



Create an Angular app (standalone, no NgModules) using **Bootstrap 5 + Bootstrap Icons** with a fixed **Header, Sidebar, Footer**, and **role-based routes** for **procurement, supplier, and admin**.

---

## 1) Create the project

```
# Make sure Angular CLI is installed
npm i -g @angular/cli

# Create a new app (choose: routing = Yes, stylesheet = CSS)
ng new sourcing-tendering --routing --style css
cd sourcing-tendering
```

## 2) Add Bootstrap & Icons

```
npm i bootstrap@5 bootstrap-icons
```

Update **angular.json** `styles` to load CSS (no JS needed):

```
{
  // ...
  "projects": {
    "sourcing-tendering": {
      "architect": {
        "build": {
          "options": {
            "styles": [
              "node_modules/bootstrap/dist/css/bootstrap.min.css",
              "node_modules/bootstrap-icons/font/bootstrap-icons.css",
              "src/styles.css"
            ]
          }
        }
      }
    }
  }
}
```

---

### 3) App structure (files to add)

```
src/
├─ main.ts
├─ app/
│   ├─ app.component.ts
│   ├─ app.routes.ts
│   ├─ models/
│   │   ├─ role.ts
│   │   └─ user.ts
│   ├─ services/
│   │   └─ auth.service.ts
│   ├─ guards/
│   │   └─ role.guard.ts
│   ├─ core/
│   │   ├─ layout.component.ts
│   │   ├─ header.component.ts
│   │   ├─ sidebar.component.ts
│   │   └─ footer.component.ts
│   ├─ pages/
│   │   ├─ login.component.ts
│   │   └─ forbidden.component.ts
│   └─ features/
│       ├─ procurement/procurement-dashboard.component.ts
│       ├─ supplier/supplier-dashboard.component.ts
│       └─ admin/admin-dashboard.component.ts
└─ styles.css
```

You can generate empty files with `ng g c` and then paste code below, or just create them manually.

---

### 4) Routing (role-guarded + lazy components)

src/app/app.routes.ts

```
import { Routes } from '@angular/router';
import { LayoutComponent } from '../core/layout.component';
import { roleGuard } from '../guards/role.guard';

export const routes: Routes = [
  { path: 'login', loadComponent: () => import('../pages/
login.component').then(m => m.LoginComponent) },
```

```

    { path: 'forbidden', loadComponent: () => import('./pages/
forbidden.component').then(m => m.ForbiddenComponent) },

    {
      path: '',
      component: LayoutComponent,
      children: [
        {
          path: 'procurement',
          canMatch: [roleGuard],
          data: { roles: ['procurement'] },
          loadComponent: () => import('./features/procurement/procurement-
dashboard.component')
            .then(m => m.ProcurementDashboardComponent)
        },
        {
          path: 'supplier',
          canMatch: [roleGuard],
          data: { roles: ['supplier'] },
          loadComponent: () => import('./features/supplier/supplier-
dashboard.component')
            .then(m => m.SupplierDashboardComponent)
        },
        {
          path: 'admin',
          canMatch: [roleGuard],
          data: { roles: ['admin'] },
          loadComponent: () => import('./features/admin/admin-
dashboard.component')
            .then(m => m.AdminDashboardComponent)
        },
        { path: '', redirectTo: 'login', pathMatch: 'full' }
      ]
    },

    { path: '**', redirectTo: 'login' }
  ];

```

## 5) Bootstrap application

src/main.ts

```

import { bootstrapApplication } from '@angular/platform-browser';
import { provideRouter } from '@angular/router';

```

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

bootstrapApplication(AppComponent, {
  providers: [provideRouter(routes)]
}).catch(err => console.error(err));
```

src/app/app.component.ts

```
import { Component } from '@angular/core';
import { RouterOutlet } from '@angular/router';

@Component({
  selector: 'app-root',
  standalone: true,
  imports: [RouterOutlet],
  template: `<router-outlet></router-outlet>`
})
export class AppComponent {}
```

---

## 6) Auth models

src/app/models/role.ts

```
export type Role = 'procurement' | 'supplier' | 'admin';
```

src/app/models/user.ts

```
import { Role } from './role';

export interface User {
  name: string;
  role: Role;
}
```

---

## 7) Auth service (very simple demo auth)

src/app/services/auth.service.ts

```

import { Injectable } from '@angular/core';
import { BehaviorSubject } from 'rxjs';
import { User } from '../models/user';
import { Role } from '../models/role';

const STORAGE_KEY = 'currentUser';

@Injectable({ providedIn: 'root' })
export class AuthService {
  private _user$ = new BehaviorSubject<User | null>(this.loadUser());
  readonly user$ = this._user$.asObservable();

  get user(): User | null { return this._user$.value; }

  login(name: string, role: Role): void {
    const user: User = { name, role };
    localStorage.setItem(STORAGE_KEY, JSON.stringify(user));
    this._user$.next(user);
  }

  logout(): void {
    localStorage.removeItem(STORAGE_KEY);
    this._user$.next(null);
  }

  private loadUser(): User | null {
    const raw = localStorage.getItem(STORAGE_KEY);
    try { return raw ? JSON.parse(raw) as User : null; } catch { return null; }
  }
}

```

## 8) Role guard

src/app/guards/role.guard.ts

```

import { inject } from '@angular/core';
import { CanMatchFn, Router } from '@angular/router';
import { AuthService } from '../services/auth.service';
import { Role } from '../models/role';

export const roleGuard: CanMatchFn = (route) => {
  const auth = inject(AuthService);
  const router = inject(Router);
  const expected: Role[] = (route.data?.['roles'] ?? []) as Role[];

```

```

const user = auth.user;
if (!user) {
  router.navigate(['/login']);
  return false;
}
if (expected.length && !expected.includes(user.role)) {
  router.navigate(['/forbidden']);
  return false;
}
return true;
};

```

## 9) Core layout

src/app/core/layout.component.ts

```

import { Component } from '@angular/core';
import { RouterOutlet } from '@angular/router';
import { HeaderComponent } from '../header.component';
import { SidebarComponent } from '../sidebar.component';
import { FooterComponent } from '../footer.component';

@Component({
  selector: 'app-layout',
  standalone: true,
  imports: [RouterOutlet, HeaderComponent, SidebarComponent, FooterComponent],
  template: `
    <div class="d-flex flex-column min-vh-100">
      <app-header (menuToggle)="toggleSidebar()"></app-header>

      <div class="container-fluid flex-grow-1">
        <div class="row flex-nowrap h-100">
          <div class="col-auto px-0">
            <app-sidebar [collapsed]="collapsed"></app-sidebar>
          </div>
          <main class="col py-3">
            <router-outlet></router-outlet>
          </main>
        </div>
      </div>

      <app-footer></app-footer>
    </div>
  `
})

```

```

    })
    export class LayoutComponent {
        collapsed = false;
        toggleSidebar() { this.collapsed = !this.collapsed; }
    }

```

src/app/core/header.component.ts

```

import { Component, EventEmitter, Output } from '@angular/core';
import { CommonModule } from '@angular/common';
import { Router, RouterLink } from '@angular/router';
import { AuthService } from '../../services/auth.service';

@Component({
    selector: 'app-header',
    standalone: true,
    imports: [CommonModule, RouterLink],
    template: `
        <header class="navbar navbar-expand-lg bg-light border-bottom">
            <div class="container-fluid">
                <button class="btn btn-outline-secondary me-2" (click)="menuToggle.emit()"
                    aria-label="Toggle menu">
                    <i class="bi bi-list"></i>
                </button>
                <a class="navbar-brand fw-semibold" [routerLink]="['/']">Sourcing &
                    Tendering</a>

                <div class="ms-auto d-flex align-items-center gap-3">
                    <ng-container *ngIf="auth.user as user; else guest">
                        <span class="text-muted small">Signed in as <strong>{{ user.name }}</strong>
                        <strong> ({{ user.role }})</span>
                        <a class="btn btn-sm btn-outline-danger" (click)="logout()">
                            <i class="bi bi-box-arrow-right me-1"></i> Logout
                        </a>
                    </ng-container>
                    <ng-template #guest>
                        <a class="btn btn-sm btn-primary" [routerLink]="['/login']">
                            <i class="bi bi-box-arrow-in-right me-1"></i> Login
                        </a>
                    </ng-template>
                </div>
            </div>
        </header>
    `
})

```

```

export class HeaderComponent {
  @Output() menuToggle = new EventEmitter<void>();
  constructor(public auth: AuthService, private router: Router) {}
  logout() {
    this.auth.logout();
    this.router.navigate(['/login']);
  }
}

```

src/app/core/sidebar.component.ts

```

import { Component, Input } from '@angular/core';
import { CommonModule } from '@angular/common';
import { RouterLink, RouterLinkActive } from '@angular/router';
import { AuthService } from '../services/auth.service';

@Component({
  selector: 'app-sidebar',
  standalone: true,
  imports: [CommonModule, RouterLink, RouterLinkActive],
  styles: [
    .sidebar { width: 260px; transition: width .2s ease; }
    .sidebar.collapsed { width: 72px; }
    .sidebar .nav-link { border-radius: .5rem; }
    .sidebar.collapsed .link-text { display: none; }
  ],
  template: `
    <aside class="sidebar bg-body-tertiary border-end h-100
p-3" [class.collapsed]="collapsed">
      <ng-container *ngIf="auth.user as user; else noUser">
        <div class="text-uppercase text-muted small mb-2">Menu</div>

        <!-- Procurement Menu -->
        <ul class="nav nav-pills flex-column gap-1" *ngIf="user.role ===
'procurement'">
          <li class="nav-item">
            <a routerLink="/procurement" routerLinkActive="active" class="nav-
link d-flex align-items-center">
              <i class="bi bi-speedometer2 me-2"></i>
              <span class="link-text">Dashboard</span>
            </a>
          </li>
          <li class="nav-item">
            <a class="nav-link d-flex align-items-center disabled"
title="Sample">
              <i class="bi bi-receipt me-2"></i>

```



```

        <span class="link-text">RFQs</span>
      </a>
    </li>
  </ul>

  <!-- Supplier Menu -->
  <ul class="nav nav-pills flex-column gap-1" *ngIf="user.role ===
'supplier'">
    <li class="nav-item">
      <a routerLink="/supplier" routerLinkActive="active" class="nav-link
d-flex align-items-center">
        <i class="bi bi-briefcase me-2"></i>
        <span class="link-text">Dashboard</span>
      </a>
    </li>
    <li class="nav-item">
      <a class="nav-link d-flex align-items-center disabled">
        <i class="bi bi-file-earmark-text me-2"></i>
        <span class="link-text">My Bids</span>
      </a>
    </li>
  </ul>

  <!-- Admin Menu -->
  <ul class="nav nav-pills flex-column gap-1" *ngIf="user.role ===
'admin'">
    <li class="nav-item">
      <a routerLink="/admin" routerLinkActive="active" class="nav-link d-
flex align-items-center">
        <i class="bi bi-gear me-2"></i>
        <span class="link-text">Dashboard</span>
      </a>
    </li>
    <li class="nav-item">
      <a class="nav-link d-flex align-items-center disabled">
        <i class="bi bi-people me-2"></i>
        <span class="link-text">Users</span>
      </a>
    </li>
  </ul>
</ng-container>
<ng-template #noUser>
  <div class="text-muted small">Please login to see the menu.</div>
</ng-template>
</aside>
,
  })
export class SidebarComponent {

```

```
@Input() collapsed = false;
constructor(public auth: AuthService) {}
}
```

src/app/core/footer.component.ts

```
import { Component } from '@angular/core';
import { CommonModule } from '@angular/common';

@Component({
  selector: 'app-footer',
  standalone: true,
  imports: [CommonModule],
  template: `
    <footer class="border-top bg-light py-3 mt-auto">
      <div class="container d-flex justify-content-between">
        <span class="text-muted">© {{ year }} Sourcing & Tendering</span>
        <span class="text-muted">v1.0 • Bootstrap 5</span>
      </div>
    </footer>
  `,
})
export class FooterComponent { year = new Date().getFullYear(); }
```

---

## 10) Login & Forbidden pages

src/app/pages/login.component.ts

```
import { Component } from '@angular/core';
import { CommonModule } from '@angular/common';
import { FormsModule } from '@angular/forms';
import { Router } from '@angular/router';
import { AuthService } from '../../services/auth.service';
import { Role } from '../../models/role';

@Component({
  selector: 'app-login',
  standalone: true,
  imports: [CommonModule, FormsModule],
  template: `
    <div class="container py-5" style="max-width: 520px;">
      <h1 class="h3 mb-4">Sign in</h1>
      <form (ngSubmit)="submit()" #f="ngForm" class="card p-4 shadow-sm">

```

```

    <div class="mb-3">
      <label class="form-label">Your name</label>
      <input class="form-control" name="name" [(ngModel)]="name" required
placeholder="e.g. Ali" />
    </div>
    <div class="mb-4">
      <label class="form-label">Role</label>
      <select class="form-select" name="role" [(ngModel)]="role" required>
        <option [ngValue]="undefined">Choose a role...</option>
        <option value="procurement">Procurement</option>
        <option value="supplier">Supplier</option>
        <option value="admin">Admin</option>
      </select>
    </div>
    <div class="d-flex gap-2">
      <button class="btn btn-primary" [disabled]="!f.form.valid"
type="submit">
        <i class="bi bi-box-arrow-in-right me-1"></i> Continue
      </button>
    </div>
  </form>
</div>
,
})
export class LoginComponent {
  name = '';
  role: Role | undefined = undefined;
  constructor(private auth: AuthService, private router: Router) {}
  submit() {
    if (!this.role) return;
    this.auth.login(this.name || 'Guest', this.role);
    this.router.navigate(['/${this.role}']);
  }
}

```

```
src/app/pages/forbidden.component.ts
```

```

import { Component } from '@angular/core';
import { CommonModule } from '@angular/common';
import { RouterLink } from '@angular/router';

@Component({
  selector: 'app-forbidden',
  standalone: true,
  imports: [CommonModule, RouterLink],
  template: `

```

```

    <div class="container py-5">
      <div class="alert alert-warning" role="alert">
        <i class="bi bi-shield-lock me-2"></i>
        You don't have permission to view that page.
      </div>
      <a routerLink="/login" class="btn btn-outline-primary">
        Go to login
      </a>
    </div>
  ,
  })
  export class ForbiddenComponent {}

```

## 11) Feature dashboards

src/app/features/procurement/procurement-dashboard.component.ts

```

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

@Component({
  selector: 'app-procurement-dashboard',
  standalone: true,
  imports: [CommonModule],
  template: `
    <h2 class="h4 mb-3">Procurement Dashboard</h2>
    <div class="row g-3">
      <div class="col-12 col-md-6 col-xl-4">
        <div class="card shadow-sm">
          <div class="card-body">
            <div class="d-flex align-items-center gap-2">
              <i class="bi bi-receipt fs-4"></i>
              <div>
                <div class="fw-semibold">Open RFQs</div>
                <div class="text-muted">12</div>
              </div>
            </div>
          </div>
        </div>
      </div>
      <div class="col-12 col-md-6 col-xl-4">
        <div class="card shadow-sm">
          <div class="card-body">
            <div class="fw-semibold mb-1">Tenders Due This Week</div>

```

```

        <div class="text-muted">3 deadlines</div>
      </div>
    </div>
  </div>
</div>
,
}))
export class ProcurementDashboardComponent {}

```

src/app/features/supplier/supplier-dashboard.component.ts

```

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

@Component({
  selector: 'app-supplier-dashboard',
  standalone: true,
  imports: [CommonModule],
  template: `
    <h2 class="h4 mb-3">Supplier Dashboard</h2>
    <div class="row g-3">
      <div class="col-12 col-md-6 col-xl-4">
        <div class="card shadow-sm">
          <div class="card-body">
            <div class="fw-semibold mb-1">Active Bids</div>
            <div class="text-muted">5</div>
          </div>
        </div>
      </div>
      <div class="col-12 col-md-6 col-xl-4">
        <div class="card shadow-sm">
          <div class="card-body">
            <div class="fw-semibold mb-1">Documents Expiring</div>
            <div class="text-muted">2 certificates</div>
          </div>
        </div>
      </div>
    </div>
  `
})
export class SupplierDashboardComponent {}

```

src/app/features/admin/admin-dashboard.component.ts

```

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

@Component({
  selector: 'app-admin-dashboard',
  standalone: true,
  imports: [CommonModule],
  template: `
    <h2 class="h4 mb-3">Admin Dashboard</h2>
    <div class="row g-3">
      <div class="col-12 col-md-6 col-xl-4">
        <div class="card shadow-sm">
          <div class="card-body">
            <div class="fw-semibold mb-1">Users</div>
            <div class="text-muted">128 total</div>
          </div>
        </div>
      </div>
      <div class="col-12 col-md-6 col-xl-4">
        <div class="card shadow-sm">
          <div class="card-body">
            <div class="fw-semibold mb-1">Pending Approvals</div>
            <div class="text-muted">4 requests</div>
          </div>
        </div>
      </div>
    </div>
  `
})
export class AdminDashboardComponent {}

```

## 12) Global styles (layout polish)

src/styles.css

```

html, body { height: 100%; }
body { background-color: #f8f9fa; }

/* Make the main area fill height next to sidebar */
.container-fluid .row { min-height: calc(100vh - 56px - 56px); /* header + footer approx */ }

/* Optional: nicer scroll for sidebar */
.sidebar { overflow: auto; max-height: calc(100vh - 56px - 56px); }

```

```
/* Cards look a bit lifted */  
.card { border-radius: 1rem; }
```

---

## 13) Run it

```
npm start  
# then open http://localhost:4200  
# 1) Go to /login  
# 2) Choose a role and name  
# 3) You'll be redirected to /procurement or /supplier or /admin
```

---

## 14) Next steps (optional)

- Replace the disabled sidebar items with real routes and components.
- Wire real authentication (JWT) instead of the simple localStorage demo.
- Move menu config to a JSON/TS constant and render dynamically.
- Add guards for canActivate (if you prefer) and HTTP interceptors.
- Add breadcrumbs and page titles per route.

---

## What you get

- Clean **Bootstrap 5** layout (header / sidebar / footer)
- **Role-based routing** for procurement, supplier, admin
- **Standalone Angular** components with lazy loading
- Minimal, readable CSS and HTML you can extend