

## Source Code

### app.component.html

```
<main>
  <router-outlet></router-outlet>
</main>
```

### app.module.ts

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

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { NavbarComponent } from './components/navbar/navbar.component';
import { HomeComponent } from './components/home/home.component';
import { RouterModule } from '@angular/router';
import { NewPostComponent } from './components/new-post/new-post.component';
import { FormsModule, ReactiveFormsModule } from '@angular/forms';
import { HttpClientModule } from '@angular/common/http';
import { ContactFormComponent } from './components/contact-form/contact-form.component';
import { ApiService } from './services/api.service';
import { LoginComponent } from './components/login/login.component';
import { SignupComponent } from './components/signup/signup.component';
import { AuthService } from './services/auth.service';
import { MyPostsComponent } from './components/my-posts/my-posts.component';

@NgModule({
  declarations: [
    AppComponent,
    NavbarComponent,
    HomeComponent,
    NewPostComponent,
    ContactFormComponent,
    LoginComponent,
    SignupComponent,
    MyPostsComponent,
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
```

```

RouterModule.forRoot([
  {path: '', component: HomeComponent},
  {path: 'new-post/add', component: NewPostComponent},
  {path: 'new-post/:id', component: NewPostComponent},
  {path: 'contact-us', component: ContactFormComponent},
  {path: 'signup', component: SignupComponent},
  {path: 'login', component: LoginComponent},
  {path: 'home', component: HomeComponent},
  {path: 'my-posts', component: MyPostsComponent},
  {path: '**', component: LoginComponent}

]),
ReactiveFormsModule,
FormsModule,
HttpClientModule
],
providers: [ApiService, AuthService],
bootstrap: [AppComponent]
}))
export class AppModule { }

```

#### login.component.html

```

<nav class="navbar navbar-expand-md navbar-dark bg-dark mb-4">
  <div class="container-fluid">
    <a class="navbar-brand" routerLink="/">Blog App</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarCollapse" aria-controls="navbarCollapse" aria-
expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarCollapse">
      <ul class="navbar-nav me-auto mb-2 mb-md-0">
        <li class="nav-item">
          <a class="nav-link" routerLink="/login">Login</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" routerLink="/signup">Signup</a>
        </li>

        <!--<li ngbDropdown *ngIf="isUserLogin" class="nav-item dropdown">
          <a ngbDropdownToggle class="nav-link dropdown-
toggle" id="dropdown01" data-bs-toggle="dropdown" aria-haspopup="true" aria-
expanded="false">{{username}}</a>

```

```

        <div ngbDropdownMenu class="dropdown-menu" aria-
labelledby="dropdown01">
            <a ngbDropdownItem class="dropdown-item" routerLink="/new-
post">New Post</a>
            <a ngbDropdownItem class="dropdown-item"
(click)="logout()">Logout</a>

        </div>
    </li>-->
</ul>

</div>
</div>
</nav>
<div class="container">
<h2 class="login text-center">Sign In</h2>
<form #f="ngForm" (ngSubmit)="onLogin(f)">
    <input ngModel #email="ngModel" name="email" type="email" class="form-
control" placeholder="Email" required email>
    <div class="alert alert-danger" *ngIf="email.touched &&
email.errors?.['required']">Email is required!</div>
    <div class="alert alert-danger" *ngIf="email.touched &&
email.errors?.['email']">Your input doesn't match email format!</div>
    <br>
    <input ngModel #pass="ngModel" name="password" type="password"
class="form-control" placeholder="Password" required minlength="4">
    <div class="alert alert-danger" *ngIf="pass.touched &&
pass.errors?.['required']">Password is required!</div>
    <div class="alert alert-danger" *ngIf="pass.touched &&
pass.errors?.['minlength']">Please, enter atleast 4 character!</div>
    <br>
    <button class="btn btn-primary" [disabled]="email.invalid ||
pass.invalid">Login</button>
</form>
</div>

```

login.component.ts

```

import { Component, OnInit } from '@angular/core';
import { NgForm } from '@angular/forms';
import { Router } from '@angular/router';
import { AuthService } from 'src/app/services/auth.service';

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

```

```

    styleUrls: ['./login.component.scss']
  })
  export class LoginComponent implements OnInit {

    constructor( private router: Router,
      private authenticationService: AuthService,) { }

    ngOnInit(): void {
    }
    onLogin(form:NgForm){
      this.authenticationService.login().subscribe(res=>{
        const user = res.find((a:any)=>{
          return a.email === form.value.email && a.password ===
form.value.password
        });
        if(user){

          localStorage.setItem('user',user.name);
          localStorage.setItem('user_id',user.email+user.password);
          localStorage.setItem('email',user.email);
          this.router.navigate(["home"])
        }else{
          alert("user not found")
        }
      },err=>{
        alert("Something went wrong")
      })
    }
  }
}

```

#### login.component.spec.ts

```

import { ComponentFixture, TestBed } from '@angular/core/testing';
import { RouterTestingModule } from '@angular/router/testing';
import { AuthService } from 'src/app/services/auth.service';
import { HttpClientTestingModule, HttpTestingController } from
'@angular/common/http/testing';
import { LoginComponent } from './login.component';
import { FormsModule, ReactiveFormsModule } from '@angular/forms';

describe('LoginComponent', () => {
  let component: LoginComponent;
  let fixture: ComponentFixture<LoginComponent>;

  beforeEach(async () => {

```

```

    await TestBed.configureTestingModule({
      declarations: [ LoginComponent ],
      imports:[RouterTestingModule,HttpClientTestingModule,FormsModule],

    })
    .compileComponents();
  });

  beforeEach(() => {
    fixture = TestBed.createComponent(LoginComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });
  it('should have text as Sign In', () => {
    const fixture = TestBed.createComponent(LoginComponent);
    fixture.detectChanges();
    const compiled = fixture.nativeElement as HTMLElement;
    expect(compiled.querySelector('.container
h2')?.textContent).toContain('Sign');
  });
});

```

signup.component.html

```

<nav class="navbar navbar-expand-md navbar-dark bg-dark mb-4">
  <div class="container-fluid">
    <a class="navbar-brand" routerLink="/">Blog App</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarCollapse" aria-controls="navbarCollapse" aria-
expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarCollapse">
      <ul class="navbar-nav me-auto mb-2 mb-md-0">
        <li class="nav-item">
          <a class="nav-link" routerLink="/login">Login</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" routerLink="/signup">Signup</a>
        </li>

        <!--<li ngbDropdown *ngIf="isUserLogin" class="nav-item dropdown">

```

```

        <a NgbDropdownToggle class="nav-link dropdown-
toggle" id="dropdown01" data-bs-toggle="dropdown" aria-haspopup="true" aria-
expanded="false">{{username}}</a>
        <div NgbDropdownMenu class="dropdown-menu" aria-
labelledby="dropdown01">
            <a NgbDropdownItem class="dropdown-item" routerLink="/new-
post">New Post</a>
            <a NgbDropdownItem class="dropdown-item"
(click)="logout()">Logout</a>

        </div>
    </li>-->
</ul>

</div>
</div>
</nav>
<div class="container">
<h2 class="text-center">Sign Up</h2>
<form #f="ngForm" (ngSubmit)="onSignup(f)">
    <input ngModel #name="ngModel" name="name" type="text" class="form-
control" placeholder="Your name" required><br>
    <div class="alert alert-danger" *ngIf="name.touched && name.invalid">Your
name is required!</div>
    <input ngModel #email="ngModel" name="email" type="email" class="form-
control" placeholder="Email" required email><br>
    <div class="alert alert-danger" *ngIf="email.touched &&
email.errors?.['required']">Email is required!</div>
    <div class="alert alert-danger" *ngIf="email.touched &&
email.errors?.['email']">Your input doesn't match email format!</div>

    <input ngModel #pass="ngModel" name="password" type="password"
class="form-control" placeholder="Password" required minlength="4"><br>
    <div class="alert alert-danger" *ngIf="pass.touched &&
pass.errors?.['required']">Password is required!</div>
    <div class="alert alert-danger" *ngIf="pass.touched &&
pass.errors?.['minlength']">Please, enter atleast 4 character!</div>

    <input ngModel #confirmPass="ngModel" name="confirmPassword"
type="password" class="form-control" placeholder="Confirm Password"
required><br>
    <div class="alert alert-danger" *ngIf="confirmPass.touched &&
confirmPass.errors?.['required']">Confirm password required!</div>
    <div class="alert alert-danger" *ngIf="confirmPass.touched && pass.value
!= confirmPass.value">Confirm Password are not equal to password!</div>

```

```

    <button class="btn btn-primary" [disabled]="f.invalid || pass.value !=
confirmPass.value">Create Account</button>
</form>
</div>

```

signup.component.ts

```

import { Component, OnInit } from '@angular/core';
import { NgForm } from '@angular/forms';
import { Router } from '@angular/router';
import { User } from 'src/app/interfaces/user';
import { AuthService } from 'src/app/services/auth.service';

@Component({
  selector: 'app-signup',
  templateUrl: './signup.component.html',
  styleUrls: ['./signup.component.scss']
})
export class SignupComponent implements OnInit {
  userObj:User=new User();
  constructor(private auth:AuthService,private route:Router) { }

  ngOnInit(): void {
  }
  onSignup(form:NgForm){
    this.userObj.id=new Date().valueOf();
    this.userObj.name=form.value.name;
    this.userObj.email=form.value.email;
    this.userObj.password=form.value.password;
    localStorage.setItem('user',form.value.name);
    localStorage.setItem('user_id',form.value.email+form.value.password);
    localStorage.setItem('email',form.value.email);
    this.auth.addUser(this.userObj).subscribe(res=>
    {

      this.route.navigate(["\home"]);

    },
    error=>{
      alert("Some error occured");
    })
  }
}

```

### signup.component.spec.ts

```
import { HttpClientTestingModule } from '@angular/common/http/testing';
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { FormsModule } from '@angular/forms';
import { RouterTestingModule } from '@angular/router/testing';

import { SignupComponent } from './signup.component';

describe('SignupComponent', () => {
  let component: SignupComponent;
  let fixture: ComponentFixture<SignupComponent>;

  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [ SignupComponent ],
      imports:[RouterTestingModule,HttpClientTestingModule,FormsModule]
    })
    .compileComponents();
  });

  beforeEach(() => {
    fixture = TestBed.createComponent(SignupComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });

  it('should have text as Sign up', () => {
    const fixture = TestBed.createComponent(SignupComponent);
    fixture.detectChanges();
    const compiled = fixture.nativeElement as HTMLElement;
    expect(compiled.querySelector('.container
h2')?.textContent).toContain('Sign Up');
  });
});
```

### navbar.component.html

```
<nav class="navbar navbar-expand-md navbar-dark bg-dark mb-4">
  <div class="container">

    <a class="navbar-brand" routerLink="/">Food Blogs</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarCollapse" aria-controls="navbarCollapse" aria-
expanded="false" aria-label="Toggle navigation">
```



```

        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarCollapse">
        <ul class="navbar-nav me-auto mb-2 mb-md-0">
          <li class="nav-item">
            <a class="nav-link" routerLinkActive="active" aria-
current="page" routerLink="/home">Home</a>
          </li>
          <li class="nav-item">
            <a class="nav-link " routerLinkActive="active" aria-
current="page" routerLink="/contact-us">Contact-us</a>
          </li>
          <li class="nav-item">
            <a class="nav-link " routerLinkActive="active" routerLink="/my-
posts" aria-current="page">My Posts</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" routerLinkActive="active"
routerLink="/login" (click)="logout()">Logout</a>
          </li>
        </ul>
        <div class="d-flex">
          <button class="btn btn-success" (click)="newPost()">New
Post</button>
        </div>

      </div>
    </div>

  </nav>

```

#### navbar.component.ts

```

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

@Component({
  selector: 'app-navbar',
  templateUrl: './navbar.component.html',
  styleUrls: ['./navbar.component.scss']
})
export class NavbarComponent implements OnInit {
  username:any;
  constructor(private router:Router) {
    let uname=localStorage.getItem('user');
    if(uname!==""){

```

```

        this.username=username;
    }
    else
    {
        this.router.navigate(['login']);
    }
}

ngOnInit(): void {

}

newPost(){
    this.router.navigate(["/new-post/add"]);
}

logout(){
    localStorage.setItem('user',"");
    localStorage.setItem('user_id',"");
    localStorage.setItem('email',"");
    this.router.navigate(['login']);
}
}

```

#### navbar.component.spec.ts

```

import { ComponentFixture, TestBed } from '@angular/core/testing';
import { RouterTestingModule } from '@angular/router/testing';

import { NavbarComponent } from './navbar.component';

describe('NavbarComponent', () => {
    let component: NavbarComponent;
    let fixture: ComponentFixture<NavbarComponent>;

    beforeEach(async () => {
        await TestBed.configureTestingModule({
            declarations: [ NavbarComponent ],
            imports: [RouterTestingModule]
        })
        .compileComponents();
    });

    beforeEach(() => {
        fixture = TestBed.createComponent(NavbarComponent);
    });

```

```

    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});

```

[home.component.html](#)

```

<app-navbar></app-navbar>
<div class="container">
<div class="jumbotron p-3 p-md-5 text-white rounded bg-dark">
  <div class="col-md-6 px-0">
    <h2 class="heading display-5 font-italic">Welcome To Food Blogs
    {{username}}!</h2>
    <p class="lead my-3">Helping you celebrate the joy of food in a totally
    non-intimidating way.</p>

  </div>
</div><br><br>
<div class="container">
  <div class="col">
    <div class="row">
      <ng-container *ngFor="let p of postData;index as i">
        <div class="col col-md-4 col-sm-12">
          <div class="card" style="width: 18rem;">
            <img [src]="p.image" class="card-img-top" alt="..."
            height="200px">
            <div class="card-body">
              <h5 class="card-title">{{p.title | titlecase}}</h5>
              <p class="card-text">{{p.content}}</p>
              <p>Posted by {{p.author}} at {{p.createdAt | date:
              'dd/MM/yyyy hh:mm a'}}</p>
            </div>
          </div>
        </div>
      </ng-container>
    </div>
    <br><br>
    <div *ngIf="(i+1)%3===0" class="w-100"><br></div>
  </div>
</div>
</div>

```

</div>

[home.component.ts](#)

```
import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { PostModel } from 'src/app/interfaces/post-model';
import { ApiService } from 'src/app/services/api.service';

@Component({
  selector: 'app-home',
  templateUrl: './home.component.html',
  styleUrls: ['./home.component.scss']
})
export class HomeComponent implements OnInit {
  postData:any;
  username:any;

  constructor(private restApi:ApiService,private route:Router) {
    let uname=localStorage.getItem('user');
    // console.log("Uname="+uname);
    if(uname!==""){
      this.username=uname;
    }
    else
    {
      this.route.navigate(['login']);
    }
  }

  ngOnInit(): void {
    this.getAllPost();
  }
  getAllPost(){

    this.restApi.getPost().subscribe(res=>{
      this.postData=res;
    })
  }
  deletePost(i:number){
    let res=confirm("Are you sure you want to delete this post?");
    if(res){
      this.restApi.deletePost(i).subscribe(res=>
        { this.getAllPost();
          alert("Post Deleted");
        })
    }
  }
}
```

```

    /* editPost(i:number){
      this.route.navigate(['\new-post\' + i]);
    }*/
  }
}

```

[home.component.spec.ts](#)

```

import { HttpClientTestingModule } from '@angular/common/http/testing';
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { RouterTestingModule } from '@angular/router/testing';
import { NavbarComponent } from '../navbar/navbar.component';

import { HomeComponent } from './home.component';

describe('HomeComponent', () => {
  let component: HomeComponent;
  let fixture: ComponentFixture<HomeComponent>;

  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [ HomeComponent ,NavbarComponent],
      imports:[RouterTestingModule,HttpClientTestingModule]
    })
    .compileComponents();
  });

  beforeEach(() => {
    fixture = TestBed.createComponent(HomeComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });

  it('should render title', () => {
    const fixture = TestBed.createComponent(HomeComponent);
    fixture.detectChanges();
    const compiled = fixture.nativeElement as HTMLElement;
    expect(compiled.querySelector('.heading')?.textContent).toContain('Welcome
To Food Blogs');
  });

  it('should render title in a h2 tag', () => {
    const fixture = TestBed.createComponent(HomeComponent);

```

```

    fixture.detectChanges();
    const compiled = fixture.debugElement.nativeElement;
    expect(compiled.querySelector('h2').textContent).toContain('Welcome To
Food Blogs');
  });
});

```

new-post.component.html

```

<app-navbar></app-navbar>
<div class="container">
<div class="row">
  <div class="col-md-6">
    <form #f="ngForm" (ngSubmit)="addNewPost(f)">
      <div class="form-group">
        <label for="title">Post Title</label>
        <input #title="ngModel" [(ngModel)]="post.title" name="title"
id="title" type="text" class="form-control" required>
        <div class="alert alert-danger" *ngIf="title.touched &&
title.invalid">
          Title is required
        </div>
      </div><br>
      <div class="form-group add-content">
        <label for="content">Add Content</label>
        <textarea #content="ngModel"
[(ngModel)]="post.content" name="content" id="content" type="text" required
class="form-control" rows="8"></textarea>
        <div class="alert alert-danger" *ngIf="content.touched &&
content.invalid">
          <div *ngIf="content.errors?.['required']">Content is
required</div>
        </div>
      </div><br>
      <div class="form-group">
        <label for="image">Image</label>
        <!--<input type="file" ngModel name="image" id="image"
class="form-control" #image required>--><br>
        <input #image="ngModel" id="image" [(ngModel)]="post.image"
name="image" type="text" class="form-control" required url>
        <div class="alert alert-danger" *ngIf="image.touched &&
image.invalid">

```

```

        <div *ngIf="image.errors?.['required']">Image is
required</div>
        <div *ngIf="image.errors?.['url']">Please enter valid
url</div>
    </div>
</div><br>
    <button class="btn btn-primary"
[disabled]="f.invalid">Save</button>
</form>
</div>
<br>
<div class="col-md-5 col-sm-12">
    <h4>Preview</h4>
    <div class="card" style="width: 20rem;">
        <img [src]="post.image" class="card-img-top" >
        <div class="card-body">
            <h5 class="card-title">{{post.title}}</h5>
            <p class="card-text">{{post.content}}</p>
        </div>
    </div>
</div>
</div>
</div>

```

#### new-post.component.ts

```

import { Component, OnInit } from '@angular/core';
import { NgForm } from '@angular/forms';
import { FormGroup, FormBuilder } from '@angular/forms';
import { ActivatedRoute, Router } from '@angular/router';
import { PostModel } from 'src/app/interfaces/post-model';
import { ApiService } from 'src/app/services/api.service';
@Component({
  selector: 'app-new-post',
  templateUrl: './new-post.component.html',
  styleUrls: ['./new-post.component.scss']
})
export class NewPostComponent implements OnInit {
  formValue!: FormGroup;
  postObj: PostModel = new PostModel();
  post: PostModel = {};
  paramId;
  constructor(private formbuilder: FormBuilder, private
restApi: ApiService, private route: Router, private
activatedRouter: ActivatedRoute)

```

```

{
  let uname=localStorage.getItem('user');
  if(uname==="){
    this.route.navigate(['login']);
  }
  this.paramId=this.activatedRouter.snapshot.paramMap.get('id');
  if(this.paramId)
  {
    this.restApi.getSinglePost(+this.paramId).subscribe(res=>this.post=res)
;
  }
}

ngOnInit(): void {
  this.formValue=this.formbuilder.group({
    id: [new Date().valueOf()],
    author:[''],
    title:[''],
    content:[''],
    image:[''],
    createdAt:[new Date()]
  })
}
addNewPost(form:NgForm){
  this.postObj.id= new Date().valueOf(),
  this.postObj.author=localStorage.getItem('user');
  this.postObj.title=form.value.title;
  this.postObj.content=form.value.content;
  this.postObj.image=form.value.image;
  this.postObj.createdAt=new Date();
  this.postObj.userid=localStorage.getItem('user_id');

  if(this.paramId){
    this.restApi.updatePost(this.postObj,+this.paramId).subscribe(res=>{
      alert("Post updated successfully");
      this.route.navigate(["/"]);
    });
  }
  else{

    this.restApi.addPost(this.postObj)
    .subscribe(res=>{
      console.log(res);
      alert("New Post added successfully");
      this.route.navigate(["/"])
    },
    error=>{
      alert("Something went wrong");

```



```

    })
  }

}

}

```

new-post.component.spec.ts

```

import { HttpClientTestingModule } from '@angular/common/http/testing';
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { FormsModule, ReactiveFormsModule } from '@angular/forms';
import { RouterTestingModule } from '@angular/router/testing';
import { NavbarComponent } from '../navbar/navbar.component';

import { NewPostComponent } from './new-post.component';

describe('NewPostComponent', () => {
  let component: NewPostComponent;
  let fixture: ComponentFixture<NewPostComponent>;

  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [ NewPostComponent ,NavbarComponent],
      imports:[RouterTestingModule,HttpClientTestingModule,FormsModule,Reactive
eFormsModule]
    })
    .compileComponents();
  });

  beforeEach(() => {
    fixture = TestBed.createComponent(NewPostComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });

  it('should have label Add Content', () => {
    const fixture = TestBed.createComponent(NewPostComponent);
    fixture.detectChanges();
    const compiled = fixture.nativeElement as HTMLElement;
    expect(compiled.querySelector('.add-content
label')?.textContent).toEqual('Add Content');
  });
});

```

my-posts.component.html

```
<app-navbar></app-navbar>

<div class="container">
  <div class="col">
    <div class="row">
      <ng-container *ngFor="let p of myPost;index as i">
        <div class="col col-md-4 col-sm-12">
          <div class="card" style="width: 18rem;">
            <img [src]="p.image" class="card-img-top" alt="..."
height="200px">
            <div class="card-body">
              <h5 class="card-title">{{p.title | titlecase}}</h5>
              <p class="card-text">{{p.content}}</p>
              <p>Posted by {{p.author}} at {{p.createdAt | date:
'dd/MM/yyyy hh:mm a'}}</p>
              <div class="card-footer">
                <a id="set-width" class="btn btn-primary btn-block
button-width" [routerLink]="['/new-post/',p.id]">Edit Post</a><br><br>
                <button id="set-width" class="btn btn-danger btn-
block button-width" (click)="deletePost(p.id)">Delete Post</button>
              </div>
            </div>
          </div>
        </div>
        <br><br>
        <div *ngIf="(i+1)%3===0" class="w-100"><br></div>
      </ng-container>
    </div>
  </div>
</div>
```

my-posts.component.ts

```
import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { ApiService } from 'src/app/services/api.service';

@Component({
  selector: 'app-my-posts',
  templateUrl: './my-posts.component.html',
  styleUrls: ['./my-posts.component.scss']
})
```

```

export class MyPostsComponent implements OnInit {
  myPost:any;
  constructor(private restApi:ApiService,private route:Router) {
    let uname=localStorage.getItem('user');
    if(uname===""){
      this.route.navigate(['login']);
    }
  }

  ngOnInit(): void {
    this.getMyPost();
  }
  getMyPost(){

    this.restApi.getMyPost(localStorage.getItem('user_id')).subscribe(res=>{
      this.myPost=res;
    })
  }
  deletePost(i:number){
    let res=confirm("Are you sure you want to delete this post?");
    if(res){
      this.restApi.deletePost(i).subscribe(res=>
        { this.getMyPost();
          alert("Post Deleted");
        })
    }
  }
}

```

#### my-posts.component.spec.ts

```

import { HttpClientTestingModule } from '@angular/common/http/testing';
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { RouterTestingModule } from '@angular/router/testing';
import { NavbarComponent } from '../navbar/navbar.component';

import { MyPostsComponent } from './my-posts.component';

describe('MyPostsComponent', () => {
  let component: MyPostsComponent;
  let fixture: ComponentFixture<MyPostsComponent>;

  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [ MyPostsComponent,NavbarComponent ],
      imports:[RouterTestingModule,HttpClientTestingModule]
    })
  })

```



```

        <label for="name" class="">Your name</label>
        <input #name="ngModel" ngModel type="text"
id="name" name="name" class="form-control" required>

    </div>
</div>
<!--Grid column-->

<!--Grid column-->
<div class="col-md-6">
    <div class="md-form mb-0">
        <label for="email" class="">Your email</label>
        <input #email="ngModel" ngModel type="email"
id="email" name="email" class="form-control" required>

    </div>
</div>
<!--Grid column-->

</div>
<!--Grid row-->

<!--Grid row-->
<div class="row">
    <div class="col-md-12">
        <div class="md-form mb-0">
            <label for="subject" class="">Subject</label>
            <input #subject="ngModel" ngModel type="text"
id="subject" name="subject" class="form-control" required>

        </div>
    </div>
</div>
<!--Grid row-->

<!--Grid row-->
<div class="row">

    <!--Grid column-->
    <div class="col-md-12">

        <div class="md-form">
            <label for="message">Your message</label>
            <textarea #message="ngModel" ngModel type="text"
id="message" name="message" rows="2" class="form-control md-textarea"
required></textarea>

        </div>
    </div>

```

```

        </div>
      </div>
      <!--Grid row-->
      <br>
      <div class="text-center text-md-left">
        <button class="btn btn-primary">Send</button>
      </div>
    </form>

    <div class="status"></div>
  </div>
  <!--Grid column-->

  <!--Grid column-->
  <div class="col-md-3 text-center">
    <ul class="list-unstyled mb-0">
      <li>
        <p>Mangalore,Karnataka,India</p>
      </li>

      <li><i class="fa fa-phone mt-4 fa-2x" aria-hidden="true" ></i>
        <p>+91 675542556</p>
      </li>

      <li><i class="fa fa-envelope mt-4 fa-2x"></i>
        <p>contact@foodblog.com</p>
      </li>
    </ul>
  </div>
  <!--Grid column-->

</div>
</section>
<!--Section: Contact v.2-->
</div>

```

contact-form.component.ts

```

import { Component, OnInit } from '@angular/core';
import { NgForm } from '@angular/forms';
import { Router } from '@angular/router';
import { ContactInfo } from 'src/app/interfaces/contact-info';
import { ApiService } from 'src/app/services/api.service';

@Component({

```

```

    selector: 'app-contact-form',
    templateUrl: './contact-form.component.html',
    styleUrls: ['./contact-form.component.scss']
  })
  export class ContactFormComponent implements OnInit {
    contact:ContactInfo=new ContactInfo();
    constructor(private restApi:ApiService,private route:Router) {
      let uname=localStorage.getItem('user');
      if(uname==="){
        this.route.navigate(['login']);
      }
    }

    ngOnInit(): void {
    }
    newContact(form:NgForm){
      // console.log("Hey");
      this.contact.name=form.value.name;
      this.contact.email=form.value.email;
      this.contact.subject=form.value.subject;
      this.contact.message=form.value.message;
      this.restApi.addContact(this.contact)
        .subscribe(res=>{
          console.log(res);
          alert("Message sent successfully");
          this.route.navigate(["/"])
        },
        error=>{
          alert("Something went wrong");
        })
    }
  }
}

```

#### contact-form.component.spec.ts

```

import { HttpClientTestingModule } from '@angular/common/http/testing';
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { FormsModule } from '@angular/forms';
import { RouterTestingModule } from '@angular/router/testing';
import { ApiService } from 'src/app/services/api.service';
import { NavbarComponent } from '../navbar/navbar.component';

import { ContactFormComponent } from './contact-form.component';

describe('ContactFormComponent', () => {

```

```

let component: ContactFormComponent;
let fixture: ComponentFixture<ContactFormComponent>;

beforeEach(async () => {
  await TestBed.configureTestingModule({
    declarations: [ ContactFormComponent ,NavbarComponent],
    imports:[RouterTestingModule,HttpClientTestingModule,FormsModule]
  })
  .compileComponents();
});

beforeEach(() => {
  fixture = TestBed.createComponent(ContactFormComponent);
  component = fixture.componentInstance;
  fixture.detectChanges();
});

it('should create', () => {
  expect(component).toBeTruthy();
});

it('should have text as Contact us', () => {
  const fixture = TestBed.createComponent(ContactFormComponent);
  fixture.detectChanges();
  const compiled = fixture.nativeElement as HTMLElement;
  expect(compiled.querySelector('.contact')?.textContent).toContain('Contact
us');
});
});

```

#### auth.service.ts

```

import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { map } from 'rxjs';

@Injectable({
  providedIn: 'root'
})
export class AuthService {

  constructor(private http:HttpClient) { }

  addUser(data:any){
    return this.http.post<any>("http://localhost:3000/profile",data)
    .pipe(map((res:any)=>{
      return res;
    })))
  }
}

```



```

login(){
  return this.http.get<any>("http://localhost:3000/profile");
}
}

```

#### api.service.ts

```

import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http'
import { map } from 'rxjs';
@Injectable({
  providedIn: 'root'
})
export class ApiService {

  constructor(private http:HttpClient) { }

  addPost(data:any){
    return this.http.post<any>("http://localhost:3000/posts",data)
    .pipe(map((res:any)=>{
      return res;
    })))
  }

  getPost(){
    return this.http.get<any>("http://localhost:3000/posts")
    .pipe(map((res:any)=>{
      return res;
    })))
  }

  updatePost(data:any,id:number){
    return this.http.put<any>("http://localhost:3000/posts/"+id,data)
    .pipe(map((res:any)=>{
      return res;
    })))
  }

  deletePost(id:number){
    return this.http.delete<any>("http://localhost:3000/posts/"+id)
    .pipe(map((res:any)=>{
      return res;
    })))
  }

  getSinglePost(id:number){
    return this.http.get<any>("http://localhost:3000/posts/"+id)
    .pipe(map((res:any)=>{
      return res;
    })))
  }
}

```

```
addContact(data:any){
  return this.http.post<any>("http://localhost:3000/comments",data)
    .pipe(map((res:any)=>{
      return res;
    })))
}
getMyPost(uid:any){
  return this.http.get<any>("http://localhost:3000/posts?q="+uid)
    .pipe(map((res:any)=>{
      return res;
    })))
}
}
```

#### contact-info.ts

```
export class ContactInfo {
  name?:string;
  email?:string;
  subject?:string;
  message?:string;
}
```

#### user.ts

```
export class User {
  id?: number;
  name?:string;
  email?: string;
  password?: string;
}
```

#### post-model.ts

```
export class PostModel {
  id?:number;
  author?:any;
  title?:string;
  content?:string;
  image?:string;
  createdAt?:Date;
```

```
userid?:any;  
}
```

db.json

```
{  
  "posts": [  
    {  
      "id": 1650372059379,  
      "author": "Akshatha Shetty",  
      "title": "Vada at Swagath Hotel",  
      "content": "Vada or bada is a category of savoury fried snacks from India. Different types of vadas can be described variously as fritters, cutlets, doughnuts, or dumplings. Alternative names for this food include wada, vade, vadai, wadeh and bara.",  
      "image":  
      "https://farm1.staticflickr.com/320/18912568232_686613c78c_o.jpg",  
      "createdAt": "2022-04-30T12:40:59.379Z",  
      "userid": "akshatha@gmail.comakshatha"  
    },  
    {  
      "id": 1650442820008,  
      "author": "Akshatha Shetty",  
      "title": "Dominos Pizza",  
      "content": "Pizza is a dish of Italian origin consisting of a usually round, flat base of leavened wheat-based dough topped with tomatoes, cheese, and often various other ingredients, which is then baked at a high temperature, traditionally in a wood-fired oven. Amazing!",  
      "image":  
      "https://static.toiimg.com/thumb/56933159.cms?imgsize=686279&width=800&height=800",  
      "createdAt": "2022-04-30T08:20:20.008Z",  
      "userid": "akshatha@gmail.comakshatha"  
    },  
    {  
      "id": 1650443047894,  
      "author": "Pavan Rai",  
      "title": "Masala Dosa at Iyengars",  
      "content": "Masala dosa is a variation of the popular South Indian dosa, which has its origins in Tuluva Udupi cuisine of Karnataka. It is made from rice, lentils, potato, fenugreek, ghee and curry leaves, and served with chutneys and sambar. Must try!",  
      "image": "https://www.vegrecipesofindia.com/wp-content/uploads/2021/06/masala-dosa-2.jpg",  
      "createdAt": "2022-04-30T08:24:31.529Z",  
      "userid": "pavan@gmail.compavan"  
    }  
  ]  
}
```

```
},
{
  "id": 1650443631342,
  "author": "Pavan Rai",
  "title": "Ideal Icecream Parlor",
  "content": "Mangaluru is truly the pearl of the Arabian Sea and guardian of the West Coast. It is a land of exquisite cultures, charms and cuisines. There are so many exciting things to see, any visit to the city would be incomplete without Ideal Ice Cream.",
  "image": "https://media-cdn.tripadvisor.com/media/photo-s/15/d4/a7/45/img-20181224-164627-largejpg.jpg",
  "createdAt": "2022-05-01T09:10:47.319Z",
  "userid": "pavan@gmail.compavan"
},
{
  "id": 1650443760140,
  "author": "Pavan Rai",
  "title": "Idli at varanga",
  "content": "Idli or idly are a type of savoury rice cake, originating from the Indian subcontinent, popular as breakfast foods in Southern India and in Sri Lanka. The cakes are made by steaming a batter consisting of fermented black lentils and rice",
  "image": "https://static.toiimg.com/thumb/53239358.cms?imgsize=176260&width=800&height=800",
  "createdAt": "2022-05-02T08:36:00.140Z",
  "userid": "pavan@gmail.compavan"
},
{
  "id": 1650445543703,
  "author": "Akshatha Shetty",
  "title": "Panipuri",
  "content": "Panipuri or phuchka or gupchup or golgappa or pani ke patashe is a type of snack that originated in India, and is one of the most common street foods there. You must try panipuri if you visit Mangalore. You can find many stalls. Must try!",
  "image": "https://cdn1.foodviva.com/static-content/food-images/snacks-recipes/pani-puri/pani-puri.jpg",
  "createdAt": "2022-05-01T10:04:08.150Z",
  "userid": "akshatha@gmail.comakshatha"
},
{
  "id": 1651226826774,
  "author": "Aishwarya Shetty",
  "title": "Momos at Tibet Restaurant",
  "content": "Momo is a type of dumpling originally from Tibet which is prominently found in the cuisines of Tibet and Nepal, as well as in the Indian
```

```
states of Ladakh and Sikkim. It has evolved and spread to many countries who
have adapted it and modified as per their preference. ",
  "image": "https://www.foodieist.com/wp-content/uploads/2020/12/veg-
momos-recipe-672x372.jpg",
  "createdAt": "2022-05-02T10:07:06.774Z",
  "userid": "aishu@gmail.comaishu"
},
{
  "id": 1651465290444,
  "author": "Akshatha Shetty",
  "title": "Biriyani at GoldFinch Hotel",
  "content": "Biriyani is a mixed rice dish originating among the Muslims
of the Indian subcontinent. It is made with Indian spices, rice, and a meat
form. Vegetarian versions of the dish replace meat with vegetables such as
potatoes. Must try!!!",
  "image": "https://vegveganmeat.com/wp-content/uploads/2020/12/mushroom-
biriyani-kalan-biriyani-chettinad-south-indian-style-720x540.jpg",
  "createdAt": "2022-05-02T04:23:08.406Z",
  "userid": "akshatha@gmail.comakshatha"
}
],
"comments": [
  {
    "id": 1
  },
  {
    "name": "Akshu",
    "email": "akshu@gmail.com",
    "subject": "message",
    "message": "Needed help with my post",
    "id": 2
  }
],
"profile": [
  {
    "id": 1650392204944,
    "name": "Akshu",
    "email": "akshu@gmail.com",
    "password": "akshu"
  },
  {
    "id": 1650392505293,
    "name": "Pavan Rai",
    "email": "pavan@gmail.com",
    "password": "pavan"
  },
  {
    "id": 1650445410466,
```

```
    "name": "Akshatha Shetty",
    "email": "akshatha@gmail.com",
    "password": "akshatha"
  },
  {
    "id": 1650446268281,
    "name": "Aishwarya Shetty",
    "email": "aishu@gmail.com",
    "password": "aishu"
  },
  {
    "id": 1651226719458,
    "name": "Aishwarya Shetty",
    "email": "aishu@gmail.com",
    "password": "aishu"
  }
]
}
```