EXP.NO. :

DATE :

TEMPLATE DRIVEN FORM VALIDATION IN ANGULAR

Aim:

To develop a web page for validating the template driven form in angular.

Description:

Template-driven forms use two-way data binding to update the data model in the component as changes are made in the template and vice versa.

Procedure:

- Create an angular project using "ng new appname" command.
- Open the application using “ng serve –o”.
- Open the application and create a new component using “ng g c component_name”.
- The new component will be created inside the app folder of src.
- To open show our component inside the app give the component name in app.component.html file.
- Then, to import the FormsModule in app.module.ts to use forms in the component.
- Design the webpage using html tags.
- Do the logic for validation in ts file.

Functions/Directives Used:

Function/Directive	Syntax	Description
ngModel	[(ngModel)]=’var_name’	The ngModel directive binds an input,select, textarea (or custom form control) to a property on the scope using NgModelController,
submit()	(submit)=”function_name	This will be used to call a function in

	($props)”	the ts file while submitting the form.
ngIf	*ngIf="condition"	A structural directive that conditionally includes a template based on the value of an expression coerced to Boolean.

States of Forms:

- $pristine No fields have been modified yet
- $dirty One or more have been modified
- $invalid The form content is not valid
- $valid The form content is valid
- $submitted The form is submitted

Program:

tempdrive.component.html:

```
<form #myform="ngForm" (submit)="Submitting(myform)">
   <h1>Template Driven form in Angular</h1>

   <input type="text" id="t1" name="email" placeholder="Enter Email Id"
      pattern = "^[a-zA-Z0-9+_.-]+@[a-zA-Z0-9.-]+$" #email="ngModel" [(ngModel)]="mail">
   <br>
   <div *ngIf="!email?.valid && (email?.dirty || email?.touched)">
   <div *ngIf="email.errors?.['pattern']">
      Invalid Email Address
   </div>
   </div>

   <input type="password" id="t2" minlength="8" maxlength="16"
   [(ngModel)]="pass" #pword="ngModel" name="pw" placeholder="Enter password" required>
   <br>
   <div *ngIf="!pword?.valid && (pword?.dirty || pword?.touched)">
   <div *ngIf="pword.errors?.['minlength']">
   Password Minimum Length is 8
   </div>
   </div>
   <button id="bt" type="submit">Login</button>
   <br>
  </form>
```

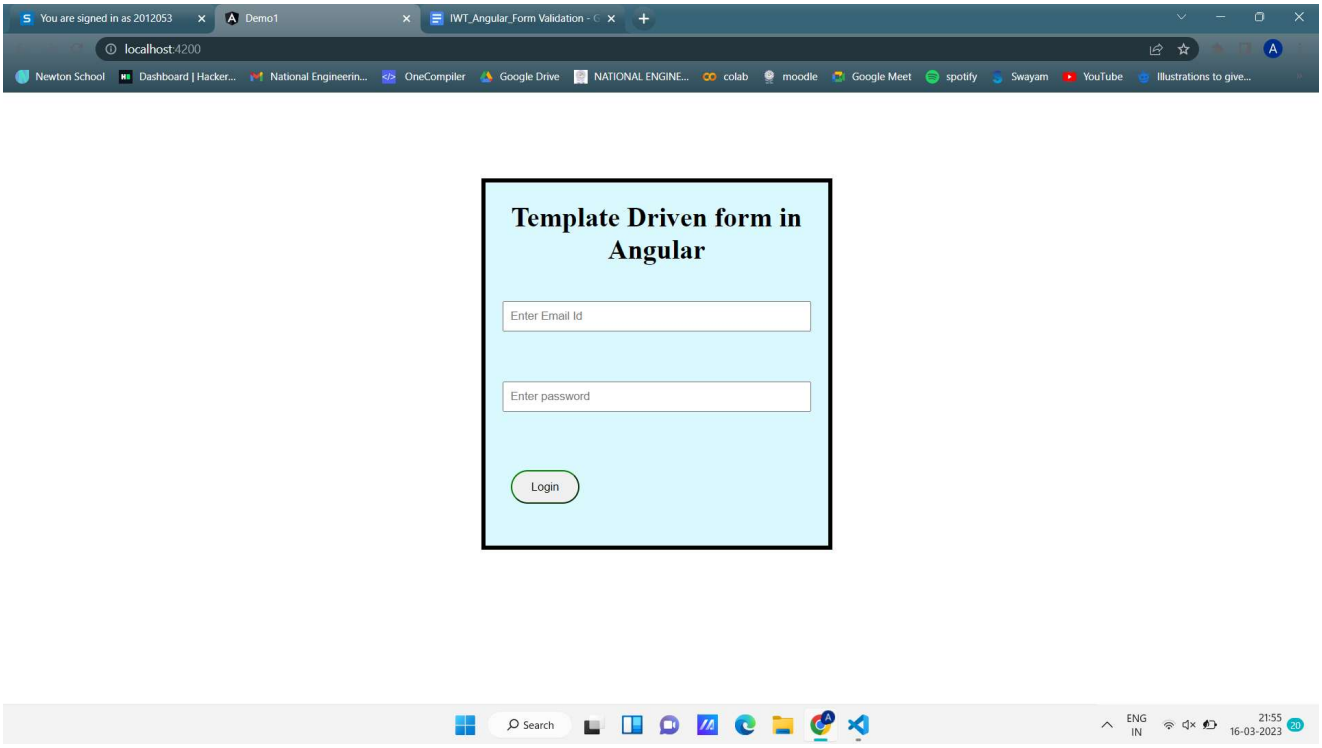
tempdrive.component.css:

```
form{
    display: flex;
    flex-direction: column;
    margin: 100px;
    margin-left: 550px;
    align-content: center;
    width:400px;
    text-align: center;
    justify-content: center;
    border: 5px solid black;
    width: 400px;
    background: rgb(218, 249, 255);
   }
   div{
    color: red;
    font-size: 20px;
   }

   #t1
   {
    margin: 20px;
    padding: 8px;
   }
   #t2
   {
    margin: 20px;
    padding: 8px;
   }
   #bt
   {
    margin: 30px;
    padding: 10px;
    border-radius: 5px;
    border-radius: 80px;
    border-color: green;
    width:80px;
   }
```

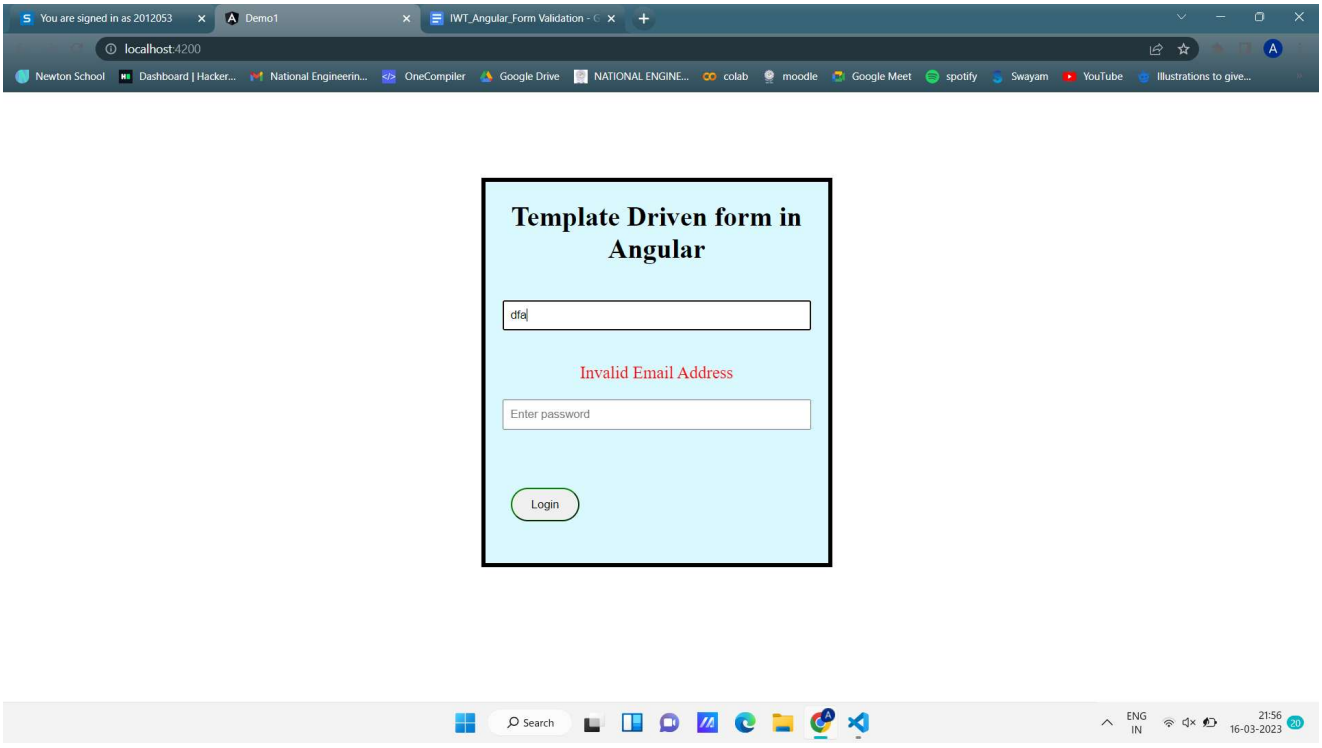
tempdrive.component.ts:

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

@Component({
  selector: 'app-tempdrive',
  templateUrl: './tempdrive.component.html',
  styleUrls: ['./tempdrive.component.css']
})
export class TempdriveComponent implements OnInit {

  mail=""
  pass=""

  constructor() { }

  ngOnInit(): void {
  }

  Submitting(myForm:any){
    if(myForm.value.email!="" && myForm.value.pw!="")
    {
    alert("Success!");
    }
    else
    {
    alert("Enter all the details");
    }

  }
}
```

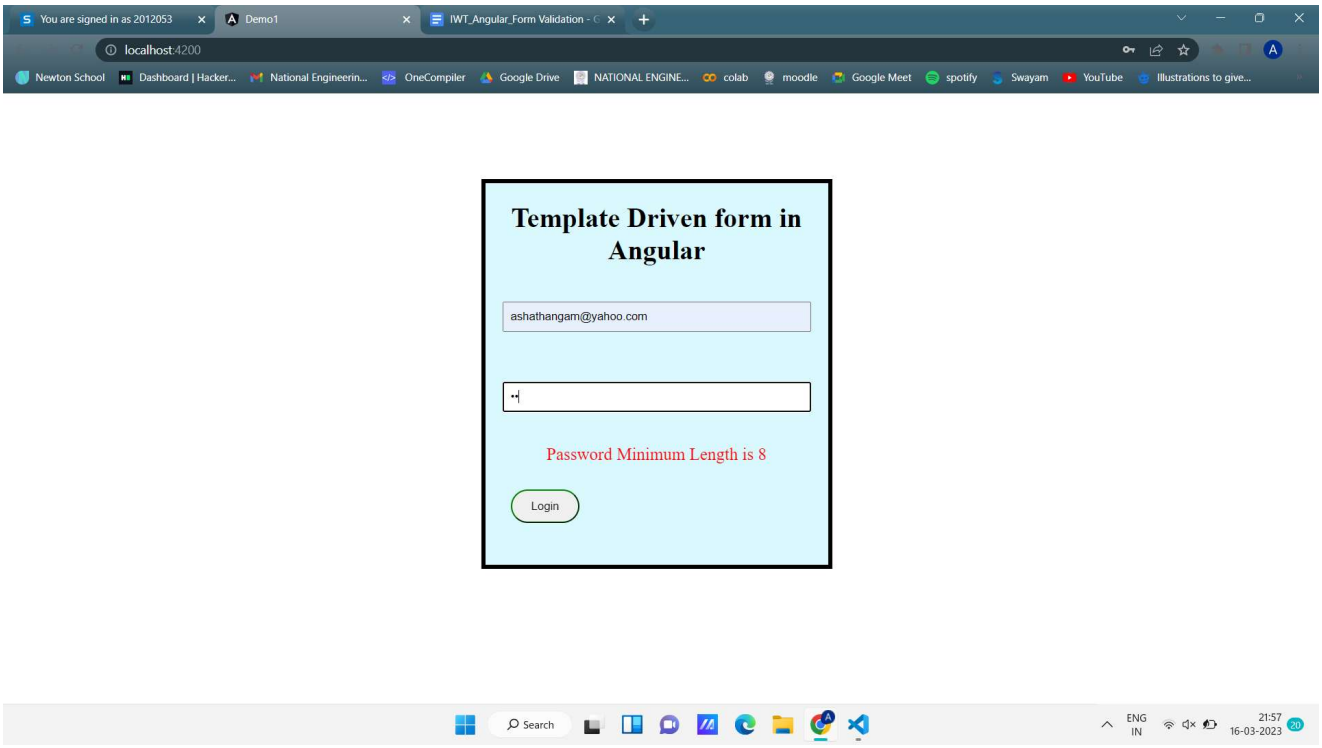
Output:



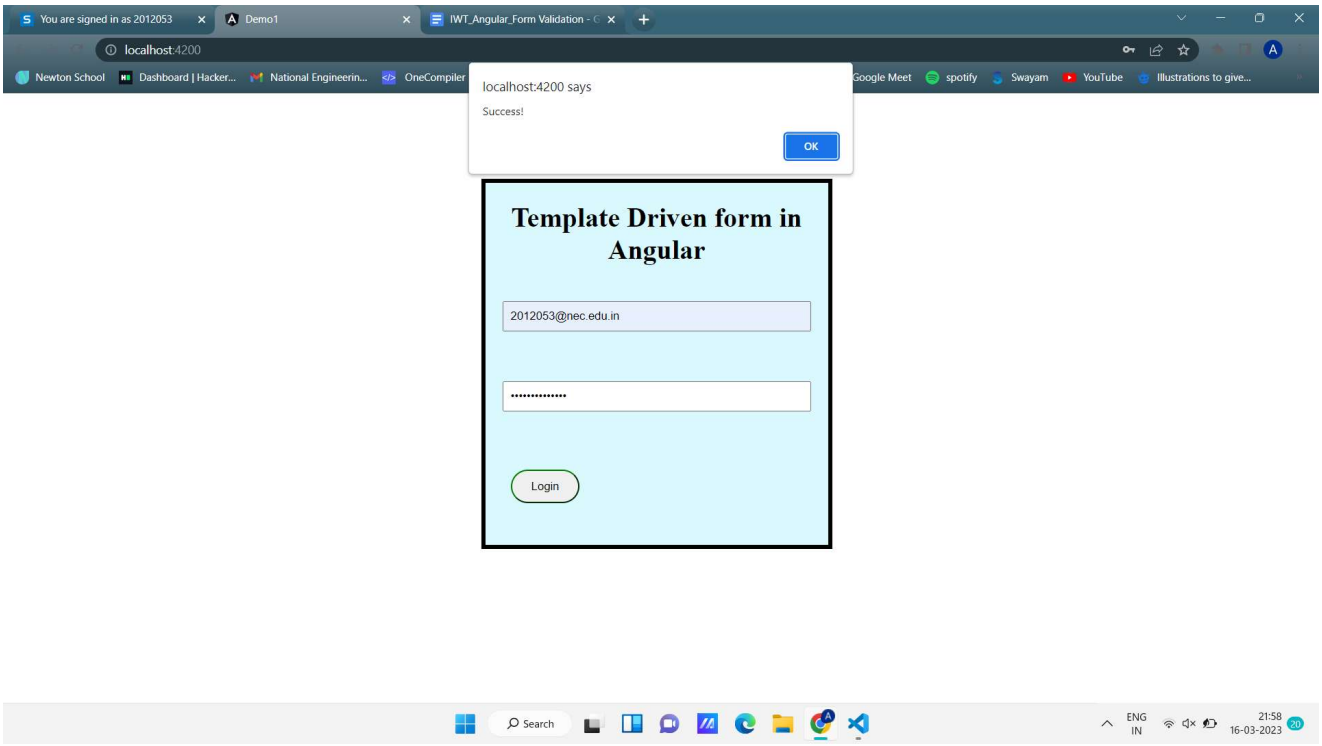
When User enters invalid mail id:



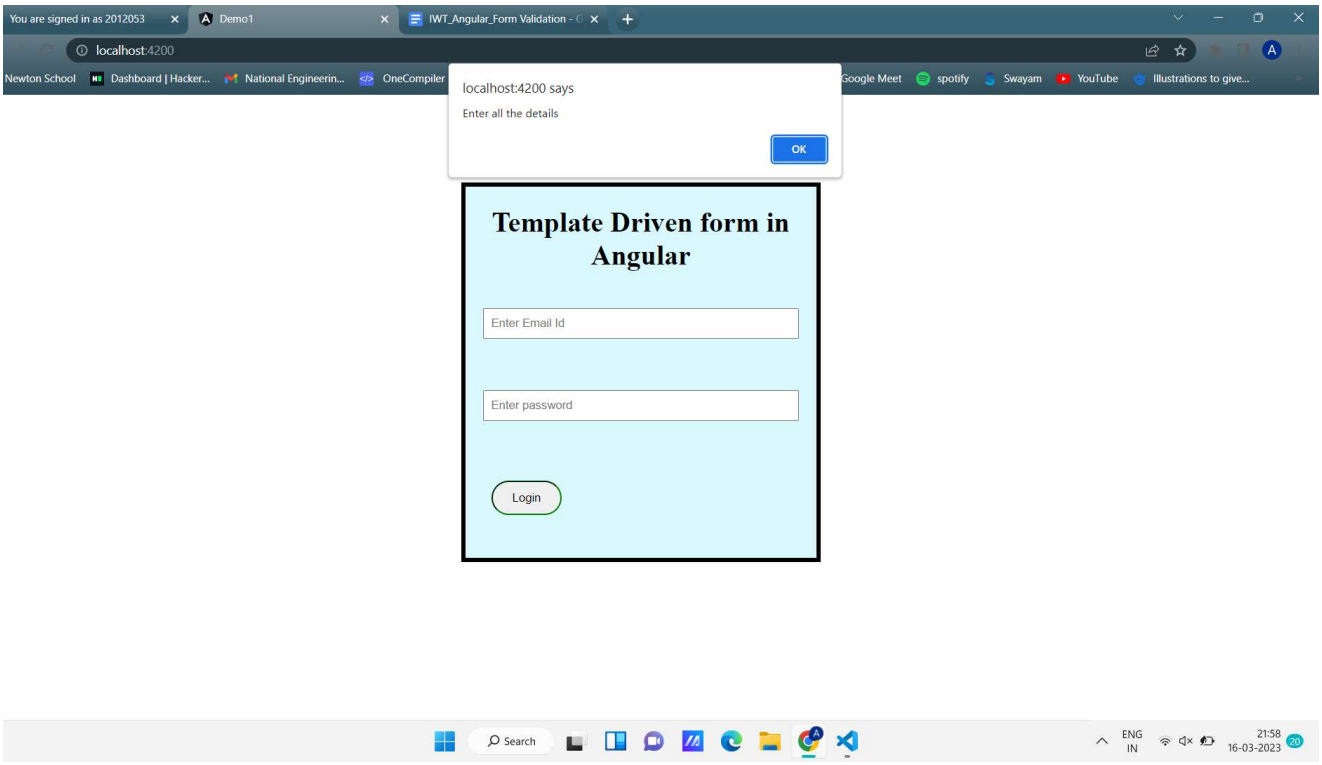
When the password length is short:



When both fields are correct:



When the user not entered input fields:



Problem Understanding	Implementation & Output	Viva	Time Management	Total

Result:

Thus the template-driven form validation has been implemented successfully and output was verified.