**EX.NO:1 DDL COMMANDS**

**Aim:**

To work with DDL commands.

**Procedure:**

The Data Definition Language (DDL) Commands are-

**CREATE:**

**-**It is used to create a new table or new database.

**i)CREATE DATABASE:**

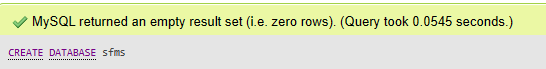
Syntax :

CREATE DATABASE database\_name;

Command :

CREATE DATABASE sfms;

Output:

****

**ii) CREATE TABLE:**

Syntax:

CREATE TABLE tablename

(column\_name1 datatype,

… …….,

column\_name\_n datatype);

Command:

CREATE TABLE `student` (

  `register` varchar(255) NOT NULL,

  `name` varchar(255) NOT NULL,

  `email` varchar(255) NOT NULL,

  `gender` varchar(100) NOT NULL,

  `updationDate` date NOT NULL,

  `password` varchar(255) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

Output:

****

**Alter:**

-The ALTER statement is used to add, delete or modify columns in an existing table.

**i) ALTER TABLE-ADD COLUMN:**

Syntax:

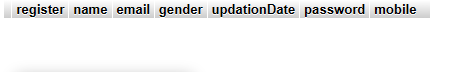
ALTER TABLE table\_name

ADD column\_name datatype;

Example:

ALTER TABLE student

ADD mobile int(11);

Output:

**ii) ALTER TABLE-DROP COLUMN:**

Syntax:

ALTER TABLE table\_name

DROP COLUMN column\_name;

Example:

ALTER TABLE student

DROP COLUMN mobile;

Output:



**iii) ALTER TABLE - RENAME COLUMN:**

Syntax:

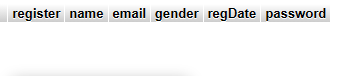
ALTER TABLE table\_name

CHANGE old\_name new\_name DATA TYPE;

Example:

[ALTER](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/alter-table.html) [TABLE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/alter-table.html)  student

CHANGE updationDate regDate  [DATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/date-and-time-types.html) [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_not) NULL;

Output:

**iv) ALTER TABLE-MODIFY DATATYPE:**

Syntax:

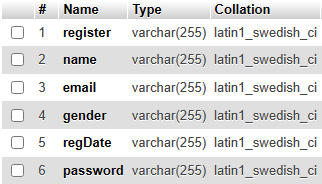
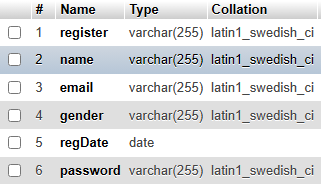
ALTER TABLE table\_name

MODIFY column\_name DATA TYPE;

Example:

[ALTER](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/alter-table.html) [TABLE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/alter-table.html)  student

MODIFY regDate varchar(255);

Output:

**Truncate:**

-This command is used to remove all the rows from the table.

Syntax:

TRUNCATE TABLE table\_name;

Example:

TRUNCATE TABLE student;

**RENAME:**

-It is used to rename the table from the database.

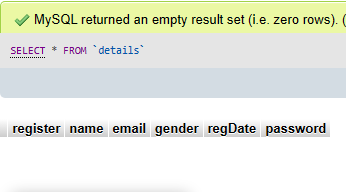
Syntax:

RENAME TABLE old\_table\_name TO new\_table\_name;

Example:

RENAME TABLE student TO details;

Output:



**DROP:**

-It is used to delete a table,index or view from database.

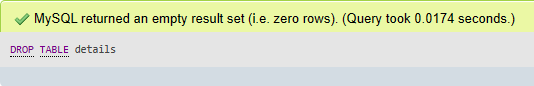
Syntax:

DROP TABLE table\_name;

Example:

DROP TABLE details;

Output:



**Result:**

The DDL COMMANDS are successfully executed and the output is verified.

**Ex No:02                                          DML COMMANDS**

**Aim:**

To work with DML commands

**Procedure:**

The Data Manipulation Language(DML) commands are-

Select - Retrieve data from the database.

Insert - Insert data into a table.

Update - Update existing data within a table.

Delete – Delete records from a database table.

**Insert:**

-Insert command is used to insert a data into a table.

Syntax:

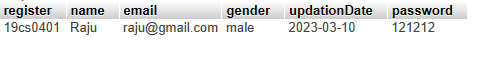
INSERT INTO table\_name(column\_list)

VALUES (column\_values);

Example:

INSERT INTO `student` (`register`, `name`, `email`, `gender`, `updationDate`, `password`) VALUES('19cs0401', 'Raju', 'raju@gmail.com', 'male', curdate(), '121212');

Output:



**Update:**

-Update command is used to update existing data with in a table.

Syntax:

UPDATE table\_name SET column\_name = column\_value WHERE CONDITION;

Example:

UPDATE student SET password = '12345678' WHERE register='19cs0401';

Output:

**SELECT:**

-Select command is used to retrieve data from the database.

Syntax:

* SELECT \* FROM table\_name;
* SELECT column\_name FROM table\_name WHERE CONDITION;

EXAPMLE:

* SELECT \* FROM student;

Output:



* SELECT name FROM student WHERE register=’19cs0401’;

Output:



**DELETE:**

-Delete command is used to delete a records from a database table.

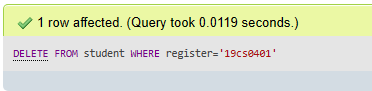
Syntax:

DELETE FROM table\_name WHERE CONDITION;

Example:

DELETE FROM student WHERE register='19cs0401';

Output:



**Result:**

The DML COMMANDS are successfully executed and the output is verified.

**Ex No:03                                          DCL COMMANDS**

**Aim:**

To work with DCL commands.

**Procedure:**

The Data Control Language (DCL) are-

**GRAND :**

- It is used to grant a privilege to a user & allow to perform specific task.

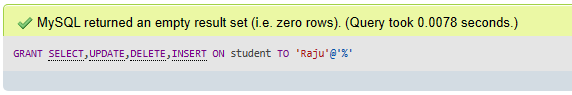
Syntax:

GRANT [privilege\_name or ALL] ON table\_name TO ‘user’@’localhost’;

Example:

GRANT SELECT,UPDATE,DELETE,INSERT ON student TO ‘Raju’@’localhost’;

Output:



**REVOKE:**

- It is used to remove granted privilege from a user.

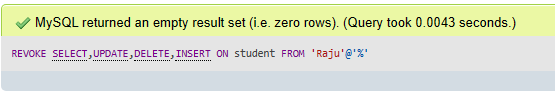
Syntax:

REVOKE [privilege\_name or ALL] ON table\_name FROM user;

Example:

REVOKE [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html),[UPDATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/update.html),[DELETE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/delete.html),[INSERT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/insert.html) ON student FROM 'Raju'@'%';

Output:



**Result:**

The DCL COMMANDS are successfully executed and the output is verified.

**EX.NO:4 SUB QUERY**

**Aim:**

To work with My SQL Sub Query.

**Procedure:**

Sub Query can be simply defined as a query within another query. In other words we can say that a subquery is a query that is embedded in where clause of another query.

**SELECT:**

**-**Sub Query in select statement(command) .

Syntax :

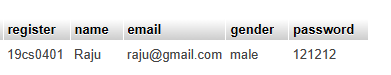
SELECT column\_name FROM table\_name

WHERE (SELECT column\_name FROM table\_name WHERE CONDITION);

Example :

SELECT \* FROM student WHERE (SELECT reg\_no FROM profile\_log WHERE id=3);

Output:



**INSERT:**

**-**Sub Query in insert command.

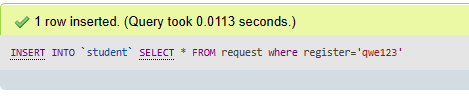
Syntax:

* Insert into table1 select\*From table2;
* Insert into table1 select\*From table2 where condition;

Example:

INSERT INTO `student` SELECT \* FROM request where register='qwe123';

Output:



**UPDATE:**

-Sub Query in update command.

Syntax:

UPDATE table\_name SET column\_Name = Value

WHERE column\_name IN (SELECT column\_name FROM table2 WHERE CONDITION);

Example:

UPDATE student SET `password` ='abcd'

WHERE register IN (SELECT reg\_no FROM profile\_log WHERE id=3);

Output:



**DELETE:**

**-**Sub Query in delete command.

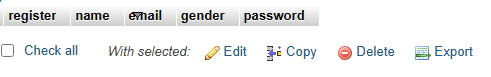
Syntax:

Delete from table where column\_name in (select column\_name from table2 where CONDITION)

Example:

DELETE FROM `student` WHERE register IN(SELECT reg\_no FROM profile\_log WHERE id=3);

Output:



**Result:**

The SUB QUERY COMMANDS are successfully executed and the output is verified.

**EX.NO:4** **SQL JOINS**

**Aim:**

To work with MySQL JOINS.

**Procedure:**

A join clause is used to combine rows from two or more tables, based on a related column between them.

**INNER JOIN:**

**-**this keyword selects records that have matching values in both tables.

Syntax :

Select column name(s) from table;

Inner Join table2

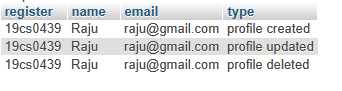
ON table1.column\_name= table2.column\_name;

Example :

SELECT student.register,student.name,student.email,profile\_log.type FROM student

INNER JOIN profile\_log on

student.register=profile\_log.reg\_no;

Output:

**LEFT JOIN:**

**-**returns all records from the left table and matching records(if any) from the right table.

Syntax:

SELECT column\_name(s) From table1;

LEFT JOIN table2 ON table1. Column\_name = table2.column\_name;

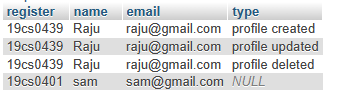
Example:

SELECT student.register,student.name,student.email,profile\_log.type

FROM student LEFT JOIN profile\_log on

student.register=profile\_log.reg\_no;

Output:



**RIGHT JOIN:**

-returns all records from the right table and matching records (if any) from the left table.

Syntax:

SELECT column\_name(s) From table1;

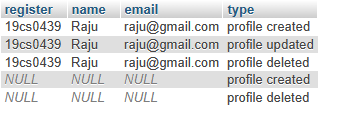
RIGHT JOIN table2 ON table1. Column\_name = table2.column\_name;

Example:

SELECT student.register,student.name,student.email,profile\_log.type FROM student

RIGHT JOIN profile\_log on student.register=profile\_log.reg\_no;

Output:



**CROSS JOIN:**

**-**returns all records from both tables.

Syntax:

Select column\_name(s)

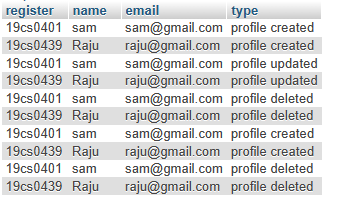
From table1.cross join table2;

Example:

SELECT student.register,student.name,student.email,profile\_log.type FROM student

CROSS JOIN profile\_log ;

Output:



Syntax 2:

Select column\_name(s)

From table1 cross join table2

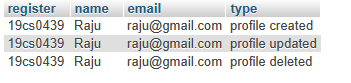
Where condition;[if using where act as a inner join];

Example:

SELECT student.register,student.name,student.email,profile\_log.type FROM student

CROSS JOIN profile\_log on student.register=profile\_log.reg\_no;

Output:



**Result:**

The SQL JOINS are successfully executed and the output is verified.

**Ex No:05 PL/SQL**

**Aim:**

To work with PL/SQL

**Procedure:**

* PL/SQL is a combination of SQL along with the procedural features of programming languages.
* PL/SQL is one of three key programming languages embedded in the Oracle Database, along with SQL itself and Java.
* The PL/SQL programs are divided and written in logical blocks of code. Each block consists of three sub-parts,
* Declare
* Begin
* Exception
* End

**SYNTAX:**

DECLARE

<declarations section>

BEGIN

<executable section commands>

EXCEPTION

<exception handling>

END;

**EXAMPLE:**

**1.Addition of Two Numbers:**

**Program Code:**

declare

x number(5);

y number(5);

z number(5);

begin

x:=50;

y:=20;

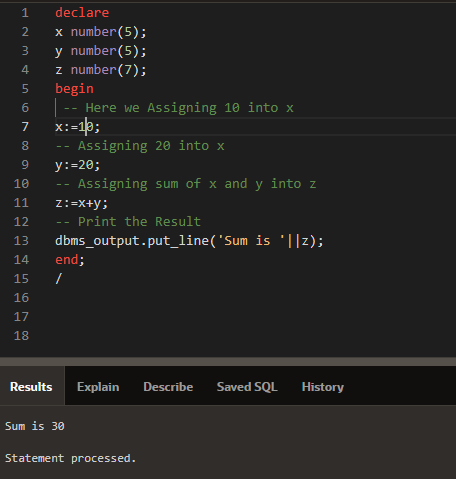
z:=x+y;

dbms\_output.put\_line('sum is'||z);

end;

/

**Output for example-1:**

****

**2.Generating Series:**

**Program Code:**

declare

n number(5);

begin

n:=1;

for i in 1..10 loop

case n

when 1 then

dbms\_output.put\_line(i);

when 2 then

if mod(i,2)=0 then

dbms\_output.put\_line(i);

end if;

when 3 then

if mod(i,2)!=0 then

dbms\_output.put\_line(i);

end if;

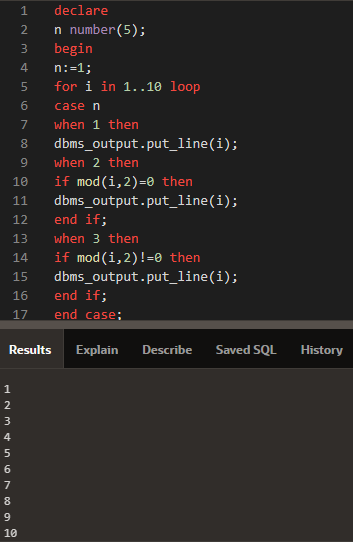
end case;

end loop;

end;

/

**Output for example-2:**



**Result:**The PL/SQL program is successfully executed and the output is verified.

**EX No:06                            Cursors, Procedures and Functions**

**Aim:**

To work with the Cursor, Procedure and Function in PL/SQL

**Procedure:**

Step 1: Create a necessary table for perform action .

Step 2: Then work with database using PL/SQL Block commands

**1.CURSOR:**

**Syntax:**

DECLARE cursor\_name CURSOR

FOR select\_statement;

OPEN cursor\_name;

FETCH NEXT FROM cursor INTO variable\_list;

CLOSE cursor\_name;

**2.PROCEDURE:**

**Syntax:**

DECLARE

CREATE PROCEDURE procedure\_name[[[IN]|OUT|INOUT]parameter\_name datatype[.parameter\_datatype]

AS

Declaration\_section

BEGIN

Executable\_section

END;

BEGIN

[procedure\_name]();

END;

**3.FUNCTION:**

**Syntax:**

CREATE [OR REPLACE] FUNCTION function\_name [parameters]

[(parameter\_name [IN | OUT | IN OUT] type [, ...])]

RETURN return\_datatype

{IS | AS}

BEGIN

   < function\_body >

END [function\_name];

**Example:**

DECLARE

n number ;

t number;

s\_id student.std\_id%type;

s\_name student.name%type;

s\_cgpa student.cgpa%type;

CURSOR s\_student is --cursor

SELECT std\_id, name, cgpa FROM student;

PROCEDURE pro AS --procedure

BEGIN

OPEN s\_student;

LOOP

FETCH s\_student into s\_id, s\_name, s\_cgpa;

EXIT WHEN s\_student%notfound;

s\_cgpa:=s\_cgpa\*10;

dbms\_output.put\_line('Id:' || s\_id || ' Name:' || s\_name || ' Percentage:' || s\_cgpa);

END LOOP;

CLOSE s\_student;

END;

FUNCTION func -- function

RETURN number IS

total number(2) := 0;

BEGIN

SELECT count(\*) into total

FROM student;

RETURN total;

END;

BEGIN --main program

n:=1;

case n

when 1 then

PRO();

when 2 then

t:=func();

dbms\_output.put\_line(t);

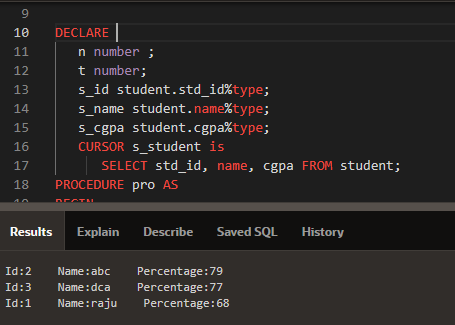
END case;

END;

/

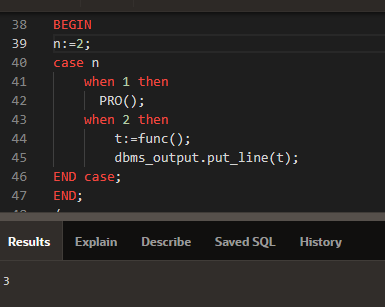
**Output:**

**when n=1:**

****

**Output:**

**when n=2**

****

**Result:**

The cursor, procedure and function in PL/SQL is successfully executed and the output is verified.

**Ex No:07                                          TRIGGERS**

**Aim:**

To work with TRIGGER in mySQL

**Procedure:**

**Syntax:**

CREATE [DEFINER = { user | CURRENT\_USER }]

TRIGGER trigger\_name

trigger\_time trigger\_event ON tbl\_name

FOR EACH ROW

trigger\_body

* trigger\_time: { BEFORE | AFTER }
* trigger\_event: { INSERT | UPDATE | DELETE }

**Example:**

1. Trigger for insert command.

**Trigger Command:**

DROP TRIGGER IF EXISTS `insert`;

CREATE DEFINER=`root`@`localhost` TRIGGER `insert`

AFTER INSERT ON `student`

FOR EACH ROW

INSERT into profile\_log VALUES(null,new.register,'profile created',curdate());

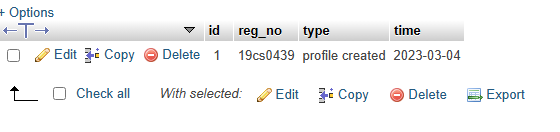
**SQL command:**

INSERT INTO `student` (`register`, `name`, `email`, `gender`, `updationDate`, `password`) VALUES ('19cs0439', 'Raju', 'rajuraj@gmail.com', 'male', CURRENT\_DATE(), 'Raju@1234');

**Table**:student



**Table:**profile\_log

****

1. Trigger for update command.

**Trigger Command:**

DROP TRIGGER IF EXISTS `update`;

CREATE DEFINER=`root`@`localhost` TRIGGER `update`

AFTER UPDATE ON `student`

FOR EACH ROW

INSERT into profile\_log VALUES(null,new.register,'profile updated',curdate());

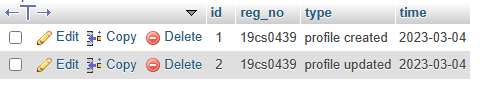
**SQL command:**

UPDATE `student` SET `email`='Raju@gmail.com',`password`=12345678 WHERE register='19cs0439';

**Table**:student

****

**Table**:profile\_log



**3)** Trigger for delete command

**Trigger Command:**

DROP TRIGGER IF EXISTS `delete`;

CREATE DEFINER=`root`@`localhost` TRIGGER `delete`

BEFORE DELETE ON `student`

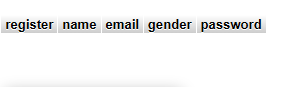
FOR EACH ROW

INSERT into profile\_log VALUES(null,old.register,'profile deleted',curdate());

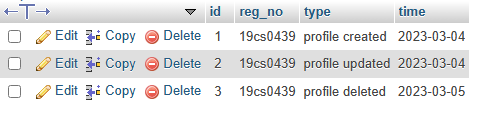
**SQL command:**

[Delete](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/delete.html) from `student` where register='19cs0439';

**Table**:student



**Table**:profile\_log



**Result:**

The Triggers in SQL is successfully executed and the output is noted.

**Ex.No:8 STUDENT’S FILE MANAGEMENT SYSTEM**

**Aim:**

To create a Web Application for STUDENT’S FILE MANAGEMENT SYSTEM(SFMS)

**ABSTRACT:**

The Student's File Management System (SFMS) is a web application designed to help students manage their academic files and assignments online. It simplifies file management and enhances productivity by eliminating the need for physical storage devices.

The SFMS provides a secure centralized location for students to store their academic files and assignments, reducing the risk of data loss or corruption. It is also highly customizable and scalable to meet the specific needs of educational institutions, departments, and individual students.

This paper presents the development and implementation of the SFMS as a web application, highlighting its key features and benefits to students. We also discuss the challenges and opportunities associated with deploying and managing a web-based file management system in an educational environment.

**MODULES:**

The entire project mainly consists of  **2** modules, which are

Admin module.

Student module.

**Software Requirements:**

* Operating system – Windows 10
* Web Server : XAMPP [5.6.4]
* Database : MYSQL [5.0.21]
* Coding Language : Web Tech (HTML, CSS, Java Script, PHP)

**LIMITATION OF EXISTING SYSTEM**

• Manual work.

• Time consuming.

• Lack of reliability.

**Database Structure:**

Database **:** sfms

Table1 : admin

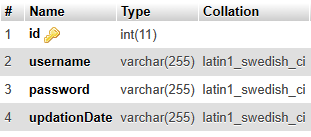


Table2 : request

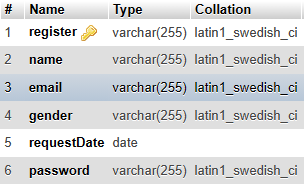
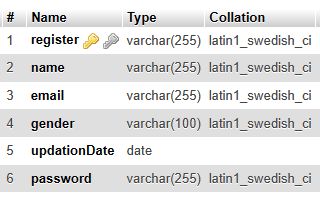


Table3 : student



**Code:**

**1)Config.php**

<?php

define('DB\_SERVER','localhost');

define('DB\_USER','root');

define('DB\_PASS' ,'');

define('DB\_NAME', 'sfms');

$con = mysqli\_connect(DB\_SERVER,DB\_USER,DB\_PASS,DB\_NAME);

// Check connection

if (mysqli\_connect\_errno())

{

echo "Failed to connect to MySQL: " . mysqli\_connect\_error();

}

?>

**2)registration.php**

<?php

include\_once('include/config.php');

if(isset($\_POST['submit']))

{

$name=$\_POST['name'];

$register=$\_POST['register'];

$gender=$\_POST['gender'];

$email=$\_POST['email'];

$password=$\_POST['password'];

$query=mysqli\_query($con,"insert into request(name,register,email,gender,password,requestDate) values('$name','$register','$email','$gender','$password',curdate())");

if($query)

{ echo "<script>alert('Successfully Requested. You can login after few minutes ');</script>";

//header('location:index.php');

}

else {

echo "<script>alert('request unsuccessfully');</script>";

}

}

?>

**3)user-login.php**

<?php

session\_start();

error\_reporting(0);

include("include/config.php");

if(isset($\_POST['submit']))

{

$ret=mysqli\_query($con,"SELECT \* FROM student WHERE register='".$\_POST['register']."' and password='".$\_POST['password']."'");

$num=mysqli\_fetch\_array($ret);

if($num>0)

{

$extra="dashboard.php";//

$\_SESSION['login']=$\_POST['register'];

$\_SESSION['id']=$num['id'];

$host=$\_SERVER['HTTP\_HOST'];

$uip=$\_SERVER['REMOTE\_ADDR'];

$status=1;

// For stroing log if user login successfull

$log=mysqli\_query($con,"insert into userlog(uid,username,userip,status) values('".$\_SESSION['id']."','".$\_SESSION['login']."','$uip','$status')");

$uri=rtrim(dirname($\_SERVER['PHP\_SELF']),'/\\');

header("location:http://$host$uri/$extra");

exit();

}

else

{ // For stroing log if user login unsuccessfull

$\_SESSION['login']=$\_POST['register'];

$uip=$\_SERVER['REMOTE\_ADDR'];

$status=0;

mysqli\_query($con,"insert into userlog(username,userip,status) values('".$\_SESSION['login']."','$uip','$status')");

$\_SESSION['errmsg']="Invalid username or password";

$extra="index.php";

$host = $\_SERVER['HTTP\_HOST'];

$uri = rtrim(dirname($\_SERVER['PHP\_SELF']),'/\\');

header("location:http://$host$uri/$extra");

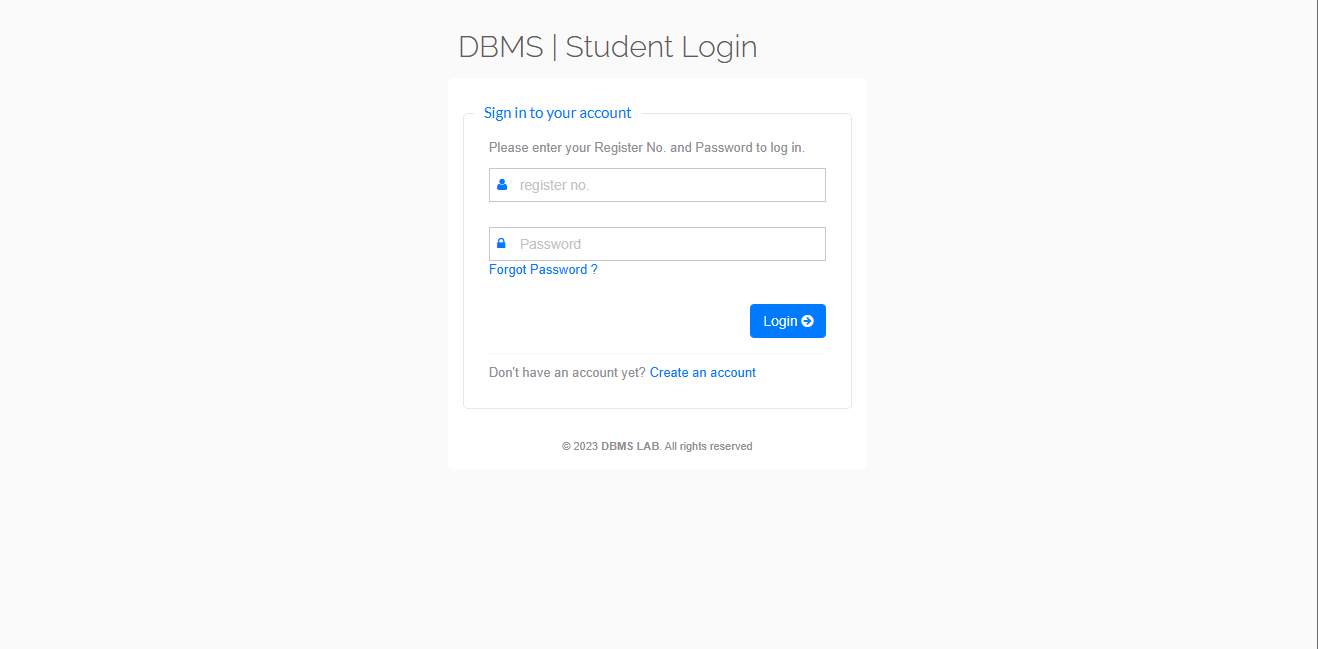
exit();

}

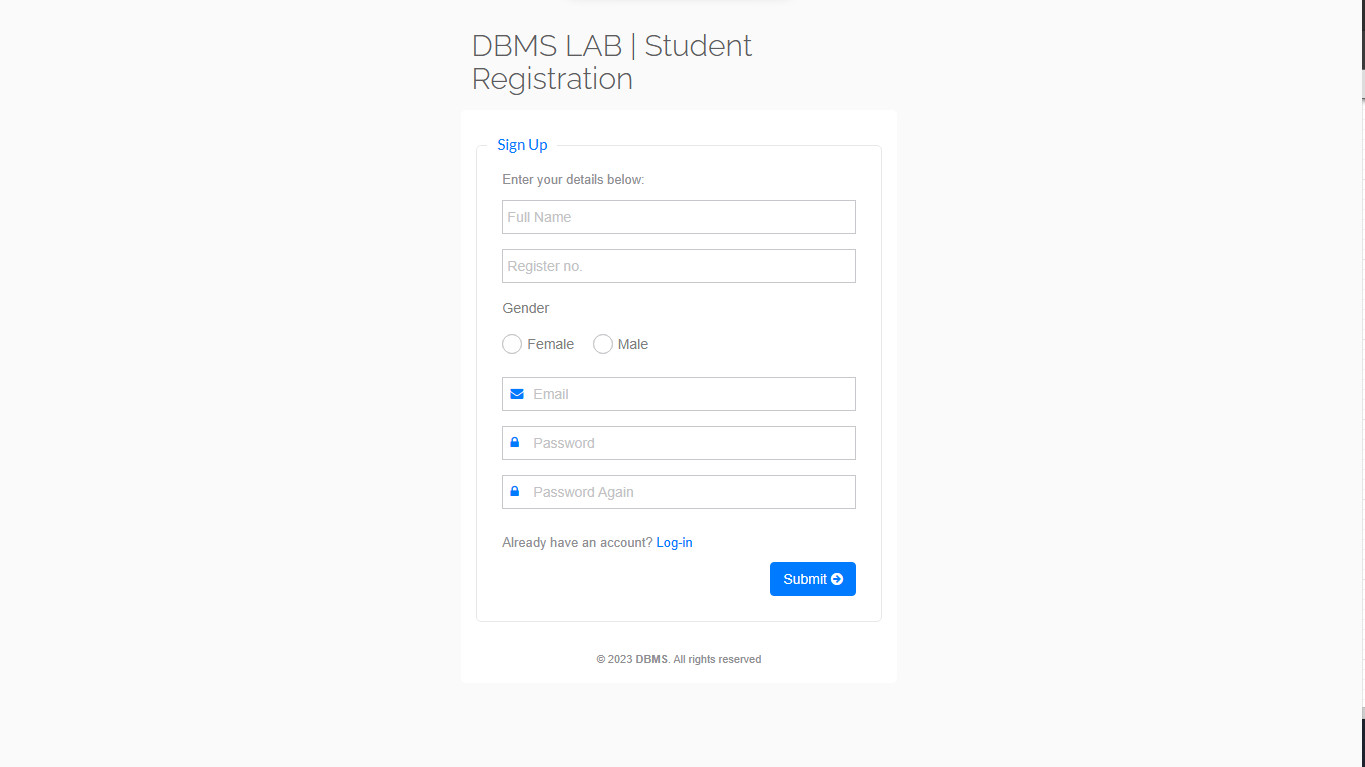
}

?>

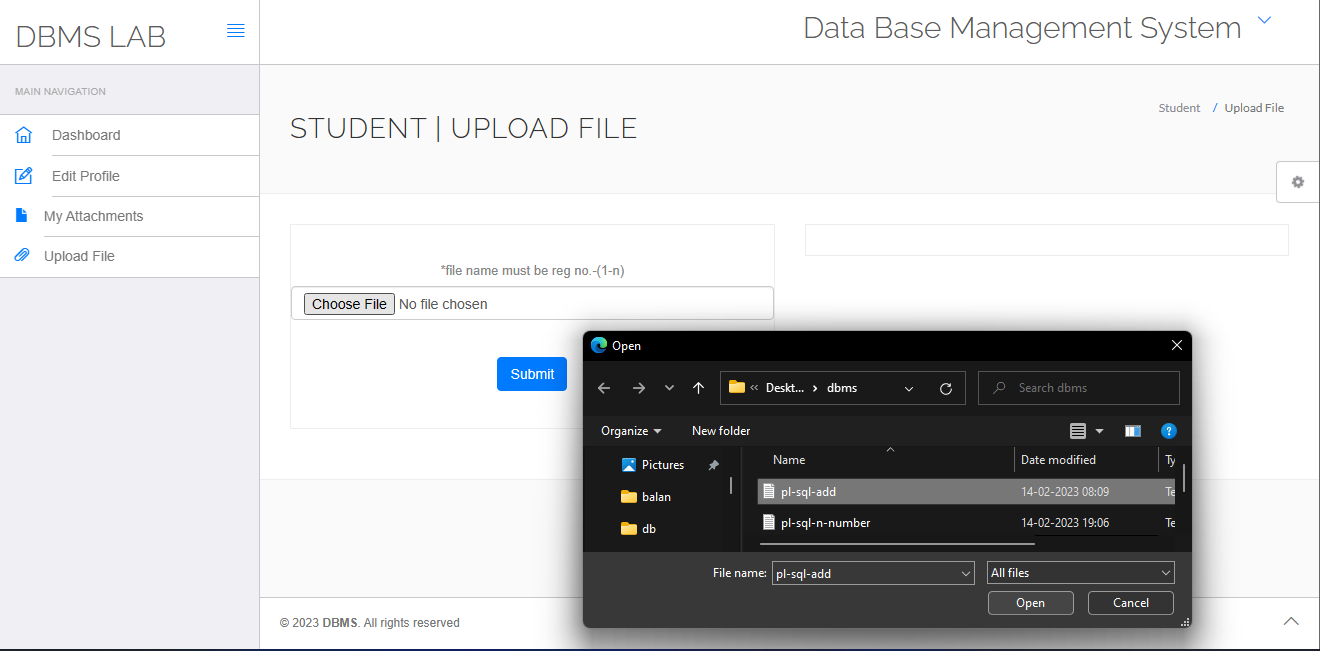
**Output:**

1)Student Login:

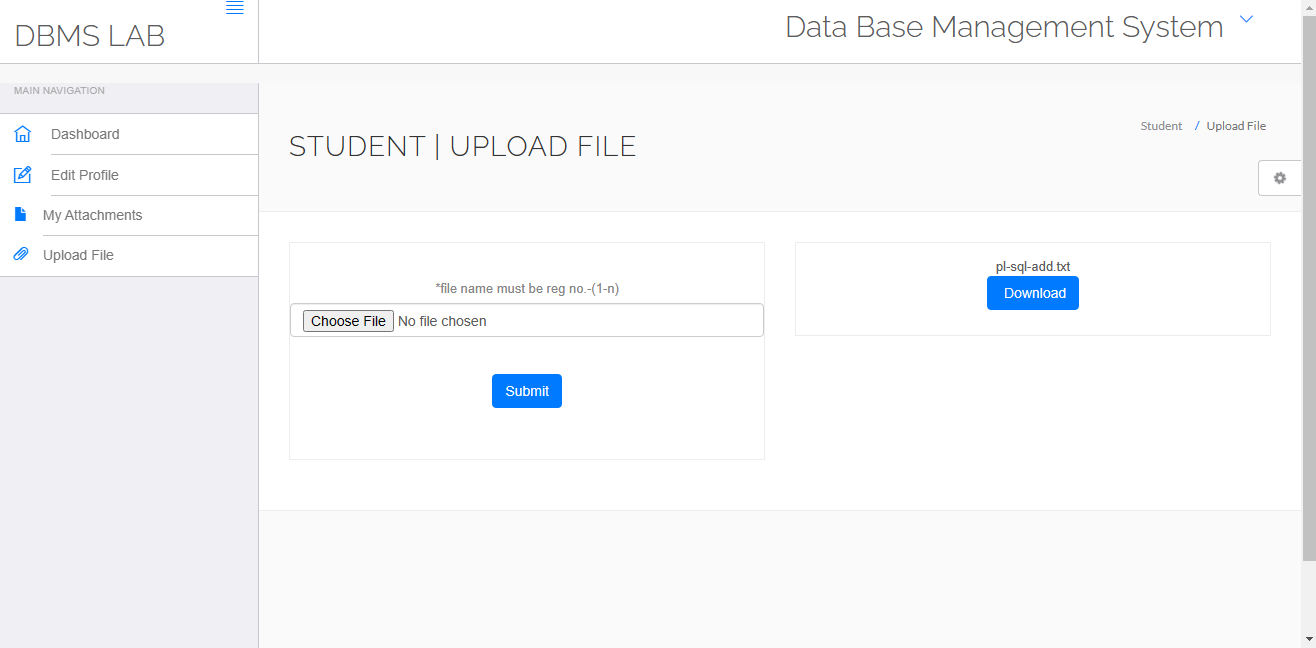
2)Student Request:



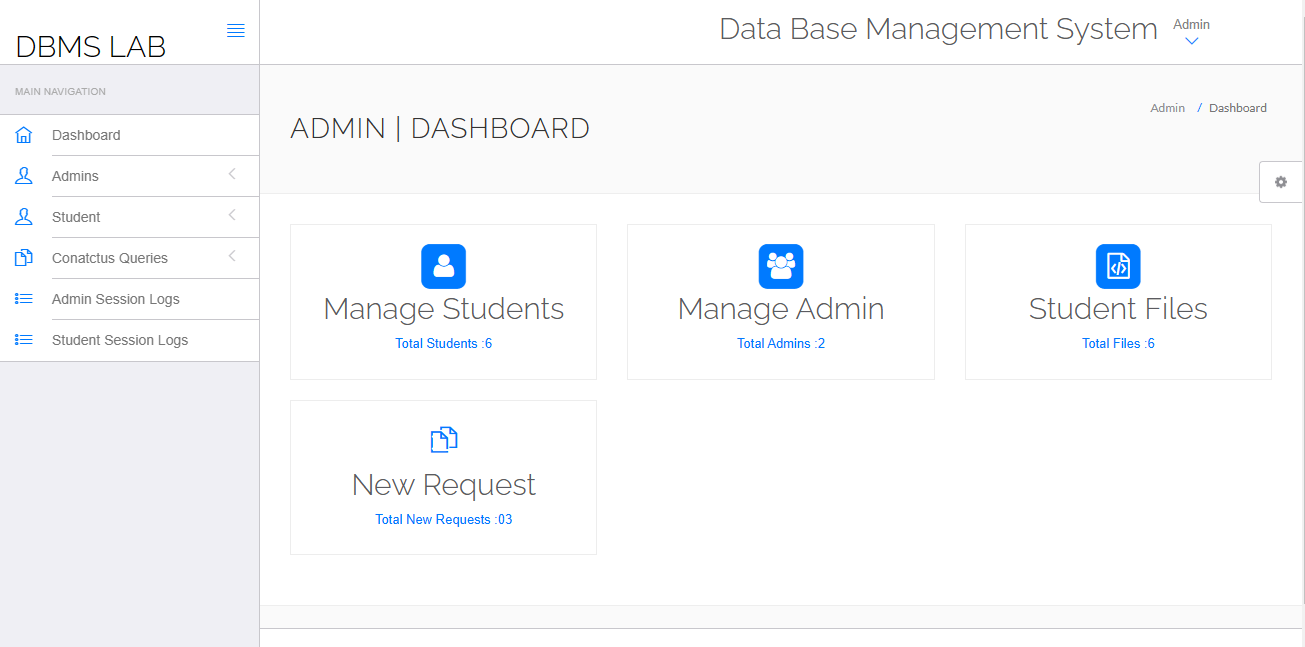
3)Student file upload:



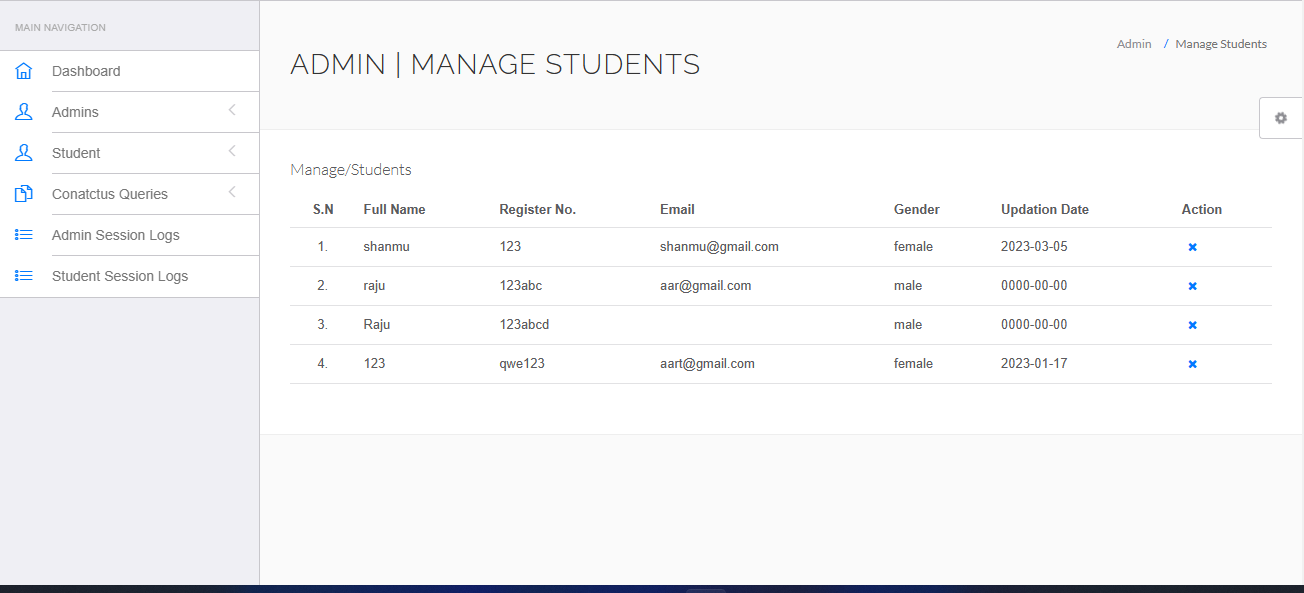
4)Student File view:

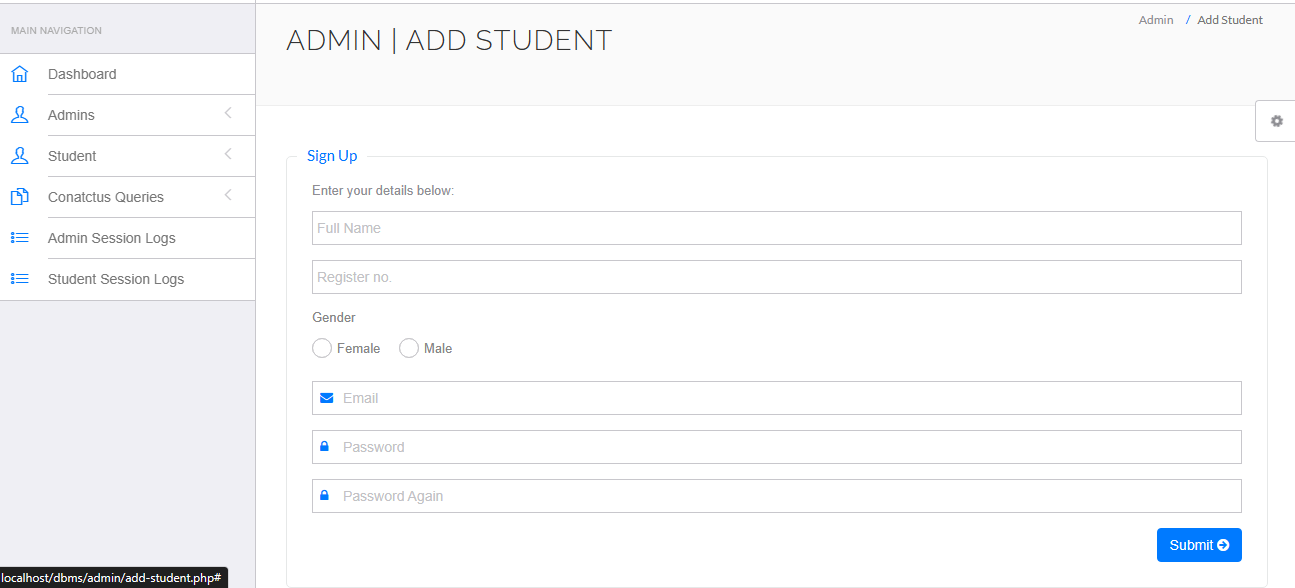


5)Admin Dashboard:



6)Admin Manage(delete) Students:



7)Admin Add Student:

8)Admin View Students Files:

