SILVER OAK UNIVERSITY
EDUCATION TO INNOVATION

NAME	:	HARISH KUMAR
EN. NO.	:	2202030400043
SUBJECT	:	OJT PRACTICALS(INTERNSHIP)
COURSE	:	B.TECH(CE)

C and C++ Practicals

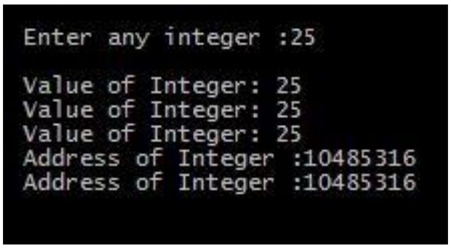
Practical-1

Aim: Write a C program to print the address of a variable using a pointer.

Code:

```
   # include < stdio.h >

   int  main( )
   {

int  a ;
int  *p ;
printf(" Enter any integer: ") ;
scanf("%d ",& a) ;
p = &a ;

printf("\n Value of Integer : %d ",a) ;
printf("\n Value of Integer : %d ",*p) ;
printf("\n Value of Integer : %d ",*(&a)) ;
printf("\n Address of Integer : %u ",p) ;
printf("\n Address of Integer : %u ",&a) ;
return ( 0 );

   }
```

Output:



```
Enter any integer :25

Value of Integer: 25
Value of Integer: 25
Value of Integer: 25
Address of Integer :10485316
Address of Integer :10485316
```

Practical-2

Aim: Write a C program to create a Calculator using a pointer.

Code:

```
 #include<stdio.h>
int main()
{
 float num1, num2, result;
 char op;
 float *ptr1 = &num1; float *ptr2 = &num2;
printf("Enter the first number: ");
scanf("%f", ptr1);


printf("Enter the operator (+, -, *, /): ");
scanf(" %c", &op);
 printf("Enter the second number: ");
scanf("%f", ptr2);

switch(op) {
case '+': result = *ptr1 + *ptr2;
 break;
case '-': result = *ptr1 - *ptr2;
 break;
case '*': result = *ptr1 * *ptr2;
 break;

case '/':
result = *ptr1 / *ptr2;
break;
 default:
 printf("Invalid operator");
return 1;
}


printf("%.2f %c %.2f = %.2f", *ptr1, op, *ptr2, result);
return 0;
 }
```

Output:

```
Enter the first number: 12

Enter the operator (+, -, *, /): *

Enter the second number: 12

 12.00 * 12.00 = 144.00
```

Practical-3

Aim: Write a C program to swap the two values using call by value and call by reference.

Code:

```
#include<stdio.h>

 void swap_value(int x, int y)

{

int temp = x;

 x = y;

 y = temp;

}

void swap_reference(int *x, int *y)

 {

 int temp = *x;

*x = *y;

 *y = temp;

}

int main()
```

```
 {
int num1, num2;
printf("Enter the Number: \n");
scanf("%d",&num1);
scanf("%d",&num2);
printf("\nBefore swapping using call by value: num1 = %d, num2 = %d\n", num1, num2);
swap_value(num1, num2); printf("After swapping using call by value: num1 = %d, num2 = %d\n\n", num1, num2);
printf("Before swapping using call by reference: num1 = %d, num2 = %d\n", num1, num2);
swap_reference(&num1, &num2); printf("After swapping using call by reference: num1 = %d, num2 = %d\n", num1, num2);
 return 0;
}
```

Output:

```
Enter the Number: 12
23
Before swapping using call by value: num1 = 12, num2 = 23
After swapping using call by value: num1 = 12, num2 = 23
Before swapping using call by reference: num1 = 12, num2 = 23
 After swapping using call by reference: num1 = 23, num2 = 12
```

Practical-4

Aim: Define a structure type struct personal that would contain person name, Date of birth and age using this. Structure to read this information of 4 people and display the same.

Code:

```
#include<stdio.h>
struct personal
 {
```

```
char name[50];
 int birth_year;
 int birth_month;
int birth_day;
 int age;
};
int main()
{
 struct personal p[4];
for (int i=0;i<4;i++)
{

printf("Enter details for person %d:\n", i+1);
 printf("Name: ");
 scanf("%s", p[i].name);
 printf("Date of Birth (YYYY-MM-DD): ");
 scanf("%d-%d-%d", &p[i].birth_year, &p[i].birth_month, &p[i].birth_day);
printf("Age: ");
 scanf("%d", &p[i].age);
 printf("\n");
 }

 printf("Information of 4 people:\n");
for (int i=0;i<4;i++)
{
printf("Person %d\n", i+1);
printf("Name: %s\n", p[i].name);
printf("Date of Birth: %d-%d-%d\n", p[i].birth_year, p[i].birth_month, p[i].birth_day);
printf("Age: %d\n", p[i].age);
printf("\n");
}
```

```
 return 0;
 }
```

Output:

```
Enter details for person 1:
 Name: yash
Date of Birth (YYYY-MM-DD): 10-09-2005
 Age: 18

Enter details for person 2:
Name: vikram
 Date of Birth (YYYY-MM-DD): 02-08-1999
Age: 24

Enter details for person 3:
Name: mayank
 Date of Birth (YYYY-MM-DD): 02-09-2005
 Age: 18

Enter details for person 4:
Name: Aditya
 Date of Birth (YYYY-MM-DD): 20-04-2006
Age: 19

Information of 4 people:
Person 1
 Name: Deepak
 Date of Birth: 10-9-2005
Age: 20

Person 2
```

```
 Name: sohan
Date of Birth: 2-8-1999
Age: 24

 Person 3
 Name: mahaveer
 Date of Birth: 2-9-2005
 Age: 18

 Person 4
 Name: nitesh
 Date of Birth: 20-4-2006
 Age: 17
```

Practical-5

Aim: Write a C program to calculate the sum of n numbers entered by the user using dynamic memory allocation.

Code:

```
#include<stdio.h>
 int main()
 {
 int n ;
 printf("Enter the number of elements: ");
scanf("%d", &n);
 int *arr = (int *) malloc(n * sizeof(int));
printf("Enter %d integers:\n", n);
for (int i = 0; i < n; i++) { scanf("%d", &arr[i]); }
int sum = 0;
for (int i = 0; i < n; i++)
{
sum += arr[i];
}
printf("Sum of %d integers is %d\n", n, sum);
 free(arr);
return 0;
}
```

Output:

```
Enter the number of elements: 2
Enter 2 integers: 12 23
Sum of 2 integers is 35
```

Practical-6

Aim: A file named “New” contains a series of integer numbers. Write a c program to read all numbers from a file and then copy all odd numbers into a file named “odd” and write all even numbers into a file named “even”. Then display the values of files odd and even on the screen

Code:

```
#include<stdio.h>
int main()
{
 FILE *fp1, *fp2, *fp3;
 int num;
fp1 = fopen("6 New.txt", "r");
 if (fp1 == NULL)
{
printf("Error: Unable to open the file.\n");
```

```
 return 1;
}
 fp2 = fopen("6 odd.txt", "w");
if (fp2 == NULL)
{
printf("Error: Unable to open the file.\n");
 return 1;
}
fp3 = fopen("6 even.txt", "w");
 if (fp3 == NULL)
{
printf("Error: Unable to open the file.\n");
return 1;
}
while (fscanf(fp1, "%d", &num) != EOF)
 {
 if (num % 2 == 0)
{
 fprintf(fp3, "%d\n", num);
 }
else {
 fprintf(fp2, "%d\n", num);
 }
}
 fclose(fp1);
 fclose(fp2);
fclose(fp3);
 printf("Odd numbers in the file:\n");
 fp2 = fopen("6 odd.txt", "r");
while (fscanf(fp2, "%d", &num) != EOF)
{
 printf("%d\n", num);
 }
 fclose(fp2);
 printf("Even numbers in the file:\n");
 fp3 = fopen("6 even.txt", "r");
while (fscanf(fp3, "%d", &num) != EOF)
{
 printf("%d\n", num); } fclose(fp3);
return 0;
}
```

Output:

```
Odd numbers in the file:
 33
35
 Even numbers in the file:
12 12 34 56 44 36
```

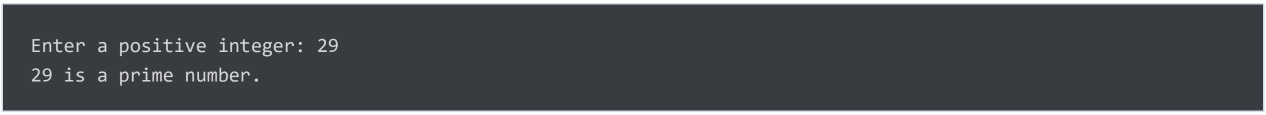
Practical-7

Aim: Write a C++ program to Check if the number is prime or not using a function

Code:

```
#include <iostream>
using namespace std;

int main() {

  int i, n;
  bool is_prime = true;

  cout << "Enter a positive integer: ";
  cin >> n;

  // 0 and 1 are not prime numbers
  if (n == 0 || n == 1) {
    is_prime = false;
  }

  // loop to check if n is prime
  for (i = 2; i <= n/2; ++i) {
    if (n % i == 0) {
      is_prime = false;
      break;
    }
  }

  if (is_prime)
    cout << n << " is a prime number";
  else
    cout << n << " is not a prime number";

  return 0;
}
```

Output:



```
Enter a positive integer: 29
29 is a prime number.
```

Practical-8

Aim: Write a C++ program that prompts the user to enter a letter and check whether a letter is a vowel or constant.

Code:

```
#include<iostream>
using namespace std;
int main()
{

   char ch;
   cout<<"Enter an Alphabet: ";
   cin>>ch;

   if(ch=='a' || ch=='e' || ch=='i' || ch=='o' || ch=='u')
      cout<<"\nIt is a Vowel";

   else if(ch=='A' || ch=='E' || ch=='I' || ch=='O' || ch=='U')
      cout<<"\nIt is a Vowel";

else

      cout<<"\nIt is a Consonant";
   cout<<endl;
   return 0;
}
```

Output:

```
Enter an Alphabet: A

It is a Vowel


Enter an Alphabet: C

It is a Consonant
```

Practical-9

Aim: Write a C++ program to demonstrate the concept of constructor and destructor.

Code:

```
#include<iostream>
using namespace std;
class MyClass
 {
 public: MyClass ()
 {
 cout<<"Constructor called."<<endl;
}
~MyClass()
 {
cout<<"Destructor called."<<endl;
}
};
int main()
{
 MyClass obj;
return 0;
}
```

Output:

```
Constructor called.
Destructor called.
```

Practical-10

Aim: Write a C++ program to implement Multilevel Inheritance.

Code:

```
#include <bits/stdc++.h>
using namespace std;
class A {
public:
int a;
void get_A_data()
{
cout << "Enter value of a: ";
cin >> a;
}
};

class B : public A {
public:
int b;
void get_B_data()
{
cout << "Enter value of b: ";
cin >> b;
}
};

class C : public B {
private:
int c;
public:
void get_C_data()
{
cout << "Enter value of c: ";
cin >> c;
}
void sum()
{
int ans = a + b + c;
cout << "sum: " << ans;
}
};
int main()
{



C obj;
obj.get_A_data();
obj.get_B_data();
obj.get_C_data();
obj.sum();
return 0;
}
```

Output:

```
Enter value of a: 4

Enter value of b: 5

Enter value of c: 9

sum: 18
```

Practical-11

Aim: Write a C++ program to overload binary + operator .

Code:

```
#include<iostream>
using namespace std;

class Complex
{
        int num1, num2;
    public:
        void accept()
        {
                cout<<"\n Enter Two Complex Numbers : ";
                cin>>num1>>num2;
        }
        Complex operator+(Complex obj)
        {
                Complex c;
                c.num1=num1+obj.num1;
                c.num2=num2+obj.num2;
                return(c);
        }
        void display()
        {
                cout<<num1<<"+"<<num2<<"i"<<"\n";
        }
};
int main()
{
        Complex c1, c2, sum;
        c1.accept();
        c2.accept();

        sum = c1+c2;

        cout<<"\n Entered Values : \n";
        cout<<"\t";
        c1.display();           cout<<"\t";
        c2.display();

        cout<<"\n Addition of Real and Imaginary Numbers : \n";
        cout<<"\t";
        sum.display();
        return 0;
}
```

Output:

```
Enter Two Complex Numbers :5 6

Enter Two Complex Numbers :7 8

Entered Values :

5+6i

7+8i

Addition of Real and Imaginary Numbers :12+14i
```

Practical-12

Aim: Write a C++ program to understand the concept of run time polymorphism.

Code:

```
#include<iostream>

 using namespace std;

 class Animal

{
```

```
public:
 virtual void sound()
 {
cout << "The animal makes a sound." << endl;
}
 };
class Dog : public Animal
{
public:
void sound()
{
cout << "The dog barks." << endl;
}
};
int main()
{
Animal* animal;

 Dog dog;

 // Assign the Dog object to the Animal pointer animal = &dog;

 // Call the virtual method using the pointer animal->sound();
 return 0;
```

Output:

```
The dog barks.
```

HTML, CSS and JS Practicals

Practical-1

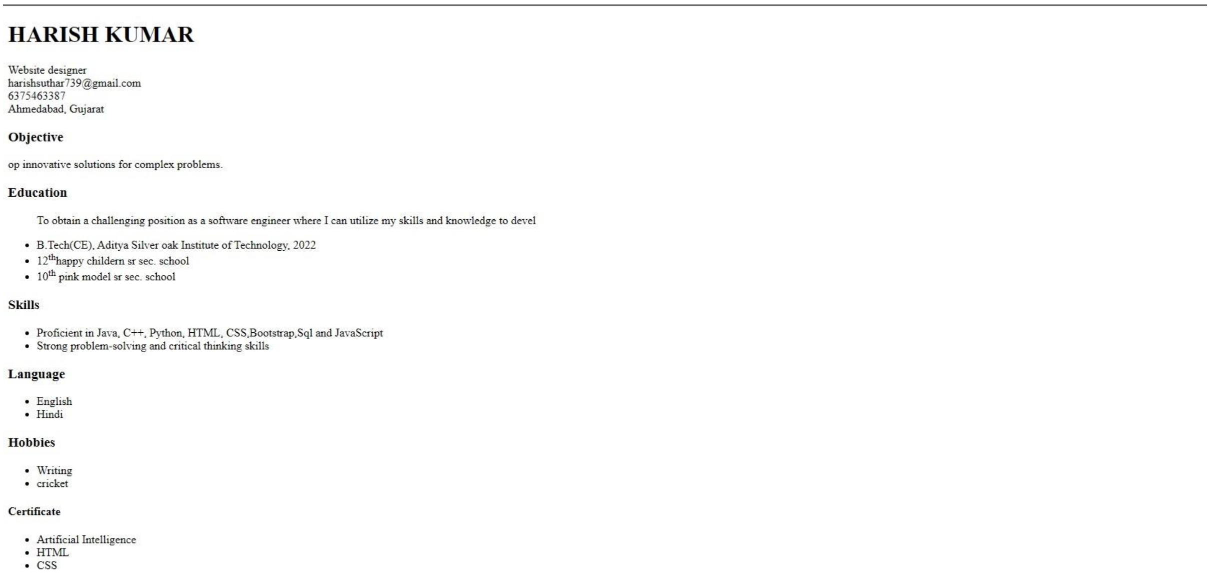
Aim: Make a Resume using the HTML tags without CSS.

Code:

```
<!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>Document</title>
</head>
<body>
   <h1>HARISH KUMAR</h1>
<div>Website designer</div>
 <div>harishsuthar739@gmail.com</div>
 <div>6375463387</div>
 <div>Ahmedabad, Gujarat</div>
<h3>Objective</h3> op innovative solutions for complex problems.</p>
<h3>Education</h3>
<ul>


<p>To obtain a challenging position as a software engineer where I can utilize my skills and
knowledge to devel
```

```
 <li>B.Tech(CE), Aditya Silver oak Institute of Technology, 2022</li>
<li>12<sup>th</sup>happy childern sr sec. school</li>

<li>10<sup>th</sup> pink model sr sec. school</li>
</ul>
<h3>Skills</h3>
<ul>
<li>Proficient in Java, C++, Python, HTML, CSS,Bootstrap,Sql and JavaScript</li>
<li>Strong problem-solving and critical thinking skills</li>
</ul>
<h3>Language</h3>
<ul>
<li>English</li>
<li>Hindi</li>
</ul>
 <h3>Hobbies</h3>
<ul>
<li>Writing</li>
<li>cricket</li>
</ul>
<h4>Certificate</h4>
<ul>
<li>Artificial Intelligence</li>
<li>HTML</li>
<li>CSS</li>
</ul>
</body>
</html>
```

Output:



HARISH KUMAR

Website designer
harishsuthar739@gmail.com
6375463387
Ahmedabad, Gujarat

Objective

op innovative solutions for complex problems.

Education

To obtain a challenging position as a software engineer where I can utilize my skills and knowledge to devel

- B.Tech(CE), Aditya Silver oak Institute of Technology, 2022
- 12thhappy childern sr sec. school
- 10th pink model sr sec. school

Skills

- Proficient in Java, C++, Python, HTML, CSS,Bootstrap,Sql and JavaScript
- Strong problem-solving and critical thinking skills

Language

- English
- Hindi

Hobbies

- Writing
- cricket

Certificate

- Artificial Intelligence
- HTML
- CSS

Practical-2

Aim: Create an HTML webpage that shows Poster Presentation using all Table Properties.

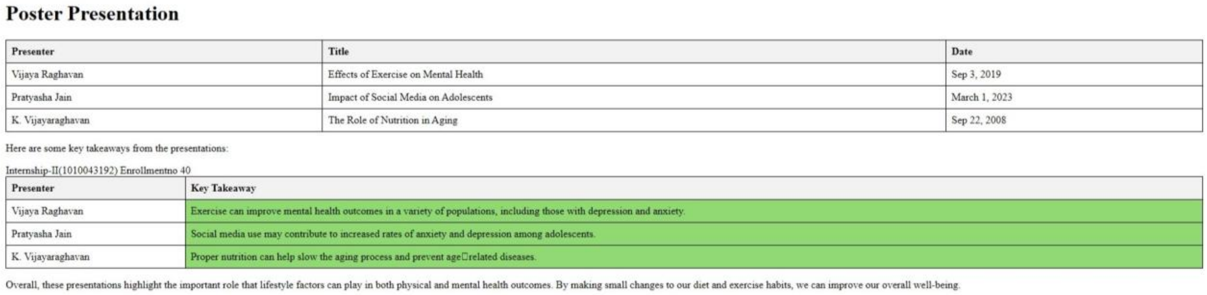
Code:

```
<!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>Document</title>
  <style>
    table {
    border-collapse:
    collapse; width: 100%;
    }
    th, td {
    border: 1px solid black;
    padding: 8px;
    text-align: left;
    }
    th {
    background-color: #f2f2f2;
    }
    .highlight {
    background-color: rgb(145, 217, 114);
    }
    </style>
</head>
<body>
  <h1>Poster Presentation</h1>
<table>
<thead>
<tr>
<th>Presenter</th>
<th>Title</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vijaya Raghavan</td>
<td>Effects of Exercise on Mental Health</td>
<td>Sep 3, 2019</td>
</tr>
<tr>
<td>Pratyasha Jain</td>
<td>Impact of Social Media on Adolescents</td>
```

```
<td>March 1, 2023</td>
</tr>
<tr>
<td>K. Vijayaraghavan</td>
<td>The Role of Nutrition in Aging</td>
<td>Sep 22, 2008</td>
</tr>
</tbody>
</table>
<p>Here are some key takeaways from the presentations:</p>
<table>
<tr>
<th>Presenter</th>
<th>Key Takeaway</th>
</tr>
<tr>
<td>Vijaya Raghavan</td>
<td class="highlight">Exercise can improve mental health outcomes in a
variety of populations, including those with depression and anxiety.</td>
Internship-II(1010043192) Enrollmentno 40
</tr>
<tr>
<td>Pratyasha Jain</td>
<td class="highlight">Social media use may contribute to increased
rates of anxiety and depression among adolescents.</td>
</tr>
<tr>
<td>K. Vijayaraghavan</td>
 <td class="highlight">Proper nutrition can help slow the aging process and prevent
age□related diseases.</td>
```

```
</tr>
</table>



<p>Overall, these presentations highlight the important role that lifestyle factors
can play in both physical and mental health outcomes. By making small changes to our diet
and exercise habits, we can improve our overall well-being.</p>
</body>
</html>
```

Output:



Poster Presentation

Presenter	Title	Date
Vijaya Raghavan	Effects of Exercise on Mental Health	Sep 3, 2019
Pratyasha Jain	Impact of Social Media on Adolescents	March 1, 2023
K. Vijayaraghavan	The Role of Nutrition in Aging	Sep 22, 2008

Here are some key takeaways from the presentations:

Internship-II(1010043192) Enrollmentno 40

Presenter	Key Takeaway
Vijaya Raghavan	Exercise can improve mental health outcomes in a variety of populations, including those with depression and anxiety.
Pratyasha Jain	Social media use may contribute to increased rates of anxiety and depression among adolescents.
K. Vijayaraghavan	Proper nutrition can help slow the aging process and prevent age□related diseases.

Overall, these presentations highlight the important role that lifestyle factors can play in both physical and mental health outcomes. By making small changes to our diet and exercise habits, we can improve our overall well-being.

Practical-3

Aim: Create an HTML page table and form.

Code:

```
<!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>Document</title>
 <style>
   table {
   border-collapse: collapse;
   width: 100%;
   }
   th, td {
   text-align: left;
   padding:
   8px;
   border-bottom: 1px solid #ddd;
   }
   th {
   background-color: #f2f2f2;
   }
   form
   { width: 50%;
   margin: 0 auto;
   }
    label {
   display: block;
   margin-bottom: 8px;
   }
   input[type="text"],
   textarea {
   width: 100%;
    padding: 12px 20px;
   margin:8px 0;
    box-sizing:border-box;
    border: 2px solid #ccc;
    border-radius: 4px;
        resize: vertical;
        }
        input[type="submit"] {
            background-color: #4CAF50;
         color: white;
        padding: 12px 20px;
         border: none;
          border-radius: 4px;
        cursor: pointer;
        }
        input[type="submit"]:hover {
        background-color: #45a049;
        }
        .form-group {
        margin-bottom: 16px;
        }

        .error
        {
        color:  red;
        font-size: 12px;
        margin-top: 4px;
        }
        </style>
</head>
<body>
    <h1>Table and Form</h1>
<table>
<thead>
<tr>
```

```
<th>Name</th>
<th>Email</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>HARISH KUMAR</td>
<td>harishsuthar739@gmail.com</td>
<td>6375463387</td>
</tr>
<tr>
<td>VIKRAM SUTHAR</td>
<td>vikramsuthar@gmail.com</td>
<td>9852456023</td>
</tr>
</tbody>
</table>
<form>


<h2>Contact Form</h2>
<div class="form-group">
<label for="name">Name</label>
<input type="text" id="name" name="name" required>
<span class="error">Please enter your name</span>
</div>
<div class="form-group">
<label for="email">Email</label>
<input type="text" id="email" name="email" required>
<span class="error">Please enter a valid email address</span>
```

```
</div>
<div class="form-group">




<label for="message">Message</label>
<textarea id="message" name="message" required></textarea>
<span class="error">Please enter a message</span>
</div>
<input type="submit" value="Send">
</form>
</body>
</html>
```

Output:

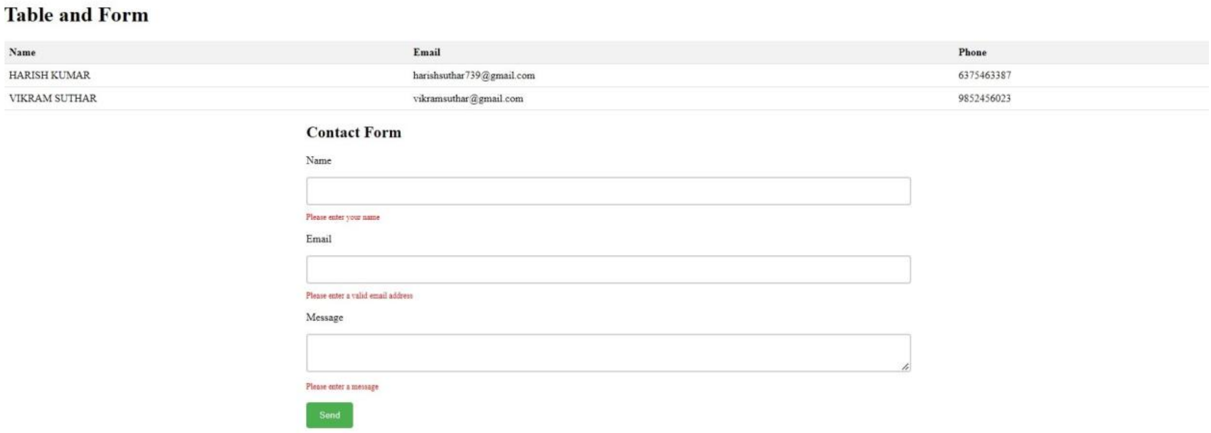


Table and Form

Name	Email	Phone
HARISH KUMAR	harishsuthar739@gmail.com	6375463387
VIKRAM SUTHAR	vikramsuthar@gmail.com	9852456023

Contact Form

Name

Please enter your name

Email

Please enter a valid email address

Message

Please enter a message

Practical-4

Aim: Create Registration form and do proper validation with HTML 5 inbuilt functionality. (Don't use JavaScript).

Code:

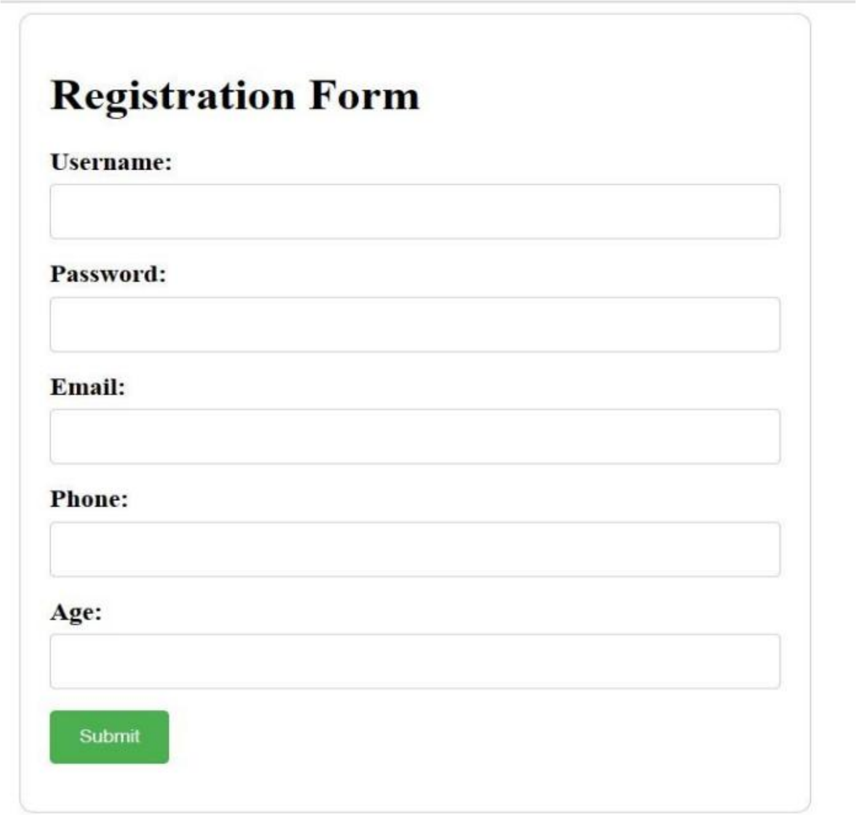
```
<html>
 <head>
 <title>Table and Form with CSS</title>
 <style>
 table {
 border-collapse: collapse;
  width: 100%;
 }
 th, td {
text-align: left;
padding: 8px;
borderbottom: 1px solid #ddd;
 }
th {
   background-color: #f2f2f2;
 }

  form {
width: 50%;
 margin: 0 auto;
 }

label {
display: block;
margin-bottom: 8px;
```

```
 }

input[type="text"], textarea {
width: 100%;
 padding: 12px 20px;
  margin: 8px 0;
  box-sizing: border-box;
  border: 2px solid #ccc;
  border-radius: 4px;
 resize: vertical;
 }



input[type="submit"] {
background-color: #4CAF50;
color: white;
 padding: 12px 20px;
 border: none;
borderradius: 4px;
cursor: pointer;
 }

 input[type="submit"]:hover { background-color: #45a049;
 }

 .form-group {
margin-bottom: 16px;
 }
```

```
.error {



  color: red;
  font-size: 12px;
   margin-top: 4px;
}
</style>
</head>
<body>
<h1>Table and Form</h1>

<table>
<thead>
<tr>
<th>Name</th>
<th>Email</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>HARISH KUMAR</td>
<td>harishsuthar@gamil.com</td> <td>6375463387</td>
</tr>
<tr>
<td>janak suthar</td>
<td>janakkumar@example.com</td>
<td>7643255678</td>
</tr>
</tbody>
</table>
```

```
 <form>
 <h2>Contact Form</h2>


 <div class="form-group">
 <label for="name">Name</label>
 <input type="text" id="name" name="name" required>
 <span class="error">Please enter your name</span>
 </div>

 <div class="form-group">
 <label for="email">Email</label>
 <input type="text" id="email" name="email" required>
 <span class="error">Please enter a valid email address</span>
 </div>

 <div class="form-group">
 <label for="message">Message</label>
 <textarea id="message" name="message" required></textarea>
 <span class="error">Please enter a message</span>
 </div>
 <input type="submit" value="Send">
</form>
</body>
</html>
```

Output:



Registration Form

Username:

Password:

Email:

Phone:

Age:

Practical-5

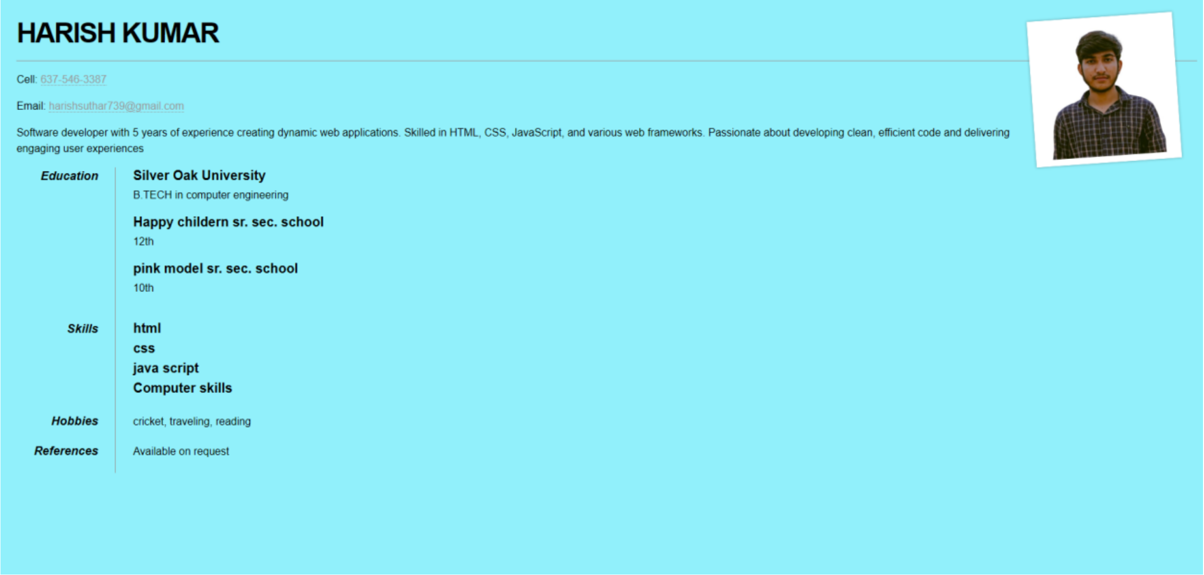
Aim: Make a Resume using the HTML tags with CSS.

Code:

```
<!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" />
```

```
    <link rel="stylesheet" href="style.css" />
    <title>Resume</title>
    <style>
    * {
  box-sizing: border-box;
}

body {
  margin: 2.2rem;
  background-color: rgb(145, 240, 251);
}

div#resume {
  min-width: 310px;
  font: 16px Helvetica, Avernir, sans-serif;
  line-height: 24px;
  color: #000;
}

div#resume h1 {
  margin: 0 0 16px 0;
  padding: 0 0 16px 0;
  font-size: 42px;
  font-weight: bold;
  letter-spacing: -2px;
  border-bottom: 1px solid #999;
  line-height: 50px;
}

div#resume h2 {
  font-size: 20px;
  margin: 0 0 6px 0;
  position: relative;
}

div#resume h2 span {
  position: absolute;
  bottom: 0;
  right: 0;
  font-style: italic;
  font-family: Georgia, serif;
  font-size: 16px;
  color: #999;
  font-weight: normal;
}

div#resume p {
  margin: 0 0 16px 0;
}

div#resume a {
  color: #999;
  text-decoration: none;
  border-bottom: 1px dotted #999;
}

div#resume a:hover {
  border-bottom-style: solid;
  color: #000;
}

div#resume p.objective {
  font-family: Georgia, serif;
  font-style: italic;
  color: #666;
}

div#resume dt {
  font-style: italic;
  font-weight: bold;
  font-size: 18px;
  text-align: right;
  padding: 0 26px 0 0;
  width: 150px;
  border-right: 1px solid #999;
}

div#resume dl {
  display: table-row;
}

div#resume dl dt,
div#resume dl dd {
  display: table-cell;
  padding-bottom: 20px;
}

div#resume dl dd {
  width: 500px;
  padding-left: 26px;
}

div#resume img {
  float: right;
  padding: 10px;
  background: #fff;
  margin: 0 30px;
  transform: rotate(-4deg);
  box-shadow: 0 0 4px rgba(0, 0, 0, 0.3);
  width: 30%;
  max-width: 220px;
}

@media screen and (max-width: 1100px) {
  div#resume h2 span {
    position: static;
    display: block;
    margin-top: 2px;
  }
}

@media screen and (max-width: 550px) {
  body {
    margin: 1rem;
  }
  div#resume img {
    transform: rotate(0deg);
  }
}

@media screen and (max-width: 400px) {
  div#resume dl dt {
    border-right: none;
    border-bottom: 1px solid #999;
  }
  div#resume dl,
  div#resume dl dd,
  div#resume dl dt {
    display: block;
    padding-left: 0;
    margin-left: 0;
    padding-bottom: 0;
    text-align: left;
    width: 100%;
  }
  div#resume dl dd {
    margin-top: 6px;
  }
  div#resume h2 {
    font-style: normal;
    font-weight: 400;
    font-size: 18px;
  }
  div#resume dt {
    font-size: 20px;
  }
  h1 {
    font-size: 36px;
    margin-right: 0;
    line-height: 0;
  }
  div#resume img {
    margin: 0;
  }
}

@media screen and (max-width: 320px) {
  body {
    margin: 0;
  }
  img {
    margin: 0;
    margin-bottom: -40px;
  }
  div#resume {
    width: 320px;
    padding: 12px;
    overflow: hidden;
  }   p,
  li {
    margin-right: 20px;
  }
}
    </style>
</head>

<body>
    <div id="resume">
        <img src="./IMG_20220820_183453_918-removebg-preview (1).jpg" alt="HARISH
KUMAR">
        <h1>HARISH KUMAR</h1>
        <p>Cell: <a href=#>637-546-3387</a>
        <p>Email: <a href=#>harishsuthar739@gmail.com</a>
        <p id="objective">Software developer with 5 years of experience creating dynamic web
applications.
            Skilled in HTML, CSS, JavaScript, and various web frameworks. Passionate about
developing
            clean, efficient code and delivering engaging user experiences
        <dl>
            <dt>Education
            <dd>
                <h2>Silver Oak University</h2>
                <p>B.TECH in computer engineering
                </p>


                <h2>Happy childern sr. sec. school</h2>
                <p>12th</p>


                <h2>pink model sr. sec. school</h2>
                <p>10th</p>
        </dl>
        <dl>
            <dt>Skills
            <dd>
                <h2>html</h2>
                <h2>css</h2>
                <h2>java script</h2>
                <h2>Computer skills</h2>

        </dl>


        <dl>
            <dt>Hobbies
            <dd>cricket, traveling, reading
        </dl>
        <dl>
            <dt>References
            <dd>Available on request
        </dl>
    </div>
</body>

</html>
```

Output:



HARISH KUMAR

Cell: 637-546-3387

Email: harishsuthar739@gmail.com

Software developer with 5 years of experience creating dynamic web applications. Skilled in HTML, CSS, JavaScript, and various web frameworks. Passionate about developing clean, efficient code and delivering engaging user experiences

Education

Silver Oak University
B.TECH in computer engineering

Happy childern sr. sec. school
12th

pink model sr. sec. school
10th

Skills

html
css
java script
Computer skills

Hobbies cricket, traveling, reading

References Available on request

Practical-6

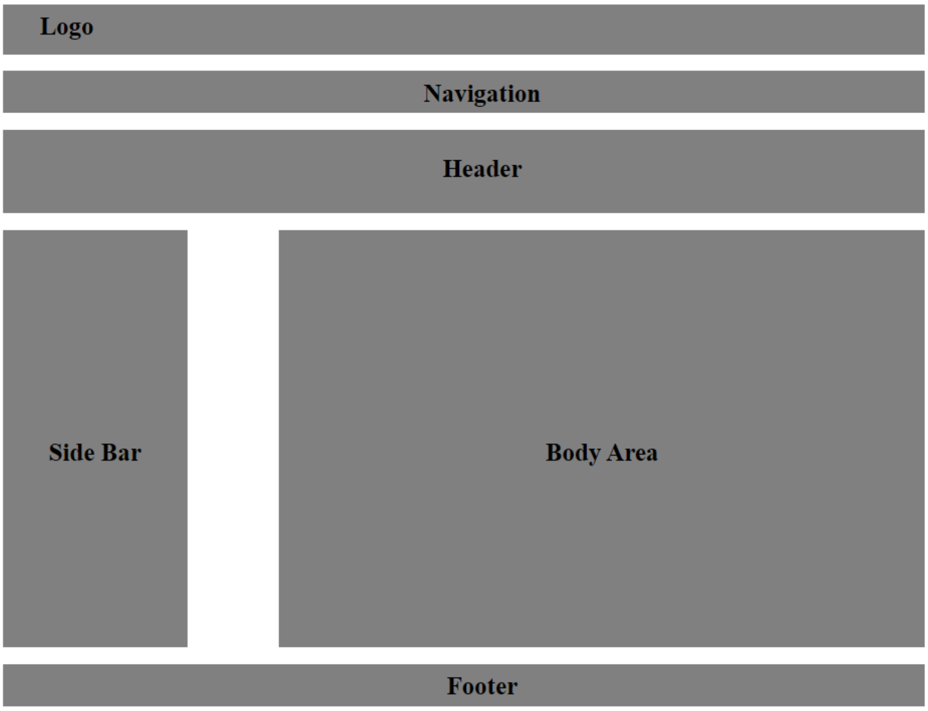
Aim: Create an HTML Page containing the following Gray Layout using CSS.

Code:

```
<!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>Document</title>
  <style>
    *{font-size: 30px;font-weight: bolder;}
    #r1{background-color: gray; height: 50px; width: 48%; padding-top: 10px;padding-left:
   2%;margin-bottom: 20px;}
```

```
    #r2,#r5{background-color: gray; height: 40px; width: 48%; padding-top: 10px;padding-left:

2%;margin-bottom: 20px; text-align: center;}
    #r3{background-color: gray; height: 70px; width: 48%; padding-top: 30px;padding-left:
   2%;margin-bottom: 20px; text-align: center;}
    #r4{height: 500px; width: 100%; margin-bottom: 20px;}
    #r4 div{float: left; background-color: gray; padding-top: 250px; height: 250px; text-align: center;}
    #r4c1{width: 10%; margin-right: 5%; }
    #r4c2{width: 35%; }
    </style>
</head>
<body>


    <div id="maindiv">
    <div id="r1">Logo</div>
    <div id="r2">Navigation</div>
    <div id="r3">Header</div>
    <div id="r4">
    <div id="r4c1">Side Bar</div>
    <div id="r4c2">Body Area</div>
    </div>
    <div id="r5">Footer</div>
    </div>
</body>
</html>
```

Output:



Practical-7

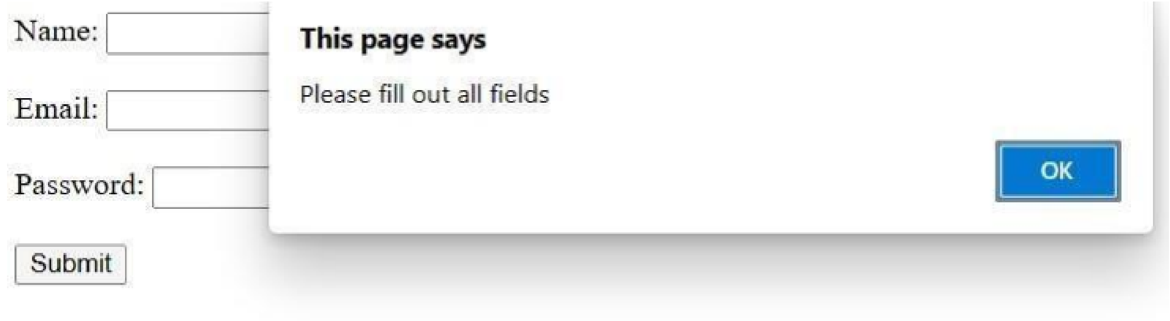
Aim: . Demonstrate JavaScript Form Validation with proper examples.

Code:

```
<!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>Document</title>
</head>
```

```
<body>
 <title>Basic Form Validation</title>
 <script>
 function validateForm() {
 var name = document.forms["myForm"]["name"].value;
var email = document.forms["myForm"]["email"].value;
var password = document.forms["myForm"]["password"].value;
 if (name == "" || email == "" || password == "") {
 alert("Please fill out all fields"); return false; }
 }



</script>
 <form name="myForm" onsubmit="return validateForm()">
 <label for="name">Name:</label>
 <input type="text" id="name" name="name">
 <br><br>
 <label for="email">Email:</label>
 <input type="email" id="email" name="email">
 <br><br>
 <label for="password">Password:</label>
 <input type="password" id="password" name="password">
 <br><br>
 <input type="submit" value="Submit">
 </form>

</body>
</html>
```

Output:



Name:

Email:

Password:

Submit

This page says

Please fill out all fields

OK

Practical-8

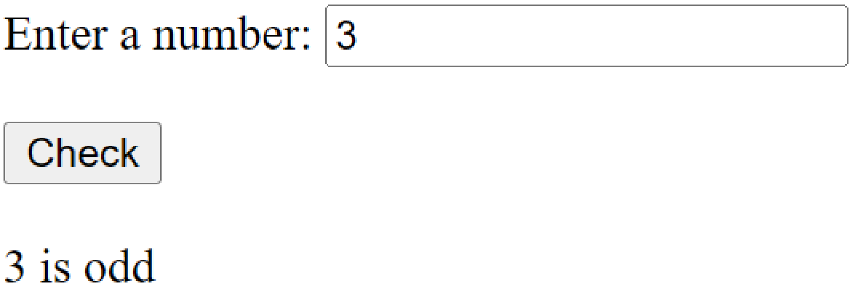
Aim: Write a javascript to check if the number is even or odd.

Code:

```
<!DOCTYPE html>
<html>
 <head>
 <title>Even or Odd Checker</title>
 <script>
 function checkEvenOrOdd()
 {
 var number = document.getElementById("number").value; if
(number%2==0)
 {
 document.getElementById("result").innerHTML = number + " is even";
 } else {
 document.getElementById("result").innerHTML = number + " is odd";
 }
 }
```

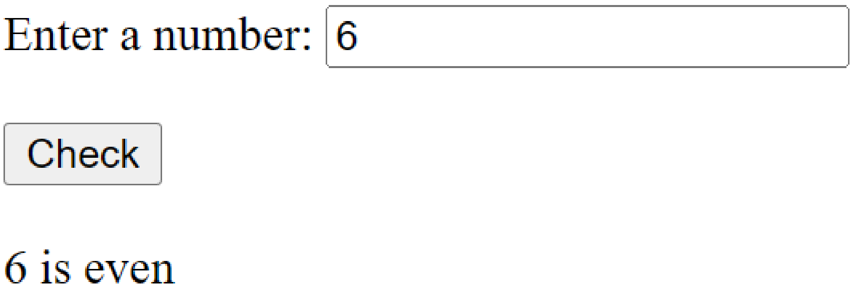
```
</script>
</head>
<body>
<label for="number">Enter a number:</label>
<input type="number" id="number">
<br><br>
<button onclick="checkEvenOrOdd()">Check</button>
<br><br>
<div id="result"></div>
</body>
</html>
```

Output:



Enter a number: 3

3 is odd



Enter a number: 6

6 is even

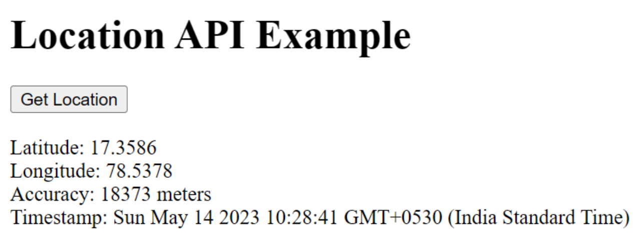
Practical-9

Aim: Create a page and access the LocationAPI.

Code:

```
<!DOCTYPE html>
<html>
 <head>
 <title>Location API Example</title>
 <script>
 function getLocation()
 {
 if (navigator.geolocation)
 {
 navigator.geolocation.getCurrentPosition(showPosition);

}
else
 {
 alert("Geolocation is not supported by this browser.");
 }
 }
 function showPosition(position)
 {
 var latitude = position.coords.latitude; var
longitude = position.coords.longitude; var
accuracy = position.coords.accuracy;
 var timestamp = new Date(position.timestamp);
 document.getElementById("latitude").innerHTML = "Latitude: " + latitude;
document.getElementById("longitude").innerHTML = "Longitude: " + longitude;
document.getElementById("accuracy").innerHTML = "Accuracy: " + accuracy + " meters";
document.getElementById("timestamp").innerHTML = "Timestamp: " + timestamp;
 }
 </script>
 </head>
 <body>
 <h1>Location API Example</h1>
 <button onclick="getLocation()">Get Location</button>
 <br><br>
 <div id="latitude"></div>
 <div id="longitude"></div>
 <div id="accuracy"></div>
 <div id="timestamp"></div>
 </body>
</html>
```

Output:



Location API Example

```
Latitude: 17.3586
Longitude: 78.5378
Accuracy: 18373 meters
Timestamp: Sun May 14 2023 10:28:41 GMT+0530 (India Standard Time)
```

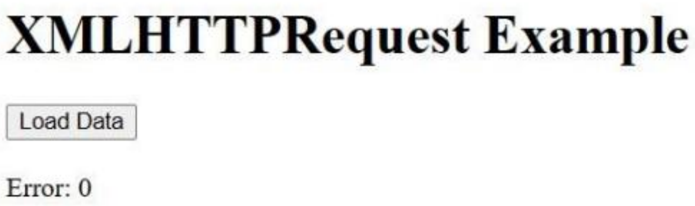
Practical-10

Aim: Create a simple XMLHTTPRequest,and retrieve the data from the text file.

Code:

```
<!DOCTYPE html>
<html>
 <head>
 <title>XMLHTTPRequest Example</title>
 <script>
 function loadData()
 {
 var xhr = new XMLHttpRequest();
xhr.open('GET', 'example.txt');
 xhr.onreadystatechange = function()
 {
 if (xhr.readyState === XMLHttpRequest.DONE)
 {
 if (xhr.status === 200)
 {
 var response = xhr.responseText;
 document.getElementById('output').innerHTML = response;

}
else
 {
 document.getElementById('output').innerHTML = 'Error: ' + xhr.status;
 }
 }
 };
 xhr.send();
 }
 </script>
 </head>
 <body>
 <h1>XMLHTTPRequest Example</h1>
 <button onclick="loadData()">Load Data</button>
 <br><br>
 <div id="output"></div>
 </body>
</html>
```

Output:



XMLHTTPRequest Example

Load Data

Error: 0

DBMS PRACTICALS

Practical-1

Aim: To study DDL-create and DML-insert commands.

Code:

DDL and DML are two types of SQL commands. DDL stands for Data Definition Language, and it is used to

create and modify the structure of database objects, such as tables, indexes, and views. DML stands for Data

Manipulation Language, and it is used to insert, update, and delete data in a database.

Here are some examples of DDL and DML commands:

DDL - CREATE TABLE:

The CREATE TABLE statement is used to create a new table in a database. Here is an example:

```
CREATE TABLE customers ( id INT
PRIMARY KEY, name VARCHAR(50),
email VARCHAR(50), phone
VARCHAR(20)
);
```

This statement creates a new table named "customers" with four columns: id, name, email, and phone. The id

column is defined as the primary key, which means that it will contain a unique value for each row in the table.

DDL - ALTER TABLE:

The ALTER TABLE statement is used to modify the structure of an existing table in a database. Here is an

example:

```
ALTER TABLE customers
ADD address VARCHAR(100);
```

This statement adds a new column named "address" to the "customers" table.

DML - INSERT INTO:

The INSERT INTO statement is used to insert new rows into a table. Here is an example:

```
INSERT INTO customers (id, name, email, phone)
VALUES (1, 'John Doe', 'john.doe@example.com', '555-1234');
```

This statement inserts a new row into the "customers" table with the specified values for the id, name, email,

and phone columns.

DML - UPDATE:

The UPDATE statement is used to modify existing rows in a table. Here is an example:

```
UPDATE customers
SET phone = '555-5678'
WHERE id = 1;
```

This statement updates the "phone" column for the row with id 1 in the "customers" table.

DML - DELETE:

The DELETE statement is used to delete rows from a table. Here is an example:

DELETE FROM customers

WHERE id = 1;

This statement deletes the row with id 1 from the "customers" table.

Practical-2

Aim: Create tables and insert sample data in tables.

Insert following values in the table Employee.

emp_id	emp_name	emp_sal	emp_comm	dept _no
101	vikram	800		20
102	sohan	1600	300	25
103	mahaveer	1100	0	20
104	janak	3000		15
105	manish	5000	50000	10
106	kajal	2450	24500	10
107	riya	2975		30

Code:

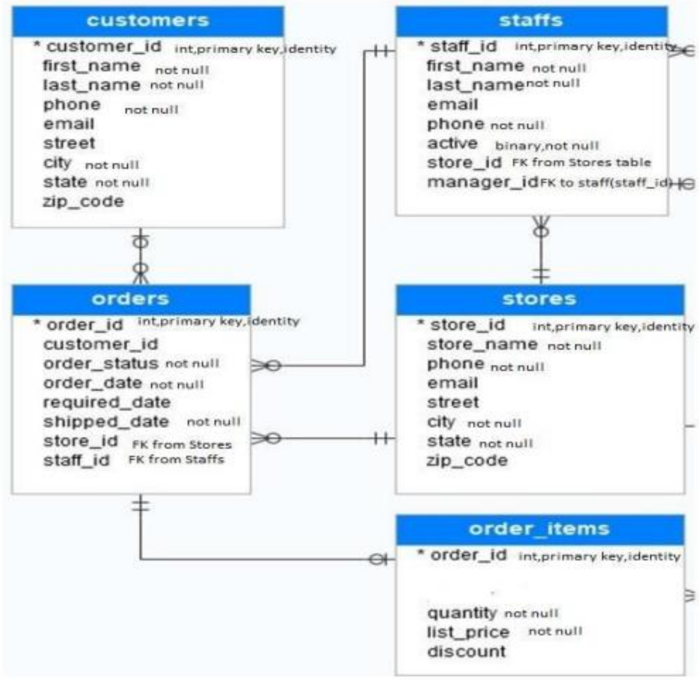
```
CREATE TABLE Employee (
 emp_no INT PRIMARY KEY, emp_name VARCHAR(30)
NOT NULL, emp_sal DECIMAL(8,2)
NOT NULL, emp_comm
DECIMAL(6,1), dept_no INT NOT


NULL
);
```

```
INSERT INTO Employee (emp_no, emp_name, emp_sal, emp_comm, dept_no) VALUES
(101, 'vikram', 800.00, 20, 0),
 (102, 'Sohan', 1600.00, 300, 25),
 (103, 'mahaveer', 1100.00, 0, 20),
 (104, 'janak', 3000.00, 15, 0),
 (105, 'manish', 5000.00, 50000, 10),
 (106, 'kajal', 2450.00, 24500, 10),
 (107, 'riya', 2975.00, 30, 0);
```

Practical-3

Aim: Write the SQL queries to provide constraints on given tables

Write the SQL queries to provide constraints on given tables



Code:

```
CREATE TABLE customers ( customer_id INT
PRIMARY
KEY, first_name
VARCHAR(50) NOT NULL,
last_name VARCHAR(50)
NOT NULL, phone
VARCHAR(20), email
VARCHAR(100), street
VARCHAR(100), city
VARCHAR(50) NOT NULL,
state VARCHAR(50) NOT NULL,

zip_code VARCHAR(20)
);

CREATE TABLE staff ( staff_id
INT PRIMARY KEY,
first_name VARCHAR(50)
NOT NULL, last_name
VARCHAR(50) NOT NULL,
email VARCHAR(100), phone
VARCHAR(20) NOT

NULL, active BOOLEAN NOT
NULL, store_id INT,
 FOREIGN KEY (store_id) REFERENCES stores(store_id)
);

CREATE TABLE stores (
store_id INT PRIMARY KEY,
store_name VARCHAR(50) NOT
NULL, phone
VARCHAR(20) NOT NULL,
 email VARCHAR(100),
street VARCHAR(100), city
VARCHAR(50) NOT
NULL, state
VARCHAR(50) NOT
NULL, zip_code
VARCHAR(20), manager_id
INT,
 FOREIGN KEY (manager_id) REFERENCES staff(staff_id)
);
```

```
CREATE TABLE orders ( order_id
INT PRIMARY KEY, order_date
DATE NOT NULL, required_date
DATE, shipped_date DATE NOT
NULL, order_status VARCHAR(20)
NOT NULL,

customer _id
INT, staff_id
INT,
store_id
INT,
 FOREIGN KEY (customer_id) REFERENCES customers(customer_id),
 FOREIGN KEY (staff_id) REFERENCES staff(staff_id),
 FOREIGN KEY (store_id) REFERENCES stores(store_id)
);

CREATE TABLE order_items ( order_id
INT, item_id INT PRIMARY KEY,
quantity INT


NOT NULL, list_price DECIMAL(10,
2) NOT NULL, discount DECIMAL(5,
2),
 FOREIGN KEY (order_id) REFERENCES orders(order_id)
);
```

Practical-4

Aim: Write the SQL queries to perform various aggregate functions on table data.

1. List total deposit from deposit.

Code:
```
SELECT SUM(amount) AS total_deposit FROM deposit;
```

2. List total amount from andheri branch

Code:
```
SELECT SUM(amount) AS total_amount FROM deposit WHERE bname = 'andheri';
```

3. Count total number of customers

Code:
```
SELECT COUNT(DISTINCT cname) AS total_customers FROM deposit;
```

4. Count total number of customer’s cities

Code:
```
SELECT COUNT(DISTINCT bname) AS total_cities FROM deposit;
```

5. Update the value dept_no to 10 where second character of emp. name is ‘m’.

Code:
```
UPDATE Employee SET dept_no = 10 WHERE emp_name LIKE '_m%';
```

6. Update the value of employee name whose employee number is 103.

Code:
```
UPDATE Employee SET emp_name = 'Adam' WHERE emp_no = 103;
```

7. Write a query to display the current date. Label the column Date

Code:
```
SELECT GETDATE() AS Date;
```

8. For each employee, display the employee number, salary, and salary in creased by 15% and expressed as a whole number. Label the column New Salary

Code:
```
SELECT emp_no, emp_sal, ROUND(emp_sal*1.15,0) AS "New Salary" FROM Employee;
```

9. Modify your previous query to add a column that subtracts the old salary from the new salary. Label the column Increment.

Code: SELECT emp_no, emp_sal, ROUND(emp_sal*1.15,0) AS "New Salary", ROUND(emp_sal*0.15,0) AS Increment FROM Employee;

Practical-5

Aim: Write the SQL queries to perform numeric,date and String functions

1. Retrieve all data from employee, jobs and deposit.

Code: SELECT * FROM employee; SELECT

* FROM jobs;

SELECT * FROM deposit;

2. Give details of account no. and deposited rupees of customers having account opened between dates 01-01-06 and 25-07-06.

Code: SELECT a_no, amount

FROM deposit

WHERE a_date BETWEEN '2006-01-01' AND '2006-07-25';

3. Display all jobs with minimum salary is greater than 4000.

Code: SELECT * FROM jobs

WHERE

min_sal > 4000;

4. Display name and salary of employee whose department no is 20. Give alias name to name of employee.

Code:
```
SELECT emp_no, emp_name AS employee_name, emp_sal, dept_no
FROM employee
WHERE dept_no = 20;
```

5. Display employee no,name and department details of those employee whose department lies in(10,20)

Code:
```
SELECT emp_no, emp_name, dept_no FROM employee WHERE
dept_no IN (10, 20);
```

6. Display all employee whose name start with ‘A’ and third character is ‘ ‘a’.

Code:
```
SELECT * FROM employee
WHERE emp_name LIKE 'A _a%';
```

7. Display name, number and salary of those employees whose name is 5 characters long and first three characters are ‘Ani’.

Code:
```
SELECT emp_name, emp_no, emp_sal FROM employee WHERE
emp_name LIKE 'Ani___';
```

8. Display the non-null values of employees and also employee name second charactershould be ‘n’ and string should be 5 character long.

Code:
```
SELECT *
FROM employee
WHERE emp_name LIKE '_n%' AND LENGTH(emp_name) = 5 AND
```

```
emp_name IS
NOT NULL;
```

9. Display the null values of employee and also employee name’s third character should be ‘a’.

Code:

```
SELECT * FROM employee
WHERE emp_name LIKE '__a%' AND emp_name IS NULL;
```