# Project Setting

- AppMneminic: Assignment

- Email : deepakshinde0007@gmail.com

- Name : Angular Project to Display Country information

## Create an Angular project that displays a list of regions, countries and a simple table using Angular, TypeScript, Observables and NGRX.

Use the Steps below to run Locally

1. Download Zip file

2. cd CountriesDetails

3. npm install

4. ng serve --open

5. Navigate to `http://localhost:4200/`

Used Library:

1. NGRX

2. Boostrap

Project Description :

This is Deepak Shinde. I have created this project for to get country information by selecting conuntry. In this Angular project that displays a list of regions, countries and a simple table using Angular, TypeScript, Observables and NGRX.

I have created common component so we can use this comonent in multiple places

1. Navigation Component

- We need navigation menu in every page so i have created this component for common purpose and placed in shared folder

- I have used Boostap navigation layout

2. Dropdown Component

- we are using two dropdown in this project so i have created common component for this and communicating this by using Input and output decorators.

Following Input Params we are using

@Input() values: any[] = [];

@Input() label?: string;

@Input() isDisabled?: boolean;

@Output() valueChange = new EventEmitter<any>();

3. Home componet for landing page so we can display all basic and about project information in Home component

4. Country Component this is feature module of our project

- inside country module we have used seperate routing module for navigation,

- created country and country details component

- created the necessary state, reducer, actions, selectors and the effect to cache the regions and their countries.

- Used of NGRX

- we have take care of everything releated to Code quality, readability, naming, formatting

- Used of types and type safety

- Selector for dropdown

- Boostrap Table to display informationselected country

- Dependant dropdown

- created sperate service

- calling API by using Effect NGRX

- calling region dropdown values from Store module ( hardCoded ) but we have used selector to get the region values

- create and Maintain State

- get data from store and update state

- cached the regions and their countries data using Store.

- checked all necessary conditions

- used Observable

- used Asyn pipe to get data and pass to dropdown selector

- used country details selector for common purpose

- If user not selected region then country dropdown values should be disabled

- created neceesary varibles and assign specific types

5. Folder structure :

Countries-info

-Component

- country component

- country Details Component

-Models

- country Model

- region model

-Service

- countries-info service

-Store

-action

-effects

-reducer

-region-country state

-region state

-selectors

-Seperate countries-info Routing module

-countries-info Module

Home component

-Component

-Home Component

-Seperate Home Routing module

-Home Module

Shared

-Component

-dropdown

-navigation

-models

-dropdownvalue.model

App routing Module

App Module

App Compoenent

6. Apply Lazy Loaded module

- we are not loading all modules we are loading country module on demand or wheen we navigate to country Module

7. Hash Routing

6. added API base url in enviorment.ts file

# Following Best Practices follow in this Project

- Naming Convention

- Safe Navigation Operator(?, !)

- Lazy Lading

- Routing Module

- using trckby \*ngFor

- Enviorment Varible

- component Reusablity

- NgRx Store

- Redux

- State Managment

- used default boostrap classes

# CountriesDetails

This project was generated with [Angular CLI](https://github.com/angular/angular-cli) version 12.2.4.

## Development server

Run `ng serve` for a dev server. Navigate to `http://localhost:4200/`. The app will automatically reload if you change any of the source files.

## Code scaffolding

Run `ng generate component component-name` to generate a new component. You can also use `ng generate directive|pipe|service|class|guard|interface|enum|module`.

## Build

Run `ng build` to build the project. The build artifacts will be stored in the `dist/` directory.

## Running unit tests

Run `ng test` to execute the unit tests via [Karma](https://karma-runner.github.io).

## Running end-to-end tests

Run `ng e2e` to execute the end-to-end tests via a platform of your choice. To use this command, you need to first add a package that implements end-to-end testing capabilities.

## Further help

To get more help on the Angular CLI use `ng help` or go check out the [Angular CLI Overview and Command Reference](https://angular.io/cli) page.