

INDEX

HTML & CSS

- 01 Create a simple HTML page which demonstrates the use of the various types of lists.
- 02 Write HTML page named home.html to create a frameset with two vertical frames: the first frame is 250 pixels wide. Fill the first frame (*left_vertical*) with *links.html*. Second frame further divided into two horizontal frames (400px, 350px). Fill the Top frame (*right_top*) with *ice.html* and Bottom (*right_bottom*) with *it.html*.
- 03 Create a HTML document giving details of your [Name, Age], [Address, Phone] and [Register Number, Class] aligned in proper order using alignment attributes of Paragraph tag.
- 04 Create a web page for internal links; when the user clicks on different links on the webpage it should go to the appropriate locations/sections in the same page.
- 05 Write HTML & CSS code for following table.

A test table with merged cells

	Average		Red eyes
	height	weight	
Males	1.9	0.003	40%
Females	1.7	0.002	43%

- 06 Write HTML for the following registration page & use CSS to beautify it as your own choice.

Personal Details

Name:

E-mail:

Phone Number:

Password:

Gender: Male Female

JavaScript

- 07 Write JavaScript to validate the following fields of the Question 06 registration page.
 - i) Name (Name should contains alphabets and the length should not be less than 6 characters).
 - ii) E-mail (should not contain any invalid and must follow the standard pattern `name@domain.com`).
 - iii) Phone Number (Phone Number should contain 10 digits only).
 - iv) Password (Password should not be less than 6 characters length).
- 08 Write a JavaScript program to calculate multiplication and division of two numbers.
- 09 Write a JavaScript for loop that will iterate from 0 to 15. For each iteration, it will check if the current number is odd or even, and display a message to the screen.

PHP

- 10 Write a PHP program to calculate Electricity bill in single page.

Conditions:

For units less 50	- Taka. 3.50/unit
For units 51 to 100	- Taka. 4.00/unit
For units 101 to 200	- Taka. 5.20/unit
For units above 250	- Taka. 6.50/unit

- 11 Write a simple calculator program using PHP in single page.

Operations:

- Addition
- Subtraction
- Multiplication
- Division

- 12 A. Solve the following **Task-1** and **Task-2**.

Task-1: Create a database called **Student** in XAMPP MySQL.

Task-2: Create a table called **Semester_Reg** in the **Student** database having the structure as shown below.

Field name	Data type	Requirement
ID	Number/Text	Mandatory and primary key
Name	Text	Mandatory
Session	Text	Must follow the format like 2017-2018
Phone_No	Text	Optional
City	Text	For example Pabna
Gender	Text	Only (Male or Female)

B. Solve the following ✓ marked tasks.

Task 3: Insert some sample data into **Semester_Reg** table using PHP program.

Task 4: Write a PHP program to show the all records of **Semester_Reg** table.

Task 5: Delete single sample data from **Semester_Reg** table using PHP program.

Task 6: Update one sample data of **Semester_Reg** table using PHP program.

- 13 A. Solve the following **Task-1** and **Task-2**.

Task-1: Create a database called **Programmer** in XAMPP MySQL.

Task-2: Create a table called **Stu_Reg** in the **Programmer** database having the structure as shown below.

Field name	Data type	Requirement
ID	Varchar (30)	Mandatory and primary key
Name	Text	Optional
Image	Varchar(400)	Optional
Password	Number/ Varchar (20)	Mandatory

B. Solve the following ✓ marked tasks.

Task 3: Insert some sample data into **Stu_Reg** table including an encryption algorithm to secure the password.

Task 4: Write a PHP program to show the all records of **Stu_Reg** table.

Task 5: Delete single sample record from **Stu_Reg** table using PHP program.

Experiment No: 01

Name of the Experiment: Create a simple HTML page which demonstrates the use of various types of lists.

Objectives:

- (i) To understand different types of lists.
- (ii) To demonstrate HTML skills.
- (iii) To improve user experience and enhancing accessibility.

Theory: HTML provides three different types of list that can be used to organize and structure content on a web page.

1. Ordered list:

Ordered list are used to create numbered lists, where each list is sequentially numbered. The '``' tag is used to start an ordered list and each list item is markup with the list '``' tag.

2. Unordered list:

Unordered list are used to create bullet list, where each list item is marked with a bullet point or another symbol. The '``' tag is used to start an unordered list and each list item is marked up with the '``' tag.

3. Definition list:

Definition list are used to define terms and provide their respective definition. The '`<dl>`' tag is used to start

a definition list and each term is marked up with `<dt>` tag while the corresponding definition is marked up with the `<dd>` tag.

Source code:

```

<!DOCTYPE html>
<html>
  <head>
    <title> list types </title>
  </head>
  <body>
    <h1> List Types </h1>
    <h2> Unordered List </h2>
    <ul>
      <li> Item1 </li>
      <li> Item2 </li>
      <li> Item-3 </li>
    </ul>
    <h2> Ordered List </h2>
    <ol>
      <li> Item-1 </li>
      <li> Item-2 </li>
      <li> Item-3 </li>
    </ol>
    <h2> Definition List </h2>
    <dl>
      <dt> Term-1 </dt>
      <dd> Definition-1 </dd>
    
```

```
<dt> Term-2 </dt>
<dd> Definition-2 </dd>
<dt> Term-3 </dt>
<dd> Definition-3 </dd>

</div>
</body>
</html>
```

Output

List Types

Unordered List

- Item-1
- Item-2
- Item-3

Ordered List

1. Item-1
2. Item-2
3. Item-3

Definition List

Term-1 Definition-1

Term-2 Definition-2

Term-3 Definition-3

Experiment No: 2

Name of the Experiment: Write HTML code to create a frameset with two vertical frames: the first frame is 250 pixels wide. Fill the first frame (left-vertical) with links.html. Second frame further divided into two horizontal frames (400px, 350px). Fill the top frame (right-top) with ice.html and Bottom (right-bottom) with it.html

Objectives:

1. To divide a webpage into multiple sections.
2. To display multiple webpage simultaneously.
3. To reduce page loading time.
4. To maintain a consistent user interface.

Theory: HTML frames are used to divide our browser windows into multiple sections where each section can load a separate HTML document. A collection of frames in the browser window is known as a frameset. The window is divided into frames in a similar way the tables are organized into rows and columns.

Source code:

home.html :

```
<!DOCTYPE html>
<html>
  <head>
    <title> Frameset Example </title>
  </head>
```

```
<body>
  <frameset cols='250'>
    <frame src='links.html' name='left-vertical'>
    <frameset rows='400,350'>
      <frame src='ice.html' name='right-top'>
      <frame src='it.html' name='right-bottom' />
    </frameset>
  </frameset>
</body>
</html>

ice.html:
<!DOCTYPE html>
<html>
  <head>
    <title>Fram-1</title>
  </head>
  <body>
    <h4>This is ice.html</h4>
  </body>
</html>
```

links.html:

```
<!DOCTYPE html>
<html>
  <head>
    <title> Frame-2 </title>
  </head>
  <body>
    <h4> This is links.html </h4>
  </body>
</html>
```

it.html:

```
<!DOCTYPE html>
<html>
  <head> <title> Frame-3 </title> </head>
  <body>
    <h4> This is it.html </h4>
  </body>
</html>
```

Output:

This is links.html

This is it.html

This is it.html

Experiment No: 03

Name of the experiment: Create a HTML document giving details of your [name, age], [Address, Phone] and [Register Number, class] aligned in proper order using alignment attributes of paragraph tag.

Objectives:

1. To practice creating basic HTML documents.
2. To learn about using paragraph tags and their alignment.
3. To understand the importance of properly formating content.
4. To practice and improve web development.

Theory HTML allows variety of alignment attribute that can be control the positioning and layout of content on a web page. HTML attributes provides additional information about HTML elements. All HTML elements can have attributes. Attributes provide additional information about elements. Attributes are always specified in the starting tag.

The basic syntax for adding an alignment attribute to an HTML element is as follow:

<element align = 'value'> Content </element>

For example, to center a paragraph, we can add the align='center' attribute to the opening <p> tag.

<p align='center'> This is a centered paragraph </p>

Source code:

```

<!DOCTYPE html>
<html>
  <head>
    <title> Personal Details </title>
  </head>
  <body>
    <h1> Personal Details </h1>
    <p align = 'center'> Name : MD. RAHATUL RABBI </p>
    <p align = 'center'> Age : 23 </p>
    <p align = 'center'> Address : Lalmonirhat, Rangpur </p>
    <p align = 'center'> Phone : 01521539767 </p>
    <p align = 'center'> Register Number : 190609 </p>
    <p align = 'center'> Class : 3rd year 1st semester </p>
  </body>
</html>

```

Output:

Name : MD. RAHATUL RABBI

Age : 23

Address : Lalmonirhat, Rangpur

Phone : 01521539767

Register Number : 190609

Class : 3rd year 1st semester

Experiment No : 04

Name of the Experiment: Create a web page for internal links, when the user clicks on different link on the webpage it should go to the appropriate location/sections in the same page.

Objectives:

1. To use improve user engagement.
2. To improve usability.
3. To simplify navigation .

Theory HTML internal link is linked within same web page.

This link can be absolute path or relative path. HTML internal link name is followed by the hash (#) sign. we have to assign an id to refer section for our page, which is referred to as an internal link to the same page. Internal links help users to navigate through a webpage easily and quickly.

Source Code:

```

<!DOCTYPE html>
<html>
  <head> <title> Internals links Example </title> </head>
  <body>
    <h1> Table of contents </h1>
    <ul>
      <li> <a href = "#section1"> Section-1 </a> </li>
      <li> <a href = "#section2"> Section-2 </a> </li>
      <li> <a href = "#section3"> Section-3 </a> </li>
    </ul>
    <hr>
    <h2 id = "section1"> Section-1 </h2>
    <p> This is Section-1 </p>
    <a href = "#top"> Back to top </a>
    <hr>
    <h2 id = "section2"> Section-2 </h2>
    <p> This is Section-2 </p>
    <a href = "#top"> Back to top </a>
    <hr>
    <h2 id = "section3"> Section-3 </h2>
    <p> This is Section-3 </p>
    <a href = "#top"> Back to top </a>
    <hr>
    <p id = "top"> <a href = "#top"> Back to top </a> </p>
  </body>
</html>

```

Output :

Table of contents

- Section - 1
- Section - 2
- Section - 3

Section - 1

This is Section - 1

Back to top

Section - 2

This is Section - 2

Back to top

Section - 3

This is Section - 3

Back to top

— - —

Experiment No: 05

Name of the Experiment: Write HTML and CSS code for following table-

A test table with merged cell

	Average		Red eyes
	Height	Weight	
Males	1.9	0.003	40%
Females	1.7	0.002	43%

Objectives:

- To design a table effectively.
- To present data in a clear and organised way.

Theory :- To create an HTML table with merged cells, we can use the colspan and rowspan attribute in our html code. The colspan attribute specifies the number of columns that a cell should span while the rowspan attribute specifies the number of rows that a cell should span.

Source code:

```
<!DOCTYPE html>
<html>
  <head> <title> Test Table </title> </head>
  <style>
    table, th, td {
      border: 1px solid black;
      border-collapse: collapse;
      padding: 10px;
      text-align: center;
    }
    th {
      font-weight: bold;
      vertical-align: middle;
    }
    th {
      vertical-align: top;
    }
  </style>
  <body>
    <table>
      <tr>
        <th rowspan='2'> </th>
        <th colspan='2'> Average </th>
      <tr>
        <th rowspan='2'> Read Eyes </th>
      </tr>
```

<tr>

<th> Height </th>

<th> Weight </th>

</tr>

<tr>

<th> Males </th>

<td> 1.9 </td>

<td> 0.003 </td>

<td> 40% </td>

</tr>

<tr>

<th> Females </th>

<td> 1.7 </td>

<td> 0.002 </td>

<td> 43% </td>

</tr>

</table>

</body>

</html>

Output:

A test table with merged cell

	Average		Red eyes
	Height	Weight	
Males	1.9	0.003	40%
Females	1.7	0.002	43%

Experiment no: 06

Name of the experiment: Write HTML for the following registration page & use CSS to beautify it as your own choice.

Personal Details

Name :

Email :

Phone No :

Password :

Gender :

 Male

 Female

Objectives:

1. To learn about HTML Form, and CSS style.
2. To design a registration form using CSS.

Theory: A registration form is a common component of many websites that allows user to create accounts and access specific content or services. A well designed registration page can attract and retain users, while a poorly design page can turn them away. HTML and CSS are two programming languages that can be used to create a visually appealing and functional registration page.

Source code:

```

<!DOCTYPE html>
<html>
  <head>
    <title> Registration Form </title>
    <style>
      * {
        color: white;
      }
      body {
        background-color: #e7dadb;
      }
      h1 {
        text-align: center; margin-top: 10px;
      }
      form {
        width: 500px; margin: auto; background: gray;
        padding: 20px;
      }
      label {
        display: block; margin-bottom: 10px; font-weight:
        bold; font-size: 18px;
      }
      .form-control > input {
        width: 100%; padding: 10px;
        border-radius: 3px; background-color: #999999;
      }
      .gender > input, .gender > label {
        display: inline;
      }
      .form-control > .submit {
        background-color: #4CAF50;
        padding: 10px 20px; border: none; cursor: pointer;
      }
      .form-control > .submit:hover {
        background-color: #3e8e41;
      }
    </style>
  </head> </html>

```

```

<body>
  <form action="" method="post">
    <h1> Personal Details </h1>
    <div>class = 'form-control'>
      <label for="name"> Name : </label>
      <input type="text" id="name" />
    </div>
    <div class = 'form-control'>
      <label for="email" id="email">
        </div>
      <div>
        <label for="phone"> Phone Number: </label>
        <input type="password" id="password" />
      </div>
    <div class = 'gender'>
      <label for="gender"> Gender : </label>
      <span style = "margin-left: 20px;">
        <input type = "radio" value = "Male" name = "gender" checked> Male
        <input type = "radio" value = "Female" name = "gender" /> Female
      </span>
    </div>
    <div class = 'form-control'>
      <input type = "submit" class = "submit" value = "Submit" />
    </div>
  </form> </body> </html>

```

Output:

Personal Details	
Name:	MD. RAHATUL RABBI
Email:	rahatul@gmail.com
Phone Number:	0152153976
Password:	*****
Gender :	<input checked="" type="radio"/> Male <input type="radio"/> Female
<input type="button" value="Submit"/>	

— o —

Experiment No: 07

Name of the Experiment: Write JavaScript to validate the following fields of the question of registration page.

- (i) Name (Name should contains alphabets and the length should not be less than 6 characters).
- (ii) E-mail (Should not contain any invalid and must follow the standard pattern name@domain.com).
- (iii) Phone Number (Phone Number should contain 10 digits only)
- (iv) Password (Password should not be less than 6 characters length)

Objectives:

- (i) To validate the name, email, phone number and password.
- (ii) To understand the javascript validation operation.

Theory:

Validating Name field: The name field is usually a text field that allows users to enter their name. Here are some basic validation rules we can apply to name field:

- Check if the field is empty.
- Check if the name field contains only alphabetic

characters.

- Check if the name field has a minimum and maximum length.

Validating password field:

Here are some basic validation rules we can apply to the password field.

- Check if the field is empty.
- Check, if the password has a minimum length
- Check, if the password contains at least one special character.

Validating Email Field:

- (i) Check, if the field is empty.
- (ii) Check if the email address is in the correct format.
- (iii) Check if the email address contains a valid domain name.

Validating Mobile Number:

- Check if the field is empty.
- Check if the resulting string contains only digits.
- Check if the resulting string has a specific length depending on the country.

Source code:

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Registration Form</title>
<link rel="stylesheet" href="style.css">
</head>
<body>
<form id="registration-form" action="" name="login" method="post">
<h1>Personal Details</h1>
<div class="form-control">
<label for="name">Name:</label>
<input type="text" id="name" name="name" required>
<i id="errorname"></i>
</div>
<div class="form-control">
<label for="email">E-mail:</label>
<input type="email" id="email" name="email" required>
<i id="erroremail"></i>
</div>
```

```
<div class="form-control">  
    <label for="phone">Phone Number:</label>  
    <input type="tel" id="phone" name="phone" required>  
    <i id="errorphone"></i>  
</div>  
  
<div class="form-control">  
    <label for="password">Password:</label>  
    <input type="password" id="password" name="password" required>  
    <i id="errorpassword"></i>  
</div>  
  
<div class="gender">  
    <label for="gender">Gender:</label>  
    <span style="margin-left: 20px;">  
        <input type="radio" value="Male" name='gender' id='male' checked> Male  
        <input type="radio" value="Female" name='gender' id='female'> Female  
    </span>  
</div>  
  
<br><br>  
  
<div class="form-control">  
    <input type="submit" class="submit" value="Submit">  
</div>  
  
</form>  
  
<div class="result" id="show_result"></div>  
  
<script src="script.js"></script>  
  
</body>  
</html>
```

Style.css

```
* { color: white; }

body { background-color: #e7dbdb; }

h1 { text-align: center; margin-top: 10px; }

form { width: 500px; margin: auto; background-color: gray; padding: 20px; border-radius: 5px; box-shadow: 0 0 10px #999999; }

label { display: block; margin-bottom: 10px; font-weight: bolder; font-size: 18px; }

.form-control>input { width: 100%; padding: 10px; border-radius: 3px; border: 1px solid #999999; margin-bottom: 20px; box-sizing: border-box; background-color: #999999; }

.gender>input,
.gender>label { display: inline; }

.form-control>.submit { background-color: #4CAF50; padding: 10px 20px; border: none; border-radius: 3px; cursor: pointer; }

.form-control>.submit:hover { background-color: #3e8e41; }

i { color: red; }

.result { text-align: center; background-color: black; }
```

script.js

```
// Get form inputs

const nameInput = document.getElementById("name");

const emailInput = document.getElementById("email");

const phoneInput = document.getElementById("phone");

const passwordInput = document.getElementById("password");

const maleInput = document.getElementById("male");
```

```

const username = document.getElementById("username");
const password = document.getElementById("password");
const email = document.getElementById("email");
const phone = document.getElementById("phone");

// Define validation functions

function validateName(name) {
    const regex = /^[a-zA-Z][a-zA-Z]{5,}$/,
        result = regex.test(name) && name.length >= 6;
    return result;
}

function validateEmail(email) {
    const regex = /^[a-zA-Z]+([a-zA-Z]+[a-zA-Z]+)*@[a-zA-Z]+\.[a-zA-Z]{2,}$/,
        result = regex.test(email);
    return result;
}

function validatePhone(phone) {
    if (phone.length === 0)
}

// Define form submit event listener
const form = document.getElementById("registration-form");
form.addEventListener("submit", function(event) {
    event.preventDefault() // Prevent form submission
});

```

Experiment No: 08

Name of the Experiment: Write a javaScript program to calculate multiplication and division of two numbers.

Objectives:

- (i) To understand basic arithmetic operations.
- (ii) To enhance user experience.
- (iii) To add functionality to our website.

Theory:

Multiplication: Let two variables are 'var x' and 'var y' and the output is 'var z'. Then the multiplication of two variables will be,

$$\text{var } z = \text{var } x * \text{var } y ;$$

Division: Let two variables are 'var x' and 'var y' and the output is 'var z'. Then the division of two variables will be,

$$\text{var } z = \text{var } x / \text{var } y ;$$

Experiment No: 09

Goal of the Experiment: Write a JavaScript for loop that will iterate from 0 to 15. For each iteration, it will check if the current number is odd or even, and display a message to the screen.

Objectives:

- (i) To understand the basic structure of for loop in JavaScript.
- (ii) To understand about iteration.
- (iii) To understand even and odd numbers.
- (iv) To enhance user experience.

Theory: In JavaScript, a for loop is used to iterate over a range of values. The for loop consists of an initialization statement, a condition and an update statement. The loop will continue to execute as long as the condition is true.

In this program, we will use for loop to iterate from 0 to 15 and check if each number is odd or even using module operator (%).

else {

document.write ("x + u is odd");

document.write ("
");

}

}

document.getElementById ("for-loop").innerHTML = "For loop from
+ first + " + to + last;

Output :

Enter for loop starting value :

Enter for loop ending value :

For loop from 0 to 15

0 is even

11 is odd

1 is odd

12 is even

2 is even

13 is odd

3 is odd

14 is even

4 is even

15 is odd

5 is odd

6 is even

7 is odd

8 is even

9 is odd

10 is even

— o —

Experiment NO: 10

Name of the Experiment: Write a PHP program to calculate electricity bill in single page.

Conditions:

- For unit less 50 - Taka 3.50 /unit.
- For unit 51 to 100 - Taka 4.00 /unit.
- For unit 101 to 200 - Taka 5.20 /unit
- For units above 250 - Taka 6.50 /unit

Objectives:

- (i) To accurately calculate the bill amount.
- (ii) To automate the billing process.
- (iii) To improve customer service.

Theory Preparing an electricity bill in PHP involves a set of processes and calculations to determine the accurate bill amount based on the consumption of electricity by a customer. First it collect user inputs. Then calculate the electricity consumption, calculate total amount payable and display the electricity bill.

PHP Source code:

<?php

\$result_str = \$result = '';

Experiment No: 11

Name of the experiment: Write a simple calculator program using PHP in single page.

Operations:

- Addition • multiplication
- Subtraction • Division

Objectives:

1. To understand basic arithmetic operations.
2. To enhance user experience.
3. To add functionality to our website.

Theory:

Addition: Let two variables are $\$x$ and $\$y$ and the output is $\$z$. Then addition of two variables will be, $\$z = \$x + \$y$;

Subtraction: If $\$x$ and $\$y$ are two variables then subtraction of these two variables $\$z$ will be, $\$z = \$x - \$y$;

Multiplication: The output of division of two numeric variable will be, $\$z = \$x * \$y$;

Division: The output of division of two numeric variable will be, $\$z = \$x / \$y$;

Experiment NO: 12

Name of the Experiment: Solve the following task1 and task2 -

Task1: Create a database called student in xampp MySQL.

Task2: Create a table called Semester_Reg in the student database having the structure as shown below:

Field Name	Data type	Requirement
ID	Number/Text	Mandatory and primary key
Name	Text	Mandatory
Session	Text	Must follow the format like 2018-19
Phone-No	Number	Optional.
City	Text	For example Lalmorirhat
Gender	Text	Only (male or female)

Task3: Insert some simple data into semester_Reg table using PHP program.

Task4: Write a PHP program to show the all records of Semester Reg table.

Task5: Delete single sample data from semester_Reg table using PHP program.

Task6: Update one sample data of Semester_Reg table using PHP program.

Objectives :

- (i) To learn how to create a database in xAMPP MySQL.
- (ii) To learn how to create a table in database
- (iii) To learn how to update, delete and insert in database using PHP.

Theory :

MySQL is an open source relational database management system. It is commonly used in web application to store and manage data.

PHP is a popular server side scripting language used to create dynamic web page. The combination of MySQL and PHP provides an efficient way to manage data in web application.

Output:

Student's Registration Form

ID : 190609Name : MD. RAHATULSession : 2018-19Phone Number: 01521539767City : LalmonirhatGender : Male FemaleInsertDeleteUpdateShow

N.B:

1. To delete a record please enter your ID, Name and Session.
2. You can update all information except ID Number.
3. To show all record enter your ID, Name and Session



Experiment No: 13

Name of the Experiment: Solve the following Task1 and Task2-

Task-1: Create a database called programmer in xampp MySQL.

Task-2: Create a table called stu-reg in programmer database having the structure as shown below:

Field Name	Data Type	Requirement
ID	Varchar(30)	Mandatory and Primary key
Name	Text	Optional
Image	Varchar(400)	Optional
Password	Number/ Varchar(20)	Mandatory

Task-3: Insert some sample data into stu-reg table including an encryption algorithm to secure the password.

Task-4: Write a PHP program to show the all records of stu-reg table.

Task-5: Delete single sample record from stu-reg table using PHP program.

Objectives:

- (i) To learn how to create a database in XAMPP MySQL.
- (ii) To learn how to create a table in database.
- (iii) To learn how to insert data in table and also to learn encryption algorithm to secure the password.

Theory:

XAMPP is an open source software package that provides a local server environment for developing and testing web application. It includes Apache, MySQL, PHP and perl. MySQL is a popular open source relational database management system used to manage and store data.

Output

ID	Name	Image
190609	RAHATUL	

Programmer Registration Form

ID:

Name:

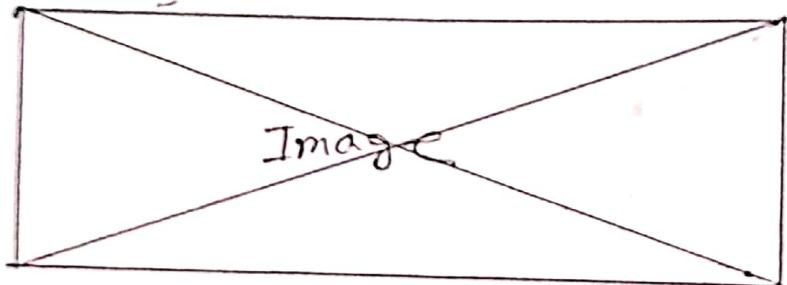


Image : No file chosen

Password

N.B:

1. To delete a record enter your ID and Password
2. To show all records enter your ID and Password