Experiment No: 01

Experiment Name: Admission Processing System.

Objective: The objective of this experiment is to create an admission processing system.

Theory:

Software Development Life Cycle (SDLC) is a framework that describes the activities performed at each stage of a software development project. The steps given below describe implementation of proposed system:

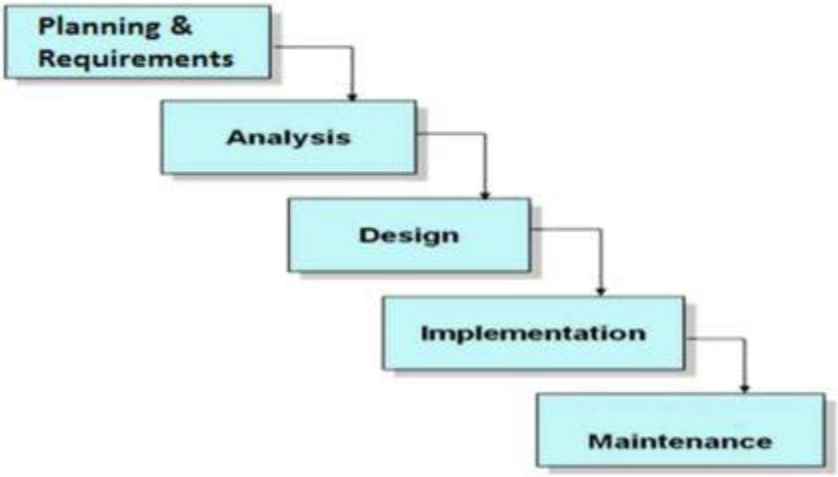


Fig: System Development Life Cycle

Planning & Requirements:

The project plan was prepared by the Project Manager to form the basis of the actual project work plan. This formed the baseline against the project being tracked and controlled

Equipment/development tools

-a PC(Any Os installed) or MAC
-XAMPP server installed
-Notepad++
-HTML,CSS,Javascipt
-Server language:PHP
-Database system: MySQL

Analysis:

Once the project was initiated, requirements gathering and analysis was the first phase in software development life cycle and a process of reviewing business's processes to determine the business needs and functional requirements that a system must meet. During this phase, requirements were analyzed to identify the application flow, the business logic implementation, and their context and usage pattern. The analysis process started with the study and analysis of the current application.

Design:

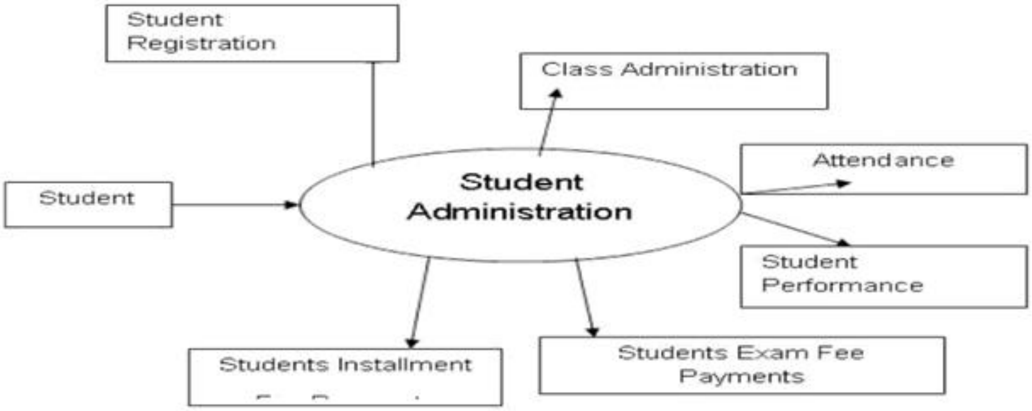


Fig: E-R diagram of admission prcessing system.

Implementation:

Procedure:

1. Turn on xampp and start Apache and MySQL
2. Go to C:\Xampp\htdocs, create a folder named admission_processing and write the codes with Notepad++
3. Go to a browser and type localhost/phpmyadmin
4. Setup database
5. Go to a browser and type localhost/admission_processing
6. Use the created system.

Sample code:

For searching:

```
<?php include "db.php" ?>
<html>
<head>
   <title>Admission Processing Sytem</title>
</head>
<body>
   <form action="search.php" method="post">
      <fieldset>
         <legend align="center">Show Candidate Information</legend>
         <table>
            <tr>
               <td>Candidate Name:</td>
               <td><input type="text" name="ca_Name" required ></td>
            </tr>
```

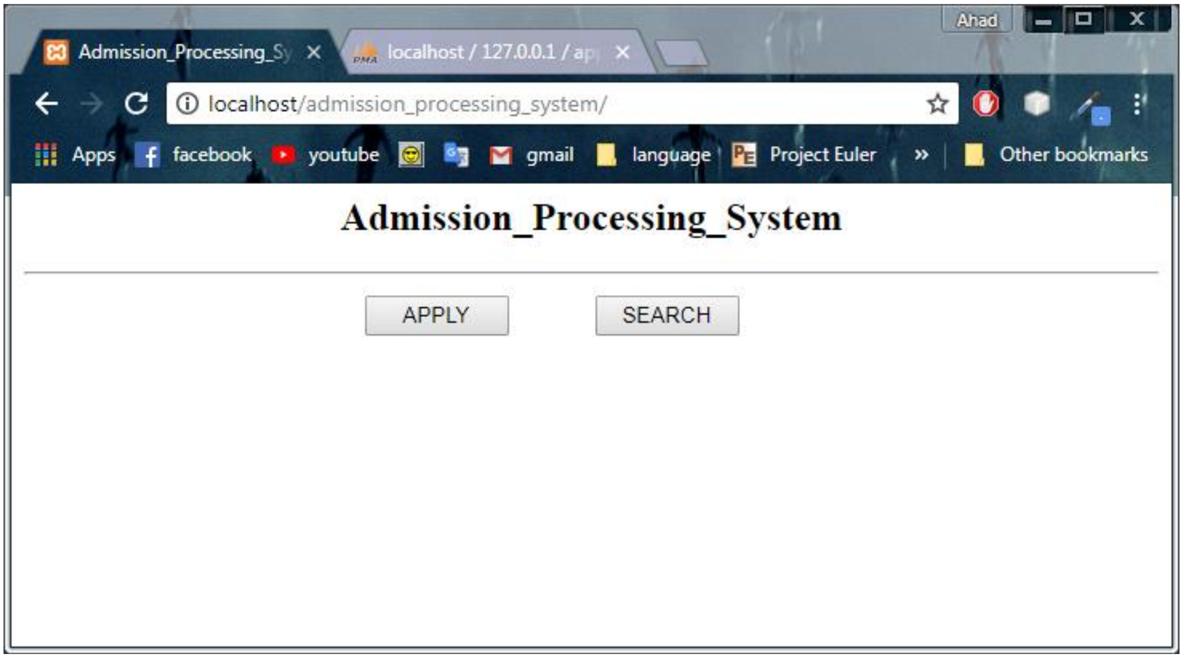
```
                <tr>
                    <td>Email:</td>
                    <td><input type="email" name="email" required ></td>
                </tr>
                <tr>
                    <td>Date Of Birth:</td>
                    <td><input type="date" name="date" required></td>
                </tr>
            </table>
            <button type="submit" name="submit">Search</button>
        </fieldset>
    </form>
</body>
</html>
<?php
if(isset($_POST["submit"]))
{
    $name = $_POST["ca_Name"];
    $email = $_POST["email"];
    $date = $_POST["date"];

    $query = "select * from admission where candidate_name = '$name' and email = '$email' and date =
'$date'";
    $result = mysqli_query($connection, $query);
    if(!$result){
        die("Query Failed". mysqli_error());
    }
    $row = mysqli_fetch_assoc($result);
    echo "<table border='1' align='center'>
                <tr align='center'>
                    <td style='width:150px'><b>Course Name</b></td>
                    <td style='width:150px'><b>".$row['cName']."</b></td>
                </tr>
                <tr align='center'>
                    <td style='width:150px'><b>Candidate Name</b></td>
                    <td style='width:150px'><b>".$row['candidate_name']."</b></td>
                </tr>
                <tr align='center'>
                    <td style='width:150px'><b>Father Name</td>
                    <td style='width:150px'><b>".$row['fName']."</b></td>
                </tr>
                <tr align='center'>
                    <td style='width:150px'><b>Date Of Birth</b></td>
                    <td style='width:150px'><b>".$row['date']."</b></td>
                </tr>
                <tr align='center'>
                    <td style='width:150px'><b>Religion</b></td>
                    <td style='width:150px'><b>".$row['religion']."</b></td>
                </tr>
                <tr align='center'>
                    <td style='width:150px'><b>Present Address</b></td>
                    <td style='width:150px'><b>".$row['present']."</b></td>
                </tr>
                <tr align='center'>
                    <td style='width:150px'><b>Parmanent Address</td>
                    <td style='width:150px'><b>".$row['parmanent']."</b></td>
                </tr>
                <tr align='center'>
                    <td style='width:150px'><b>Nationality</td>
                    <td style='width:150px'><b>".$row['nationality']."</b></td>
                </tr>
                <tr align='center'>
                    <td style='width:150px'><b>Mobile Number</td>
                    <td style='width:150px'><b>".$row['mobile_no']."</b></td>
                </tr>
                <tr align='center'>
                    <td style='width:150px'><b>Email</td>
                    <td style='width:150px'><b>".$row['email']."</b></td>
                </tr>
            </table>";
}

?>
```

For database connection:

```
<?php
    $connection = mysqli_connect("localhost", "root", "", "application");
    if(!$connection){
        die("Connection failed");
    }
 ?>
```

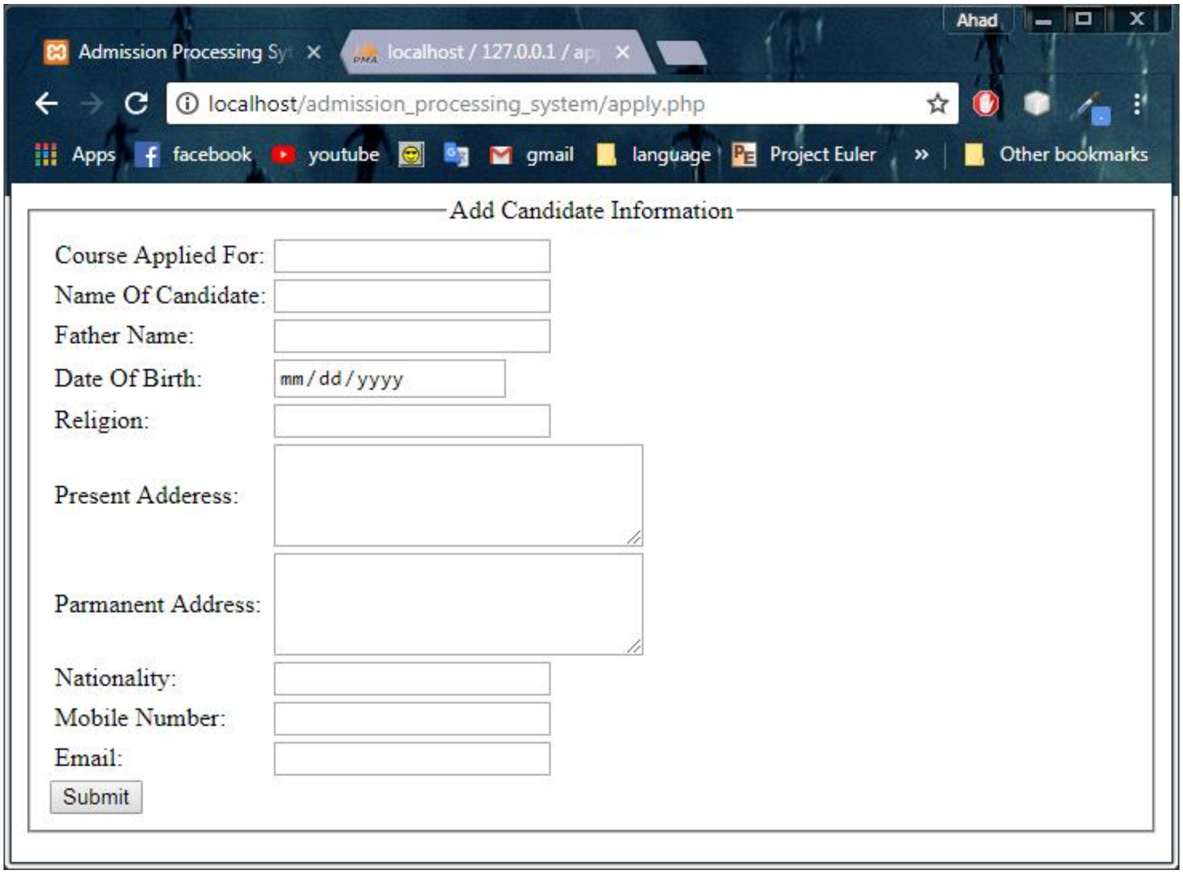
System snapshots:

1.



Admission_Processing_System

2.



Add Candidate Information

Course Applied For:

Name Of Candidate:

Father Name:

Date Of Birth: mm/dd/yyyy

Religion:

Present Adderess:

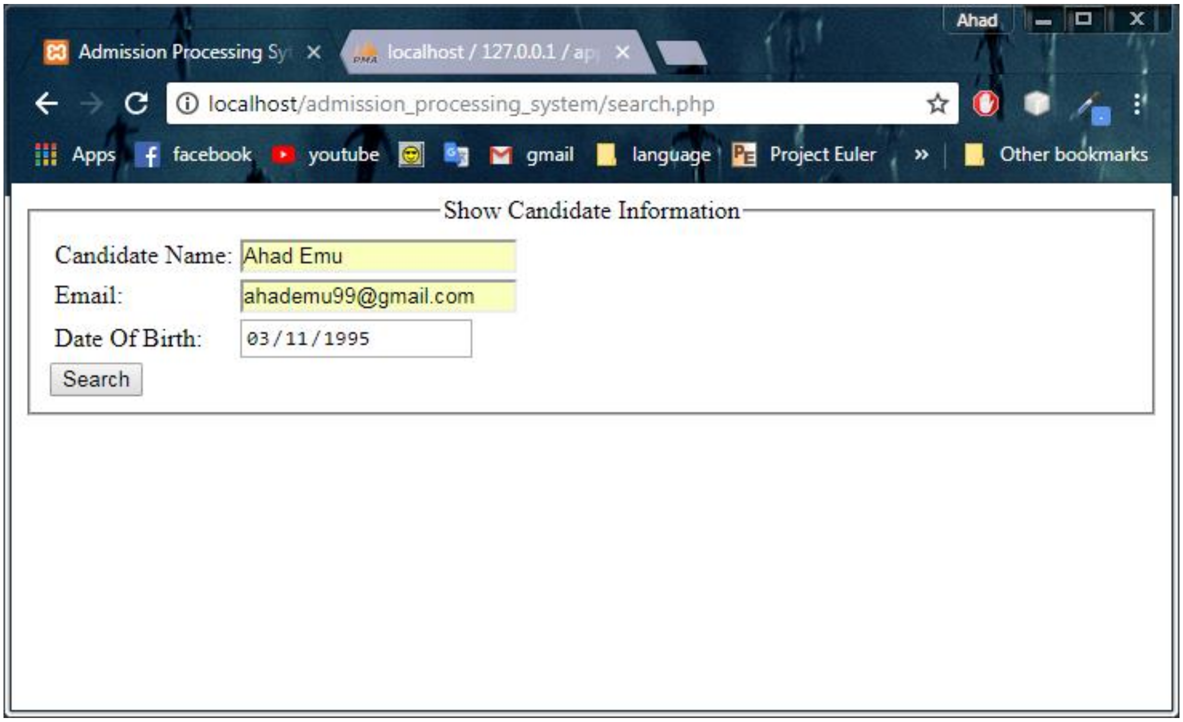
Parmanent Address:

Nationality:

Mobile Number:

Email:

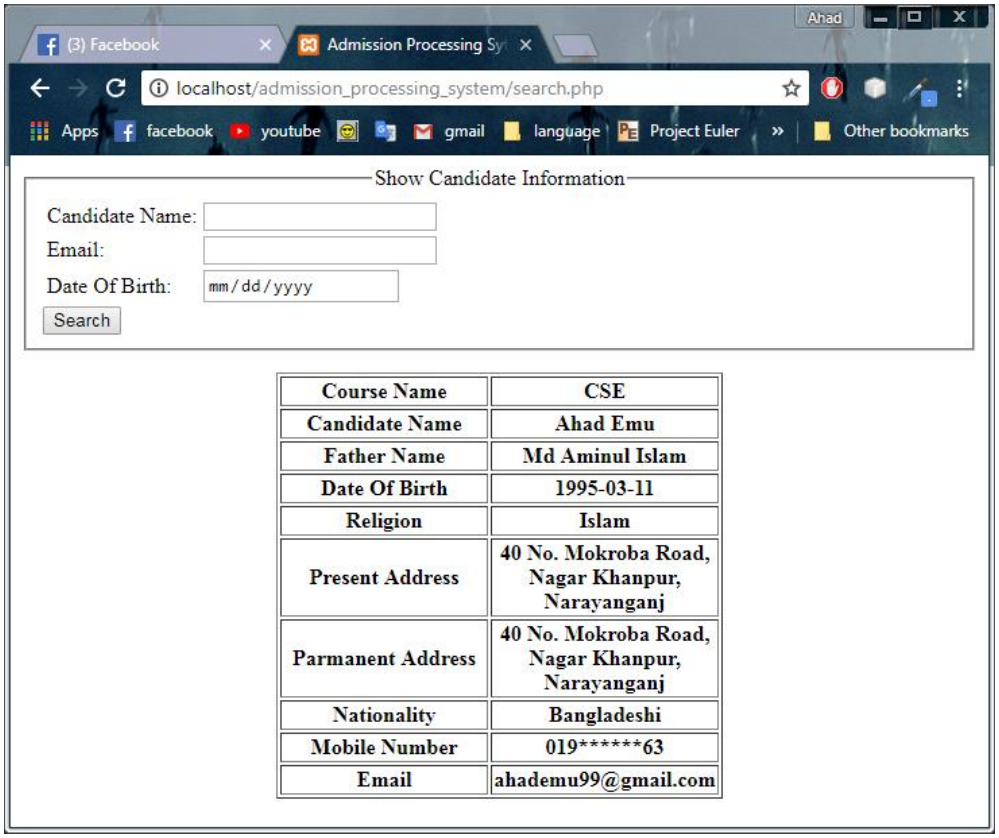
3.



Show Candidate Information

Candidate Name:	Ahad Emu
Email:	ahademu99@gmail.com
Date Of Birth:	03/11/1995

4.



Show Candidate Information

Candidate Name:	
Email:	
Date Of Birth:	mm/dd/yyyy

Course Name	CSE
Candidate Name	Ahad Emu
Father Name	Md Aminul Islam
Date Of Birth	1995-03-11
Religion	Islam
Present Address	40 No. Mokroba Road, Nagar Khanpur, Narayanganj
Parmanent Address	40 No. Mokroba Road, Nagar Khanpur, Narayanganj
Nationality	Bangladeshi
Mobile Number	019******63
Email	ahademu99@gmail.com

Maintenance:

1. A customer/user support structure and any other necessary operational support processes should be in place.
2. Any planned changes to the system or software should be scheduled, communicated, and documented.
3. Continuous security penetration testing is conducted on the system or software throughout its life cycle at regularly scheduled intervals.

Mandatory security testing is conducted when any major configuration or architecture change is made

Experiment No: 02

Experiment Name: Write a program to make Student Information software with the help of database

Theory:

Software Development Life Cycle (SDLC) is a framework that describes the activities performed at each stage of a software development project . The steps given below describe implementation of proposed system:

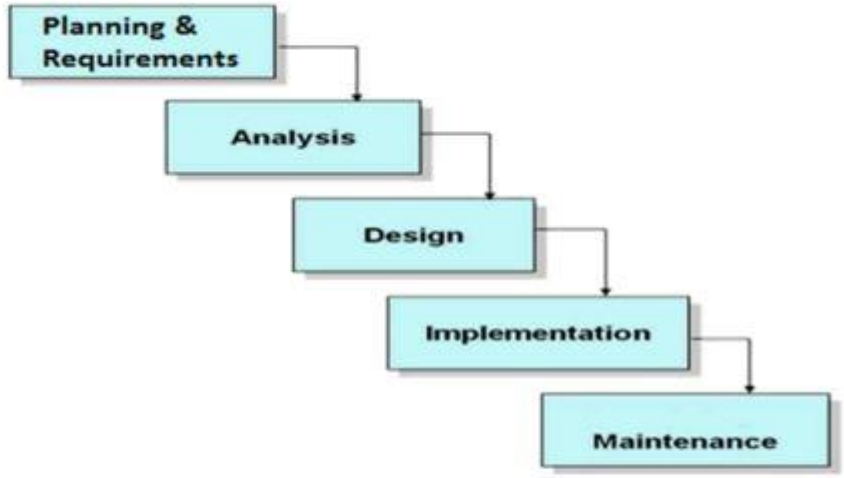


Fig: System Development Life Cycle

Planning & Requirements:

The user of this module is students and parents. They can view or print the student report card information. to access this module they must enter key in student ID.

Equipment/development tools

-a PC(Any Os installed) or MAC
-XAMPP server installed
-Notepad++
-HTML,CSS,Javascipt
-Server language:PHP
-Database system: MySQL

Analysis:

The system is going to be known as the Student Information Management System for secondary school . The system is about the student management. The main reason of developing the system is to overcome the problems that are faced using manual way. The groups of people that are going to use the system are the clerk, Principal, teachers and parent.
Data retrieval, distribution, usage, maintenance, and storage are the important components of the information management. Accurate information, updated, and
Reliable to achieve when needed is as critical to make an effective decision.

Design:

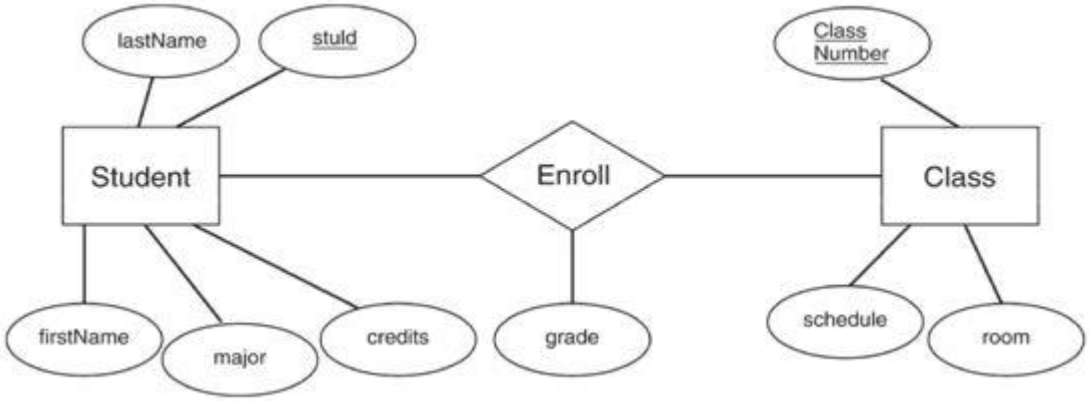


Fig: E-R diagram of student information system

Implementation:

Procedure:

1. Turn on xampp and start Apache and MySQL
2. Go to C:\Xampp\htdocs, create a folder named student_information and write the codes with Notepad++
3. Go to a browser and type localhost/phpmyadmin
4. Setup database
5. Go to a browser and type localhost/ student_information
6. Use the created system.

Sample code:

For Deletion:

```
<?php include "db.php" ?>
<?php
if(isset($_POST["submit"])){
   $roll = $_POST["search"];
   $query = "DELETE FROM `student_information` where roll=$roll";
   $result = mysqli_query($connection, $query);
   if(!$result){
      die("Query Failed". mysqli_error());
   }
}
?>
<html>
<head>
   <title></title>
</head>
</body>
   <h2>Delete Student Information</h2>
   <form action="delete.php" method="post">
```

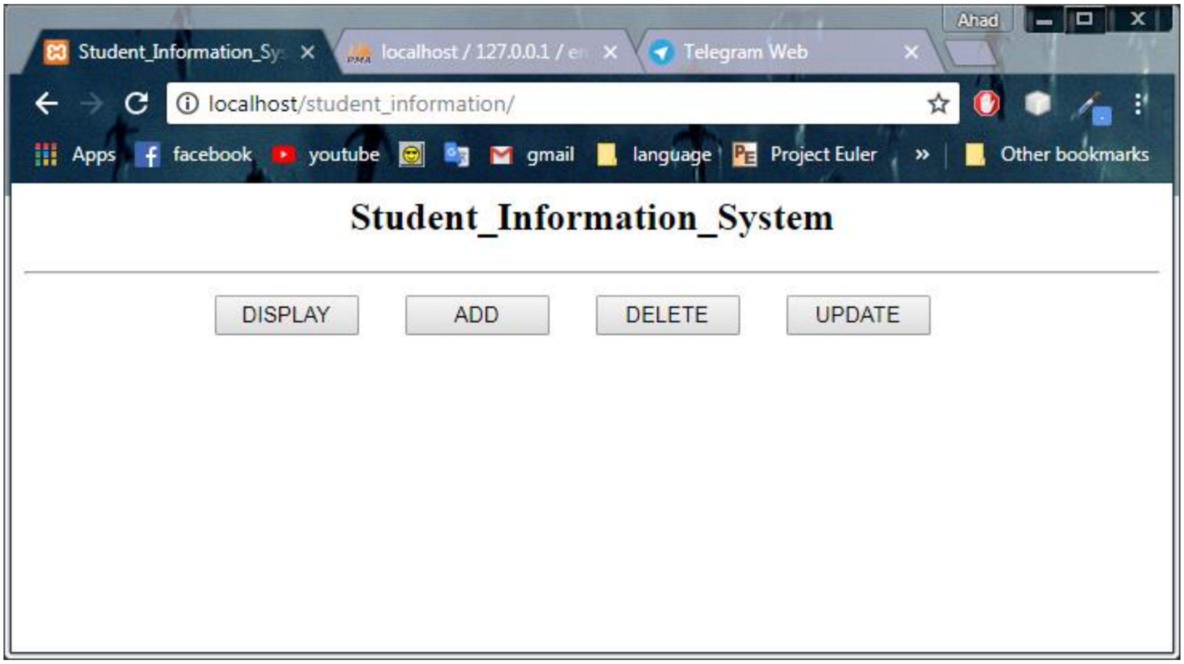
```
        <input type="number" name="search">
        <input type="submit" name="submit">
    </form>
</body>
</html>
```

For Database connection:

```
<?php
    $connection = mysqli_connect("localhost", "root", "", "student");
    if(!$connection){
        die("Connection failed");
    }
 ?>
```

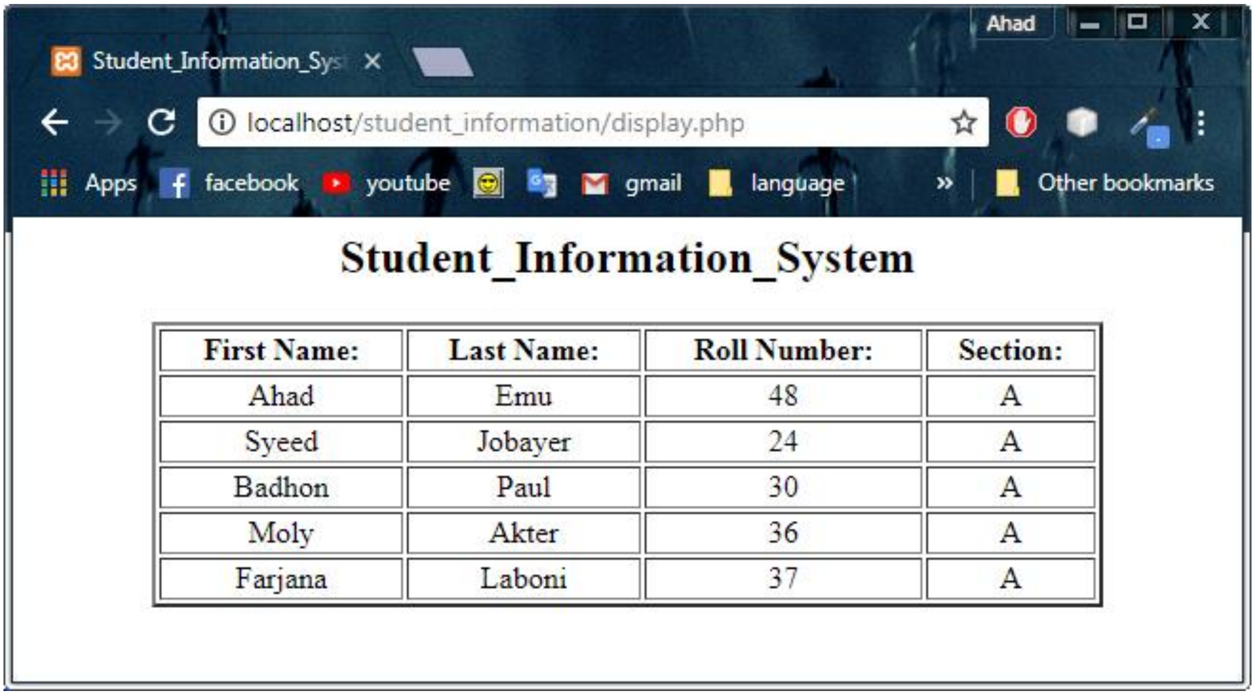
System snapshots:

1.



Student_Information_System

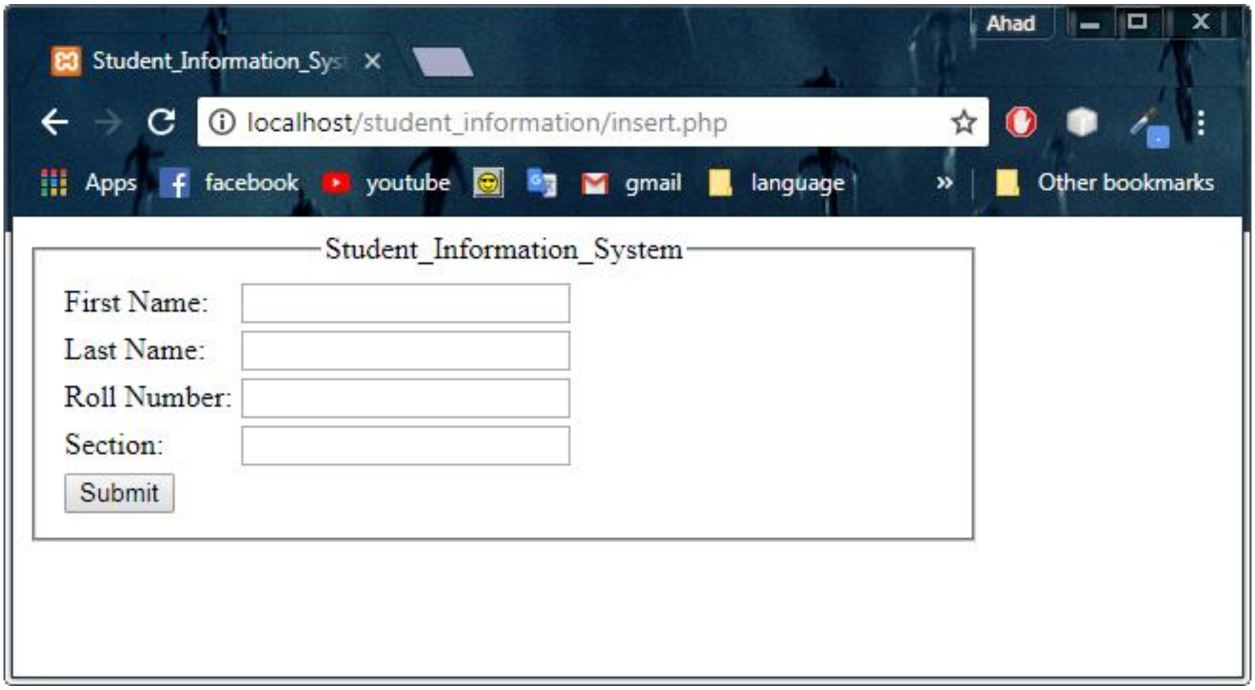
2.



Student_Information_System

First Name:	Last Name:	Roll Number:	Section:
Ahad	Emu	48	A
Syeed	Jobayer	24	A
Badhon	Paul	30	A
Moly	Akter	36	A
Farjana	Laboni	37	A

3.



Student_Information_System

First Name:

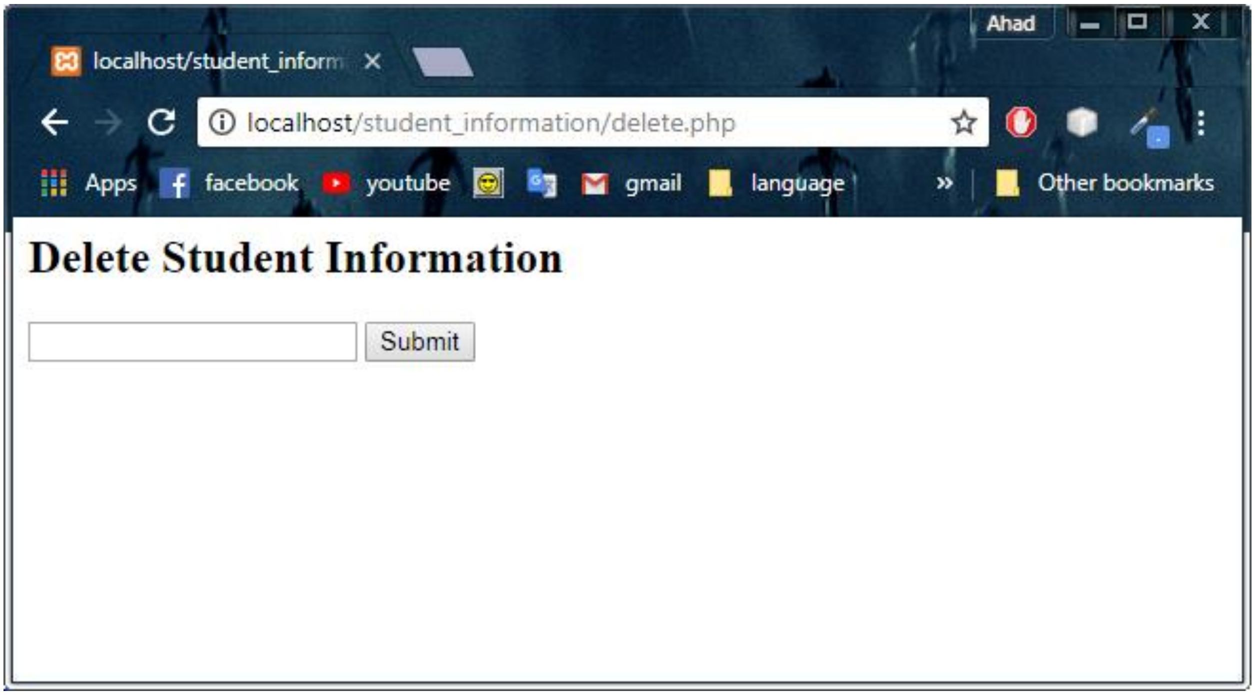
Last Name:

Roll Number:

Section:

Submit

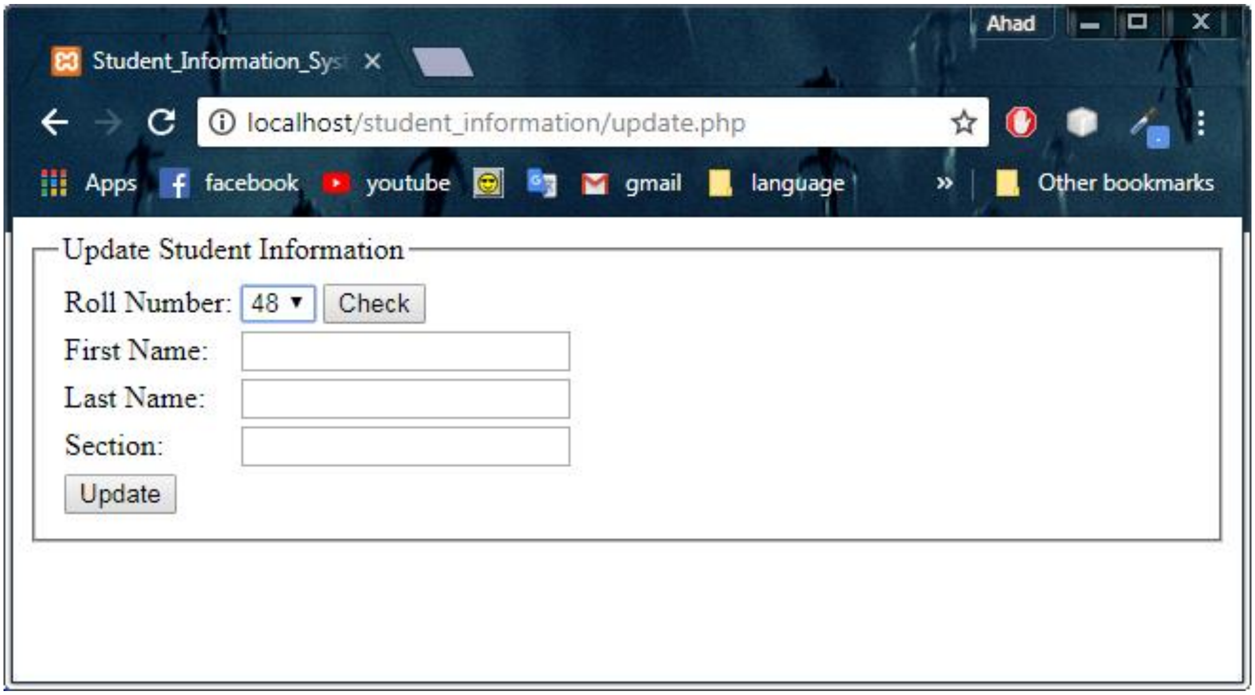
4.



Delete Student Information

Submit

5.



Update Student Information

Roll Number: 48 ▼ Check

First Name:

Last Name:

Section:

Update

Maintenance:

- A user support structure and any other necessary operational support processes should be in place.
- Any planned changes to the system or software should be scheduled, communicated, and documented.
- Continuous security penetration testing is conducted on the system or software throughout its life cycle at regularly scheduled intervals.
- Mandatory security testing is conducted when any major configuration or architecture change is made.

Experiment No: 03

Experiment Name: Write a program in make a **Employee informationSystem** software with the help of database.

Objective: To write a program in manage employee information system to store information in database then display it.

Planning & Requirements:

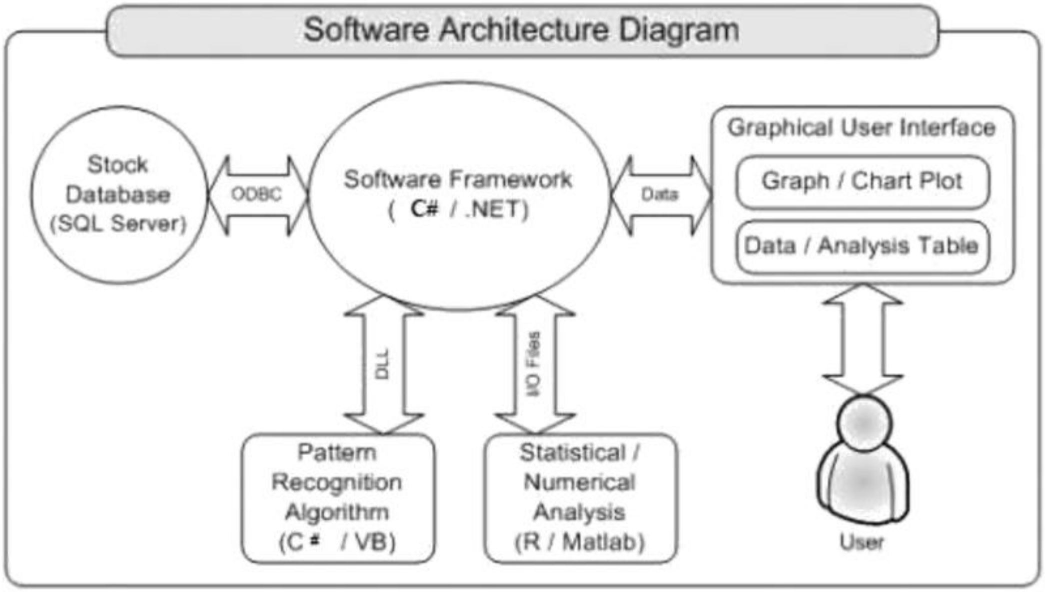
Functional requirements:

1. User post employee information.
2. User put information in user form.
3. Store the information in database
4. Display the information in data grid view.

Equipment/development tools

5. -a PC(Any Os installed) or MAC
6. -XAMPP server installed
7. -Notepad++
8. -HTML,CSS,Javascipt
9. -Server language:PHP
10. -Database system: MySQL

Design:


Software Architecture Diagram
Stock Database (SQL Server)
ODBC
Software Framework (C# / .NET)
Data
Graphical User Interface
Graph / Chart Plot
Data / Analysis Table
DLL
I/O Files
Pattern Recognition Algorithm (C# / VB)
Statistical / Numerical Analysis (R / Matlab)
User

Implementation:

Procedure:

1. Turn on xampp and start Apache and MySQL
2. Go to C:\Xampp\htdocs, create a folder named employee_information and write the codes with Notepad++
3. Go to a browser and type localhost/phpmyadmin
4. Setup database
5. Go to a browser and type localhost/employee_information
6. Use the created system.

Sample code:

For displaying:

```
<?php include "db.php" ?>
```

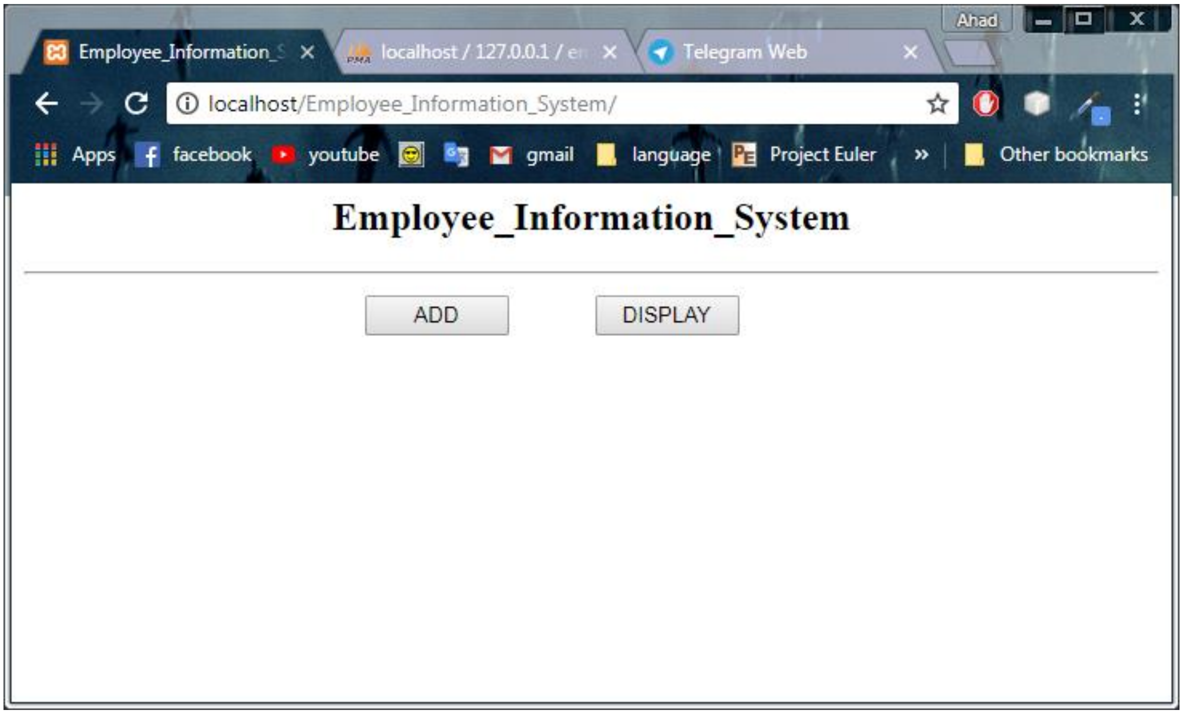
```
<?php
 $query = "select * from employee_information Order by employee_id";
 $result = mysqli_query($connection, $query);
 if(!$result){
     die("Query Failed". mysqli_error());
 }
?>
<html>
<head>
    <title>Employee_Information_System</title>
</head>
<body>
    <h2 align="center">Employee_Information_System</h2>
    <hr />
    <table border="2" height"auto" width="500px" align="center">
        <tr>
            <th>Employee ID:</th>
            <th>Employee Name:</th>
            <th>Designation:</th>
            <th>Join Date:</th>
            <th>Address:</th>
        </tr>
    <?php
        while($row = mysqli_fetch_assoc($result)){
    ?>
        <tr>
            <td align="center"><?php echo $row['employee_id']; ?></td>
            <td align="center"><?php echo $row['employee_name']; ?></td>
            <td align="center"><?php echo $row['designation']; ?></td>
            <td align="center" width="100px"><?php echo$row['join_date']; ?></td>
            <td align="center" width="150px"><?php echo$row['address']; ?></td>
        </tr>
    <?php
        }
     ?>
     </table>
</body>
</html>
```

For database connection:

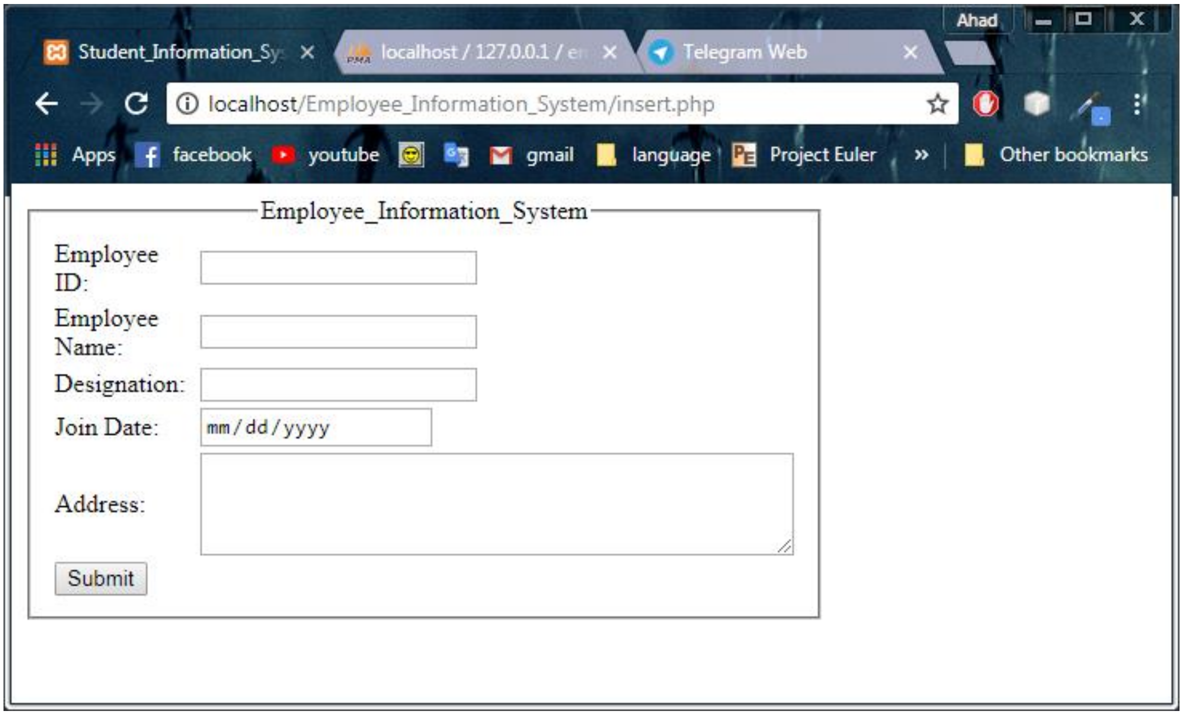
```
<?php
    $connection = mysqli_connect("localhost", "root", "", "employee");
    if(!$connection){
        die("Connection failed");
    }
 ?>
```

System snapshots:

1.



2.



Employee_Information_System

Employee ID:

Employee Name:

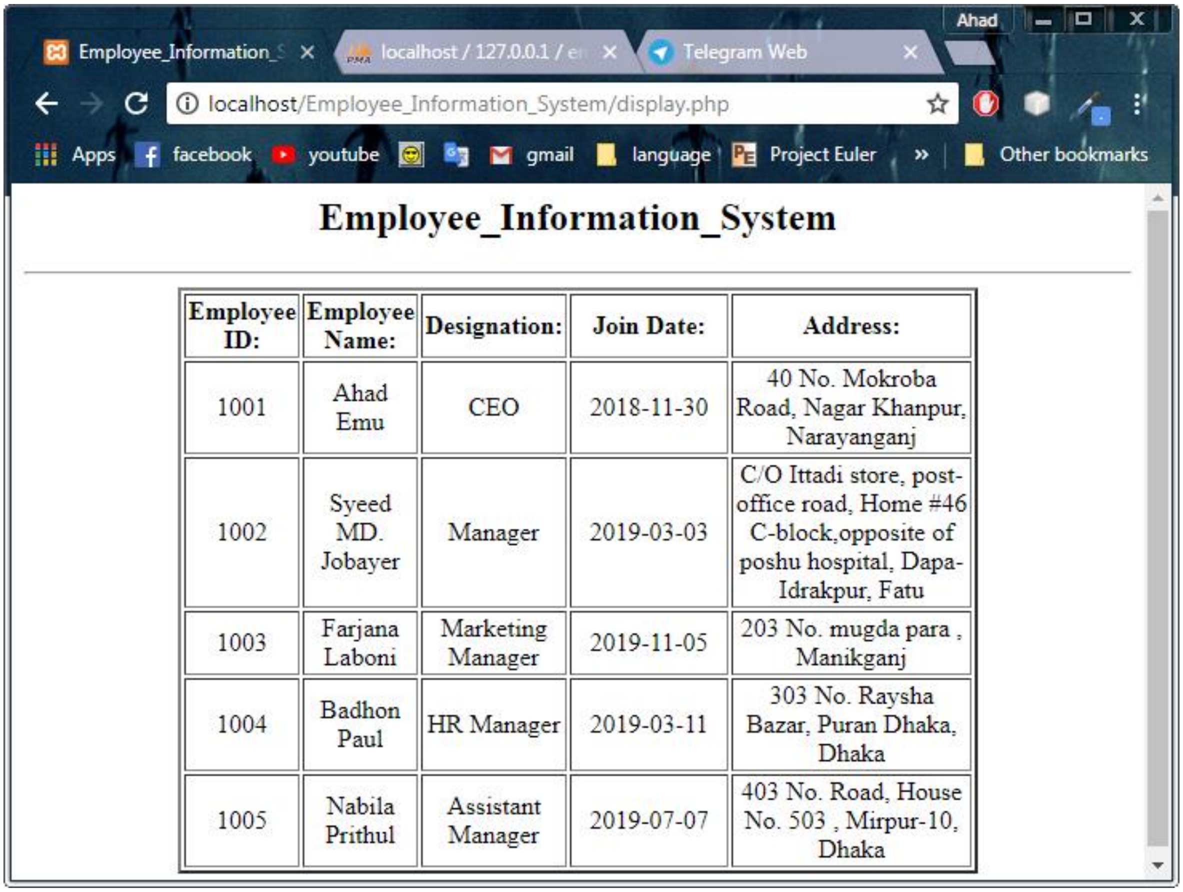
Designation:

Join Date: mm/dd/yyyy

Address:

Submit

3.



Employee_Information_System

Employee ID:	Employee Name:	Designation:	Join Date:	Address:
1001	Ahad Emu	CEO	2018-11-30	40 No. Mokroba Road, Nagar Khanpur, Narayanganj
1002	Syeed MD. Jobayer	Manager	2019-03-03	C/O Ittadi store, post-office road, Home #46 C-block,opposite of poshu hospital, Dapa-Idrakpur, Fatu
1003	Farjana Laboni	Marketing Manager	2019-11-05	203 No. mugda para , Manikganj
1004	Badhon Paul	HR Manager	2019-03-11	303 No. Raysha Bazar, Puran Dhaka, Dhaka
1005	Nabila Prithul	Assistant Manager	2019-07-07	403 No. Road, House No. 503 , Mirpur-10, Dhaka

Verification & Validation:

1. Requirements of all needs of user are full filled.
2. Implementation are tested.
3. Final system are tested.
4. Maintainece for further errors and development are tested.

Maintenance:

As requrements evolve and bug reports come in from the field

- prioritize changes
- make changes (a kind of mini-s.d.p.)
- validate changes
  - new test cases
- validate that change does not break previously working code
  - regression testing

Experiment No. 4

Experiment Name: Write a program to make Result **Processing System** software with the help of database.

Objective: To write a program to make a result processing system and store result in database then display it.

Requirements:

Functional requirements:

1. User post student information.
2. User post student marks only.
3. Result of each marks will converted into grades.
4. Give a total CGPA.
5. Give a total CGPA grade.

Equipment/development tools

-a PC(Any Os installed) or MAC

-XAMPP server installed

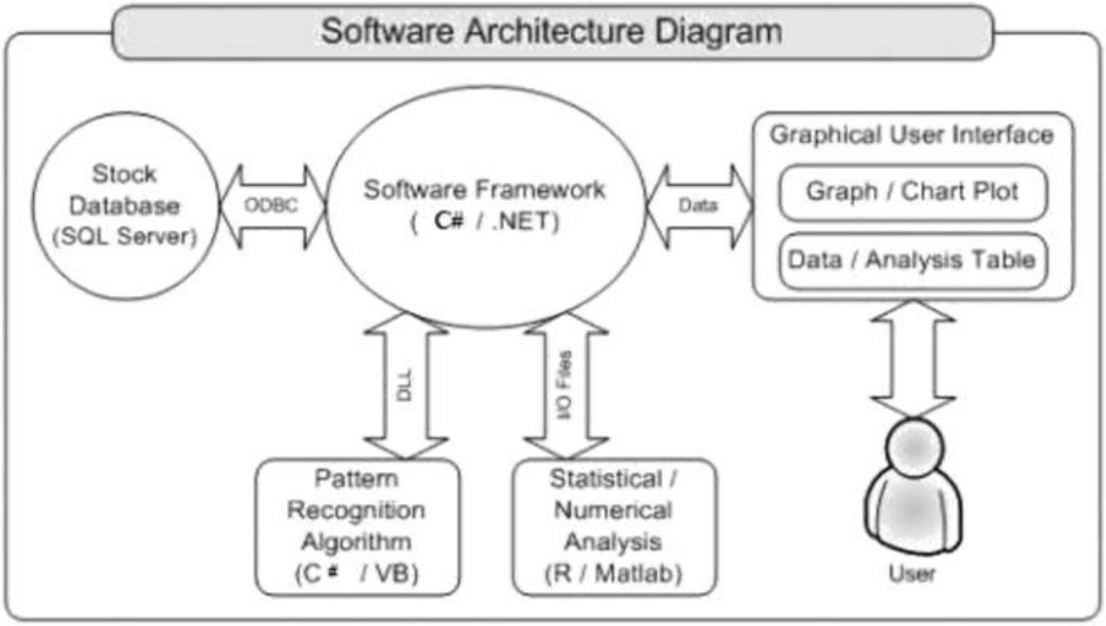
-Notepad++

-HTML,CSS,Javascipt

-Server language:PHP

-Database system: MySQL

Design:


Software Architecture Diagram
Stock Database (SQL Server)
ODBC
Software Framework (C# / .NET)
Data
Graphical User Interface
Graph / Chart Plot
Data / Analysis Table
DLL
I/O Files
Pattern Recognition Algorithm (C# / VB)
Statistical / Numerical Analysis (R / Matlab)
User

Implementation:

Procedure:

7. Turn on xampp and start Apache and MySQL
8. Go to C:\Xampp\htdocs, create a folder named result_processing and write the codes with Notepad++

9. Go to a browser and type localhost/phpmyadmin
10. Setup database
11. Go to a browser and type localhost/result_processing
12. Use the created system.

Sample code:

For display:

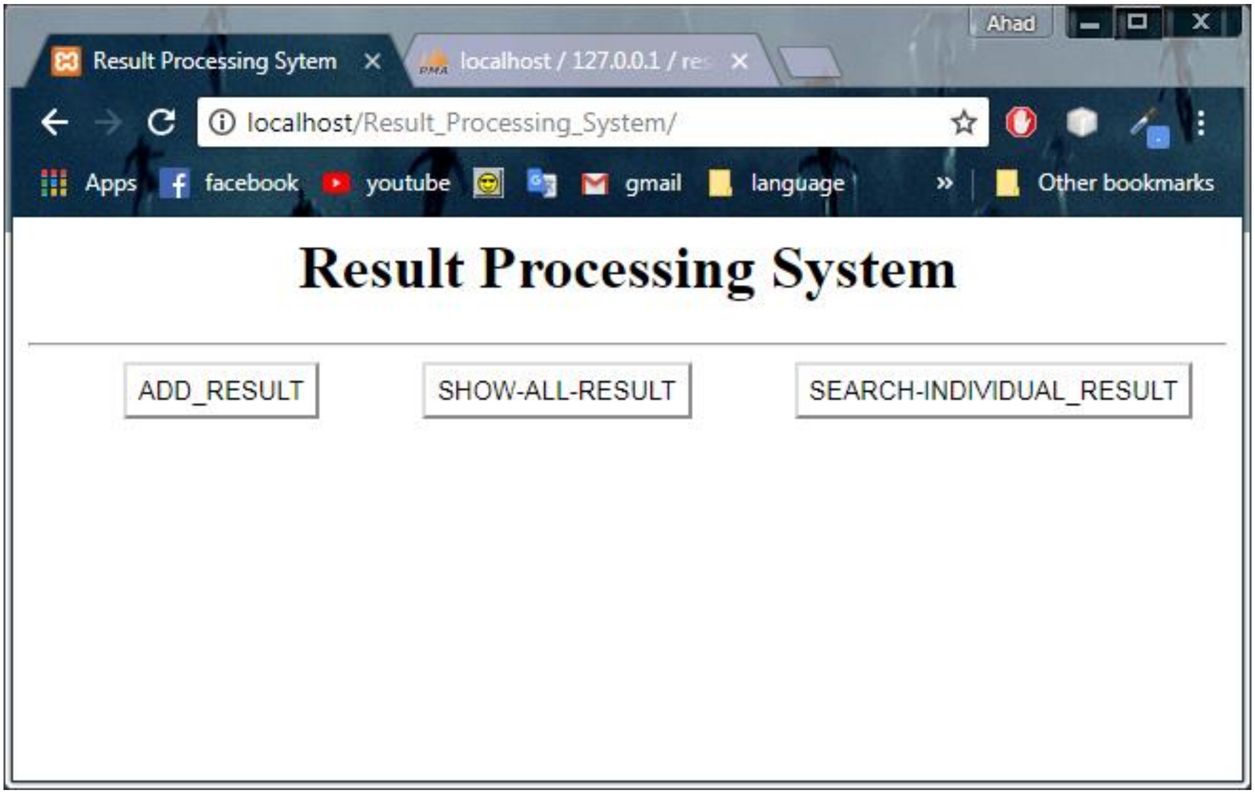
```
<?php include "db.php" ?>
<?php
 $query = "select * from Student_Result";
 $result = mysqli_query($connection, $query);
 if(!$result){
     die("Query Failed". mysqli_error());
 }
?>
<html>
<head>
    <title>Result Processing Sytem</title>
</head>
<body>
    <h2 align="center">Result_Processing_Sytem</h2>
    <table border="2" height"auto" width="500px" align="center">
        <tr>
            <th>Name</th>
            <th>Department Name</th>
            <th>Roll Number</th>
            <th>Semester</th>
            <th>Session</th>
            <th>First_Sub</th>
            <th>Second_Sub</th>
            <th>Third_sub</th>
            <th>Fourth_sub</th>
            <th>Fifth_sub</th>
            <th>Sixth_sub</th>
            <th>CGPA</th>
        </tr>
    <?php
        while($row = mysqli_fetch_assoc($result)){
    ?>
        <tr>
            <td align="center"><?php echo $row['Name']; ?></td>
            <td align="center"><?php echo $row['Department_Name']; ?></td>
            <td align="center"><?php echo $row['Roll']; ?></td>
            <td align="center"><?php echo$row['Semester']; ?></td>
            <td align="center"><?php echo $row['Session']; ?></td>
            <td align="center"><?php echo $row['1st_Subject']; ?></td>
            <td align="center"><?php echo $row['2nd_Subject']; ?></td>
            <td align="center"><?php echo$row['3rd_Subject']; ?></td>
            <td align="center"><?php echo $row['4th_Subject']; ?></td>
            <td align="center"><?php echo $row['5th_Subject']; ?></td>
            <td align="center"><?php echo $row['6th_Subject']; ?></td>
            <td align="center"><?php echo$row['Total_CGPA']; ?></td>
        </tr>
    <?php
        }
     ?>
     </table>
</body>
</html>
```

For Database connection:

```
<?php
    $connection = mysqli_connect("localhost", "root", "", "result");
    if(!$connection){
        die("Connection failed");
    }
 ?>
```

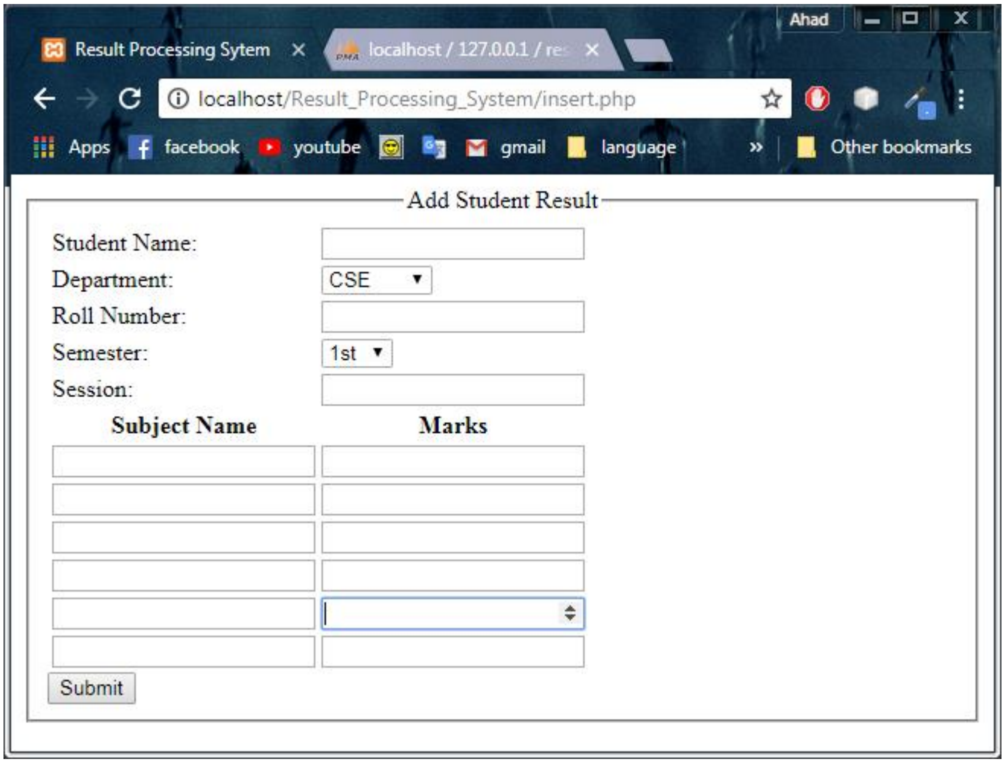
System snapshots:

1.



Result Processing System

2.



Add Student Result

Student Name:

Department: CSE

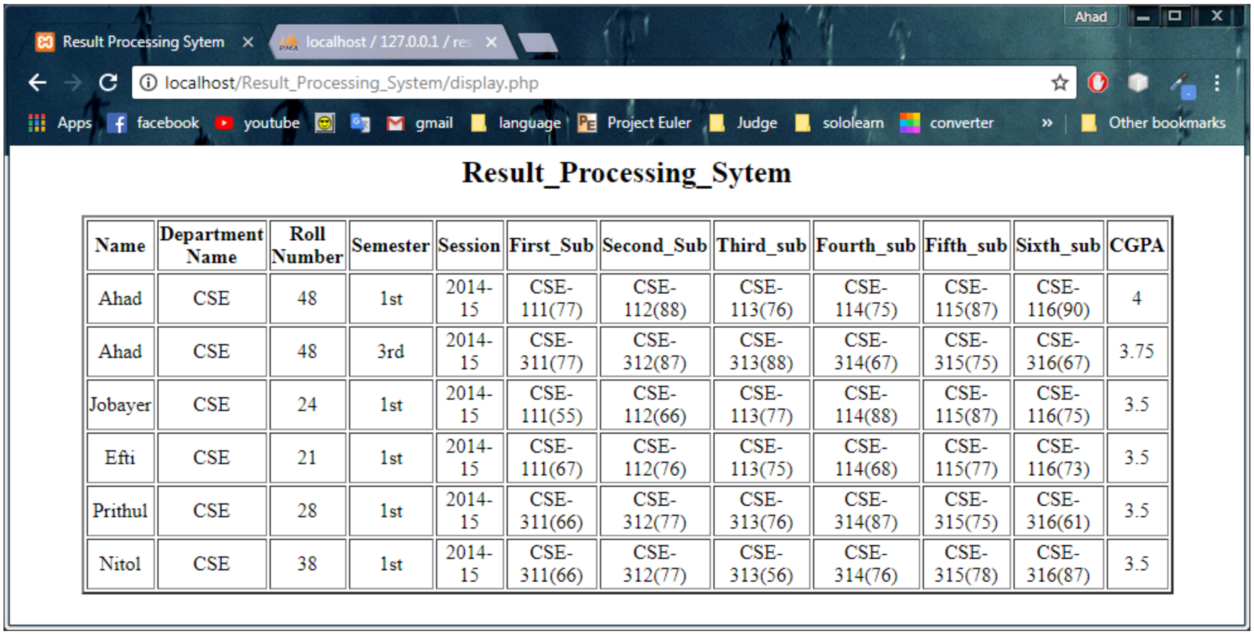
Roll Number:

Semester: 1st

Session:

Subject Name **Marks**

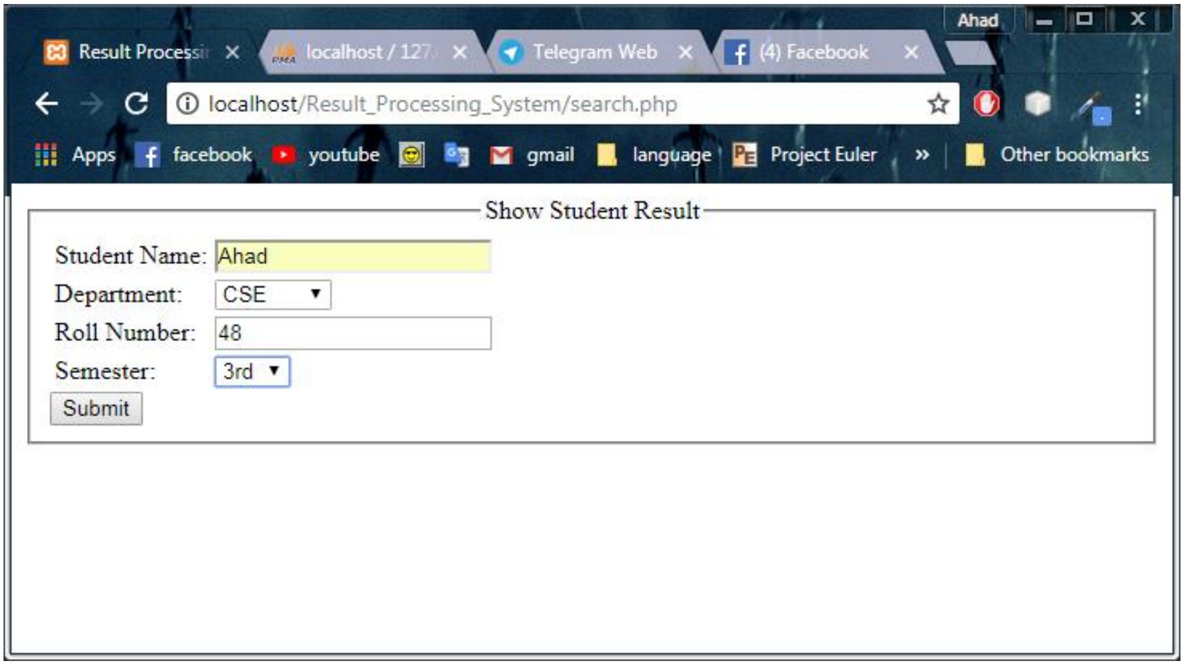
3.



Result_Processing_Sytem

Name	Department Name	Roll Number	Semester	Session	First_Sub	Second_Sub	Third_sub	Fourth_sub	Fifth_sub	Sixth_sub	CGPA
Ahad	CSE	48	1st	2014-15	CSE-111(77)	CSE-112(88)	CSE-113(76)	CSE-114(75)	CSE-115(87)	CSE-116(90)	4
Ahad	CSE	48	3rd	2014-15	CSE-311(77)	CSE-312(87)	CSE-313(88)	CSE-314(67)	CSE-315(75)	CSE-316(67)	3.75
Jobayer	CSE	24	1st	2014-15	CSE-111(55)	CSE-112(66)	CSE-113(77)	CSE-114(88)	CSE-115(87)	CSE-116(75)	3.5
Efti	CSE	21	1st	2014-15	CSE-111(67)	CSE-112(76)	CSE-113(75)	CSE-114(68)	CSE-115(77)	CSE-116(73)	3.5
Prithul	CSE	28	1st	2014-15	CSE-311(66)	CSE-312(77)	CSE-313(76)	CSE-314(87)	CSE-315(75)	CSE-316(61)	3.5
Nitol	CSE	38	1st	2014-15	CSE-311(66)	CSE-312(77)	CSE-313(56)	CSE-314(76)	CSE-315(78)	CSE-316(87)	3.5

4.

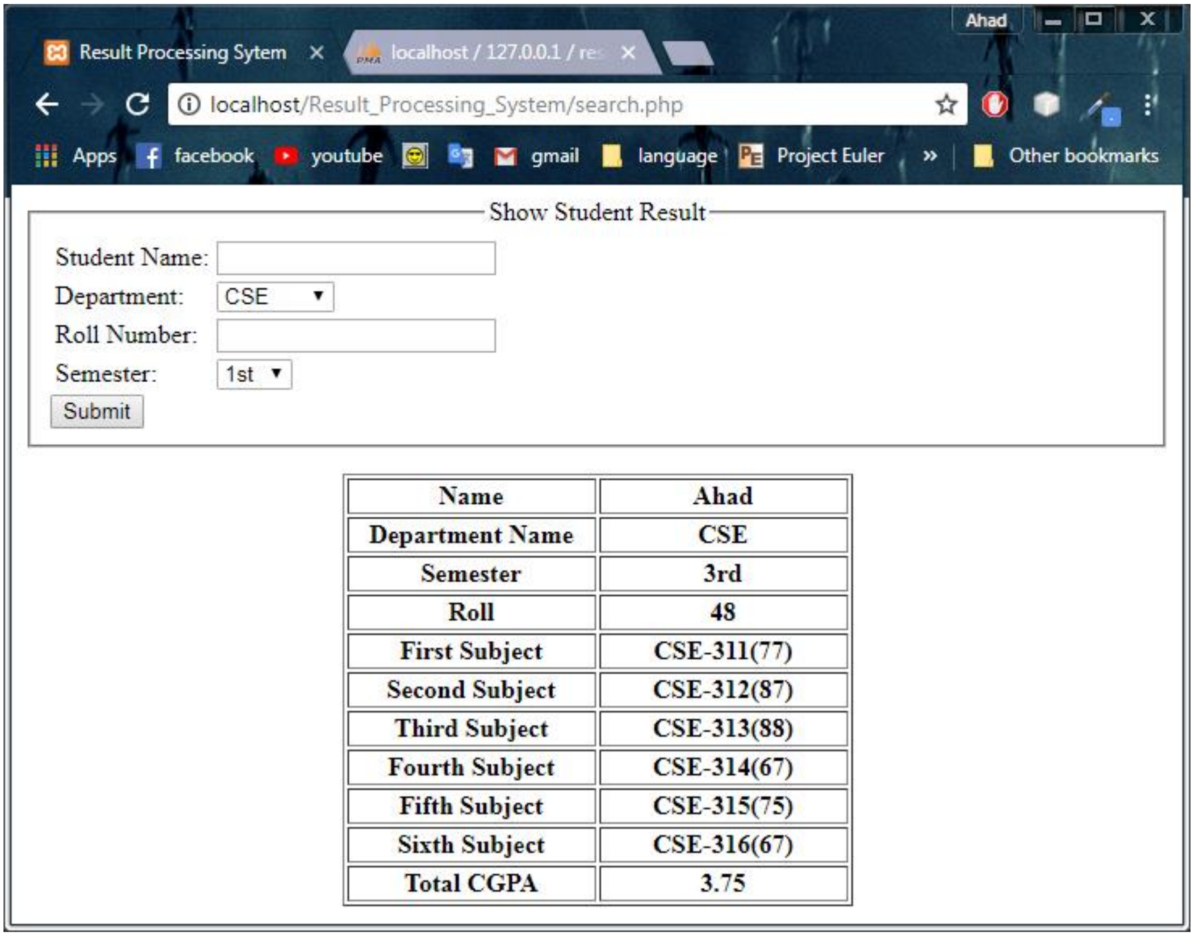


Show Student Result

Student Name:	Ahad
Department:	CSE
Roll Number:	48
Semester:	3rd

Submit

5.



Show Student Result

Student Name:	
Department:	CSE
Roll Number:	
Semester:	1st

Submit

Name	Ahad
Department Name	CSE
Semester	3rd
Roll	48
First Subject	CSE-311(77)
Second Subject	CSE-312(87)
Third Subject	CSE-313(88)
Fourth Subject	CSE-314(67)
Fifth Subject	CSE-315(75)
Sixth Subject	CSE-316(67)
Total CGPA	3.75

Verification & Validation:

1. Requirements of all needs of user are full filled.
2. Implementation is tested.
3. Final systems are tested.
4. Maintainece for further errors and development are tested.

Maintenance:

As requirements evolve and bug reports come in from the field

- prioritize changes
- make changes (a kind of mini-s.d.p.)
- validate changes
  - new test cases
- validate that change does not break previously working code
  - regression testing

Experiment No: 06

Experiment Name:Write a program to make a **Library Management System** software with the help of database.

Objective: To Write a program to make a Library Managementsystem and store Data in database then display it.

Requirements:

Functional requirements:

1. Books post information.
2. Book Author Information
3. Book Edition information

Equipment/development tools

-a PC(Any Os installed) or MAC

-XAMPP server installed

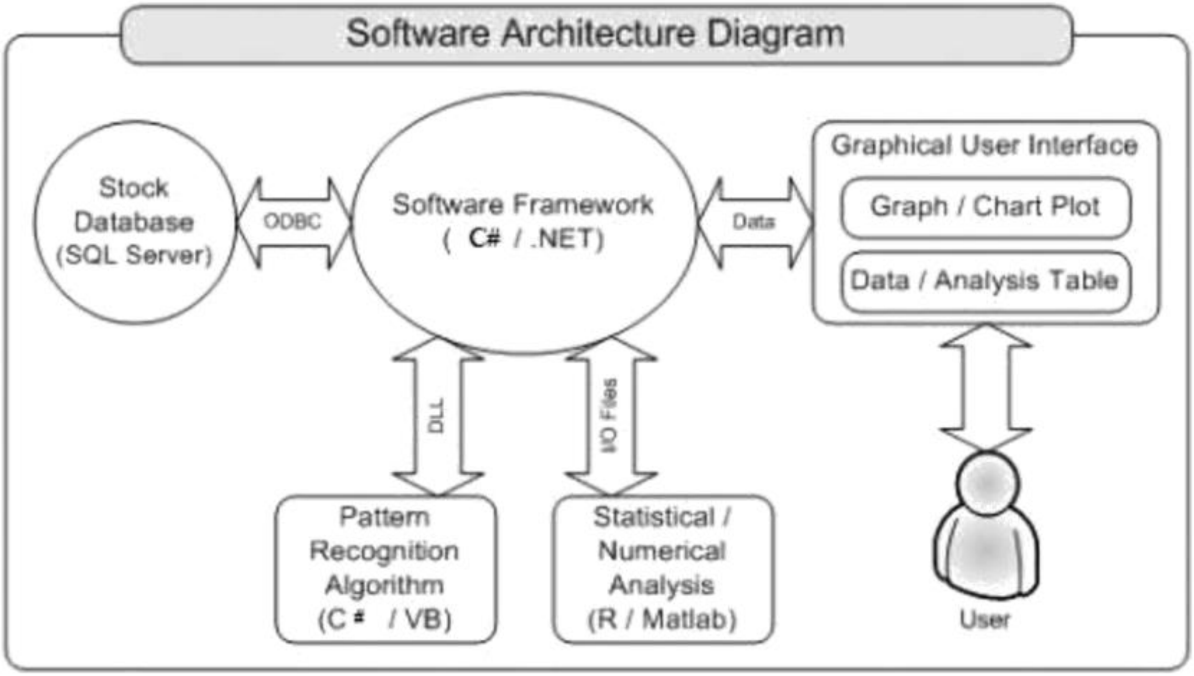
-Notepad++

-HTML,CSS,Javascipt

-Server language:PHP

-Database system: MySQL

Design:


Software Architecture Diagram
Stock Database (SQL Server)
ODBC
Software Framework (C# / .NET)
Data
Graphical User Interface
Graph / Chart Plot
Data / Analysis Table
DLL
I/O Files
Pattern Recognition Algorithm (C# / VB)
Statistical / Numerical Analysis (R / Matlab)
User

Implementation:

Procedure:

1. Turn on xampp and start Apache and MySQL
2. Go to C:\Xampp\htdocs, create a folder named library_management_system and write the codes with Notepad++
3. Go to a browser and type localhost/phpmyadmin

4. Setup database
5. Go to a browser and type localhost/library_management_system
6. Use the created system.

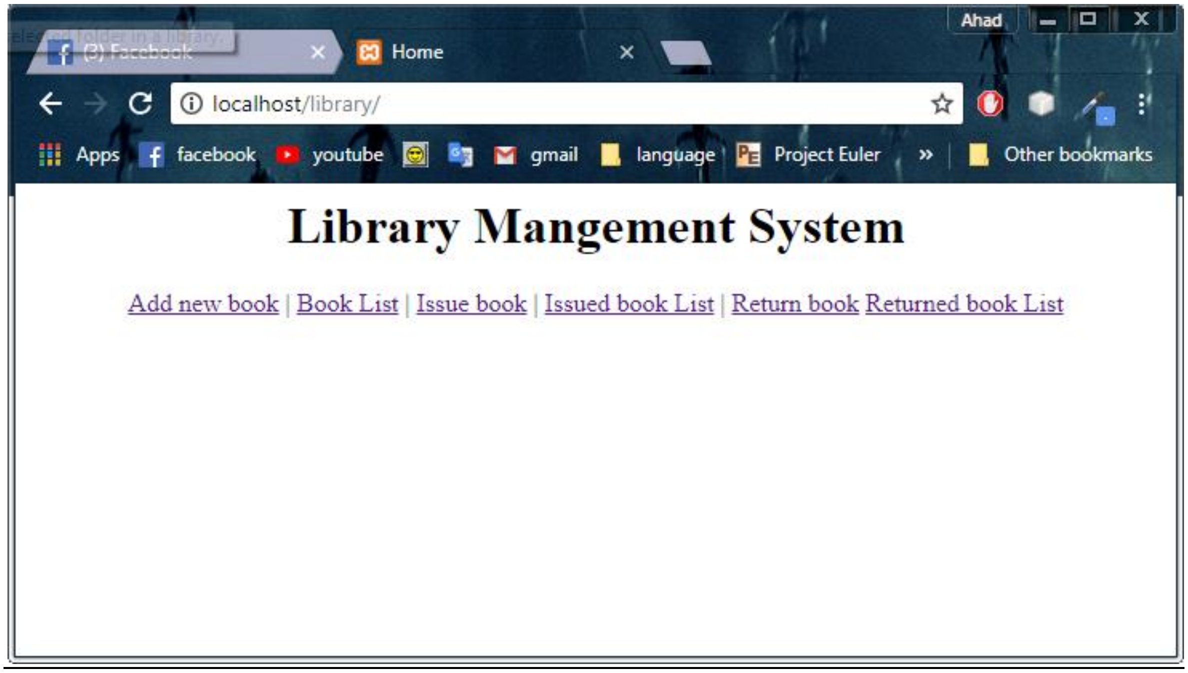
Sample code:

For adding new books:

```
<html>
	<head>
		<title>Add New Book </title>
	</head>
	<body>
		<h1 align="center">Add New Book </h1>

		<form method="post" action="save.php">
			<table>
			<tr>
				<td>Title: </td>
				<td><input type="text" name="title" /></td>
			</tr>

			<tr>
				<td>Writer: </td>
				<td><input type="text" name="writer" /></td>
			</tr>
			<tr>
				<td>Quantity: </td>
				<td><input type="text" name="quantity" /></td>
			</tr>
			<tr>
				<td></td>
				<td><input type="submit" name="submit" value="Submit" /></td>
			</tr>

		</form>
	</body>
</html>
```

System snapshots:

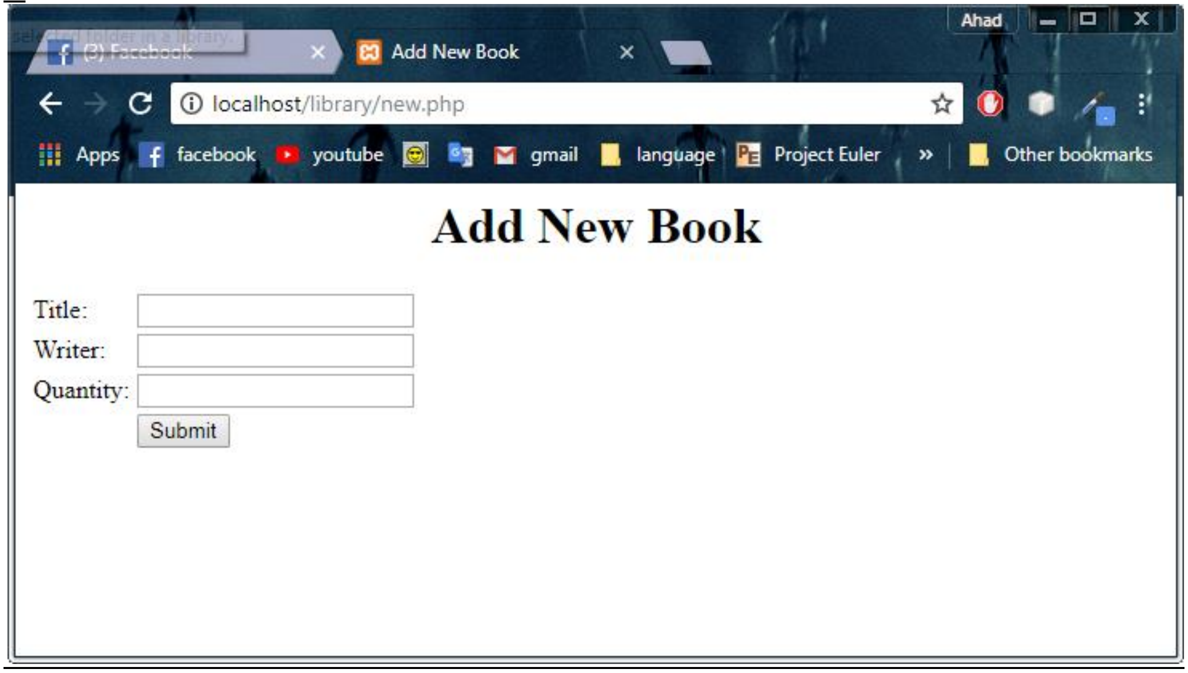
1.



Library Mangement System

Add new book | Book List | Issue book | Issued book List | Return book Returned book List

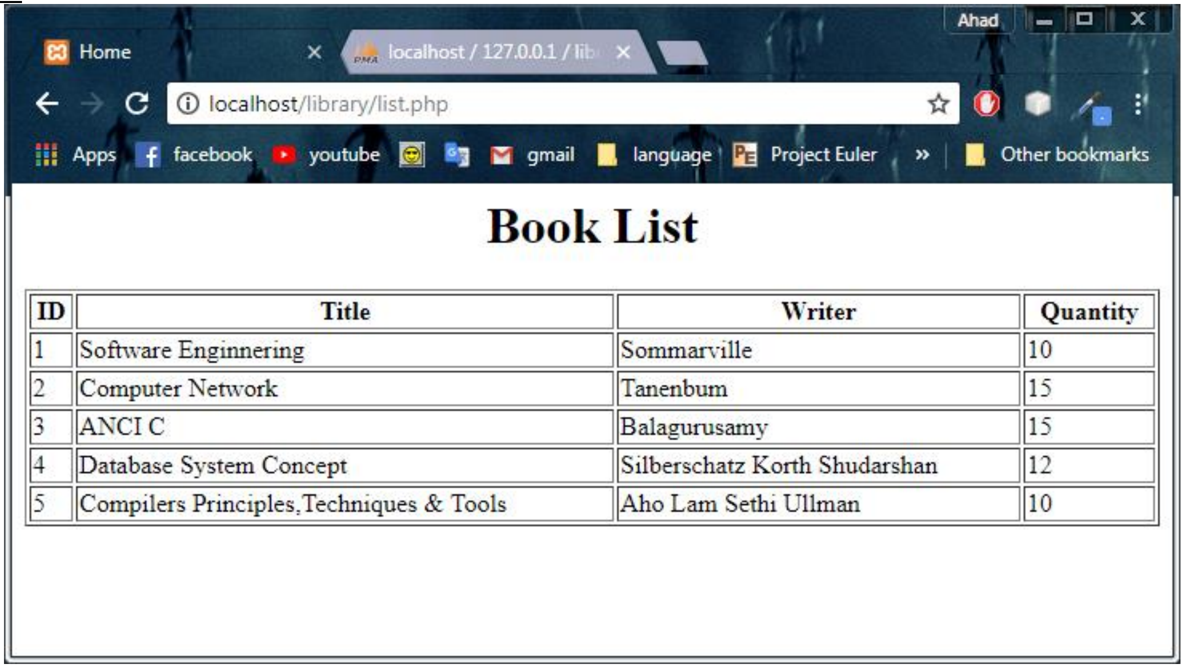
2.



Add New Book

Title:
Writer:
Quantity:
Submit

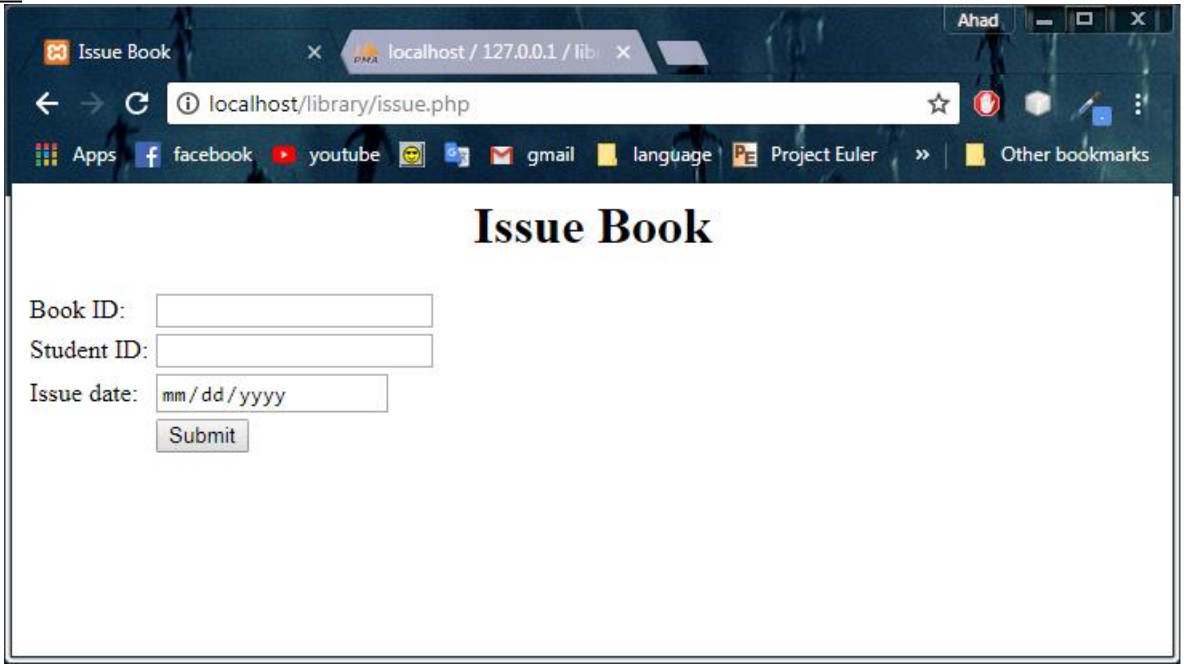
3.



Book List

ID	Title	Writer	Quantity
1	Software Enginnering	Sommarville	10
2	Computer Network	Tanenbum	15
3	ANCI C	Balagurusamy	15
4	Database System Concept	Silberschatz Korth Shudarshan	12
5	Compilers Principles,Techniques & Tools	Aho Lam Sethi Ullman	10

4.



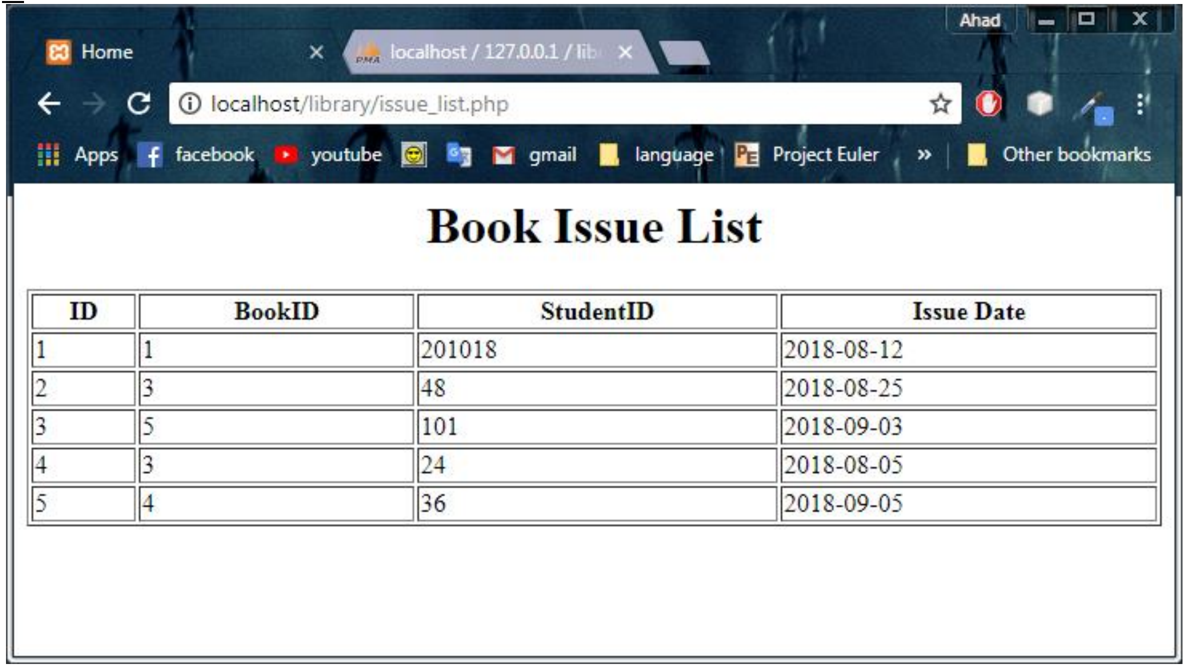
Issue Book

Book ID:

Student ID:

Issue date: mm/dd/yyyy

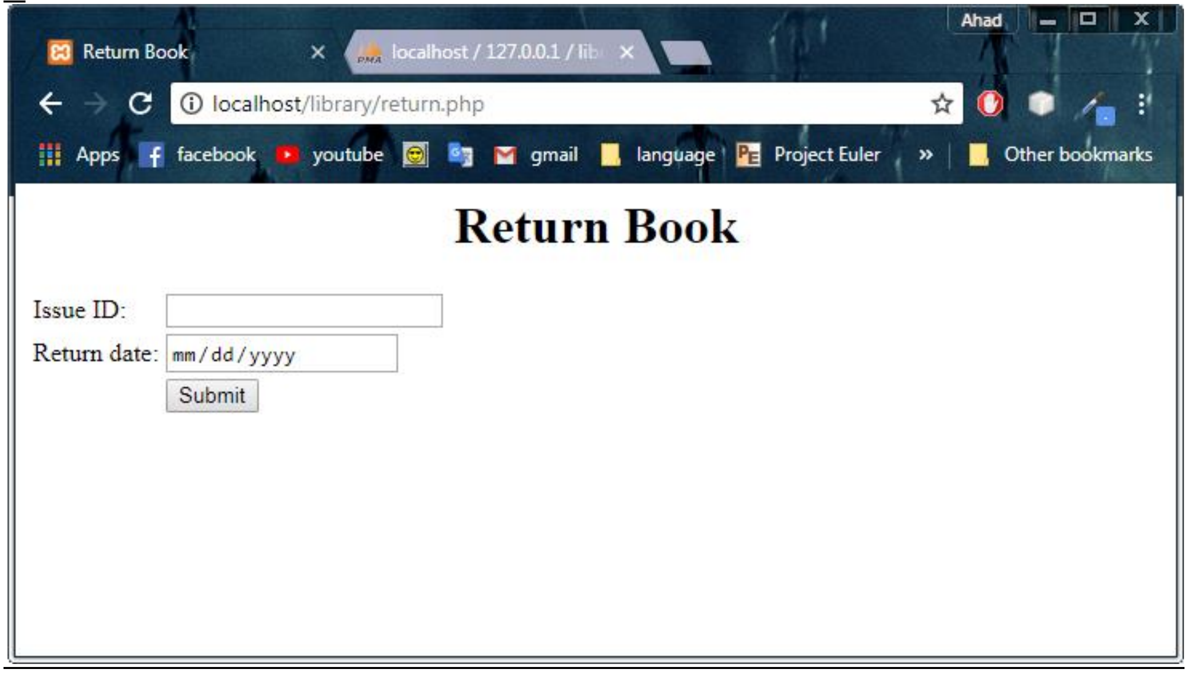
5.



Book Issue List

ID	BookID	StudentID	Issue Date
1	1	201018	2018-08-12
2	3	48	2018-08-25
3	5	101	2018-09-03
4	3	24	2018-08-05
5	4	36	2018-09-05

6.

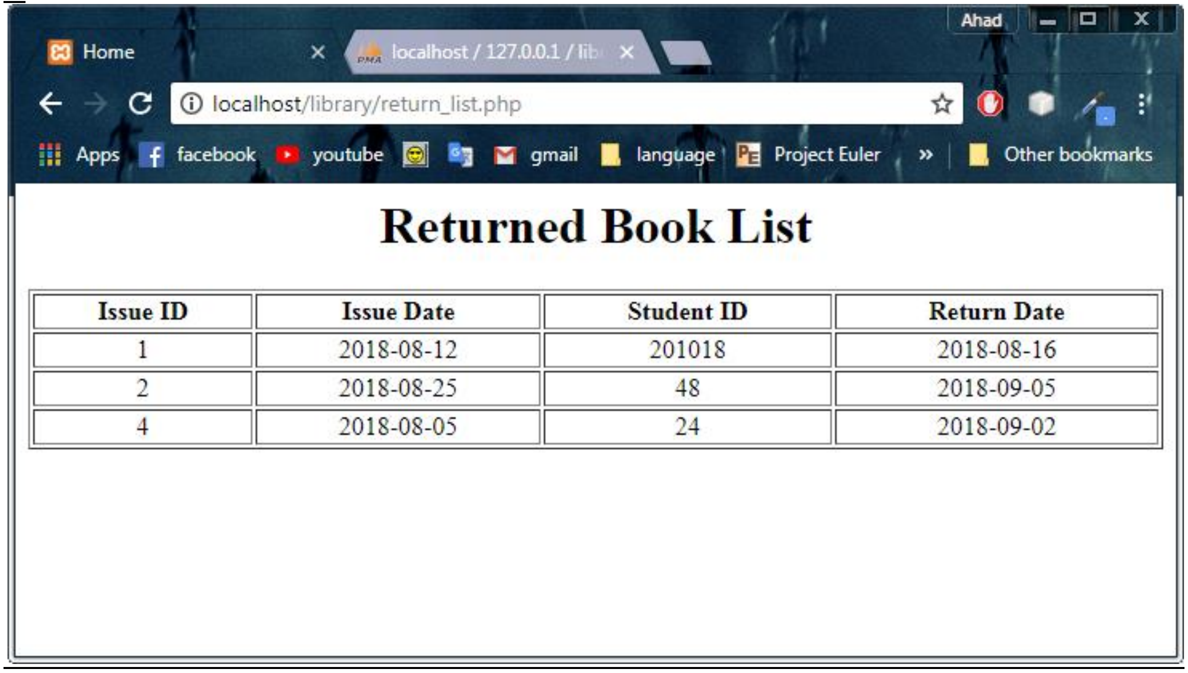


Return Book

Issue ID:

Return date: mm/dd/yyyy

7.



Returned Book List

Issue ID	Issue Date	Student ID	Return Date
1	2018-08-12	201018	2018-08-16
2	2018-08-25	48	2018-09-05
4	2018-08-05	24	2018-09-02

Verification & Validation:

1. Requirements of all needs of user are full filled.
2. Implementation are tested.
3. Final system are tested.
4. Maintainece for further errors and development are tested.

Maintenance: As requirements evolve and bug reports come in from the field

- prioritize changes
- make changes (a kind of mini-s.d.p.)
- validate changes
  - new test cases
- validate that change does not break previously working code
  - regression testing