

## Slip1

Write the HTML5 code for generating the form as shown below. Apply the internal CSS to the following form to change the font size of the heading to 6pt and change the color to red and also change the background color to yellow.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF=8">
    <meta name="viewport " content="width=device-width,initial-scale=1.0">
    <title>Project Mangment</title>
    <style>

      .container{
        max-width:300px;
        margin:0 auto;
        padding:20px;
        background-color:yellow;
        border-radius:5px;

      }
      h1{
        font-size: 24pt;
        color:red;
        text-align: center;
      }

      .button {
        border: none;
        color:blue;
        padding: 4px 8px;
        text-align: center;
        text-decoration: none;
        display: inline-block;
        font-size: 16px;
        margin: 2px 2px;
        cursor: pointer;
      }
      .success {background-color: #04AA6D;} /* Green */
      .primary {background-color: #2500ba;} /* Blue */
    </style>
  </head>
  <body>
    <h1>Project Managment</h1>
    <div class="container">
```

```

<label for="Project name">Project Name </label>
<input id="pname" name="pname" value="pname"><br>
<label for="Assigned To">Assigned To</label>
<select id="teacher" name="teacher">
    <option value="ABC">AbC</option>
    <option value="XYZ">XYZ</option>
    <option value="C">AC</option>
</select><br>
<label for="start date">Start Date</label>
<input type="date" value="start date"><br>
<label for="end date">End Date</label>
<input type="date" value="End date"><br>
<label for="Priority">Priority</label>
<input type="radio" value="high">High</input>
<input type="radio" value="Average">Average</input>
<input type="radio" value="Low">Low</input><br>
<label for="Description">Description</label>
<input type="textarea"><br>
<button type="button" class="button success">Submit</button>
<button type="button" class="button primary">Clear</button>
</div>
</body>
</html>

```

2.

```
db.propertylist.insert({area:"Nashik",rate:200000,city:"Nashik",owner:[{name:"Mr.Patil",age:50}]})
```

```
db.propertylist.insert({area:"FC",rate:150000,owner:[{name:"Mr.Patil",age:50}]})
```

```
db.propertylist.insert({area:"Nashik",rate:100000,city:"Nashik",owner:[{name:"Mr.Jadhav",age:55}]})
```

```
db.propertylist.insert({area:"Mumbai",rate:95000,city:"Mumbai",owner:[{name:"Mr.Roy",age:55}]})
```

```
db.propertylist.insert({area:"Mumbai",rate:80000,city:"Mumbai",owner:[{name:"Mr.Patil",age:55}]})
```

a) Display area wise property details

```
`db.propertylist.find({}, { area: 1, rate: 1, _id: 0 })
```

b) Display property owned by 'Mr. Patil' having minimum rate

```
db.propertylist.find({"owner.name":"Mr.Patil"}).sort({"rate":1}).limit(1)
```

c) Give the details of owner whose property is at "Nashik".

**db.propertylist.find({area:"Nashik"})**

d) Display area of property whose rate is less than 100000.

**db.propertylist.find({rate:{\$lt:100000}})**

## slip2

Create a container add row inside it and add 3 columns inside row using Bootstrap.

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZJEdK2Kadq2F9CUG65"
crossorigin="anonymous">

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-
awesome/4.4.0/css/font-awesome.min.css" />

  </head>
  <body>

    <div class="container text center mt-5">
      <table class="table" width='50%'>
        <tr>
          <th>x</th>
          <th>y</th>
          <th>z</th>
        </tr>
        <tr><td>1</td><td>John</td><td>Berg</td></tr>
        <tr><td>2</td><td>rohan</td><td>Berg</td></tr>
      </table>
    </div>
```

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

2.

```
db.publisher.insertMany([
  { name: "Newspaper House", state:"Gujrat" },
  { name: "Bhaskar", state:"Maharashtra" },
  { name: "Mumbai Mirror", state:"Gujrat" },
  { name: "Sandesh", state:"Gujrat" },
  { name:"ABC",state:"Maharashtra" }
])
```

```
db.newspaper.insertMany([
  { name: "Nashik Times", language: "English",sale:600,city:[ {name:"Nashik",state:
"Maharashtra" } ] },
  { name: "Lockmat", language: "English", sale:450,city:[ {name:"Nashik",state:
"Maharashtra" } ] },
  { name: "Times of India", language: "English",sale:500,city:[ {name:"Nashik",state:
"Maharashtra" } ] },
  { name: "Sandesh", language: "Marathi", sale:350,city:[ {name:"Dhule",state: "UP" } ] },
  { name: "Divya Marathi", language: "English",sale:400, city:[ {name:"Pune",state: "UP" }
]}
])
```

```
db.publisher.insertMany([
...   { name: "Newspaper House", state:"Gujarat" },
...   { name: "Bhaskar", state:"Maharashtra" },
...   { name: "Mumbai Mirror", state:"Gujarat" },
...   { name: "Sandesh", state:"Gujarat" },
...   { name:"ABC",state:"Maharashtra" }
... ])
```

a.List all newspapers available "NASHIK" city

```
db.newspaper.find({"city.name":"Nashik"})
```

b. List all the newspaper of "Marathi" language

```
db.newspaper.find({language:"Marathi"})
```

c. Count no. of publishers of "Gujrat" state

```
db.publisher.countDocuments({ state: "Gujarat" })
```

d. Write a cursor to show newspapers with highest sale in Maharashtra State

```
let cursor = db.newspaper.find({"city.state": "Maharashtra" }).sort({ sale: -1 }).limit(1)
while (cursor.hasNext()) { printjson(cursor.next()); }
```

## slip3

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap demo</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZJEdK2Kadq2F9CUG65"
crossorigin="anonymous">

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-
awesome/4.4.0/css/font-awesome.min.css" />
    <style >

  </style>

</head>
<body>

  <div class="container text center mt-5">
    <h4>Image Gallery</h4>
    <div class='row'>
      <div class='col-sm-6 col-lg-3'>
        
      </div>
      <div class='col-sm-6 col-lg-3'>
        
      </div>
      <div class='col-sm-6 col-lg-3'>
```

```

        
    </div>
    <div class='col-sm-6 col-lg-3'>
        
    </div>
</div>
<div class='clear-fix'>&nbsp;</div>

    <div class='row'>
    <div class='col-sm-6 col-lg-3'>
        
    </div>
    <div class='col-sm-6 col-lg-3'>
        
    </div>
    <div class='col-sm-6 col-lg-3'>
        
    </div>
    <div class='col-sm-6 col-lg-3'>
        
    </div>
</div>

</body>
</html>

```

## 2.

```

db.emp.insertMany([ {name: "John Doe", age: 30, position: "Manager", salary: 70000,
department:[ { name: "Sales", employees_count: 5} ]},
{ name: "Jane Doe", age: 28, position: "Sales Rep", salary: 50000, department:[ { name:
"Sales", employees_count: 8} ] },
{name: "Bob Smith", age: 26, position: "IT Specialist", salary: 65000, department:[ { name:
"Computer science", employees_count: 4} ] },
{ name: "Alice Johnson", age: 24, position: "HR Assistant", salary: 45000, department:[ {
name: "Electroinices", employees_count: 3} ] },

```

```
{name: "Chris Lee", age: 22, position: "Marketing Manager", salary: 60000,  
department: [{name: "Accounting", employees_count: 3}]}])
```

a. Display name of employee who has highest salary

```
db.emp.find().sort({ salary: -1 }).limit(1).pretty()
```

b. Display biggest department with max. no. of employees

```
db.emp.find({}, {"department.name": 1, "department.employees_count": 1, _id:  
0}).sort({"department.employees_count": -1}).limit(1)
```

c. Write a cursor which shows department wise employee information

```
let cursor = db.emp.aggregate([{$match: {"department.name": "Accounting" }}]) while  
(cursor.hasNext()) { printjson(cursor.next()); }
```

d. List all the employees who work in Sales dept and salary > 50000

```
db.emp.find({$and:[{ "department.name":"Sales" , salary: { $gt: 50000 } } ]})
```

## slip4

1.

```
<!doctype html>  
<html lang="en">  
  <head>  
    <meta charset="utf-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1">  
    <title>Bootstrap demo</title>  
    <link  
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css"  
rel="stylesheet" integrity="sha384-  
rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZJEdK2Kadq2F9CUG65"  
crossorigin="anonymous">  
  
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-  
awesome/4.4.0/css/font-awesome.min.css" />  
    <style >  
    </style>  
  
  </head>  
  <body>
```

```

<div class="container text center mt-5">
<h4>Students List</h4>
<table class="table" width='50%'>
<tr>
<th>ID</th>
<th>First Name</th>
<th>Last Name</th>
<th>Email</th>
</tr>

<tr><td>1</td><td>John</td><td>Berg</td><td>johntest1@gmail.com</td></tr>
<tr><td>2</td><td>Thomas</td><td>Wright</td><td>thomastest1@gmail.com</td>
</tr>
<tr><td>3</td><td>Charles</td><td>Clerk</td><td>williamtest1@gmail.com</td>
></tr>
<tr><td>4</td><td>David</td><td>Williams</td><td>davidtest1@gmail.com</td>
</tr>
<tr><td>5</td><td>Richard</td><td>Rose</td><td>test1@gmail.com</td></tr>

</table>

</div>

</body>
</html>

```

2.

```

db.hospital.insertMany(
[
{
'Hospital Name': 'City Hospital',
Location: 'Nashik',
Specializations: [ 'Pediatric', 'Gynaec' ],
Rating: 5,
Name: 'Dr. Deshmukh'
},

```



```

{
  'Hospital Name': 'ABC Hospital',
  Location: 'Nashik',
  Specializations: [ 'Pediatric', 'Gynaec', 'Orthopedic' ],
  Rating: 4,
  Name: 'Dr.Deshmukh'
},
{
  'Hospital Name': 'XY Hospital',
  Location: 'Nashik',
  Specializations: [ 'Pediatric' ],
  Rating: 5,
  Name: 'Dr. Bhadane'
},
{
  'Hospital Name': 'Ojas Hospital',
  Location: 'Nashik',
  Specializations: [ 'Pediatric' ],
  Rating: 3,
  Name: 'Dr. Mavale'
}
])

```

a. List the names of hospitals with ..... specialization.

**db.hospital.find({"Specializations": "Orthopedic"})**

b. List the Names of all hospital located in..... city

**db.hospital.find({"Location": "Nashik"})**

c. List the names of hospitals where Dr. Deshmukh visits

**db.hospital.find({ "Name": "Dr. Deshmukh" }, { "hospital Name": 1, "\_id": 0 })**

d. List the names of hospitals whose rating >=4

**db.hospital.find({ "Rating": { "\$gte": 4 } })**

## slip5

```
1. <!doctype html>

<html>
  <head>
    <style>

      table, th, td {
border: 1px solid black;
border-collapse: collapse;
text-align: center;
border-radius: 10px 12px 14px 16px;
}

    </style>
  </head>
  <body>

    <table class="head">
      <tr colspan="3"><th>List Of Person</th></tr>

      <tr><th>Srno</th><th>Person
name</th><th>age</th><th>Country</th></tr>
      <tr><td>1</td> <td></td><td> </td><td> </td></tr>
      <tr><td>2</td> <td></td><td> </td><td> </td></tr>
      <tr><td>3</td> <td></td><td> </td><td> </td></tr>
    </table>

  </body>
</html>
```

2.

db.emp.insert({ename:"Mr.Patil" ,proj:{pname:"UI",pduration:3,ptype:"Web development"}})

db.emp.insert({ename:"Mr.Kapoor" ,proj:{pname:"UI",pduration:3,ptype:"Web development"}})

a.list all name of project where proj\_type

**db.emp.find({"proj.ptype":"Web development"})**

b. b.duration greater than 3

**db.emp.find({"proj.pduration":{\$gt:2}})**

c. count no. of emp working on \_\_\_Project

**db.emp.find({ "proj.pname":"UI" }).count()**

d.list name of project on which Mr.Patil is working

**db.emp.find({"ename":"Mr.Patil"},"proj.pname":1})**

## slip7

1.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>3D Text Effect</title>
  <style>
    body {
      font-family: 'Arial', sans-serif;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
      margin: 0;
      background-color: #f0f0f0;
    }

    .threeD-text {
      font-size: 36px;
      font-weight: bold;
      color: #3498db;
      position: relative;
      transition: transform 0.7s ease-in-out;
    }

    .threeD-text:hover {
      transform: perspective(400px) rotateX(60deg);
    }
  </style>
</head>
<body>

  <div class="threeD-text">Hover over me</div>

</body>
</html>
```

2.

```
db.custinfo.insert({"first name": "Amit","last name":"Dev","DOB":"22-12;1974","Branch":"Satpur","Account": [{"Type": "saving","open date": "12-2-2021"}], "Transaction":[{"Transaction Type": "Withdraw","date": "23-2-2021","Amount": 500}], "Services":[{"type":"loan","amount":"200000","duration":"5 years"}]})
```

```
db.custinfo.insert({"first name": "Sumit","last name":"Rai","DOB":"1-6-2002","Branch":"Nashik","Account": [{"Type": "saving","open date": "20-2-2021"}], "Transaction":[{"Transaction Type": "Withdraw","date": "23-2-2021","Amount": 500},{ "Transaction Type": "deposit","date": "29-3-2021","Amount": 5000}], "Services":[{"type":"loan","amount":"150000","duration":"4 years"}]})
```

```
db.custinfo.insert({"first name": "Sakshi","last name":"Patil","DOB":"2-7-1996","Branch":"Satpur","Account": [{"Type": "current","open date": "20-12-2020"}], "Transaction":[{"Transaction Type": "Withdraw","date": "28-5-2023","Amount": 23400},{ "Transaction Type": "deposit","date": "20-8-2023","Amount": 20000}], "Services":[{"type":"Policy","amount":"10000","duration":"10 years"}]})
```

```
db.custinfo.insert({"first name": "Subhash","last name":"Dev","DOB":"22-12-1974","Branch":"Nashik","Account": [{"Type": "current","open date": "12-12-2021"}], "Transaction":[{"Transaction Type": "Withdraw","date": "2-6-2022","Amount": 500}], "Services":[{"type":"Cradit card","amount":"80000","duration":"1 years"}]})
```

a. List names of all customers whose first name starts with a "S"

```
db.custinfo.find({ "first name": /^S/ })
```

b. List all customers who has open an account on 1/1/2020 in

```
db.custinfo.find({ "Account.open date": "1-1-2020" })
```

c. List the names customers where acctype="Saving"

```
db.custinfo.find({ "Account.Type": "saving" }, { "first name": 1, "last name": 1 })
```

d. Count total no. of loan account holder of ...branch

```
db.custinfo.countDocuments({"Services.type":"loan"}, {"Branch":"Nashik"})
```

**OR**

```
db.custinfo.find({"Services.type":"loan"}, {"Branch":"Nashik"}).count()
```

# slip8

1.

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap</title>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css"
integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">

    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZJEdK2Kadq2F9CUG65"
crossorigin="anonymous">

  </head>
  <body>
    <div class="container">
      <button type="button" class="btn btn-primary">Primary</button>
      <button type="button" class="btn btn-secondary">Secondary</button>
      <button type="button" class="btn btn-success">Success</button>
      <button type="button" class="btn btn-danger">Danger</button>
      <button type="button" class="btn btn-warning">Warning</button>
      <button type="button" class="btn btn-info">Info</button>
      <button type="button" class="btn btn-error">Error</button>
    </div>
  </body>
</html>
```

2.

```
b.inven.insertMany([
{id:01,iname:"Chair",tags:["Furniture","Comfortable","Wooden"],status:"B",height:
10,warehouse:[{wname:"Skillful store",quantity:40}]},

...
{id:02,iname:"Planner",tags:["Stationary","Paper","Organizer","Task","Activities"],
status:"B",height:3,warehouse:[{wname:"Notehouse",quantity:15},{wname:"Bookstor
e",quantity:50}]},
```

...

```
{id:03,iname:"Laptop",tags:["Electronics","Technology"],status:"A",height:5,warehouse:[{wname:"Digistore",quantity:500}]},
```

...

```
{id:04,iname:"Bicycle",tags:["Travel","Transport","Luxury","Easy","Affordable","Environment friendly"],status:"A",height:14,warehouse:[{wname:"Decathlon",quantity:600}]},
```

```
... {id:05,iname:"Clock",tags:["Time","Daily use"],status:"A",height:4,warehouse:[{wname:"Clockstore",quantity:100},{wname:"Clockworks",quantity:200}]}
```

... ])

a. List all the items qty is greater than 300

```
db.inven.find({"warehouse.quantity":{"$gt:300}})
```

b. List all items which have tags less than 5

```
db.inven.find({ "tags": { $size:{ $lt:5 } } })
```

c. List all items having status equal to "B" or having quantity less than 50 and height of the product should be greater than 8

```
db.inven.find({$and:[{$or:[{"warehouse.quantity":{"$lt:50"}},{status:"B"}]},{height:{$gt:8}}])
```

d. . Find all warehouse that keeps item "Planner" and having in stock quantity less than 20

```
db.inven.find({$and:[{iname:"Planner"},{"warehouse.quantity":{"$lt:20}}])
```

# slip9

1.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport " content="width=device-width,initial-scale=1.0">
<title>HTML5 Input Types</title>
</head>
<body>
<h1>HTML 5 Input Types</h1>
<!-- Date Time Input-->
<label for="datetime">Date and Time:</label>
<input type="datetime-local" id="datetime"><br><br>
<!--Email Input-->
<label for="email">Email:</label>
<input type="email" id="email" name="email"><br><br>
<!--Submit Button-->
<label for="search">search:</label>
<input type="search" id="search" name="search"><br><br>
<!--submit Button-->
<input type="submit" value="submit">
</body>
</html>
```

```
2. db.custloaninfo.insert({"name": "Darshan","Address":"ABC","DOB":"3-2-2002","City":"Pimpri","Loan": [{"Type": "A loan","amount":200000,"premium":20000}, {"Type": " B loan","amount":500000,"premium":40000}]})
```

```
db.custloaninfo.insert({"name": "Mr.Patil","Address":"XYZ","DOB":"23-5-1999","City":"Nashik","Loan": [{"Type": "A loan","amount":90000,"premium":1000}]})
```

```
db.custloaninfo.insert({"name": "Mr.Patil","Address":"XYZ","DOB":"23-5-1999","City":"Nashik","Loan": [{"Type": "A loan","amount":90000,"premium":1000}, {"Type": "y loan","amount":2000000,"premium":25000}]})
```

```
db.custloaninfo.insert({"name": "Divya","Address":"aaa","DOB":"2-5-1998","City":"Pimpri","Loan": [{"Type": "A loan","amount":90000,"premium":1000}, {"Type": "y loan","amount":600000,"premium":2500}]})
```

a. List all customers whose name starts with 'D' character

```
db.custloaninfo.find({"name":/^D/},{ "name":1})
```

b. List the names of customer in descending order who has taken a loan from Pimpri city.

**db.custloaninfo.find({ "City": "Pimpri" }).sort({ "name": -1 })**

c. Display customer details having maximum loan amount.

**db.custloaninfo.find().sort({ "Loan.amount": -1 }).limit(1)**

d. Update the address of customer whose name is "Mr. Patil" and loan\_amt is greater than 100000.

**db.custloaninfo.updateMany( { "name": "Mr.Patil", "Loan.amount": { \$gt: 100000 } }, { \$set: { "Address": "New Address" } } )**

## slip10

```
1. <!DOCTYPE html>

<html>
<head>
  <title>CSS Transition</title>

  <style>
    h1 {
      color: #640bd1;
      text-align: center;
    }

    div.one {
      height: 150px;
      width: 150px;
      border: 1px dashed rgb(218, 66, 66);
      margin: 0 auto;
      background: #474eda;
      transition-delay: 2s;
    }

    div.one:hover {
      height: 300px;
      width: 300px;
      background:red;
    }
  </style>
</head>
```



```

<body>
  <h1>Transition</h1>

  <div class="one">
  </div>

</body>
</html>

```

2.

```
db.custonlineshope.insert({name:"A",city:"Nashik",billamt:"60000",Brand:[ { name:"X"
,pname:"AC",warranty:"10 years",dateofPurchase:"15/8/2023",rating:5}]})
```

```
db.custonlineshope.insert({ name: "B", city: "Pune", billamt: "40000", Brand: [ { name: "y",
pname: "TV", warranty: "5 years", dateofPurchase: "1/10/2023", rating: 4 } ] } )
```

```
db.custonlineshope.insert({ name: "C", city: "Pune", billamt: "55000", Brand: [ { name: "H",
pname: "Washing machine", warranty: "8 years", dateofPurchase: "8/1/2023", rating: 3 } ] } )
```

```
db.custonlineshope.insert({ name: "D", city: "Mumbai", billamt: "50000", Brand: [ { name:
"A", pname: "Samart Phone", warranty: "1 years", dateofPurchase: "8/1/2023", rating: 5 } ] } )
```

a. List the names of product whose warranty period is one year

```
db.custonlineshope.find({"Brand.warranty":"1 years"})
```

b. List the customers has done purchase on "15/08/2023".

```
db.custonlineshope.find({"Brand.dateofPurchase":"15/8/2023"})
```

c. Display the names of products with brand which have highest rating.

```
db.custonlineshope.find({},{"Brand.pname":1,"Brand.rating":1}).sort({"Brand.rating"
:-1}).limit(1)
```

d. Display customers who stay in city and billamt >50000.

```
db.custonlineshope.find({$and:[{city:"Nashik"},{billamt:{$gt:50000}}]}) OR
db.custonlineshope.find({ "city": "Nashik", "billamt": { $gt: 50000 } })
```

# slip14

1.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Travel Plan Booking Form</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f4f4f4;
      margin: 20px;
    }

    form {
      max-width: 500px;
      margin: 0 auto;
      background-color: #fff;
      padding: 20px;
      border-radius: 5px;
      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    }

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

    input[type="text"],
    input[type="tel"],
    select {
      width: 100%;
      padding: 8px;
      margin-bottom: 10px;
      box-sizing: border-box;
    }

    input[type="checkbox"] {
      margin-bottom: 10px;
    }

    input[type="submit"],
    input[type="reset"] {
      background-color: #4caf50;
      color: #fff;
      padding: 10px 15px;
```

```

        border: none;
        border-radius: 5px;
        cursor: pointer;
        font-size: 16px;
    }

    input[type="submit"]:hover,
    input[type="reset"]:hover {
        background-color: #45a049;
    }
</style>
</head>
<body>

<form action="#" method="post">
    <label for="name">Name:</label>
    <input type="text" id="name" name="name" required>

    <label for="address">Address:</label>
    <input type="text" id="address" name="address" required>

    <label for="contact">Contact Number:</label>
    <input type="tel" id="contact" name="contact" pattern="[0-9]{10}"
required>

    <label>Gender:</label>
    <label for="male"><input type="radio" id="male" name="gender" value="male"
required> Male</label>
    <label for="female"><input type="radio" id="female" name="gender"
value="female" required> Female</label>

    <label>Preferred Season:</label>
    <label for="spring"><input type="checkbox" id="spring" name="season[]"
value="spring"> Spring</label>
    <label for="summer"><input type="checkbox" id="summer" name="season[]"
value="summer"> Summer</label>
    <label for="autumn"><input type="checkbox" id="autumn" name="season[]"
value="autumn"> Autumn</label>
    <label for="winter"><input type="checkbox" id="winter" name="season[]"
value="winter"> Winter</label>

    <label for="location">Location Type:</label>
    <select id="location" name="location" required>
        <option value="" disabled selected>Select Location Type</option>
        <option value="beach">Beach</option>
        <option value="mountain">Mountain</option>
        <option value="city">City</option>
        <option value="countryside">Countryside</option>

```

```

</select>

<input type="submit" value="Submit">
<input type="reset" value="Reset">
</form>

</body>
</html>

```

2. CREATE(:Government {name: 'Government'})

CREATE (:Scholarship {name: 'S1', category: 'OBC', year: '2020-2021', income\_limit: 50000})

CREATE (:Scholarship {name: 'S2', category: 'General', year: '2020-2021', income\_limit: 60000})

CREATE (:Student {name: 'C1', category: 'OBC', year\_of\_study: '2020-2021'})

CREATE (:Student {name: 'C2', category: 'General', year\_of\_study: '2020-2021'})

CREATE (:FamilyMember {name: 'F1'})

relationships:

1)MATCH(g:Government)

MATCH(s:Scholarship{name:'S1'})

CREATE(g)-[:PROVIDES]->(s)

2)MATCH(g:Government)

MATCH(s:Scholarship{name:'S2'})

CREATE(g)-[:PROVIDES]->(s)

3)MATCH(c:Student{name:'C1'})

MATCH(s:Scholarship{name:'S1'})

CREATE(c)-[:Can\_Apply]->(s)

4)MATCH(c:Student{name:'C2'})

MATCH(s:Scholarship{name:'S2'})

CREATE(c)-[:Can\_Apply]->(s)

5)MATCH(c:Student{name:'C1'})

MATCH(s:Scholarship{name:'S1'})

CREATE(c)-[:get\_benifit]->(s)

6)MATCH(c:Student{name:'C2'})

MATCH(s:Scholarship{name:'S2'})

CREATE(c)-[:get\_benifit]->(s)

7)MATCH(c:Student{name:'C1'})

MATCH(f:FamilyMember)

CREATE(c)-[:recommended]->(f)

QUERIES:

1)MATCH(s:Scholarship{category:'OBC'})

RETURN s.name

2)MATCH (c:Student)-[:get\_benifit]->(s:Scholarship {name: 'S1', year: '2020-2021'})

RETURN COUNT(c) AS numberOfStudents

3)MATCH (s:Scholarship {name: 'S1'})

SET s.income\_limit = 55000

RETURN s

and

MATCH (s:Scholarship {name: 'S2'})

SET s.income\_limit = 75000

RETURN s

4)MATCH (s:Scholarship)

OPTIONAL MATCH (s:Student)-[:get\_benifit]->(s)

WITH s, COUNT(s) AS numberOfStudents

ORDER BY numberOfStudents DESC

RETURN s.name, numberOfStudents

LIMIT 1

## Slip15

1.

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <!--<meta name="viewport" content="width=device-width, initial-
scale=1">-->
    <title>Bootstrap demo</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLWZJEdK2Kadq2F9CUG65"
crossorigin="anonymous">

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-
awesome/4.4.0/css/font-awesome.min.css" />
    <style >
    .container {
background-color:#879191
    }

  </style>

</head>
<body>
  <div class="container text center">
    <div class="col-lg-10">
      <h3 align="center" class="mt-5 mb-5 pt-2">Registration
Form</h3><!--p=padding t=top b=bottom m=margin-->
      <form class="was-validatedxx" action="/action_page.php" ><!--
validation -->

        <div class="input-group">
          <label class="form-label col-lg-2" for="firstName">
            First Name &nbsp;&nbsp;&nbsp;<!--non breaking space-->
          </label>
          <span class="input-group-text"><i class="fa fa-
user"></i> </span>

          <input
            id="firstName"
            type="text"
            class="form-control"
            placeholder="Firstname"
          />

        </div>
```

```
<div class="clear-fix">&nbsp;</div><!--cerate space  
between two text box-->  
  
    <div class="input-group">  
        <label class="form-label col-lg-2" for="lastName">  
            Last Name &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
</label>  
        <span class="input-group-text"><i class="fa fa-user"></i> </span>  
  
        <input  
            id="lastName"  
            type="text"  
            class="form-control"  
            placeholder="Last Name"  
        />  
    </div>  
    <div class="clear-fix">&nbsp;</div>  
    <div class="input-group">  
        <label class="form-label col-lg-2" for="dept">  
            Department/Office &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
</label>  
        <span class="input-group-text"><i class="fa fa-list"></i> </span>  
  
        <select class="form-control" id="dept">  
            <option value="">Select your department </option>  
            <option value="arts">Arts </option>  
            <option value="commerce">Commerce </option>  
            <option value="science">science </option>  
  
        </select>  
    </div>  
    <div class="clear-fix">&nbsp;</div>  
    <div class="input-group">  
        <label class="form-label col-lg-2" for="username">  
            Username &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
</label>  
        <span class="input-group-text"><i class="fa fa-user"></i> </span>  
  
        <input  
            id="username"  
            type="text"  
            class="form-control"  
            placeholder="User Name"  
        />  
  
        </select>  
    </div>
```

```
<div class="clear-fix">&nbsp;</div>  
<div class="input-group">  
  <label class="form-label col-lg-2" for="password">  
    Password &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
  </label>  
  <span class="input-group-text"><i class="fa fa-user"></i> </span>  
  
  <input  
    id="password"  
    type="password"  
    class="form-control"  
    placeholder="Password"  
  
  />  
  
</div>  
<div class="clear-fix">&nbsp;</div>  
  <div class="input-group">  
    <label class="form-label col-lg-2" for="cpassword">  
      Confirm Password &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
    </label>  
    <span class="input-group-text"><i class="fa fa-user"></i> </span>  
  
    <input  
      id="cpassword"  
      type="password"  
      class="form-control"  
      placeholder="Confirm Password"  
  
    />  
  
  </div>  
  
<div class="clear-fix">&nbsp;</div>  
  <div class="input-group">  
    <label class="form-label col-lg-2 text-right" for="email">  
      Email &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
    </label>  
    <span class="input-group-text"><i class="fa fa-envelope"></i> </span>  
  
    <input  
      id="email"  
      type="email"  
      class="form-control"  
      placeholder="Email"  
      aria-label="email"  
  
    />  
  
  </div>
```



[illegible]

```

        /*// Example starter JavaScript for disabling form submissions if
there are invalid fields
        (() => {
            'use strict';

            // Fetch all the forms we want to apply custom Bootstrap
validation styles to
            const forms = document.querySelectorAll('.needs-validation');

            // Loop over them and prevent submission
            Array.prototype.slice.call(forms).forEach((form) => {
                form.addEventListener('submit', (event) => {
                    if (!form.checkValidity()) {
                        event.preventDefault();
                        event.stopPropagation();
                    }
                    form.classList.add('was-validated');
                }, false);
            });
        })();
    */
</script>
</body>
</html>

```

2.

```

CREATE (:Movie {title: "M1"})
CREATE (:Movie {title: "M2"})
CREATE (:Actor {name: "A1"})
CREATE (:Actor {name: "A2"})
CREATE (:Actor {name: "Shahrukh Khan"})
CREATE (:Business {revenue: 1000000})
CREATE (:Business {revenue: 1500000})
CREATE (:Award {name: "Best Film"})
CREATE (:Award {name: "Best Actor"})
CREATE (:Award {name: "Best Director"})

MATCH(a:Actor{name:"A1"})

```

MATCH(m:Movie{title:"M1"})

CREATE(a)-[:ACTED\_IN]->(m)

MATCH(a:Actor{name:"A2"})

MATCH(m:Movie{title:"M1"})

CREATE(a)-[:ACTED\_IN]->(m)

MATCH(a:Actor{name:"Shahrukh Khan"})

MATCH(m:Movie{title:"M2"})

CREATE(a)-[:ACTED\_IN]->(m)

MATCH(m:Movie{title:"M1"})

MATCH(b:Business{revenue: 1000000})

CREATE(m)-[:MADE\_BUSINESS]->(b)

MATCH(m:Movie{title:"M2"})

MATCH(b:Business{revenue: 1500000})

CREATE(m)-[:MADE\_BUSINESS]->(b)

MATCH(m:Movie{title:"M2"})

MATCH(a:Award{name: "Best Film"})

CREATE(m)-[:RECEIVED\_AWARD]->(a)

MATCH(m:Movie{title:"M1"})

MATCH(a:Award{name: "Best Actor" })

CREATE(m)-[:RECEIVED\_AWARD]->(a)

Bu

MATCH(m:Movie{title:"M2"})

MATCH(a:Award{name:"Best Director" })

CREATE(m)-[:RECEIVED\_AWARD]->(a)

QUERIES:

1)MATCH (m:Movie)-[mb:MADE\_BUSINESS]->(b:Business)

RETURN m, mb, b

ORDER BY b.revenue DESC

LIMIT 1;

2)MATCH (b:Actor)-[a:ACTED\_IN]->(m:Movie)

RETURN b,a,m;

3)MATCH (a:Actor {name: "Shahrukh Khan"})-[:ACTED\_IN]->(m:Movie)

RETURN m.title;

4)MATCH (m:Movie)-[ra:RECEIVED\_AWARD]->()

WITH m, COUNT(ra) AS awardsCount

WHERE awardsCount > 1

RETURN m, awardsCount;

## Slip17

```
1. <!doctype html>

<head>
<style>
    .outer-div
    {
        margin-left:200px;
        padding:100px;
        width:600px;
        height:400px;
        background-color:orange;
        border:4px solid;
    }
    .container
    {
        background-color:yellow;
        width:400px;
        height:200px;
        border:4px solid black;
        padding:1px;
        padding-right:100px;
        padding-left:100px;
        padding-top:100px;
        padding-bottom:100px;
        margin:0 auto;
    }
</style>
</head>
</body>
<div class="outer-div">
<div class="container">
<h4 style="background-color:yellow;border:2px solid;padding:20px;text-align:center">M.Sc(comp.sci)</h4>
<h4 style="background-color:yellow;border:2px solid;padding:20px;text-align:center">Academic year 2023-2024</h4>
</div>
</div>
</body>
</html>
```

**2.**

CREATE (:Author {name: "A1"})

CREATE (:Author {name: "A2"})

CREATE (:Book {title: "Novel1"})

CREATE (:Book {title: "Novel2"})

CREATE (:Book {title: "Comics"})

CREATE (:Publisher {name: "Sage"})

CREATE (:Publisher {name: 'New Age'})

CREATE (:Reader {name: "R1"})

CREATE (:Reader {name: "R2"})

MATCH (a:Author{name:'A2'})

MATCH (b:Book{title:'Novel1'})

CREATE (a)-[:WROTE]->(b)

MATCH (a:Author{name:'A2'})

MATCH (b:Book{title:'Novel2'})

CREATE (a)-[:WROTE]->(b)

MATCH (a:Author{name:'A2'})

MATCH (b:Book{title:'Comics'})

CREATE (a)-[:WROTE]->(b)

MATCH (p:Publisher{name:'New Age'})

MATCH (b:Book{title:'Novel1'})

CREATE (a)-[:PUBLISHED]->(b)

MATCH (p:Publisher{name:'Sage'})

MATCH (b:Book{title:'Comics'})

CREATE (a)-[:PUBLISHED]->(b)

MATCH (p:Publisher{name:'Sage'})

MATCH (b:Book{title:'Comics'})

CREATE (a)-[:PUBLISHED]->(b)

MATCH (r:Reader{name:'R1'})

MATCH (b:Book{title:'Comics'})

CREATE (a)-[:READ]->(b)

MATCH (r:Reader{name:'R2'})

MATCH (b:Book{title:'Novel1'})

CREATE (a)-[:READ]->(b)

MATCH (r:Reader{name:'R1'})

MATCH (b:Book{title:'Comics'})

CREATE (r)-[:REVIEWED {rating: 4}]->(b);

MATCH (r:Reader{name:'R2'})

MATCH (b:Book{title:'Novel1'})

CREATE (r)-[:REVIEWED {rating:5}]->(b);

1)MATCH (a:Author)-[:WROTE]->(b:Book {title: 'Comics'})

RETURN a.name; done

2)MATCH (:Book {title: 'Comics'})<-[:READ]-(r:Reader) -[:READ]->(:Book)<-[:PUBLISHED]-(p:Publisher {name: 'Sage'})

RETURN COUNT(DISTINCT r) AS numberOfReader; done

3)MATCH (p:Publisher)

WHERE p.name STARTS WITH 'N'

RETURN DISTINCT p.name;

4)MATCH (ra:Reader)-[r:REVIEWED]->(b:Book {title: 'Comics'})

WHERE r.rating >=3

RETURN ra.name;



## Slip24

1.

```
<!doctype html>
<html>
  <body>
    <h1>Input Type Example</h1>
    <label for="date">Date:</label>
    <input type="date"><br>
    <label for="datetime">DateTime:</label>
    <input type="datetime"><br>
    <label for="datetime-local">Datetime-local</label>
    <input type="datetime-local"><br>
    <label for="month">Month:</label>
    <input type="month"><br>
    <label for="time">Time:</label>
    <input type="Time"><br>
    <label for="week">Week:</label>
    <input type="week"><br>
    <button>Submit</button>

  </body>
</html>
```

2.

```
CREATE (:Book {title: 'Book1'})
      CREATE (:Book {title: 'Book2'})
      CREATE (:Student {name: 'S1'})
      CREATE (:Student {name: 'S2'})
      CREATE (:BookType {name: 'Text'})
      CREATE (:BookType {name: 'Reference'})
      CREATE (:BookType {name: 'Bibliography'})

MATCH(s:Student{name:'S1'})
MATCH(b:Book)
CREATE(s)-[:BOUGHT]->(b)
MATCH(s:Student{name:'S2'})
```

```

MATCH(b:Book)
CREATE(s)-[:BOUGHT]->(b)
MATCH(s:Student{name:'S1'})
MATCH(b:Book{title:'Book1'})
CREATE(s)-[:RECOMMENDED]->(b)
MATCH(s:Student{name:'S2'})
MATCH(b:Book{title:'Book1'})
CREATE(s)-[:RECOMMENDED]->(b)
MATCH(b:Book{title:'Book1'})
MATCH(bt:BookType {name: 'Text'})
CREATE(b)-[:TYPE_OF]->(bt)
MATCH(b:Book{title:'Book2'})
MATCH(bt:BookType {name: 'Reference'})
CREATE(b)-[:TYPE_OF]->(bt)
MATCH(b:Book{title:'Book2'})
MATCH(bt:BookType {name: 'Bibliography'})
CREATE(b)-[:TYPE_OF]->(bt)

```

1.List the books of type “text” [3]

```
=>MATCH (b:Book)-[:TYPE_OF]->(bt:BookType {name:'Text'})
```

```
RETURN b.title
```

```
Ans="Book1"
```

2.List the name of student who bought a text and reference types books.[3]

```
MATCH (s:Student)-[:BOUGHT]->(b:Book)-[:TYPE_OF]->(t:BookType)
```

```
WHERE t.name IN ['Text', 'Reference']
```

```
RETURN DISTINCT s.name
```

```
Ans="S2"
```

```
"S1"
```

3.List the most recommended book type.[4]

```
=>MATCH (b:Book)<-[:RECOMMENDED]-(s:Student)
```

```
WITH b, COUNT(s) AS recommendations
```

```
MATCH (book)-[:TYPE_OF]->(t:BookType)
```

RETURN t.name, SUM(recommendations) AS totalRecommendations

ORDER BY totalRecommendations DESC

LIMIT 1

Ans="Text"      2

4.List the student who buy the more than one type of book [4]

=>MATCH (s:Student)-[:BOUGHT]->(b:Book)-[:TYPE\_OF]->(t:BookType)

WITH s, COUNT(DISTINCT t) AS numTypes

WHERE numTypes > 1

RETURN s.name

## Slip25

```
1. <!doctype html>

<html>
  <head>
    <title>Entry form</title>
    <style>
      .container{
        max-width:600px;
        margin:0 auto;
        padding:20px;
        background-color:skyblue;
        border-radius:5px;

      }
      h1{
        font-size: 24pt;

        text-align: center;
      }
    </style>
  </head>
  <body>
    <div class="container">
      <h1>Entry form</h1>
    </div>
  </body>
</html>
```

```

    </style>
</head>
<body>

    <div class="container">
        <form>
            <h1>Entry Form </h1>
            <label for="name">Enter your name </label>
            <input id="pname" name="pname" ><br>
            <label for="age">Age</label>
            <input type="integer" name="age"><br>
            <label for="Address">Address</label>
            <input type="textarea"><br>
            <label for="gender">Sex</label><br>
            <input type="radio" value="Male">Male</input><br>
            <input type="radio" value="Female">Female</input><br>
            <label for="Nationality"> Nationality </label>
            <select id="item" name="Nationality">
                <option value="Indian">Indian</option>
                <option value="American">American</option>
                <option value="Other">Other</option>

            </select><br>
            <label for="Language">language Known</label><br>
            <input type="checkbox" value="c">C</input><br>
            <input type="checkbox" value="c++">C++</input><br>
            <input type="checkbox" value="c#">C#</input><br>
            <input type="checkbox" value="Java">Java</input><br>
            <input type="checkbox" value="Other">Other</input><br>
            <label for="Password">Enter Your Password</label>
            <input type="text" name="Password"><br>
            <button type="button" class="button success">Submit</button>
            <button type="button" class="button primary">Reset</button>
        </form>
    </div>
</body>
</html>

```

**2. CREATE (:Department {name: 'Physics'})**

CREATE (:Department {name: 'Geography'})

CREATE (:Department {name: 'Computer'})

//Create Course

CREATE (:Course {name: 'Physics Course 1'})

CREATE (:Course {name: 'Physics Course 2'})

```

CREATE (:Course {name: 'Geography Course 1'})
CREATE (:Course {name: 'Computer Course 1'})
CREATE (:Course {name: 'Computer Course 2'})
//Relationships:
MATCH(d:Department{name:'Physics'})
MATCH(c:Course{name:'Physics Course 1'})
CREATE(d)-[:CONDUCTS]->(c)
MATCH(d:Department{name:'Physics'})
MATCH(c:Course{name:'Physics Course 2'})
CREATE(d)-[:CONDUCTS]->(c)
MATCH(d:Department{name:'Physics'})
MATCH(c:Course{name:'Computer Course 2'})
CREATE(d)-[:CONDUCTS]->(c)
MATCH(d:Department{name:'Geography'})
MATCH(c:Course{name:'Geography Course 1'})
CREATE(d)-[:CONDUCTS]->(c)

MATCH(d:Department{name:'Computer'})
MATCH(c:Course{name:'Computer Course 1'})
CREATE(d)-[:CONDUCTS]->(c)
MATCH(d:Department{name:'Computer'})
MATCH(c:Course{name:'Computer Course 2'})
CREATE(d)-[:CONDUCTS]->(c)
MATCH(p:Person{name:'P1'})
MATCH(c:Course{name: 'Physics Course 1'})
CREATE(p)-[:RECOMMENDS]->(c)
MATCH(p:Person{name:'P2'})
MATCH(c:Course{name: 'Physics Course 2'})
CREATE(p)-[:RECOMMENDS]->(c)
MATCH(p:Person{name:'P2'})
MATCH(c:Course{name: 'Computer Course 1'})

```

```

CREATE(p)-[:RECOMMENDS]->(c)
MATCH(p:Person{name:'P1'})
MATCH(c:Course{name: 'Geography Course 1'})
CREATE(p)-[:RECOMMENDS]->(c)
MATCH(p:Person{name:'P2'})
MATCH(c:Course{name: 'Geography Course 1'})
CREATE(p)-[:RECOMMENDS]->(c)
MATCH(p:Person{name:'P3'})
MATCH(c:Course{name: 'Geography Course 1'})
CREATE(p)-[:RECOMMENDS]->(c)

```

1.List the details of all the departments in the university. [3]

```
=>MATCH (d:Department)
```

```
RETURN d;
```

2.List the names of the courses provided by Physics department. [3]

```
=>MATCH (d:Department {name: 'Physics'})-[:CONDUCTS]->(c:Course)
```

```
RETURN c.name;
```

```
ANS="Physics Course 2"
```

```
"Physics Course 1"
```

3.List the most recommended course in Geography department. [4]

```
=>MATCH (d:Department {name: 'Geography'})-[:CONDUCTS]->(c:Course)<-[:RECOMMENDS]-
(p:Person)
```

```
RETURN c.name, COUNT(p) AS recommendations
```

```
ORDER BY recommendations DESC
```

```
LIMIT 1;
```

```
ANS="Geography Course 1"      3
```

4.d. List the names of common courses across Mathematics and computer department.[4]

```
=>MATCH (p:Department {name: 'Physics'})-[:CONDUCTS]->(pCourse:Course)
```

```
MATCH (c:Department {name: 'Computer'})-[:CONDUCTS]->(cCourse:Course)
```

```
WHERE pCourse.name = cCourse.name
```

```
RETURN pCourse.name;
```

ANS="Computer Course 2"