



Angular Essentials

Lab Guides


Document Code	25e-BM/HR/HDCV/FSOFT
Version	1.1
Effective Date	30/06/2020

RECORD OF CHANGES

No	Effective Date	Change Description	Reason	Reviewer	Approver

Table of Contents

Objectives	4
Technical Requirements:.....	4
Specifications	4
Guidelines.....	4
Step 1: Create project structure	4
Step 2: Open project in Visual Studio Code	5
Step 3: Change to MobileStore Directory	6
Step 4: run project	6
Step 5: Access MobileStore website by google chrome	7
Step 6: Create login component	7
Step 7: Create header component	7
Step 8: Add Bootstrap 4	8
Step 9: Open app.module.ts file and insert code	8
Step 10: Open app.component.ts file and insert code	9
Step 11: Open app.component.html file and insert code	9
Step 12: Open app-routing.module.ts file and insert code	9
Step 13: Open header.component.html file and insert code	10
Step 14: Open header.component.ts file and insert code	11
Step 15: Create model.ts file and insert code	11
Step 16: Create authentication.service.ts file and insert code	11
Step 17: Open login.component.html file and insert code.....	13
Step 18: Open login.component.ts file and insert code.....	13
Step 19: Run project again	14

	<table><tr><td>CODE:</td><td>Angular</td></tr><tr><td>TYPE:</td><td>Medium</td></tr><tr><td>LOC:</td><td></td></tr><tr><td>DURATION:</td><td>90 MINUTES</td></tr></table>	CODE:	Angular	TYPE:	Medium	LOC:		DURATION:	90 MINUTES
CODE:	Angular								
TYPE:	Medium								
LOC:									
DURATION:	90 MINUTES								

Objectives

- Understand the basic concepts of TypeScript
- Understand Type Systems of TypeScript
- Understand Angular architecture
- Understand Interfaces/Class
- Understand Functions
- Understand Generics
- Understand Decorators
- Understand Service (HTTPClient)
- Import/export

Technical Requirements:

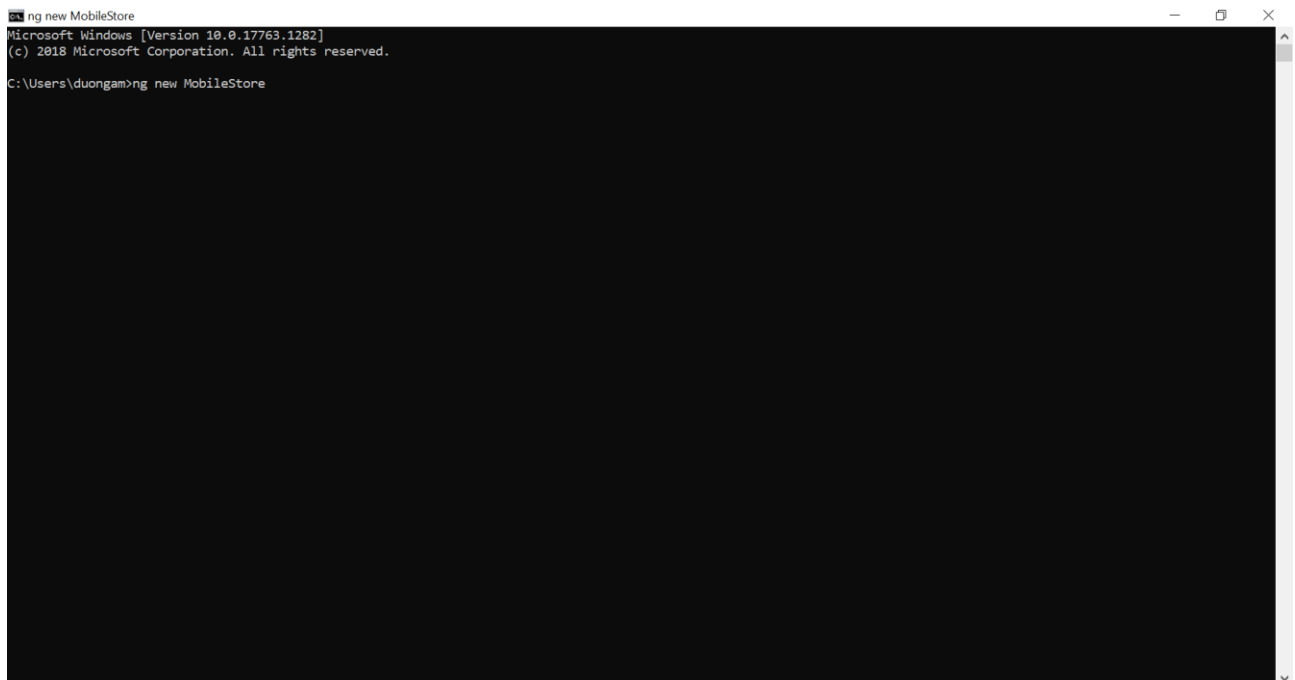
- Must use TypeScript

Specifications

Guidelines

Step 1: Create project structure

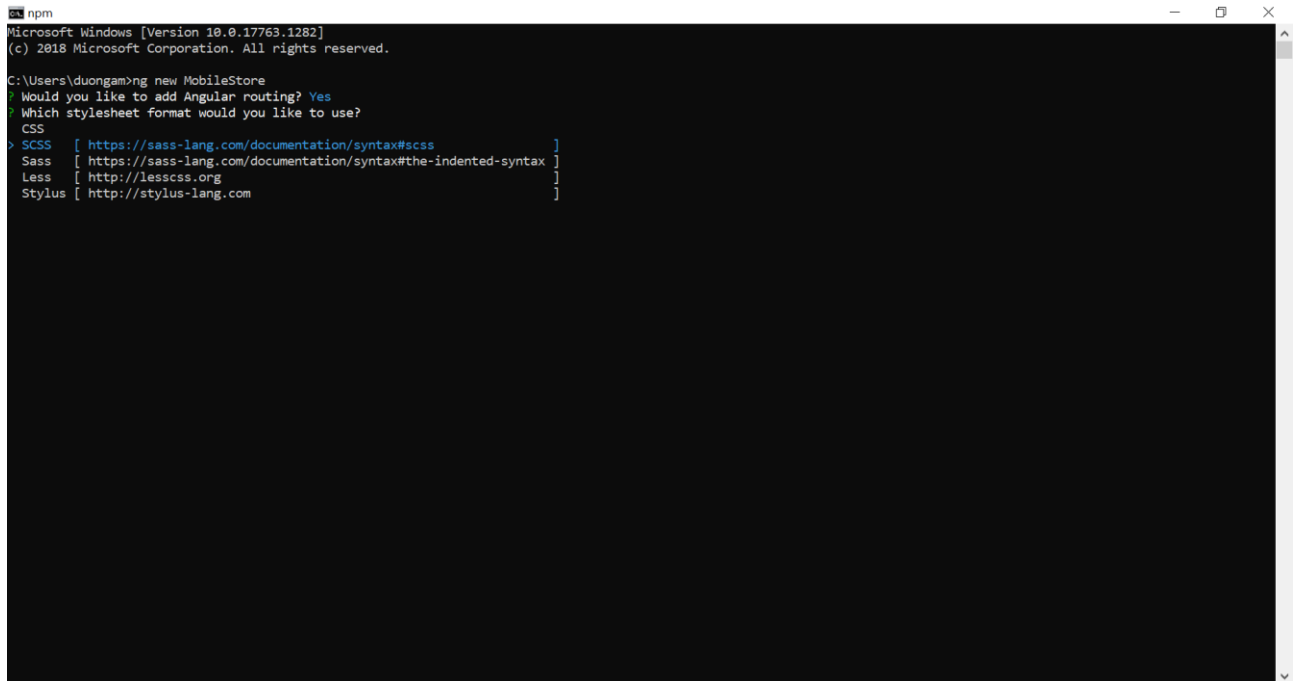
- Open “cmd”
- Typing “ng new MobileStore”



```
ng new MobileStore
Microsoft Windows [Version 10.0.17763.1282]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\duongam>ng new MobileStore
```

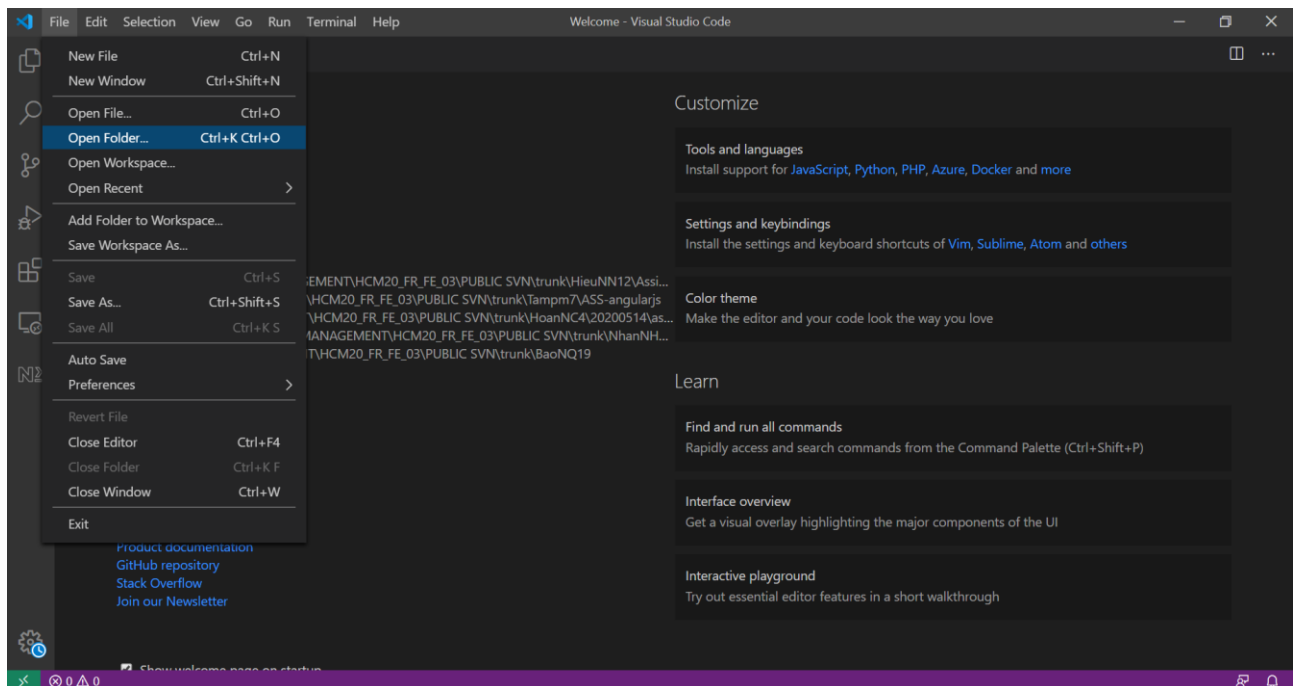
- Choose “SCSS” → Enter



```
npm
Microsoft Windows [Version 10.0.17763.1282]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\duongam>ng new MobileStore
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use?
  CSS
> SCSS [ https://sass-lang.com/documentation/syntax#scss ]
  Sass [ https://sass-lang.com/documentation/syntax#the-indented-syntax ]
  Less  [ http://lesscss.org ]
  Stylus [ http://stylus-lang.com ]
```

Step 2: Open project in Visual Studio Code



Step 3: Change to MobileStore Directory

- typing “cd MobileStore”

```
ng serve --open
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in package.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/app/app-routing.module.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/app/app.component.html.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/app/app.component.spec.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/app/app.component.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/app/app.module.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/environments/environment.prod.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/environments/environment.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/index.html.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/main.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/polyfills.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/styles.scss.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/test.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in tsconfig.app.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in tsconfig.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in tsconfig.spec.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in tslint.json.
The file will have its original line endings in your working directory
Successfully initialized git.

C:\Users\duongam>cd MobileStore

C:\Users\duongam\MobileStore>ng serve --open
```

Step 4: run project

- Typing “ng serve --open”

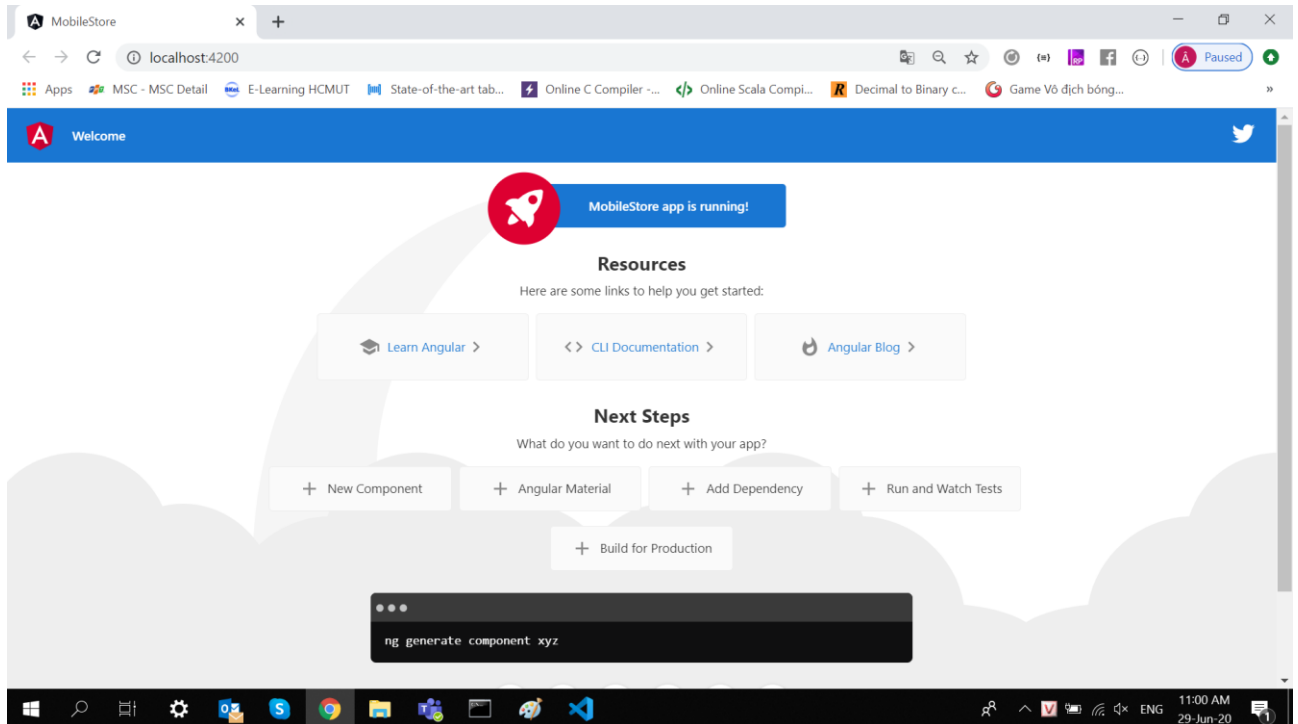
```
ng serve --open
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/environments/environment.prod.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/environments/environment.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/index.html.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/main.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/polyfills.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/styles.scss.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/test.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in tsconfig.app.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in tsconfig.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in tsconfig.spec.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in tslint.json.
The file will have its original line endings in your working directory
Successfully initialized git.

C:\Users\duongam>cd MobileStore

C:\Users\duongam\MobileStore>ng serve --open
10% building 3/3 modules 0 active @wdwds: Project is running at http://localhost:4200/webpack-dev-server/
i @wdwds: webpack output is served from /
i @wdwds: 404s will fallback to //index.html

chunk {main} main.js, main.js.map (main) 50.7 kB [initial] [rendered]
chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 269 kB [initial] [rendered]
chunk {runtime} runtime.js, runtime.js.map (runtime) 6.15 kB [entry] [rendered]
chunk {styles} styles.js, styles.js.map (styles) 10.1 kB [initial] [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 4.09 MB [initial] [rendered]
Date: 2020-06-29T03:51:37.687Z - Hash: 54f3934c98d41a660e98 - Time: 42761ms
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **
i @wdwds: Compiled successfully.
```

Step 5: Access MobileStore website by google chrome



Step 6: Create login component

- Typing “ng generate component login”

```
Select ng serve --open
The file will have its original line endings in your working directory
Successfully initialized git.

C:\Users\duongam>cd MobileStore

C:\Users\duongam\MobileStore>ng serve --open
10% building 3/3 modules 0 active @wdm: Project is running at http://localhost:4200/webpack-dev-server/
i @wdm: webpack output is served from /
i @wdm: 404s will fallback to //index.html

chunk {main} main.js, main.js.map (main) 50.7 kB [initial] [rendered]
chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 269 kB [initial] [rendered]
chunk {runtime} runtime.js, runtime.js.map (runtime) 6.15 kB [entry] [rendered]
chunk {styles} styles.js, styles.js.map (styles) 10.1 kB [initial] [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 4.09 MB [initial] [rendered]
Date: 2020-06-29T03:51:37.687Z - Hash: 54f3934c98d41a660e98 - Time: 42761ms
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **
i @wdm: Compiled successfully.
^CTerminate batch job (Y/N)? y

C:\Users\duongam\MobileStore>ng generate component login
CREATE src/app/login/login.component.html (20 bytes)
CREATE src/app/login/login.component.spec.ts (621 bytes)
CREATE src/app/login/login.component.ts (266 bytes)
CREATE src/app/login/login.component.scss (0 bytes)
UPDATE src/app/app.module.ts (471 bytes)

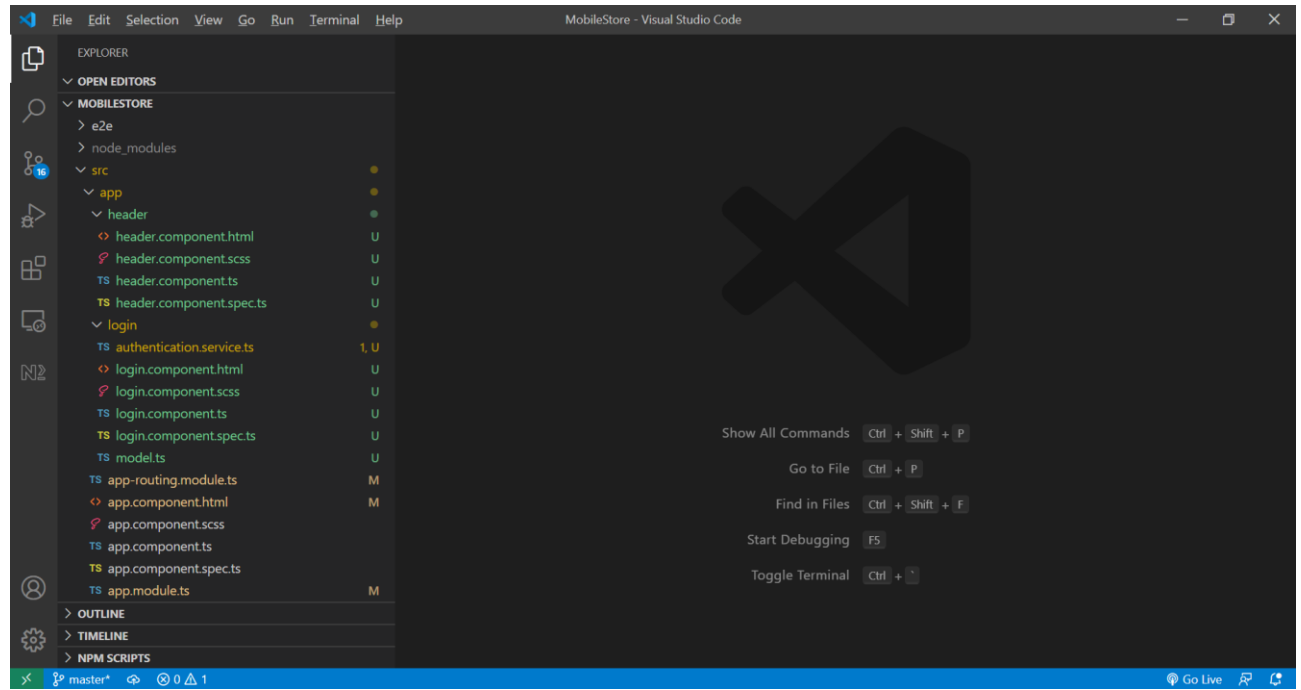
C:\Users\duongam\MobileStore>ng serve --open
10% building 3/3 modules 0 active @wdm: Project is running at http://localhost:4200/webpack-dev-server/
i @wdm: webpack output is served from /
i @wdm: 404s will fallback to //index.html

chunk {main} main.js, main.js.map (main) 53.6 kB [initial] [rendered]
chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 269 kB [initial] [rendered]
chunk {runtime} runtime.js, runtime.js.map (runtime) 6.15 kB [entry] [rendered]
chunk {styles} styles.js, styles.js.map (styles) 10.1 kB [initial] [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 4.09 MB [initial] [rendered]
Date: 2020-06-29T04:26:07.542Z - Hash: dfbc06c1325fcd9fcea0 - Time: 22758ms
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **
i @wdm: Compiled successfully.
```

Step 7: Create header component

- Typing “ng generate component header” into cmd

➔ Current project like this Image:



Step 8: Add Bootstrap 4

- Typing “npm install --save bootstrap” into cmd
- Open “Styles.css” file and insert code

```
1. /* You can add global styles to this file, and also import other style files
   */
2. @import '~bootstrap/dist/css/bootstrap.css';
```

Step 9: Open app.module.ts file and insert code

```
1. import { BrowserModule } from '@angular/platform-browser';
2. import { NgModule } from '@angular/core';
3.
4. import { AppRoutingModule } from './app-routing.module';
5. import { AppComponent } from './app.component';
6. import { LoginComponent } from './login/login.component';
7. import { HeaderComponent } from './header/header.component';
8. import { HttpClientModule } from '@angular/common/http';
9. import { FormsModule } from '@angular/forms';
10.
11. @NgModule({
12.   declarations: [
13.     AppComponent,
14.     LoginComponent,
15.     HeaderComponent
```



```
16. ],
17. imports: [
18.   BrowserModule,
19.   AppRoutingModule,
20.   FormsModule,
21.   HttpClientModule,
22. ],
23. providers: [],
24. bootstrap: [AppComponent]
25.})
26.export class AppModule { }
```

Step 10: Open app.component.ts file and insert code

```
1. import { Component } from '@angular/core';
2.
3. @Component({
4.   selector: 'app-root',
5.   templateUrl: './app.component.html',
6.   styleUrls: ['./app.component.scss']
7. })
8. export class AppComponent {
9.   title = 'MobileStore';
10.}
```

Step 11: Open app.component.html file and insert code

```
1. <router-outlet></router-outlet>
```

Step 12: Open app-routing.module.ts file and insert code

```
1. import { NgModule } from '@angular/core';
2. import { Routes, RouterModule } from '@angular/router';
3. import { LoginComponent } from '../login/login.component';
4.
5. const routes: Routes = [
6.   {
7.     path: 'login',
8.     component: LoginComponent
9.   },
10. ];
11.
12. @NgModule({
13.   imports: [RouterModule.forRoot(routes)],
14.   exports: [RouterModule]
15. })
16. export class AppRoutingModule { }
```

Step 13: Open header.component.html file and insert code

```
1. <div
2.   class="container-fluid"
3.   style="
4.     height: 25vh;
5.
6.     background-color: rgb(238, 238, 238);
7.   "
8. >
9.   <div class="container">
10.    <div class="row">
11.      <div class="col">
12.        <h1 style="font-size: 5rem; padding-top: 50px;">
13.          {{ title }}
14.        </h1>
15.        <span class="lead">{{ subtitle }}</span>
16.      </div>
17.      <div class="col text-right">
18.        <button
19.          *ngIf="button === 'logout'"
20.          type="button"
21.          style="margin-top: 30px;"
22.          class="btn btn-danger"
23.          (click)='onLogout()'
24.        >
25.          logout
26.        </button>
27.        <button
28.          *ngIf="button === 'view-cart'"
29.          type="button"
30.          style="margin-top: 30px;"
31.          class="btn btn-light"
32.          routerLink="/cart"
33.        >
34.          <i class="fas fa-shopping-
35.            cart" style="transform: rotateY(180deg) !important;" ></i>
36.          view cart
37.        </button>
38.      </div>
39.    </div>
40.  </div>
41.
42. <br />
```

Step 14: Open header.component.ts file and insert code

```
1. import { Component, OnInit, Input } from '@angular/core';
2.
3. import { Router } from '@angular/router';
4. import { AuthenticationService } from '../login/authentication.service';
5.
6. @Component({
7.   selector: 'app-header',
8.   templateUrl: './header.component.html',
9.   styleUrls: ['./header.component.scss'],
10. })
11. export class HeaderComponent implements OnInit {
12.   @Input() title = '';
13.   @Input() subtitle = '';
14.   @Input() button = '';
15.
16.   constructor(
17.     private authenticationService: AuthenticationService,
18.     private router: Router
19.   ) {}
20.
21.   ngOnInit(): void {}
22.   public onLogout = () => {
23.     this.authenticationService.logout();
24.     this.router.navigateByUrl('/login');
25.   }
26. }
```

Step 15: Create model.ts file and insert code

```
1. export class User {
2.   id: Int32Array;
3.   username: string;
4.   password: string;
5. }
```

Step 16: Create authentication.service.ts file and insert code

```
1. import { Injectable } from '@angular/core';
2. import { BehaviorSubject, Observable } from 'rxjs';
3. import { User } from './model';
4. import { HttpClient, HttpHeaders } from '@angular/common/http';
5. import { map } from 'rxjs/operators';
6. import { Console } from 'console';
7.
8. @Injectable({
9.   providedIn: 'root',
10. })
```

```
11. export class AuthenticationService {
12.   private currentUserSubject: BehaviorSubject<User>;
13.   public currentUser: Observable<User>;
14.   private urlAPI = 'http://localhost:8080';
15.   constructor(private http: HttpClient) {
16.     this.currentUserSubject = new BehaviorSubject<User>(
17.       JSON.parse(localStorage.getItem('currentUser'))
18.     );
19.     this.currentUser = this.currentUserSubject.asObservable();
20.   }
21.
22.   public get currentUserValue(): User {
23.     return this.currentUserSubject.value;
24.   }
25.
26.   public login = (username: string, password: string) => {
27.     console.log(username);
28.     console.log(password);
29.     const loginUrl = `${this.urlAPI}/api/v1/user/signin`;
30.     console.log(loginUrl);
31.     return this.http
32.       .post<any>(loginUrl, {
33.         username,
34.         password,
35.       })
36.       .pipe(
37.         map((user) => {
38.           // console.log(user);
39.           if (user != null){
40.             const newUser = {} as User;
41.             newUser.id = user.id;
42.             newUser.username = user.username;
43.             newUser.password = user.password;
44.             this.currentUserSubject.next(newUser);
45.             return user;
46.           } else {
47.             return null;
48.           }
49.         })
50.       );
51.   }
52.
53.   public logout = () => {
54.     localStorage.removeItem('username');
55.     localStorage.removeItem('password');
56.     this.currentUserSubject.next(null);
57.   }
58. }
```

Step 17: Open login.component.html file and insert code

```
1. <app-header title="Mobile Store" subtitle="" button=""></app-header>
2. <div class="container">
3.   <div class="card" style="width: 30vw; margin: 1rem auto; min-width: 24rem;">
4.     <div class="card-header">
5.       Please sign in
6.     </div>
7.     <div class="card-body">
8.       <form (ngSubmit)="login()" #loginForm="ngForm">
9.         <div class="form-group">
10.          <input
11.            type="text"
12.            class="form-control"
13.            id="inputUser"
14.            placeholder="User Name"
15.            [(ngModel)]= "userName"
16.            [ngModelOptions]="{ standalone: true }"
17.            required
18.          />
19.        </div>
20.        <div class="form-group">
21.          <input
22.            type="password"
23.            class="form-control"
24.            id="inputPassword"
25.            placeholder="Password"
26.            [(ngModel)]= "password"
27.            [ngModelOptions]="{ standalone: true }"
28.          />
29.        </div>
30.        <button type="submit" class="btn btn-success btn-md btn-block">
31.          Login
32.        </button>
33.      </form>
34.    </div>
35.  </div>
36.</div>
```

Step 18: Open login.component.ts file and insert code

```
1. import { Component, OnInit } from '@angular/core';
2. import { AuthenticationService } from '../authentication.service';
3. import { Router } from '@angular/router';
4.
5. @Component({
6.   selector: 'app-login',
7.   templateUrl: './login.component.html',
8.   styleUrls: ['./login.component.scss']
9. })
10. export class LoginComponent implements OnInit {
11.   constructor(private authService: AuthenticationService, private router: Router) {}
12.   ngOnInit(): void {}
13.   login(): void {}
14. }
```

```
9. })
10. export class LoginComponent implements OnInit {
11.
12.   public userName = 'aumauduong';
13.   public password = 'aumauduong';
14.
15.   constructor(
16.     private authenticationService: AuthenticationService,
17.     private router: Router
18.   ) {}
19.
20.   login = () => {
21.     this.authenticationService.login(this.userName, this.password).subscribe(
22.       (data) => {
23.         if (data != null && data.username) {
24.           localStorage.setItem('username', data.username);
25.           localStorage.setItem('password', data.password);
26.           console.log('login Success');
27.           // this.router.navigateByUrl('/productList');
28.         } else {
29.           console.log('login fail');
30.         }
31.       },
32.       (err) => console.error(err)
33.     );
34.   };
35.
36.   ngOnInit(): void {}
37.
38. }
```

Step 19: Run project again

- Typing “ng serve --open”