1.Create an angular project which has HTML template which has three courses (angular, NodeJS and Java Script).Create custom component of three above courses. If you click on Register button then the message (You have registered _____ Course) will be display on the web page.[Note: Use services and dependency Injection]

app-routing.module.ts

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';

const routes: Routes = [];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

app.component.html

```
<div>
  <app-javascript></app-javascript>
  <app-angular></app-angular>
  <app-node></app-node>
</div>
```

app.component.ts

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'course';
```

```
}
```

app.module.ts

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { JavascriptComponent } from './javascript/javascript.component';
import { AngularComponent } from './angular/angular.component';
import { NodeComponent } from './node/node.component';

@NgModule({
  declarations: [
    AppComponent,
    JavascriptComponent,
    AngularComponent,
    NodeComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

angular.component.css

```
.container{
    width:200px;
    height:200px;
```

```
    display:inline-block;
    margin-left:200px;
    margin-top:0px;
}
button{
    padding:20px 20px;
    width:100px;
}
```

angular.component.html

```
<div class="container">
    <div><img src="assets/angular.png" style="width:240px; height:180px;"></div>
    <div style="text-align:center;">
        <h3>{{title}}</h3>
    </div>
    <div style="text-align:center; padding:20px 0px;background-color: aqua;">
        <button (click)="OnEnroll()">Enroll</button>
    </div>
</div>
```

angular.component.ts

```
import { Component } from '@angular/core';
import { EnrollService } from '../Services/enroll.service';

@Component({
  selector: 'app-angular',
  templateUrl: './angular.component.html',
  styleUrls: ['./angular.component.css'],
  providers:[EnrollService]
})
export class AngularComponent {
  title="Angular";
```

```
  constructor(private enrollService:EnrollService){
  }
  OnEnroll(){
     this.enrollService.OnEnrollClicked(this.title);
  }
}
```

javascript.component.css

```
.container{
    width:200px;
    height:200px;
    display:inline-block;
    margin-left:200px;
    margin-top:0px;
}
button{
    padding:20px 20px;
    width:100px;
}
```

javascript.component.html

```
<div class="container">
    <div><img src="assets/js.png" style="width:240px; height:180px;"></div>
    <div style="text-align:center;">
        <h3>{{title}}</h3>
    </div>
    <div style="text-align:center; padding:20px 0px;background-color: blue;">
        <button (click)="OnEnroll()">Enroll</button>
    </div>
</div>
```

javascript.component.ts

```
import { Component } from '@angular/core';
import { EnrollService } from '../Services/enroll.service';

@Component({
  selector: 'app-javascript',
  templateUrl: './javascript.component.html',
  styleUrls: ['./javascript.component.css'],
  providers:[EnrollService]
})
export class JavascriptComponent {
title="JavaScript";
constructor(private enrollService:EnrollService){
}

OnEnroll(){
   this.enrollService.OnEnrollClicked(this.title);
}
}
```

node.component.css

```
.container{
    width:200px;
    height:200px;
    display:inline-block;
    margin-left:200px;
    margin-top:0px;
}
button{
    padding:20px 20px;
    width:100px;
}
```

node.component.html

```
<div class="container">
   <div><img src="assets/node.png" style="width:240px; height:180px;"></div>
   <div style="text-align:center;">
      <h3>{{title}}</h3>
   </div>
   <div style="text-align:center; padding:20px 0px;background-color: rgb(255, 0, 64);">
      <button (click)="OnEnroll()">Enroll</button>
   </div>
</div>
```

node.component.ts

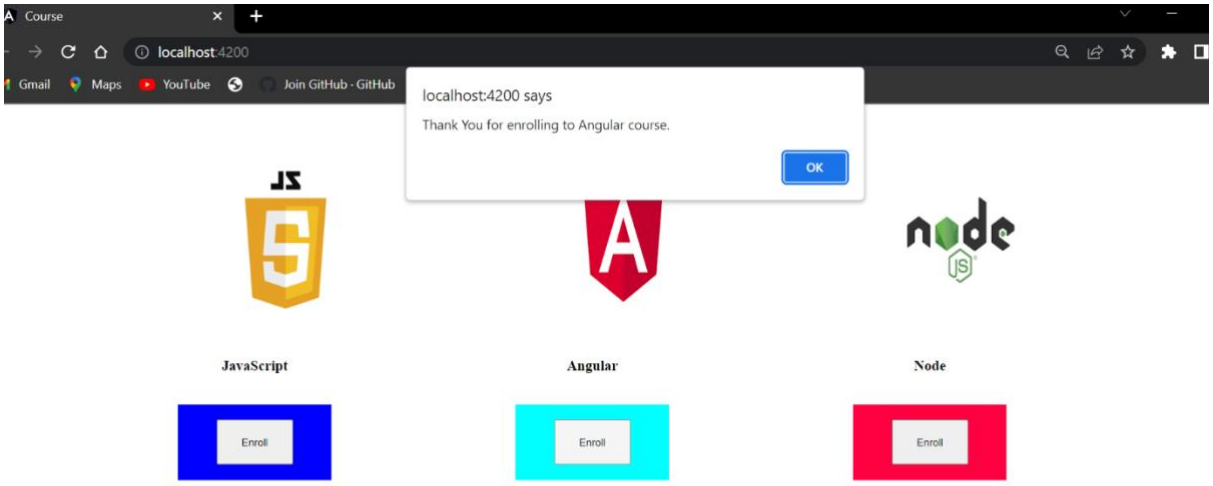
```
import { Component } from '@angular/core';
import { EnrollService } from '../Services/enroll.service';

@Component({
  selector: 'app-node',
  templateUrl: './node.component.html',
  styleUrls: ['./node.component.css'],
  providers:[EnrollService]
})
export class NodeComponent {
  title = "Node";
  constructor(private enrollService:EnrollService){
  }
  OnEnroll(){
     this.enrollService.OnEnrollClicked(this.title);
  }
}
```

enroll.services.ts

```
export class EnrollService{
```

```
  OnEnrollClicked(title:string){
    alert("Thank You for enrolling to "+title+" course.");
  }
}
```

Output :



2. Create an angular project which has HTML template which has the following format. When you click on any link respective component should be display on the same page. [Use Routing] (e.g. If you click on Home link then "You are on Home Page." This message should be display on the page.)

Home About Us Products Contact Us

You are on Home Page.

app-routing.modules.ts

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { HomeComponent } from './home/home.component';
import { ContactComponent } from './contact/contact.component';
import { AboutComponent } from './about/about.component';
import { ProductComponent } from './product/product.component';

const routes: Routes = [
  { path: '', redirectTo: 'home ', pathMatch: 'full' },
  { path: 'home', component: HomeComponent },
  { path: 'about', component: AboutComponent },
  { path: 'contact', component: ContactComponent },
  { path: 'products', component: ProductComponent }
];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

app.component.css

```
nav {
  background-color: #f0f0f0;
  padding: 10px;
```

```
}

ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
}

li {
  display: inline;
  margin-right: 10px;
}

a {
  color: #333;
  text-decoration: none;
  padding: 5px;
}

a:hover {
  background-color: #333;
  color: #fff;
}

.active {
  background-color: #333;
  color: #fff;
}
```

app.component.html

```
<nav>
  <ul>
```

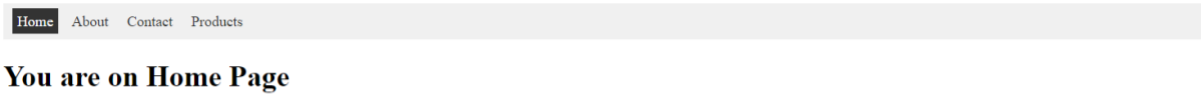
```
    <li><a routerLink="/home" routerLinkActive="active">Home</a></li>
    <li><a routerLink="/about" routerLinkActive="active">About</a></li>
    <li><a routerLink="/contact" routerLinkActive="active">Contact</a></li>
    <li><a routerLink="/products" routerLinkActive="active">Products</a></li>
  </ul>
</nav>
<div>
  <router-outlet></router-outlet>
</div>
```

app.module.ts

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { HomeComponent } from './home/home.component';
import { AboutComponent } from './about/about.component';
import { ContactComponent } from './contact/contact.component';
import { ProductComponent } from './product/product.component';




@NgModule({
  declarations: [
    AppComponent,
    HomeComponent,
    AboutComponent,
    ContactComponent,
    ProductComponent
  ],
  imports: [
```

```
    BrowserModule,
    AppRoutingModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

Output :



3. Using Angular display the 10 student details in Table format (using directive and use Array to store data).

app.component.html

```
<h1>Student Details</h1>
<app-student-table></app-student-table>
```

student-table.component.html

```
<table>
  <thead>
    <tr>
      <th>ID</th>
      <th>Name</th>
      <th>Age</th>
      <th>Grade</th>
    </tr>
  </thead>
  <tbody>
    <tr *ngFor="let student of students">
      <td>{{ student.id }}</td>
      <td>{{ student.name }}</td>
      <td>{{ student.age }}</td>
      <td>{{ student.grade }}</td>
    </tr>
  </tbody>
</table>
```

student-table.component.css

```
table {
  width: 100%;
  border-collapse: collapse;
}

th, td {
```

```
    border: 1px solid #ccc;
    padding: 8px;
  }

  th {
    background-color: #f2f2f2;
    font-weight: bold;
  }
```

student-table.component.ts

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

@Component({
  selector: 'app-student-table',
  templateUrl: './student-table.component.html',
  styleUrls: ['./student-table.component.css']
})
export class StudentTableComponent {
  students: any[] = [
    { id: 1, name: 'John Doe', age: 20, grade: 'A' },
    { id: 2, name: 'Jane Smith', age: 22, grade: 'B' },
    { id: 3, name: 'Michael Johnson', age: 21, grade: 'A+' },
    { id: 4, name: 'Emily Davis', age: 19, grade: 'B+' },
    { id: 5, name: 'Daniel Wilson', age: 23, grade: 'A' },
    { id: 6, name: 'Sophia Thompson', age: 20, grade: 'A-' },
    { id: 7, name: 'William Anderson', age: 22, grade: 'B' },
    { id: 8, name: 'Olivia Rodriguez', age: 21, grade: 'A+' },
    { id: 9, name: 'James Lee', age: 19, grade: 'B+' },
    { id: 10, name: 'Ava Martinez', age: 23, grade: 'A' }
  ];
}
```

Output :

Student Details

ID	Name	Age	Grade
1	John Doe	20	A
2	Jane Smith	22	B
3	Michael Johnson	21	A+
4	Emily Davis	19	B+
5	Daniel Wilson	23	A
6	Sophia Thompson	20	A-
7	William Anderson	22	B
8	Olivia Rodriguez	21	A+
9	James Lee	19	B+
10	Ava Martinez	23	A

4. Create an angular program which accept the employees information(eid, ename ,department, salary, DOJ) and display all information in tabular format on the same page.(Use-Two way Data Binding).

app.component.html

```
<div style="text-align:center">
  <h1>Welcome to Employee Management</h1>
  <app-employee></app-employee>
</div>
```

app.module.ts

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

import { AppComponent } from './app.component';
import { EmployeeComponent } from './employee/employee.component';
import { FormsModule } from '@angular/forms';



@NgModule({
  declarations: [
    AppComponent,
    EmployeeComponent
  ],
  imports: [
    BrowserModule,
    FormsModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

employee.component.css

```
table {
  border-collapse: collapse;
  width: 100%;
}

th, td {
  border: 1px solid black;
  padding: 8px;
}

th {
  background-color: #f2f2f2;
}

input {
  margin-bottom: 10px;
}
```

employee.component.html

```
<h2>Add Employee</h2>
<div>
  <label>Employee ID:</label>
  <input [(ngModel)]="employee.eid" type="text" placeholder="Employee ID">
</div>
<div>
  <label>Employee Name:</label>
  <input [(ngModel)]="employee.ename" type="text" placeholder="Employee Name">
</div>
<div>
  <label>Department:</label>
  <input [(ngModel)]="employee.department" type="text" placeholder="Department">
```

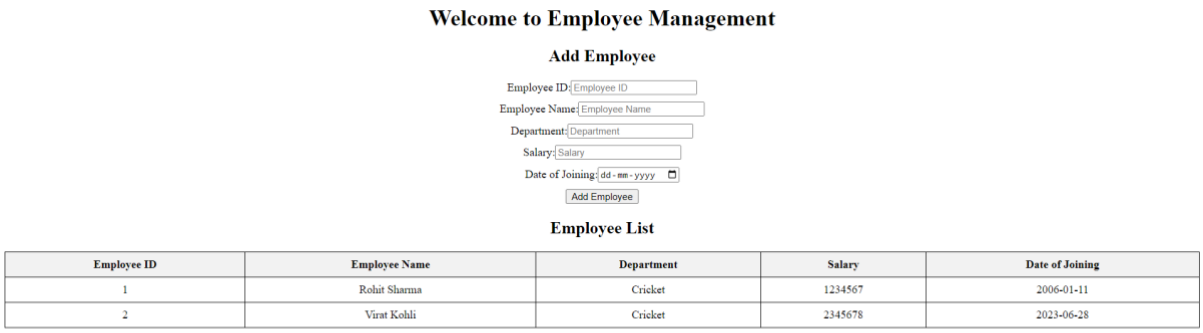
```
</div>
<div>
  <label>Salary:</label>
  <input [(ngModel)]="employee.salary" type="number" placeholder="Salary">
</div>
<div>
  <label>Date of Joining:</label>
  <input [(ngModel)]="employee.DOJ" type="date">
</div>

<button (click)="addEmployee()">Add Employee</button>

<h2>Employee List</h2>
<table>
  <tr>
    <th>Employee ID</th>
    <th>Employee Name</th>
    <th>Department</th>
    <th>Salary</th>
    <th>Date of Joining</th>
  </tr>
  <tr *ngFor="let emp of employees">
    <td>{{ emp.eid }}</td>
    <td>{{ emp.ename }}</td>
    <td>{{ emp.department }}</td>
    <td>{{ emp.salary }}</td>
    <td>{{ emp.DOJ }}</td>
  </tr>
</table>
```

employee.component.ts

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

```
@Component({
  selector: 'app-employee',
  templateUrl: './employee.component.html',
  styleUrls: ['./employee.component.css']
})
export class EmployeeComponent {
  employee: any = {};
  employees: any[] = [];

  addEmployee() {
    this.employees.push({ ...this.employee });
    this.employee = {};
  }
}
```

Output:



5. Create an angular program which accept a number from user and check whether it is even number or odd number.[Use ngif Directive]

app.component.html

```
<div style="text-align:center">
  <h1>Welcome to Number Checker</h1>
  <app-number-check></app-number-check>
</div>
```

app.module.ts

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { AppComponent } from './app.component';
import { NumberCheckComponent } from './number-check/number-check.component';
import { FormsModule } from '@angular/forms';

@NgModule({
  declarations: [
    AppComponent,
    NumberCheckComponent
  ],
  imports: [
    BrowserModule,
    FormsModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

number-check.component.css

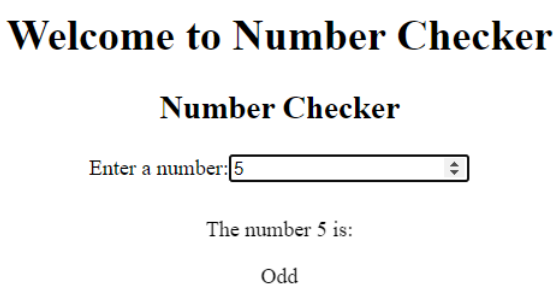
```
input {
    margin-bottom: 10px;
  }
```

number-check.component.html

```
<h2>Number Checker</h2>
<div>
  <label>Enter a number:</label>
  <input [(ngModel)]="inputNumber" type="number">
</div>
<div *ngIf="inputNumber !== null">
  <p>The number {{ inputNumber }} is:</p>
  <p *ngIf="isEven(inputNumber)">Even</p>
  <p *ngIf="!isEven(inputNumber)">Odd</p>
</div>
```

number-check.component.ts

```
import { Component } from '@angular/core';
@Component({
  selector: 'app-number-check',
  templateUrl: './number-check.component.html',
  styleUrls: ['./number-check.component.css']
})
export class NumberCheckComponent {
  inputNumber: number | null = null;

  isEven(num: number): boolean {
    return num % 2 === 0;
  }
}
```



6. Create an angular program which display the products detail(pid,pname,price,quantity,available,image). The products which are "Available"-apply background color -green "not available" –apply background color- red [Use attribute directives and array to store the data].

app.component.html

```
<div style="text-align:center">
  <h1>Welcome to Product Details</h1>
  <app-product-list></app-product-list>
</div>
```

availability-color.directive.ts

```
import { Directive, ElementRef, Input, OnInit } from '@angular/core';

@Directive({
  selector: '[appAvailabilityColor]'
})
export class AvailabilityColorDirective implements OnInit {
  @Input('appAvailabilityColor') availability: string = '';

  constructor(private elementRef: ElementRef) {}

  ngOnInit() {
    this.elementRef.nativeElement.style.backgroundColor = this.availability === 'Available' ? 'green' : 'red';
  }
}
```

product-list.component.css

```
table {
    border-collapse: collapse;
    width: 100%;
  }
```

```
 th, td {
   border: 1px solid black;
   padding: 8px;
 }

 th {
   background-color: #f2f2f2;
 }

 img {
   max-width: 100px;
   max-height: 100px;
 }
```

product-list.component.html

```
<h2>Product List</h2>

<table>
  <tr>
    <th>Product ID</th>
    <th>Product Name</th>
    <th>Price</th>
    <th>Quantity</th>
    <th>Availability</th>
    <th>Image</th>
  </tr>
  <tr *ngFor="let product of products" [appAvailabilityColor]="product.available">
    <td>{{ product.pid }}</td>
    <td>{{ product.pname }}</td>
    <td>{{ product.price }}</td>
    <td>{{ product.quantity }}</td>
    <td>{{ product.available }}</td>
```

```
    <td><img [src]="product.image" alt="Product Image"></td>
  </tr>
</table>
```

product-list.component.ts

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


@Component({
  selector: 'app-product-list',
  templateUrl: './product-list.component.html',
  styleUrls: ['./product-list.component.css']
})
export class ProductListComponent {
  products: any[] = [
    { pid: 1, pname: 'Product 1', price: 10, quantity: 5, available: 'Available', image: '/assets/product1.jpg' },
    { pid: 2, pname: 'Product 2', price: 15, quantity: 0, available: 'Not Available', image: '/assets/product2.jpg' },
    { pid: 1, pname: 'Product 3', price: 10, quantity: 5, available: 'Available', image: '/assets/product3.jpg' },
    { pid: 2, pname: 'Product 4', price: 15, quantity: 0, available: 'Not Available', image: '/assets/product4.jpg' }
  ];
}
```

Output :

Welcome to Product Details

Product List

Product ID	Product Name	Price	Quantity	Availability	Image
1	Product 1	10	5	Available	
2	Product 2	15	0	Not Available	
3	Product 3	10	5	Available	
4	Product 4	15	0	Not Available	

7. Using angular create a SPA that to accept the details such as name, mobile number, pincode, email address and perform validations name should contains character only, mobile number should contains only 10 digit, pin code should contains 6 digits.

app.component.html

```
<div>
  <app-details-form></app-details-form>
</div>
```

details-form.component.css

```
h2 {
    color: #333;
    font-size: 24px;
    margin-bottom: 20px;
  }
  form {
    margin-top: 20px;
    margin: 0 auto; /* Center align the form */
    width: 70%; /* Adjust the width as needed */
  }
  label {
    display: block;
    margin-bottom: 10px;
    font-weight: bold;
  }
  input[type="text"],
  input[type="tel"],
  input[type="email"] {
    width: 100%;
    padding: 10px;
    margin-bottom: 10px;
    border: 1px solid #ccc;
    border-radius: 4px;
  }
  .error-message {
```

```
  color: red;
  margin-top: 5px;
  font-size: 14px;
 }
 button[type="submit"] {
  padding: 10px 20px;
  background-color: #333;
  color: #fff;
  border: none;
  border-radius: 4px;
  cursor: pointer;
 }
 button[type="submit"]:disabled {
  background-color: #ccc;
  cursor: not-allowed;
 }
```

details-form.component.html

```
<div>
  <form #detailsForm="ngForm" (ngSubmit)="submitForm(detailsForm)">
    <div>
      <label for="name">Name:</label>
      <input type="text" id="name" name="name" [(ngModel)]="name" required pattern="[a-zA-Z
]*">
      <div *ngIf="detailsForm.controls['name'].invalid && detailsForm.controls['name'].touched"
        class="error-message">
        Name is required and should contain only characters.
      </div>
    </div>
    <div>
      <label for="mobile">Mobile Number:</label>
      <input type="tel" id="mobile" name="mobile" [(ngModel)]="mobile" required pattern="[0-
9]{10}">
```

```
        <div *ngIf="detailsForm.controls['mobile'].invalid &&
detailsForm.controls['mobile'].touched"
            class="error-message">
            Mobile number is required and should contain 10 digits.
        </div>
      </div>
      <div>
        <label for="pincode">Pincode:</label>
        <input type="text" id="pincode" name="pincode" [(ngModel)]="pincode" required
pattern="[0-9]{6}">
        <div *ngIf="detailsForm.controls['pincode'].invalid &&
detailsForm.controls['pincode'].touched"
            class="error-message">
            Pincode is required and should contain 6 digits.
        </div>
      </div>
      <div>
        <label for="email">Email Address:</label>
        <input type="email" id="email" name="email" [(ngModel)]="email" required>
        <div *ngIf="detailsForm.controls['email'].invalid && detailsForm.controls['email'].touched"
            class="error-message">
            Email is required and should be a valid email address.
        </div>
      </div>
      <button type="submit" [disabled]="!detailsForm.valid">Submit</button>
    </form>
</div>
```

details-form.component.ts

```
import { Component } from '@angular/core';
@Component({
  selector: 'app-details-form',
  templateUrl: './details-form.component.html',
```

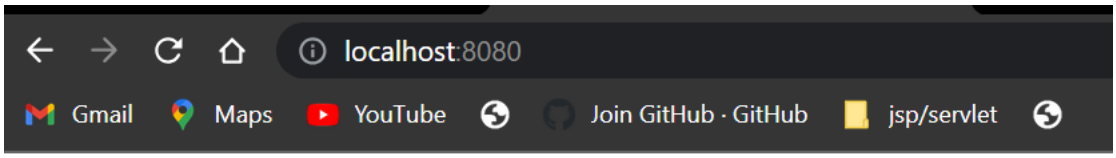
```
  styleUrls: ['./details-form.component.css']
})
export class DetailsFormComponent {
  name: string = '';
  mobile: string = '';
  pincode: string = '';
  email: string = '';
  formSubmitted: any;
  submitForm(form: any) {
    if (form.valid) {
      console.log('Form submitted successfully.');
      console.log('Name:', this.name);
      console.log('Mobile Number:', this.mobile);
      console.log('Pincode:', this.pincode);
      console.log('Email Address:', this.email);
      this.formSubmitted = true;
    }
  }
}
```

Output :

8. Create a node.js file that will convert the output “dypimcam akurdi” into the uppercase letters.

```
var http = require('http')
var uc = require('upper-case')
const port = 8080
const server = http.createServer(function(request,response){
    response.writeHead(200,{'Content-Type' : 'text/html'});
    response.write(uc.upperCase('dypimca'));
    response.end();
}).listen(port)
```

Output :



DYPIMCA

9. Create a Node.js Application that uses user defined module circle.js which exports functions area() and circumference() and display details on console.

Que9.js

```
exports.area = function (r) {
   const pi = 3.14
   console.log(pi*r**2)
}
exports.circumference = function (r) {
   const pi = 3.14
   console.log(2*pi*r)
}
```

Que9.1.js

```
var arth = require("./Que9");
var a = 10;
var b = 11;
arth.area(a)
arth.circumference(a)
```

Output :

```
314
62.800000000000004
```

10. Write a program to read the query string using url property in node js

```
const url = require('url')
const address = 'https://abc.com:9000/search?name=HelloWorld&page=1';
const parsed =url.parse(address,true)
console.log(parsed)
```

Output:

```
Url {
  protocol: 'https:',
  slashes: true,
  auth: null,
  host: 'abc.com:9000',
  port: '9000',
  hostname: 'abc.com',
  hash: null,
  search: '?name=HelloWorld&page=1',
  query: [Object: null prototype] { name: 'HelloWorld', page: '1' },
  pathname: '/search',
  path: '/search?name=HelloWorld&page=1',
  href: 'https://abc.com:9000/search?name=HelloWorld&page=1'
}
```