

## App.component.spec.ts=

```
import { TestBed } from 'c:/Users/abhis/Downloads/Online Test
Application/phase4_BhawnaGunwani-master/src/app/app.component';
import { AppComponent } from './app.component';

describe('AppComponent', () => {
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [
        AppComponent
      ],
    }).compileComponents();
  });
  it('should create the app', () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app).toBeTruthy();
  });

  it(`should have as title 'online-test-application'`, () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app.title).toEqual('online-test-application');
  });

  it('should render title', () => {
    const fixture = TestBed.createComponent(AppComponent);
    fixture.detectChanges();
    const compiled = fixture.nativeElement;
    expect(compiled.querySelector('.content
span').textContent).toContain('online-test-application app is running!');
  });
})
function expect(app: any) {
  throw new Error('Function not implemented.');
```

## App.component.html=

```
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
  <a class="navbar-brand ml-2 fw-bolder" routerLink="/quiz">Online Quiz</a>
  <button
    class="navbar-toggler"
    type="button"
    data-toggle="collapse"
    data-target="#navbarNav"
```

## App.component.spec.ts=

```
        aria-controls="navbarNav"
        aria-expanded="false"
        aria-label="Toggle navigation"
      >
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav ms-auto">
          <li class="nav-item">
            <a
              class="nav-link btn-outline-primary btn-sm text-light fw-bolder"
              routerLink="/quiz"
            >Home</a>
          </li>
        </ul>
      </div>
    </nav>

    <div class="container">
      <router-outlet></router-outlet>
    </div>

    <footer class="bg-dark text-center text-lg-start mt-5">
      <!-- Copyright -->
      <div class="text-center p-3 text-light fw-bold">
        © 2021 Copyright:
        <a class="text-light fw-bold" routerLink="/quiz">Online Quiz
Application</a>
      </div>
      <!-- Copyright -->
    </footer>
```

## App.component.ts=

```
import { Component } from "c:/Users/abhis/Downloads/Online Test
Application/phase4_BhawnaGunwani-master/src/app/app.component";

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

## App.component.spec.ts=

```
export class AppComponent {  
  title = 'online-test-application';  
}
```

## App.module.ts=

```
import { NgModule } from '@angular/core';  
import { FormsModule } from '@angular/forms';  
import { BrowserModule } from '@angular/platform-browser';  
  
import { AppComponent } from './app.component';  
import { HttpClientModule } from '@angular/common/http';  
import { QuizService } from './services/quiz.service';  
import { QuizListComponent } from './components/quiz-list/quiz-list.component';  
import { QuizComponent } from './components/quiz/quiz.component';  
import { RouterModule, Routes } from '@angular/router';  
  
const routes: Routes = [  
  { path: 'quiz/:quizName', component: QuizComponent },  
  { path: 'quiz', component: QuizListComponent },  
  { path: '', redirectTo: '/quiz', pathMatch: 'full' },  
  { path: '**', redirectTo: '/quiz', pathMatch: 'full' },  
];  
  
@NgModule({  
  declarations: [AppComponent, QuizListComponent, QuizComponent],  
  imports: [  
    BrowserModule,  
    FormsModule,  
    HttpClientModule,  
    RouterModule.forRoot(routes),  
  ],  
  providers: [QuizService],  
  bootstrap: [AppComponent],  
})  
export class AppModule {}
```

## app.e2e.spec.ts=

```
import { browser, logging } from 'protractor';  
import { AppPage } from './app.po';
```

## App.component.spec.ts=

```
describe('workspace-project App', () => {
  let page: AppPage;

  beforeEach(() => {
    page = new AppPage();
  });

  it('should display welcome message', async () => {
    await page.navigateTo();
    expect(await page.getTitleText()).toEqual('online-test-application app is
running!');
  });

  afterEach(async () => {
    // Assert that there are no errors emitted from the browser
    const logs = await browser.manage().logs().get(logging.Type.BROWSER);
    expect(logs).not.toContain(jasmine.objectContaining({
      level: logging.Level.SEVERE,
    } as logging.Entry));
  });
});
```

## App.po.ts=

```
import { browser, by, element } from 'protractor';

export class AppPage {
  async navigateTo(): Promise<unknown> {
    return browser.get(browser.baseUrl);
  }

  async getTitleText(): Promise<string> {
    return element(by.css('app-root .content span')).getText();
  }
}
```