSUED_BY,LBI.BORROWED_FROM_DATE,LBI.BORROWED_TO_DATE
from LMS_BOOK_ISSUE LBI where LBI.BORROWER_TYPE= 'STUDENT' and BOOK_RT_STATUS =1
```

Those queries works the same way but the difference is one shows the staff detail, as another one shows the student details, the condition is set that if the **BOOK_RT_STATUS** is 1, it will show the information because if the RT status is 1 that means the person has borrowed the books. It will also show the information of the book which is matched by the book title of the issue table.

- **Book Submission / Return:** It's exactly opposite as borrowing the books, we have set some items in a region for the user to input the book details; we have set an option to input the book name and if the borrower submitting is Faculty or Student. According to our choice the return option will appear by the following sql query :

```
select LBI.BOOK_ISSUE_ID||'---'||
(select STAFF_NAME from LMS_STAFF_DETAIL where STAFF_ID =
(select LB.ENTITY_OWNER_FK from LMS_BORROWER LB WHERE LB.BORROWER_ID=LBI.BORROWER_ID)) as display_value,
LBI.BOOK_ISSUE_ID as return_value from LMS_BOOK_ISSUE LBI where LBI.BORROWER_TYPE='FACULTY' and BOOK_RT_STATUS =1
```

```
select LBI.BOOK_ISSUE_ID||'---'||
(select STUDENT_NAME from LMS_STUDENT_DETAILS where STUDENT_ID =
(select LB.ENTITY_OWNER_FK from LMS_BORROWER LB WHERE LB.BORROWER_ID=LBI.BORROWER_ID)) as display_value,
LBI.BOOK_ISSUE_ID as return_value from LMS_BOOK_ISSUE LBI where LBI.BORROWER_TYPE='STUDENT' and BOOK_RT_STATUS =1
and BORROWER_TYPE='STUDENT';
```

After selecting appropriate option we have to click the submit button which was created to imply some sql queries which will calculate the fine value for every late day. Store the information is **LMS_BOOK_RETURN** and update the **LMS_BOOK_DETAILS** and increase a number of copy, and make the **BOOK_RT_STATUS** 0 thus it is no longer borrowed.

```
declare
B_ID number(7);
Bo_ID number(7);
FINE number(3);
B_IS_ID number(7);
begin

    if(:P11_BOOK_RETURN_FORM = 'FACULTY') then
                select BOOK_ID into B_ID from LMS_BOOK_ISSUE where BOOK_ISSUE_ID =  :P11_BOOK_ISSUE_ID_FAC;

                select BORROWER_ID into Bo_ID from LMS_BOOK_ISSUE where BOOK_ISSUE_ID = :P11_BOOK_ISSUE_ID_FAC;

                select 10 * (to_number(to_date (sysdate,'MM-DD-YYYY') - to_date(BORROWED_TO_DATE,'MM-DD-YYYY'))) into
 FINE from LMS_BOOK_ISSUE where BOOK_ISSUE_ID = :P11_BOOK_ISSUE_ID_FAC;
                :P11_DUE_FINE_FACULTY := FINE;
                insert into LMS_BOOK_RETURN(BOOK_RETURN_ID,BOOK_ID,BORROWER_ID,BOOK_ISSUE_ID,RETURN_DATE,FINE)
                values(LMS_BOOK_RETURN_SEQ.nextval, B_ID,Bo_ID,:P11_BOOK_ISSUE_ID_FAC,sysdate, FINE  );

                update LMS_BOOK_DETAIL set BOOK_NO_OF_COPIES_CURRENT = BOOK_NO_OF_COPIES_CURRENT+ 1 where BOOK_ID= B_ID;

                update LMS_BOOK_ISSUE set BOOK_RT_STATUS = 0 where BOOK_ISSUE_ID = :P11_BOOK_ISSUE_ID_FAC;


     else
                select BOOK_ID into B_ID from LMS_BOOK_ISSUE where BOOK_ISSUE_ID = :P11_BOOK_ISSUE_ID_STU;

                select BORROWER_ID into Bo_ID from LMS_BOOK_ISSUE where BOOK_ISSUE_ID = :P11_BOOK_ISSUE_ID_STU;

                select 10 * (to_number(to_date (sysdate,'MM-DD-YYYY') - to_date(BORROWED_TO_DATE,'MM-DD-YYYY')))  into
 FINE from LMS_BOOK_ISSUE where BOOK_ISSUE_ID = :P11_BOOK_ISSUE_ID_STU;
                :P11_DUE_FINE_STUDENT := FINE;

                 insert into LMS_BOOK_RETURN(BOOK_RETURN_ID,BOOK_ID,BORROWER_ID,BOOK_ISSUE_ID,RETURN_DATE,FINE)
                values(LMS_BOOK_RETURN_SEQ.nextval, B_ID,Bo_ID,:P11_BOOK_ISSUE_ID_STU,sysdate, FINE  );

                update LMS_BOOK_DETAIL set BOOK_NO_OF_COPIES_CURRENT = BOOK_NO_OF_COPIES_CURRENT+ 1 where BOOK_ID= B_ID;

                update LMS_BOOK_ISSUE set BOOK_RT_STATUS = 0 where BOOK_ISSUE_ID = :P11_BOOK_ISSUE_ID_STU;

     end if;
   /* insert into LMS_BOOK_RETURN(BOOK_RETURN_ID,BOOK_ID,BORROWER_ID,BOOK_ISSUE_ID,RETURN_DATE,FINE)
    values(1,1,1,1,'1-1-2000',20);*/
end;
```

- **Homepage:** The homepage was also created by default; we have added some css and added template codes to make the dashboard.

**The Codes can be accessed through my Github profile:**

**https://github.com/smahmuddz/LMS**