

SPRING BOOT ASSIGNMENT

1.

```
@RestController
public class AssignmentController {

    @Autowired
    private AssignmentService assignmentService;

    @GetMapping("/welcome")
    public String welcomeMessage() {
        return assignmentService.welcomeMessage();
    }

}
```

```
1 package com.service;
2
3 import org.springframework.stereotype.Service;
4
5 @Service
6 public class AssignmentService {
7
8
9     public String welcomeMessage() {
10         return "Welcome to the Service Message";
11     }
12 }
13
```

2.

```
@GetMapping("/add")
public int addTwoNumbers(@RequestParam int a,@RequestParam int b)
{
    return assignmentService.addTwoNumbers(a, b);
}
```

```
public int addTwoNumbers(int a,int b)
{
    int result=a+b;
    return result;
}
```

3.

```
public String evenOdd(int num)
{
    if(num%2==0)
    {
        return "The Given Number is Even"+num;
    }
    else
    {
        return "The Given Number is Odd"+num;
    }
}
```

```
@GetMapping("/evenodd/{num}")
public String evenOdd(@PathVariable int num)
{
    return assignmentService.evenOdd(num);
}
```

4.

```
        }

    @GetMapping("age/{age}")
    public String ageCategory(@PathVariable int age) {
        return assignmentService.ageCategory(age);
    }

    public String ageCategory(int age)
    {
        if(age<18)
        {
            return "Child";
        }
        else if(age<=58 && age>=18)
        {
            return "Adult";
        }
        else {
            return "Senior";
        }
    }
}
```

5.

```
@GetMapping("/grade/{marks}")
public String GradeCalculation(@PathVariable int marks)
{
    return assignmentService.GradeCalculation(marks);
}
```

```
public String GradeCalculation(int marks) {  
    if(marks>=90)  
    {  
        return "A";  
    }  
    else if(marks<=90 && marks>=75)  
    {  
        return "B";  
    }  
    else if(marks<=75 && marks>=60)  
    {  
        return "C";  
    }  
    else {  
        return "F";  
    }  
}
```

6.

```
@GetMapping("/celsius-to-fahrenheit/{celsius}")  
public int CelsiusToFahrenheit(@PathVariable int celsius)  
{  
    return assignmentService.CelsiusToFahrenheit(celsius);  
}
```

```
    }  
    public int CelsiusToFahrenheit(int celsius)  
    {  
        int result =(celsius*9/5)+32;  
        return result;  
    }  
}
```

7.

```
}

public String reverseString(String toreverse)
{
    StringBuilder sb=new StringBuilder();
    sb.append(toreverse);
    String reversed=sb.reverse().toString();

    return reversed;
}

@GetMapping("/reverse/{text}")
public String reverseString(@PathVariable String text)
{
    return assignmentService.reverseString(text);
}
```

8.

```

public String Palindrome(String palindrome)
{
    boolean result=false;
    char palindromeArray[] = palindrome.toCharArray();
    int j=palindrome.length()-1;
    for(int i=0;i<j/2;i++,j--)
    {
        if(palindromeArray[i]==palindromeArray[j])
        {
            result=true;
        }
        else {
            result=false;
            break;
        }
    }
    if(result==true)
    {
        return "String is Palindrome";
    }
    else {
        return "String is not a palindrome";
    }
}

@GetMapping("/palindrome/{word}")
public String Palindrome(@PathVariable String word)
{
    return assignmentService.Palindrome(word);
}

```

9.

```

public int getFactorial(int number) {
    int result=1;
    for(int i=1;i<number+1;i++)
    {
        result=result*i;
    }
    return result;
}

```

```
@GetMapping("/factorial/{number}")
public int getFactorial(@PathVariable int number)
{
    return assignmentService.getFactorial(number);
}
```

10.

```
@GetMapping("/prime/{num}")
public String isPrime(@PathVariable int num)
{
    return assignmentService.isPrime(num);
}
```

```
public String isPrime(int number) {  
  
    boolean isPrime=false;  
    int count=0;  
  
    for(int i=1;i<number/2+1;i++)  
    {  
        if(number%i==0)  
        {  
            count++;  
        }  
    }  
  
    if(count>=2)  
    {  
        isPrime=false;  
    }  
    else {  
        isPrime=true;  
    }  
  
    if(isPrime)  
    {  
        return "Prime Number";  
    }  
    else {  
        return "Not a Prime Number";  
    }  
}
```

```
public int discountCalculation(int price,int discount)
{
    int result=price-(price*discount/100);
    return result;
}
```

```
@GetMapping("/discount")
public int discountCalculation(@RequestParam int price,@RequestParam int discount)
{
    return assignmentService.discountCalculation(price, discount);
}
```

12.

```
@GetMapping("/login")
public String loginValidation(@RequestBody Map<String,String> body)
{
    return assignmentService.loginValidation(body.get("username"), body.get("password"));
}

public String loginValidation(String username,String password)
{
    if(username.equals("admin") && password.equals("admin123"))
    {
        return "Login Successfull";
    }
    else {
        return "Login Failed";
    }
}
```

13.

```
@GetMapping("/interest")
public int simpleInterest(@RequestParam int p,@RequestParam int r,@RequestParam int t)
{
    return assignmentService.simpleInterest(p, r, t);
}
```

```
public int simpleInterest(int p,int r,int t)
{
    int result=(p*r*t)/100;
    return result;
}
```

14.

```
}
```

```
public int wordCount(String sentence)
{
    String arr[]=sentence.split(" ");
    int result= arr.length;
    return result;
}
```

```
@PostMapping("/wordcount")
public int wordCount(@RequestBody Map <String, String> body)
{
    return assignmentService.wordCount(body.get("sentence"));
}
```

15.

```
public int max(int a,int b,int c)
{
    int result;
    if(a<b && b>c)
    {
        result=b;
    }
    else if(a<c && c>b)
    {
        result=c;
    }
    else {
        result=a;
    }
    return result;
}

@GetMapping("/max")
public int max(@RequestParam int a,@RequestParam int b,@RequestParam int c)
{
    return assignmentService.max(a, b, c);
}
```