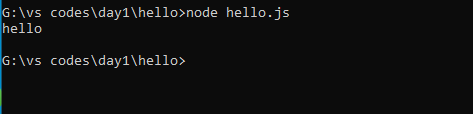
**Day 1**

1. Install node js .

2. Create simple helloworld node js app and test.

console.log("hello");



3. Create new lib.js file with Display, show and addition, subtraction , multiplication function implementation

function show(){

 var count=67;

 count++;

 console.log(count);

}

function addition(num1,num2)

{

    var result=num1+num2;

    return result;

}

function substraction(num1,num2)

{

    var result1=num1-num2;

    return result1;

}

function multiplication(num1,num2)

{

    var result2=num1\*num2;

    return result2;

}

function display(company){

    console.log( "Company name = " + company);

};

show();

var number1=20;

var number2=30;

var total=addition(number1,number2);

console.log(total);

substraction(5,2);

multiplication(2,3);

var provider="ibm.pvt.ltd,pune";

display(provider);

4. Test each functions using NOdejs command



function processJob3(){

    console.log("Processing  request by Handler 3");

    console.log("Performing task 3");

    console.log("Generating Response 3");

    console.log("Sending Response 3");

};

//Nonbloking function

function processJob2(){

    console.log("Processing  request by Handler 2");

    console.log("Performing task 2");

    console.log("Generating Response 2");

    console.log("Sending Response 2");

};

function processJob1(){

    console.log("Processing  request by Handler 1");

    console.log("Performing task 1");

    console.log("Generating Response 1");

    console.log("Sending Response 1 ");

}

setInterval(processJob1,500);

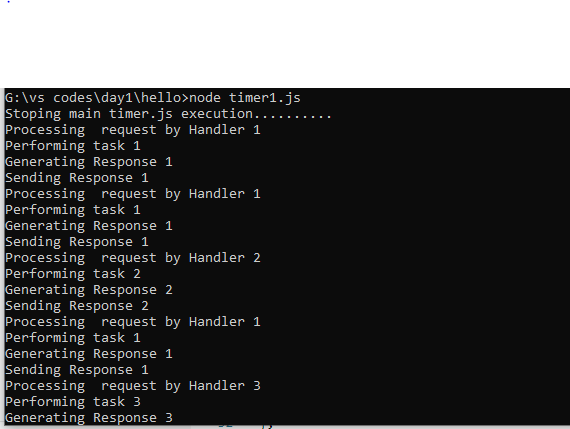
setInterval(processJob2,1000);

setInterval(processJob3,2000);

console.log("Stoping main timer.js execution..........")

5. Implement simple Timer logic and test its execution.

6. Explore Node JS Application Execution



**Day 2**

1)Banking APP:

1)handler.js

//Modularity Event handling Module

//Event Listeners

//Event handler logic

//expose functions to be used outside of Javascript file

// we are publishing functions outside by using exports keyword

exports.payIncomeTax=function(){

    console.log("25% income tax to be applied against account");

}

exports.payServiceTax=function(){

    console.log("18% Service tax to be applied against account");

}

exports.blockAccount=function(){

    console.log("The Account has been blocked Temp");

};

exports.sendEmail=function(){

    console.log("Email is sent to registered email ID");

};

exports.sendSMS=function(){

    console.log("SMS is sent to registered phone number");

};

2)account.js

//Domain: Banking

//Operations: Credit, Debit,

//Business Entity: Account

//Rule: Contstraints, Policy

//Events against Account

//Account:

        //state: Balance

        //operations: withdraw, deposit

        //Rule: Balance threshold

        //Event:underBalance, overBalance

//classical Javascript syntax

//SOC:Seperation of Cocern

//Loosely Coupled , Highly Cohesive

var handlers=require("./handler"); // custom module

var events=require("events");     // builtin module available in nodejs

var emitter=new events.EventEmitter();

var Account=function(amount){

     var balance=amount;

     //Inner function

     var getBalance=function(){

         return balance;

     };

     var monitor=function(){

        if(balance < 5000){

           // handlers.blockAccount();

            //emit (trigger) event underBalance

            emitter.emit("underBalance");

        }

        else if(balance>=250000){

            //handlers.payIncomeTax();

             // emit (trigger event overBalance)

             emitter.emit("overBalance");

        }

     };

     var withdraw=function(amount){

         balance=balance-amount;

         monitor();

     };

     var deposit=function(amount){

        balance=balance+amount;

        monitor();

     }

     //publish inner functions to outside world

     return{

         receiveBalance:getBalance,

         debit:withdraw,

         credit:deposit

     }

}

emitter.on("underBalance",handlers.blockAccount);

emitter.on("underBalance",handlers.sendEmail);

emitter.on("underBalance",handlers.sendSMS);

emitter.on("overBalance",handlers.payServiceTax);

//Step 2: Create objects

var acct123=new Account(55000);                 // creating an instance

var acct124=new Account(5000);

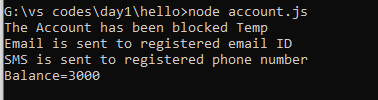
//var data=acct123.receiveBalance();              // invoking member function

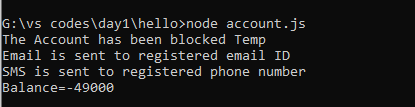
//step 3 : Start invoking their operations

acct123.debit(52000);//

var data=acct123.receiveBalance();

console.log("Balance=" + data);





2)html app:

// Custom HTTP Web server using Node JS

var http=require('http');

// This is a rule for web programming

// To define first parameter always as request object

//           second prameter always as  response object

//One JSON object

var person={

        'firstname':'Prathamesh',

        'lastname':'Raut',

        'email':'renushelar50@gmail.com',

        'contactnumber':'9881735801'

};

//Array of JSON objects

var people=[

    {'firstname':'Ravi','lastname':'Tambade','email':'renushelar50@gmail.com','contactnumber':'9892536646'},

    {'firstname':'Sachin','lastname':'Barman','email':'renushelar50@gmail.com','contactnumber':'91111737801'},

    {'firstname':'Suresh','lastname':'Jadhav','email':'renushelar50@gmail.com','contactnumber':'9881735801'}

];

var data="Transflower Learning Pvt. Ltd.";

// custom (Programmer Defined Function)

// HTTP Request Handler function

var onRequestHandler=function(request, response){

  console.log("Request is received.....");  //inbuit functions

  //This code will send JSON data as a response to Client

  response.writeHead(200,{'Content-Type':"text/json"});

  // Serialization --Deserialization

   response.write(JSON.stringify(people));

  //This code will send HTML data  as  a response to Client

 // response.writeHead(200,{'Content-Type':"text/html"});

  //response.write("<h1>Welcome to the real programming world</h1>");

  response.end();

};

var server=http.createServer(onRequestHandler);  // inbuilt function

server.listen(1154);  //0 to 1024 port number    // inbuilt function

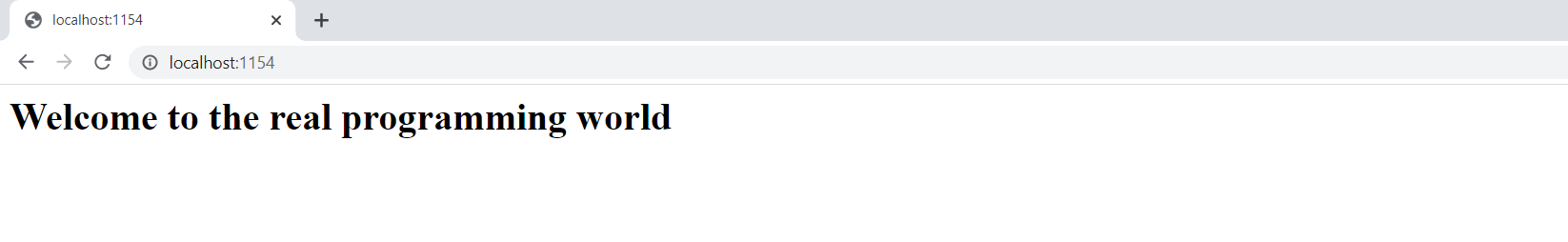
console.log("Server is listening on Port no. 1154");

//Serialization (Encoding)

// convert json object into string

//DeSerailization (Decoding)

//convert string into json object

****

**Day 3**

1.Create simple Node JS Application to read text file using file IO.

var fs=require('fs');

var fileName="./data/transflower.txt";

//callback function

//

var onFileRead=function(err, data){

    if(err){

      throw err;

    }

    console.log("Data from File available");

    console.log(data.toString());

};

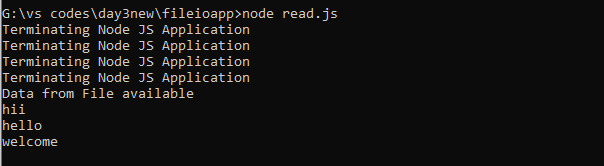
fs.readFile(fileName,onFileRead);     //Non Blokcing call

console.log("Terminating Node JS Application");

console.log("Terminating Node JS Application");

console.log("Terminating Node JS Application");

console.log("Terminating Node JS Application");



2.Create simple Node JS Application to write content to text file using file IO.

var fs= require('fs');

var fileName="./data/result.txt";

var dataTobeWritten=" This data is written by Node JS Application....";

var onFileWrite=function(err){

    if(err){

        throw err;

    }

    console.log("Contnet has been successfully written into file");

};

// first parameter: path of the file

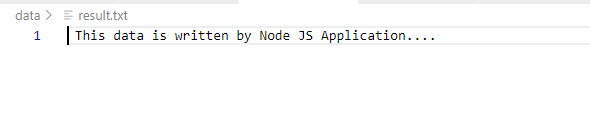
// second parameter:data to be written

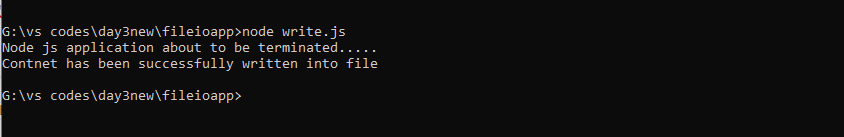
// third parameter:callback function name, which will be called atuomatically

// after write operation is done by system

fs.writeFile(fileName,dataTobeWritten,onFileWrite);

console.log("Node js application about to be terminated.....");





3.Create simplle Node JS Application for Transflower Store to show index.html file.

a.add new index.html file to node js application folder.

b.build simple index.html file using basic html elements such as html, head, body, h1, a &ol

|  |  |
| --- | --- |
|  | <!DOCTYPE html> |
|  | <html> |
|  | <head> |
|  | <title>renu website </title> |
|  | </head> |
|  | <body> |
|  | <h2> new start </h2> |
|  | <hr/> |
|  | <a href="[index.html](file:///G:\vs%20codes\day3\fileapp\data\index.html)"> home</a> |
|  | <hr/> |
|  | <ol> |
|  | <li>udit</li> |
|  | <li>radha</li> |
|  | </ol> |
|  | </body> |
|  | </html> |

c.add server.js file.

d.import http and fs modules.

var http=require('http');   //Step 1

var fs =require('fs');      //Step 2

var onRequestHandler=function(request, response){    //Step 4

    var pathname="./"+request.url;

    console.log(pathname);

    fs.readFile(pathname,(err,data)=>{              //Step 5

        if(err){

            console.log("Something has gone wrong");

            console.log(err);

            response.writeHead(404,{'Content-Type':'text/html'});

        }

        else{

            response.writeHead(200,{'Content-Type':'text/html'});

            response.write(data.toString());

        }

        response.end();

    });

};

var server=http.createServer(onRequestHandler);   //Step 3

server.listen(2200); //avoid port no. 0 to 1024   //Step 6

console.log("HTTP Server is listening on port number 2200");

e.CreateHttpserver

f.Implement Callback function onReuestHandler to send html file based on url provided.

g.Keep server in listen mode

// Custom HTTP Web server using Node JS

var http=require('http');

// This is a rule for web programming

// To define first parameter always as request object

//           second prameter always as  response object

//One JSON object

var person={

        'firstname':'sonu',

        'lastname':'raut',

        'email':'radhika@transflower.in',

        'contactnumber':'978900034'

};

//Array of JSON objects

var people=[

    {'firstname':'sonu','lastname':'raut','email':'radha.punekar@transflower.in','contactnumber':'9881735801'},

    {'firstname':'Sachin','lastname':'Barman','email':'barman.tambade@transflower.in','contactnumber':'9881735801'},

    {'firstname':'Suresh','lastname':'Jadhav','email':'ravi.jadhav@transflower.in','contactnumber':'9881735801'},

    {'firstname':'Ramesh','lastname':'Verma','email':'verma.tambade@transflower.in','contactnumber':'9881735801'}

];

var data="Transflower Learning Pvt. Ltd.";

// custom (Programmer Defined Function)

// HTTP Request Handler function

var onRequestHandler=function(request, response){

  console.log("Request is received.....");  //inbuit functions

  //This code will send JSON data as a response to Client

  response.writeHead(200,{'Content-Type':"text/json"});

  // Serialization --Deserialization

   response.write(JSON.stringify(people));

  //This code will send HTML data  as  a response to Client

  //response.writeHead(200,{'Content-Type':"text/html"});

  //response.write("<h1>Welcome to the real programming world</h1>");

  response.end();

};

var server=http.createServer(onRequestHandler);  // inbuilt function

server.listen(7788);  //0 to 1024 port number    // inbuilt function

console.log("Server is listening on Port no. 7788");

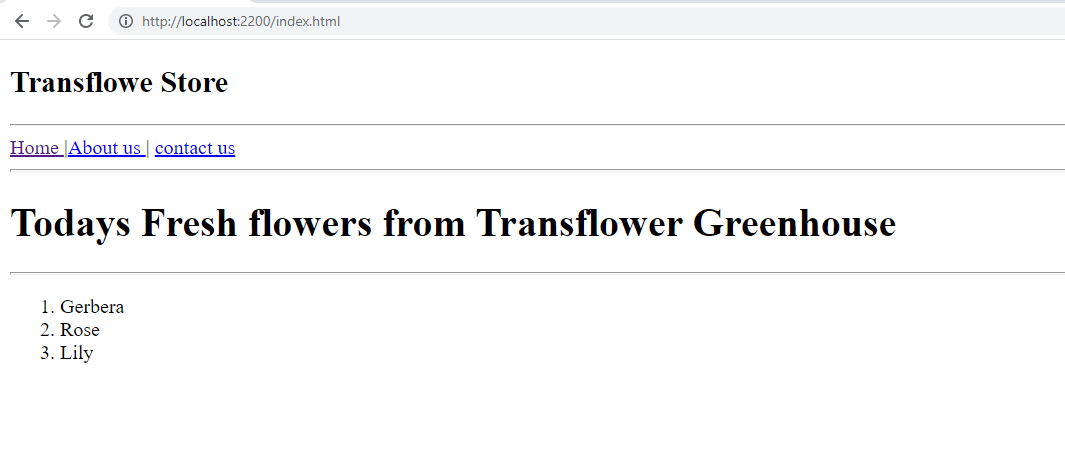
//Serialization (Encoding)

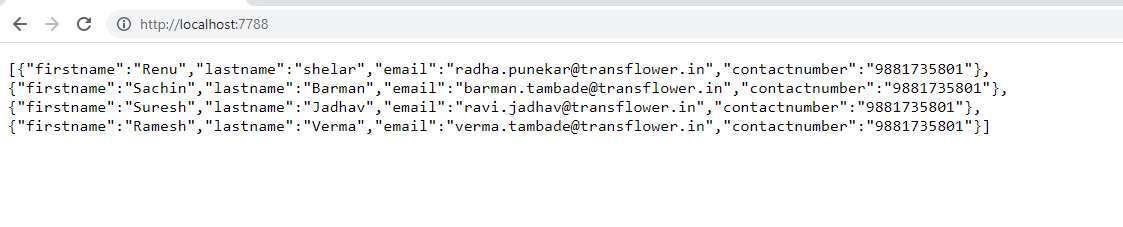
// convert json object into string

//DeSerailization (Decoding)

//convert string into json object

4.Run and Test Node Js Application.





5.Add two web pages about.html and contact.html to show basic information using paragraph and h1

1)About.html

<!DOCTYPE html>

<html>

    <head>

        <title>Transflower Store</title>

    </head>

    <body>

        <h1>Transflower Store</h1>

        <hr/>

        <a href="index.html">Home </a> |<a href="about.html">About us </a> |  <a href="contact.html">contact us </a>

        <hr/>

        <p> <u> Doing ordinary things extra ordinarily.....</u></p>

        <ul>

            <li>Smart Agricuture Platform</li>

            <li>AI in Agriculture</li>

            <li>Robotics for Greenhouse Automation</li>

        </ul>

    </body>

</html>

2)contact.html

<!DOCTYPE html>

<html>

    <head>

        <title>Transflower Store</title>

    </head>

    <body>

        <h2>Transflowe Store</h2>

        <hr/>

        <a href="idnex.html">Home </a> |<a href="about.html">About us </a> |  <a href="contact.html">contact us </a>

        <hr/>

        <ol>

            <li>radha89@flower.in</li>

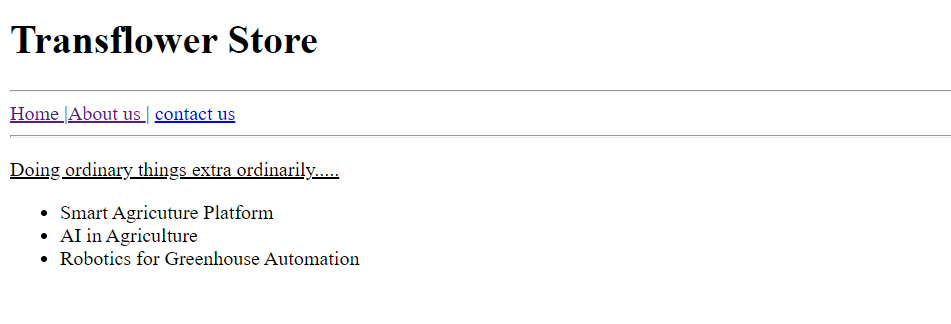
            <li>Contact Number: 356272888828</li>

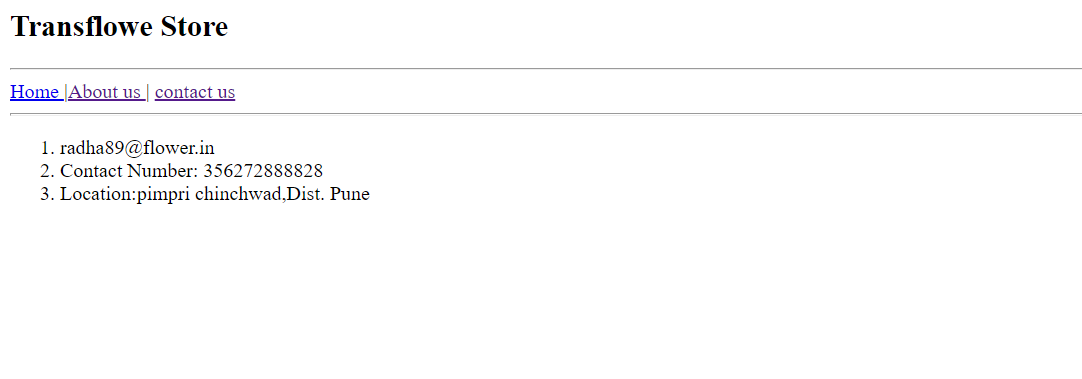
            <li>Location:pimpri chinchwad,Dist. Pune</li>

        </ol>

    </body>

</html>





**Day 4**

Assignment 4

1.Create a folder ExpressApp

2.Add server.js file to newly created folder ExpressApp

3.Add package.json file to ExpressApp using npminit command

{

  "name": "practice",

  "version": "1.0.0",

  "description": "yyyyy gggg hhh",

  "main": "server.js",

  "scripts": {

    "test": "echo \"Error: no test specified\" && exit 1",

    "start": "node server.js"

  },

  "author": "renu",

  "license": "ISC",

  "dependencies": {

    "express": "^4.17.1",

    "mongoose": "^5.13.3",

    "mysql": "^2.18.1"

  }

}

4.Use npm install express --save command to download express Framework to ExpressApp folder

5.User npm install mysql --save command to download mysql to ExpressApp

6.Use npm install mongoose --save comand to downlowad mongoose to ExpressApp

7.Import Express module in server.js

const { urlencoded } = require('body-parser');

var express=require('express');

var path=require('path');

const app=express();

// configuring middleware

app.use(urlencoded({extended:true}));

app.use(express.json());

//Configure public folder as static folder  to express module

var staticMiddleware=express.static(path.join(\_\_dirname,"public"));

app.use(staticMiddleware);

// 3 HTTP Request Handlers

// how many get handlers  : 1

// how many many post handlers: 2

//Logic for sending default html file to browser

app.get("/", (req, res)=>{

    res.sendFile(path.join(\_\_dirname + '/index.html'));

});

//Logic for Login form posted Data

app.post("/login",(request, response)=>{

   console.log( request.body);

   var user=request.body;

   //User Credential logic

   if(user.email=="praut366@gmail.com" &&

      user.password=="seed"){

          console.log("valid user");

          response.send("Valid user");

      }

      else

      {

         console.log("Invalid User");

         response.send("InValid user");

      }

    console.log("Login form is posted by  browser is received at server");

});

//Logic for  Register form posted Data

app.post("/register",(request, response)=>{

    console.log( request.body);

    console.log("REgister  form is posted by  browser is received at server");

});

app.listen(7000);

console.log("Website is being hosted on port no 7000");

8.Create public folder to Express App

9.Add basic index.html , about.html, contact.html, login.html, register.html file to public

1)index.html

<!DOCTYPE html>

<html>

    <head>

        <Title>Transflower Web</Title>

    </head>

    <body>

        <h1>Online Flower Shop </h1>

        <a href="/">Home</a> | <a href="/about.html">About us</a> |<a href="/contact.html">Contact</a>|<a href="/login.html">Sign In</a>

        <hr/>

    </body>

</html>

2)about.html

<!DOCTYPE html>

<html>

    <head>

        <Title>Transflower Web</Title>

    </head>

    <body>

        <h1>About Us </h1>

        <a href="/">Home</a> | <a href="/about.html">About us</a> |<a href="/contact.html">Contact</a>|<a href="/login.html">Sign In</a>

        <hr/>

3)contact.html

<!DOCTYPE html>

<html>

    <head>

        <Title>Transflower Web</Title>

    </head>

    <body>

        <h1>Contact Us </h1>

        <a href="/">Home</a> | <a href="/about.html">About us</a> |<a href="/contact.html">Contact</a>

        <hr/>

    </body>

</html>

    </body>

</html>

10.Add images folder to public folder

11.Add videos folder to public folder

12.Add audios folder to public folder

13.Import path module to server.js

14.Configure staticmidleware to configure public folder.

15.Configure json, urlencodingmiddlerware to server.js

16.Implement app.get for "/" url

17.Implement app.post for "/login" url

18.Implement app.postfor "/register" url

19.Implement logic for login.html file using form element.

Login.html

<!DOCTYPE html>

<html>

    <head>

        <Title>Transflower Web</Title>

    </head>

    <body>

        <h1>Login </h1>

        <a href="/">Home</a> | <a href="/about.html">About us</a> |<a href="/contact.html">Contact</a>|<a href="/login.html">Sign In</a>

        <hr/>

        <form method="POST" action="/login">

            <table>

                <tr>

                    <td> <label>Email</label></td>

                    <td> <input type="email"  name="email" required/></td>

                </tr>

                <tr>

                    <td> <label>Password</label></td>

                    <td> <input type="password" name="password" required/></td>

                </tr>

                <tr>

                    <td></td>

                    <td><button type="submit">Login</button></td>

                </tr>

            </table>

        </form>

        <a href="/register.html">Register new user</a>

    </body>

</html>

20.Implment basic validation using required to input elements.

21.Implment registration UI logic to register.html using form element.

Register.html

<!DOCTYPE html>

<html>

    <head>

        <Title>Transflower Web</Title>

    </head>

    <body>

        <h1>Register </h1>

        <a href="/">Home</a> | <a href="/about.html">About us</a> |<a href="/contact.html">Contact</a>|<a href="/login.html">Sign In</a>

        <hr/>

        <form method="POST" action="/register">

            <table>

                <tr>

                    <td> <label>Email</label></td>

                    <td> <input type="text" name="email" required/></td>

                </tr>

                <tr>

                    <td> <label>Password</label></td>

                    <td> <input type="password" name="password" required/></td>

                </tr>

                <tr>

                    <td> <label>Confirm Password</label></td>

                    <td> <input type="password" name="Password" required/></td>

                </tr>

                <tr>

                    <td> <label>First Name</label></td>

                    <td> <input type="text" name="first name" required /></td>

                </tr>

                <tr>

                    <td> <label>Last  Name</label></td>

                    <td> <input type="text" name="last name" required/></td>

                </tr>

                <tr>

                    <td> <label>City</label></td>

                    <td> <input type="text" name="city"/></td>

                </tr>

                <tr>

                    <td> <label>Birthdate</label></td>

                    <td> <input type="date"/></td>

                </tr>

                <tr>

                    <td></td>

                    <td><button type="submit">Register</button></td>

                </tr>

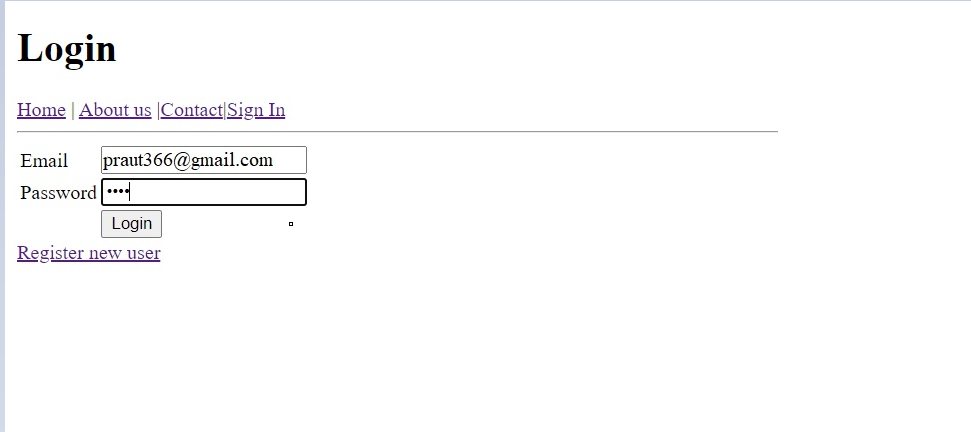
            </table>

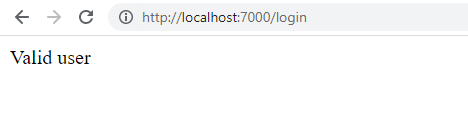
        </form>

    </body>

</html>

22.Test login form and register form are working with Client UI and server side post logic







**Day 5**

1.Create new Node JS Application TFLStore.

2.Add package.json file

**Package.json**

{

  "name": "tflstore",

  "version": "1.0.0",

  "description": "new tlf project",

  "main": "server.js",

  "scripts": {

    "test": "echo \"Error: no test specified\" && exit 1",

    "start": "node server.js"

  },

  "author": "renu",

  "license": "ISC",

  "dependencies": {

    "express": "^4.17.1",

    "mongoose": "^5.13.3",

    "mysql": "^2.18.1"

  }

}

3.Install express JS node Module.

4.Build server.js file with sending index.html file to client for <http://localhost:7000>

**Server.js**

const { urlencoded } = require('body-parser');

var express=require('express');

var path=require('path');

const app=express();

// configuring middleware

app.use(urlencoded({extended:true}));

app.use(express.json());

//Configure public folder as static folder  to express module

var staticMiddleware=express.static(path.join(\_\_dirname,"public"));

app.use(staticMiddleware);

// 3 HTTP Request Handlers

// how many get handlers  : 1

// how many many post handlers: 2

//Server Side

//Logic for sending default html file to browser

app.get("/", (req, res)=>{

    res.sendFile(path.join(\_\_dirname + '/index.html'));

});

//Logic for Login form posted Data

app.post("/login",(request, response)=>{

   console.log( request.body);

   var user=request.body;

   //User Credential logic

   if(user.email=="praut366@gmail.com" &&

      user.password=="seed"){

          console.log("valid user");

          response.send("Valid user");

      }

      else

      {

         console.log("Invalid User");

         response.send("InValid user");

      }

    console.log("Login form is posted by  browser is received at server");

});

//Logic for  Register form posted Data

app.post("/register",(request, response)=>{

    console.log( request.body);

    console.log("REgister  form is posted by  browser is received at server");

});

app.listen(8500);

console.log("Website is being hosted on port no 8500");

5add public folder to TFLStore

6.Add images, videos, audio, scripts , styles folder to TFLStore

7.Add index.html file, list.html, about.html file, contact.html , catalog.html, login.html, register.html

**Contact.html**

<!DOCTYPE html>

<html>

    <head>

        <Title>Transflower Web</Title>

    </head>

    <body>

        <h1>Contact Us </h1>

        <a href="/">Home</a> | <a href="/about.html">About us</a> |<a href="/contact.html">Contact</a>|<a href="/login.html">Sign In</a>

        |<a href="/list.html">list In</a>

        |<a href="/catalog.html">catalog In</a>

        |<a href="/multimedia.html">multimedia In</a>

        <hr/>

        <p id="para1">Seed Infotech Ltd.....</p>

        <p id="para2">Microsoft India Pvt. Ltd....</p>

        <button onclick="onButtonClick()">Change content</button>

        <script>

            var onButtonClick=function(){

                //DOM

                //alert("Button is pressed......")

                var element=document.getElementById("para2");

                element.innerHTML="Transflower Learning Pvt. Ltd."

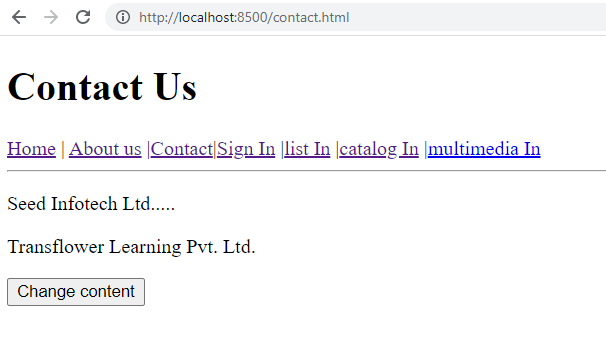
            }

        </script>

    </body>

</html>





8.Design index.html file with images, text, hyerlinks to navigate to other pages.

**Index.html**

<!DOCTYPE html>

<html>

    <head>

        <Title>Transflower Web</Title>

    </head>

    <body>

        <h1>Online Flower Shop </h1>

        <a href="/">Home</a> | <a href="/about.html">About us</a> |<a href="/contact.html">Contact</a>|<a href="/detail.html">Sign In</a>

        |<a href="/list.html">list In</a>

        |<a href="/catalog.html">catalog In</a>

        <hr/>

        <iframe src="www.flipkart.in"

        title="Prathamesh Raut">

</iframe>

    </body>

</html>

9.Design register.html page with standard html elements input types

**Register.html**

<!DOCTYPE html>

<html>

    <head>

        <Title>Transflower Web</Title>

    </head>

    <body>

        <h1>Register </h1>

        <a href="/">Home</a> | <a href="/about.html">About us</a> |<a href="/contact.html">Contact</a>|<a href="/login.html">Sign In</a>

        |<a href="/list.html">list In</a>

        |<a href="/catalog.html">catalog In</a>

        <hr/>

        <form method="POST" action="/register">

            <table>

                <tr>

                    <td> <label>Email</label></td>

                    <td> <input type="text" name="email"/></td>

                </tr>

                <tr>

                    <td> <label>Password</label></td>

                    <td> <input type="password" name="password"/></td>

                </tr>

                <tr>

                    <td> <label>Confirm Password</label></td>

                    <td> <input type="password" name="confirm password"/></td>

                </tr>

                <tr>

                    <td> <label>First Name</label></td>

                    <td> <input type="text" name="first name"/></td>

                </tr>

                <tr>

                    <td> <label>Last  Name</label></td>

                    <td> <input type="text" name="last name"/></td>

                </tr>

                <tr>

                    <td> <label>City</label></td>

                    <td> <input type="text" name="city"/></td>

                </tr>

                <tr>

                    <td> <label>Birthdate</label></td>

                    <td> <input type="date" name="birthdate"/></td>

                </tr>

                <tr>

                    <td></td>

                    <td><button type="submit">Register</button></td>

                </tr>

            </table>

        </form>

    </body>

</html>



10.Deisng login.html page with table and form, input text to accept user credentials

**Login.html**

<!DOCTYPE html>

<html>

    <head>

        <Title>Transflower Web</Title>

    </head>

    <body>

        <h1>Login </h1>

        <a href="/">Home</a> | <a href="/about.html">About us</a> |<a href="/contact.html">Contact</a>|<a href="/login.html">Sign In</a>

        |<a href="/list.html">list In</a>

        |<a href="/catalog.html">catalog In</a>

        <hr/>

        <form method="POST" action="/login">

            <table>

                <tr>

                    <td> <label>Email</label></td>

                    <td> <input type="email"  name="email" required/></td>

                </tr>

                <tr>

                    <td> <label>Password</label></td>

                    <td> <input type="password" name="password" required/></td>

                </tr>

                <tr>

                    <td></td>

                    <td><button type="submit">Login</button></td>

                </tr>

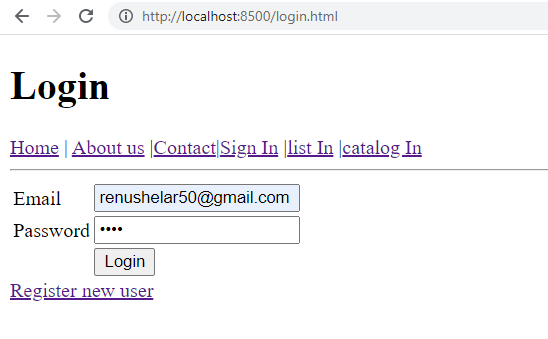
            </table>

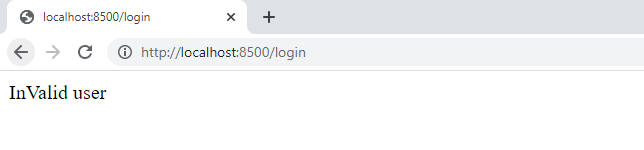
        </form>

        <a href="/register.html">Register new user</a>

    </body>

</html>





11.Write Client Side javascript code to print data entered in browser console. for both login.html, register.html

**Register.html**

<!DOCTYPE html>

<html>

    <head>

        <Title>Transflower Web</Title>

    </head>

    <body>

        <h1>Flower Details </h1>

        <a href="/">Home</a> | <a href="/about.html">About us</a> |<a href="/contact.html">Contact</a>|<a href="/login.html">Sign In</a>

      <hr/>

        <input  id="emailid" type="text" value="praut366@gmail.com"/>

        <br/>

        <input  id="password" type="text" value="sonu"/>

        <br/>

        <input  id="firstname" type="text" value="sonu"/>

        <br/>

        <input  id="lastname" type="text" value="raut"/>

        <br/>

        <input  id="location" type="text" value="pune"/>

        <br/>

        <input  id="birthdate" type="date" value=""/>

        <br/>

        <button onClick="onButtonClick()">register</button>

        <hr/>

        <p id="paraemail">email</p>

        <p id="parapassword">password</p>

        <p id="parafirstname">firstname</p>

        <p id="paralastname">lastname</p>

        <p id="paralocation">location</p>

        <p id="parabirthdate">birthdate</p>

        <script>

                var onButtonClick=function(){

                    //getting reference of element whose id is username

                    var txtUserName=document.getElementById("emailid");

                    var dataUserName=txtUserName.value;

                    var txtpassword=document.getElementById("password");

                    var datapassword=txtpassword.value;

                    var txtfirstname=document.getElementById("firstname");

                    var datafirstname=txtfirstname.value;

                    var txtlastname=document.getElementById("lastname");

                    var datalastname=txtlastname.value;

                    var txtlocation=document.getElementById("location");

                    var datalocation=txtlocation.value;

                    var txtbirthdate=document.getElementById("birthdate");

                    var databirthdate=txtbirthdate.value;

                    //get the reference of element whose id is para1

                    var elementUserName=document.getElementById("paraemail");

                    elementUserName.innerHTML=dataUserName;

                    var elementpassword=document.getElementById("parapassword");

                    elementpassword.innerHTML=datapassword;

                    var elementfn=document.getElementById("parafirstname");

                    elementfn.innerHTML=datafirstname;

                    var elementln=document.getElementById("paralastname");

                    elementln.innerHTML=datalastname;

                    var elementlocation=document.getElementById("paralocation");

                    elementlocation.innerHTML=datalocation;

                    var elementdate=document.getElementById("parabirthdate");

                    elementdate.innerHTML=databirthdate;

                }

        </script>

    </body>

</html>



12.Show list of customers in list.html with information id, firstname, lastname, contact, email, location as static content.

**List.html**

<!DOCTYPE html>

<html>

    <head>

        <Title>TFL Store</Title>

    </head>

    <!--- Markup language syntax-->

    <!-- HyperText Markup Language-->

    <!-- HTML-->

    <body>

        <h1>Online Flower Shop </h1>

        <a href="/">Home</a> | <a href="/about.html">About us</a> |<a href="/contact.html">Contact</a>|<a href="/login.html">Sign In</a>

        |<a href="/list.html">list In</a>

        |<a href="/catalog.html">catalog In</a>

        <hr/>

        <h2>customer Information </h2>

        <hr/>

        <marquee width="60%" direction="up" height="100px">

            welcome to TFL store

        </marquee>

        <p>List of Customer</p>

        <table style="width:100%">

            <tr>

                <th>id</th>

                <th>Firstname</th>

                <th>Lastname</th>

                <th>Contact Number</th>

                <th>location</th>

              </tr>

            <tr>

                <td>1</td>

                <td>Prathamesh</td>

                <td>Raut</td>

                <td>7864567902</td>

                <td>pune</td>

            </tr>

            <tr>

                <td>2</td>

                <td>Sameer</td>

                <td>Pande</td>

                <td>6781735841</td>

                <td>mumbai</td>

            </tr>

            <tr>

                <td>3</td>

                <td>Seema</td>

                <td>Nene</td>

                <td>9081735901</td>

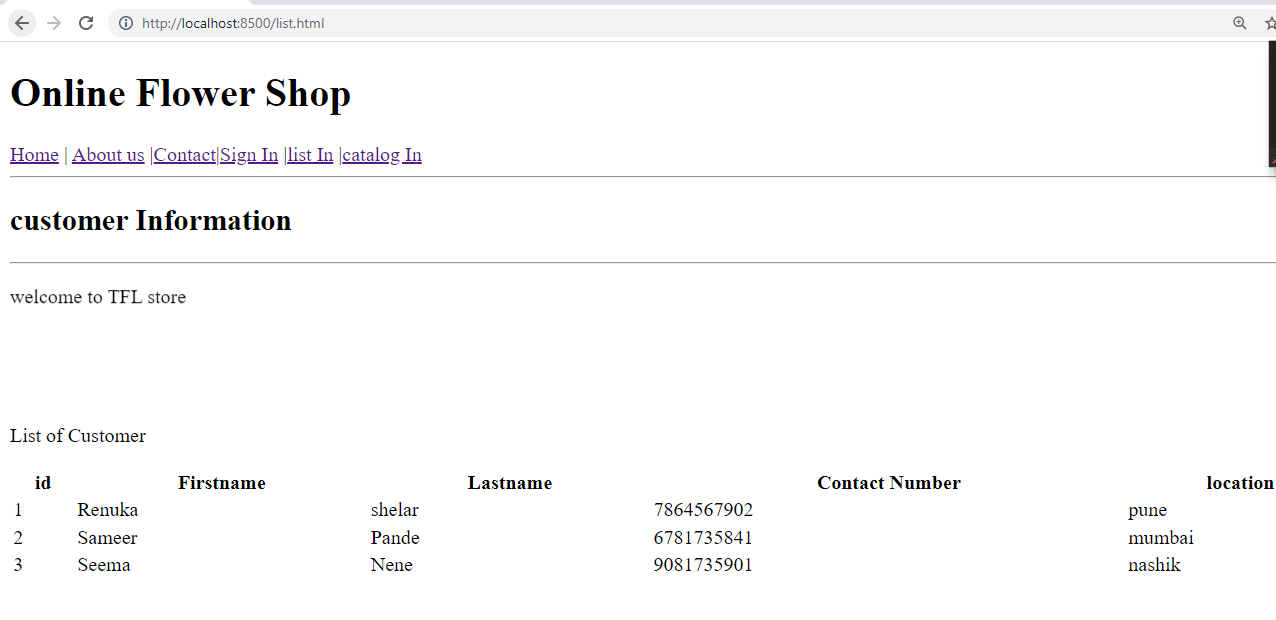
                <td>nashik</td>

            </tr>

        </table>

    </body>

</html>



13.Show list of flowers in catalog.html with flower image, title, description, unitprice, quantity and linke as static content

using standard HTML elments

**catlog.html**

<!DOCTYPE html>

<html>

    <head>

        <Title>Transflower Web</Title>

    </head>

    <!--- Markup language syntax-->

    <!-- HyperText Markup Language-->

    <!-- HTML-->

    <body>

        <h1>Online Flower Shop </h1>

        <a href="/">Home</a> | <a href="/about.html">About us</a> |<a href="/contact.html">Contact</a>|<a href="/login.html">Sign In</a>

        |<a href="/list.html">list In</a>

        |<a href="/catalog.html">catalog In</a>

        <hr/>

        <h2>list of flowers </h2>

        <hr/>

        <marquee width="60%" direction="up" height="100px">

            welcome to TFL store

        </marquee>

        <p>List of Customer</p>

        <table style="width:100%">

            <tr>

                <th>Image</th>

                <th>title</th>

                <th>description</th>

                <th>unitprice</th>

                <th>quantity</th>

              </tr>

            <tr>

                <td><img src="image/lily.jpg" width="200" height="200"/>

                </td>

                <td>lily</td>

                <td>beautiful flower</td>

                <td>20</td>

                <td>100</td>

            </tr>

            <tr>

                <td><img src="image/rose.jpeg" width="200" height="200" /> </td>

                <td>rose</td>

                <td>pink colour</td>

                <td>30</td>

                <td>100</td>

            </tr>

            <tr>

                <td><img src="image/gerbera.jpg" width="200" height="200" /> </td>

                <td>gerbera</td>

                <td>red colour</td>

                <td>50</td>

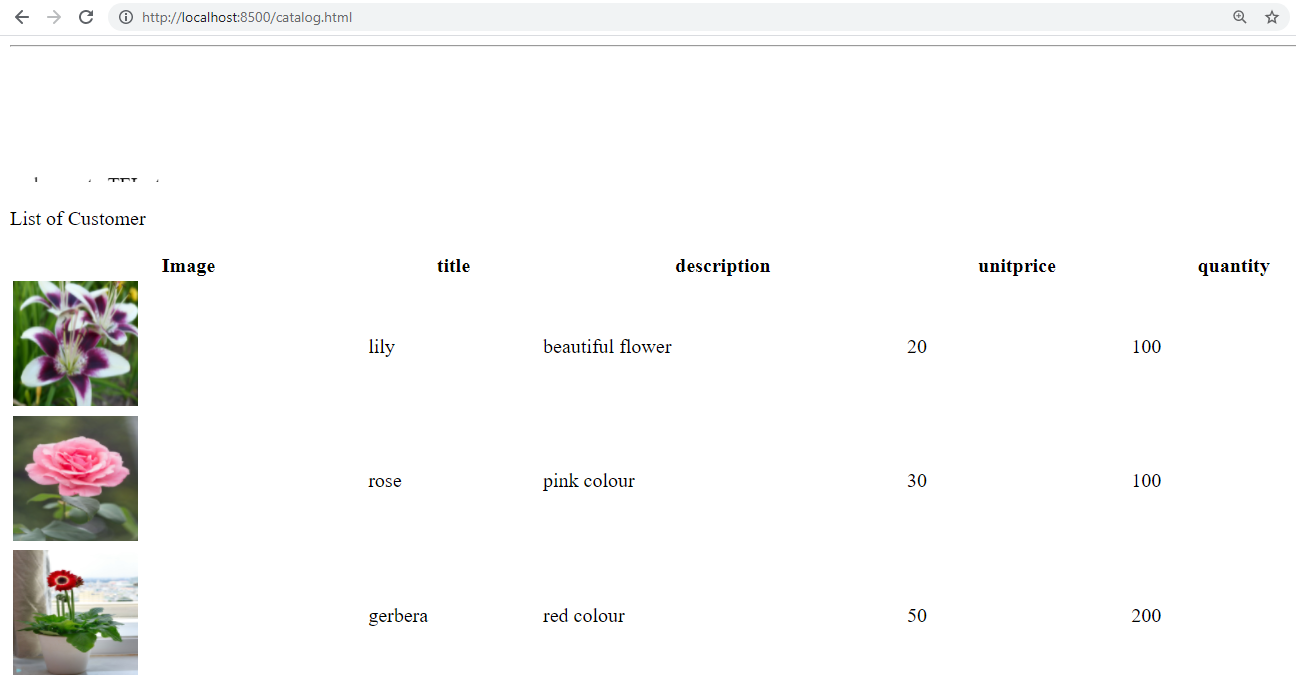
                <td>200</td>

            </tr>

        </table>

    </body>

</html>



15.show data added by user using register.html to paragraph element using DOM API

14.Implement Client Side Javascript code to validate user credentials on signINbuttton clicked.

**Detail.html**

<!DOCTYPE html>

<html>

    <head>

        <Title>Transflower Web</Title>

    </head>

    <body>

        <h1>login </h1>

        <a href="/">Home</a> | <a href="/about.html">About us</a> |<a href="/contact.html">Contact</a>|<a href="/login.html">Sign In</a>

        |<a href="/list.html">list In</a>

        |<a href="/catalog.html">catalog In</a>

      <hr/>

         <!--to display box we will use input here -->

        <input  id="emailid" type="text" value="praut366@gmail.com"/>

        <br/>

        <input  id="password" type="text" value="renu"/>

        <br/>

        <button onClick="onButtonClick()">Show Data</button>

        <hr/>

        <p id="paraemailid">emailid</p>

        <p id="parapassword">password</p>

        <script>

                var onButtonClick=function(){

                    //getting reference of element whose id is username

                    var txtUserName=document.getElementById("emailid");

                    var dataUserName=txtUserName.value;

                    var txtLocation=document.getElementById("password");

                    var dataLocation=txtLocation.value;

                    //get the reference of element whose id is para1

                    var elementUserName=document.getElementById("paraemailid");

                    elementUserName.innerHTML=dataUserName;

                    var elementLocation=document.getElementById("parapassword");

                    elementLocation.innerHTML=dataLocation;

                    // DOM API :

                    // Document Object Model

                    // Application Programmers Interface ( built in functions  by DOM)

                    //document.createElement----------will create new html element

                    //document.removeChild------------will remove existing html element

                    //document.appendChild------------will append child to html document

                    //document.replaceChild

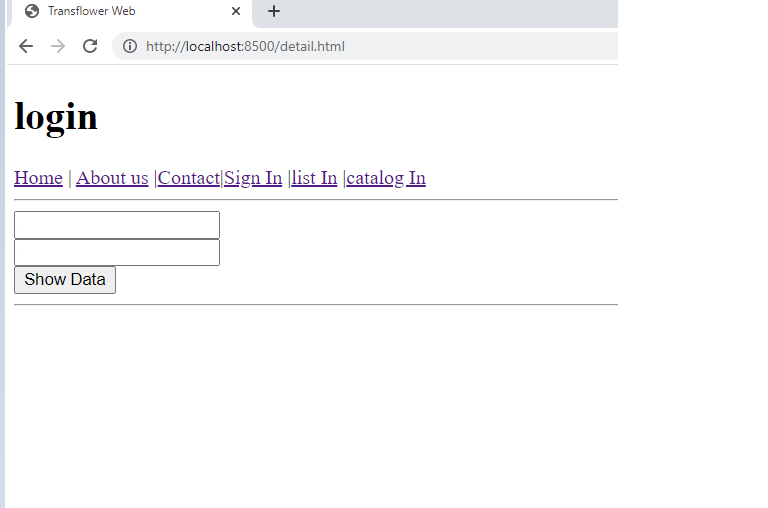
                    //document.write

                }

        </script>

    </body>

</html>



16.Show videos about transflower Greenhouse on about.html page

**About.html**

<!DOCTYPE html>

<html>

    <head>

        <Title>Transflower Web</Title>

    </head>

    <body>

        <h1>Multimedia Resources </h1>

        <a href="/">Home</a> | <a href="/about.html">About us</a> |<a href="/contact.html">Contact</a>|<a href="/login.html">Sign In</a>

        |<a href="/list.html">list In</a>

        |<a href="/catalog.html">catalog In</a>

        <hr/>

        <video width="320" height="240" controls>

            <source src="upsc.mp4" type="vedios/mp4">

        </video>

    </body>

</html>



**Day 6**

1.Create new Node JS experss Application with server.js, package.json and public folder

install express JS

2.Add new event.html file to handle EventListeners for Mouse, Keyboard events

**Event.html**

<!DOCTYPE html>

<html>

<head>

    <title>Transflower Store</title>

</head>

<body>

    <!--<button onClick="alert('button is pressed')">click me</button>-->

    <button id="myBtn">Press me</button>

    <p id="demo">Hello</p>

    <script>

        // What is the root node DOM Tree

        //Document

        var btn=document.getElementById("myBtn");

        btn.addEventListener("click", displayDate);

        document.addEventListener("mousedown",onMouseDown);

        document.addEventListener("mousemove",onMouseMove);

        document.addEventListener("mouseup",onMouseUp);

        document.addEventListener("keydown",onKeyDown);

        //Client Side Event Resgister

        //event handler registration for click event

        // Event Handlers

        // Event Recievers

        // Event Listeners

        function displayDate() {

            document.getElementById("demo").innerHTML = Date();

        };

        function onMouseDown(event){

            console.log("Mouse Down");

            var x = event.clientX;

            var y = event.clientY;

            var coords =  x + "," + y;

            document.getElementById("demo").innerHTML =coords;

        };

        function onMouseUp(event){

            console.log("Mouse UP");

            var x = event.clientX;

            var y = event.clientY;

            var coords =  x + "," + y;

            document.getElementById("demo").innerHTML =coords;

        };

        function onMouseMove(event){

            console.log("Mouse Move");

            var x = event.clientX;

            var y = event.clientY;

            var coords =  x + ", " + y;

            document.getElementById("demo").innerHTML =coords;

        };

        function onKeyDown(){

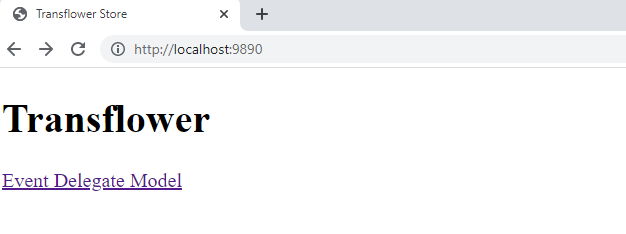
            console.log("Key Down");

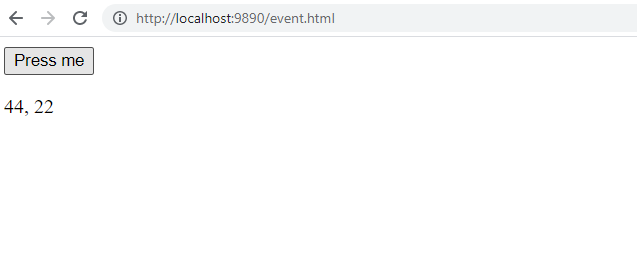
        }

    </script>

</body>

</html>





3.Add new paintbrush file to draw Graphics shaped line, rectangle, circle, filled using canvas.

**Paintbrush.html**

<!DOCTYPE html>

<html>

    <head>

        <title>Paint Brush </title>

    </head>

    <body>

        <canvas id="canvas" width="300" height="300" style="background-color:azure;"></canvas>

        <h1 style="color: red;">Transflower Paint Brush</h1>

        <p style="color:brown;font-size: large; font-family:'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif">Doing ordinary things extraordinarily</p>

        <script>

            // HTML 5.0 Behaviour

            function drawLine(){

                //DOM

                var theCanvas=document.getElementById("canvas");

                var context=theCanvas.getContext("2d");

                context.moveTo(10,200);

                context.lineTo(200,200);

                context.stroke();

            }

            function drawRectangle(){

                var theCanvas=document.getElementById("canvas");

                var context=theCanvas.getContext("2d");

               // context.fillStyle="blue";

                context.fillRect(50,50,200,150);

            }

            function drawTriangle() {

                var canvas = document.getElementById('canvas');

                if (canvas.getContext) {

                    var ctx = canvas.getContext('2d');

                    ctx.beginPath();

                    ctx.moveTo(75, 50);

                    ctx.lineTo(100, 75);

                    ctx.lineTo(100, 25);

                    ctx.fill();

                }

            }

            function drawCircle(){

                var canvas = document.getElementById('canvas');

                if (canvas.getContext) {

                    var ctx = canvas.getContext('2d');

                    ctx.beginPath();

                    ctx.arc(50,50,40,0,2\* Math.PI);

                    ctx.stroke();

                    //ctx.fill();

                }

            }

           // drawLine();

           //drawRectangle();

           //drawTriangle();

           function drawSmily() {

                var canvas = document.getElementById('canvas');

                if (canvas.getContext) {

                    var ctx = canvas.getContext('2d');

                    ctx.beginPath();

                    ctx.arc(75, 75, 50, 0, Math.PI \* 2, true); // Outer circle

                    ctx.moveTo(110, 75);

                    ctx.arc(75, 75, 35, 0, Math.PI, false);  // Mouth (clockwise)

                    ctx.moveTo(65, 65);

                    ctx.arc(60, 65, 5, 0, Math.PI \* 2, true);  // Left eye

                    ctx.moveTo(95, 65);

                    ctx.arc(90, 65, 5, 0, Math.PI \* 2, true);  // Right eye

                    ctx.stroke();

                }

            }

           //drawCircle();

           drawSmily();

        </script>

    </body>

</html>

<!---

      basics

      syntax,

      rules

      keyword

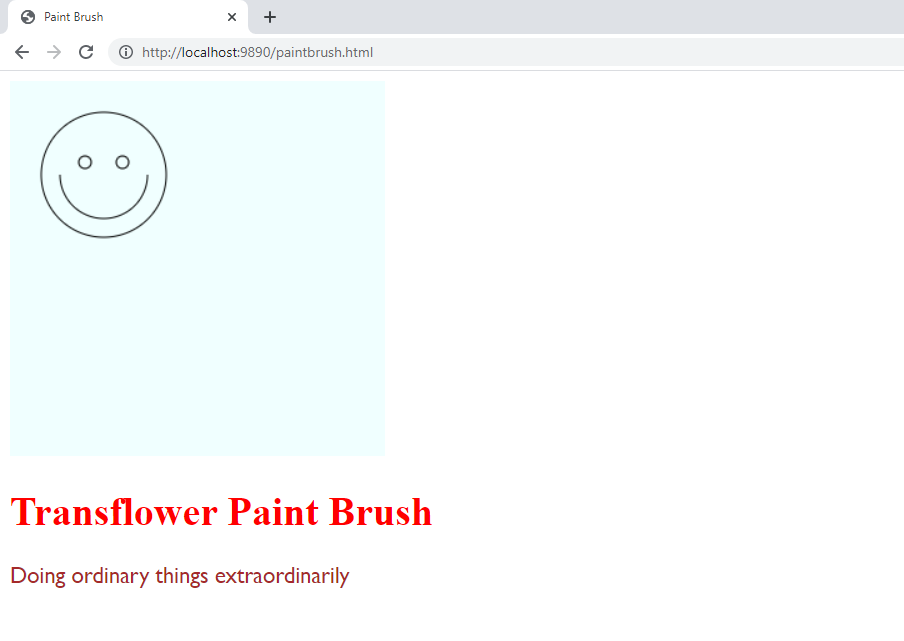
      using inbuilt library functions

      searching online

      patience to read documentation

      courage to apply  and solve possible errors

-->



4.Add new svg.html file to show Graphics shaped and smily face using HTML elements.

<!DOCTYPE html>

<html>

<head>

    <title>RJ enterprises</title>

</head>

<body>

        <h1>Smiley Face using SVG</h1>

        <svg height="200" width="600">

             <circle id="yellowcircle" cx="50" cy="50" r="50" fill="yellow"/>

            <circle id="blackcircle" cx="30" cy="30" r="10" fill="black"/>

            <circle id="blackcircle" cx="70" cy="30" r="10" fill="black"/>

            <ellipse id="blackellipse" cx="50" cy="75" rx="20" ry="8" fill="black"/>

            <ellipse id="yellowellipse" cx="50" cy="73" rx="20" ry="8" fill="yellow"/>

        </svg>

</body>

</html>



5.Add new storage.html to store user entered data into localstorage.

<!DOCTYPE html>

<html>

<head>

    <title>WebStorage Demo</title>

</head>

<body>

       <h1>Transflower</h1>

       <hr/>

        <input type="text" id="txtName"/>

        <button onClick="onSetData()">Set Data</button>

        <hr/>

        <div id="result"></div>

        <script>

            function onSetData(){

                //logic for setting data into storage

                if(typeof(Storage)!=="undefiend"){

                    var txtInput=document.getElementById("txtName");

                    var name=txtInput.value;

                    localStorage.setItem("person",name);

                }

                else{

                    var divId=document.getElementById("result");

                    divId.innerHTML="Sorry, your browser does not support HTML 5 feature :WebStorage"

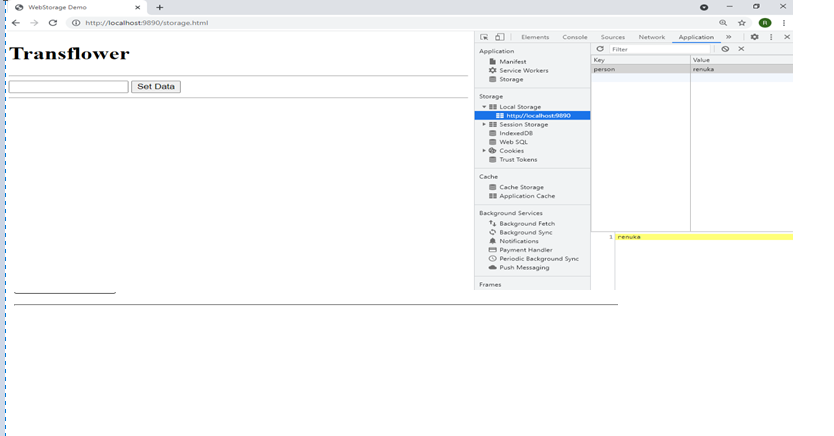
                }

            }

        </script>

</body>

</html>



6.Add new register.html to store user details into localstorage.

<!DOCTYPE html>

<html>

<head>

    <title>Regisgter</title>

</head>

<body>

    <h1>Register new User</h1>

    <hr/>

    <label>User Name :</label> <input type="text" id="username" name="username" required />

    <br/>

    <label>Password</label> <input type="password" id="password" name="password"  required/>

    <br/>

    <hr/>

    <!--<input type="button" value="submit"/>-->

    <button onclick="onRegister()">Regisger</button>

    <script>

        var onRegister=function()

        {

            if(typeof(Storage)!=="undefiend"){

                var txtUserName=document.getElementById("username");

                var userName=txtUserName.value;

                var txtPassword=document.getElementById("password");

                var password=txtPassword.value;

                var validUser={

                    "username":userName,

                    "password":password

                }

                localStorage.setItem("validuser",JSON.stringify(validUser));

            }

            else{

                var divId=document.getElementById("result");

                divId.innerHTML="Sorry, your browser does not support HTML 5 feature :WebStorage"

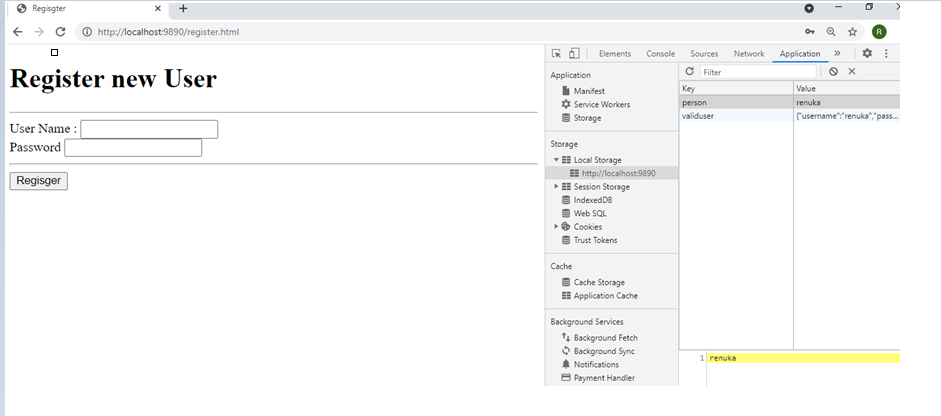
            }

        }

    </script>

</body>

</html>



7.Add new login.html file to verfiyusers credentials using data from localstorage.

<!DOCTYPE html>

<html>

<head>

    <title>Login</title>

</head>

<body>

        <h1>Login</h1>

        <hr/>

        <label>User Name :</label> <input type="text"  id="username" name="username"  required />

        <br/>

        <label>Password</label> <input type="password"  id="password" name="password"   required/>

        <br/>

        <hr/>

        <!--<input type="button" value="submit"/>-->

        <button onclick="validateUserClientSide()">Login</button>

        <div id="result"></div>

    <script>

        var validateUserClientSide=function()

        {

            if(typeof(Storage)!=="undefiend")

            {

                //get data from localstorage

                var strValidUser=localStorage.getItem("validuser");

                var validUser=JSON.parse(strValidUser);

                // get data from HTML controls of login.html

                var txtUserName=document.getElementById("username");

                var userName=txtUserName.value;

                var txtPassword=document.getElementById("password");

                var password=txtPassword.value;

                var status=false;

                if(validUser.username == userName && validUser.password== password){

                    status=true;

                }

                if(status){

                    var divId=document.getElementById("result");

                    divId.innerHTML="Valid User !"

                }

                else{

                    var divId=document.getElementById("result");

                    divId.innerHTML="Invalid User !"

                }

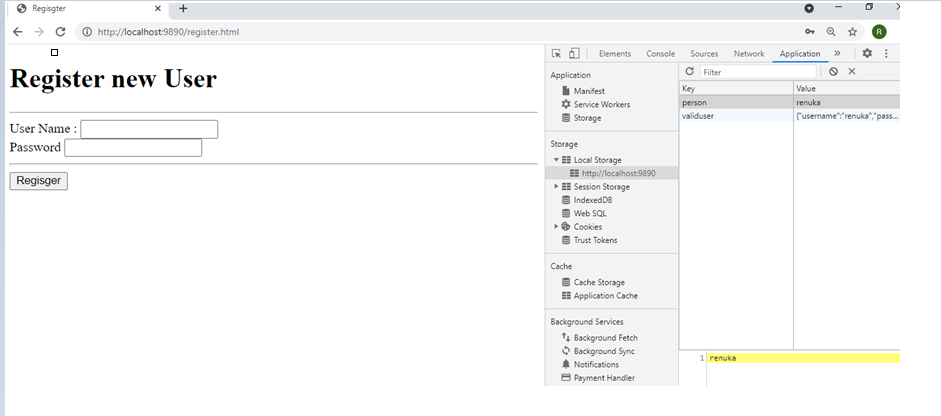
            }

        }

    </script>

</body>

</html>



8.Add insert.html file to insert flower details into array maintained in localstorage

**insert.html:**

<!-- 8.Add insert.html file to insert flower details into array maintained in localstorage -->

<!DOCTYPE html>

<html>

    <head>

        <title>RJ ENTERPRISE</title>

    </head>

    <body>

        <h1>Flower Details</h1>

        <a href="index.html">Home</a> | <a href="event.html">Event</a>

        <hr/>

        <table>

            <tr>

                <td><label>Flower Name</label></td>

                <td><input type="text" id="in1"/></td>

            </tr>

            <tr>

                <td><label>Description</label></td>

                <td><input type="text" id="in2"/></td>

            </tr>

            <tr>

                <td><label>Price</label></td>

                <td><input type="text" id="in3"/></td>

            </tr>

            <tr>

                <td><label>Quantity</label></td>

                <td><input type="text" id="in4"/>    </td>

            </tr>

            <tr>

                <td></td>

                <td><button onClick="storeData()">Submit</button></td>

            </tr>

            <script>

                function storeData(){

                    var fn=document.getElementById("in1");

                    var fn1=fn.value;

                    var fd=document.getElementById("in2");

                    var fd1=fd.value;

                    var fp=document.getElementById("in3");

                    var fp1=fp.value;

                    var fq=document.getElementById("in4");

                    var fq1=fq.value;

                    var flowerDetails={

                        "flowername":fn1,

                        "description":fd1,

                        "price":fp1,

                        "quantity":fq1

                    }

                    if(typeof(Storage)!=="undefined")

                    {

                        localStorage.setItem("flowerDetails",JSON.stringify(flowerDetails));

                    }

                    else{

                       var divid=document.getElementById("result");

                        divid.innerHTML="Sorry your browser does not support HTML 5 feature web storage";

                    }

                }

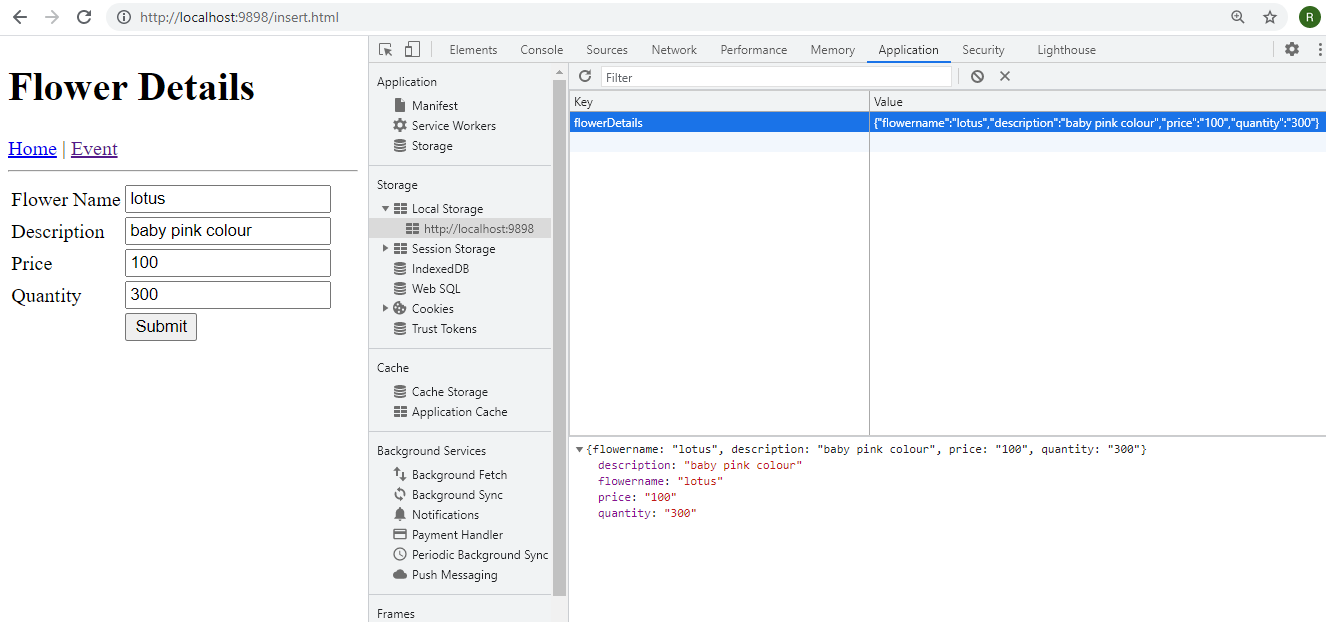
            </script>

        </table>

<div id="result"></div>

    </body>

</html>



9.Add list.html file to show list of all flowers stored in localstorage using dynamic list

10.Add rubberband.html file show rubber band line effect in canvas using MouseDown, Mousemove and MouseUp events and canvs

11.Implement logic to store all lines into localstorage and on refresh to show all lines drawn on web page.

**Rubberband.html**

<!doctype html>

<html>

<head>

    <title>Rubber band Effect</title>

</head>

<body>

    <h4>Rubberband line effect</h4>

    <canvas id="canvas" width=600 height=800></canvas>

    <script>

   var canvas=document.getElementById("canvas");

   canvas.addEventListener("mousedown",handleMouseDown);

   canvas.addEventListener("mouseup",handleMouseUp);

   canvas.addEventListener("mousemove",handleMouseMove);

   canvas.addEventListener("mouseout",handleMouseOut);

    var ctx=canvas.getContext("2d");

    var offsetX=canvas.offsetLeft;

    var offsetY=canvas.offsetTop;

    var startX;

    var startY;

    var isDown=false;

    function drawOval(x,y){

        ctx.clearRect(0,0,canvas.width,canvas.height);

        ctx.beginPath();

        ctx.moveTo(startX, startY + (y-startY)/2);

        ctx.bezierCurveTo(startX, startY, x, startY, x, startY + (y-startY)/2);

        ctx.bezierCurveTo(x, y, startX, y, startX, startY + (y-startY)/2);

        ctx.closePath();

        ctx.stroke();

    }

    function drawLine(x, y) {

      ctx.clearRect(0, 0, canvas.width, canvas.height);

        ctx.beginPath();

        ctx.moveTo(startX, startY);

       // ctx.lineTo(startX, startY, x, startY, x, startY + (y - startY) / 2);

       ctx.lineTo(x, y, startX, y, startX, startY + (y - startY) / 2);

        ctx.closePath();

        ctx.stroke();

    }

    function drawRectangle(x, y) {

        ctx.clearRect(0, 0, canvas.width, canvas.height);

        ctx.beginPath();

        //ctx.moveTo(startX, startY);

        //PenStyle

        ctx.strokeStyle = 'rgba(255,0,0,0.8)';

        //BrushStyle

        ctx.fillStyle = 'rgba(0,0,255,0.8)';

        ctx.rect(startX, startY, x, y);

        ctx.fillRect(startX, startY, x, y);

        ctx.closePath();

        ctx.stroke();

    }

    function handleMouseDown(e){

      e.preventDefault();

      e.stopPropagation();

      startX=parseInt(e.clientX-offsetX);

      startY=parseInt(e.clientY-offsetY);

      isDown=true;

    }

    function handleMouseUp(e){

      if(!isDown){ return; }

      e.preventDefault();

      e.stopPropagation();

      isDown=false;

    }

    function handleMouseOut(e){

      if(!isDown){ return; }

      e.preventDefault();

      e.stopPropagation();

      isDown=false;

    }

    function handleMouseMove(e){

      if(!isDown){ return; }

      e.preventDefault();

      e.stopPropagation();

      mouseX=parseInt(e.clientX-offsetX);

      mouseY=parseInt(e.clientY-offsetY);

      //drawOval(mouseX,mouseY);

      drawLine(mouseX,mouseY);

     //drawRectangle(mouseX, mouseY);

    }

    </script>

</body>

</html>

