Template Driven Forms lab Manual

Create a new project

Ng new ngforms

Cd ngforms

Ng serve –open

Install bootstrap

* Npm install boostrap

Step:1 import forms module in rootmodeule

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

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

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

import { AppRoutingModule } from './app-routing.module';

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

@NgModule({

  declarations: [

    AppComponent

  ],

  imports: [

    BrowserModule,

    AppRoutingModule,

FormsModule

  ],

  providers: [],

  bootstrap: [AppComponent]

})

export class AppModule { }

Go to app.component.html

And

Type

B4$ and press tab we get all booststrap ready made

App.component.html

<!doctype html>

<html lang="en">

<head>

    <title>Title</title>

    <!-- Required meta tags -->

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bootstrap CSS -->

    <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">

</head>

<body>

    <!-- Optional JavaScript -->

    <!-- jQuery first, then Popper.js, then Bootstrap JS -->

    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>

    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1" crossorigin="anonymous"></script>

    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>

    <br/>

    <br/>

    <div class="container">

        <div class="row">

            <div class="form-bg">

                <form #studentForm="ngForm" (ngSubmit)="RegisterStudent(studentForm)">

                    <div class="panel panel-primary">

                        <div class="panel-heading">

                            <h3 class="panel-title">Student Registration</h3>

                        </div>

                        <div class="panel-body">

                            <div class="form-group">

                                <label for="firstName">First Name</label>

                                <input id="firstName" type="text" class="form-control" name="firstName" ngModel>

                            </div>

                            <div class="form-group">

                                <label for="lastName">Last Name</label>

                                <input id="lastName" type="text" class="form-control" name="lastName" ngModel>

                            </div>

                            <div class="form-group">

                                <label for="email">Email</label>

                                <input id="email" type="text" class="form-control" name="email" ngModel>

                            </div>

                            <div class="form-group">

                                <label for="Gender">Gender</label><br>

                                <input type="radio" name="gender" value="male" checked [(ngModel)]="gender">Male

                                <input type="radio" name="gender" value="female" [(ngModel)]="gender">Female </div>

                            <div class="form-group">

                                <div class="form-controlf">

                                    <label class="checkbox-inline"></label>

                                    <input type="checkbox" name="isaccepted" ngModel>AcceptTerms and Condition</div>

                            </div>

                            <div class="form-group">

                                <div class="form-control">

                                    <select name="branch" id="branch" ngModel class="form-control">

                                        <!-- <option value="1">ComputerScience</option>

                                        <option value="2">Mechanical</option>

                                        <option value="3">ai and ml</option>

                                        <option value="4">Electrical and Electronics</option>

                                        <option value="5">Electronics and Communication</option> -->

                                     <option \*ngFor="let branch of Branches" [value]="branch.name">

                                        {{branch.name}}

                                      </option>

                                    </select>

                                </div>

                            </div>

                        </div>

                        <div class=" panel-footer ">

                            <button class="btn btn-primary " type="submit ">Submit</button>

                        </div>

                    </div>

                </form>

            </div>

        </div>

    </div>

</body>

</html>

**Appcomponent.html.ts**

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

import { NgForm } from '@angular/forms';

@Component({

  selector: 'app-root',

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

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  title = 'ngforms';

   gender:string="";

  RegisterStudent(studentForm:NgForm):void{

    var firstname=studentForm.controls['firstName'].value;

    var lastname=studentForm.controls['lastName'].value;

    var email=studentForm.controls['email'].value;

    this. gender=studentForm.controls['gender'].value

    var isaccepted=studentForm.controls['isaccepted'].value;

    console.log(studentForm.value);

  }

Generally we get data from backend but we are writing here in a arry and binding to a dropdown

  Branches:any[]=[

    {id:1,name:'computerscience'},

    {id:3,name:'Mechanical'},

    {id:3,name:'Ai amd ML'},

    {id:4,name:'Electrical and Electronics'},

    {id:5,name:'Electronics and Communication'}

  ]

}

**Angular 11/10 Multi Select Dropdown Example**

By Hardik Savani August 31, 2020 Category : Angular

Video Player is loading.

Pause

Unmute

Loaded: 0.51%

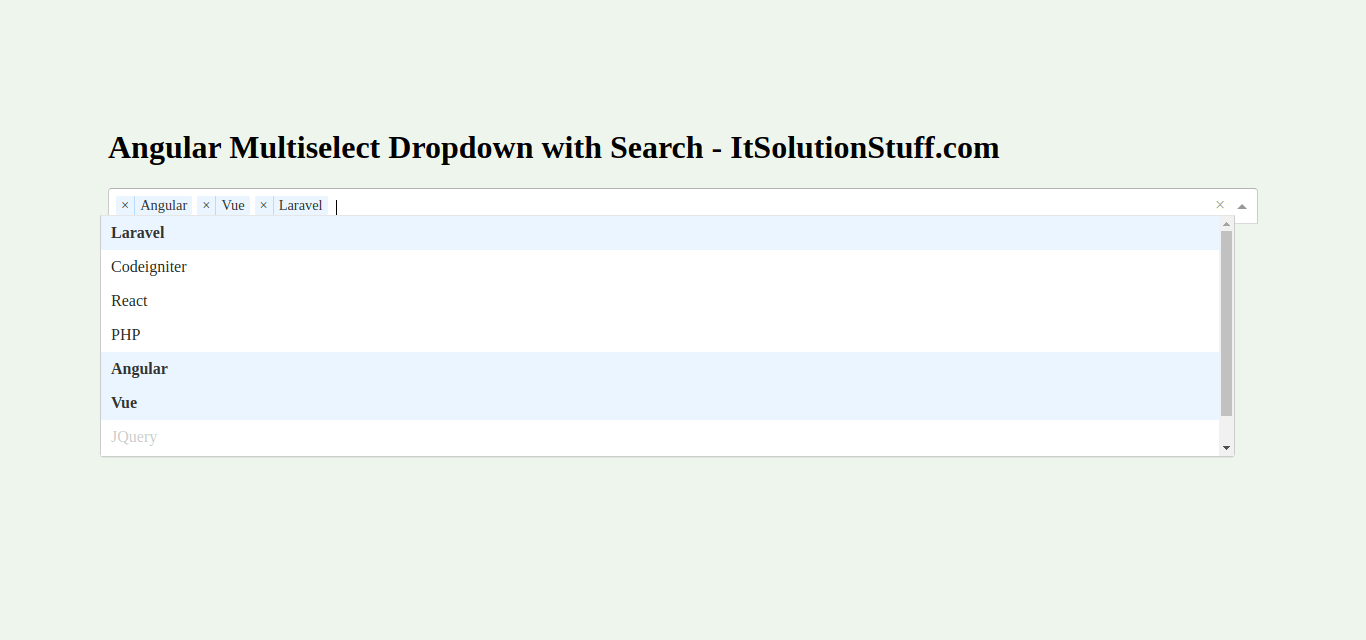
Fullscreen

In this tute, we will discuss angular 10 multiple select dropdown. step by step explain angular 11/10 multi select dropdown. you can understand a concept of multi select dropdown in angular 10. if you have question about how to use multi select dropdown in angular 10 then i will give simple example with solution.

If you are looking for good example of multiselect dropdown in angular 10 app then i will help you how to use multi select dropdown in angular 10 application. we will use ng-select npm package for multiselect dropdown in angular. they provide set methods to give dropdown option and change events to getting selected option values etc.

Here, we will look step by step example how to use multiselect dropdown in angular 8/9 application.

So, let's see very simple step and get it very simple example here:



Step 1: Create New App

You can easily create your angular application using bellow command:

ng new myMultiselect

Step 2: Install Npm Packages

In this step, we will install @ng-select/ng-select npm package for creating chart using multi select dropdown in angular 8/9. so let's run bellow command:

npm install --save @ng-select/ng-select

Read Also: [Angular 10 Reactive Forms Validation Example](https://www.itsolutionstuff.com/post/angular-10-reactive-forms-validation-exampleexample.html)

Step 3: Import NgSelectModule

Now, here we will import NgSelectModule from ng-select and then we add on declarations part. so let's update app.module.ts file as like bellow:

**src/app/app.module.ts**

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

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

**import** { FormsModule } **from** '@angular/forms';

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

**import** { NgSelectModule } **from** '@ng-select/ng-select';

@NgModule({

imports: [ BrowserModule, FormsModule, NgSelectModule ],

declarations: [ AppComponent ],

bootstrap: [ AppComponent ]

})

**export** **class** AppModule { }

Step 4: Import CSS File

now in this step, we will import ng-select theme css file. so we can get multi select dropdown box design so let's import in styles.css file.

**src/styles.css**

/\* Add application styles & imports to this file! \*/

@import "~@ng-select/ng-select/themes/default.theme.css";

Step 5: Update Ts File

Here, we will update app.component.ts file here, in this file we will take "categories" array with list of category so we can create dropdown box option. we will also create "selected" array with give default selected option if you need then. Another we will create getSelectedValue() that call on click event and get the selected values.

You can update as bellow app.component.ts file.

**src/app/app.component.ts**

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

@Component({

selector: 'my-app',

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

styleUrls: [ './app.component.css' ]

})

**export** **class** AppComponent {

name = 'Angular';

categories = [

{id: 1, name: 'Laravel'},

{id: 2, name: 'Codeigniter'},

{id: 3, name: 'React'},

{id: 4, name: 'PHP'},

{id: 5, name: 'Angular'},

{id: 6, name: 'Vue'},

{id: 7, name: 'JQuery', disabled: **true**},

{id: 8, name: 'Javascript'},

];

selected = [

{id: 5, name: 'Angular'},

{id: 6, name: 'Vue'}

];

getSelectedValue(){

console.log(**this**.selected);

}

}

Step 6: Update Layout File

Here, we will update html file as like bellow, so update it as like bellow:

**src/app/app.component.html**

**<h1>**Angular Multiselect Dropdown with Search - ItSolutionStuff.com**</h1>**

**<ng-select** [**items**]="categories"

**bindLabel**="name"

**placeholder**="Select Category"

**appendTo**="body"

[**multiple**]="true"

[(**ngModel**)]="selected"**>**

**</ng-select>**

**<button** (**click**)="getSelectedValue()"**>**Get Selected Values**</button>**

Now you can run angular 8 app:

## ****DropDownList in Angular Template Driven Forms****

In this article, I am going to discuss **DropDownList in Angular Template Driven Forms** in detail. Please read our previous article as it is a continuation part to that article where we discussed [**Checkbox in Angular Template Driven Forms**](https://dotnettutorials.net/lesson/checkbox-angular-template-driven-forms/). At the end of this article, you will understand what is DropDownList and when and how to use DropDownList in Angular Template Driven Forms.

##### **What is a DropDownList?**

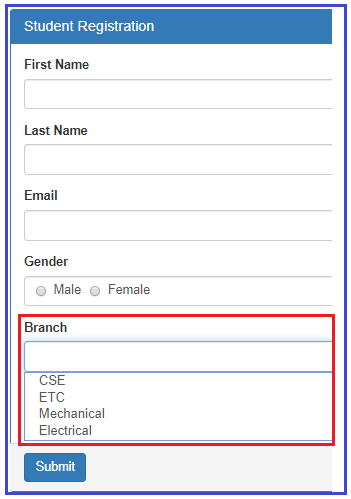
A DropDownList is an HTML Element which is nothing but a collection of list items from which it will allows the user to select a single list item. Depending on your business requirement you may either hard code the values or you may retrieve the values from a database table.

In this article, I am going to discuss both the approaches. First, we will discuss creating the DropDownList using the hard-coded value then we will see how to create the DropDownList with the values coming from a component.

##### **Example to understand DropDownList in Angular Template Driven Forms:**

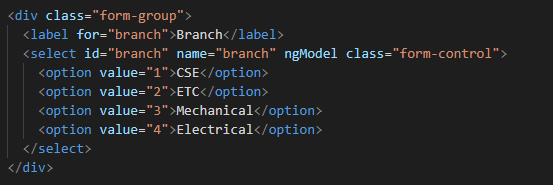
Let us understand how to create and use **DropDownList**in Angular Template Driven Forms with an example. We are going to work with the same example that we worked in our previous article.

Now, we want to include Branches dropdownlist in the student registration form as shown in the below image. When the user select a particular branch from the dropdownlist and click on the “Submit” button, then we want to display the selected drop down list value on the console.



##### **How to create the dropdownlist in angular using template driven forms?**

Please have a look at the following code which will create a Drop Down List with the required items.



##### **Code Explanation**

As shown in the code, we have hard coded the drop down list options in the HTML. Notice each option also has a corresponding value attribute and its value is the branch id which is what we want to save in the database table when the form is submitted. We will discuss, saving the data to a database table in our upcoming article.

##### **The complete code of app.component.html:**

Following is the complete code of **app.component.htm**l file.

**<br/>**

**<div** class="container"**>**

**<div** class="row"**>**

**<div** class="form-bg"**>**

**<form** #studentForm="ngForm" (ngSubmit)="RegisterStudent(studentForm)" **>**

**<div** class="panel panel-primary"**>**

**<div** class="panel-heading"**>**

**<h3** class="panel-title"**>**Student Registration**</h3>**

**</div>**

**<div** class="panel-body"**>**

**<div** class="form-group"**>**

**<label** for="firstName"**>**First Name**</label>**

**<input** id="firstName" type="text" class="form-control"

name="firstName" ngModel**>**

**</div>**

**<div** class="form-group"**>**

**<label** for="lastName"**>**Last Name**</label>**

**<input** id="lastName" type="text" class="form-control"

name="lastName" ngModel**>**

**</div>**

**<div** class="form-group"**>**

**<label** for="email"**>**Email**</label>**

**<input** id="email" type="text" class="form-control"

name="email" ngModel**>**

**</div>**

**<div** class="form-group"**>**

**<label>**Gender**</label>**

**<div** class="form-control"**>**

**<label** class="radio-inline"**>**

**<input** type="radio" name="gender" value="male" ngModel**>**

Male

**</label>**

**<label** class="radio-inline"**>**

**<input** type="radio" name="gender" value="female" ngModel**>**

Female

**</label>**

**</div>**

**</div>**

**<div** class="form-group"**>**

**<label** for="branch"**>**Branch**</label>**

**<select** id="branch" name="branch" ngModel class="form-control"**>**

**<option** value="1"**>**CSE**</option>**

**<option** value="2"**>**ETC**</option>**

**<option** value="3"**>**Mechanical**</option>**

**<option** value="4"**>**Electrical**</option>**

**</select>**

**</div>**

**<div** class="form-group"**>**

**<div** class="form-control"**>**

**<label** class="checkbox-inline"**>**

**<input** type="checkbox" name="isAccept" ngModel**>**Accept Terms & Conditions

**</label>**

**</div>**

**</div>**

**</div>**

**<div** class="panel-footer"**>**

**<button** class="btn btn-primary" type="submit"**>**Submit**</button>**

**</div>**

**</div>**

**</form>**

**</div>**

**</div>**

**</div>**

##### **Modifying app.component.ts file:**

We want to log the posted form values into the console tab. So, modify the **app.component.ts**file as shown below.

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

**import** **{** NgForm **}** from '@angular/forms'

@Component**({**

selector: 'app-root',

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

styleUrls: **[**'./app.component.css'**]**

**})**

**export** **class** AppComponent **{**

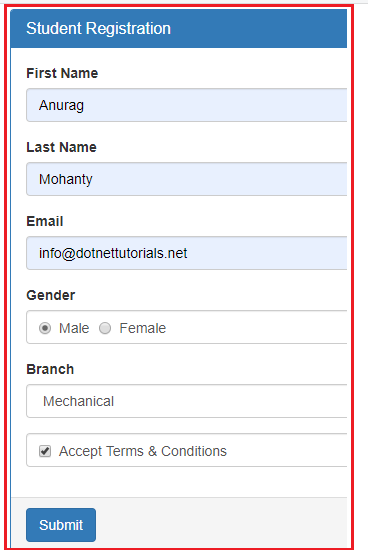
RegisterStudent**(**studentForm: NgForm**)**: **void** **{**

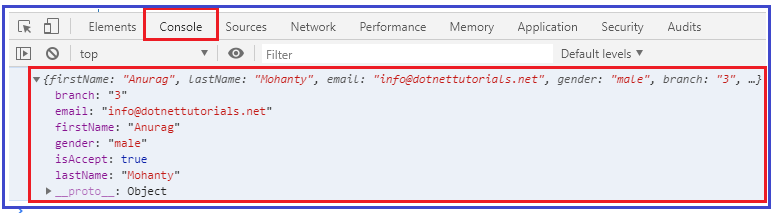
console.log**(**studentForm.value**)**;

**}**

**}**

Save the changes and browse the application, then open browser developers tool by pressing F12 key and click on the console tab. Fill the form, select one value from the drop down list and click on the submit button and you should see the posted form values in the console tab as shown in the below image.

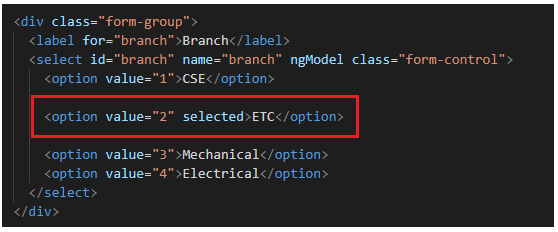




##### **How to have one of the dropdownlist item selected by default in Angular?**

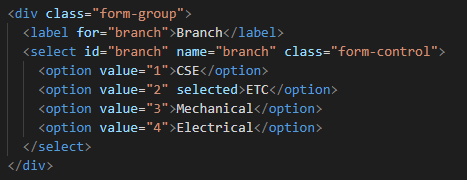
As we know when working with real-time applications, many a times we need to provide one option to be selected in the drop down list by default when the form load. And we normally do this by adding the **selected**attribute on one of the option of drop down list.

If we include the **selected**attribute on the **dropdownlist**, then we may expect that option or item to be selected by default. But in angular template driven forms, that will not work. Lets include the “**selected**” attribute on the ETC branch option to verify this. So. Modify the Drop Down List HTML code as shown below.



With the above changes in place, now browse the application and you will see that the ETC department is not selected by default when the page load.

However, if you remove the “ngModel” directive from the select list as shown below, then you will see that the ETC branch is selected when the form is load.



As we already discussed, we use the “ngModel” directive in angular for two-way data binding. So when we put the ngModel directive back into the control then the “**selected**” attribute will not work on the drop down list or select list. If we remove the ngModel directive from the control then **selected**attribute work but two way data binding will not work.

##### **How to make it works?**

In order to make it work, what we need to do is, we need to add a property lets say “BranchId” in the component class and initialize its value with the branch value that you want to be selected when the page load. As we want ETC branch to be selected by default and as its value is 1, so, we need to BranchId property and initialize its value to 2 in the AppComponent class. So, modify the **app.component.ts** file as shown below.

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

**import** **{** NgForm **}** from '@angular/forms'

@Component**({**

selector: 'app-root',

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

styleUrls: **[**'./app.component.css'**]**

**})**

**export** **class** AppComponent **{**

BranchId = "2";

RegisterStudent**(**studentForm: NgForm**)**: **void** **{**

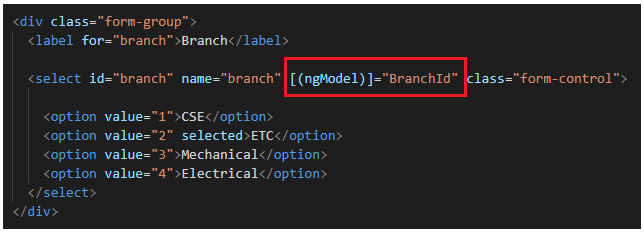
console.log**(**studentForm.value**)**;

**}**

**}**

##### **Modifying the app.component.html file**

Next, we include to include the ngModel directive and bind it with the component property BranchId. To do so, modify the Select List code in the **app.component.html** file as shown below.



With the above changes in place, now if you browse the application, then you should see the ETC branch is selected by default in the Branch Drop Down List when the form loads .

Now, even if you remove the “selected” attribute from the Option, then also it will work i.e. it will select the ETC by default. This is possible because of the two-way data binding which is provided by angular. In our example, we do not want the ETC to be selected by default, so we remove the “selected” attribute and the “BranchId” property from the component class and ngModel directive.

##### **How to disable a Drop Down List in Angular Template Driven Forms?**

In order to disable a drop down list in Angular Template Driven Form, we need use the disabled attribute on the select element as shown below.

**<select id=”branch” name=”branch” class=”form-control” ngModel disabled>**

With the above changes, now it is not possible to select any item from the drop down list. As we already discussed, by default, the disabled form controls are not included in the Angular auto generated form model.

In our example, we do not want the dropdownlist to be disabled, so please remove the disabled attribute from the select element.

##### **How to get the select list options from the component class?**

As of now we have hard-coded the select list options in HTML itself. In most of the real-time applications, you will get this data from a database. So, modify the **app.component.ts** file as shown below. Here, we created one property called **Branches**which will return the list of items that we want to show in the drop down list. It has two properties id and name.

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

**import** **{** NgForm **}** from '@angular/forms'

@Component**({**

selector: 'app-root',

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

styleUrls: **[**'./app.component.css'**]**

**})**

**export** **class** AppComponent **{**

RegisterStudent**(**studentForm: NgForm**)**: **void** **{**

console.log**(**studentForm.value**)**;

**}**

Branches: **any[]** = **[**

**{** id: 1, name: 'CSE' **}**,

**{** id: 2, name: 'ETC' **}**,

**{** id: 3, name: 'Mechanical' **}**,

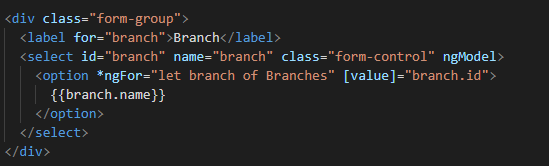
**{** id: 4, name: 'Electrical' **}**

**]**;

**}**

##### **Modifying the app.component.html file:**

Next we need to modify the drop down list code in the app.component.html file as shown below.



**Step 1** – So let create a project – This tutorial assumes that you have already installed “Angular CLI”, if not please visit [here](https://github.com/angular/angular-cli#prerequisites) and install it. After successful installation you should able to run below command to create and to serve project.



|  |  |
| --- | --- |
|  | ng new PROJECT-NAME  cd PROJECT-NAME  ng serve |

**Step2** – Add *FormsModule* module to imports array in the *app.module.ts* file.

**Step 3**– Edit app.component.ts file. We have added few methods to supply sample data and *onSelectCountry,onSelectState*methods to detect the change in the parent drop down.



|  |  |
| --- | --- |
|  | import { Component } from '@angular/core';      @Component({    selector: 'app-root',    templateUrl: './app.component.html',    styleUrls: ['./app.component.css']  })  export class AppComponent {      selectedCountry = 0;    selectedState = 0;      title = 'app';    states = [];    cities = [];        onSelectCountry(country\_id: number) {      this.selectedCountry = country\_id;      this.selectedState = 0;      this.cities = [];      this.states = this.getStates().filter((item) => {        return item.country\_id === Number(country\_id)      });    }      onSelectState(state\_id: number) {      this.selectedState = state\_id;      this.cities = this.getCity().filter((item) => {        return item.state\_id === Number(state\_id)      });    }      getCountries() {      return [        { id: 1, name: 'India' },        { id: 2, name: 'USA' },        { id: 3, name: 'Australia' }      ];    }      getStates() {      return [        { id: 1, country\_id: 1, name: 'Andhra Pradesh' },        { id: 2, country\_id: 1, name: 'Madhya Pradesh' },        { id: 3, country\_id: 2, name: 'San Francisco' },        { id: 4, country\_id: 2, name: 'Los Angeles' },        { id: 5, country\_id: 3, name: 'New South Wales' },        { id: 6, country\_id: 3, name: 'Victoria' },      ]    }      getCity() {      return [        { id: 1, state\_id: 1, name: 'Guntur' },        { id: 2, state\_id: 1, name: 'Vijayawada' },        { id: 3, state\_id: 1, name: 'Nellore' },        { id: 4, state\_id: 1, name: 'Kadapa' },        { id: 5, state\_id: 2, name: 'Warangal' },        { id: 6, state\_id: 2, name: 'Hyderabad' },        { id: 7, state\_id: 2, name: 'Karimnagar' },        { id: 8, state\_id: 2, name: 'Kadapa' },        { id: 9, state\_id: 3, name: 'SOMA' },        { id: 10, state\_id: 3, name: 'Richmond' },        { id: 11, state\_id: 3, name: 'Sunset' },        { id: 12, state\_id: 4, name: 'Burbank' },        { id: 13, state\_id: 4, name: 'Hollywood' },        { id: 14, state\_id: 5, name: 'Sunset' },        { id: 15, state\_id: 5, name: 'Burbank' },        { id: 16, state\_id: 5, name: 'Hollywood' },        { id: 17, state\_id: 6, name: 'Benalla' },        { id: 18, state\_id: 6, name: 'Melbourne' },      ]    }  } |

step 4 – now update your *app.component.html* view file with below content.



|  |  |
| --- | --- |
|  | <!--The content below is only a placeholder and can be replaced.-->  <div class="container">    <div style="text-align:center">      <h1>        Welcome to {{title}}!!      </h1>    </div>      <div class="form-group">      <label class="control-label" for="Country">Country:</label>      <select \*ngIf="getCountries()" [(ngModel)]="selectedCountry" (change)="onSelectCountry($event.target.value)" class="form-control input-lg" id="country"         >         <option value="0">Select Country</option>        <option \*ngFor="let country of getCountries()" value= {{country.id}}>{{country.name}}</option>      </select>    </div>      <div class="form-group">      <label class="control-label" for="States">States:</label>      <select \*ngIf="states" [(ngModel)]="selectedState" (change)="onSelectState($event.target.value)" class="form-control input-lg" id="state">          <option value="0">Select State</option>        <option \*ngFor="let state of states" value= {{state.id}}>{{state.name}}</option>      </select>    </div>      <div class="form-group">      <label class="control-label" for="City">City:</label>      <select class="form-control input-lg" id="city">          <option \*ngIf="!selectedState" value="0">Select City</option>          <option \*ngFor="let city of cities" value= {{city.id}}>{{city.name}}</option>      </select>    </div>  </div> |

step 5 – As you already started server in the step 1, now you should able to see output something like below shown image. If your is not running, issue *ng serve* from the root of your project.

Template Driven Validation example- angular 12 class example

<!doctype html>

<html lang="en">

<head>

    <title>Title</title>

    <!-- Required meta tags -->

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bootstrap CSS -->

    <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">

</head>

<body>

    <!-- Optional JavaScript -->

    <!-- jQuery first, then Popper.js, then Bootstrap JS -->

    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>

    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1" crossorigin="anonymous"></script>

    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>

    <br/>

    <br/>

    <div class="container">

        <div class="row">

            <div class="form-bg">

                <form #studentForm="ngForm" (ngSubmit)="RegisterStudent(studentForm)">

                    <div class="panel panel-primary">

                        <div class="panel-heading">

                            <h3 class="panel-title">Student Registration</h3>

                        </div>

                        <div class="panel-body">

                            <div class="form-group">

                                <label for="firstName">First Name</label>

                                <input id="firstName" type="text" class="form-control" name="firstName" #firstname="ngModel" required [(ngModel)]="studentForm.value.firstName">

                                <div \*ngIf="!firstname?.valid && (firstname?.dirty || firstname?.touched)">

                                    Invalid First Name

                                </div>

                                <div \*ngIf="firstname.errors?.['required']">

                                    First Name is required

                                </div>

                            </div>

                            <div class="form-group ">

                                <label for="lastName ">Last Name</label>

                                <input id="lastName " type="text " class="form-control " name="lastName " #lastName="ngModel" required [(ngModel)]="studentForm.value.lastName ">

                                <div \*ngIf=" !lastName?.valid && (lastName?.dirty || lastName?.touched) ">

                                    Invalid First Name

                                </div>

                                <div \*ngIf="lastName.errors?.[ 'required'] ">

                                    lastName is required

                                </div>

                            </div>

                            <div class="form-group ">

                                <label for="email ">Email</label>

                                <input id="email " type="text " class="form-control " name="email" #email="ngModel" [(ngModel)]="studentForm.value.email" pattern="^[a-z0-9.\_%+-]+@[a-z0-9.-]+\.[a-z]{2,4}$" required>

                                <div \*ngIf="!email?.valid && (email?.dirty || email?.touched)" class="error">

                                    <div \*ngIf="email.errors?.['required']">

                                        Email is required

                                    </div>

                                    <div \*ngIf="email.errors?.['email']">

                                        Invalid Email Address

                                    </div>

                                    <div \*ngIf="email.errors?.['pattern']">

                                        invalid email

                                    </div>

                                </div>

                            </div>

                            <div class=" form-group ">

                                <label for="Gender ">Gender</label><br>

                                <input type="radio" name="gender " value="male " checked [(ngModel)]="gender ">Male

                                <input type="radio" name="gender " value="female " [(ngModel)]="gender ">Female </div>

                            <div class="form-group">

                                <div class="form-control">

                                    <input type="checkbox " name="isaccepted " ngModel>AcceptTerms and Condition</div>

                            </div>

                            <div class="form-group ">

                                <div class="form-control ">

                                    <select name="branch " id="branch " ngModel class="form-control ">

                                        <!-- <option value="1 ">ComputerScience</option>

                                        <option value="2 ">Mechanical</option>

                                        <option value="3 ">ai and ml</option>

                                        <option value="4 ">Electrical and Electronics</option>

                                        <option value="5 ">Electronics and Communication</option> -->

                                     <option \*ngFor="let branch of Branches " [value]="branch.name ">

                                        {{branch.name}}

                                      </option>

                                    </select>

                                </div>

                            </div>

                        </div>

                        <div class=" panel-footer ">

                            <button class="btn btn-primary " type="submit ">Submit</button>

                        </div>

                    </div>

                </form>

            </div>

        </div>

    </div>

</body>

</html>

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

import { NgForm } from '@angular/forms';

@Component({

  selector: 'app-root',

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

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  title = 'ngforms';

   gender:any="";

   firstName:any="";

   lastName:any=""

   email:any="";

   isaccepted:any;

  RegisterStudent(studentForm:NgForm):void{

    this. firstName=studentForm.controls['firstName'].value;

    this. lastName=studentForm.controls['lastName'].value;

    this. email=studentForm.controls['email'].value;

    this. gender=studentForm.controls['gender'].value

    var isaccepted=studentForm.controls['isaccepted'].value;

    console.log(studentForm.value);

  }

  Branches:any[]=[

    {id:1,name:'computerscience'},

    {id:3,name:'Mechanical'},

    {id:3,name:'Ai amd ML'},

    {id:4,name:'Electrical and Electronics'},

    {id:5,name:'Electronics and Communication'}

  ]

}