Slip1

Write an AngularJS script for addition of two numbers using ng-init, ng-model & ng-bind. And also demonstrate ng-show, ng-disabled, ng-click directives on button component.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>AngularJS Addition Example</title>
  <script src="angular.min.js"></script>
</head>
<body ng-app="myApp" ng-controller="myController">

  <!-- Input fields for two numbers with ng-model -->
  <div ng-init="num1=0; num2=0; result=0">
    <label>Number 1:</label>
    <input type="number" ng-model="num1" placeholder="Enter first number" />
    <br>
    <label>Number 2:</label>
    <input type="number" ng-model="num2" placeholder="Enter second number" />
  </div>

  <!-- Button for calculating the result -->
  <button
    ng-click="addNumbers()"
    ng-disabled="num1 === null || num2 === null"
    ng-show="num1 !== null && num2 !== null">
    Add Numbers
  </button>

  <!-- Display the result with ng-bind -->
  <div>
    <h3>Result: <span ng-bind="result"></span></h3>
  </div>

  <!-- Show message if either of the inputs is empty -->
  <p ng-show="num1 === null || num2 === null" style="color: red;">
    Please enter both numbers to enable addition.
  </p>

  <script>
    // Define the AngularJS application
    angular.module('myApp', [])
      .controller('myController', function($scope) {
        // Function to add numbers
        $scope.addNumbers = function() {
          $scope.result = $scope.num1 + $scope.num2;
```

```
        };
      });
  </script>

</body>
</html>
```

Slip:2

Write an AngularJS script to print details of bank (bank name, MICR code, IFC code, address etc.) in tabular form using ng-repeat.

```
<!DOCTYPE html>
<html lang="en" ng-app="bankApp">
<head>
 <meta charset="UTF-8">
 <title>Bank Details</title>
 <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-controller="BankController">

 <h1>Bank Details</h1>

 <table border="1">
  <tr>
   <th>Bank Name</th>
   <th>MICR Code</th>
   <th>IFSC Code</th>
   <th>Address</th>
  </tr>
  <tr ng-repeat="bank in banks">
   <td>{{ bank.name }}</td>
   <td>{{ bank.micr }}</td>
   <td>{{ bank.ifsc }}</td>
   <td>{{ bank.address }}</td>
  </tr>
 </table>

 <script>
  angular.module('bankApp', [])
   .controller('BankController', function($scope) {
    $scope.banks = [
```

```
      { name: 'SBI', micr: '123456789', ifsc: 'BOA0001234', address: '123 Main St, New York, NY' },
      { name: 'TJSB', micr: '987654321', ifsc: 'CHAS0009876', address: '456 Elm St, Los Angeles, CA' },
      { name: 'HDFC', micr: '456123789', ifsc: 'WF0004567', address: '789 Pine St, Chicago, IL' }
    ];
  });
</script>

</body>
</html>
```

Slip-3

Write an AngularJS script to display list of games stored in an array on click of button using ng-click and also demonstrate ng-init, ng-bind directive of AngularJS.

```
<!DOCTYPE html>
<html lang="en" ng-app="gameApp">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Game List with AngularJS</title>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-controller="GameController" ng-init="initializeGames()">

  <h1>Game List</h1>

  <!-- ng-bind directive to display the title -->
  <p ng-bind="gameListTitle"></p>

  <!-- ng-click directive to load and display the list of games -->
  <button ng-click="showGames()">Show Games</button>

  <!-- ng-repeat to display the list of games -->
  <ul>
    <li ng-repeat="game in games">{{ game }}</li>
  </ul>

  <script>
    // Define the AngularJS module
    var app = angular.module('gameApp', []);

    // Define the GameController
```

```
    app.controller('GameController', function($scope) {
      // ng-init to initialize data (used here to set the title)
      $scope.initializeGames = function() {
        $scope.gameListTitle = "Click the button to see the list of games:";
        $scope.games = []; // Empty list to start with
      };

      // ng-click function to show the games
      $scope.showGames = function() {
        // Example array of games
        $scope.games = [
          "Minecraft",
          "The Witcher 3",
          "Cyberpunk 2077",
          "Among Us",
          "Fortnite"
        ];
      };
    });
  </script>

</body>
</html>
```

Slip3B

Find a company with a workforce greater than 30 in the array (use find by id method)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Find Company by Workforce</title>
</head>
<body>

  <h1>Find Company with Workforce Greater Than 30</h1>

  <button onclick="findCompany()">Find Company</button>

  <p id="company-info"></p>

  <script>
    // Array of company objects
```

```
    const companies = [
      { id: 1, name: "Company A", workforce: 25 },
      { id: 2, name: "Company B", workforce: 26},
      { id: 3, name: "Company C", workforce: 10 },
      { id: 4, name: "Company D", workforce: 35 },
      { id: 5, name: "Company E", workforce: 15 }
    ];

    // Function to find a company with workforce greater than 30
    function findCompany() {
      // Use the find() method to search by id and workforce condition
      const company = companies.find(c => c.workforce > 30);

      // Display the result in the paragraph with id 'company-info'
      if (company) {
        document.getElementById("company-info").textContent =
          `Found company: ${company.name} with a workforce of ${company.workforce}`;
      } else {
        document.getElementById("company-info").textContent = "No company found with a
workforce greater than 30.";
      }
    }
  </script>

</body>
</html>
```

Slip4

Fetch the details using ng-repeat in AngularJS.

```
<!DOCTYPE html>
<html lang="en" ng-app="studentApp">
<head>
 <meta charset="UTF-8">
 <title>Student Details</title>
 <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-controller="StudentController">

 <h1>Student Details</h1>

 <table border="1">
  <tr>
   <th>Name</th>
   <th>Roll Number</th>
```

```
    <th>Class</th>
    <th>Grade</th>
  </tr>
  <tr ng-repeat="student in students">
    <td>{{ student.name }}</td>
    <td>{{ student.rollNumber }}</td>
    <td>{{ student.class }}</td>
    <td>{{ student.grade }}</td>
  </tr>
 </table>

 <script>
  angular.module('studentApp', [])
    .controller('StudentController', function($scope) {
      // Defining student data directly in the controller
      $scope.students = [
        {
          name: 'Ram Kale',
          rollNumber: '101',
          class: '10th Grade',
          grade: 'A'
        },
        {
          name: 'Tushar Maske',
          rollNumber: '102',
          class: '10th Grade',
          grade: 'B+'
        },
        {
          name: 'Rohit mane',
          rollNumber: '103',
          class: '10th Grade',
          grade: 'A-'
        },
        {
          name: 'Rahul kale',
          rollNumber: '104',
          class: '10th Grade',
          grade: 'B'
        }
      ];
    });
 </script>

</body>
```

```
</html>
```

Slip11

Create Angular application that print the name of students who play basketball using filter and map method.

```
ng new student-basketball-app
cd student-basketball-app
ng generate component student-list
```

Modify the student-list.component.ts file

```
import { Component, OnInit } from '@angular/core';

interface Student {
  name: string;
  playsBasketball: boolean;
}

@Component({
  selector: 'app-student-list',
  templateUrl: './student-list.component.html',
  styleUrls: ['./student-list.component.css']
})
export class StudentListComponent implements OnInit {
  students: Student[] = [
    { name: 'Alice', playsBasketball: true },
    { name: 'Bob', playsBasketball: false },
    { name: 'Charlie', playsBasketball: true },
    { name: 'David', playsBasketball: false },
    { name: 'Eva', playsBasketball: true }
  ];

  basketballPlayers: string[] = [];

  ngOnInit() {
    // Use filter to get students who play basketball, then map to extract their names
    this.basketballPlayers = this.students
      .filter(student => student.playsBasketball)  // filter students who play basketball
      .map(student => student.name);              // map to only get the names
  }
}
```

Update the student-list.component.html file

```
<div>
```

```
 <h2>Students who play Basketball:</h2>
 <ul>
  <li *ngFor="let player of basketballPlayers">{{ player }}</li>
 </ul>
</div>
```

Modify the app.component.html file

```
<app-student-list></app-student-list>
```

Run the Application

```
ng serve
```

slip13

Write an AngularJS script to print details of Employee (employee name, employee Id,Pin code, address etc.) in tabular form using ng-repeat.

```
<!DOCTYPE html>
<html lang="en" ng-app="employeeApp">
<head>
 <meta charset="UTF-8">
 <title>Employee Details</title>
 <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-controller="EmployeeController">

 <h1>Employee Details</h1>

 <table border="1">
  <tr>
   <th>Employee Name</th>
   <th>Employee ID</th>
   <th>Pin Code</th>
   <th>Address</th>
  </tr>
  <tr ng-repeat="employee in employees">
   <td>{{ employee.name }}</td>
   <td>{{ employee.id }}</td>
   <td>{{ employee.pinCode }}</td>
   <td>{{ employee.address }}</td>
  </tr>
 </table>

 <script>
  angular.module('employeeApp', [])
   .controller('EmployeeController', function($scope) {
    // Defining employee data directly in the controller
```

```
    $scope.employees = [
      {
        name: 'Alice Johnson',
        id: 'E101',
        pinCode: '123456',
        address: '123 Oak Street, Springfield'
      },
      {
        name: 'Bob Smith',
        id: 'E102',
        pinCode: '654321',
        address: '456 Maple Avenue, Riverside'
      },
      {
        name: 'Carol Williams',
        id: 'E103',
        pinCode: '789012',
        address: '789 Pine Road, Hilltop'
      },
      {
        name: 'David Brown',
        id: 'E104',
        pinCode: '345678',
        address: '101 Cedar Lane, Greenfield'
      }
    ];
  });
</script>

</body>
</html>
```

Slip5A)

Slip5B)

Implement a simple server using Node.js.

```
// server.js
const http = require('http');

// Define the hostname and port
const hostname = 'localhost';
const port = 3000;

// Create the server
```

```
const server = http.createServer((req, res) => {
  // Set the response HTTP header with status and content type
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');

  // Handle different routes
  if (req.url === '/') {
    res.end('Welcome to the Home Page!');
  } else if (req.url === '/about') {
    res.end('This is the About Page.');
  } else if (req.url === '/contact') {
    res.end('Contact us at: contact@example.com');
  } else {
    res.statusCode = 404;
    res.end('404 - Page Not Found');
  }
});

// Start the server
server.listen(port, hostname, () => {
  console.log('server contected.....');
});
```

Slip5a)

Create a simple Angular component that takes input data and displays it.

ng new slip5a

cd slip5a

Step 1: Generate the Student Component

ng generate component student

step2:

// src/app/student/student.component.ts

```
import { Component, Input } from '@angular/core';

@Component({
 selector: 'app-student',
 templateUrl: './student.component.html',
 styleUrls: ['./student.component.css']
})
export class StudentComponent {
 @Input() student: { name: string; age: number; grade: string } | undefined;
}
```

Step 3: Display the Input Data in the Template

```
<p>student works!</p>
```

```
<!-- src/app/student/student.component.html -->
<div *ngIf="student">
    <h2>Student Information</h2>
    <p>Name: {{ student.name }}</p>
    <p>Age:{{ student.age }}</p>
    <p>Grade:{{ student.grade }}</p>
  </div>
```

Step 4: Use the `StudentComponent` in a Parent Component

a)

```
// src/app/app.component.ts
import { Component } from '@angular/core';

@Component({
 selector: 'app-root',
 templateUrl: './app.component.html',
 styleUrls: ['./app.component.css']
})
export class AppComponent {
 // Define a student object to pass as input
 student = {
  name: 'John Doe',
  age: 18,
  grade: 'A'
 };
}
```

b)

```
<!-- src/app/app.component.html -->
<div style="text-align:center">
 <h1>Student Details</h1>
 <app-student [student]="student"></app-student>
</div>
```

Step 5: Run the Application

ng serve –o

Slip6A)

Develop an Express.js application that defines routes for Create and Read operations on a resource (products).

Slip7B

Develop an Express.js application that defines routes for Create and Read operations on a resource (User)

Slip8B
Develop an Express.js application that defines routes for Create, Update operations on a resource (Employee).

Slip11B
Develop an Express.js application that defines routes for Create operations on a resource (Movie).
Slip12B
Develop an Express.js application that defines routes for Create operations on a resource (User).
Slip14B
Develop an Express.js application that defines routes for Create, Update operations on a resource (Employee).

Refer-Practical 11

Slip 15)
Find an emp with a Salary greater than 25000 in the array. (Using find by id method)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Find Employee by ID and Salary</title>
</head>
<body>
    <h1>Find Employee</h1>

    <label for="employeeId">Enter Employee ID:</label>
    <input type="number" id="employeeId" placeholder="Enter ID" required>
    <button onclick="findEmployee()">Find Employee</button>

    <h2>Employee Details:</h2>
    <div id="result"></div>

    <script>
        // Array of employee objects
        const employees = [
            { id: 1, name: "Ram", salary: 20000 },
            { id: 2, name: "sham", salary: 30000 },
            { id: 3, name: "Rahul", salary: 25000 },
            { id: 4, name: "David", salary: 40000 }
        ];

        // Function to find employee by ID and check salary
        function findEmployee() {
            const id = parseInt(document.getElementById("employeeId").value,
10);
```

```
            const salaryLimit = 25000;
            const resultDiv = document.getElementById("result");

            // Clear previous result
            resultDiv.innerHTML = '';

            // Find the employee
            const employee = employees.find(emp => emp.id === id && emp.salary
> salaryLimit);

            if (employee) {
                resultDiv.innerHTML = `
                    <p>Name: ${employee.name}</p>
                    <p>Salary: ${employee.salary}</p>
                `;
            } else {
                resultDiv.innerHTML = "<p>No employee found with the specified
criteria.</p>";
            }
        }
    </script>
</body>
</html>
```

Slip14a) and Slip15b)

Create Angular application that print the name of students who got 85% using filter and map method.

ng new slip14a
cd slip14a
ng generate component student-list

Step 2: Define the Student Data and Filter Logic

1. Open `student-list.component.ts`.

```
// src/app/student-list/student-list.component.ts
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-student-list',
  templateUrl: './student-list.component.html',
  styleUrls: ['./student-list.component.css']
})
export class StudentListComponent implements OnInit {
  // Array of students with name and percentage
```

```
  students = [
    { name: 'Ram', percentage: 90 },
    { name: 'sham', percentage: 75 },
    { name: 'Rahul', percentage: 88 },
    { name: 'David', percentage: 83 },
    { name: 'Rohit', percentage: 92 }
  ];

  // Array to store names of students who scored above 85%
  highScorers: string[] = [];

  ngOnInit(): void {
    this.highScorers = this.students
      .filter(student => student.percentage > 85)  // Filter students with
percentage > 85
      .map(student => student.name);              // Map to get only the
names of filtered students
  }
}
2.
```

Step 3: Display the Filtered Names in the Template

In `student-list.component.html`

```
<p>student-list works!</p>
<!-- src/app/student-list/student-list.component.html -->
<h2>Students Who Scored Above 85%</h2>
<ul>
  <li *ngFor="let name of highScorers">{{ name }}</li>
</ul>
```

Step 4: Add the Component to the App

Open `app.component.html`

<!-- src/app/app.component.html -->
<app-student-list></app-student-list>

Step 5: Run the Application

Run the Angular application to see the list of students who scored above 85%:

```
ng serve
```