**NAME - MAYANK SANKLIYA**

|  |  |
| --- | --- |
| **EN. NO.** | **2202030400162** |
| **SUBJECT** | **OJT PRACTICALS (INTERNSHIP**  **- II)** |
| **COURSE** | **B.TECH(C.E.)** |

OJT PRACTICAL

# PRACTICAL-1

**AIM** :- Write a C program to print the address of a variable using a pointer.

### CODE :-

#include

<stdio.h>

int main()

{

Int num;

int \*ptr = &num;

printf("Enter The Number: \n"); scanf("%d",&num);

printf("Address of Number Variable: %p", ptr);

return 0;

}

**OUTPUT :-** Enter The Number:

12

Address of Number Variable: 000000000062FE14

# 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("Invali d 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\_yea r; int birth\_mo nth; 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: rutu

Date of Birth (YYYY-MM-DD): 02-08-1999

Age: 24

Enter details for person 3:



Name: Guddu

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: Abhisek

Date of Birth: 2-8- 1999 Age: 24

Person 3 Name:

Rahul

Date of Birth: 2-9- 2005 Age: 18

Person 4 Name:

Mnish

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> #include

<stdlib.h>

int main()

{

i n t 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; bool Prime(int num)

{

if(num<=1

)

{

return false;

}

for (int i=2;i<=num/2;i++)

{

if (num%i==0)

{

return false;

}

}

return true;

}

int main()

{

int num; cout<<"Enter a

number: ";

> num;

cin>

if(Prime(num))

{

cout<<num<<" is a prime number."<<endl;

}

e l s e

{

cout<<num<< " is not a prime number."<< endl;

}

return 0;

}

**OUTPUT :-**

Enter a number: 34 34 is not a prime number.

Enter a number: 13

13 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 letter; cout<<"Enter a letter: "; cin>>letter; if(letter=='a' || letter=='e' || letter=='i' || letter=='o' || letter=='u' || letter=='A' || letter=='E' || letter=='I' || letter=='O' || letter=='U')

{

cout<<letter<<" is a vowel."<<endl;

}

else

{

cout<<letter<<" is a consonant."<<endl;

}

return 0;

}

## OUTPUT :-

Enter a letter: a a is a vowel.

Enter a letter: c c is a conson a nt.

# PRACTICAL-9

**AIM** :- Write a C++ program to demonstrate the concept of constructor and destructor.

CODE :-

#include

<iostream> using namespace std;

class MyClass

{

public: MyClas s ()

{

cout<<"Constructor called."<<endl;

}

~MyClass()

{

cout<<"Destructor called."<<endl;

}

};

int main()

{

MyClass obj; return 0;

}

## CODE :-

Constructor called.

Destructor called.

# PRACTICAL-10

**AIM** :- Create a class student that stores roll\_no, name. Create a class test that stores marks obtained in five subjects. Class result derived from student and test contains the total marks and percentage obtained in test. Input and display information of a student.

CODE :-

#include<io st ream> #include<str i ng> using namespace std;

class Student

{ public:

int roll\_no; string name;

};

class Test{ public

:

int subject\_marks[5];

int total\_marks(){ int total = 0; for(int i=0; i<5; i++){

total += subject\_marks[i];

}

return total;

}

double percentage(){

return (total\_marks() / 5.0);

}

};

class Result : public Student, public Test{ public:

Result(int roll\_no, string name, int marks[]){ this->roll\_no =

roll\_no; this->name

= name; for(int i=0; i<5; i++){

subject\_marks[i] = marks[i];

}

}

};

int main(){ int marks[] =

{80, 90, 85, 75, 95};

Result r(1, "John Doe", marks);

cout << "Roll No: " << r.roll\_no

<< endl; cout << "Name: "

<< r.name

<< endl;

cout << "Marks: ";

for(int i=0; i<5; i++){

cout << r.subject\_marks[i] << " ";

}

cout << endl; cout << "Total Marks: "

<< r.total\_marks() << endl;

cout << "Percentage: " << r.percentage() << endl;

return 0;

}

## OUTPUT :-

Roll No: 1

Name: John Doe Marks: 80 90 85 75

95

Total Marks: 425

Percentage: 85

# PRACTICAL-11

**AIM** :- Write a C++ program to overload binary + operator.

### CODE :-

#include

<iostream>

using namespace std;

class MyClass

{

private

: int

value; public

:

MyCla s s(int v)

{

value = v;

}

MyClass operator+(const MyClass& other) const

{

int result = value + other.value; return MyClass(result);

}

int getValue() const

{

return value;

}

};

int main()

{

int x,y;

cout<<"Enter the value: ";

cin>>x

;

cin>>y

;

MyClass a(x); MyClass b(y);

MyClass c=a+b;

cout<<"a = "<<a.getValue()<< endl; cout<<"b = "<<b.getValue()<<endl;

cout<<"c = a + b = "<<c.getValue()<< endl;

return 0;

}

OUTPUT :-

Enter the value: 23

32 a = 23

b = 32 c

= a + b = 55

# PRACTICAL-12

**AIM** :- Create a base class called 'SHAPE' having two data members of type double, member function get\_data( ) to initialize base class data members, pure virtual member function display\_area( ) to compute and display the area of the geometrical object. Derive two specific classes 'TRIANGLE' and 'RECTANGLE' from the base class. Using these three classes design a program that will accept dimension of a triangle / rectangle interactively and display the area.

CODE :-

#include

<iostream> using namespace std;

class Shape

{

public: virtual void draw()

{

cout<<"Drawing a shape"<<endl;

}

};

class Circle:public Shape

{

publ ic: voi d dra w()

{

cout<<"Drawing a circle"<<endl;

}

};

class Rectangle:public Shape

{

publ ic: void dra w()

{

cout<<"Drawing a rectangle"<<endl;

}

};

int main()

{

Shape\* s = new Shape(); Shape\* c = new Circle(); Shape\* r = new Rectangle();

s->draw();

1. >draw(); r-

>draw(); return 0;

}

## OUTPUT :-

Drawing a shape Drawing a circle Drawing a rectangle

# PRACTICAL-13

**AIM** :- To study DDL-create and DML-insert commands.

DDL (Data Definition Language) and DML (Data Manipulation Language) are two different types of SQL (Structured Query Language) commands used to manage and manipulate databases.

DDL commands are used to define and manipulate the structure of the database, including creating tables, altering tables, adding columns, and deleting tables.

DML commands, on the other hand, are used to manipulate the data within the database, including inserting, updating, and deleting data in tables.

Let's start by looking at the syntax and usage of DDL-create and DML-insert commands: DDL-Create Command:

The create command is used to create a new table in the database. The basic syntax of the create command is as follows:

CODE :-

CREATE TABLE

table\_name ( column1 datatype, column2 datatype, column3 datatype,

....

);

For example, to create a table named "customers" with columns for "id", "name", "email", and "phone", the following command would be used:

## CODE :-

CREATE TABLE

customers ( id INT PRIMARY KEY,

name VARCHAR(5 0),

email VARCHAR(5 0),

phone VARCHAR(20)

);

DML-Insert Command:

The insert command is used to insert data into a table in the database. The basic syntax of the insert command is as follows:

## CODE :-

INSERT INTO table\_name (column1, column2, column3, ...) VALUES (value1, value2, value3, ...);

For example, to insert a new row of data into the "customers" table created in the previous example, the following command would be used:

## CODE :-

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

This command would insert a new row into the "customers" table with the values specified for the "id", "name", "email", and "phone" columns.

I hope this helps you understand the basics of DDL-create and DML-insert commands in SQL. If you have any further questions, feel free to ask!

# PRACTICAL-14

**AIM** :- Create following Table

* 1. Job (job\_id, job\_title, min\_sal, max\_sal)

|  |  |
| --- | --- |
| COLUMN NAME | DATA TYPE |
| job\_id | Varchar(15) |
| job\_title | Varchar(30) |
| min\_sal | Int |
| max\_sal | Int |

### CODE :-

CREATE TABLE Job (

job\_id VARCHAR(15) PRIMARY KEY,

job\_title VARCHAR(30), min\_sal INT,

max\_sal INT

);

1. Employee (emp\_no, emp\_name, emp\_sal, emp\_comm, dept\_no)

|  |  |
| --- | --- |
| COLUMN NAME | DATA TYPE |
| emp\_no | Int |
| emp\_name | Varchar(30) |
| emp\_sal | decimal(8,2) |
| emp\_comm | decimal(6,1) |
| dept\_no | Int |

**CODE :-** CREATE TABLE

Employee ( emp\_no INT PRIMARY KEY,

emp\_name VARCHAR(3 0),

emp\_sal DECIMAL(8, 2),

emp\_comm DECIMAL(6, 1),

dept\_no INT

);

1. deposit(a\_no,cname,bname,amount,a\_date)

|  |  |
| --- | --- |
| COLUMN NAME | DATA TYPE |
| a\_no | Int,identity |
| cname | Varchar(50) |
| bname | Varchar(30) |
| amount | Decimal(4,2) |
| a\_date | Date |

**CODE :-** CREATE TABLE

deposit ( a\_no INT IDENTITY PRIMARY

KEY, cname VARCHAR(50),

bname VARCHAR(30),

amount DECIMAL(4,2),

a\_date DATE

);

1. borrow(loanno,cname,bname,amount)

|  |  |
| --- | --- |
| COLUMN NAME | DATA TYPE |
| loanno | Int |
| cname | Varchar(25) |
| bname | Varchar(20) |

|  |  |
| --- | --- |
| amount | Decimal(6,2) |

CODE :-

CREATE TABLE borrow

( loanno INT PRIMARY KEY, cname VARCHAR(25),

bname VARCHAR(20),

amount DECIMAL(6,2)

);

# PRACTICAL-15

**AIM** :- Create tables and insert sample data in tables.

Write SQL queries to insert following data into tables

Insert following values in the table **Employee.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **emp\_n** | **emp\_name** | **emp\_sal** | **emp\_comm** | **dept \_no** |
| 101 | Smith | 800 |  | 20 |
| 102 | Snehal | 1600 | 300 | 25 |
| 103 | Adama | 1100 | 0 | 20 |
| 104 | Aman | 3000 |  | 15 |
| 105 | Anita | 5000 | 50000 | 10 |
| 106 | Sneha | 2450 | 24500 | 10 |
| 107 | Anamika | 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, 'Smith', 800.00, 20, 0),

(102, 'Snehal', 1600.00, 300, 25),

(103, 'Adama', 1100.00, 0, 20),

(104, 'Aman', 3000.00, 15, 0),

(105, 'Anita', 5000.00, 50000, 10),

(106, 'Sneha', 2450.00, 24500, 10),

(107, 'Anamika', 2975.00, 30, 0);

**AIM** :- Create tables and insert sample data in tables.

Write SQL queries to insert following data into tables

Insert following values in the table **Job.**

|  |  |  |  |
| --- | --- | --- | --- |
| job\_id | **job\_name** | **min\_sal** | **max\_sal** |
| IT\_PROG | Programmer | 4000 | 10000 |
| MK\_MGR | Marketing manager | 9000 | 15000 |
| FI\_MGR | Finance manager | 8200 | 12000 |
| FI\_ACC | Account | 4200 | 9000 |
| LEC | Lecturer | 6000 | 17000 |
| COMP\_OP | Computer Operator | 1500 | 3000 |

**CODE :-** CREATE TABLE Job (

job\_id VARCHAR(15) PRIMARY KEY, job\_name VARCHAR(30) NOT NULL,

min\_sal DECIMAL(10,2) NOT NULL,

max\_sal DECIMAL(10,2) NOT NULL

);

INSERT INTO Job (job\_id, job\_name, min\_sal, max\_sal) VALUES ('IT\_PROG', 'Programmer', 4000.00, 10000.00),

('MK\_MGR', 'Marketing manager', 9000.00, 15000.00),



('FI\_MGR', 'Finance manager', 8200.00, 12000.00),

('FI\_ACC', 'Account', 4200.00, 9000.00),

('LEC', 'Lecturer', 6000.00, 17000.00),

('COMP\_OP', 'Computer Operator', 1500.00, 3000.00);

**AIM** :- Create tables and insert sample data in tables.

Write SQL queries to insert following data into table.

Insert following values in the table **deposit**.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A\_no** | **cname** | **Bname** | **Amount** | **date** |
| 101 | Anil | andheri | 7000 | 01-jan-06 |
| 102 | sunil | virar | 5000 | 15-jul-06 |
| 103 | jay | villeparle | 6500 | 12-mar-06 |
| 104 | vijay | andheri | 8000 | 17-sep-06 |
| 105 | keyur | dadar | 7500 | 19-nov-06 |
| 106 | mayur | borivali | 5500 | 21-dec-06 |

## CODE :-

CREATE TABLE

deposit ( a\_no INT IDENTITY PRIMARY KEY,

cname VARCHAR(50) NOT NULL,

bname VARCHAR(30) NOT NULL, amount DECIMAL(8,2) NOT NULL,

a\_date DATE NOT NULL

);

INSERT INTO deposit (cname, bname, amount, a\_date) VALUES ('Anil', 'andheri', 7000.00, '2006-01-01'),

('sunil', 'virar', 5000.00, '2006-07-15'),

('jay', 'villeparle', 6500.00, '2006-03-12'),

('vijay', 'andheri', 8000.00, '2006-09-17'),

('keyur', 'dadar', 7500.00, '2006-11-19'),

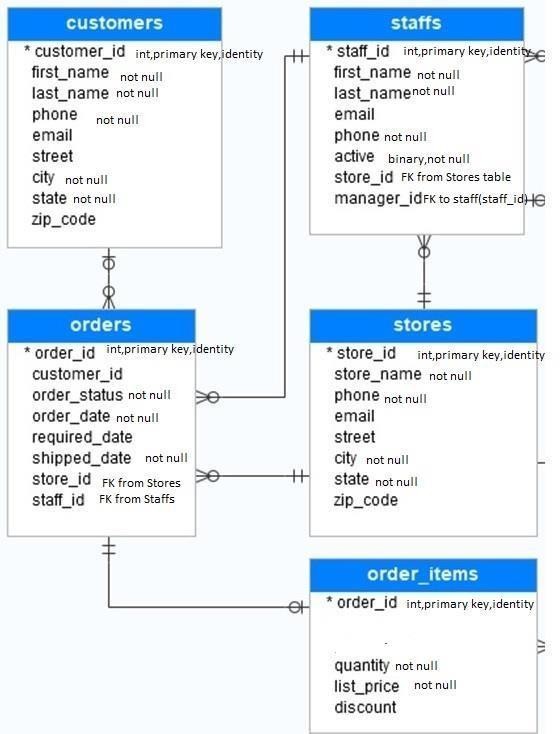
('mayur', 'borivali', 5500.00, '2006-12-21');



# PRACTICAL-16

**AIM** :- Write the SQL queries to provide constraints on given tables.

Create A Database Sales and Write SQL Queries to create following tables with all constrains mentioned in image.



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\_i d 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-17

**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;

1. List total amount from andheri branch

**CODE :-** SELECT SUM(amount) AS total\_amount FROM deposit WHERE bname = 'andheri';

1. Count total number of customers

**CODE :-** SELECT COUNT(DISTINCT cname) AS total\_customers FROM deposit;

1. Count total number of customer‟s cities

**CODE :-** SELECT COUNT(DISTINCT bname) AS total\_cities FROM deposit;

1. 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%';

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

**CODE :-** UPDATE Employee SET emp\_name = 'Adam' WHERE emp\_no = 103;

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

**CODE :-** SELECT GETDATE() AS Date;

1. 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;

1. 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-18

**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;

* 1. 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';

* 1. Display all jobs with minimum salary is greater than 4000.

CODE:- SELECT \* FROM

jobs WHERE min\_sal > 4000;

* 1. 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;

* 1. 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);

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

**CODE :-** SELECT \* FROM employee WHERE emp\_name LIKE 'A

\_a%';

* 1. 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 ';

* 1. 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;

* 1. 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;

# PRACTICAL-19

**AIM** :- Make a Resume using the HTML tags without CSS.

### CODE :-

<!DOCTYP

E html>

<html>

<head>

<title>My Resume</title>

</head>

<body>

<h1>Guddu Giri</h1>

<div>Website designer</div>

<di[v>guddugiri216@gmail.com](mailto:guddugiri216@gmail.com)</div>

<div>7985248023</div>

<div>Ahmedabad, Gujarat</div>

<h3>Objective</h3>

<p>To obtain a challenging position as a software engineer where I can utilize my skills and knowledge to develop innovative solutions for complex problems.</p>

<h3>Education</h3>

<ul>

<li>B.Tech(IT), Aditya Silver oak Institute of Technology, 2022</li>

<li>12<sup>th</sup> Gurukul Mission Sr.sec. School UP 2020-2021</li>

<li>10<sup>th</sup> Smart Move Academy , 2018-2029</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>

</ul>

<li>English</li>

<li>Hindi</li>

<h3>Hobbies</h3>

<ul>

</ul>

<li>Writing</li>

<li>Cooking</li>

<li>Network marketing</li>

<h4>Certificate</h4>

<ul>

<li>Artificial Intelligence</li>



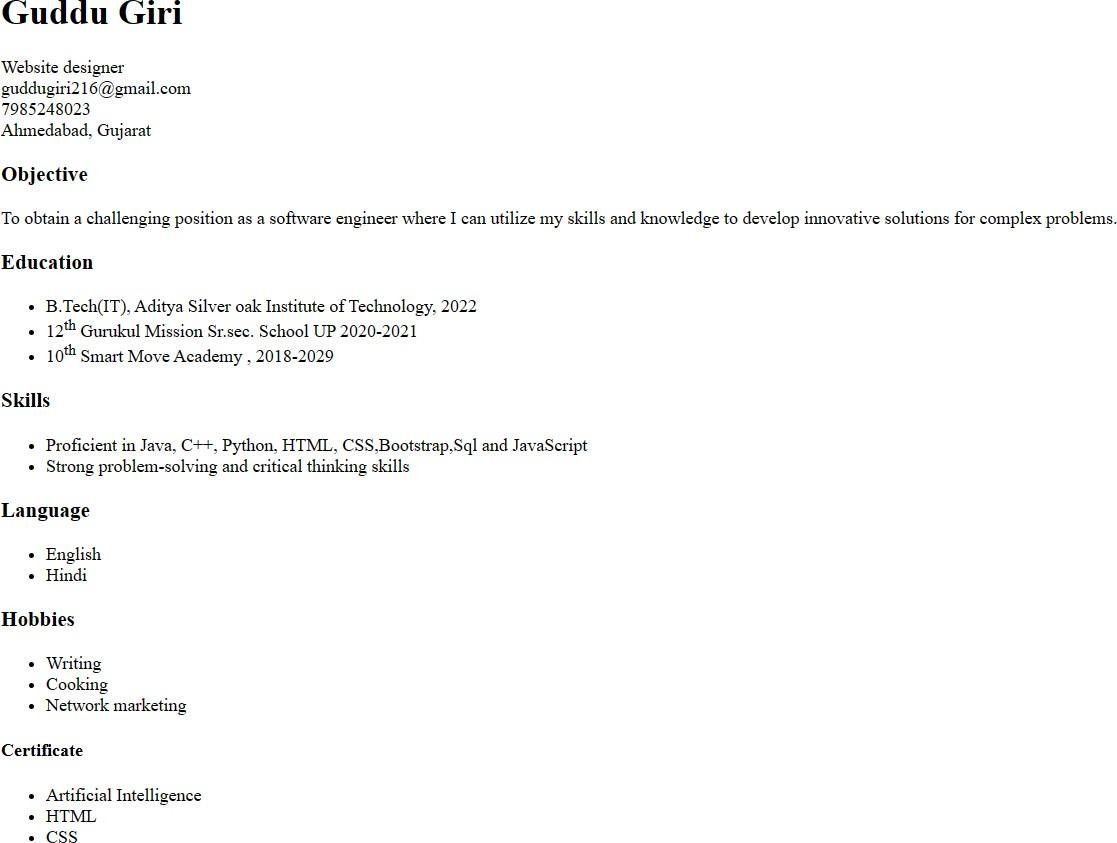
<li>HTML</li>

<li>CSS</li>

</ul>

</body>

</html>



# PRACTICAL-20

**AIM** :- Create an HTML webpage that shows Poster Presentation using all Table Properties.

### CODE :-

<!DOCTYP

E html>

<html>



<head>

<title>Poster Presentation</title>

<style>

table {

border-collapse:

collapse; width: 100%;

}

th, td {

}

th {

}

border: 1px solid black; padding: 8px; text-align: left;

background-color: #f2f2f2;

.highlight {

background-color: yellow;

}

</style>

</head>

<body>

<h1>Poster Presentation</h1>

<table>

<thead>

<tr>

</tr>

<th>Presenter</th>

<th>Title</th>

<th>Date</th>



</thead>

<tbody>

<tr>

</tr>

<tr>

</tr>

<tr>

</tr>

</tbody>

</table>

<td>Vijaya Raghavan</td>

<td>Effects of Exercise on Mental Health</td>

<td>Sep 3, 2019</td>

<td>Pratyasha Jain</td>

<td>Impact of Social Media on Adolescents</td>

<td>March 1, 2023</td>

<td>K. Vijayaraghavan</td>

<td>The Role of Nutrition in Aging</td>

<td>Sep 22, 2008</td>

<p>Here are some key takeaways from the presentations:</p>

<table>

<tr>

</tr>

<tr>

<th>Presenter</th>

<th>Key Takeaway</th>

<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>



</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>

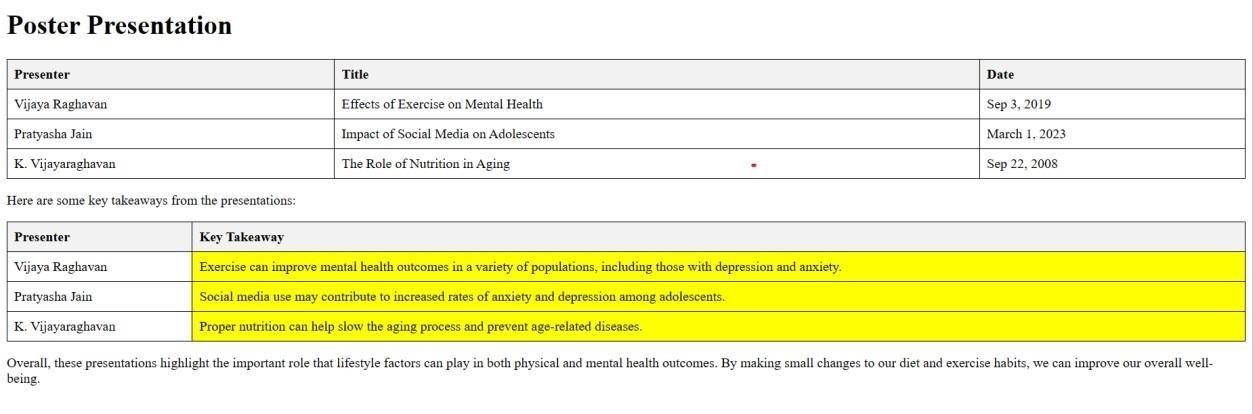
</table>

<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 :-



# PRACTICAL-21

**AIM** :- Create an HTML page table and form

**CODE :-** <!DOCTYPE html>

<html>

<head>

<title>Table and Form with CSS</title>

<style>

/\* Table Styles

\*/ table {

border-collapse: collapse; width: 100%;

}

th, td

{ text- align:

left; padding

:

8px;

border-bottom: 1px solid #ddd;

}

th {

background-color: #f2f2f2;

}

/\* Form Styles

\*/ 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="submi t "] { 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; margi n- 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>Guddu giri</td>

<td[>guduxt@example.com](mailto:guduxt@example.com)</td>

<td>8955858555</td>

</tr>

<tr>

<td>Smit Gajera</td>

<t[d>gajera@ex](mailto:gajera@example.com)a[mple.com](mailto:gajera@example.com)</td>



<td>7985248023</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>

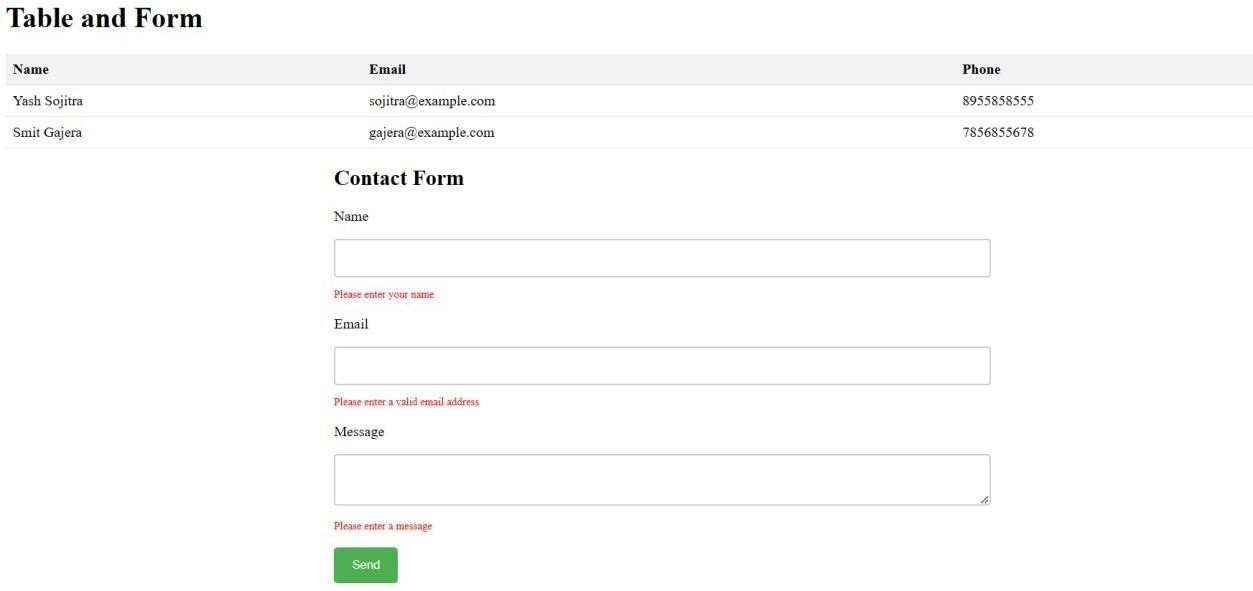
</div>

<input type="submit" value="Send">

</form>

</body>

</html>

OUTPUT :-

# PRACTICAL-22

**AIM** :- Create Registration form and do proper validation with HTML 5 inbuilt functionality. (Don‟t use JavaScript).

### CODE :-

<!DOCTYP

E html>

<html>

<head>

<title>Table and Form with CSS</title>

<style>

/\* Table Styles

\*/ table {

border-collapse: collapse; width: 100%;

}

th, td

{ text- align: left; padding: 8px;

}

th {

background-color: #f2f2f2;

}

/\* Form Styles \*/ 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;

}

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>Yash Sojitra</td>

<td[>sojitra@example.com](mailto:sojitra@example.com)</td>



<td>8955858555</td>

</tr>

<tr>

<td>Smit Gajera</td>

<td[>gajera@ex](mailto:gajera@example.com)a[mple.com](mailto:gajera@example.com)</td>

<td>7856855678</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>

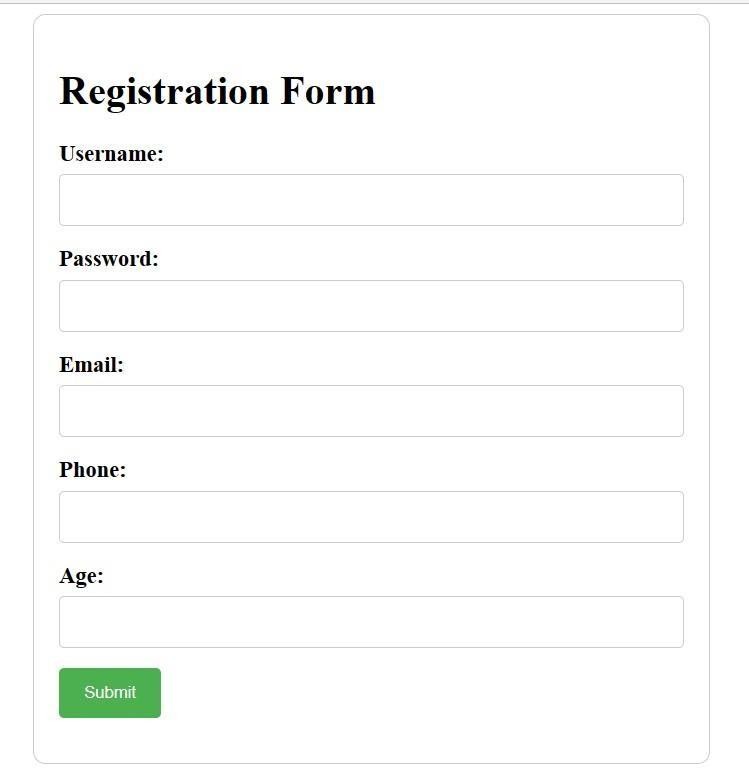
<span class="error">Please enter a message</span>

</div>

<input type="submit" value="Send">

</body>

</html>

OUTPUT :-

# PRACTICAL-23

**AIM** : Make a Resume using the HTML tags with CSS.

### CODE :-

<!DOCTYP

E html>

<html>

<head>

<style

>

body

{

font-family: Arial, sans-serif;

font- size: 16px; line- height: 1.6;

color: #333;

}

h1, h2, h3

{ margin-top: 0;

}

h1 { font- size: 36px; color: #0077b 5;

}

h2 { font- size: 28px; color: #222;

}

h3 { font- size: 24px; color: #555;

}

p {

i n:

0;

.container

{ max-width: 800px; margin: 0 auto; padding:

20px;

}

.header { text-align:

center; margin- bottom: 30px;

}

.header h1 { margin-bottom: 10px;

}

.contact-info { margin-bottom: 30px;

}

.contact-info p { margin-bottom: 10px;

}

.section { margin- bottom: 30px;

}

.section h2 {

margin-bottom: 20px;

}

{ list-style: none; margin: 0;

padding: 0;

}

.section li {

margin-bottom: 10px;

}

.section li span

{ display: inline-block; width: 150px;

font-weight: bold;

}

</style>

</head>

<body>

<div class="container">

<div class="header">

<h1>Guddu giri</h1>

<p>Web Developer</p>

</div>

<div class="contact-info">

<p><strong>Email:</strong> [guddugiri216@example.com](mailto:guddugiri216@example.com)</p>

<p><strong>Phone:</strong> 7985248023</p>

<p><strong>Website:</strong> crazycodes.com</p>

</div>

<div class="section">

<h2>Summary</h2>

<p>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.</p>

</div>

<div class="section">

<h2>Skills</h2>

<ul>

<li><span>HTML:</span> Advanced</li>

<li><span>CSS:</span> Advanced</li>

<li><span>JavaScript:</span> Advanced</li>

<li><span>React:</span> Intermediate</li>

<li><span>Angular:</span> Intermediate</li>

<li><span>Vue:</span> Beginner</li>

</ul>

</div>

<div class="section">

<h2>Experience</h2>

<ul>

<li>

<span>Web Developer</span>

<p>Google corp</p>

<p>January 2023 - Present</p>

<ul>

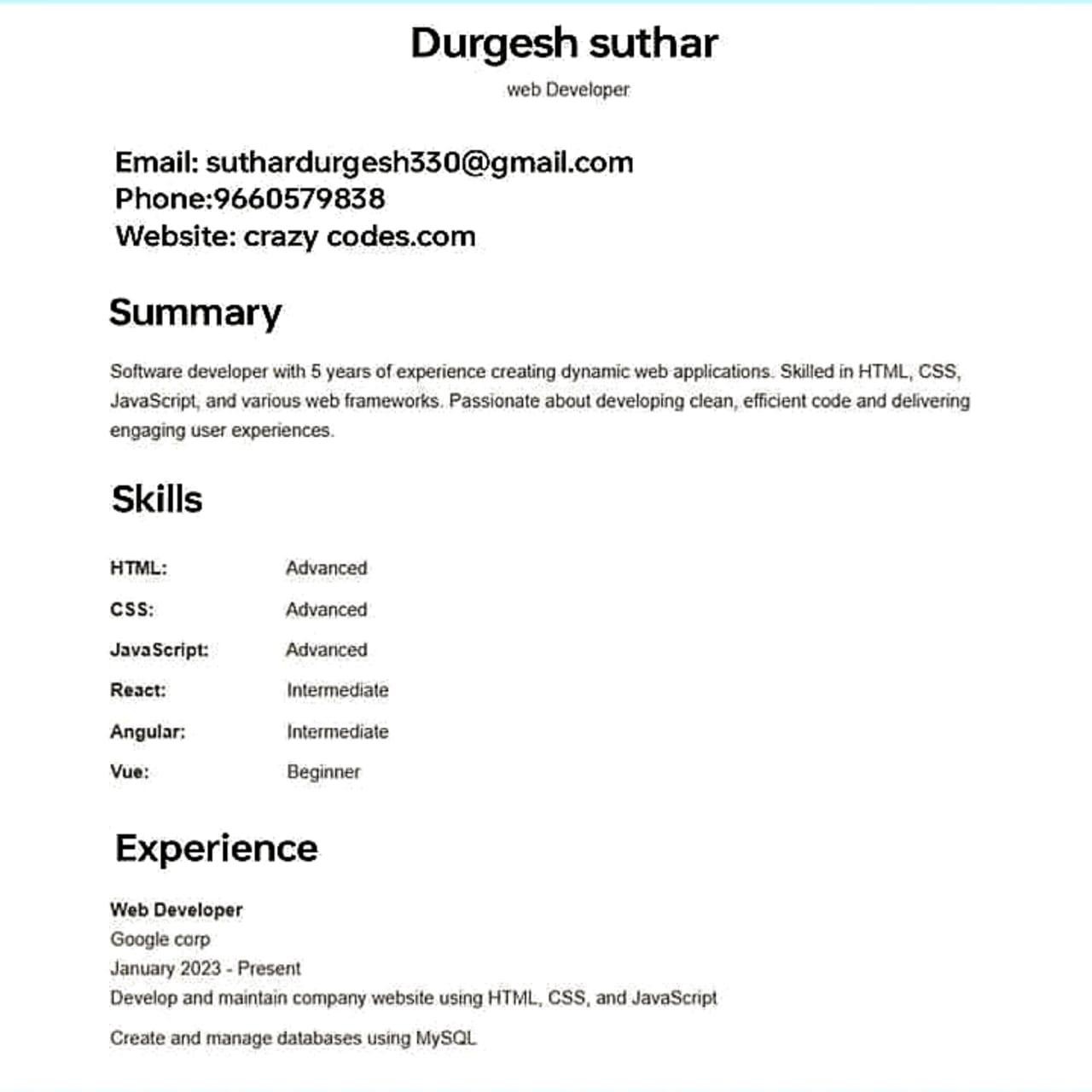
<li>Develop and maintain company website using HTML, CSS, and JavaScript</li>

<li>Create and manage databases using MySQL

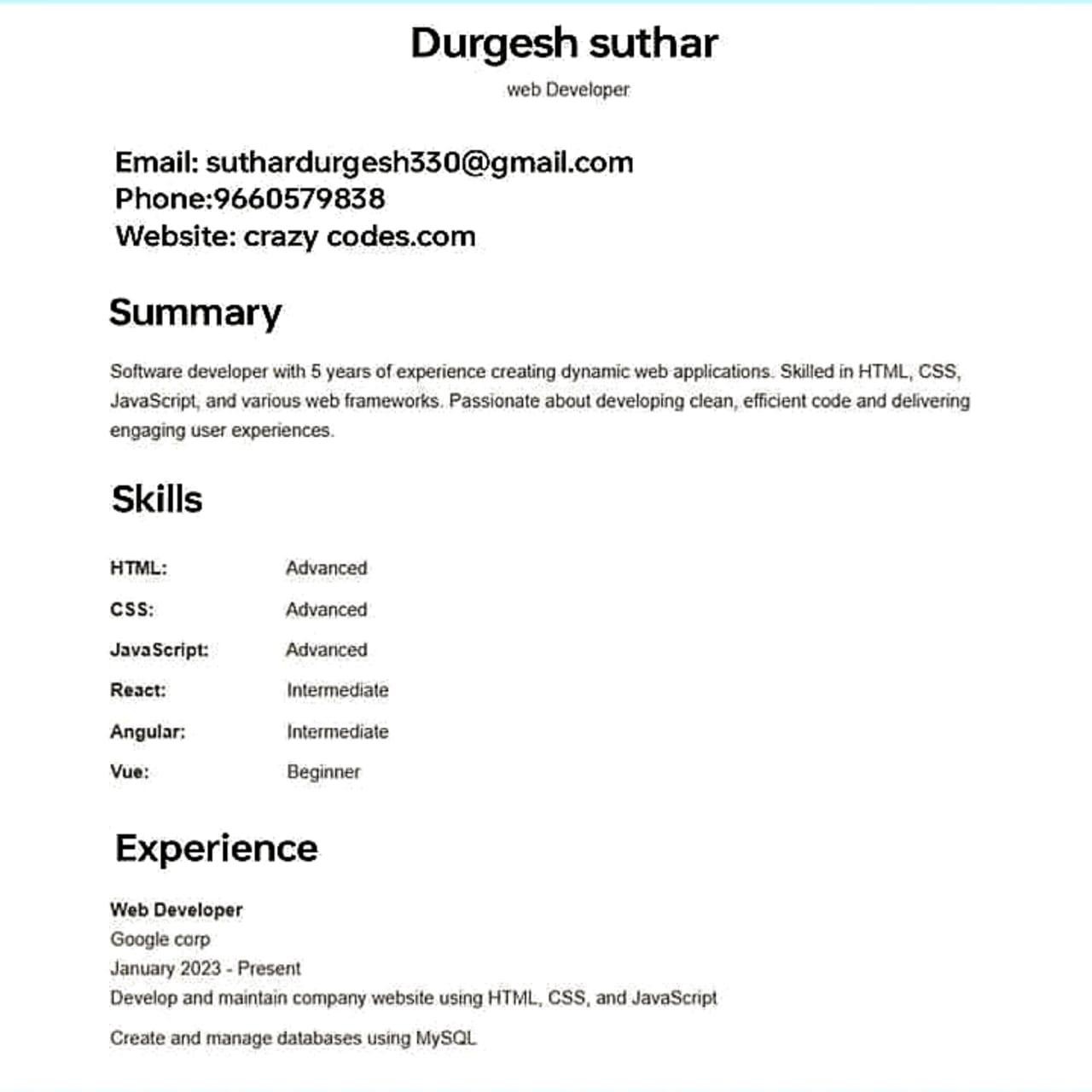
</body>

</html

> OUTPUT:-



# PRACTICAL-24



**AIM** :- Create an HTML Page containing the following Gray Layout using CSS

### CODE :-

<!DOCTYP

E 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>Gray layout 1</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>

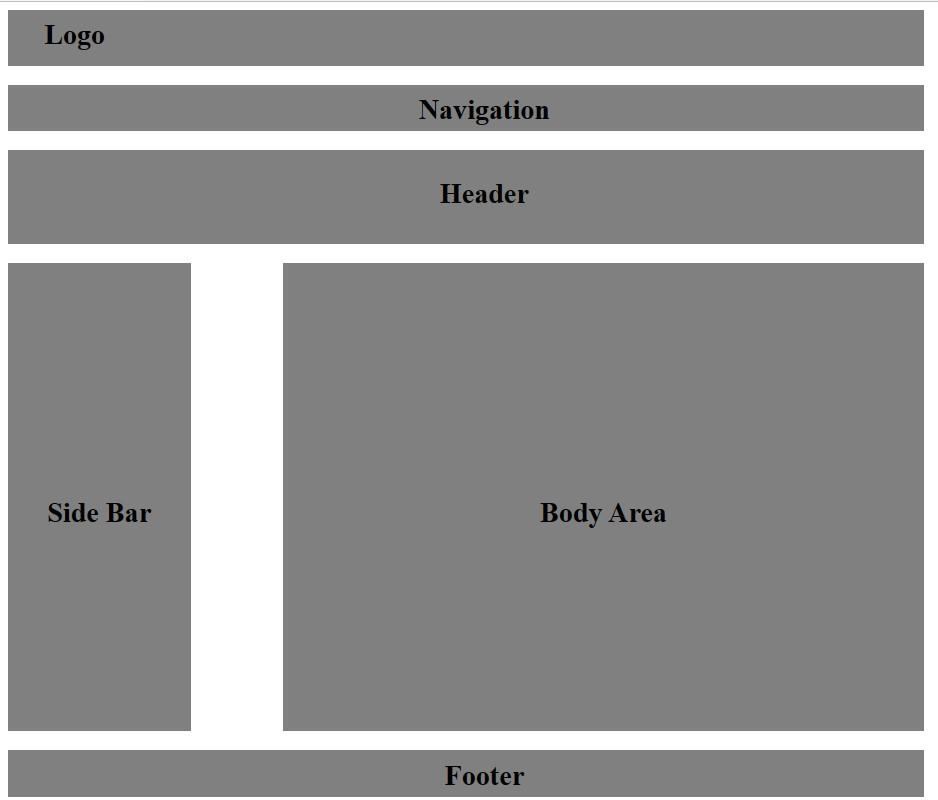
</div>

<div id="r5">Footer</div>

</div>

</body>

</html>

OUTPUT :-

### CODE :-

<!DOCTYP

E 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>Gray Layout 2</title>

<style>

#r1{background-color: gray; width: 78%;padding-top: 15px;padding-left: 2%; margin- left: 10%; height: 40px;}

#r1c1{float: left; font-size: larger;}



#r1c2{margin-left:80%;padding-top: 5px;}

#r2, #r3{width: 80%;margin-left: 10%;height: 60px;padding-top: 30px; background- color: gray;margin-top: 10px; text-align: center;}

#r4 div{float: left; background-color: gray; font-weight: bolder;} #r4c1{margin-left: 10%;text-align: center;width: 25%;height:

100px;margin-top: 10px;paddingtop: 100px;}

#r4c2{margin-left: 2.5%;text-align: center;width: 25%;height: 100px;margin-top: 10px;paddingtop: 100px;}

#r4c3{margin-left: 2.5%;text-align: center;width: 25%;height: 100px;margin-top: 10px;paddingtop: 100px;}

#r5{width: 80%;margin-left: 10%;background-color: graY;margin-top: 220px;height: 30px;padding-top: 20px;text-align: center;}

</style>

</head>

<body>

<div id="r1">

<div id="r1c1">Logo</div>

<div id="r1c2">Navigation</div>

</div>

<div id="r2">Header</div>

<div id="r3">Intro Text Area</div>

<div id="r4">

<div id="r4c1">Box 1</div>

<div id="r4c2">Box 2</div>

<div id="r4c3">Box 3</div>

</div>

<div id="r5">Footer</div>

</body>

</html>

## OUTPUT :-

### CODE :-

<!DOCTYP

E 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>Gray Layout 3</title>

<style>

#r1{padding-top: 30px;height: 30px; margin-top: 10px;margin-left: 10%;width: 78%;paddingleft: 2%; background-color: gray;font-size: larger;}

#r2 div{float: left;margin-left: 10%;margin-top: 10px;height: 40px;padding-top: 40px; text-align: center;}

#r2c1{ width: 20%;margin-right: 5%;font-size:

large;} #r2c2{width: 45%; background-color:

gray;}

#r3{margin-bottom: 10px; margin-left: 45%;width: 45%; height: 250px;padding-top: 250px;textalign: center;background-color: gray;margin-top: 100px;}

#r4{margin-left: 10%; width: 80%;text-align: center;padding-top: 20px;height: 30px;backgroundcolor: gray;}



</style>

</head>



<body>

<div id="r1">Logo</div>

<div id="r2">

<div id="r2c1">Navigation</div>

<div id="r2c2">Header</div>

</div>

<div id="r3">Body Area</div>

<div id="r4">Footer</div>

</body>

</html>

## OUTPUT :-

### CODE :-

<!DOCTYP

E 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>Gray Layout 4</title>

<style>

#r1{background-color: gray; width: 78%;padding-top: 15px;padding-left: 2%; margin- left: 10%; height: 40px;}

#r1c1{float: left; font-size: larger;} #r1c2{margin-left:80%;padding-top:

5px;}

#r3{margin-bottom: 10px;height: 100px; margin-left: 10%;margin-top: 10px;} #c1,#c2,#c3{float: left; background-color: gray; width: 10%;margin-right:

15%; text- align: center;}

#c5,#c6,#c7{float: left;background-color: gray; width: 10%; margin-right: 15%;text-align: center;} #c9,#c10,#c11{float: left;background-color: gray; width: 10%; margin-right: 15%;text-align:

center;}

#c13,#c14,#c15{float: left;background-color: gray; width: 10%; margin-right: 15%;text- align:

center;}

#r4 div{float: left; background-color: gray; font-weight: bolder;} #r4c1{margin-left: 10%;text-align: center;width: 25%;height:

100px;margin-top: 10px;paddingtop: 100px;}

#r4c2{margin-left: 2.5%;text-align: center;width: 25%;height: 100px;margin-top: 10px;paddingtop: 100px;}

#r4c3{margin-left: 2.5%;text-align: center;width: 25%;height: 100px;margin-top: 10px;paddingtop: 100px;}

#r5{width: 80%;margin-left: 10%;background-color: graY;margin-top: 250px;height: 30px;padding-top: 20px;text-align: center;}

</style>

</head>

<body>

<div id="r1">



<div id="r1c1">Logo</div>

<div id="r1c2">Navigation</div>

</div>

<div>

<div></div>

</div>

<div id="r3">

<div id="c1">1</div>

<div id="c2">2</div>

<div id="c3">3</div>

<div id="c4">4</div><br>

<div id="c5">5</div>

<div id="c6">6</div>

<div id="c7">7</div>

<div id="c8">8</div><br>

<div id="c9">9</div>

<div id="c10">10</div>

<div id="c11">11</div>

<div id="c12">12</div><br>

<div id="c13">13</div>

<div id="c14">14</div>

<div id="c15">15</div>

<div id="c16">16</div>

</div>

<div id="r4">

<div id="r4c1">Box 1</div>

<div id="r4c2">Box 2</div>

<div id="r4c3">Box 3</div>

</div>

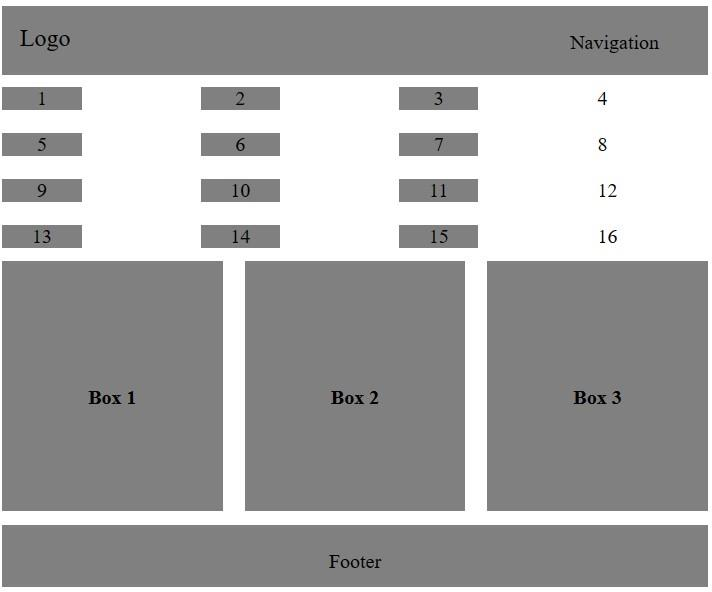


<div id="r5">Footer</div>

</body>

</html>

OUTPUT:-



# PRACTICAL-25

**AIM** :- Demonstrate JavaScript Form Validation with proper examples

### CODE :-

<!DOCTYP

E html>

<html>



<head>

<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>

</head>

<body>

<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 :-

# PRACTICAL-26

**AIM** :- Write a javascript to check if the number is even or odd.

### CODE :-

<!DOCTYP

E 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>

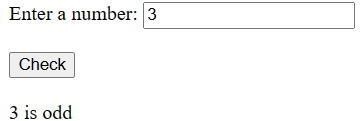
<button onclick="checkEvenOrOdd()">Check</button>

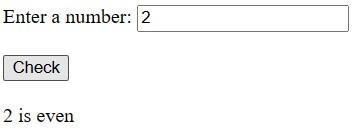
<br><br>

<div id="result"></div>

</body>

</html>

OUTPUT :-



# PRACTICAL-27

**AIM** :- Create a page and access the LocationAPI

### CODE :-

<!DOCTYP

E html>

<html>

<head>

<title>Location API Example</title>

<script>



function getLocation()

{

if (navigator.geolocation)

{

navigator.geolocation.getCurrentPosition(showPosition);

}

e l s e

{

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>

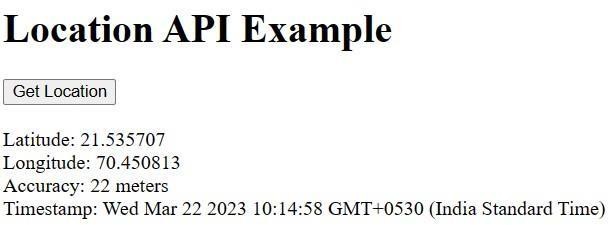
<div id="longitude"></div>

<div id="accuracy"></div>

<div id="timestamp"></div>

</body>

</html>

OUTPUT :-

# PRACTICAL-28

**AIM** :- Create a simple XMLHTTPRequest,and retrieve the data from the text file.

### CODE :-

<!DOCTYP

E 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;

}

el s e

{

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 :-



THANK YOU