**Angular Assessment**

**24-May-2024**

**Batch: EDS\_AMJ24\_Angular\_1**

Question 1: Create an angular web application to create a web form to accept the Customer data for an e-shopping application.

Solution:

I have used bootstrap cdn for my code. So please add the link to your index.html, so that the code will look exactly same according to the screenshots attached below:

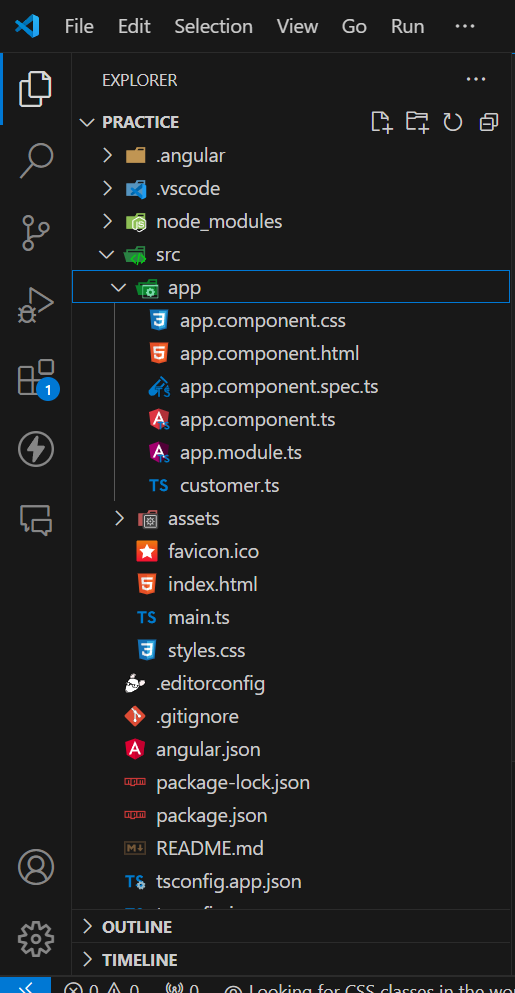
Boostrap CDN link:

  <link rel="stylesheet" href="

  https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"

    integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">

Please find the folder structure as per the codes:



Please find the attached codes according to the files:

app.module.ts

import { NgModule } from '@angular/core';

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

import { AppComponent } from './app.component';

import { FormsModule, ReactiveFormsModule } from '@angular/forms';

@NgModule({

  declarations: [

    AppComponent

  ],

  imports: [

    BrowserModule, FormsModule, ReactiveFormsModule

  ],

  providers: [],

  bootstrap: [AppComponent]

})

export class AppModule { }

Customer Model (customer.ts)

export class Customer{

    cust\_id?:number;

    cust\_name?:string;

    cust\_address?:string;

    cust\_orders?:number;

}

app.component.ts

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

import { FormBuilder, FormGroup, Validators } from '@angular/forms';

@Component({

  selector: 'app-root',

  templateUrl: './app.component.html',

  styleUrls: ['./app.component.css']

})

export class AppComponent implements OnInit{

  customerForm!: FormGroup;

  constructor(private fb: FormBuilder){}

  displayMessage: boolean = false;

  customerCategory?:string;

  ngOnInit(): void {

    this.customerForm = this.fb.group({

      customerId:['',[Validators.required, Validators.min(10000),Validators.max(99999)]],

      customerName: ['',[Validators.required,Validators.minLength(3),Validators.maxLength(30)]],

      customerAddress:['',[Validators.required]],

      customerOrders:['',[Validators.required,Validators.min(0),Validators.max(99999)]]

    });

  }

  //getters for the input form controls

  get Id(){

    return this.customerForm.get('customerId');

  }

  get Name(){

    return this.customerForm.get('customerName');

  }

  get Address(){

    return this.customerForm.get('customerAddress');

  }

  get Orders(){

    return this.customerForm.get('customerOrders');

  }

  onFormSubmit(){

    this.displayMessage = true;

    const val = this.Orders?.value;

    if(val==0){

      this.customerCategory = 'Inactive Customer';

    }

    else if(val>=1 && val<=9){

      this.customerCategory = 'Elite Customer';

    }

    else{

      this.customerCategory = 'Premium Customer';

    }

  }

}

app.component.html

<div class="container">

  <h1>Customer Form Validation</h1>

  <form [formGroup]="customerForm" (ngSubmit)="onFormSubmit()">

    <div class="form-group">

      <label>Customer Id</label>

      <input type="number" class="form-control" formControlName = "customerId" >

      <small class="text-danger" \*ngIf="Id?.invalid && Id?.touched">Id should be of 5 digit number</small>

    </div>

    <div class="form-group">

      <label>Customer Name</label>

      <input type="text" class="form-control" formControlName = "customerName" >

      <small class="text-danger" \*ngIf="Name?.invalid && Name?.touched">Name should be between 3 and 30 characters</small>

    </div>

    <div class="form-group">

      <label>Customer Address</label>

      <input type="text" class="form-control" formControlName = "customerAddress" >

      <small class="text-danger" \*ngIf="Address?.invalid && Address?.touched">Addres is a required field</small>

    </div>

    <div class="form-group">

      <label>Customer Orders</label>

      <input type="number" class="form-control" formControlName = "customerOrders" >

      <small class="text-danger" \*ngIf="Orders?.invalid && Orders?.touched">Orders should be between 0 and 99999</small>

    </div>

    <button class="btn btn-success" [disabled]="customerForm.invalid">Submit Data</button>

  </form>

</div>

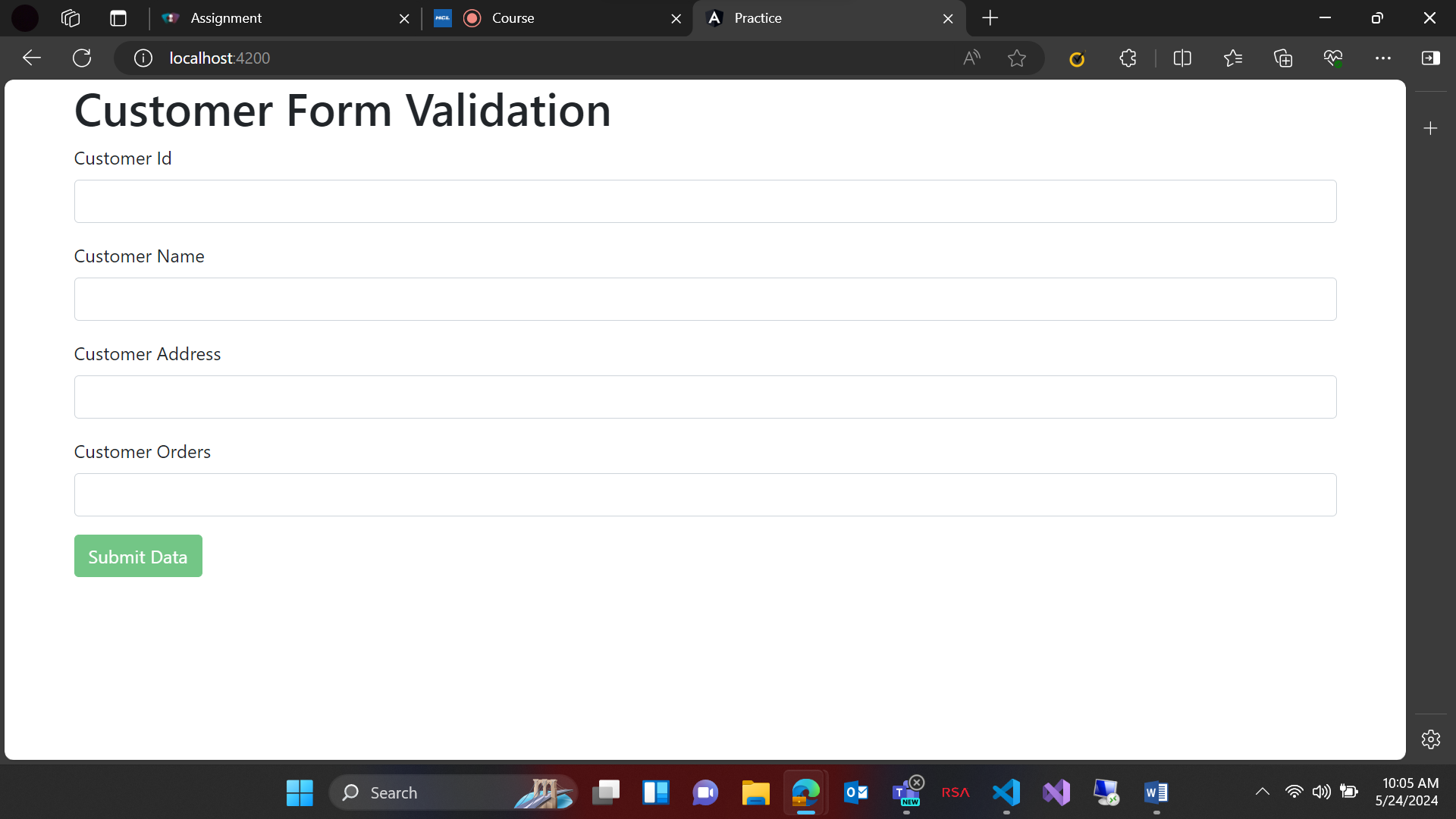
<div class="container" \*ngIf="displayMessage">

  <h3>Customer Category: {{customerCategory}}</h3>

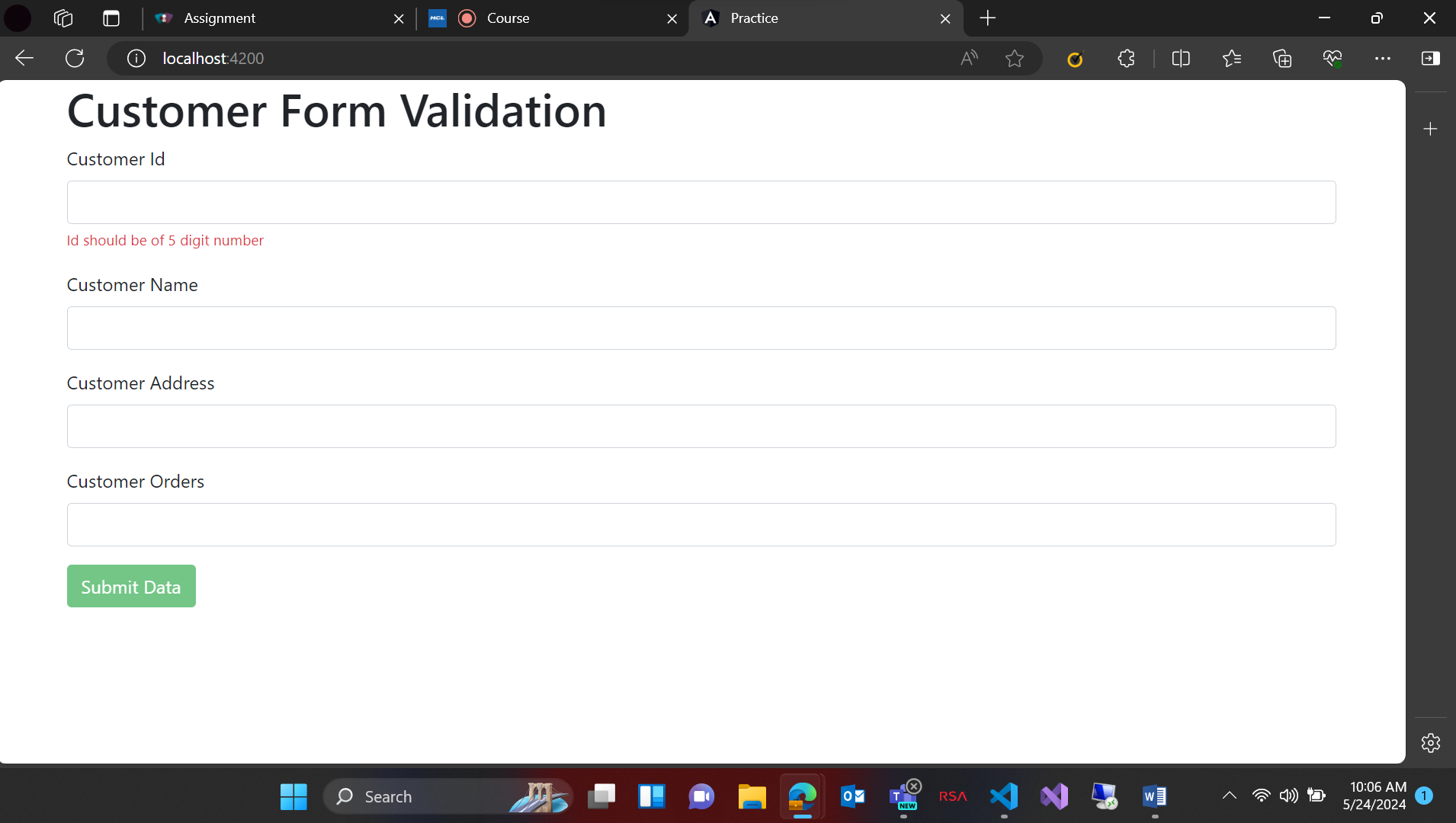
</div>

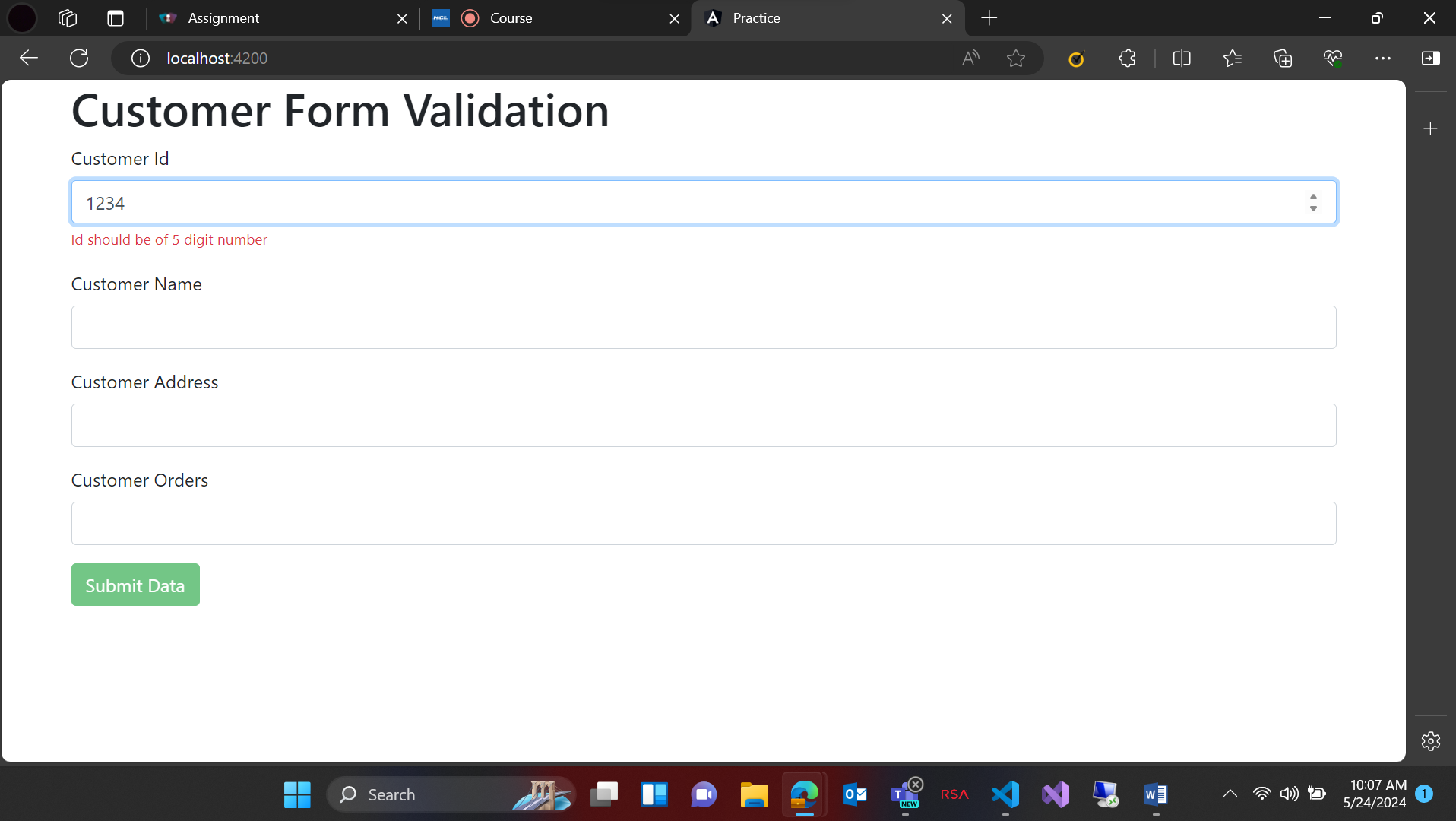
Please find the attached Output Screenshot for each scenario:

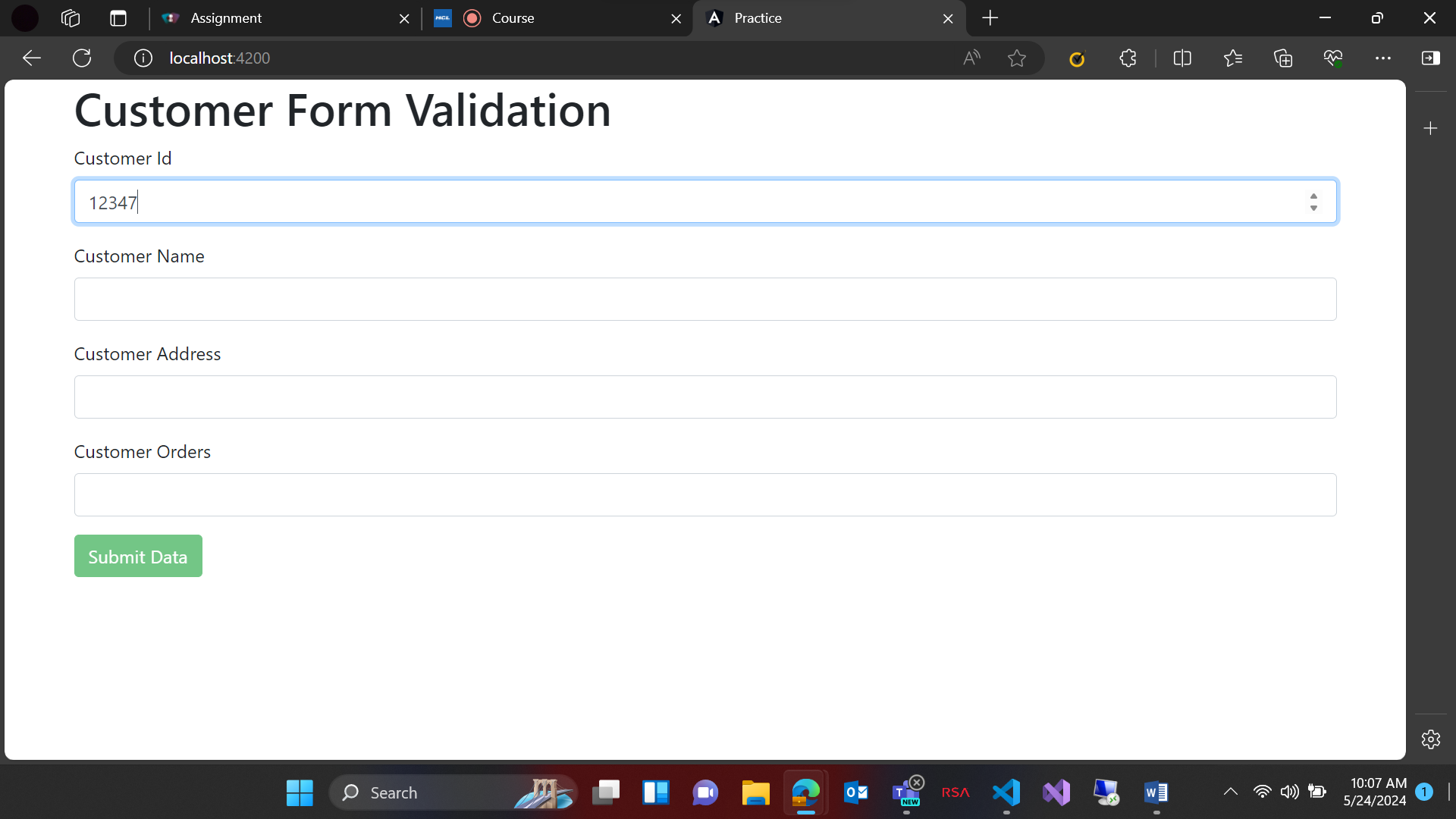
Output(view) of the code

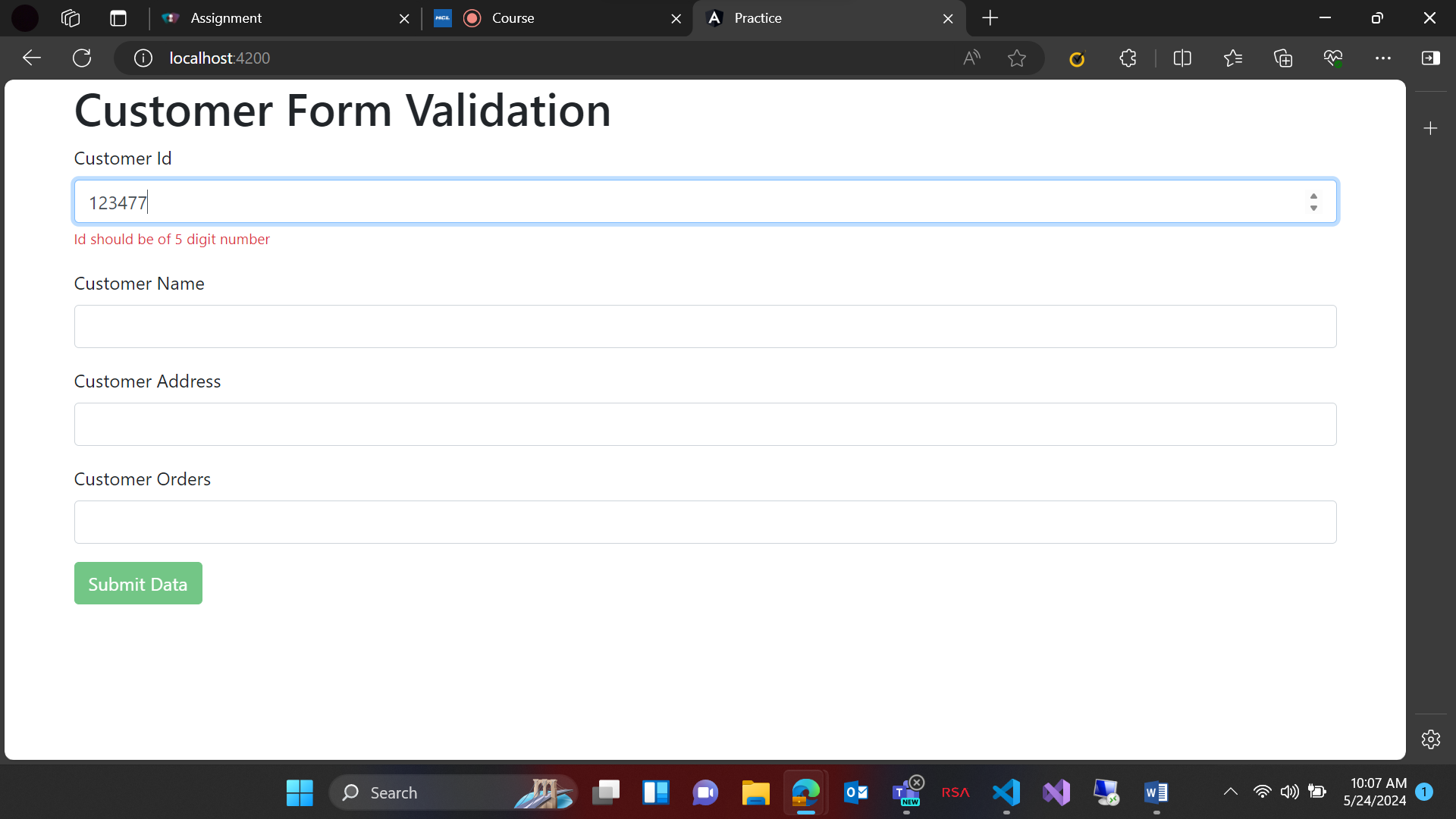


Id validation:

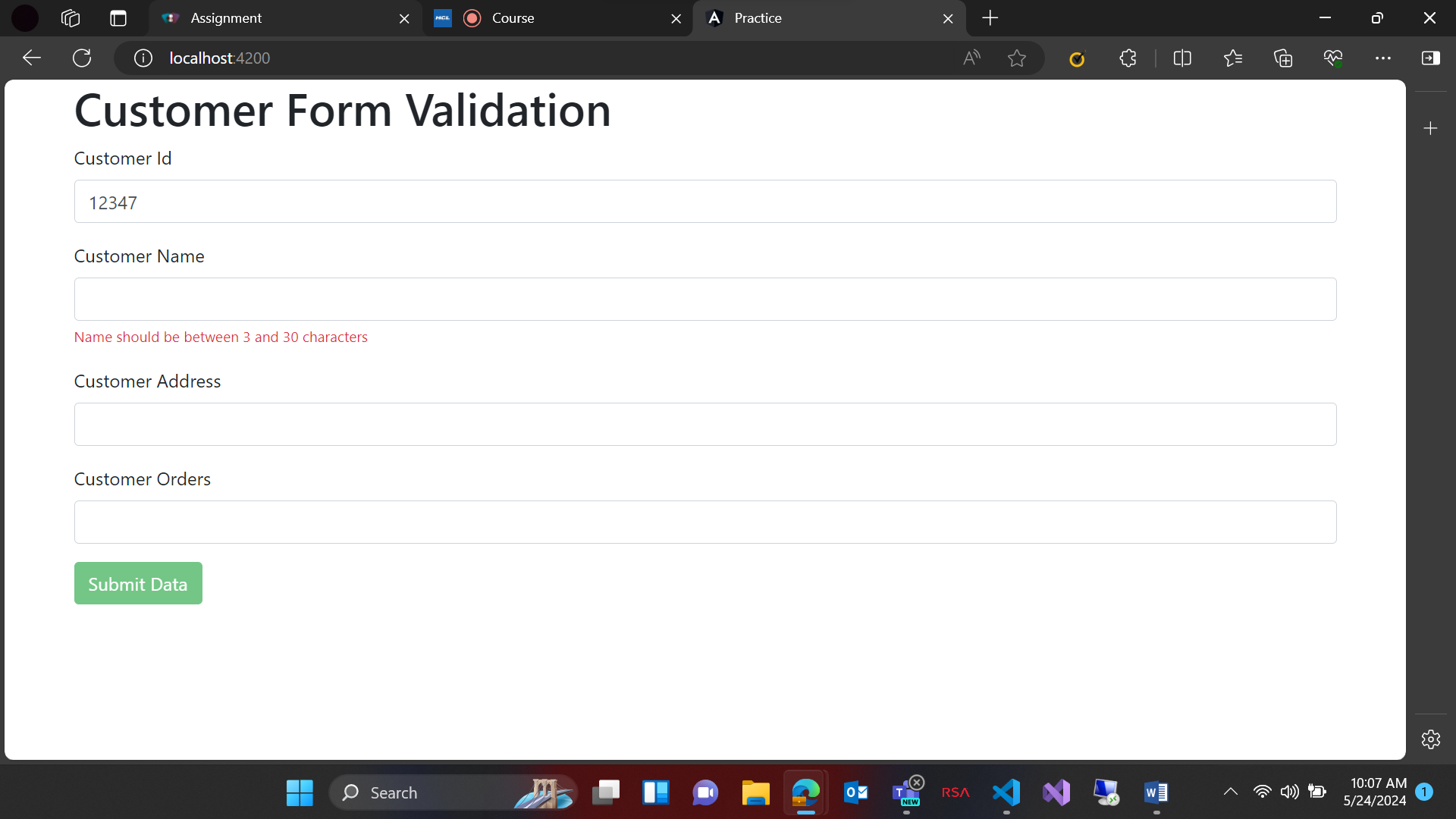


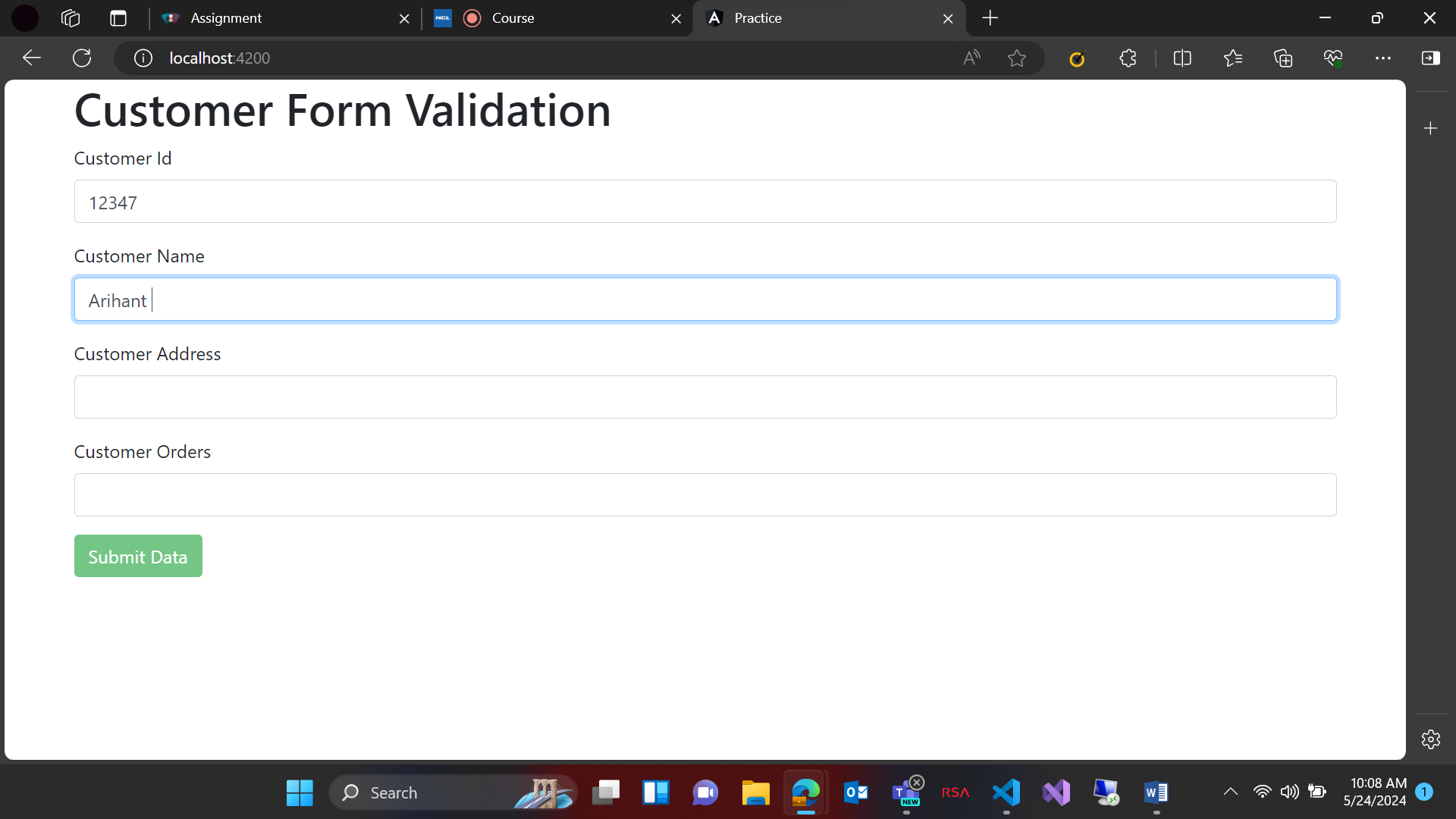


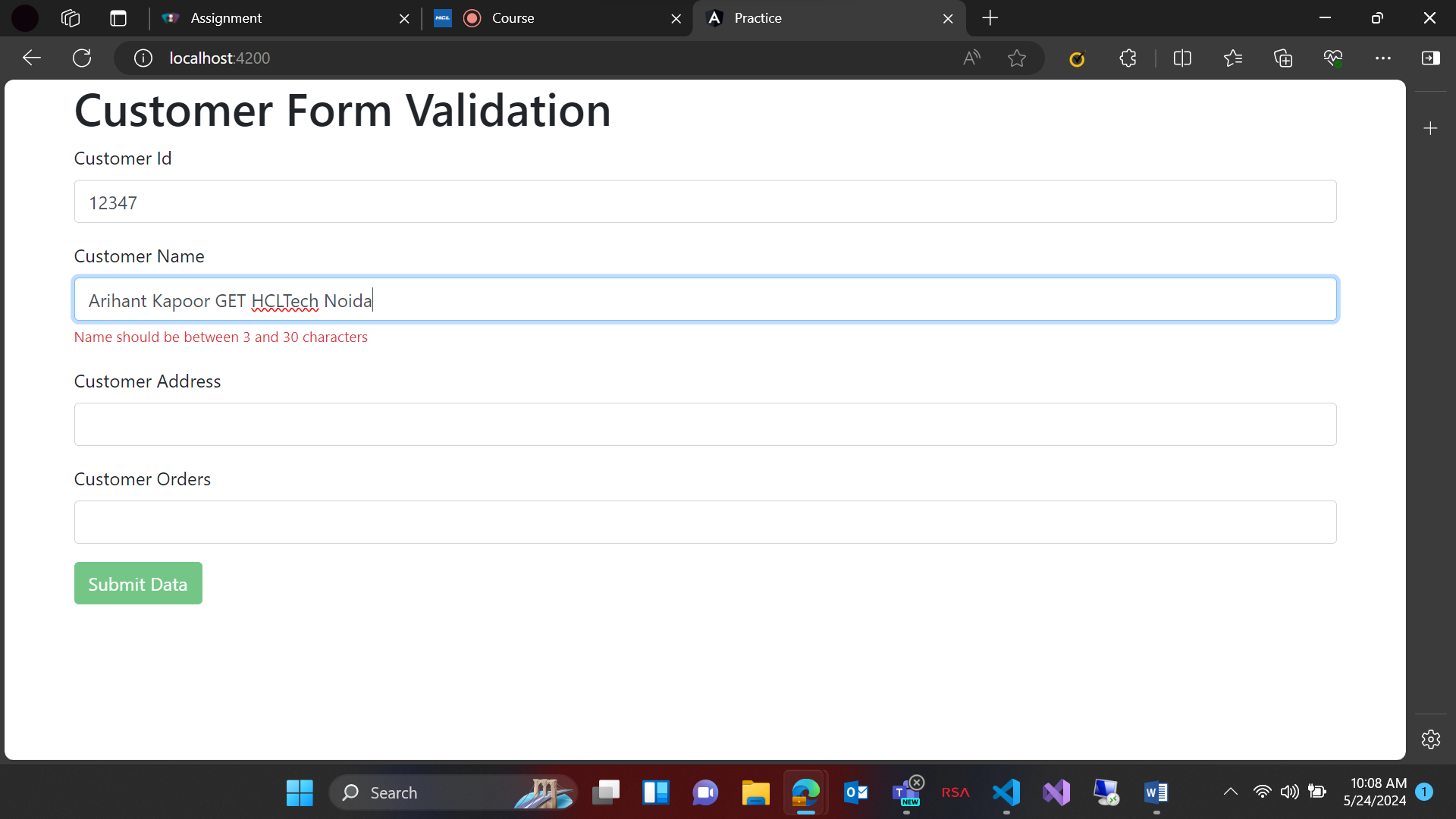




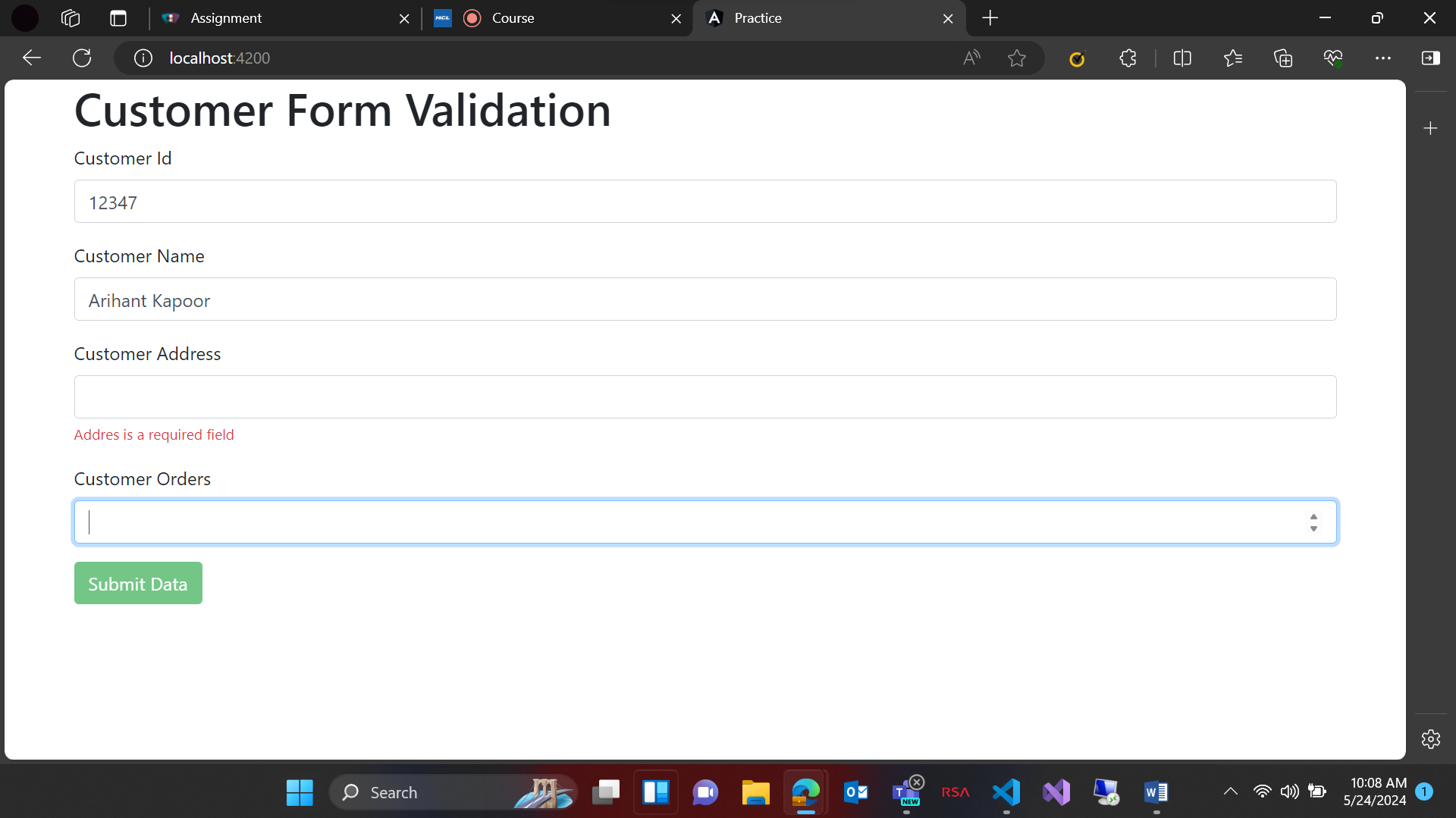
Name validation:

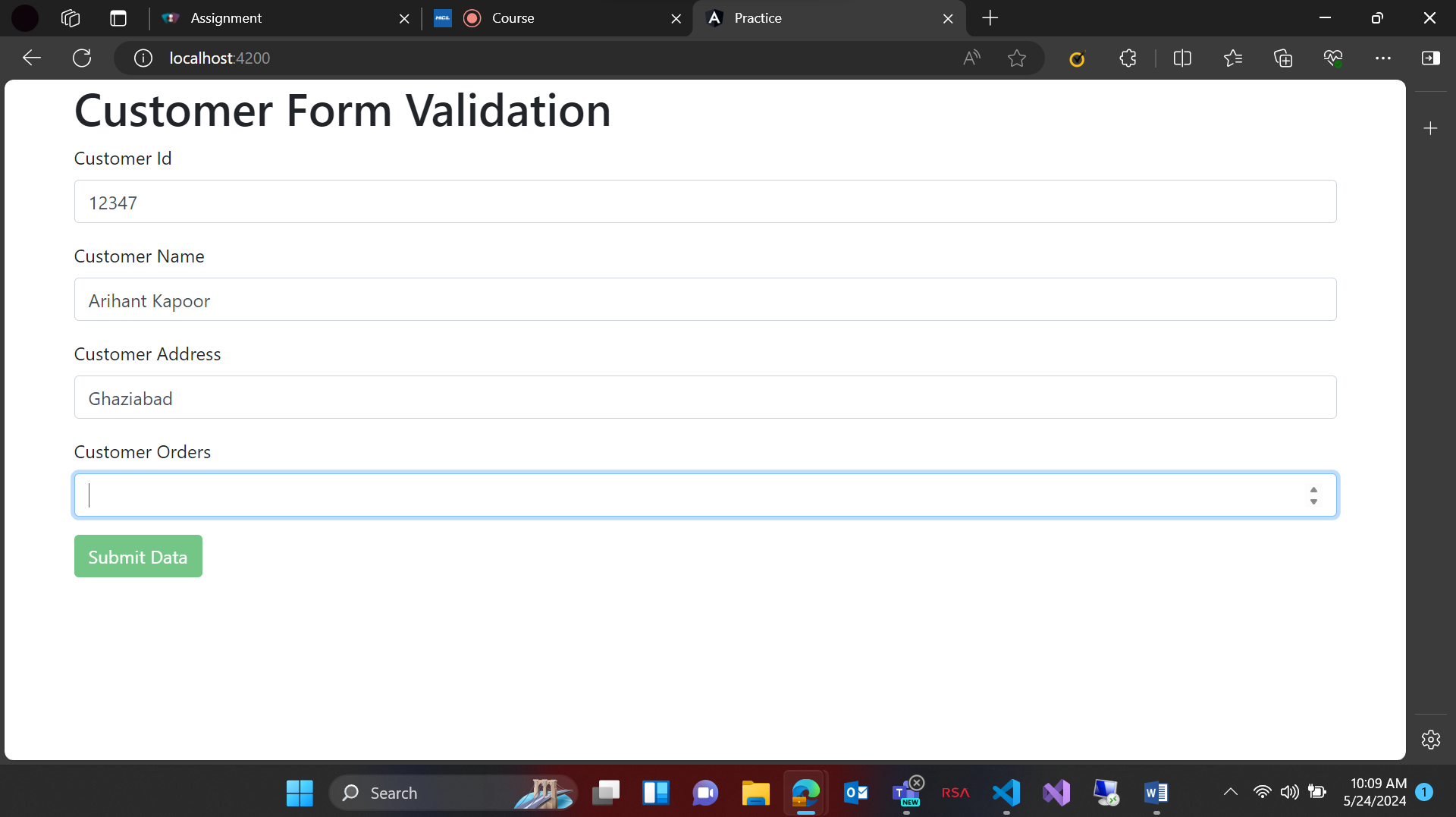






Address validation:





Orders Validation:

