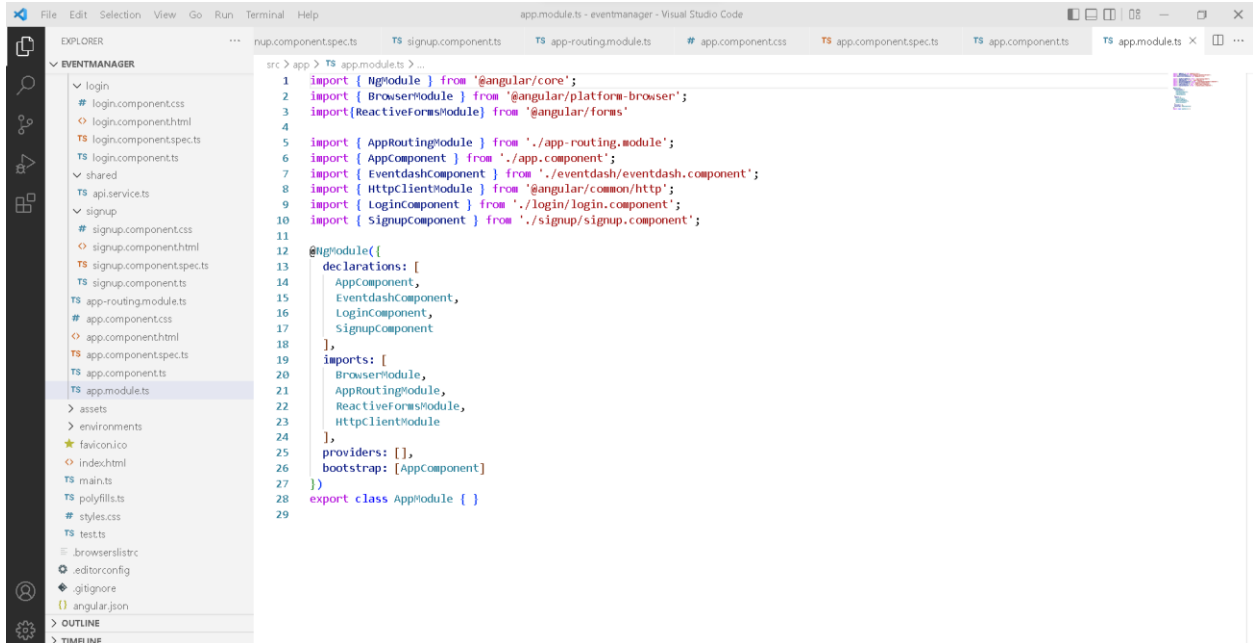


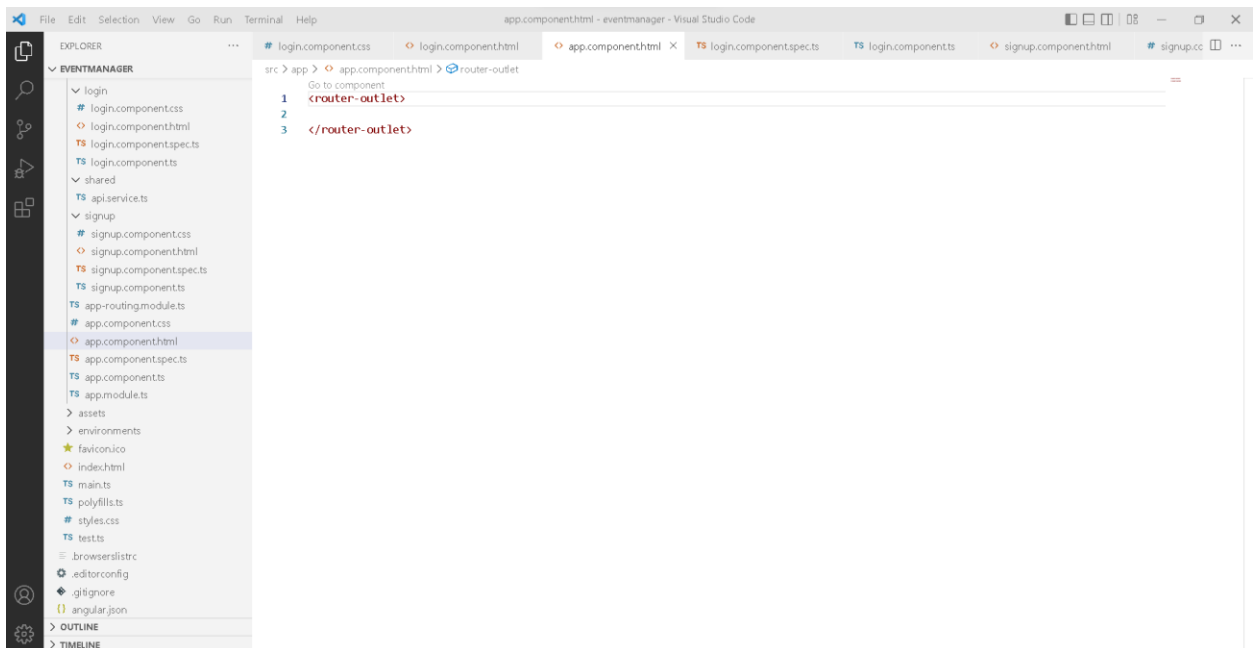
Event Management App

Code:



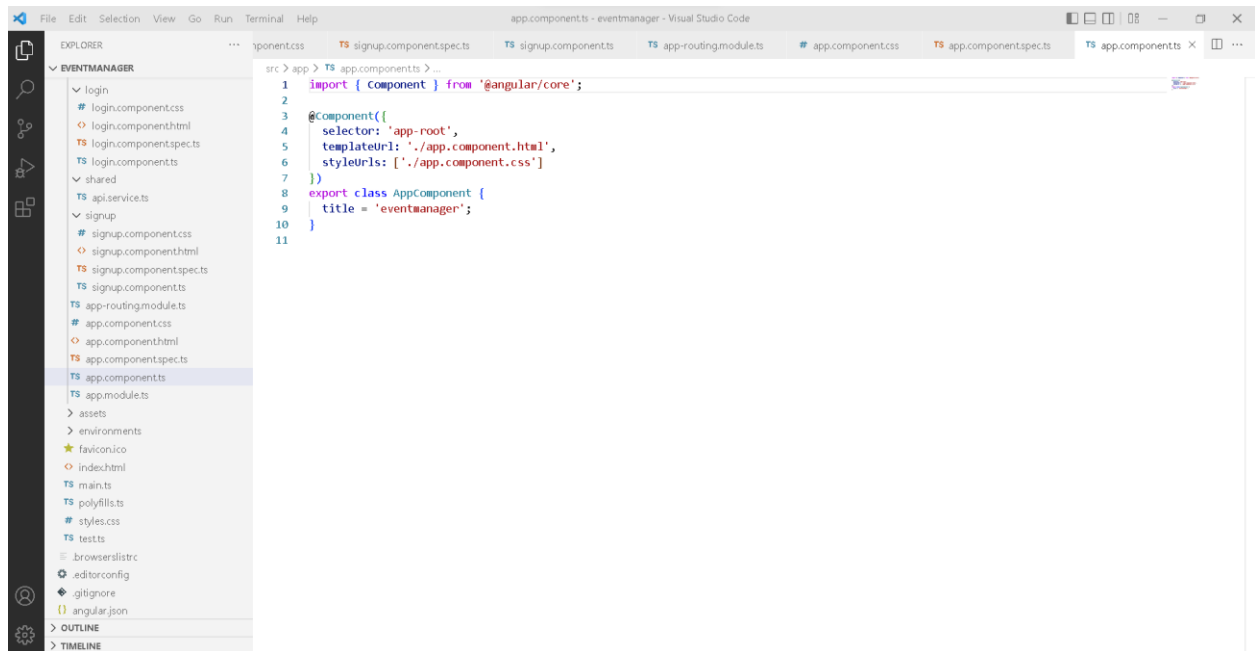
The screenshot shows the Visual Studio Code editor with the 'app.module.ts' file open. The Explorer sidebar on the left shows the project structure for 'EVENTMANAGER', including folders like 'login', 'shared', 'signup', and 'app'. The main editor area displays the following TypeScript code:

```
src > app > TS app.module.ts > ...
1 import { NgModule } from '@angular/core';
2 import { BrowserModule } from '@angular/platform-browser';
3 import { ReactiveFormsModule } from '@angular/forms'
4
5 import { AppRoutingModule } from './app-routing.module';
6 import { AppComponent } from './app.component';
7 import { EventdashComponent } from './eventdash/eventdash.component';
8 import { HttpClientModule } from '@angular/common/http';
9 import { LoginComponent } from './login/login.component';
10 import { SignupComponent } from './signup/signup.component';
11
12 @NgModule({
13   declarations: [
14     AppComponent,
15     EventdashComponent,
16     LoginComponent,
17     SignupComponent
18   ],
19   imports: [
20     BrowserModule,
21     AppRoutingModule,
22     ReactiveFormsModule,
23     HttpClientModule
24   ],
25   providers: [],
26   bootstrap: [AppComponent]
27 })
28 export class AppModule { }
29
```

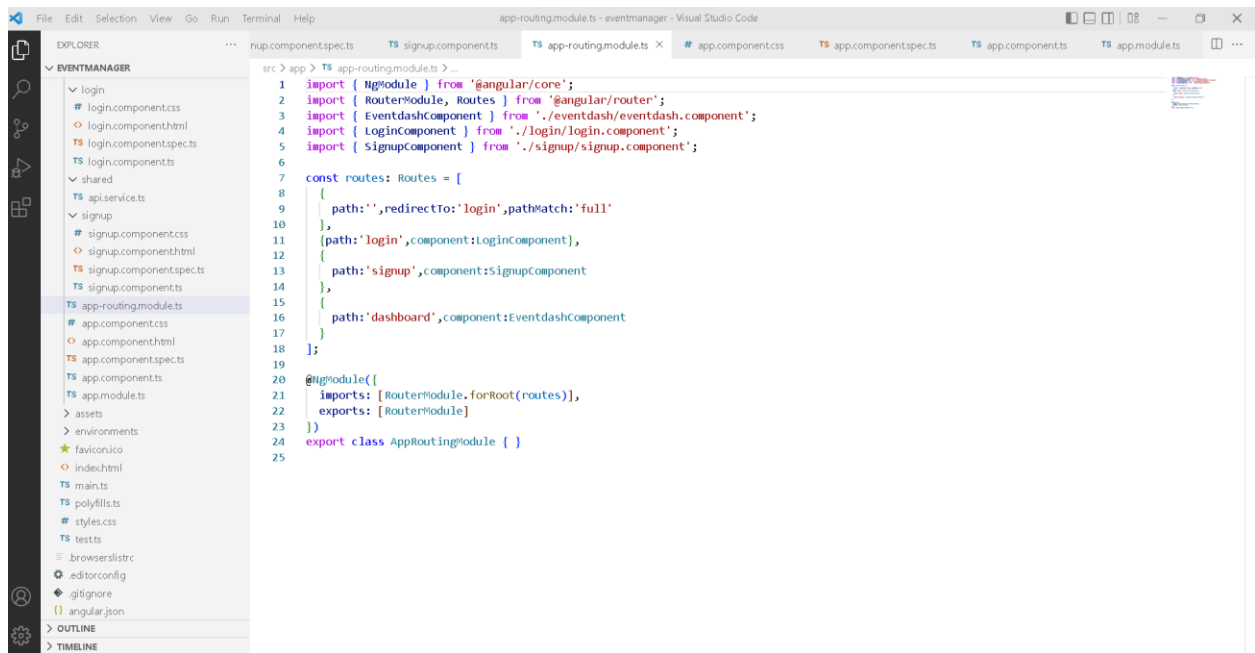


The screenshot shows the Visual Studio Code editor with the 'app.component.html' file open. The Explorer sidebar on the left shows the project structure for 'EVENTMANAGER'. The main editor area displays the following HTML code:

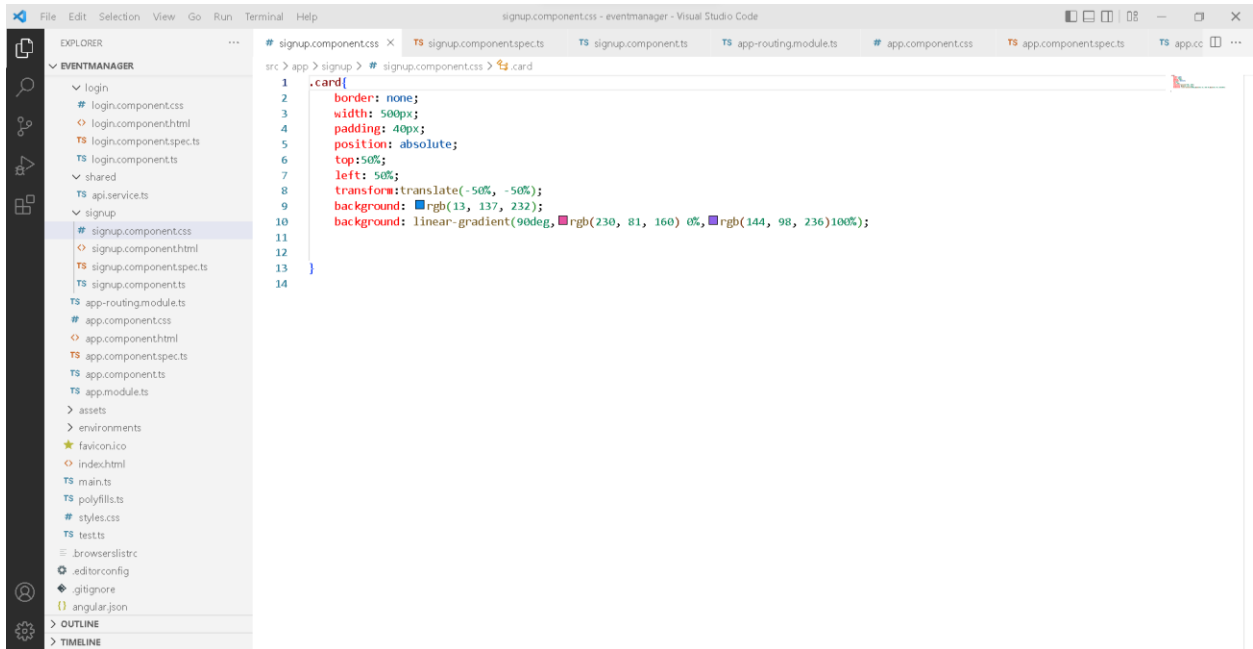
```
src > app > TS app.component.html > router-outlet
Go to component
1 <router-outlet>
2
3 </router-outlet>
```



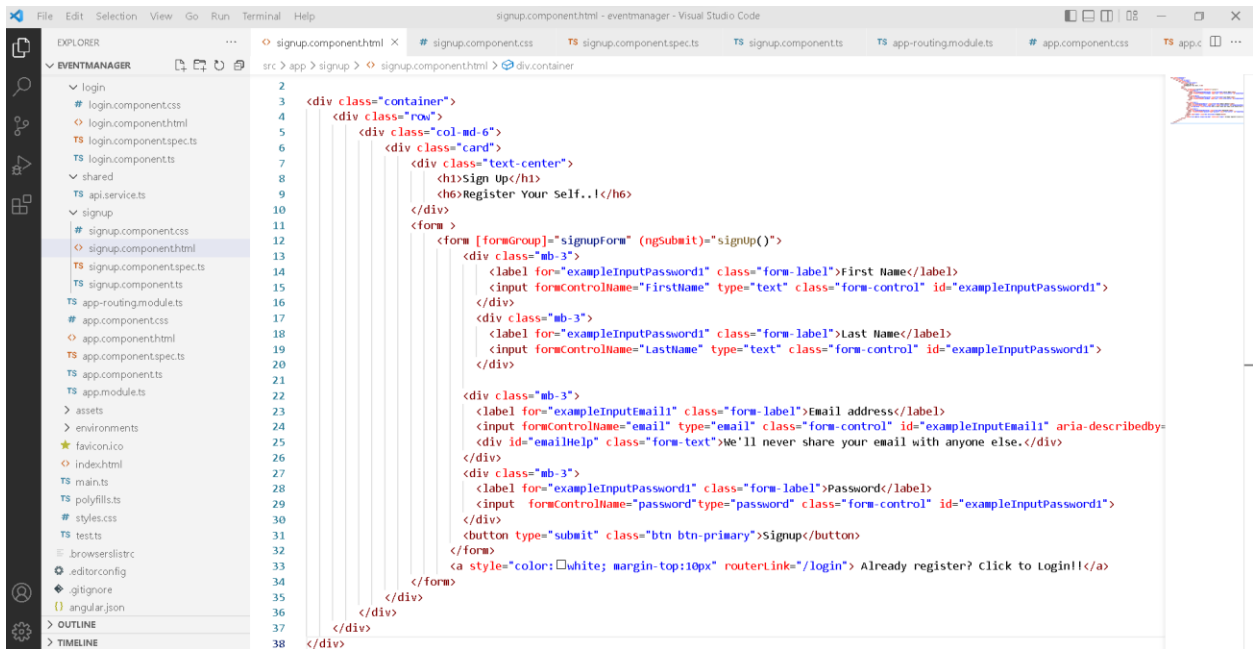
```
1 import { Component } from '@angular/core';
2
3 @component({
4   selector: 'app-root',
5   templateUrl: './app.component.html',
6   styleUrls: ['./app.component.css']
7 })
8 export class AppComponent {
9   title = 'eventmanager';
10 }
11
```



```
1 import { NgModule } from '@angular/core';
2 import { RouterModule, Routes } from '@angular/router';
3 import { EventdashComponent } from './eventdash/eventdash.component';
4 import { LoginComponent } from './login/login.component';
5 import { SignupComponent } from './signup/signup.component';
6
7 const routes: Routes = [
8   {
9     path: '', redirectTo: 'login', pathMatch: 'full'
10   },
11   { path: 'login', component: LoginComponent },
12   { path: 'signup', component: SignupComponent },
13   { path: 'dashboard', component: EventdashComponent },
14 ];
15
16 @NgModule({
17   imports: [RouterModule.forRoot(routes)],
18   exports: [RouterModule]
19 })
20 export class AppRoutingModule { }
21
```



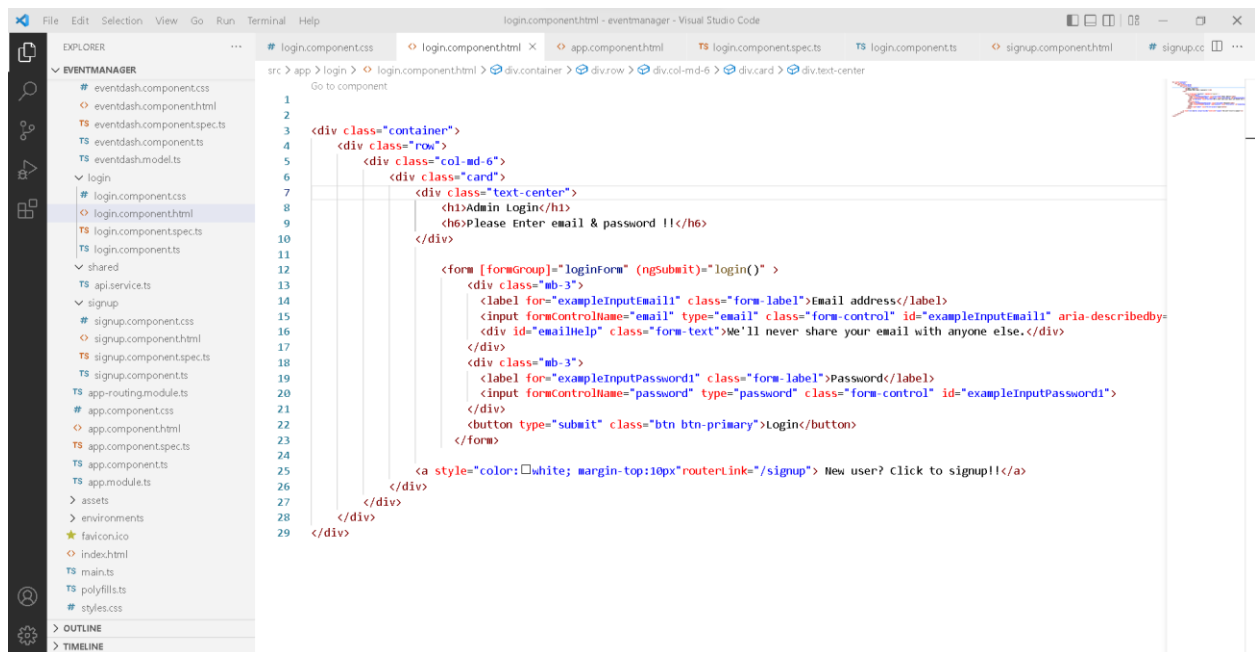
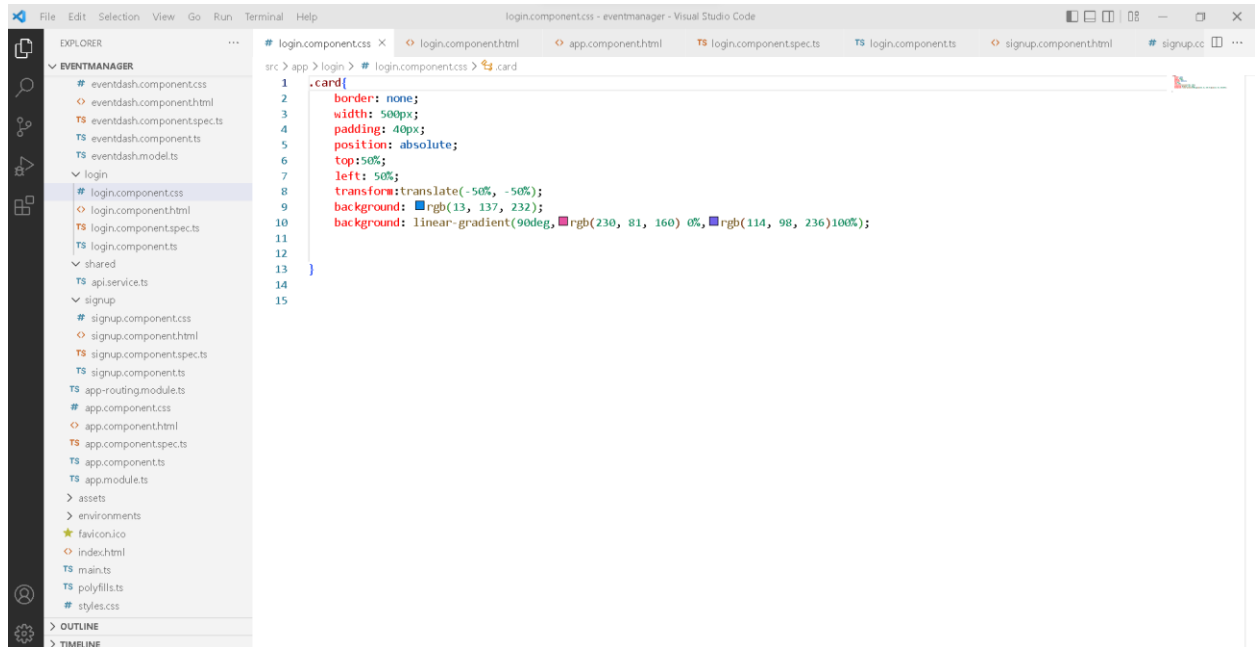
```
1 .card{
2   border: none;
3   width: 500px;
4   padding: 40px;
5   position: absolute;
6   top:50%;
7   left: 50%;
8   transform:translate(-50%, -50%);
9   background: linear-gradient(90deg, #000080 13%, #000080 137%, #000080 232%);
10  background: linear-gradient(90deg, #000080 0%, #000080 230, 81, 160) 0%, #000080 144, 98, 236)100%);
11 }
12
13
14
```



```
1
2
3 <div class="container">
4   <div class="row">
5     <div class="col-md-6">
6       <div class="card">
7         <div class="text-center">
8           <h1>Sign Up</h1>
9           <h2>Register Your Self..!</h2>
10        </div>
11        <form [formGroup]="signupForm" (ngSubmit)="signup()">
12          <div class="mb-3">
13            <label for="exampleInputPassword1" class="form-label">First Name</label>
14            <input formControlName="FirstName" type="text" class="form-control" id="exampleInputPassword1">
15          </div>
16          <div class="mb-3">
17            <label for="exampleInputPassword1" class="form-label">Last Name</label>
18            <input formControlName="LastName" type="text" class="form-control" id="exampleInputPassword1">
19          </div>
20          <div class="mb-3">
21            <label for="exampleInputEmail1" class="form-label">Email address</label>
22            <input formControlName="email" type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp">
23            <div id="emailHelp" class="form-text">We'll never share your email with anyone else.</div>
24          </div>
25          <div class="mb-3">
26            <label for="exampleInputPassword1" class="form-label">Password</label>
27            <input formControlName="password" type="password" class="form-control" id="exampleInputPassword1">
28          </div>
29          <button type="submit" class="btn btn-primary">Signup</button>
30        </form>
31        <a style="color: #000080; margin-top:10px" routerLink="/login"> Already register? click to Login!</a>
32      </div>
33    </div>
34  </div>
35 </div>
36
37 </div>
38 </div>
```

```
1 import { HttpClientModule } from '@angular/common/http';
2 import { Component, OnInit } from '@angular/core';
3 import { FormGroup, FormBuilder } from '@angular/forms';
4 import { Router } from '@angular/router';
5
6 @Component({
7   selector: 'app-signup',
8   templateUrl: './signup.component.html',
9   styleUrls: ['./signup.component.css']
10 })
11 export class SignupComponent implements OnInit {
12
13   public signupForm !: FormGroup;
14
15   constructor(private formBuilder : FormBuilder, private http:HttpClient, private router:Router) { }
16
17   ngOnInit(): void {
18     this.signupForm=this.formBuilder.group({
19       firstName:'',
20       LastName:'',
21       email:'',
22       password:'',
23     })
24   }
25
26   signup(){
27     this.http.post<any>("http://localhost:3000/signupusers/", this.signupForm.value)
28       .subscribe(res=>{
29         alert("Signup Successfull !!");
30         this.signupForm.reset();
31         this.router.navigate(['login']);
32       },err=>{
33         alert("Something went wrong !!")
34       })
35   }
36 }
37
```

```
1 import { Injectable } from '@angular/core';
2 import { HttpClient } from '@angular/common/http';
3 import { map } from 'rxjs/operators';
4
5 @Injectable({
6   providedIn: 'root'
7 })
8 export class ApiService {
9
10   constructor(private http:HttpClient) { }
11
12   postEmployee( data :any){
13     return this.http.post<any>("http://localhost:3000/posts",data)
14       .pipe(map((res:any)=>{
15         return res;
16       })))
17   }
18
19   getEmployee(){
20     return this.http.get<any>("http://localhost:3000/posts")
21       .pipe(map((res:any)=>{
22         return res;
23       })))
24   }
25
26   updateEmployee(data :any,id:number){
27     return this.http.put<any>("http://localhost:3000/posts/"+id,data)
28       .pipe(map((res:any)=>{
29         return res;
30       })))
31   }
32
33   deleteEmployee(id : number){
34     return this.http.delete<any>("http://localhost:3000/posts/"+id)
35       .pipe(map((res:any)=>{
36         return res;
37       })))
38   }
39 }
40
```



eventdash.component.html - eventmanager - Visual Studio Code

```
37 <div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
38   <div class="modal-content">
39     <div class="modal-header">
40       <h5 class="modal-title" id="exampleModalLabel">Employee Details</h5>
41       <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
42     </div>
43     <div class="modal-body">
44       <form [formGroup]="formValue">
45         <div class="mb-3">
46           <label for="exampleInputEmail" class="form-label">First Name</label>
47           <input type="text" formControlName="firstName" class="form-control" id="exampleInputEmail" aria-describedby="em
48         </div>
49         <div class="mb-3">
50           <label for="exampleInputPassword1" class="form-label">Last Name</label>
51           <input type="text" formControlName="lastName" class="form-control" id="exampleInputPassword1">
52         </div>
53         <div class="mb-3">
54           <label for="exampleInputPassword1" class="form-label">Email</label>
55           <input type="text" formControlName="email" class="form-control" id="exampleInputPassword1">
56         </div>
57       </form>
58     </div>
59     <div class="modal-footer">
60       <button type="button" id="cancel" class="btn btn-secondary" data-bs-dismiss="modal">Close</button>
61       <button (click)="postEmployeeDetails()" *ngIf="showAdd" type="button" class="btn btn-primary">Add</button>
62       <button (click)="updateEmployeeDetails()" *ngIf="showUpdate" type="button" class="btn btn-primary">Update</button>
63     </div>
64   </div>
65 </div>
66 </div>
67 </div>
68 </div>
```

eventdash.components.ts - eventmanager - Visual Studio Code

```
1 import { Component, OnInit } from '@angular/core';
2 import { FormBuilder, FormGroup } from '@angular/forms';
3 import { ApiService } from '../shared/api.service';
4 import { EmployeeModel } from './eventdash.model';
5
6
7 @Component({
8   selector: 'app-eventdash',
9   templateUrl: './eventdash.component.html',
10   styleUrls: ['./eventdash.component.css']
11 })
12 export class EventdashComponent implements OnInit {
13   formValue!: FormGroup;
14   employeeModelObj!: EmployeeModel;
15   employeeData!: any;
16   showAdd!: boolean;
17   showUpdate!: boolean;
18
19   constructor( private formBuilder: FormBuilder,
20     private api: ApiService ) { }
21
22   ngOnInit(): void {
23     this.formValue = this.formBuilder.group({
24       firstName: [''],
25       lastName: [''],
26       email: ['']
27     });
28     this.getAllEmployee();
29   }
30   clickAddEmployee(){
31     this.formValue.reset();
32     this.showAdd=true;
33     this.showUpdate=false;
34   }
35   postEmployeeDetails(){
36     this.employeeModelObj.firstName = this.formValue.value.firstName;
37     this.employeeModelObj.lastName = this.formValue.value.lastName;
```

```
37 this.employeeModelObj.lastName = this.formValue.value.lastName;
38 this.employeeModelObj.email = this.formValue.value.email;
39
40 this.api.postEmployee(this.employeeModelObj)
41 .subscribe(res=>{
42   console.log(res);
43   alert("Employee Added Successfully");
44   let ref=document.getElementById('cancel');
45   ref?.click();
46   this.formValue.reset();
47   this.getAllEmployee();
48 },
49 err=>{
50   alert("Something went wrong");
51 })
52
53
54
55 getAllEmployee(){
56   this.api.getEmployee()
57   .subscribe(res=>{
58     this.employeeData=res;
59   })
60 }
61
62
63 deleteEmployee(row : any){
64   this.api.deleteEmployee(row.id)
65   .subscribe(res=>{
66     alert("Employee deleted")
67     this.getAllEmployee();
68   })
69 }
70
71 onEdit(row :any){
72   this.showAdd=false;
73   this.showUpdate=true;
```

```
74 this.employeeModelObj.id=row.id;
75 this.formValue.controls['firstName'].setValue(row.firstName);
76 this.formValue.controls['lastName'].setValue(row.lastName);
77 this.formValue.controls['email'].setValue(row.email);
78
79 }
80
81 updateEmployeeDetails(){
82   this.employeeModelObj.firstName =this.formValue.value.firstName;
83   this.employeeModelObj.lastName =this.formValue.value.lastName;
84   this.employeeModelObj.email =this.formValue.value.email;
85
86   this.api.updateEmployee(this.employeeModelObj,this.employeeModelObj.id)
87   .subscribe(res=>{
88     alert("Updated Successfully");
89     let ref=document.getElementById('cancel');
90     ref?.click();
91     this.formValue.reset();
92     this.getAllEmployee();
93   })
94 }
95
96 }
97
```


eventdash.model.ts - eventmanager - Visual Studio Code

```
1 export class EmployeeModel {
2   id : number = 0;
3   firstName : string = '';
4   lastName : string = '';
5   email : string = '';
6 }
```

Ln 1, Col 1 · Spaces: 4 · UTF-8 · CRLF · {} TypeScript · Go Live

index.html - eventmanager - Visual Studio Code

```
1 <!doctype html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <title>EventManager</title>
6   <base href="/">
7   <meta name="viewport" content="width=device-width, initial-scale=1">
8   <link rel="icon" type="image/x-icon" href="favicon.ico">
9   <!-- CSS only -->
10  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-ivqeczEYfBK"
11  <!-- JavaScript Bundle with Popper -->
12  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js" integrity="sha384-u10knCvxwV5kf"
13
14 </head>
15 <body>
16   <app-root></app-root>
17 </body>
18 </html>
19
```

Ln 1, Col 1 · Spaces: 2 · UTF-8 · LF · HTML · Go Live

```
1 {
2   "posts": [
3     {
4       "id": 1,
5       "firstName": "Sebastian",
6       "lastName": "Eschweiler",
7       "email": "sebastian@codingthesmartway.com"
8     },
9     {
10      "id": 2,
11      "firstName": "Steve",
12      "lastName": "Palmer",
13      "email": "steve@codingthesmartway.com"
14    },
15    {
16      "id": 3,
17      "firstName": "Ann",
18      "lastName": "Smith",
19      "email": "ann@codingthesmartway.com"
20    }
21  ],
22  "signupUsers": [
23    {
24      "firstName": "Uma",
25      "lastName": "Lakshmi",
26      "email": "umalakshwiamanthula@gmail.com",
27      "password": "Uma@1472",
28      "id": 3
29    },
30    {
31      "firstName": "UMA",
32      "lastName": "LAKSHMI",
33      "email": "umalakshwiggmail.com",
34      "password": "Uma@123",
35      "id": 4
36    },
37    {
```

```
37 {
38   "firstName": "ani",
39   "lastName": "",
40   "email": "aniggmail.com",
41   "password": "Ani@1123",
42   "id": 5
43 },
44 {
45   "firstName": "UMA",
46   "lastName": "Lakshmi",
47   "email": "umalakshwiggmail.com",
48   "password": "Uma@123",
49   "id": 6
50 },
51 ],
52 "profile": {
53   "name": "typicode"
54 }
55 }
```

```
1 import { enableProdMode } from '@angular/core';
2 import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';
3
4 import { AppModule } from './app/app.module';
5 import { environment } from './environments/environment';
6
7 if (environment.production) {
8   enableProdMode();
9 }
10
11 platformBrowserDynamic().bootstrapModule(AppModule)
12   .catch(err => console.error(err));
13
```

```
1 import { Injectable } from '@angular/core';
2 import { HttpClient } from '@angular/common/http';
3 import { map } from 'rxjs/operators';
4
5 @Injectable({
6   providedIn: 'root'
7 })
8 export class ApiService {
9
10   constructor(private http: HttpClient) { }
11
12   postEmployee( data :any){
13     return this.http.post<any>("http://localhost:3000/posts",data)
14       .pipe(map((res:any)=>{
15         return res;
16       }));
17   }
18   getEmployee(){
19     return this.http.get<any>("http://localhost:3000/posts")
20       .pipe(map((res:any)=>{
21         return res;
22       }));
23   }
24   updateEmployee(data :any,id:number){
25     return this.http.put<any>("http://localhost:3000/posts/"+id,data)
26       .pipe(map((res:any)=>{
27         return res;
28       }));
29   }
30   deleteEmployee(id : number){
31     return this.http.delete<any>("http://localhost:3000/posts/"+id)
32       .pipe(map((res:any)=>{
33         return res;
34       }));
35   }
36 }
37
```

OUTPUT:



Sign Up

Register Your Self..!

First Name

Last Name

Email address

We'll never share your email with anyone else.

Password

[Signup](#)

[Already register? Click to Login!!](#)



localhost:4200 says
Signup Successfull !!
[OK](#)

Sign Up

Register Your Self..!

First Name

Last Name

Email address

We'll never share your email with anyone else.

Password

[Signup](#)

[Already register? Click to Login!!](#)

Admin Login

Please Enter email & password !!

Email address

We'll never share your email with anyone else.

Password

Login

[New user? Click to signup!!](#)

localhost:4200 says
Login Success!!

OK

Admin Login

Please Enter email & password !!

Email address

We'll never share your email with anyone else.

Password

Login

[New user? Click to signup!!](#)

Event Manager

[Add Employee](#)[Logout](#)

Employee Id	first Name	last Name	Email Id	Action	
1	Sebastian	Eschweiler	sebastian@codingthesmartway.com	Edit	Delete
2	Steve	Palmer	steve@codingthesmartway.com	Edit	Delete
3	Ann	Smith	ann@codingthesmartway.com	Edit	Delete

Event Manager

[Add Employee](#)[Logout](#)

Employee Id	first Name	last Name
1	Sebastian	Eschweiler
2	Steve	Palmer
3	Ann	Smith

Employee Details

First Name

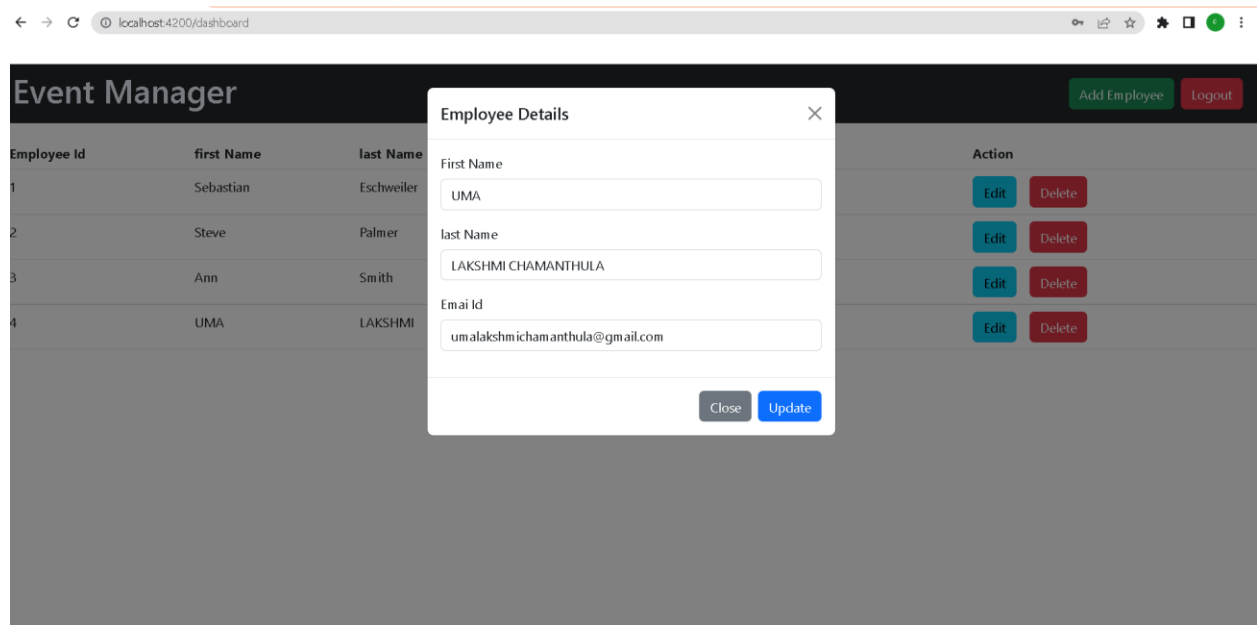
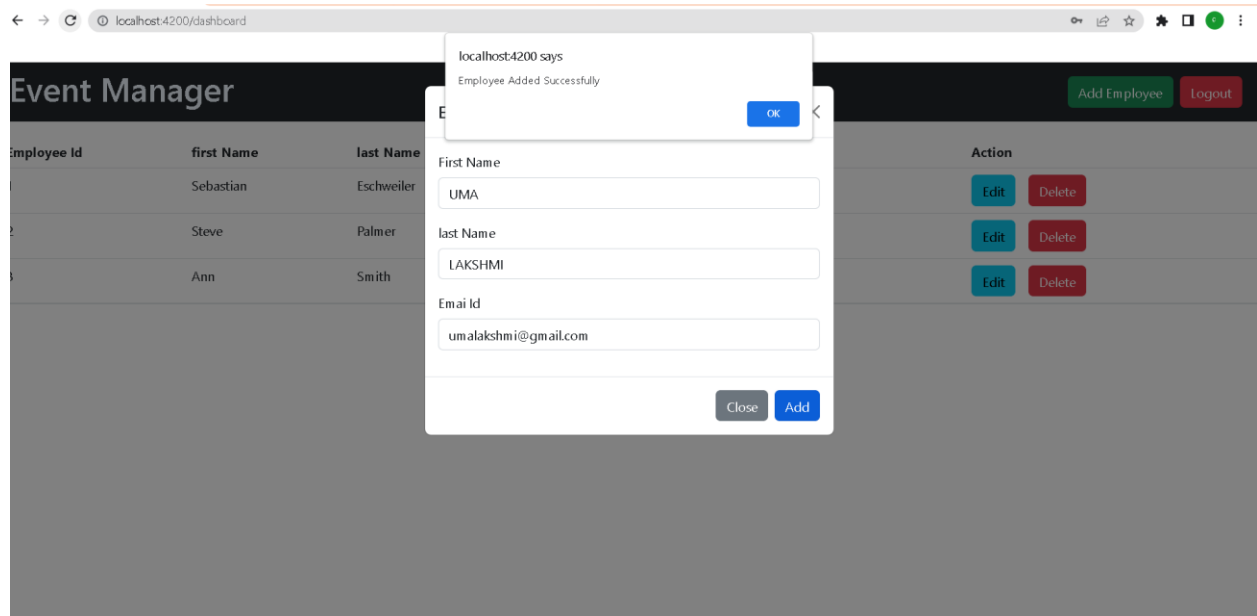
last Name

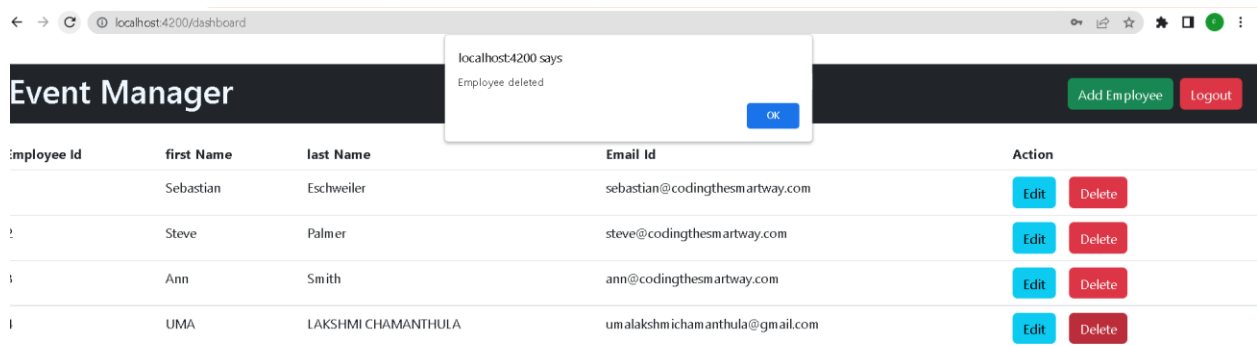
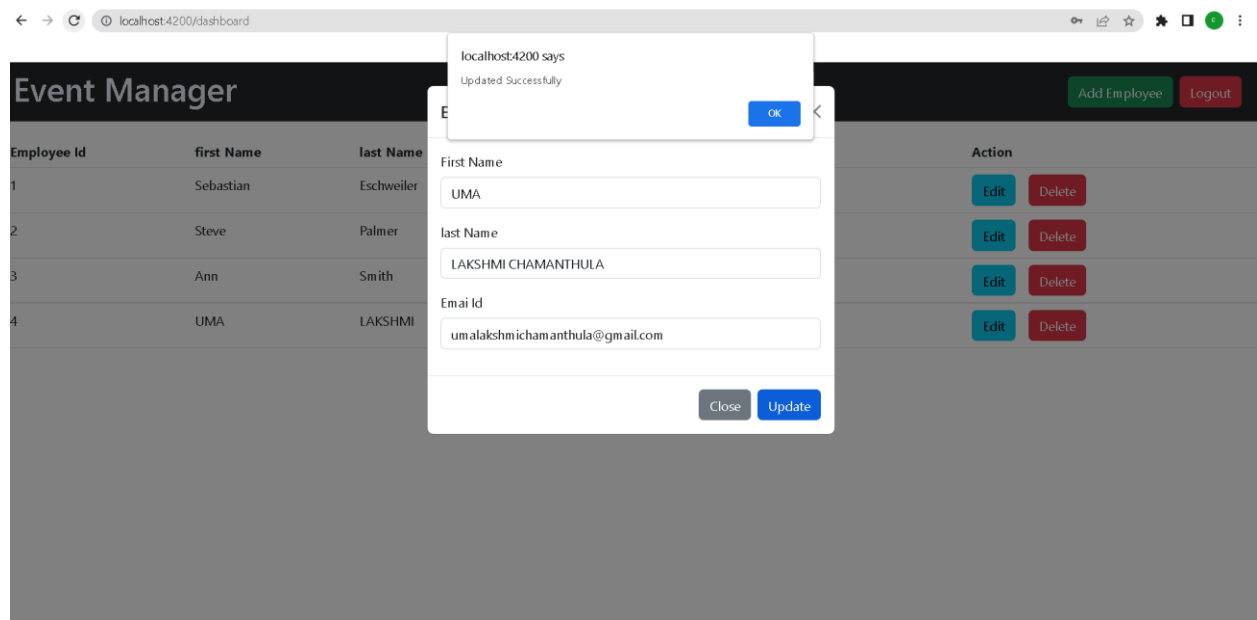
Email Id

[Close](#)[Add](#)

Action

[Edit](#)[Delete](#)[Edit](#)[Delete](#)[Edit](#)[Delete](#)





Event Manager

Add Employee

Logout

Employee Id	first Name	last Name	Email Id	Action	
1	Sebastian	Eschweiler	sebastian@codingthesmartway.com	Edit	Delete
2	Steve	Palmer	steve@codingthesmartway.com	Edit	Delete
3	Ann	Smith	ann@codingthesmartway.com	Edit	Delete