JNAN VIKAS MANDAL'S

PADMASHREE DR. R.T.DOSHI DEGREE COLLEGE OF INFORMATION TECHNOLOGY

OHANLAL RAICHAND MEHTA COLLEGE OF COMMERCE

DIWALIMAA DEGREE COLLEGE OF SCIENCE

CERTIFICATE

This is to certify that the **Mr. Prathamesh Gurav** of S.Y.B.Sc.CS Semester IV has completed the practical work in the subject of **Advanced Application Development** during the Academic year 2023-24 under the guidance of **PROF**.**Vinaya Mangnale** being the partial requirement for the fulfillment of the curriculum of Degree of Bachelor of Science in Computer Science, University of Mumbai.

Place: **Date:**

Sign of Subject In Charge

Sign of External Examiner

Sign of In charge / H.O.D

Index

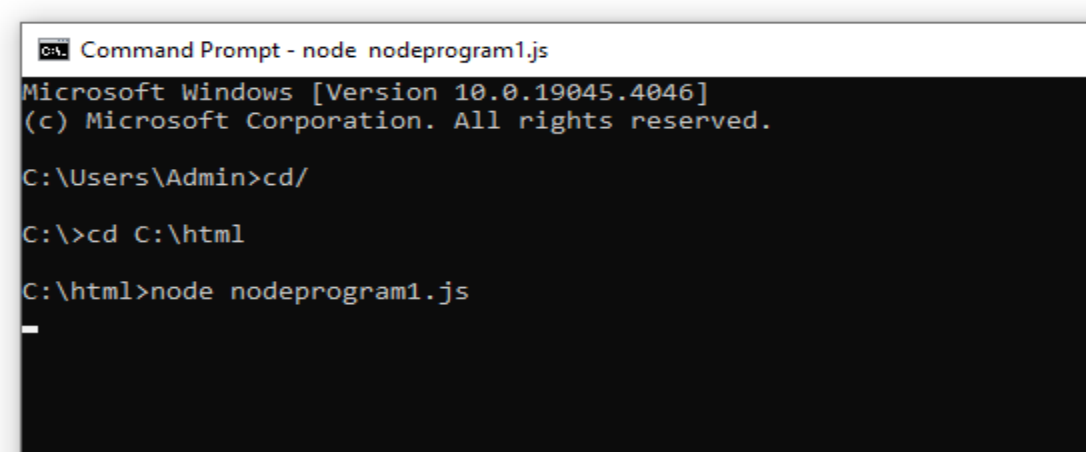
Sr. No.	Name of Practicals	Date	Signature
1	Write a simple program to print any message using node.js.		
2	Write a program to implement CRUD operations in MongoDB.		
3	Write a program to implement MongoDB data models.		
4	Create a simple html project using AngularJS Framework and apply ng-model ,ng-controller and expressions.		
5	Write a program to create simple web application using AngularJS.		
6	Write a program to perform validation of a form using AngularJS.		
7	Create an application for Customer/Students records using AngularJS.		
8	Write a program to implement Error Handling in AngularJS.		

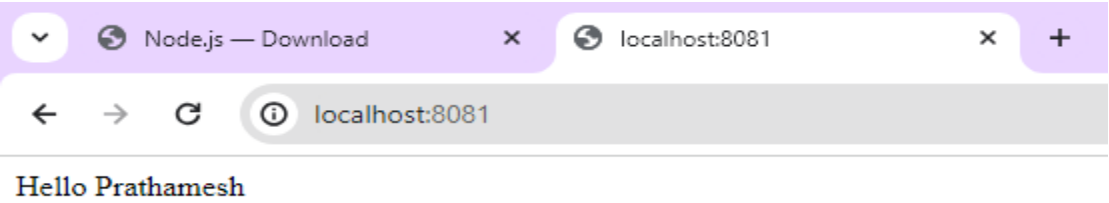
Practical 1

Write a simple program to print any message using node.js.

```
var http=require('http');

http.createServer(function(req,res){

res.writeHead(200,{'Content-

Type':'text/html'}); res.write('Hello

Prathamesh');

res.end();
}).listen(8081);
```

Output:-





Practical 2

Write a program to implement CRUD operations in MongoDB.

1.use Prathamesh //database name

2.Insert documents inside collection i.e. Creating Records.

db.prathamesh.insertOne({ename:"prathamesh", empid:1,esal:50000})

db.prathamesh.insertOne({ename:"aditya", empid:2,esal:30000})

db.prathamesh.insertOne({ename:"harshad", empid:3,esal:40000})

db.prathamesh.insertOne({ename:"arya", empid:4,esal:25000})

Output:-

```
>_MONGOSH

> use prathamesh
< switched to db prathamesh
> db.prathamesh.insertOne({ename:"prathamesh", empid:1,esal:50000})
  db.prathamesh.insertOne({ename:"aditya", empid:2,esal:30000})
  db.prathamesh.insertOne({ename:"harshad", empid:3,esal:40000})
  db.prathamesh.insertOne({ename:"arya", empid:4,esal:25000})
< {
    acknowledged: true,
    insertedId: ObjectId('65e19fc1a9261de7f24a157a')
  }
```

3. Retrieving documents from the collection.

db.prathamesh.find().pretty()

Output:-

```
>_MONGOSH
> db.prathamesh.find().pretty()
< {
    _id: ObjectId('65e19fc0a9261de7f24a1577'),
    ename: 'prathamesh',
    empid: 1,
    esal: 50000
  }
  {
    _id: ObjectId('65e19fc0a9261de7f24a1578'),
    ename: 'aditya',
    empid: 2,
    esal: 30000
  }
  {
    _id: ObjectId('65e19fc0a9261de7f24a1579'),
    ename: 'harshad',
    empid: 3,
    esal: 40000
  }
  {
    _id: ObjectId('65e19fc1a9261de7f24a157a'),
    ename: 'arya',
```

4.Updating documents in the collection.

db.prathamesh.updateOne({ename:"arya"},{$set:{esal:30000}})

Output:-

```
> db.prathamesh.updateOne({ename:"arya"},{$set:{esal:30000}})
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
> db.prathamesh.findOne({ename:"arya"})
< {
    _id: ObjectId('65e19fc1a9261de7f24a157a'),
    ename: 'arya',
    empid: 4,
    esal: 30000
  }
```

5. Deleting documents from the collections.

db.prathamesh.deleteOne({ename:"arya"})
db.prathamesh.find().pretty()

Output:-

```
>_MONGOSH
    deletedCount: 1
  }
> db.prathamesh.find().pretty()
< {
    _id: ObjectId('65e19fc0a9261de7f24a1577'),
    ename: 'prathamesh',
    empid: 1,
    esal: 50000
  }
  {
    _id: ObjectId('65e19fc0a9261de7f24a1578'),
    ename: 'aditya',
    empid: 2,
    esal: 30000
  }
  {
    _id: ObjectId('65e19fc0a9261de7f24a1579'),
    ename: 'harshad',
    empid: 3,
    esal: 40000
  }
prathamesh>
```

Practical 3

Write a program to implement MongoDB data models.

```
db.prathamesh.insertOne({ename:"abc", empid:1,esal:2000});
db.prathamesh.insertOne({ename:"xyz", empid:2,esal:3000})

db.prathamesh.find().pretty()
```

Output:-

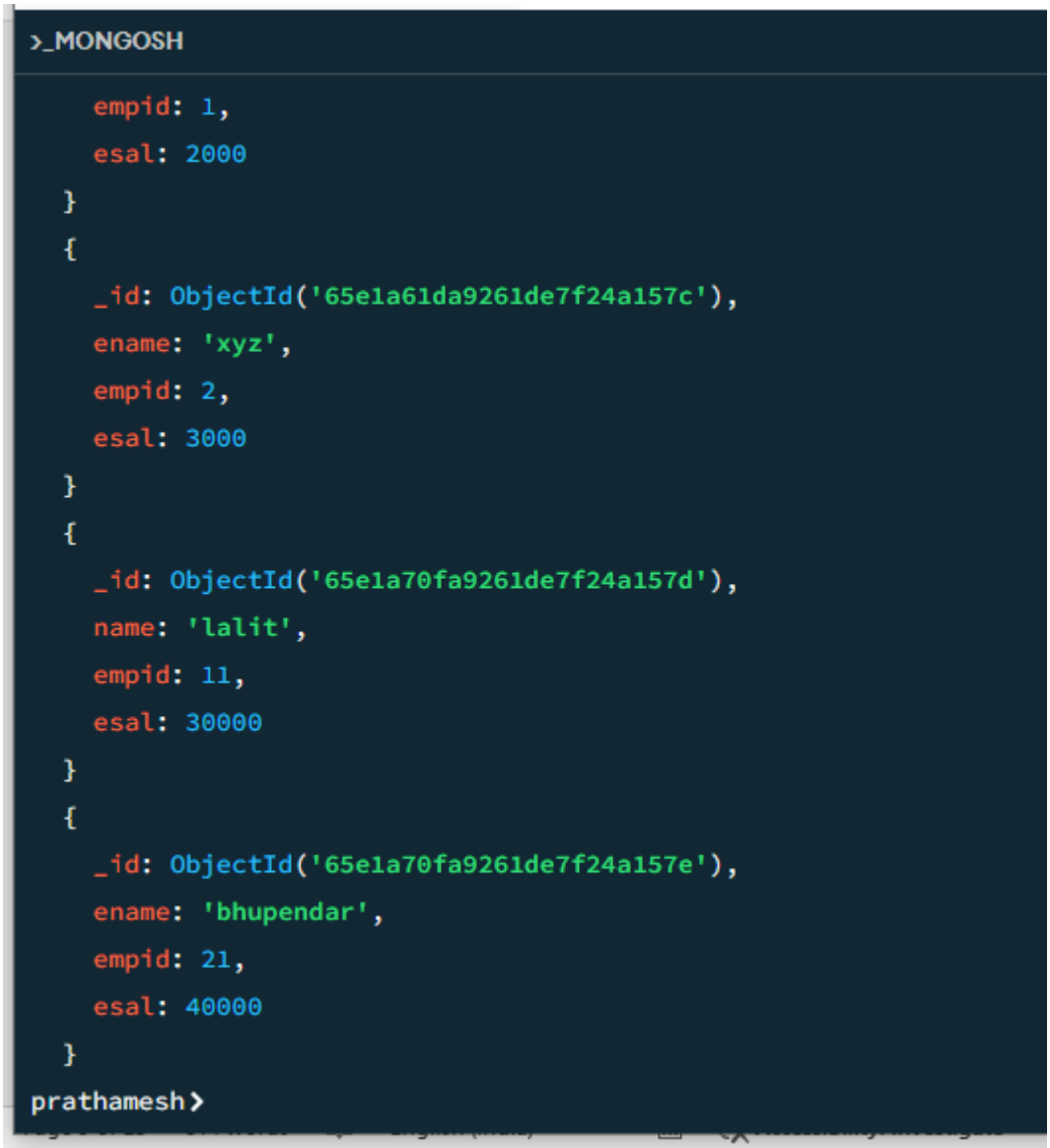
```
>_MONGOSH
    empid: 2,
    esal: 30000
  }
  {
    _id: ObjectId('65e19fc0a9261de7f24a1579'),
    ename: 'harshad',
    empid: 3,
    esal: 40000
  }
  {
    _id: ObjectId('65e1a61da9261de7f24a157b'),
    ename: 'abc',
    empid: 1,
    esal: 2000
  }
  {
    _id: ObjectId('65e1a61da9261de7f24a157c'),
    ename: 'xyz',
    empid: 2,
    esal: 3000
  }
prathamesh>
```

Embedded Models.

db.prathamesh.insertMany([{name:"lalit",empid:11,esal:30000},{ename:"bhupendar",empid:21,esal:40000}])

db.prathamesh.find().pretty()

Output:-



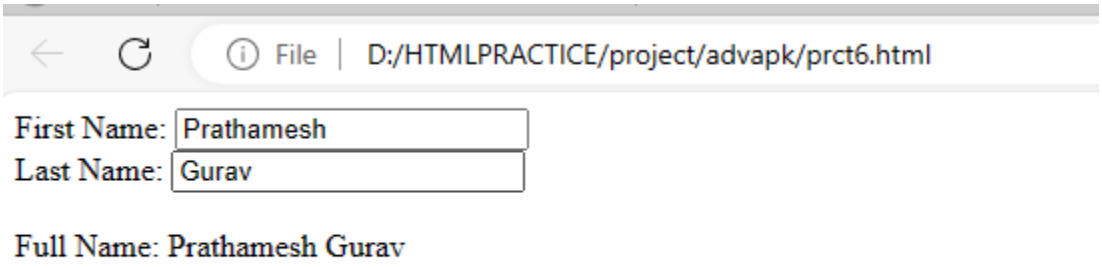
Practical 4

Create a simple html project using AngularJS Framework andapply ng-model ,ng-controller and expressions.

```
<!DOCTYPE html>

<html>

<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script
>

<body>

<div ng-app="myApp" ng-controller="personCtrl">

First Name: <input type="text" ng-model="firstName"><br>

Last Name: <input type="text" ng-model="lastName"><br>

<br>

Full Name: {{fullName()}}

</div>

<script>

var app = angular.module('myApp', []);

app.controller('personCtrl', function($scope) {

    $scope.firstName = "Prathamesh";

    $scope.lastName = "Gurav";

    $scope.fullName = function() {

        return $scope.firstName + " " + $scope.lastName;

    };

});
```

```
</script>

</body>

</html>
```

Output:-

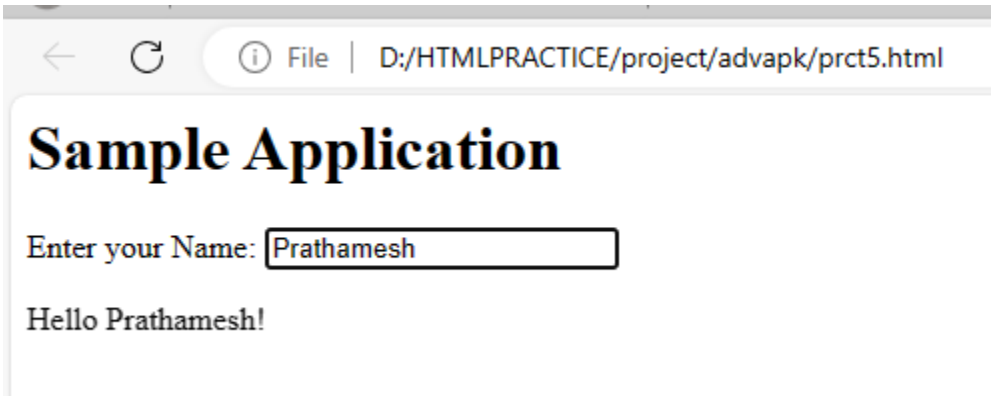


Practical 5

Write a program to create simple web application usingAngularJS.

```
<html>
  <head>
    <title>AngularJS First Application</title>
  </head>
        <body>
           <h1>Sample Application</h1>
              <div ng-app = "">
                 <p>Enter your Name: <input type = "text" ng-model =
          "name"></p>
                 <p>Hello <span ng-bind = "name"></span>!</p>
              </div>
              <script src =
          "https://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.mi
          n.js">
              </script
        </body>
</html>
```

Output:-



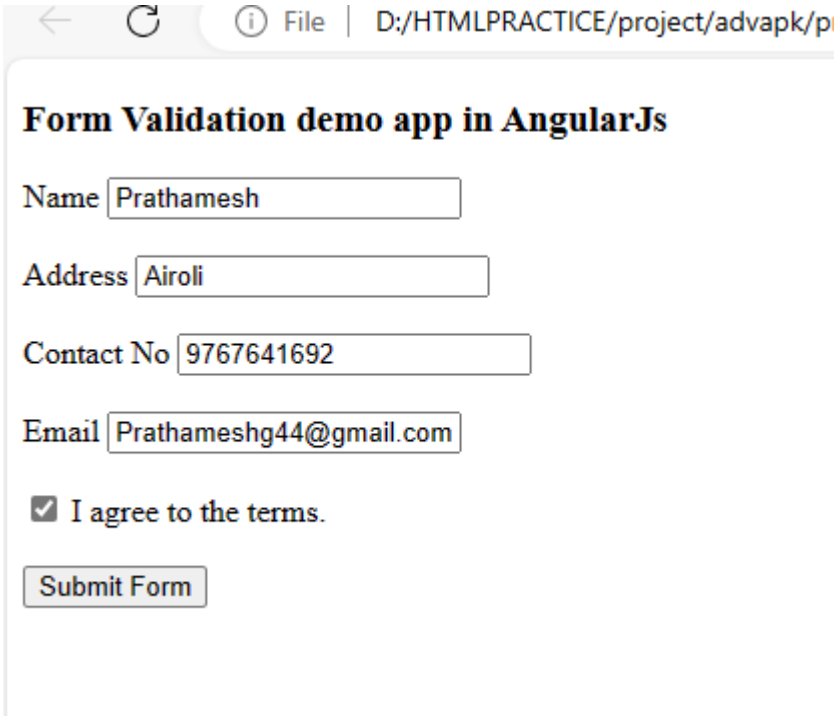
Practical 6

Write a program to perform validation of a form usingAngularJS.

```
<!DOCTYPE html>
<html>
<head>
<title>AngularJS Form Validation</title>
<script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.
min.js"></script>
<script>
var app=angular.module('formApp',[]); app.controller('formCtrl',
function($scope) {
$scope.sendForm=function(){ window.open("welcome.html");
$scope.msg='Form Submitted Successfully';
};
$scope.getClass=function(color){ return color.toString();
}
});
</script>
</head>
<body ng-app="formApp" ng-controller="formCtrl">
<h3>Form Validation demo app in AngularJs</h3>
<form name="personForm" ng-submit="sendForm()">
<label for="name">Name</label>
<input id="name" name="name" type="text" ng-
model="person.name" required/>
<span class="error" ng-
show="personForm.name.$error.required">Required!</span>
<br><br>
<label for="address">Address</label>
<input id="address" name="address" type="text" ng-
model="person.address" required/>
```

```
        <span class="error" ng-
        show="personForm.address.$error.required">Required!</span>
        <br><br>
        <label for="contact">Contact No</label>

        <input id="mobile" name="mobile" type="number" ng-
        model="person.mobile" required/>
        <span class="error" ng-
        show="personForm.mobile.$error.required">Required!</span>
        <span class="error" ng-
        show="personForm.mobile.$error.mobile">Invalid
        Mobile</span>
        <br><br>
        <label for="email">Email</label>
        <input id="email" name="email" type="email" ng-
        model="person.email" required/>
        <span class="error" ng-
        show="personForm.email.$error.required">Required!</span>
        <span class="error" ng-
        show="personForm.email.$error.email">Invalid email</span>
        <br><br>
        <input type="checkbox" ng-model="terms" name="terms"
        id="terms" required/>
        <label for="terms">I agree to the terms.</label>
        <span class="error" ng-
        show="personForm.terms.$error.required">You must agree to
        the terms</span>
        <br><br/>
        <button type="submit">Submit Form</button>
        <br><br/>
        <span>{{msg}}</span>
        </form>
        </body>
</html>
```

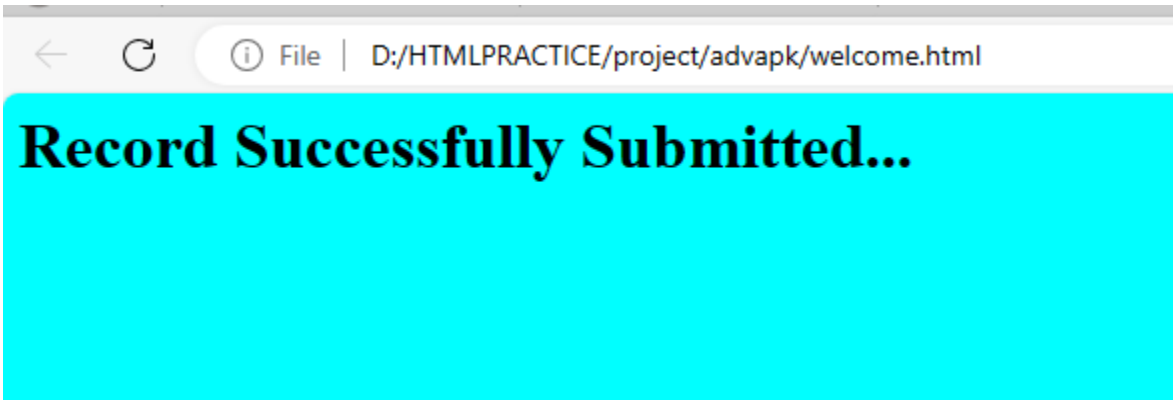
Output:-


File | D:/HTMLPRACTICE/project/advapk/p
Form Validation demo app in AngularJs
Name Prathamesh
Address Airoli
Contact No 9767641692
Email Prathameshg44@gmail.com
I agree to the terms.
Submit Form

Welcome.html

```
<html>
<head>
<title>Welcome Page</title>
</head>
<body bgcolor="aqua">
<h1>Record Successfully Submitted...</h1>
</body>
</html>
```

Output:-



Practical 7

Create an application for Customer/Students records using AngularJS.

```
<!DOCTYPE html>

<html ng-app="studentApp">

<head>

<scriptsrc="https://ajax.googleapis.com/ajax/libs/angularjs/1.3.16/angular.min.js">
</script>

</head>

<body ng-controller="studentController">

<h1>

Student Information:

</h1>

<form ng-submit="submitStudentForm()">

<label for="firstName">First Name:</label><br/>

<input type="text"id="firstName"ng-model="student.firstName"/><br/>

<label for="lastName">Last Name:</label><br/>

<input type=<text"id="lastName"ng-model="student.lastName"/><br/>

<label for="dob">DOB</label><br/>

<input type="date"id="dob"ng-model="student.DOB"/><br/>

<label for="gender">Gender</label><br/>

<select id="gender"ng-model="student.gender">

<option value="male">Male</option>
```

<option value="Female">Female</option>

</select>

Training Type:

<label><input value="online" type="radio" name="training" ng-model="student.trainingType"/>Online</label>

<label><input value="onsite" type="radio" name="training" ng-model="student.trainingType"/>Onsite</label>

Subjects

<label><input type="checkbox"ng-model="student.maths"/>Maths</label>

<label><input type="checkbox"ng-model="student.physics"/>Physics</label>

<label><input type="checkbox"ng-model="student.chemistry"/>Chemistry</label>

<input type="submit"value="Submit"/>

<input type="reset"ng-click="resetForm()"value="Reset"/>

</form>

<script>

var studentApp = angular.module('studentApp',[]);

studentApp.controller("studentController",function($scope,$http){

$scope.originalStudent={

firstName:'Prathamesh',

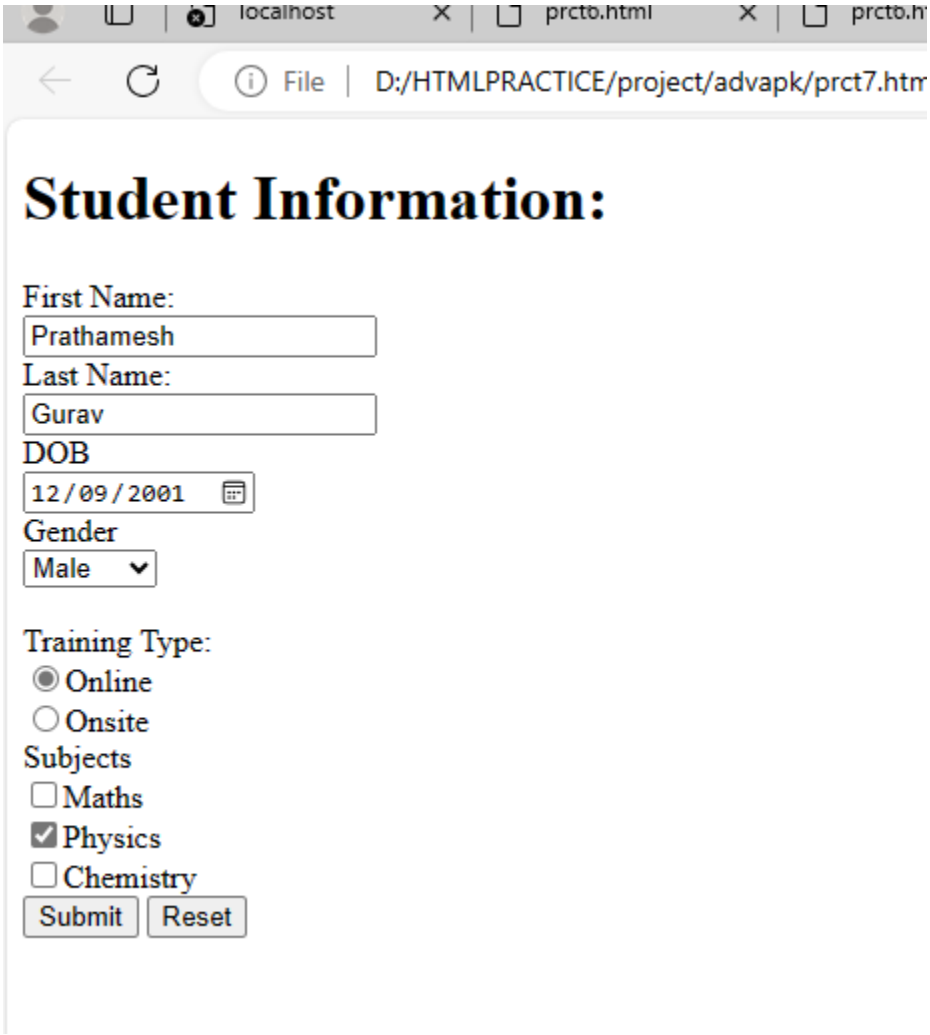
lastName:'Gurav',

DOB:Date('12/09/2001'),

gender:'female',

```
trainingType:'Online',
maths:false,
physics:false,
chemistry:true
};
$scope.student=angular.copy($scope.originalStudent);
$scope.submitStudentForm=function(){
};
$scope.resetForm=function(){
$scope.student=angular.copy($scope.originalStudent);
};
});
</script>
</body>
</html>
```

Output:-


localhost
prcto.html
File
D:/HTMLPRACTICE/project/advapk/prct7.htm
Student Information:
First Name:
Prathamesh
Last Name:
Gurav
DOB
12/09/2001
Gender
Male
Training Type:
Online
Onsite
Subjects
Maths
Physics
Chemistry
Submit
Reset

Practical 8

Write a program to implement Error Handling in AngularJS.

```
<!DOCTYPE html>
<html>

<head>
   <title>Exception Handling in AngularJS</title>
   <style>
      body{
         background: aqua;
      }
      input {
         width: 100px;
         padding: 5px 15px;
         margin: 5px 0;
         box-sizing: border-box;
         border: 2px solid #ccc;
         border-radius: 4px;
      }

      button {
         width: 80px;
         background-color: #4caf50;
         color: white;
         padding: 6px 12px;
         margin: 5px 0;
         border: none;
         border-radius: 4px;
         cursor: pointer;
      }
```

```
        h1 {
            color: green;
        }
    </style>
</head>

<body ng-app="myApp">

    <center>
        <h1> WELCOME</h1>
    </center>
    <div ng-controller="myCtrl">
        <p>Enter a number:</p>
        <input type="number" ng-model="num" />
        <button ng-click="calculate()">Calculate</button>
        <p>Result: {{result}}</p>
        <p>Any Exception: {{errorMessage}}</p>
    </div>

    <script src=
"https://ajax.googleapis.com/ajax/libs/angularjs/1.7.9/angular.min.js">
    </script>
    <script>
        angular.module('myApp', [])
            .controller('myCtrl', function ($scope, $exceptionHandler) {
                $scope.calculate = function () {
                    try {
                        if ($scope.num < 0) {
                            $scope.result = "Not Possible";
                            throw new Error("Number must be positive.");
                        }
                        $scope.result = Math.sqrt($scope.num);
```

```
                $scope.errorMessage = "No Exceptions";
            } catch (e) {
                $scope.errorMessage = e.message;
                $exceptionHandler(e);
            }
        }
    });
</script>
</body>
</html>
```

Output:-