

# Code

## App-Module:

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

import { AppComponent } from './app.component';
import { RouterModule, Routes } from '@angular/router';
import { HomeComponent } from './home/home.component';
import { AddBlogComponent } from './view-blogs/add-blog/add-blog.component';
import { FormsModule } from '@angular/forms';
import { ViewBlogsComponent } from './view-blogs/view-blogs.component';
import { BlogComponent } from './view-blogs/blog/blog.component';
import { HttpClientModule } from '@angular/common/http';
import { ContentLengthPipe } from './pipes/content-length.pipe';
import { ModifyBlogsComponent } from './view-blogs/modify-blogs/modify-blogs.component';
import { ModifyBlogComponent } from './view-blogs/modify-blogs/blog/blog.component';
import { MatIconModule } from '@angular/material/icon';
import { BrowserAnimationsModule } from '@angular/platform-browser/animations';

const appRoutes: Routes = [
  { path: '', component: HomeComponent },
  { path: 'AddBlog', component: AddBlogComponent },
  { path: 'ViewBlogs', component: ViewBlogsComponent },
  { path: 'ModifyBlogs', component: ModifyBlogsComponent },
];

@NgModule({
  declarations: [
    AppComponent,
    HomeComponent,
    AddBlogComponent,
    ViewBlogsComponent,
    BlogComponent,
    ContentLengthPipe,
    ModifyBlogsComponent,
    ModifyBlogComponent,
  ],
  imports: [
    BrowserModule,
    RouterModule.forRoot(appRoutes),
    FormsModule,
    HttpClientModule,
    MatIconModule,
    BrowserAnimationsModule,
  ],
  providers: [],
  bootstrap: [AppComponent],
})
export class AppModule {}
```

## App Component:

### HTML:

```
<nav class="navbar navbar-expand-sm nav-bg navbar-dark fixed-top">
  <div class="container">
    <a href="#" class="navbar-brand">Blog App</a>
    <button
      class="navbar-toggler"
```

```

    type="button"
    data-toggle="collapse"
    data-target="#navbarContent"
  >
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarContent">
    <ul class="navbar-nav">
      <li
        class="nav-item"
        role="presentation"
        routerLinkActive="route-active"
        [routerLinkActiveOptions]="{ exact: true }"
      >
        <a routerLink="/" class="nav-link" id="router-link">Home</a>
      </li>
      <li
        class="nav-item"
        role="presentation"
        routerLinkActive="route-active"
        [routerLinkActiveOptions]="{ exact: true }"
      >
        <a routerLink="/AddBlog" class="nav-link" id="router-link"
          >Add Blog</a>
        </li>
      <li
        class="nav-item"
        role="presentation"
        routerLinkActive="route-active"
        [routerLinkActiveOptions]="{ exact: true }"
      >
        <a routerLink="/ViewBlogs" class="nav-link" id="router-link"
          >View Blogs</a>
        </li>
      <li
        class="nav-item"
        role="presentation"
        routerLinkActive="route-active"
        [routerLinkActiveOptions]="{ exact: true }"
      >
        <a routerLink="/ModifyBlogs" class="nav-link" id="router-link"
          >Modify Blogs</a>
        </li>
    </ul>
  </div>
</div>
</nav>
<div class="router-display">
  <router-outlet></router-outlet>
</div>

```

CSS:

```

.navbar {
  padding-bottom: 0 !important;
  background-color: var(--secondary-color1) !important;
}
.navbar-nav {
  justify-content: space-evenly;
  width: 100%;
}

```

```

.navbar-nav > li {
  font-weight: bolder;
  padding: 0.2px 20px;
}

nav .navbar-nav li a {
  color: white !important;
}

.route-active {
  border-bottom: 0 !important;
  border: 2px solid white;
  border-radius: 5px 5px 0px 0px;
  background-color: white;
}

.route-active > #router-link {
  color: var(--secondary-color1) !important;
  font-weight: bold;
}

```

Type Script:

```

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'BlogApplication';
}

```

Home Component:

Html:

```

<div class="home-div w-100 d-flex justify-content-center align-items-center">
  <div class="home-div-child animate__animated animate__bounce">
    <h1>Welcome to Blog Application</h1>
    <p class="h5 text-center font-weight-bold m-5">Developed By Manoj M</p>
  </div>
</div>

```

CSS:

```

.home-div {
  width: 100vw;
  height: 100vh !important;
}

.home-div-child {
  border: 5px solid #ffd6a5;
  padding: 20px;
}

```

TypeScript:

```

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

```

```

@Component({
  selector: 'app-home',
  templateUrl: './home.component.html',
  styleUrls: ['./home.component.css']
})
export class HomeComponent {
}

```

## Content Length Pipe:

```

import { Pipe, PipeTransform } from '@angular/core';

@Pipe({
  name: 'contentLength',
})
export class ContentLengthPipe implements PipeTransform {
  transform(value: any, end: number): any {
    if (value.length == end) return value;
    else if (value.length > 500) return value.substr(0, end) + '...';
  }
}

```

## Blog Service:

```

import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { blog } from '../view-blogs/blog/blog.interface';
import { map } from 'rxjs/operators';
import { Observable } from 'rxjs';

@Injectable({
  providedIn: 'root',
})
export class BlogAppService {
  constructor(private httpClient: HttpClient) {}

  URL: string = 'http://localhost:3000/posts';

  createPost(postData: blog) {
    return this.httpClient.post(this.URL, postData);
  }

  fetchposts() {
    return this.httpClient.get(this.URL).pipe(
      map((resp) => {
        const blogArr = [];
        for (const key in resp) {
          blogArr.push(resp[key]);
        }
        return blogArr;
      })
    );
  }

  DeletePost(id: number) {
    return this.httpClient.delete(`${this.URL}/${id}`);
  }
}

```

```

updatePost(updatedPost: blog) {
  return this.httpClient.put(`${this.URL}/${updatedPost.id}`, updatedPost);
}
}

```

## Add Blog Component:

### HTML

```

<div class="container wrapper">
  <div
    class="alert alert-success alert-dismissible fade show text-center"
    role="alert"
    *ngIf="blogAdded"
  >
    <strong>{{ title }}</strong> Posted Successfully
    <button type="button" class="close" data-dismiss="alert" aria-label="Close">
      <span aria-hidden="true">&times;</span></button>
    </div>
  <div class="container w-50 mt-5 mb-5">
    <h1 class="mt-5 text-center">Add Posts</h1>
    <form #f="ngForm">
      <div class="form">
        <div class="form-group">
          <label for="title" class="form-label">
            >Author <span class="imp">*</span></label>
          <input
            type="text"
            placeholder="Enter Author Name"
            name="author"
            class="form-control"
            id="title"
            ngModel
            required
          />
        </div>
        <div class="form-group">
          <label for="title" class="form-label">
            >Title <span class="imp">*</span></label>
          <input
            type="text"
            placeholder="Enter Title"
            name="title"
            class="form-control"
            id="title"
            ngModel
            required
          />
        </div>
        <div class="form-group">
          <label for="content">Content <span class="imp">*</span></label>
          <textarea
            type="text"
            placeholder="Enter content"
            name="content"
            class="form-control"
            id="content"
            cols="35"

```

```

        rows="4"
        ngModel
        [(ngModel)]="content"
        minLength="200"
        required
        (keyup)="updateContentLen()"
    ></textarea>
    <small class="form-text text-muted"
        >Min: {{ contentLen }}/200 letters</small>
    >
</div>
<div class="form-group">
    <label for="img" class="form-label"
        >Add Images <span class="imp">*</span></label>
    >
    <input
        type="url"
        name="image"
        class="form-control"
        id="img"
        placeholder="Enter Image URI"
        accept="image/png, image/jpeg,image/jpg"
        url
        required
        ngModel
        (change)="onChange($event)"
    />
    <small class="form-text text-muted">
        You can add only one image</small>
    >
</div>
<div class="form-group">
    <input
        type="submit"
        placeholder="Enter Content"
        name="username"
        [disabled]="!f.valid || contentLen < 200"
        class="form-control btn btn-primary"
        id="title"
        (click)="onBlogSubmit(f.value, f)"
    />
</div>
</div>
</form>
</div>
</div>

```

CSS:

```

.imp {
    color: red;
    font-weight: bold;
}

```

TypeScript:

```

import { Component } from '@angular/core';
import { blog } from '../blog/blog.interface';
import { NgForm } from '@angular/forms';
import { BlogAppService } from '../../services/blog-app.service';

@Component({
    selector: 'app-add-blog',
    templateUrl: './add-blog.component.html',

```

```

    styleUrls: ['./add-blog.component.css'],
  })
export class AddBlogComponent {
  content: string = '';
  contentLen: number = 0;
  blogAdded: boolean = false;
  title: string;
  file: File = null;

  constructor(private blogService: BlogAppService) {}

  onChange(event) {
    this.file = event.target.files[0];
  }

  updateContentLen() {
    const content_list = this.content.split(' ').filter((elem) => elem != '');
    this.contentLen = content_list.length;
  }

  onBlogSubmit(blogData: blog, form: NgForm) {
    console.log(blogData);
    this.blogService.createPost(blogData).subscribe(() => {
      this.title = blogData.title;
      this.blogAdded = true;
      form.reset();
      this.contentLen = 0;
      setTimeout(() => {
        this.blogAdded = false;
      }, 5000);
    });
  }
}

```

## Blog Interface:

```

export interface blog {
  author: string;
  title: string;
  content: string;
  id?: number;
  image?: URL;
}

```

## Blog Component:

### HTML:

```

<div class="card m-4 border-left p-2 animate__animated">
  <div class="">
    
    <div class="card-body">
      <div>
        <h4 class="card-title d-inline-block">{{ blogInp.title }}</h4>
        <small class="text-muted">
          Written By {{ blogInp.author | uppercase }}.</small>
        >
        <p class="card-text">

```

```

        {{ blogInp.content | contentLength : contentLen }}
      </p>
      <p class="card-text"></p>
      <button
        [disabled]="blogInp.content.length <= 200"
        href="#"
        class="btn btn-danger btn-sm bg-UTOrange"
        (click)="onExpandMore()"
      >
        {{ btnStatus }}
      </button>
    </div>
  </div>
</div>

```

CSS:

```

.transition {
  transition: all ease-in-out 2000ms;
}

.border-Left {
  border-left: 5px solid var(--selectiveYellow) !important;
}

#card-img {
  width: 15vw;
  height: 35vh;
  float: left;
  margin: 10px;
}

.card {
  box-shadow: rgba(0, 0, 0, 0.35) 0px 5px 15px;
  transition: box-shadow ease-in-out 300ms;
}

.card:hover {
  box-shadow: rgba(50, 50, 93, 0.25) 0px 50px 100px -20px,
    rgba(0, 0, 0, 0.3) 0px 30px 60px -30px;
  cursor: pointer;
}

```

Typescript:

```

import { Component, EventEmitter, Input, Output } from '@angular/core';
import { blog } from './blog.interface';

@Component({
  selector: 'app-blog',
  templateUrl: './blog.component.html',
  styleUrls: ['./blog.component.css'],
})
export class BlogComponent {
  @Input()
  blogInp: blog;
  btnStatus: string = 'Read More';

  contentLen: number = 500;

  onExpandMore() {
    if (this.contentLen == 500) {
      this.contentLen = this.blogInp.content.length;
    }
  }
}

```



```

    this.btnStatus = 'Read Less';
  } else {
    this.contentLen = 500;
    this.btnStatus = 'Read More';
  }
}
}

```

## View Blog Component:

HTML:

```

<div class="container wrapper">
  <h1 class="text-center m-2">Blogs</h1>
  <div class="alert alert-danger text-center" *ngIf="error">
    <strong>Server Error: </strong>{{ error }}
  </div>
  <div class="alert alert-info text-center" *ngIf="fetching && !error">
    <strong>Loading...</strong>
  </div>
  <div *ngIf="!fetching && !error">
    <div
      class="alert alert-info bg-UTOrange text-light text-center"
      *ngIf="blogs.length <= 0"
    >
      <strong>No Post Available</strong>
    </div>
    <app-blog *ngFor="let myBlog of blogs" [blogInp]="myBlog">{{
      myBlog.title
    }}</app-blog>
  </div>
</div>

```

Typescript:

```

import { Component, OnInit } from '@angular/core';
import { BlogAppService } from '../services/blog-app.service';
import { blog } from '../blog/blog.interface';

@Component({
  selector: 'app-view-blogs',
  templateUrl: './view-blogs.component.html',
  styleUrls: ['./view-blogs.component.css'],
})
export class ViewBlogsComponent implements OnInit {
  constructor(private blogService: BlogAppService) {}
  ngOnInit(): void {
    this.fetchposts();
  }

  blogs: blog[] = [];
  error: string;
  fetching: boolean = true;

  fetchposts() {
    this.blogService.fetchposts().subscribe(
      (response) => {
        this.blogs = response;
        console.log(this.blogs);
        this.fetching = false;
      },
      (err) => {

```

```

        this.error = err.message;
    }
    });
}
}

```

## Modify Blog Component:

HTML:

```

<div class="container wrapper">
  <h1 class="text-center m-2">Blogs</h1>
  <div class="alert alert-danger text-center" *ngIf="error">
    <strong>Server Error: </strong>{{ error }}
  </div>
  <div class="alert alert-info text-center" *ngIf="fetching && !error">
    <strong>Loading...</strong>
  </div>
  <div *ngIf="!fetching">
    <div
      class="alert alert-info bg-UTOrange text-light text-center"
      *ngIf="dbPosts.length <= 0"
    >
      <strong>No Post Available</strong>
    </div>
    <app-blog-modify
      *ngFor="let blog of dbPosts"
      [blogInp]="blog"
      (postDeleted)="onPostDelete()"
    ></app-blog-modify>
  </div>
</div>

```

Typescript:

```

import { Component, OnInit } from '@angular/core';
import { BlogAppService } from 'src/app/services/blog-app.service';
import { blog } from '../blog/blog.interface';

@Component({
  selector: 'app-modify-blogs',
  templateUrl: './modify-blogs.component.html',
  styleUrls: ['./modify-blogs.component.css'],
})
export class ModifyBlogsComponent implements OnInit {
  constructor(private blogService: BlogAppService) {}
  ngOnInit(): void {
    this.fetchPosts();
  }

  dbPosts: blog[] = [];
  error: string;
  fetching: boolean = true;

  fetchPosts() {
    this.blogService.fetchposts().subscribe(
      (response: any) => {
        this.fetching = false;
        this.dbPosts = response;
        console.log(response.currentValue);
        console.log(
          Math.floor((response.currentValue / response.totalValue) * 100)
        );
      }
    );
  }
}

```

```

    );
  },
  (err) => {
    this.error = err.message;
  }
);
}
onPostDelete() {
  this.fetchPosts();
}
}

```

## Modify Blog / Blog Component

HTML:

```

<div
  class="card m-4 border-left"
  [ngClass]="{
    'border-full-modify': editContent,
  }"
  *ngIf="editContent"
>
  <div class="card-body">
    <div class="form-group">
      <label for="img">Title: </label>
      <input
        placeholder="Enter Title"
        class="form-control"
        [readOnly]="!editContent"
        [(ngModel)]="blogInp.title"
        required
      />
    </div>
    <div class="form-group">
      <label for="img">Image url: </label>
      <input
        type="url"
        name="image"
        class="form-control"
        id="img"
        placeholder="Enter Image URI"
        [(ngModel)]="blogInp.image"
        required
      />
    </div>
    <div class="form-group">
      <label for="'txt' + {{ blogInp.id }}">Content: </label>
      <textarea
        class="form-control"
        [(ngModel)]="blogInp.content"
        [readOnly]="!editContent"
        [id]="'txt' + blogInp.id"
        required
        (click)="onTextareaLoad()"
      ></textarea>
    </div>

    <div class="btn-align">
      <button href="#" class="btn btn-sm btn-danger" (click)="onConcelEdit()">
        Cancel
      </button>
    </div>
  </div>

```

```

</button>
<button
  href="#"
  class="btn btn-danger btn-sm bg-UTOrange"
  (click)="onEditText()"
>
  {{ modifyBtn }}
</button>
<div class="alert alert-success" *ngIf="postSaved">
  Successfully Updated
</div>
</div>
</div>
</div>
</div>

<!-- Displaying -->
<div class="card m-4 border-left p-2" *ngIf="!editContent">
  <div class="alert alert-success text-center" *ngIf="postSaved">
    Successfully Updated
  </div>
  <div class="">
    
    <div class="card-body">
      <div>
        <h4 class="card-title d-inline-block">{{ blogInp.title }}</h4>
        <small class="text-muted">
          Written By {{ blogInp.author | uppercase }}.</small>
        </div>
        <p class="card-text">
          {{ blogInp.content | contentLength : contentLen }}
        </p>
        <button
          [disabled]="blogInp.content.length <= 200"
          href="#"
          class="btn btn-danger btn-sm bg-UTOrange"
          (click)="onExpandMore()"
        >
          {{ btnStatus }}
        </button>
        <button href="#" class="btn btn-sm btn-danger" (click)="onDeletePost()">
          Delete
        </button>
        <button
          href="#"
          class="btn btn-danger btn-sm bg-UTOrange"
          (click)="onEditText()"
        >
          {{ modifyBtn }}
        </button>
      </div>
    </div>
  </div>
</div>
</div>
</div>

```

CSS:

```

.btn-align {
  width: 100%;
  display: flex;

```

```

    align-items: center !important;
    gap: 10px;
    flex-wrap: wrap !important;
}
.border-Left {
    border-left: 5px solid var(--selectiveYellow) !important;
    transition: border ease-in-out 500ms;
}

.border-full-modify {
    border: 5px solid var(--UTOrange) !important;
    transition: border ease-in-out 500ms;
}

.alert {
    padding: 3px !important;
}

#card-img {
    width: 15vw;
    height: 35vh;
    float: left;
    margin: 10px;
}

.card-body > div > button {
    margin: 4px !important;
}

.card {
    box-shadow: rgba(0, 0, 0, 0.35) 0px 5px 15px;
    transition: box-shadow ease-in-out 300ms;
}

.card:hover {
    box-shadow: rgba(50, 50, 93, 0.25) 0px 50px 100px -20px,
        rgba(0, 0, 0, 0.3) 0px 30px 60px -30px;
    cursor: pointer;
}

```

## Typescript:

```

import {
    Component,
    EventEmitter,
    Input,
    OnChanges,
    OnInit,
    Output,
    SimpleChanges,
} from '@angular/core';
import { blog } from '../blog/blog.interface';
import { BlogAppService } from 'src/app/services/blog-app.service';

@Component({
    selector: 'app-blog-modify',
    templateUrl: './blog.component.html',
    styleUrls: ['./blog.component.css'],
})
export class ModifyBlogComponent implements OnInit {
    constructor(private blogService: BlogAppService) {}
    ngOnInit(): void {}

    @Input()

```

```

blogInp: blog;

btnStatus: string = 'Read More';

contentLen: number = 500;

@Output()
postDeleted = new EventEmitter<{ deleted: boolean }>();

editContent: boolean = false;
modifyBtn: string = 'Modify Post';
postSaved: boolean = false;

onDeletePost() {
  this.blogService.DeletePost(this.blogInp.id).subscribe(() => {
    this.postDeleted.emit({ deleted: true });
  });
}
onEditText() {
  if (this.editContent == true) {
    this.modifyBtn = 'Modify Post';
    this.blogService.updatePost(this.blogInp).subscribe((resp) => {
      this.postSaved = true;
      setTimeout(() => {
        this.postSaved = false;
      }, 3000);
    });
    this.editContent = false;
  } else {
    this.modifyBtn = 'Save Poast';
    this.editContent = true;
  }
}

onExpandMore() {
  if (this.contentLen == 500) {
    this.contentLen = this.blogInp.content.length;
    this.btnStatus = 'Read Less';
  } else {
    this.contentLen = 500;
    this.btnStatus = 'Read More';
  }
}

onConcelEdit() {
  this.editContent = false;
  this.modifyBtn = 'Modify Post';
}

onTextareaLoad() {
  const textarea: HTMLTextAreaElement = document.querySelector(
    `#txt${this.blogInp.id}`
  );
  textarea.style.height = 'auto';
  textarea.style.height = textarea.scrollHeight + 'px';
}
}

```