**Advanced Java Script Real time Exmaples:**

1. **prototype Design Pattern:**

**Index.html:**

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Document</title>

  <link rel="stylesheet" href="index.css">

  <script type="text/javascript" src="./horoscope.js"></script>

  <script type="text/javascript" src="./person.js"></script>

  <script type="text/javascript" src="./index.js"></script>

</head>

<body>

  <div>

    <h1>Example for ProtoType Pattern using Person Data</h1>

    <label>Enter Name:</label> <input id="name" placeholder="enter name">

    <label>Enter Age:</label> <input id="age" placeholder="enter age">

    <label>Enter Location: </label><input id="loc"placeholder="enter location">

    <button id="add" onclick="addPerson();">Add Person</button>

  </div>

  <div>

    <br>

    <h2>Get Persons List so far along with Today Horoscope</h2>

    <label>Enter age:</label> <input id="ageCondition" placeholder="enter 0 for all">

    <button id="ageConditionBrn" onclick="getByAgeCondition();">Get</button>

    <p id="personsList"> empty</p>

  </div>

</body>

</html>

**Index.js:**

var Persons = [] ;//Empty Array. This is supposed to hold Persons Objects

function addPerson (){

//1. Get Person details from DOM

var name=document.getElementById("name").value;

var age=document.getElementById("age").value;

var location=document.getElementById("loc").value;

//2. Create Persons Object

var person = new Person();

//person.init("modi", 65);

person.init(name,age,location);

//3.Add it to the list

Persons.push(person);

}

function getByAgeCondition() {

//1. Get Age condition using html input ageCondition

var ageCondition=document.getElementById("ageCondition").value;

//2. if Age is zero get all persons description

//3. if age is some value, filter persons whose age is lower than given

if(ageCondition == 0) {

var personsList = "";

for( person of Persons) { //This is ES6 for loop

personsList += person.describeWithHoroscope();

    }

}

else {

var personsList = "";

for(person of Persons) {

if(person.age < ageCondition) {

personsList += person.describeWithHoroscope();

      }

  }

}

/\*else {

    var personsList = "";

    personsList = getHoroscopeByIndex(ageCondition);

  }\*/

//4. update personsList

var listElement = document.getElementById("personsList");

listElement.innerHTML = personsList;

}

**Horoscope.js:**

function Horoscopes() {

};

Horoscopes.prototype.arrPredictions = [

    "Today is your lucky day",

    "You will meet your soulmate",

    "You will find your destiny",

    "you will never forget today rest of your life",

    "you need to hold your anger",

    "Patience is your weapon",

    "you are born leader",

    "get your life back",

    "change your surroundings",

    "you will be surprised",

    "stay healthy",

    "time to take easy"

  ];

Horoscopes.prototype.get = function(name, age) {

  //algorithm to determine horoscope based on age

  var len = this.arrPredictions.length;

  var ind = Math.floor(Math.random() \* Math.floor(age % len));

  ind += Math.floor(Math.random() \* Math.floor(name.length%len));

  ind = ind % len;

  return this.arrPredictions[ind];

}

var horoscopes = new Horoscopes();

function getHoroscope(name, age) {

  return horoscopes.get(name, age);

}

**Index.css:**

#personsList {

  width: 400px;

  height: 400px;

  padding: 2px, 2px, 2px, 2px;

  border: 1px solid red;

};

**Person.js:**

function Person() {

};

Person.name = "";

Person.age = "";

Person.location = "";

Person.prototype.init = function(name, age,location) {

  this.name = name;

  this.age = age;

  this.location = location;

}

Person.prototype.describe = function() {

  var description = "Person Name is : " +this.name +" <br> Person age is :"+String(this.age)+"<br>Person Location is : "+this.location;

  return description;

}

Person.prototype.describeWithHoroscope = function() {

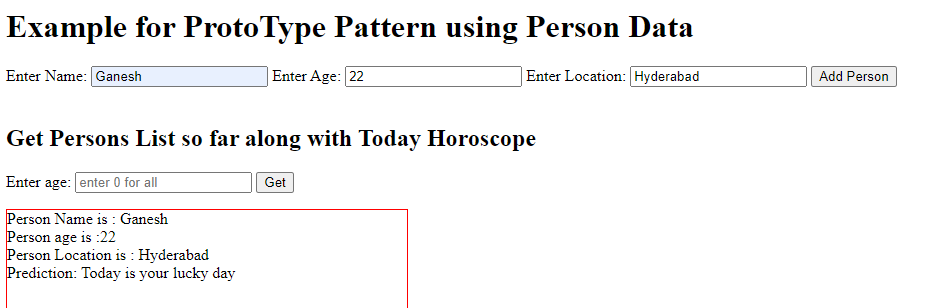
  var description = this.describe();

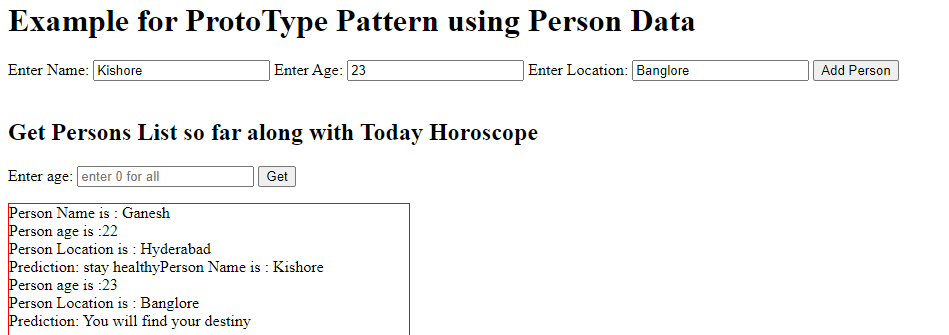
  description += " <br>Prediction: " + getHoroscope(this.name, this.age,this.location);

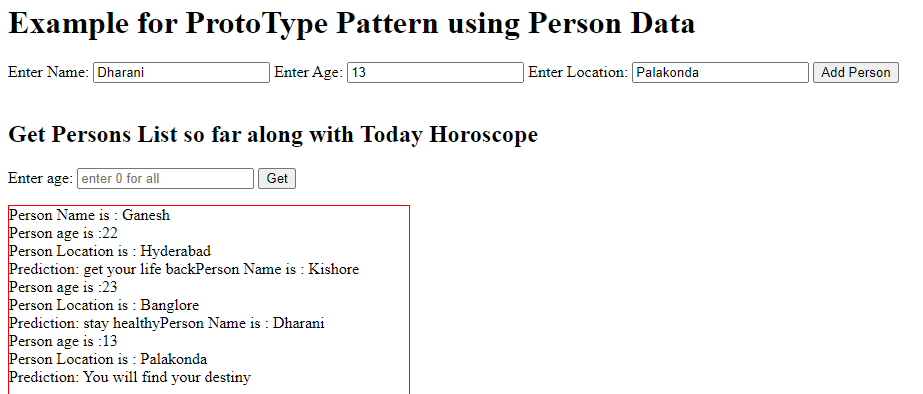
  return description;

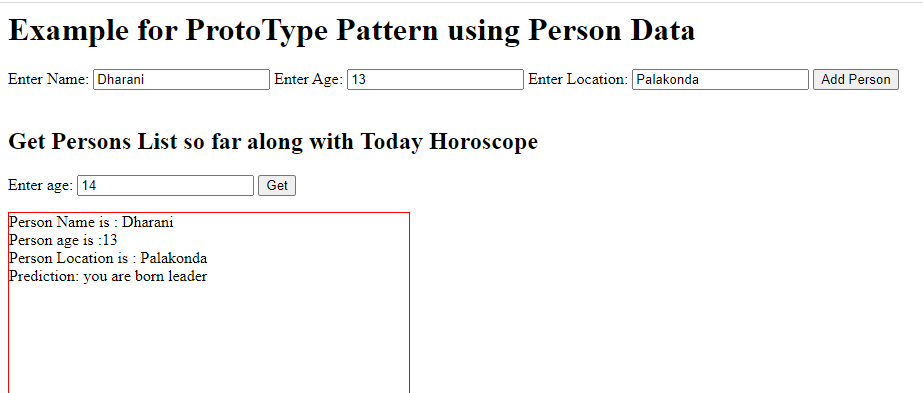
}

Output:









1. **Constructor Desing Pattern with DOM elements:**

**Index.html:**

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Document</title>

  <link rel="stylesheet" href="index.css">

  <script type="text/javascript" src="./person.js"></script>

  <script type="text/javascript" src="./index.js"></script>

  <script type="text/javascript" src="./address.js"></script>

  <script type="text/javascript" src="./personDOM.js"></script>

</head>

<body>

  <div>

    <h1>Example for Constructor Pattern using Person Data</h1>

    <label>Enter Name:</label> <input id="name" placeholder="enter name">

    <label>Enter Age:</label> <input id="age" placeholder="enter age"> <br>

    <label for=""><b><h1>Address</h1></b></label><br>

    <label for ="">House No:</label><input type ="text" id="hno">

    <label for ="">Street:</label><input type ="text" id="street">

    <label for ="">City:</label><input type ="text" id="city">

    <label for ="">Pin No:</label><input type ="text" id="pin">

    <button id="add" onclick="addPerson();">Add Person</button>

  </div>

  <div>

    <br>

    <h2>Get Persons List so far</h2>

    <label>Enter age:</label> <input id="ageCondition" placeholder="enter 0 for all">

    <button id="ageConditionBrn" onclick="getByAgeCondition();">Get</button>

    <p id="personsList"> empty</p>

  </div>

</body>

</html>

**Index.js:**

var Persons = [] ;//Empty Array. This is supposed to hold Persons Objects

function addPerson (){

//1. Get Person details from DOM

//2. Create Persons Object

//let pDom = new personDOM();

var person = new Person(personDOM.getName(), personDOM.getAge(),personDOM.getHouseNo(),personDOM.getStreet(),personDOM.getCity(),personDOM.getPin());

//3.Add it to the list

Persons.push(person);

}

function getByAgeCondition() {

let ageCondition =personDOM.getAgeCondition();

//1. Get Age condition using html input ageCondition

//2. if Age is zero get all persons description

//3. if age is some value, filter persons whose age is lower than given

if(ageCondition == 0) {

var personsList = "";

for( person of Persons) { //This is ES6 for loop

personsList += person.describe();

  }

  }else {

    var personsList = "";

    for(person of Persons) {

    if(person.age < ageCondition) {

    personsList += person.describe();

          }

      }

    }

//4. update personsList

var listElement = document.getElementById("personsList");

listElement.innerHTML = personsList;

}

**Index.css:**

#personsList {

  width: 400px;

  height: 400px;

  padding: 2px, 2px, 2px, 2px;

  border: 1px solid red;

};

**Person.js:**

function Person(name,age,houseNo,street,city,pin) {

//create Person properties

this.name = name;

this.age = age;

this.address = new Address(houseNo, street, city, pin);

this.describe = function() {

let description = "<br>Name is: "+this.name+"<br>Person age is: "+this.age;

    description += this.address.describe();

return description;

  }

};

**Address.js:**

function Address(houseNo,street,city,pin) {

    this.houseNo = houseNo;

    this.street = street;

    this.city = city;

    this.pin = pin;

    this.describe = function() {

    var description = "<br>House No is: "+this.houseNo+"<br>Street is : "+this.street+"<br>city is : "+this.city+"<br>pincode is: "+this.pin+"<br>";

    return description;

      }

    };

**PersonDOM.js:**

var personDOM = (function() {

    let NAME = 'name';

    let AGE= 'age';

    let HOUSENO = 'hno';

    let STREET = 'street';

    let CITY = 'city';

    let PIN = 'pin';

    let AGECONDITION='ageCondition';

    var dom = new Object();

    dom.getName = function() {

        let nameElement = document.getElementById(NAME);

        let name = '';

        if(typeof nameElement.value ==='string')

            name = nameElement.value;   //string we can omit

        return name;

    }

    dom.getAge = function() {

        let ageElement = document.getElementById(AGE);

        let age = 0;

        if(!isNaN(ageElement.value ))

            age = Number(ageElement.value);   //string we can omit

        return age;

    }

    dom.getHouseNo = function() {

        let houseNoElement = document.getElementById(HOUSENO);

        let houseNo = '';

        if(isNaN(houseNoElement.value ))

            houseNo = String(houseNoElement.value);   //string we can omit

        return houseNo;

    }

    dom.getStreet = function() {

        let streetElement = document.getElementById(STREET);

        let street = '';

        if(typeof streetElement.value ==='string')

            street = String(streetElement.value);   //string we can omit

        return street;

    }

    dom.getCity = function() {

        let cityElement = document.getElementById(CITY);

        let city = '';

        if(typeof cityElement.value ==='string')

            city = String(cityElement.value);   //string we can omit

        return city;

    }

    dom.getPin = function() {

        let pinElement = document.getElementById(PIN);

        let pin = '';

        if(!isNaN(pinElement.value ))

            pin = pinElement.value;   //string we can omit

        return pin;

    }

    dom.getAgeCondition = function() {

        let ageConditionElement =document.getElementById(AGECONDITION);

        let ageCondition = Number(0);

        if(!isNaN(ageConditionElement.value ))

            ageCondition = Number(ageConditionElement.value);   //string we can omit

        return ageCondition;

    }

    return dom;

})();

// const name = document.getElementById("name").value;

// const age = document.getElementById("age").value;

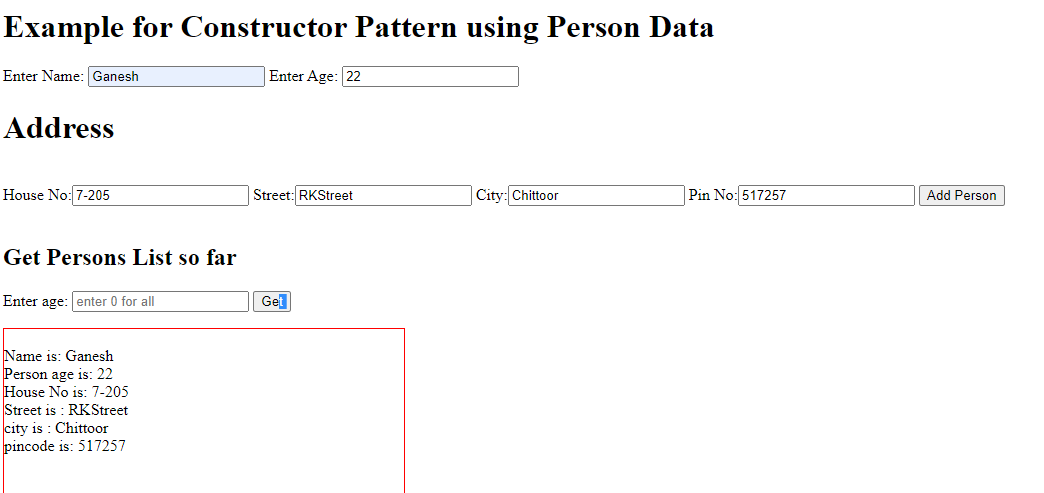
// const houseNo = document.getElementById("hno").value;

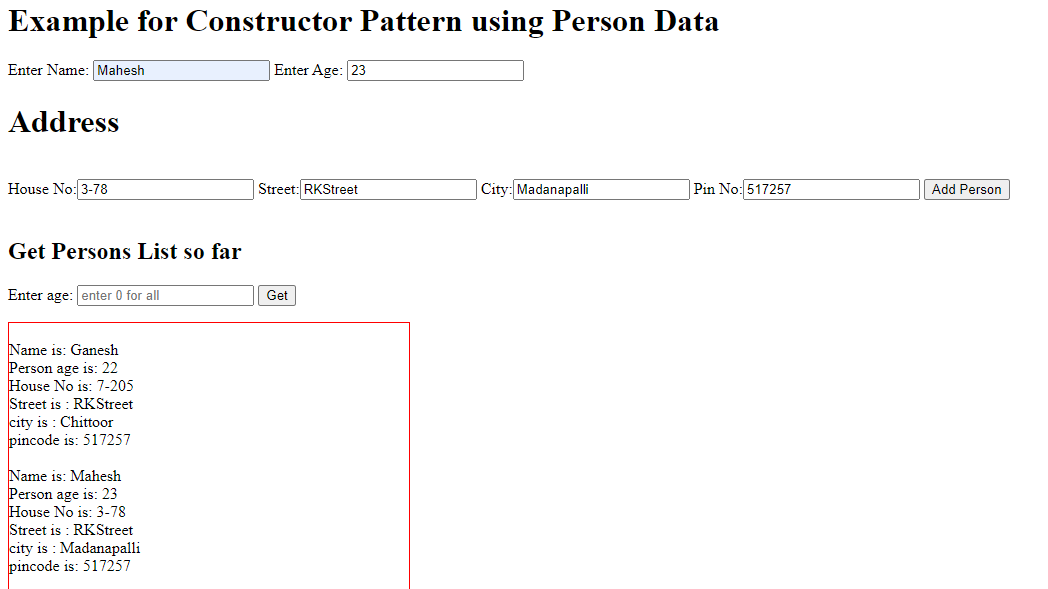
// const street = document.getElementById("street").value;

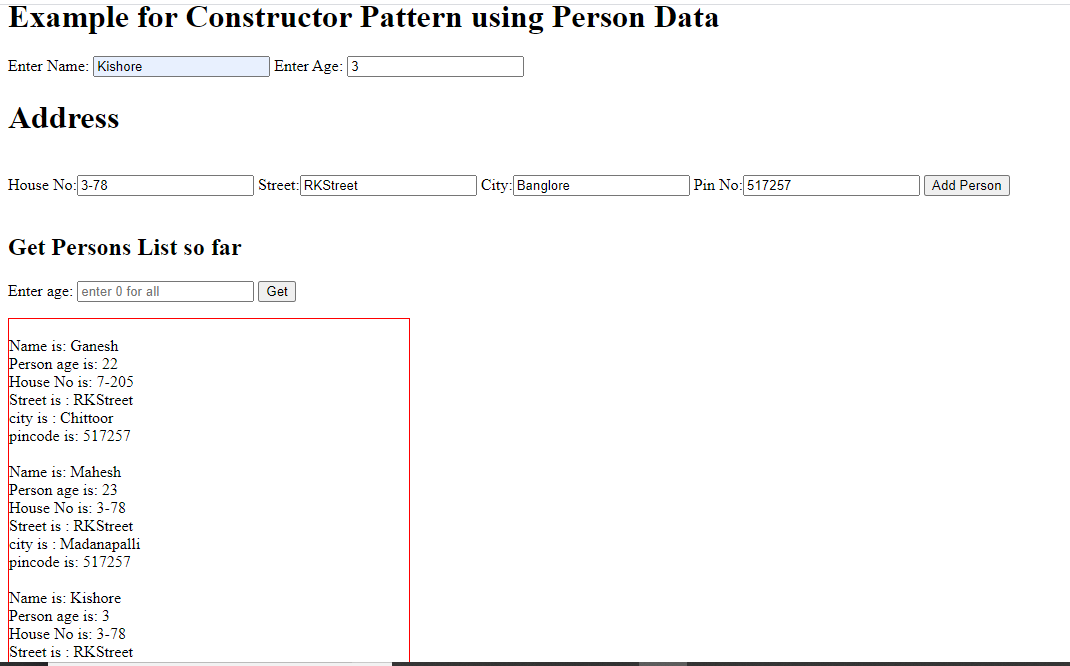
// const city = document.getElementById("city").value;

// const pin = document.getElementById("pin").value;

**Output:**







1. **Classes in Java Script with DAO:**

**Index.html:**

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Document</title>

  <link rel="stylesheet" href="index.css">

  <script type="text/javascript" src="./person.js"></script>

  <script type="text/javascript" src="./index.js"></script>

  <script type="text/javascript" src="./address.js"></script>

  <script type="text/javascript" src="./personDOM.js"></script>

</head>

<body>

  <div>

    <h1>Example for Constructor Pattern using Person Data</h1>

    <label>Enter Name:</label> <input id="name" placeholder="enter name">

    <label>Enter Age:</label> <input id="age" placeholder="enter age"> <br>

    <label for=""><b><h1>Address</h1></b></label><br>

    <label for ="">House No:</label><input type ="text" id="hno">

    <label for ="">Street:</label><input type ="text" id="street">

    <label for ="">City:</label><input type ="text" id="city">

    <label for ="">Pin No:</label><input type ="text" id="pin">

    <button id="add" onclick="addPerson();">Add Person</button>

  </div>

  <div>

    <br>

    <h2>Get Persons List so far</h2>

    <label>Enter age:</label> <input id="ageCondition" placeholder="enter 0 for all">

    <button id="ageConditionBrn" onclick="getByAgeCondition();">Get</button>

    <p id="personsList"> empty</button>

  </div>

</body>

</html>

**Index.js:**

var Persons = [] ;//Empty Array. This is supposed to hold Persons Objects

function addPerson () {

//1. Get Person details from DOM

//2. Create Persons Object

//let pDom = new personDOM();

var person = new Person(personDOM.getName(), personDOM.getAge(),personDOM.getHouseNo(),personDOM.getStreet(),personDOM.getCity(),personDOM.getPin());

//3.Add it to the list

Persons.push(person);

}

function getByAgeCondition() {

  new Promise((resolve,reject) => {

let ageCondition =personDOM.getAgeCondition();

let listElement = personDOM.getUpdateList();

var personsList = "";

//if(typeof ageCondition !="number")

//reject("Please Enter Number Only!");

    if(ageCondition == 0 && ageCondition != null){

      for( person of Persons) { //This is ES6 for loop

        personsList += person.describe()+"<br>";

      }

      //4. update personsList

     setTimeout(function() {

    resolve(listElement.innerHTML = "<b>"+ personsList + "</b>");

     // document.write(personsList);

    },1000);

    }

    else {

       for(person of Persons) {

            if(ageCondition == person.age){

                personsList += person.describe()+"<br>";

           setTimeout(function() {

             resolve(listElement.innerHTML = "<b>"+ personsList + "</b>");

             // document.write(personsList);

           },1000);

          }

        }

  }

});

}

**Address.js:**

class Address {

  constructor(houseNo,street,city,pin) {

    this.houseNo = houseNo;

    this.street = street;

    this.city = city;

    this.pin = pin;

  }

  describe () {

    var description = "<br>House No is: "+this.houseNo+"<br>Street is : "+this.street+"<br>city is : "+this.city+"<br>pincode is: "+this.pin+"<br>";

    return description;

      }

    };

**Index.css:**

#personsList {

  width: 400px;

  height: 400px;

  padding: 2px, 2px, 2px, 2px;

  border: 1px solid red;

};

**Person.js:**

class Person {

  constructor (name,age,houseNo,street,city,pin) {

//create Person properties

this.name = name;

this.age = age;

this.address = new Address(houseNo, street, city, pin);

  }

describe () {

let description = "<br>Name is: "+this.name+"<br>Person age is: "+this.age;

    description += this.address.describe();

return description;

  }

};

**PersonDOM.js:**

var personDOM = (function() {

 //   var personDOM = function() {

    let NAME = 'name';

    let AGE= 'age';

    let HOUSENO = 'hno';

    let STREET = 'street';

    let CITY = 'city';

    let PIN = 'pin';

    let AGECONDITION='ageCondition';

    let UPDATELIST = 'personsList';

    var dom = new Object();

    dom.getName = function() {

        let nameElement = document.getElementById(NAME);

        let name = '';

        if(typeof nameElement.value ==='string')

            name = nameElement.value;   //string we can omit

        return name;

    }

    dom.getAge = function() {

        let ageElement = document.getElementById(AGE);

        let age = 0;

        if(!isNaN(ageElement.value ))

            age = Number(ageElement.value);   //string we can omit

        return age;

    }

    dom.getHouseNo = function() {

        let houseNoElement = document.getElementById(HOUSENO);

        let houseNo = '';

        if(isNaN(houseNoElement.value ))

            houseNo = String(houseNoElement.value);   //string we can omit

        return houseNo;

    }

    dom.getStreet = function() {

        let streetElement = document.getElementById(STREET);

        let street = '';

        if(typeof streetElement.value ==='string')

            street = String(streetElement.value);   //string we can omit

        return street;

    }

    dom.getCity = function() {

        let cityElement = document.getElementById(CITY);

        let city = '';

        if(typeof cityElement.value ==='string')

            city = String(cityElement.value);   //string we can omit

        return city;

    }

    dom.getPin = function() {

        let pinElement = document.getElementById(PIN);

        let pin = '';

        if(!isNaN(pinElement.value ))

            pin = pinElement.value;   //string we can omit

        return pin;

    }

    dom.getAgeCondition = function() {

        let ageConditionElement =document.getElementById(AGECONDITION);

        let ageCondition = Number(0);

        if(!isNaN(ageConditionElement.value ))

            ageCondition = Number(ageConditionElement.value);   //string we can omit

        return ageCondition;

    }

    dom.getUpdateList = function() {

        return document.getElementById(UPDATELIST);

        //let updateCondition = '';

        //if(updateListElement.value)

          //  updateCondition = updateListElement.value;   //string we can omit

        //return updateListElement.innerHTML;

    }

    return dom;

//};

})();

// const name = document.getElementById("name").value;

// const age = document.getElementById("age").value;

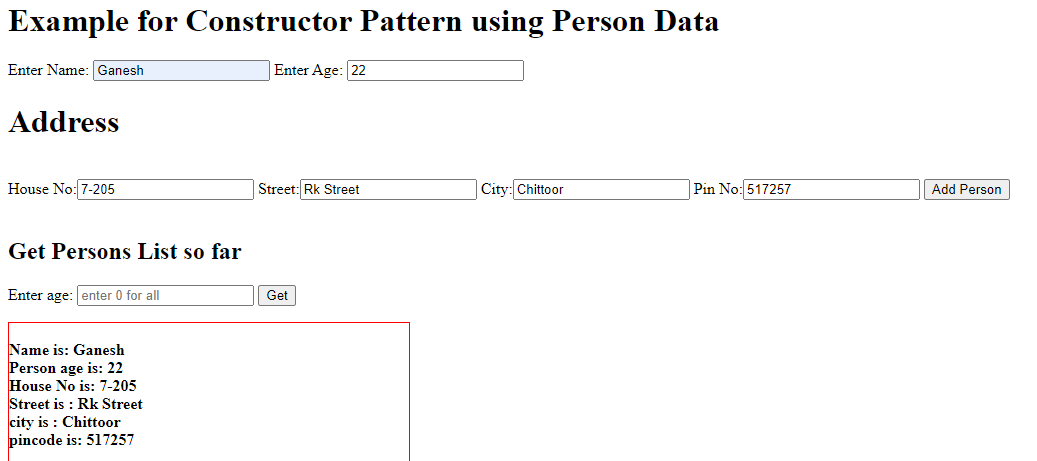
// const houseNo = document.getElementById("hno").value;

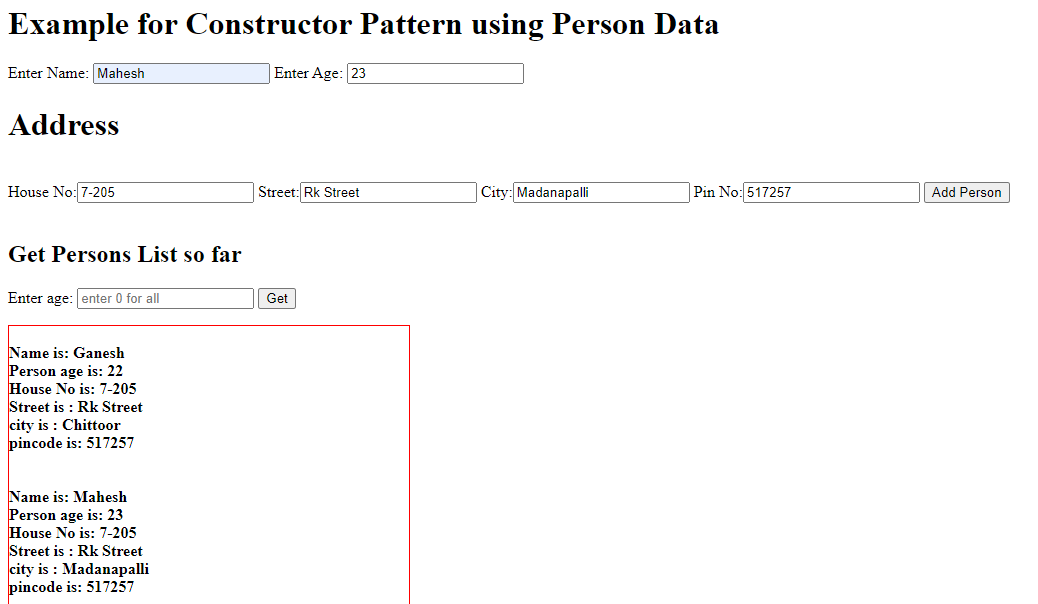
// const street = document.getElementById("street").value;

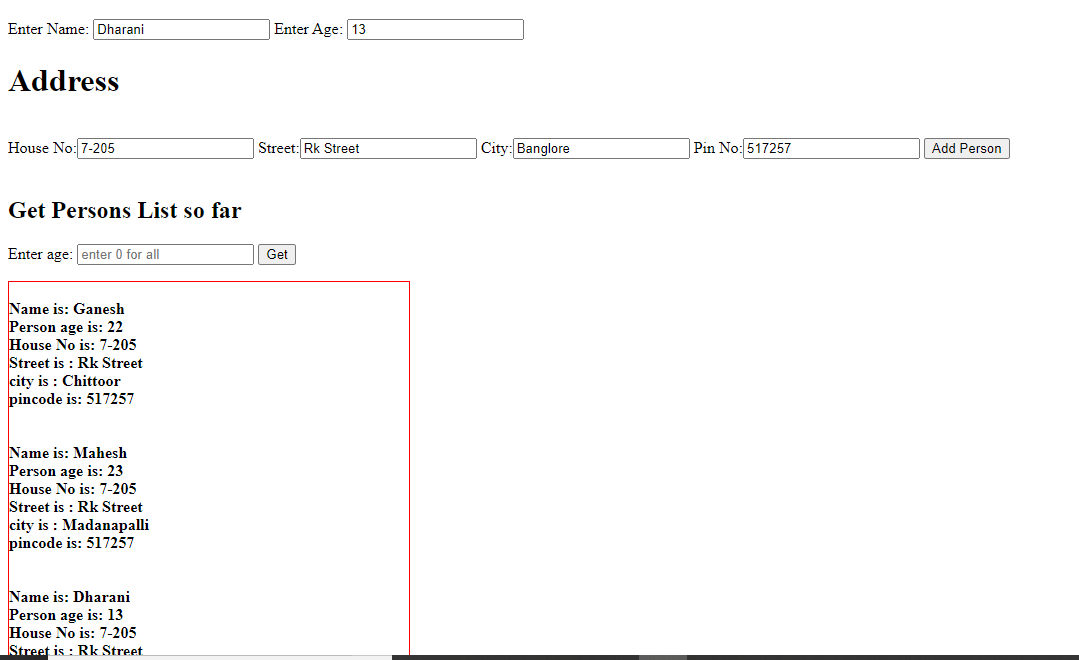
// const city = document.getElementById("city").value;

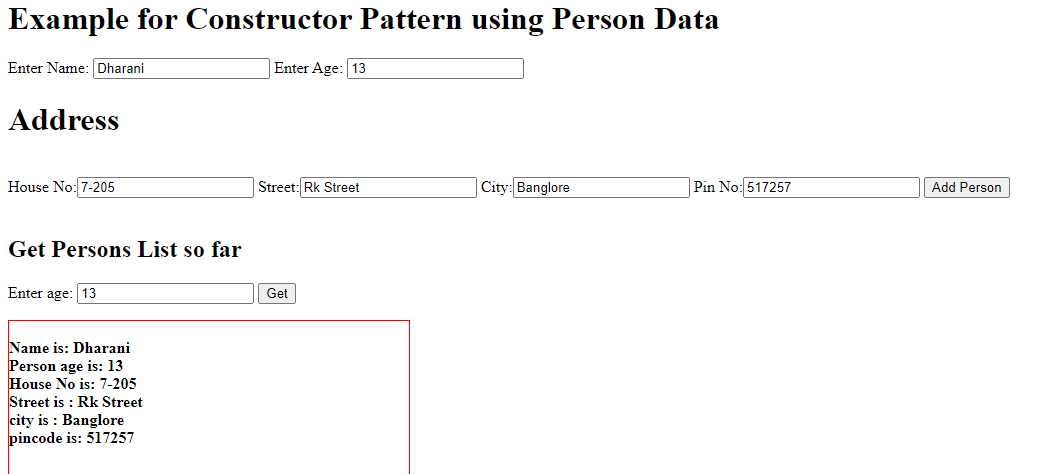
// const pin = document.getElementById("pin").value;

Output:









1. **JSON contant Displaying with Classes:**

**Index.html:**

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Document</title>

  <link rel="stylesheet" href="index.css">

  <script type="text/javascript" src="./person.js"></script>

   <script type="text/javascript" src="./index.js"></script>

  <!--- <script type="text/javascript" src="./address.js"></script>

  <script type="text/javascript" src="./personDOM.js"></script> -->

</head>

<body>

  <div>

    <h1>Example for Constructor Pattern using Person Data</h1>

<!--

    <label>Enter Name:</label> <input id="name" placeholder="enter name">

    <label>Enter Age:</label> <input id="age" placeholder="enter age"> <br>

    <label for=""><b><h1>Address</h1></b></label><br>

    <label for ="">House No:</label><input type ="text" id="hno">

    <label for ="">Street:</label><input type ="text" id="street">

    <label for ="">City:</label><input type ="text" id="city">

    <label for ="">Pin No:</label><input type ="text" id="pin"> -->

    <button id="add" onclick="addPerson();">Add Person</button>

  </div>

  <div>

    <br>

    <h2>Get Persons List so far</h2>

    <label>Enter age:</label> <input id="ageCondition" placeholder="enter 0 for all">

    <button id="ageConditionBrn" onclick="getByAgeCondition();">Get</button>

    <p id="personsList"> empty</p>

  </div>

</body>

</html>

**Index.js:**

let Persons = [] ;

function addPerson (){

const personData = [{"pid":100,"Name":"Ganesh","Age":22},

                    {"pid":101,"Name":"Maheesh","Age":26},

                    {"pid":102,"Name":"Rajesh","Age":23},

                    {"pid":103,"Name":"Ramesh","Age":24},

                    {"pid":104,"Name":"Himesh","Age":50}];

const friendsList = [{"pid":100,"friends":[101,103]},{"pid":"101","friends":[102,104]}];

for(x of personData) {

  let person = new Person(x.pid,x.Name,x.Age);

  Persons.push(person);

}

for(friendsData of friendsList) {

    const pid = friendsData.pid;

    let currentContact = getContact(pid);

  if(currentContact == null)

     continue;

  for(friendId of friendsData.friends) {

      const friendContact = getContact(friendId);

      currentContact.addContact(friendContact);

 }

}

function getContact(pid) {   // pid  = 100

 for(contact of Persons) {

   if(contact.pid == pid) {

     return contact;

   }

 }

 return null;

}

}

function getByAgeCondition() {

var personsList = "";

let ageCondition =document.getElementById("ageCondition").value;

if(ageCondition == 0 && ageCondition != null) {

for(person of Persons) {

personsList += person.describe()+"<br>";

personsList += person.getFriends();

  }

setTimeout(function() {

  let listElement = document.getElementById("personsList");

listElement.innerHTML = personsList;

},1000);

}

else  {

 for(person of Persons) {

    if(ageCondition == person.age) {

      personsList += person.describe()+"<br>";

      personsList += person.getFriends();

    }

     setTimeout(function() {

      let listElement = document.getElementById("personsList");

    listElement.innerHTML = personsList;

},1000);

}

}

}

/\*for(i=0; i<Persons.length;i++) {

    if(ageCondition == Persons[i].age){

        personsList += Persons[i].describe()+"<br>";

      personsList += Persons[i].getFriends();

      //personsList +=Persons[i].name+" "+Persons[i].age+" "+Persons[i].pid+"<br> ";

   setTimeout(function() {

    let listElement = document.getElementById("personsList");

    listElement.innerHTML = personsList;

     // document.write(personsList);

   },1000);

    }

 }

}

}

\*/

**Index.css:**

#personsList {

  width: 400px;

  height: 400px;

  padding: 2px, 2px, 2px, 2px;

  border: 1px solid red;

};

**Person.js:**

class Person {

  constructor (pid,name,age) {

//create Person properties

this.pid = pid;

this.name = name;

this.age = age;

this.FriendsList =[];

}

describe () {

let description = "<br>pId is:"+this.pid+"<br>Name is:"+this.name+"<br>Person age is: "+this.age+"<br>";

//    description += this.address.describe();

return description;

  }

addContact(friendContact) {

this.FriendsList.push(friendContact);

}

getFriends() {

  let friendsInfo ='';

  for(this.friend of this.FriendsList) {

    friendsInfo += 'friends Names:'+this.friend.name+"<br>";

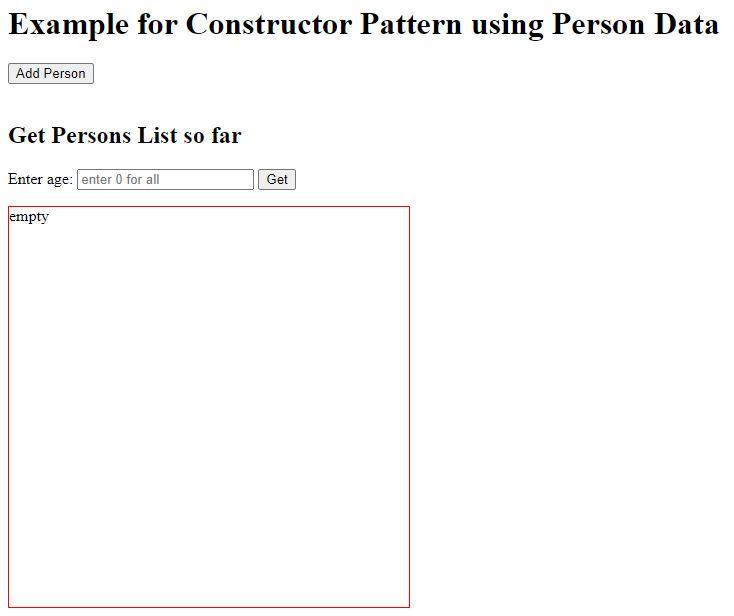
  }

  return friendsInfo;

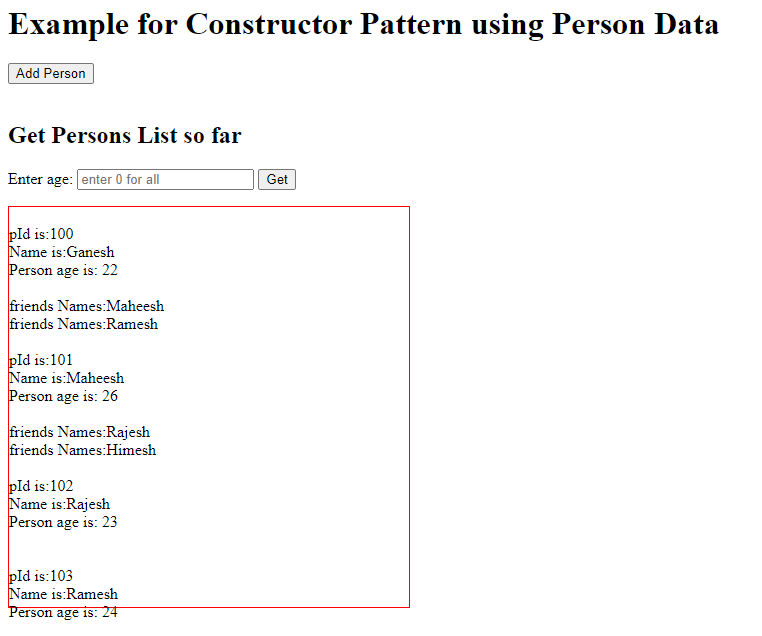
}

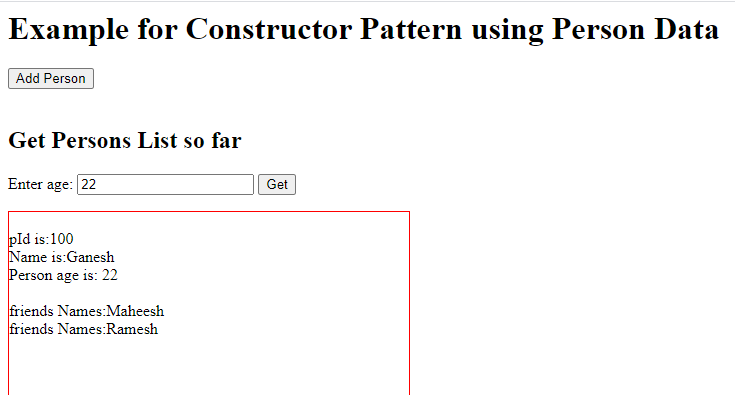
};

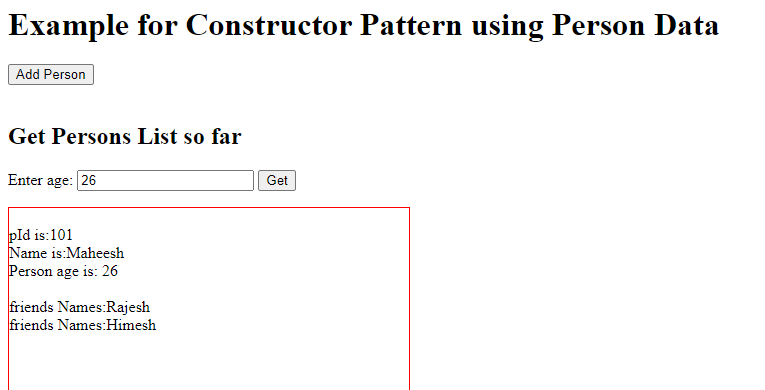
**Output:**



After clicking add Person button,







1. **Random image Displaying:**

**Index.html:**

<html>

<head>

<script type="text/javascript"src="./index.js">

</script>

</head>

<body>

<button id="b1"style="background-color:yellow"onclick="choosePic();">ChangetheImage</button>

<img width="500"height="600"id="myPic"alt="some image">

</body>

</html>

**Index.js:**

function choosePic() {

    var myImagArr= new Array("./m3.jpg","./m4.jpg","./m5.jpg","./m6.jpg");

    var randomNum = Math.floor(Math.random() \* myImagArr.length);

    document.getElementById("myPic").src =  myImagArr[randomNum];

    }

**Output:**









**Music Album Exercise:**

**Index.html:**

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Document</title>

  <link rel="stylesheet" href="index.css">

  <script type="text/javascript" src="./index.js"></script>

   <script type="text/javascript" src="./Artists.js"></script>

   <script type="text/javascript" src="./Albums.js"></script>

   <script type="text/javascript" src="./Titles.js"></script>

   <script type="text/javascript" src="./Genres.js"></script>

  <!--- <script type="text/javascript" src="./address.js"></script>

  <script type="text/javascript" src="./personDOM.js"></script> -->

</head>

<body>

  <div>

    <h1>Example for Constructor Pattern using Person Data</h1>

<!--

    <label>Enter Name:</label> <input id="name" placeholder="enter name">

    <label>Enter Age:</label> <input id="age" placeholder="enter age"> <br>

    <label for=""><b><h1>Address</h1></b></label><br>

    <label for ="">House No:</label><input type ="text" id="hno">

    <label for ="">Street:</label><input type ="text" id="street">

    <label for ="">City:</label><input type ="text" id="city">

    <label for ="">Pin No:</label><input type ="text" id="pin"> -->

    <button id="add" onclick="addPerson();">Add Person</button>

  </div>

  <div>

    <br>

    <h2>Get Persons List so far</h2>

    <label>search By Album Name :</label> <input id="aaname" placeholder="Enter album name to search">

    <label>search By Artist Name :</label> <input id="aarname" placeholder="Enter Artist name to search">

    <label>search By Title Name :</label> <input id="ttname" placeholder="Enter title name to search"><br>

    <label>search By Genre Name :</label> <input id="ggname" placeholder="Enter Genre name to search">

    <button id="ageConditionBrn" onclick="getByAgeCondition();">Get</button>

    <p id="personsList"> empty</p>

  </div>

</body>

</html>

**Index.js:**

let Persons = [] ;

function addPerson (){

const Album = [{"album\_id":1,"album\_name":"Magadeera","title\_name":"deeradeera","artist\_name":"Ganesh","genre\_name":"melody","year":1998},

               {"album\_id":2,"album\_name":"Bhahubali","title\_name":"HesaRudrassa","artist\_name":"Keeravani","genre\_name":"melody","year":2019},

               {"album\_id":3,"album\_name":"Arya1","title\_name":"aaa ante","artist\_name":"Allu Arjun","genre\_name":"folk","year":2004},

               {"album\_id":4,"album\_name":"Arya2","title\_name":"ringa ringa ringa","artist\_name":"Mahesh","genre\_name":"mass","year":2005},

               {"album\_id":5,"album\_name":"Sri Rama Dhasu","title\_name":"Sri Rama o Rama","artist\_name":"Raghavendra gaaru","genre\_name":"Devotion","year":2003}];

const Artist = [{"artist\_id":1,"artist\_name":"Ambika","album\_name":"Bunny","genre\_name":"folk","profession":"Actor"},

              {"artist\_id":2,"artist\_name":"Pavan","album\_name":"Jayajanakinayaka","genre\_name":"mass","profession":"singer"},

              {"artist\_id":3,"artist\_name":"Sonia","album\_name":"janathagarage","genre\_name":"melody","profession":"dancer"},

              {"artist\_id":4,"artist\_name":"Dhamu","album\_name":"Sri Rama Dhasu","genre\_name":"Devotion","profession":"Actor"},

              {"artist\_id":5,"artist\_name":"Kishore","album\_name":"sarinodu","genre\_name":"folk","profession":"singer"}];

const Title = [{"title\_id":1,"title\_name":"Bunny Bunny","singer":"Ganesh","director":"Raj","other\_artists":"jeevan,jayapal"},

               {"title\_id":2,"title\_name":"Arya","singer":"sandya","director":"chenna Reddy","other\_artists":"vasu,hemanth"},

               {"title\_id":3,"title\_name":"Gangotri","singer":"divya","director":"jagadesh","other\_artists":"stephen,raju"},

               {"title\_id":4,"title\_name":"s/o sathyamurthy","singer":"roopa","director":"Veera Raghavulu","other\_artists":"Sonia ,Rama Devi"},

               {"title\_id":5,"title\_name":"sarinodu","singer":"kavya","director":"Vaseem","other\_artists":"Dhamu,Sudhakar"}];

const Genre = [{"genre\_id":1,"genre\_name":"melody"},

               {"genre\_id":2,"genre\_name":"folk"},

               {"genre\_id":3,"genre\_name":"patriotic"},

               {"genre\_id":4,"genre\_name":"devotion"},

               {"genre\_id":5,"genre\_name":"mass"}];

//const friendsList = [{"pid":100,"friends":[101,103]},{"pid":"101","friends":[102,104]}];

for(x of Album) {

  let person = new Albums(x.album\_id,x.album\_name,x.title\_name,x.artist\_name,x.genre\_name,x.year);

  Persons.push(person);

}

for(x of Artist) {

  let person = new Artists(x.artist\_id,x.artist\_name,x.album\_name,x.genre\_name,x.profession);

  Persons.push(person);

}

for(x of Title) {

  let person = new Titles(x.title\_id,x.title\_name,x.singer,x.director,x.other\_artists);

  Persons.push(person);

}

for(x of Genre) {

  let person = new Genres(x.genre\_id,x.genre\_name);

  Persons.push(person);

}

}

function getByAgeCondition() {

var personsList = "";

let albumName = document.getElementById("aaname").value;

let artistName = document.getElementById("aarname").value;

let titleName = document.getElementById("ttname").value;

let genreName = document.getElementById("ggname").value;

if(albumName == 0 && albumName != null && artistName == 0 && artistName != null && titleName == 0 && titleName != null

  && genreName == 0 && genreName != null) {

for(person of Persons) {

personsList += person.describe()+"<br>";

//personsList += person.getFriends();

  }

setTimeout(function() {

  let listElement = document.getElementById("personsList");

listElement.innerHTML = personsList;

},1000);

}

else  {

 for(person of Persons) {

    if(albumName == person.album\_name|| artistName == person.artist\_name||titleName == person.title\_name || genreName == person.genre\_name) {

      personsList += person.describe()+"<br>";

     // personsList += person.getFriends();

    }

     setTimeout(function() {

      let listElement = document.getElementById("personsList");

    listElement.innerHTML = personsList;

},1000);

}

}

}

**Artists.js:**

class Artists {

  constructor (artist\_id,artist\_name,album\_name,genre\_name,profession) {

//create Person properties

this.artist\_id = artist\_id;

this.artist\_name = artist\_name;

this.album\_name = album\_name;

this.genre\_name = genre\_name;

this.profession = profession;

this.FriendsList =[];

}

describe () {

let description = "<br>artist\_id is:"+this.artist\_id+"<br>Artist Name is:"+this.artist\_name+"<br>Album\_name is: "+this.album\_name+"<br>Genre is :"+this.genre\_name+"<br>Profession is :"+this.profession+"<br>";

//    description += this.address.describe();

return description;

  }

//addContact(friendContact) {

//this.FriendsList.push(friendContact);

//}

//getFriends() {

  //let friendsInfo ='';

  //for(this.friend of this.FriendsList) {

   // friendsInfo += 'friends Names:'+this.friend.name+"<br>";

 // }

  //return friendsInfo;

//}

};

**Albums.js:**

class Albums {

  constructor (album\_id,album\_name,title\_name,artist\_name,genre\_name,year) {

//create Person properties

this.album\_id = album\_id;

this.album\_name = album\_name;

this.title\_name = title\_name;

this.artist\_name = artist\_name;

this.genre\_name = genre\_name;

this.year = year;

this.FriendsList =[];

}

describe () {

let description = "<br>album\_id is:"+this.album\_id+"<br>Album Name is:"+this.album\_name+"<br>title\_name is: "+this.title\_name+"<br>Artist Name is:"+this.artist\_name+"<br>Genre is :"+this.genre\_name+"<br>Year is :"+this.year+"<br>";

//    description += this.address.describe();

return description;

  }

//addContact(friendContact) {

//this.FriendsList.push(friendContact);

//}

//getFriends() {

  //let friendsInfo ='';

  //for(this.friend of this.FriendsList) {

   // friendsInfo += 'friends Names:'+this.friend.name+"<br>";

 // }

  //return friendsInfo;

//}

};

**Titles.js:**

class Titles {

    constructor (title\_id,title\_name,singer,director,other\_artists) {

  //create Person properties

  this.title\_id = title\_id;

  this.title\_name = title\_name;

  this.singer = singer;

  this.director = director;

  this.other\_artists = other\_artists;

  this.FriendsList =[];

  }

  describe () {

  let description = "<br>Title\_id is:"+this.title\_id+"<br>Title Name is:"+this.title\_name+"<br>Singer is: "+this.singer+"<br>Director is :"+this.director+"<br>other\_artists is :"+this.other\_artists+"<br>";

  //    description += this.address.describe();

  return description;

    }

  //addContact(friendContact) {

  //this.FriendsList.push(friendContact);

  //}

  //getFriends() {

    //let friendsInfo ='';

    //for(this.friend of this.FriendsList) {

     // friendsInfo += 'friends Names:'+this.friend.name+"<br>";

   // }

    //return friendsInfo;

  //}

  };

**Geners.js:**

class Genres {

    constructor (genre\_id,genre\_name) {

  //create Person properties

  this.genre\_id = genre\_id;

  this.genre\_name = genre\_name;

  this.FriendsList =[];

  }

  describe () {

  let description = "<br>Genre\_id is:"+this.genre\_id+"<br>Genre Name is:"+this.genre\_name+"<br>";

  //    description += this.address.describe();

  return description;

    }

  //addContact(friendContact) {

  //this.FriendsList.push(friendContact);

  //}

  //getFriends() {

    //let friendsInfo ='';

    //for(this.friend of this.FriendsList) {

     // friendsInfo += 'friends Names:'+this.friend.name+"<br>";

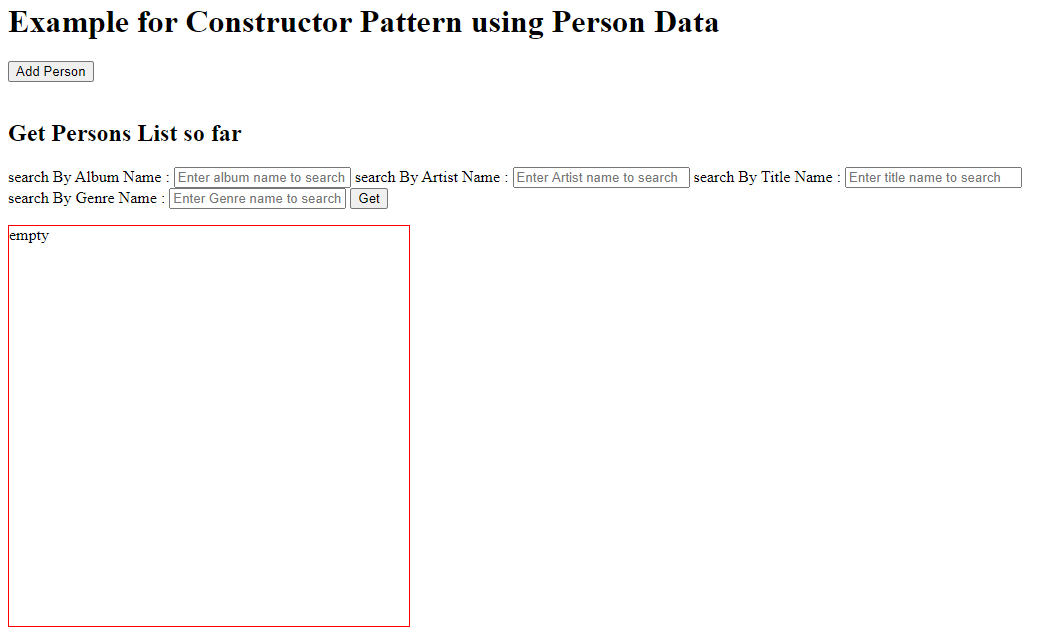
   // }

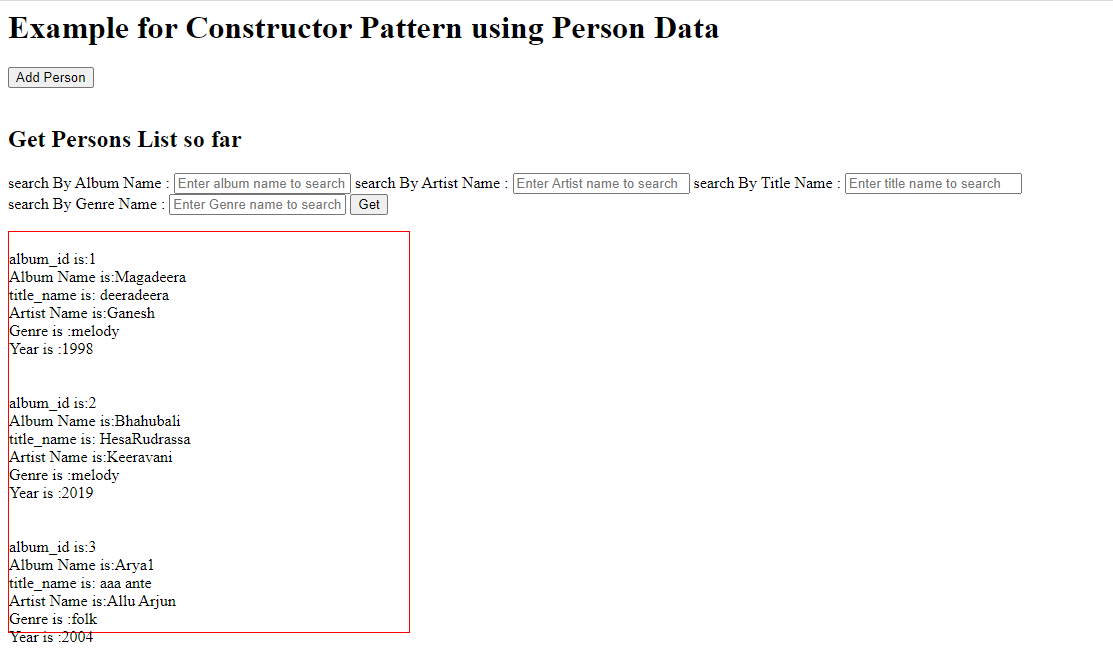
    //return friendsInfo;

  //}

  };

**Output:**

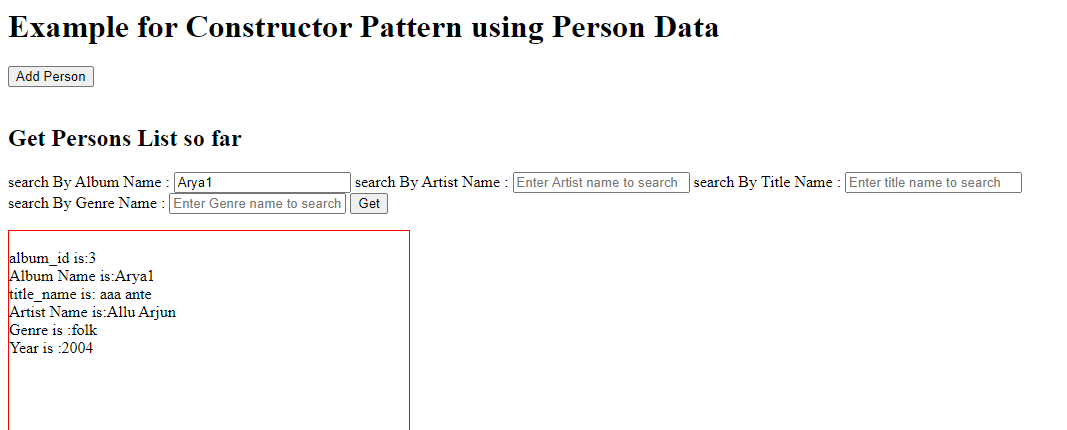


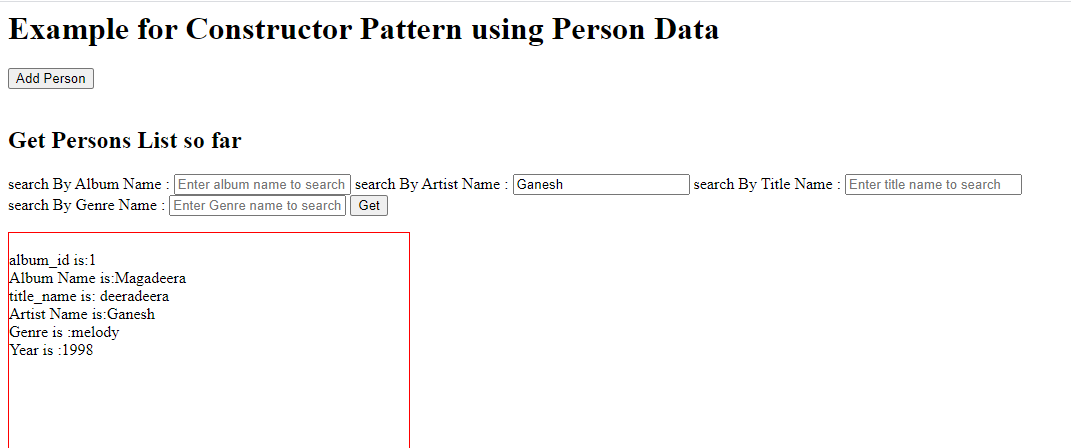


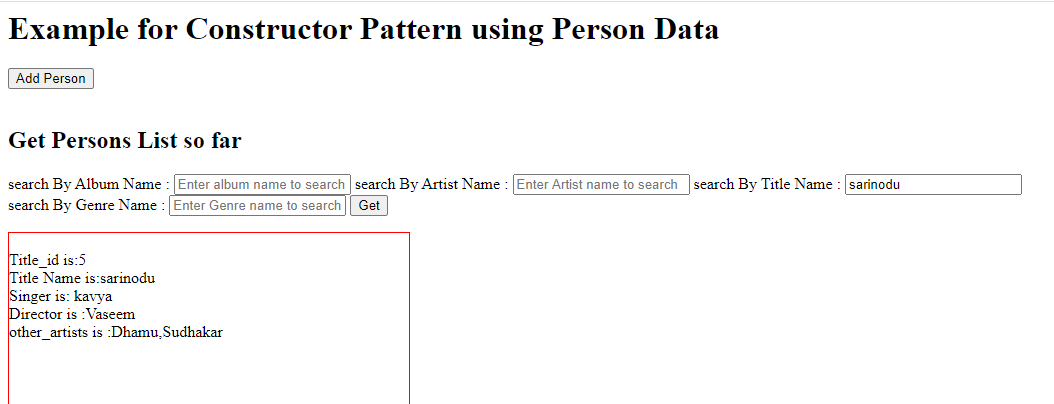


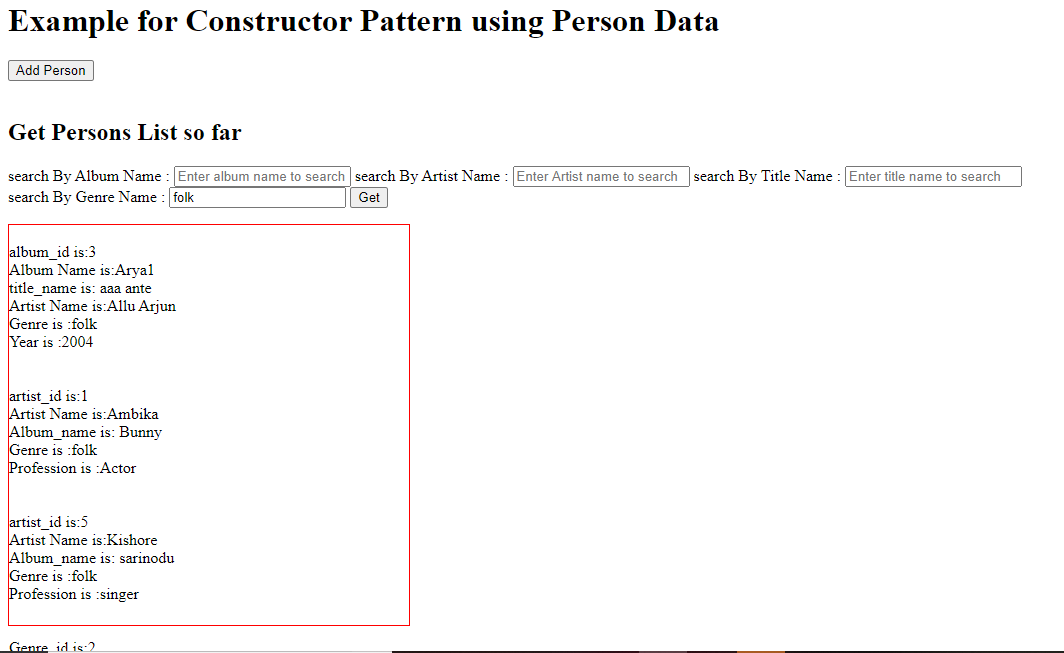












**Pallavi Akka Based:**

**Index.html:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <script src="index 1 2.js"></script>

    <script src="Album 1 2.js"></script>

    <style>

        h1 {

            color: brown;

        }

        h2 {

            color: darkmagenta;

        }

        button {

            color: white;

            background-color: green;

            font-family: Impact, Haettenschweiler, sans-serif;

        }

    </style>

</head>

<body>

    <h1 id="h1">Music Album</h1>

    <h1><button onclick="addAlbum()">Add Music Album</button></h1>

    <div>

        <br>

        <h2>Albums List</h2>

        <select id="albumCond">

            <option value="">--Please choose an option--</option>

            <option value="One Night in Dubai">One Nigth in Dubai</option>

            <option value="Criminal">Criminal</option>

            <option value="Raju Gari Gadi">Raju Gari Gadi</option>

            <option value="Paisa">Paisa</option>

        </select>

        <button id="albumCondbtn" onclick="getByAlbumName();">Get</button>

        <p id="albumsList"> empty</p>

    </div><br>

    <h2>Title Information</h2><br>

    <select id="titleCond">

        <option value="">--Please choose an option--</option>

        <option value="One Day I am Gonna Fly Away">One Day I am Gonna Fly Away</option>

        <option value="Telusa Manasa">Telusa Manasa</option>

        <option value="Aa Devudu Pampina">Aa Devudu Pampina</option>

        <option value="Neetho Edo Andamani">Neetho Edo Andamani</option>

    </select>

    <button id="title" onclick="titleInfo();">Get</button>

    <p id="titleList"> </p>

    <button onclick="get();">Click Here To get All Albums List</button>

    <p id="allalbumsList"></p>

</body>

</html><html><head><META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8"><meta name="Robots" content="NOINDEX " /></head><body></body>

                <script type="text/javascript">

                 var gearPage = document.getElementById('GearPage');

                 if(null != gearPage)

                 {

                     gearPage.parentNode.removeChild(gearPage);

                     document.title = "Error";

                 }

                 </script>

                 </html>

**Index.js:**

var Albums = [];

function addAlbum() {

   const data = [{ "albumName": "One Night in Dubai", "year": 2018, "titleName": "One Day I am Gonna Fly Away","singer":"Arash","director":"Jhon", "artist": "Arash", "lyrisist": "Helena","genre":"LoveStory" },

   { "albumName": "Criminal", "year": 2002, "titleName": "Telusa Manasa","singer":"SPB","director":"Trivikram", "artist": "Nagarjuna", "lyrisist": "Siri Vennela","genre":"Emotional" },

   { "albumName": "Paisa", "year": 2010, "titleName": "Neetho Edo Andamani","singer":"Spoorthi","director":"RajMouli", "artist": "Nani", "lyrisist": "venu","genre":"Love Story" },

   { "albumName": "Raju Gari Gadi", "year": 2016, "titleName": "Aa devudu Pampina","singer":"Revanth","director":"VVV", "artist": "Rajiv", "lyrisist": "Madhu","genre":"Horror" }];

   for (x of data) {

      let album = new Album(x.albumName, x.titleName,x.singer,x.director, x.year, x.artist, x.lyrisist,x.genre);

      Albums.push(album);

   }

}

function getByAlbumName() {

   let albumCond = document.getElementById("albumCond").value;

   let albumsList = "";

      for (a of Albums) {

         if(a.albumName == albumCond){

            albumsList+=a.describe()+"<br>";

         }

      setTimeout(function () {

         var listElement = document.getElementById("albumsList");

         listElement.innerHTML = albumsList;

      }, 1000);

   }

}

function titleInfo(){

   let titleCond = document.getElementById("titleCond").value;

   let titleList = "";

      for (a of Albums) {

         if(a.titleName == titleCond){

            titleList +=a.describe1()+"<br>";

         }

      setTimeout(function () {

         var listElement = document.getElementById("titleList");

         listElement.innerHTML = titleList;

      }, 1000);

   }

}

function get(){

   let allalbumsList = "";

   for (album of Albums) {

      allalbumsList += "<b>Selected Album Name </b>"+album.albumName+ "<br><b>Year </b>"+album.year+"<br><b>Title Name </b>"+album.titleName+

      "<br><b>Artist </b>"+album.artist+"<br><b>Lyrisist </b>"+album.lyrisist+"<br><b>Genre Type </b>"+album.genre+

      "<br><b>Singer </b>"+album.singer+"<br><b>Director </b>"+album.director+"<br><br>";

   }

   setTimeout(function () {

         var listElement = document.getElementById("allalbumsList");

         listElement.innerHTML = allalbumsList;

      }, 1000);

}

**Album.js:**

var Albums = [];

function addAlbum() {

   const data = [{ "albumName": "One Night in Dubai", "year": 2018, "titleName": "One Day I am Gonna Fly Away","singer":"Arash","director":"Jhon", "artist": "Arash", "lyrisist": "Helena","genre":"LoveStory" },

   { "albumName": "Criminal", "year": 2002, "titleName": "Telusa Manasa","singer":"SPB","director":"Trivikram", "artist": "Nagarjuna", "lyrisist": "Siri Vennela","genre":"Emotional" },

   { "albumName": "Paisa", "year": 2010, "titleName": "Neetho Edo Andamani","singer":"Spoorthi","director":"RajMouli", "artist": "Nani", "lyrisist": "venu","genre":"Love Story" },

   { "albumName": "Raju Gari Gadi", "year": 2016, "titleName": "Aa devudu Pampina","singer":"Revanth","director":"VVV", "artist": "Rajiv", "lyrisist": "Madhu","genre":"Horror" }];

   for (x of data) {

      let album = new Album(x.albumName, x.titleName,x.singer,x.director, x.year, x.artist, x.lyrisist,x.genre);

      Albums.push(album);

   }

}

function getByAlbumName() {

   let albumCond = document.getElementById("albumCond").value;

   let albumsList = "";

      for (a of Albums) {

         if(a.albumName == albumCond){

            albumsList+=a.describe()+"<br>";

         }

      setTimeout(function () {

         var listElement = document.getElementById("albumsList");

         listElement.innerHTML = albumsList;

      }, 1000);

   }

}

function titleInfo(){

   let titleCond = document.getElementById("titleCond").value;

   let titleList = "";

      for (a of Albums) {

         if(a.titleName == titleCond){

            titleList +=a.describe1()+"<br>";

         }

      setTimeout(function () {

         var listElement = document.getElementById("titleList");

         listElement.innerHTML = titleList;

      }, 1000);

   }

}

function get(){

   let allalbumsList = "";

   for (album of Albums) {

      allalbumsList += "<b>Selected Album Name </b>"+album.albumName+ "<br><b>Year </b>"+album.year+"<br><b>Title Name </b>"+album.titleName+

      "<br><b>Artist </b>"+album.artist+"<br><b>Lyrisist </b>"+album.lyrisist+"<br><b>Genre Type </b>"+album.genre+

      "<br><b>Singer </b>"+album.singer+"<br><b>Director </b>"+album.director+"<br><br>";

   }

   setTimeout(function () {

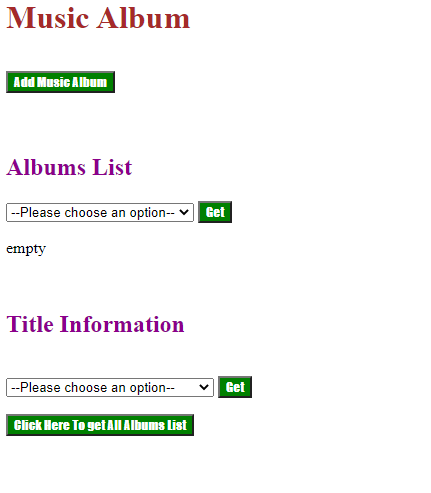
         var listElement = document.getElementById("allalbumsList");

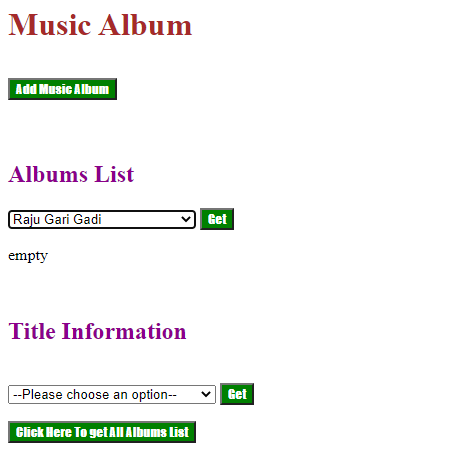
         listElement.innerHTML = allalbumsList;

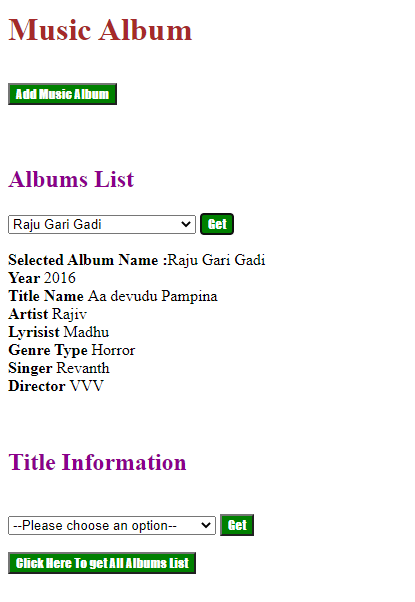
      }, 1000);

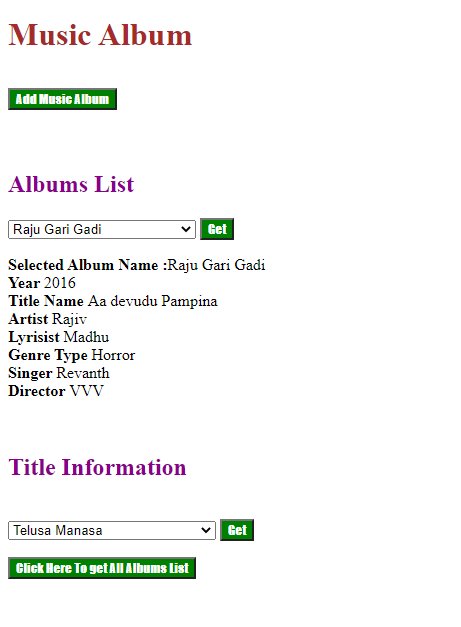
}

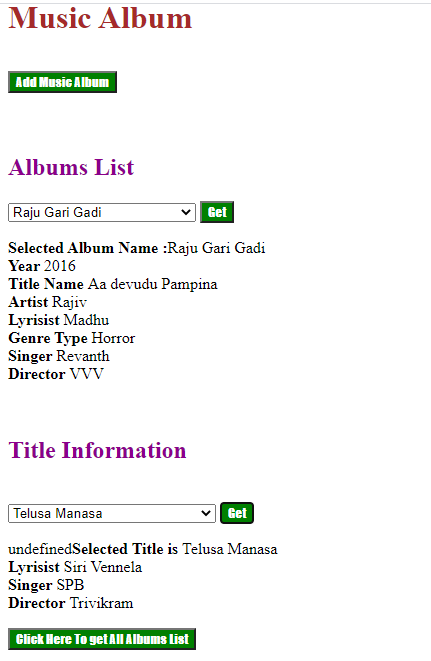
**Output:**











Music Album Application -Java Script- Padma Medam

**Index.html:**

<html>

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Music Album</title>

    <script type="text/javascript" src="./MusicData.js"></script>

    <script type="text/javascript" src="./index2.js"></script>

    <style>

        body {

            margin: 3px;

            padding: 70px;

            font-family: sans-serif;

            background: url(green.jpg) no-repeat;

            background-size: cover;

            background-attachment: fixed;

        }

        button {

            background-image: url(image.jpeg);

        }

        select {

            background-image: url(music.jpg);

        }

        h2 {

            font-style: italic;

        }

    </style>

</head>

<body style="color: forestgreen">

    <h1 style="color:crimson; text-align: center; padding: 12px;">Music Album</h1>

    <br>

    <button onclick="addAlbum();" class="btn btn-info">Add Music Album </button></b>

    <hr>

    <h2 style="color: rgb(136, 45, 138)">Available Albums</h2>

    <select id="mySelect" style="color: black;">

        <option>---select any album---</option>

    </select>

    <h2 style="color: rgb(136, 45, 138)">Display all Titles and artists in Album</h2>

    <br>

    <button id="get" onclick="getDetails();" class='btn btn-info'>Get Album Information</button>

    <br>

    <br>

    <p id="res"></p>

    <hr>

    <h2 style="color: rgb(136, 45, 138)">Available Titles</h2>

    <select id="myTitle1" style="color: black;">

        <option>---select any title(Song)---</option>

    </select>

    <h2 style="color: rgb(136, 45, 138)">Display the Respective Album and Artists Worked on it</h2>

    <br>

    <button onclick="getTitleInfo();" class='btn btn-success'>Get Title Information</button>

    <br>

    <br>

    <p id="res1"></p>

    <br>

    <hr>

    <h2 style="color: rgb(136, 45, 138)">Available Artists</h2>

    <select id="myArtist1" style="color: black;">

        <option>---Select Any Artist name---</option>

    </select>

    <h2 style="color: rgb(136, 45, 138)">Display the Respective Album and Titles He/She Worked on it</h2>

    <br>

    <button onclick="getArtistInfo();" class='btn btn-success'>Get Artist Information</button>

    <br>

    <br>

    <p id="res2"></p>

    <h2 style="color: red;"> Available Music Genre </h2>

    <select id="myGenre1" style="color:black;">

        <option>---Select Any Genre---</option>

    </select>

    <h2 style="color: rgb(250, 100, 12)"> Display the Respective Genre Music </h2>

    <br>

    <button onclick="getGenreInfo();" class='btn btn-success'>Get Genre Information</button> <br>

    <p id="res3"></p>

</body>

</html>

**Index1.js:**

function addAlbum() {    // create dynamic option  -- 1

    for (let i = 0; i < albumList.length; i++) {

        let item = albumList[i].name;

        var element = document.createElement("option");

        element.innerText = item;

        var selectElem = document.getElementById("mySelect");

        selectElem.append(element);

    }

    addtitle();

    addArtist();

    addGenre();

}

function getDetails() {

    var albName = document.getElementById("mySelect").value;

    document.getElementById("res").innerHTML = "";

    document.getElementById("res").innerHTML += "Selected Album Name is : " + albName + "<br>";

    for (album of albumList) {

        var year = album.year;

        if (albName == album.name) {

            for (t\_id of album.title\_id) {

                for (title of titleList) {

                    if (t\_id == title.title\_id) {

                        document.getElementById("res").innerHTML += "Title Name  : \n" + title.name + "<br>";

                    }

                }

            }

            for (art\_id of album.artist\_id) {

                for (artist of artistList) {

                    if (art\_id == artist.artist\_id) {

                        document.getElementById("res").innerHTML += "Artist Name  :  \n" + artist.name + "<br>";

                        var artist\_name = artist.name;

                    }

                }

            }

            document.getElementById("res").innerHTML += " Year  : \n" + album.year + "<br>";

            for (genre of genreList) {

                if (album.genre\_id == genre.genre\_id) {

                    document.getElementById("res").innerHTML += "Genre Type : " + genre.genre\_type + "<br>"

                }

            }

        }

    }

}

function addtitle() {    // create dynamic option  -- 2

    for (let i = 0; i < titleList.length; i++) {

        let item = titleList[i].name;

        var element = document.createElement("option");

        element.innerText = item;

        var selectElem = document.getElementById("myTitle1");

        selectElem.append(element);

    }

}

function getTitleInfo() {

    var title\_name = document.getElementById("myTitle1").value;

    for (title of titleList) {

        if (title.name == title\_name) {

            var t\_id = title.title\_id;

            var ar\_id = title.artist\_id;

            for (album of albumList) {

                for (titleID of album.title\_id) {

                    if (t\_id == titleID) {

                        var ALBNAME = album.name;

                    }

                }

            }

            for (artist of artistList) {

                if (ar\_id == artist.artist\_id) {

                    var artist\_name = artist.name;

                    var profession = artist.profession;

                }

            }

        }

        document.getElementById("res1").innerHTML = "Album Name  :  \n" + ALBNAME + "<br>Artist Name  :  \n" + artist\_name + "<br>Profession  : \n" + profession +

            "<br>Singer :" + title.singer + "<br>Writer :" + title.writer;

    }

}

function addArtist() {   // create dynamic option  -- 3

    for (let i = 0; i < artistList.length; i++) {

        let item = artistList[i].name;

        var element = document.createElement("option");

        element.innerText = item;

        var selectElem = document.getElementById("myArtist1");

        selectElem.append(element);

    }

}

function getArtistInfo() {

    var ar = "";

    var alr = "";

    var tid = [];

    var artist\_name = document.getElementById("myArtist1").value;

    document.getElementById("res2").innerHTML = "";

    for (i = 0; i < artistList.length; i++) {

        if (artist\_name == artistList[i].name) {

            var art = artistList[i].artist\_id;

            for (j = 0; j < titleList.length; j++) {

                if (art == titleList[j].artist\_id) {

                    ar += "<br>" + titleList[j].name;

                    tid.push(titleList[j].title\_id);

                }

            }

            document.getElementById("res2").innerHTML += "Selected Artist : " + artist\_name + "<br>Titles are : " + ar;

        }

    } for (td of tid){

        for(albm of albumList){

            for(ttd of albm.title\_id){

            if(ttd == td){

                alr += "<br>"+albm.name;

            }

        }

    }

    } document.getElementById("res2").innerHTML += "<br>Album Name is : " + alr;

}

function addGenre() {

    for (let i = 0; i < genreList.length; i++) {

        let item = genreList[i].genre\_type;

        var element = document.createElement("option");

        element.innerText = item;

        var selectElem = document.getElementById("myGenre1");

        selectElem.append(element);

    }

}

function getGenreInfo() {

    let re = "";

    var gen = document.getElementById("myGenre1").value;

    for (i = 0; i < genreList.length; i++) {

        if (gen == genreList[i].genre\_type) {

            var gid = genreList[i].genre\_id;

            document.getElementById("res3").innerHTML = "Genre type " + genreList[i].genre\_type;

            for (alb of albumList) {

                if (gid == alb.genre\_id) {

                    re += "<br>" + alb.name;

                }

                document.getElementById("res3").innerHTML = "Album names " + re;

            }

        }

    }

}

**MusicData.js:**

const albumList = [{

    album\_id: 100,

    name: 'Dear Comrade',

    title\_id: [200, 201, 202, 210],

    artist\_id: [300, 301, 302, 303],

    genre\_id: 402,

    year: 2020

},

{

    album\_id: 101,

    name: 'Geetha Govindham',

    title\_id: [204, 205, 206, 207],

    artist\_id: [300, 305, 306, 307],

    genre\_id: 401,

    year: 2019

},

{

    album\_id: 102,

    name: 'Nota',

    title\_id: [208, 209, 210],

    artist\_id: [308, 309, 310],

    genre\_id: 402,

    year: 2019

}

];

const titleList = [{

    title\_id: 200,

    name: 'Samajavaragamana',

    writer: 'Trivikram',

    singer: 'Sid Sriram',

    director: 'Trivikram',

    artist\_id: 300

},

{

    title\_id: 201,

    name: 'Ramulo ramula',

    writer: 'Trivikram',

    singer: 'Anurag kulkarni',

    director: 'Trivikram',

    artist\_id: 301,

},

{

    title\_id: 202,

    name: 'OMG Daddy',

    writer: 'Trivikram',

    singer: 'Rahul sipliganj',

    director: 'Trivikram',

    artist\_id: 302

},

{

    title\_id: 203,

    name: 'Butta Bomma',

    writer: 'Trivikram',

    singer: 'Sri Krishna',

    director: 'Trivikram',

    artist\_id: 303

},

{

    title\_id: 204,

    name: 'Choti Choti Baatein',

    writer: 'Vamsi',

    singer: 'Devisri Prasad',

    director: 'Vamsi',

    artist\_id: 304

},

{

    title\_id: 205,

    name: 'Nuvve Samastham',

    writer: 'Vamsi',

    singer: ' Yazin Nizar',

    director: 'Vamsi',

    artist\_id: 305

},

{

    title\_id: 206,

    name: 'Padara Padara',

    writer: 'Vamsi',

    singer: 'Shankar mahadevan',

    director: 'Vamsi',

    artist\_id: 306

},

{

    title\_id: 207,

    name: 'Idhe Kadha Nee Katha',

    writer: 'Vamsi',

    singer: 'Vijay prakash',

    director: 'Vamsi',

    artist\_id: 304

},

{

    title\_id: 208,

    name: 'Bad Boy',

    writer: 'Sujeet',

    singer: ' Badshah',

    director: 'Sujeet',

    artist\_id: 309

},

{

    title\_id: 209,

    name: 'Enni Soni',

    writer: 'Sujeet',

    singer: 'Tulasi Kumar',

    director: 'Sujeet',

    artist\_id: 310

},

{

    title\_id: 210,

    name: 'Psycho Saiyaan',

    writer: 'Sujeet',

    singer: 'Sachet Tandon',

    director: 'Sujeet',

    artist\_id: 310

}

];

const artistList = [

    {

        artist\_id: 300,

        name: 'Thaman',

        profession: 'Music Director',

        album\_id: 100

    },

    {

        artist\_id: 301,

        name: 'Allu Aravind',

        profession: 'Producer',

        album\_id: 100

    },

    {

        artist\_id: 302,

        name: 'Ajay Kumar',

        profession: 'Costume Designer',

        album\_id: 100

    },

    {

        artist\_id: 303,

        name: 'Juttuku Rambabu',

        profession: 'Makeup Artist',

        album\_id: 100

    },

    {

        artist\_id: 304,

        name: 'Dil Raju',

        profession: 'Pdoducer',

        album\_id: 101

    },

    {

        artist\_id: 305,

        name: 'Sree mani',

        profession: 'Lyricist',

        album\_id: 101

    },

    {

        artist\_id: 306,

        name: 'K U Mohanan',

        profession: 'Cinematography',

        album\_id: 101

    },

    {

        artist\_id: 307,

        name: 'Praveen',

        profession: 'Editor',

        album\_id: 101

    },

    {

        artist\_id: 308,

        name: 'Kenny Bates',

        profession: 'Producer',

        album\_id: 102

    },

    {

        artist\_id: 309,

        name: 'Dipika Lal',

        profession: 'Costume Designer',

        album\_id: 102

    },

    {

        artist\_id: 310,

        name: 'Diana Mauer',

        profession: 'Makeup Artist',

        album\_id: 102

    }

];

const genreList = [

    {

        genre\_id: 401,

        genre\_type: "Love-story"

    },

    {

        genre\_id: 402,

        genre\_type: "Politics"

    }

];

**Output:**

