

```
TS app.component.ts M    <> app.component.html M    TS app.config.ts M    TS app.routes.ts X
src > app > TS app.routes.ts > [0] routes
1  import { Routes } from '@angular/router';
2  import { MasterComponent } from '../components/master/master.component';
3
4  export const routes: Routes = [
5    {
6      path: '',
7      redirectTo: 'master',
8      pathMatch: 'full'
9    },
10   {
11     path: 'master',
12     component: MasterComponent
13   }
14 ];
```

```
PS D:\v-18\angular18_tutorial\angular_18_tutorial\src\app\components> ng g c employee
CREATE src/app/components/employee/employee.component.html (24 bytes)
CREATE src/app/components/employee/employee.component.spec.ts (629 bytes)
CREATE src/app/components/employee/employee.component.ts (254 bytes)
CREATE src/app/components/employee/employee.component.css (0 bytes)
PS D:\v-18\angular18_tutorial\angular_18_tutorial\src\app\components> ng g c client
```

```
14   component: MasterComponent
15 },
16 {
17   path: 'employee',
18   component: EmployeeComponent
19 },
20 {
21   path: 'client',
22   component: ClientComponent
23 }
```

BS5 Progress Bars

BS5 Spinners

BS5 Pagination

BS5 List Groups

BS5 Cards

BS5 Dropdowns

BS5 Collapses

A navigation bar is a navigation header that is placed at the top of the page:

Logo Link Link Link Search Search

Elements

Console

Network

Sources

Performance

Memory

Application

Lighthouse

Angular

Styles

Computed

Layout

Filter

:hov

element.style {

}

.bg-dark {

background-color: #212

}

@media (min-width: 576px) {

.navbar-expand-sm {

flex-wrap: nowrap;

}

}

EXPLORER

ANGULAR_18_TUTORIAL

- > .angular
- > .vscode
- > node_modules
- > public
- ✓ src
 - ✓ app
 - > components
 - > model
 - > services
 - # app.component.css
 - <> app.component.html M
 - TS app.component.spec.ts
 - TS app.component.ts M
 - TS app.config.ts M
 - TS app.routes.ts M
 - <> index.html
 - TS main.ts
 - # styles.css
 - editorconfig
 - gitignore

src > app > <> app.component.html > nav.navbar.navbar-expand-sm.navbar-dark.bg-dark > div.container-fluid > div#mynavbar.collapse.nav

Go to component

```
1 <nav class="navbar navbar-expand-sm navbar-dark bg-dark">
2   <div class="container-fluid">
3     <a class="navbar-brand" href="javascript:void(0)">Logo</a>
4     <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#mynavbar">
5       <span class="navbar-toggler-icon"></span>
6     </button>
7     <div class="collapse navbar-collapse" id="mynavbar">
8       <ul class="navbar-nav me-auto">
9         <li class="nav-item" style="margin-top:-12px;">
10          <a class="nav-link active" href="javascript:void(0)">Link</a>
11        </li>
12        <li class="nav-item" style="margin-top:-12px;">
13          <a class="nav-link" href="javascript:void(0)">Link</a>
14        </li>
15        <li class="nav-item" style="margin-top:-12px;">
16          <a class="nav-link" href="javascript:void(0)">Link</a>
17        </li>
18      </ul>
19      <form class="d-flex">
20        <input class="form-control me-2" type="text" placeholder="Search">
21        <button class="btn btn-primary" type="button">Search</button>
22      </form>
23    </div>
24  </div>
25 </nav>
26 <div class="container pt-1">
27   <router-outlet></router-outlet>
28 </div>
29
```

```
23   </div>
24 </div>
25 </nav>
26 <div class="container pt-1">
27   <router-outlet></router-outlet>
28 </div>
29
```

Roles		Designation
Sr No	Role Name	
1	Super Admin	
2	Admin	
3	TL	

Elements Console Network Sources Performance Memory Application Lighthouse Angular

Preserve log Disable cache Slow 4G

Filter

All Fetch/XHR Doc CSS JS Font Img Media Manifest WS Wasm Other

Blocked response cookies Blocked requests 3rd-party requests

- BSS Colors
- BSS Tables
- BSS Images
- BSS Jumbotron
- BSS Alerts
- BSS Buttons
- BSS Button Groups
- BSS Badges
- BSS Progress Bars
- BSS Spinners**
- BSS Pagination
- BSS List Groups
- BSS Cards
- BSS Dropdowns
- BSS Collapse

Example

```
<div class="spinner-border text-muted"></div>
<div class="spinner-border text-primary"></div>
<div class="spinner-border text-success"></div>
<div class="spinner-border text-info"></div>
<div class="spinner-border text-warning"></div>
<div class="spinner-border text-danger"></div>
<div class="spinner-border text-secondary"></div>
<div class="spinner-border text-dark"></div>
<div class="spinner-border text-light"></div>
```

[Try it Yourself »](#)

```
> .angular
> .vscode
> node_modules
> public
> src
  > app
    > components
    > client
    > designation
      # designation.component.css U
      <> designation.component.html U
      TS designation.components... U
      TS designation.components.ts U
    > employee
    > master
    > roles
    > model
    > services
  # app.component.css
  <> app.component.html M
```

```
2
3  @if (isLoading == true) {
4    <div class="col-12 text-center">
5      <div class="spinner-border text-muted"></div>
6      <div class="spinner-border text-primary"></div>
7      <div class="spinner-border text-success"></div>
8      <div class="spinner-border text-info"></div>
9      <div class="spinner-border text-warning"></div>
10     <div class="spinner-border text-danger"></div>
11     <div class="spinner-border text-secondary"></div>
12     <div class="spinner-border text-dark"></div>
13     <div class="spinner-border text-light"></div>
14   </div>
15 } @else {
16   @for (item of designationList; track $index) {
17     <div class="col-3">
18       <div class="card">
19         <div class="card-body">
20           <h4 class="card-title">{{item.designation}}</h4>
21           <a href="#" class="card-link">[dit</a>
22           <a href="#" class="card-link">Uelet</a>
23         </div>
24       </div>
```

style="margin-top:12px;" Aa ab * No results ↑ ↓

Replace AB

```
12 export class DesignationComponent implements OnInit {
13
14   designationList: IDesignation[] = [];
15   isLoading: boolean = true;
16   masterService = inject(MasterService);
17
18   ngOnInit(): void {
19     this.masterService.getDesignations().subscribe((result: APIResponseModel) => {
20       this.designationList = result.data;
21       this.isLoading = false;
22     }, error => {
23       alert("API error/ Network Down")
24       this.isLoading = false;
25     });
26   }
27 }
```

```

8     standalone: true,
9     imports: [RouterOutlet, MasterComponent, RouterLink],
10    templateUrl: './app.component.html',
11    styleUrls: ['./app.component.css']
12  })
13  export class AppComponent {
14    title = 'angular_18_tutorial';
15  }
16

```

```

6      </button>
7      <div class="collapse navbar-collapse" id="myNavbar">
8        <ul class="navbar-nav me-auto">
9          <li class="nav-item">
10             <a class="nav-link active" routerLink="/master">Master</a>
11          </li>
12          <li class="nav-item">
13             <a class="nav-link" [routerLink]="'/employee'">Employee</a>
14          </li>
15          <li class="nav-item">
16             <a class="nav-link" routerLink="/client">Client</a>
17          </li>
18        </ul>
19        <form class="d-flex">
20          <input class="form-control me-2" type="text" placeholder="Search">
21          <button class="btn btn-primary" type="button">Search</button>
22        </form>
23      </div>
24    </div>
25  </div>
26

```

```

6  @Component({
7    selector: 'app-root',
8    standalone: true,
9    imports: [RouterOutlet, MasterComponent, RouterLink, RouterLinkActive],
10   templateUrl: './app.component.html',
11   styleUrls: ['./app.component.css']
12 })
13 export class AppComponent {
14   title = 'angular_18_tutorial';
15 }
16

```

```

<li class="nav-item">
  <a class="nav-link" routerLink="/master" routerLinkActive="active">Master</a>
</li>
<li class="nav-item">
  <a class="nav-link" [routerLink]="'/employee'" routerLinkActive="active">Employee</a>
</li>
<li class="nav-item">
  <a class="nav-link" routerLink="/client" routerLinkActive="active">Client</a>
</li>
</ul>

```

POST /api/ClientStrive/AddUpdateClient

Try it out

No parameters

Request body application/json-patch+json

Example Value | Schema

```

{
  "clientId": 0,
  "contactPersonName": "string",
  "companyName": "string",
  "address": "string",
  "city": "string",
  "pincode": "string",
  "state": "string",
  "EmployeeStrength": 0,
  "gstNo": "string",
  "contactNo": "string",
  "regNo": "string"
}

```


The screenshot shows the VS Code interface with the Explorer sidebar on the left displaying the project structure for 'ANGULAR_18_TUTORIAL'. The main editor shows the file 'client.component.html' with the following code:

```
1 <div class="row">
2   <div class="col-8">
3     <div class="card">
4       <div class="card-header bg-success">
5         Client List
6       </div>
7       <div class="card-body">
8         <div class="row">
9           <div class="col-12">
10            <table class="table table-bordered">
11              <thead>
12                <tr>
13                  <th>Sr No</th>
14                  <th>Role Name</th>
15                </tr>
16              </thead>
17              <tbody>
18                <tr *ngFor="let role of roleList; let srno = index">
19                  <td>{{srno+1}}</td>
20                  <td>{{role.role}}</td>
21                </tr>
22              </tbody>
23            </table>
24          </div>
25        </div>
26      </div>
27    </div>
28  </div>
```

This close-up view of the code editor shows the 'New Client' form structure within the 'card-body' of the 'col-4' container:

```
26 <div class="col-4">
27   <div class="card">
28     <div class="card-header bg-success">
29       New Client
30     </div>
31     <div class="card-body">
32       <div class="row">
33         <div class="col-6">
34           <label for="">Name</label>
35           <input type="text" class="form-control">
36         </div>
37         <div class="col-6">
38           <label for="">Name</label>
39           <input type="text" class="form-control">
40         </div>
41         <div class="col-6">
42           <label for="">Name</label>
43           <input type="text" class="form-control">
44         </div>
45       </div>
```

This screenshot shows the continuation of the form code in the editor and a JSON schema in the right sidebar. The code in the editor is:

```
30 </div>
31 <div class="card-body">
32   <div class="row">
33     <div class="col-6">
34       <label for="">Person Name</label>
35       <input type="text" class="form-control">
36     </div>
37     <div class="col-6">
38       <label for="">Company Name</label>
39       <input type="text" class="form-control">
40     </div>
41     <div class="col-6">
42       <label for="">City</label>
43       <input type="text" class="form-control">
44     </div>
45     <div class="col-6">
46       <label for="">Name</label>
47       <input type="text" class="form-control">
48     </div>
49     <div class="col-6">
50       <label for="">Name</label>
51       <input type="text" class="form-control">
```

The right sidebar displays a JSON schema under the heading 'Example Value : Schema':

```
{
  "clientId": 0,
  "contactPersonName": "string",
  "companyName": "string",
  "address": "string",
  "city": "string",
  "pincode": "string",
  "state": "string",
  "EmployeeStrength": 0,
  "gstNo": "string",
  "contactNo": "string",
  "regNo": "string"
}
```

37

<div class="col-6">

<label for="">Company Name</label>

<input type="text" class="form-control">

</div>

<div class="col-6">

<label for="">City</label>

<input type="text" class="form-control">

</div>

<div class="col-6">

<label for="">State</label>

<input type="text" class="form-control">

</div>

<div class="col-6">

<label for="">Pincode</label>

<input type="text" class="form-control">

</div>

<div class="col-6">

<label for="">Address</label>

<textarea class="form-control"></textarea>

</div>

<div class="col-6">

<label for="">Employee Strength</label>

<input type="text" class="form-control">

</div>

</div>

</div>

Parameters

Try it out

No parameters

Request body

application/json-patch+json

Example Value | Schema

```
{
  "clientId": 0,
  "contactPersonName": "string",
  "companyName": "string",
  "address": "string",
  "city": "string",
  "pincode": "string",
  "state": "string",
  "EmployeeStrength": 0,
  "gstNo": "string",
  "contactNo": "string",
  "regNo": "string"
}
```

Responses

64

<input type="text" class="form-control">

</div>

<div class="col-6">

<label for="">gstNo</label>

<input type="text" class="form-control">

</div>

<div class="col-6">

<label for="">regNo</label>

<input type="text" class="form-control">

</div>

</div>

</div>

Example Value | Schema

```
{
  "clientId": 0,
  "contactPersonName": "st
  "companyName": "string",
  "address": "string",
  "city": "string",
  "pincode": "string",
  "state": "string",
  "EmployeeStrength": 0,
  "gstNo": "string",
  "contactNo": "string",
  "regNo": "string"
}
```

Logo

Master

Employee

Client

Search

Search

Client List

Sr No	Role Name
-------	-----------

New Client

Person Name

Company Name

Contact No

City

State

Pincode

Address

Employee Strength

GstNo

RegNo

```

74     </div>
75     <div class="row pt-2">
76       <div class="col-6 text-center">
77         <button class="btn btn-secondary">Reset</button>
78       </div>
79       <div class="col-6 text-center">
80         <button class="btn btn-success">Save Client</button>
81       </div>
82     </div>
83   </div>
84 </div>
85 </div>
86 </div>

```

State	Pincode
Address	Employee Strength
GstNo	RegNo
<button>Reset</button>	<button>Save Client</button>

GET /api/ClientStrive/GetAllClients		
Parameters		
No parameters		
Responses		
Code	Description	Links
200	Success	No links

Client List				
Sr No	Company Name	Person Name	Contact No	City

Person Name	Company Name
Contact No	City
State	Pincode

<https://jsonformatter.org/json-to-typescript>

JSON BEAUTIFIER JSON PARSER XML FORMATTER JSBEAUTIFIER SAVE RECENT LINKS LOGIN

JSON to Typescript

Sample

1 {
2 "clientId": 0,
3 "contactPersonName": "string",
4 "companyName": "string",
5 "address": "string",
6 "city": "string",
7 "pincode": "string",
8 "state": "string",
9 "EmployeeStrength": 0,
10 "gstNo": "string",
11 "contactNo": "string",
12 "regNo": "string"
13 }

Load Data

Build with Gemini API

JSON to Typescript

1 export interface Welcome8 {
2 clientId: number;
3 contactPersonName: string;
4 companyName: string;
5 address: string;
6 city: string;
7 pincode: string;
8 state: string;
9 EmployeeStrength: number;
10 gstNo: string;
11 contactNo: string;
12 regNo: string;
13 }
14

Angular 18 Full Course | Angular 18 Tutorial With Project

EXPLORER

ANGULAR_18_TUTORIAL

src

app

components

client

client.component.css

client.component.html

TS client.component.spec.ts

TS client.component.ts

designation

employee

master

roles

model

class

TS Client.ts

interface

services

app.component.css

app.component.html

TS app.component.spec.ts

TS app.component.ts

TS app.config.ts

TS app.routes.ts

index.html

OUTLINE

TIMELINE

TS client.component.ts U

client.component.css U

client.component.html U

TS Client.ts U X

roles.component.html U

```
src > app > model > class > TS Client.ts > Client > regNo
1  export class Client {
2      clientId: number;
3      contactPersonName: string;
4      companyName: string;
5      address: string;
6      city: string;
7      pincode: string;
8      state: string;
9      employeeStrength: number;
10     gstNo: string;
11     contactNo: string;
12     regNo: string;
13
14     constructor() {
15         this.clientId = 0;
16         this.employeeStrength = 0;
17         this.address = '';
18         this.city = '';
19         this.companyName = '';
20         this.contactNo = '';
21         this.state = '';
22         this.regNo = '';
23         this.gstNo = '';
24         this.pincode = '';
25         this.contactPersonName = '';
26     }
27
28 }
```

Ln 12, Col 19 Spaces: 4 UTF-8 CRLF TypeScript Go Live Prettier

1:48:02 / 3:43:27 Create a Class >

TS client.component.ts U X

client.component.html 6, U

```
src > app > components > client > TS client.component.ts > ClientComponent.component.html > div.row > div.col-4 > div.card > div.card-body > div.row > div.col-6 > input.form-control
1  import { Component } from '@angular/core';
2  import { Client } from '../model/class/Client';
3  import { FormsModule } from '@angular/forms';
4
5  @Component({
6      selector: 'app-client',
7      standalone: true,
8      imports: [FormsModule],
9      templateUrl: './client.component.html',
10     styleUrls: ['./client.component.css']
11 })
12 export class ClientComponent {
13
14     clientObj: Client = new Client();
15     clientList: Client[] = [];
16
17 }
18
19
20
21
```

```
34 </div>
35 <div class="col-4">
36 <div class="card">
37 <div class="card-header">
38     New Client
39 </div>
40 <div class="card-body">
41 <div class="row">
42 <div class="col-6">
43 <label for=""> Person Name</label>
44 <input type="text" [(ngModel)]="clientObj." [(ngModel)]="clientObj."
45 </div>
46 <div class="col-6">
47 <label for="">Company Name</label>
48 <input type="text" [(ngModel)]="clientObj." class="form-control">
49 </div>
50 <div class="col-6">
51 <label for="">Contact No</label>
52 <input type="text" [(ngModel)]="clientObj." class="form-control">
53 </div>
54 <div class="col-6">
55 <label for="">City</label>
56 <input type="text" [(ngModel)]="clientObj." class="form-control">
57 </div>
58
59 <div class="col-6">
60 <label for="">State</label>
61 <input type="text" [(ngModel)]="clientObj." class="form-control">
62 </div>
63 <div class="col-6">
```

Ln 44, Col 89 (45 selected) Spaces: 4 UTF-8 CRLF HTML Go Live Prettier


```
<div class="card-body">
  <div class="row">
    <div class="col">
      <label for=""> Person Name</label>
      <input type="text" [(ngModel)]="clientObj.contactPersonName" class="form-control">
    </div>
    <div class="col-6">
      <label for="">Company Name</label>
      <input type="text" [(ngModel)]="clientObj.companyName" class="form-control">
    </div>
    <div class="col-6">
      <label for="">Contact No</label>
      <input type="text" [(ngModel)]="clientObj.contactNo" class="form-control">
    </div>
    <div class="col-6">
      <label for="">City</label>
      <input type="text" [(ngModel)]="clientObj.city" class="form-control">
    </div>
    <div class="col-6">
      <label for="">State</label>
      <input type="text" [(ngModel)]="clientObj.state" class="form-control">
    </div>
  </div>
</div>
```

```
11 })
12 export class ClientComponent {
13
14   clientObj: Client = new Client();
15   clientlist: Client[] = [];
16
17   onSaveClient() {
18     |
19   }
20
21 }
22
23
81 <input type="text" [(ngModel)]="clientObj.regNo" class="form-control">
82 </div>
83 </div>
84 <div class="row pt-2">
85   <div class="col-6 text-center">
86     <button class="btn btn-secondary">Reset</button>
87   </div>
88   <div class="col-6 text-center">
89     <button class="btn btn-success" (click)="onSaveClient()">Save Client
90   </div>
91 </div>
92 </div>
93 </div>
94 </div>
95 </div>
```

```
> model
  services
    TS clients.service.spec.ts U
    TS clients.service.ts U
    TS master.service.spec.ts U
    TS master.service.ts U
  # app.component.css
  < app.component.html M
  TS app.component.spec.ts
  TS app.component.ts M
  TS app.config.ts M
  TS app.routes.ts M
  < index.html
  TS main.ts
  # styles.css

export class MasterService {
  constructor(private http: HttpClient) { }

  getDesignations():Observable<APIResponseModel> {
    return this.http.get<APIResponseModel>("https://freeapi.miniprojectideas.com/api/ClientStrive/GetAllDesignation")
  }
}
```

```
> app
  > environments
    < index.html
    TS main.ts
    # styles.css
  .editorconfig
  .gitignore
  {} angular.json M
  {} package-lock.json M
  {} package.json M
  README.md
  {} tsconfig.app.json
  {} tsconfig.json
  {} tsconfig.spec.json

providedIn: 'root'
})
export class ClientService {
  constructor(private http: HttpClient) { }

  getAllClients():Observable<Client> {
    return this.http.get("https://freeapi.miniprojectideas.com/api/ClientStrive/GetAllDesignation")
  }
}
```

```
TS clientcomponents.ts U • TS master.service.ts U • TS client.service.ts 1, U • {} angular.json M • TS environments.ts U •
src > environments > TS environments.ts > [environment] > API_URL
1 export const environment = {
2   API_URL: 'https://freeapi.miniprojectideas.com/api/ClientStrive/'
3 };
4
```

```
TS master.service.ts U • TS client.service.ts 1, U • {} angular.json M • TS environments.ts U • TS environment.development.ts U •
src > environments > TS environment.development.ts > [environment] > API_URL
1 export const environment = {}
2   API_URL: 'https://freeapi.miniprojectideas.com/api/ClientStrive/'
3 };
4
```

```
TS client.service.spec.ts U • TS client.service.ts U • TS master.service.spec.ts U • TS master.service.ts U •
# app.component.css M
< app.component.html M
TS app.component.spec.ts M
TS app.component.ts M
TS app.config.ts M
TS app.routes.ts M
environments
  client.service.ts 7 @Injectable({
  8   providedIn: 'root'
  9 })
  10 export class ClientService {
  11
  12   constructor(private http: HttpClient) { }
  13
  14   getAllClients(): Observable<Client> {
  15     return this.http.get<Client>(environment.API_URL + "GetAllDesignation")
  16   }
  17 }
```

(query)

Execute

Clear

Responses

Curl

```
curl -X 'DELETE' \
  'https://freeapi.miniprojectideas.com/api/ClientStrive/DeleteClientById?clientId=1' \
  -H 'accept: text/plain'
```

Request URL

```
https://freeapi.miniprojectideas.com/api/ClientStrive/DeleteClientById?clientId=1
```

Server response

Code	Details
200	

```
TS client.service.spec.ts U • TS client.service.ts U • TS master.service.spec.ts U • TS master.service.ts U •
# app.component.css M
< app.component.html M
TS app.component.spec.ts M
TS app.component.ts M
TS app.config.ts M
TS app.routes.ts M
environments
  client.service.ts 12 constructor(private http: HttpClient) {}
  13
  14   getAllClients(): Observable<APIResponseModel> {
  15     return this.http.get<APIResponseModel>(environment.API_URL + "GetAllDesignation")
  16   }
  17   addUpdate (obj:Client):Observable<APIResponseModel> {
  18     return this.http.post<APIResponseModel>(environment.API_URL + "GetAllDesignation",obj)
  19   }
  20   deleteClientById (id:number):Observable<APIResponseModel> {
  21     return this.http.get<APIResponseModel>(environment.API_URL + "DeleteClientById?clientId="+id)
  22   }
  23 }
  24
```

```
14 export class ClientComponent implements OnInit {
15
16   clientObj: Client = new Client();
17   clientList: Client[] = [];
18
19   clientService = inject(ClientService);
20
21   ngOnInit(): void {
22   }
23
24   loadClient() {
25     this.clientService.getAllClients().subscribe((res: APIResponseModel) => {
26       this.clientList = res.data;
27     });
28   }
29
30   onSaveClient() {
```

```
17 clientList: Client[] = [];
18
19 clientService = inject(ClientService);
20
21 ngOnInit(): void {
22   this.loadClient();
23 }
24
25 loadClient() {
26   this.clientService.getAllClients().subscribe((res: APIResponseModel) => {
27     this.clientList = res.data;
28   });
29 }
30
31 onSaveClient() {
32   this.clientService.addUpdate(this.clientObj).subscribe((res: APIResponseModel) => {
33     if (res.result) {
34       alert("Client Created Successs");
35     } else {
36       alert(res.message);
37     }
38   });
39 }
```

```
21 ngOnInit(): void {
22   this.loadClient();
23 }
24
25 loadClient() {
26   this.clientService.getAllClients().subscribe((res: APIResponseModel) => {
27     this.clientList = res.data;
28   });
29 }
30
31 onSaveClient() {
32   debugger;
33   this.clientService.addUpdate(this.clientObj).subscribe((res: APIResponseModel) => {
34     if (res.result) {
35       alert("Client Created Successs");
36       this.loadClient();
37     } else {
38       alert(res.message);
39     }
40   });
41 }
```

```
14 <th>Company Name</th>
15 <th>Person Name</th>
16 <th>Contact No</th>
17 <th>City</th>
18 </tr>
19 </thead>
20 <tbody>
21   @For (item of clientList; track $index) {
22     <tr>
23       <td>{{ $index + 1 }}</td>
24       <td>{{ item.companyName }}</td>
25       <td>{{ item.contactPersonName }}</td>
26       <td>{{ item.contactNo }}</td>
27       <td>{{ item.city }}</td>
28     </tr>
29   }
30 </tbody>
31 </table>
```

```

13 constructor(private http: HttpClient) { }
14
15 getAllClients():Observable<APIResponseModel> {
16     return this.http.get<APIResponseModel>(environment.API_URL + "GetAllClients")
17 }
18
19 addUpdate (obj:Client):Observable<APIResponseModel> {
20     return this.http.post<APIResponseModel>(environment.API_URL + "AddUpdateClient",obj)
21 }
22
23 deleteClientById (id:number):Observable<APIResponseModel> {
24     return this.http.get<APIResponseModel>(environment.API_URL + "DeleteClientById?clientId="+id)
25 }
26 }
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43

```

```

30
31 onSaveClient() {
32     debugger;
33     this.clientService.addUpdate(this.clientObj).subscribe((res:APIResponseModel)=>{
34         if(res.result) {
35             alert("Cleint Created Success");
36             this.loadClient();
37             this.clientObj = new Client();
38         } else {
39             alert(res.message)
40         }
41     });
42 }
43

```

```

16 <th>Contact No</th>
17 <th>City</th>
18 <th>Action</th>
19 </tr>
20 </thead>
21 <tbody>
22     @for (item of clientList; track $index) {
23         <tr>
24             <td>{{index+1}}</td>
25             <td>{{item.companyName}}</td>
26             <td>{{item.contactPersonName}}</td>
27             <td>{{item.contactNo}}</td>
28             <td>{{item.city}}</td>
29             <td>
30                 <button class="btn btn-primary">Edit</button>
31                 <button class="btn btn-danger">Delete</button>
32             </td>
33         </tr>

```

```

15 <th>Person Name</th>
16 <th>Contact No</th>
17 <th>City</th>
18 <th>Action</th>
19 </tr>
20 </thead>
21 <tbody>
22     @for (item of clientList; track $index) {
23         <tr>
24             <td>{{index+1}}</td>
25             <td>{{item.companyName}}</td>
26             <td>{{item.contactPersonName}}</td>
27             <td>{{item.contactNo}}</td>
28             <td>{{item.city}}</td>
29             <td>
30                 <button class="btn btn-primary">Edit</button>
31                 <button class="btn btn-danger" (click)="onDelete(item.clientId)">Delete</button>
32             </td>
33         </tr>

```



```
> .vscode
> node_modules
> public
  src
    app
      components
        client
          # client.component.css U
          < client.component.html U
          TS client.component.spec.ts U
          TS client.components.ts U
        designation
        employee
        master
        roles
        model
```

```
43
44
45   onDelete(id: number) {
46     const isDelete = confirm("Are you sure want to delete");
47     if(isDelete) {
48       this.clientService.deleteClientById(id).subscribe((res:APIResponseModel)=>{
49         if(res.result) {
50           alert("Client Delete Success");
51           this.loadClient();
52         } else {
53           alert(res.message)
54         }
55       })
56     }
57   }
58 }
59
60
61 }
```

```
TS client.component.ts U 22
> designation           23
> employee              24
> master               25
> roles                26
> model                27
  services
    TS client.service.spec.ts U
    TS client.service.ts 1 U
```

```
23
24
25   deleteClientById (id:number):Observable<APIResponseModel> {
26     return this.http.delete<APIResponseModel>(environment.API_URL + "DeleteClientById?clientId="+id)
27   }
28 }
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
> client
  # client.component.css U
  < client.component.html 1 U
  TS client.component.spec.ts U
  TS client.component.ts U
  > designation
  > employee
  > master
  > roles
  > model
  > services
```

```
21
22
23   @for (item of clientList; track $index) {
24     <tr>
25       <td>{{$index+1}}</td>
26       <td>{{item.companyName}}</td>
27       <td>{{item.contactPersonName}}</td>
28       <td>{{item.contactNo}}</td>
29       <td>{{item.city}}</td>
30       <td>
31         <button class="btn btn-primary" (click)="onEdit(item)">Edit</button>
32         <button class="btn btn-danger" (click)="onDelete(item.clientId)">Delete</button>
33       </td>
34     </tr>
35   }
```

```
> app
  components
    client
      # client.component.css U
      < client.component.html U
      TS client.component.spec.ts U
      TS client.component.ts U
    > designation
```

```
43
44
45   onEdit(data: Client) {
46     this.clientObj = data;
47   }
48
49   onDelete(id: number) {
50     const isDelete = confirm("Are you sure want to delete");
51     if(isDelete) {
52       this.clientService.deleteClientById(id).subscribe((res:APIResponseModel)=>{
```

```
> components
  client
    # client.component.css U
    < client.component.html U
    TS client.component.spec.ts U
    TS client.component.ts U
  > designation
  > employee
```

```
94
95
96   <button class="btn btn-secondary" >Reset</button>
97 </div>
98   <div class="col-6 text-center">
99     <button class="btn btn-success" (click)="onSaveClient()">
100       {{clientObj.clientId != 0 ? 'Update Client': ' Save Client'}}
101     </button>
102   </div>
103 </div>
104 </div>
```

```
49
50   <div class="row">
51     <div class="col-6">
52       <label for=""> Person Name</label>
53       <input type="text" #personName="ngModel" required [(ngModel)]="clientObj.contactPersonName" class="form-control">
54       <div class="text-danger">
55         @if(personName.errors?.['required']) {
56           <span>This Is Required</span>
57         }
58     </div>
```

```

48     <div class="card-body">
49         <div class="row">
50             <div class="col-6">
51                 <label for=""> Person Name</label>
52                 <input type="text" #personName="ngModel" required [(ngModel)]="clientObj.contactPersonName" class="form-control">
53                 @if (personName.touched) {
54                     <div class="text-danger">
55                         @if(personName?.errors?.['required']) {
56                             <span>This Is Required</span>
57                         }
58                     </div>
59                 }
60             </div>
61         </div>

```

```

48     <div class="card-body">
49         <div class="row">
50             <div class="col-6">
51                 <label for=""> Person Name</label>
52                 <input type="text" #personName="ngModel" minlength="5" required [(ngModel)]="clientObj.contactPersonName" class="form-control">
53                 @if (personName.touched) {
54                     <div class="text-danger">
55                         @if(personName?.errors?.['required']) {
56                             <span>This Is Required</span>
57                         }
58                         @if(personName?.errors?.['minlength']) {
59                             <span>Min % Chars Required</span>
60                         }
61                     </div>
62                 }
63             </div>
64         </div>

```

```

<div class="card-body">
  <form #form="ngForm">
    <div class="row">
      <div class="col-6">
        <label for=""> Person Name</label>
        <input type="text" #personName="ngModel" minlength="5" required name="contactPersonName" [(ngModel)]="clientObj.contactPersonName" class="form-control">
        @if (personName.touched) {
          <div class="text-danger">
            @if(personName?.errors?.['required']) {
              <span>This Is Required</span>
            }
            @if(personName?.errors?.['minlength']) {
              <span>Min % Chars Required</span>
            }
          </div>
        }
      </div>
    </div>

```

```

src / app / components / client / client.component.html
69     <input type="text" [(ngModel)]="clientObj.companyName" class="form-control" name="companyName">
70   </div>
71   <div class="col-6">
72     <label for="">Contact No</label>
73     <input type="text" [(ngModel)]="clientObj.contactNo" class="form-control" name="contactNo">
74   </div>
75   <div class="col-6">
76     <label for="">City</label>
77     <input type="text" [(ngModel)]="clientObj.city" class="form-control" name="city">
78   </div>
79 </div>

```

```

107     <button class="btn btn-secondary">Reset</button>
108   </div>
109   <div class="col-6 text-center">
110     <button class="btn btn-success" [disabled]="!form.valid" (click)="onSaveClient()">
111       {{clientObj.clientId != 0 ? 'Update Client' : 'Save Client'}}
112     </button>
113   </div>
114 </div>
115 </form>

```

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

● PS D:\v-18\angular18_tutorial\angular_18_tutorial\src\app\components> ng g c clientProject
CREATE src/app/components/client-project/client-project.component.html (30 bytes)
CREATE src/app/components/client-project/client-project.component.spec.ts (665 bytes)
CREATE src/app/components/client-project/client-project.component.ts (277 bytes)
CREATE src/app/components/client-project/client-project.component.css (0 bytes)
○ PS D:\v-18\angular18_tutorial\angular_18_tutorial\src\app\components>

```

```

16 |         <a class="nav-link footer-link" routerLink="/client" routerLinkActive="active">Client</a>
17 |     </li>
18 |     <li class="nav-item">
19 |         <a class="nav-link" routerLink="/" routerLinkActive="active">Client Project</a>
20 |     </li>
21 | </ul>
22 | <form class="d-flex">
23 |     <input class="form-control me-2" type="text" placeholder="Search">
24 |     <button class="btn btn-primary" type="button">Search</button>
25 | </form>

```

```

src > app > components > client-project > client-project.component.html > div.row > div.col-4 > div.card > div.card-body > form > div.row > div.col-12
47     </div>
48     <div class="card-body">
49       <form >
50         <div class="row">
51           <div class="col-6">
52             <label for=""> Project Name</label>
53             <input type="text" class="form-control" >
54           </div>
55           <div class="col-6">
56             <label for=""> Start Date</label>
57             <input type="date" class="form-control" >
58           </div>
59           <div class="col-6">
60             <label for=""> Expected End Date </label>
61             <input type="date" class="form-control" >
62           </div>
63           <div class="col-6">
64             <label for=""> Contact Name</label>
65             <input type="text" class="form-control" >
66           </div>
67           <div class="col-6">
68             <label for=""> Contact Mobile</label>
69             <input type="text" class="form-control" >

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- CREATE src/app/components/client-project/client-project.component.html (30 bytes)
- CREATE src/app/components/client-project/client-project.component.spec.ts (665 bytes)
- CREATE src/app/components/client-project/client-project.component.ts (277 bytes)
- CREATE src/app/components/client-project/client-project.component.css (0 bytes)

```

7     </div>
8     <div class="card-body">
9         <form >
10             <div class="row">
11                 <div class="col-6">
12                     <label for=""> Project Name</label>
13                     <input type="text" class="form-control" >
14                 </div>
15                 <div class="col-6">
16                     <label for=""> Start Date</label>
17                     <input type="date" class="form-control" >
18                 </div>
19                 <div class="col-6">
20                     <label for=""> Expected End Date </label>
21                     <input type="date" class="form-control" >
22                 </div>
23                 <div class="col-6">
24                     <label for=""> Contact Name</label>
25                     <input type="text" class="form-control" >
26                 </div>
27                 <div class="col-6">
28                     <label for=""> Contact Mobile</label>
29                     <input type="text" class="form-control" >

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

TS client-project.component.ts U
src > app > components > client-project > TS client-project.component.ts > ClientProjectComponent > projectForm
6     standalone: true,
7     imports: [],
8     templateUrl: './client-project.component.html',
9     styleUrls: ['./client-project.component.css']
10 })
11 export class ClientProjectComponent {
12
13
14     projectForm: FormGroup = new FormGroup({
15         clientProjectId: new FormControl(0),
16         projectName: new FormControl(""),
17         startDate: new FormControl(""),
18         expectedEndDate: new FormControl(""),
19         leadByEmpId: new FormControl(""),
20         completedDate: new FormControl(""),
21         contactPerson: new FormControl(""),
22         contactPersonContactNo: new FormControl(""),
23         totalEmpWorking: new FormControl(""),
24         projectCost: new FormControl(""),
25         projectDetails: new FormControl(""),
26         contactPersonEmailId: new FormControl(""),
27         clientId: new FormControl("")
28     })

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

CREATE src/app/components/client-project/client-project.component.html (30 bytes)

CREATE src/app/components/client-project/client-project.component.css (555 bytes)


```
TS client-project.component.ts U
src > app > components > client-project > TS client-project.component.ts > ClientProjectComponent
6
7   standalone: true,
8   imports: [ReactiveFormsModule],
9   templateUrl: './client-project.component.html',
10  styleUrls: ['./client-project.component.css']
11  })
12  export class ClientProjectComponent {
13
14    projectForm: FormGroup = new FormGroup({
15      clientProjectId: new FormControl(0),
16      projectName: new FormControl(""),
17      startDate: new FormControl(""),
18      expectedEndDate: new FormControl(""),
19      leadByEmpId: new FormControl(""),
20      completedDate: new FormControl(""),
21      contactPerson: new FormControl(""),
22      contactPersonContactNo: new FormControl(""),
23      totalEmpWorking: new FormControl(""),
24      projectCost: new FormControl(""),
25      projectDetails: new FormControl(""),
26      contactPersonEmailId: new FormControl(""),
27      clientId: new FormControl("")
28    })
29  }
30  }
31

client-project.component.html U
41 </div>
42 <div>
43   <div class="col-4">
44     <div class="card">
45       <div class="card-header bg-success">
46         New Client
47       </div>
48       <div class="card-body">
49         <form [formGroup]="projectForm">
50           <div class="row">
51             <div class="col-6">
52               <label for=""> Project Name</label>
53               <input type="text" class="form-control" >
54             </div>
55             <div class="col-6">
56               <label for=""> Start Date</label>
57               <input type="date" class="form-control" >
58             </div>
59             <div class="col-6">
60               <label for=""> Expected End Date </label>
61               <input type="date" class="form-control" >
62             </div>
63             <div class="col-6">
64               <label for=""> Contact Name</label>
65               <input type="text" class="form-control" >
66             </div>
67           </div>
68         </form>
69       </div>
70     </div>
71   </div>
72 </div>
```

form model the model you have to add in
import statement fine

```
4 @Component({
5   selector: 'app-client-project',
6   standalone: true,
7   imports: [ReactiveFormsModule],
8   templateUrl: './client-project.component.html',
9   styleUrls: ['./client-project.component.css']
10 })
11 export class ClientProjectComponent {
12
13
14   projectForm: FormGroup = new FormGroup({
15     clientProjectId: new FormControl(0),
16     projectName: new FormControl(""),
17     startDate: new FormControl(""),
18     expectedEndDate: new FormControl(""),
19     leadByEmpId: new FormControl(""),
20     completedDate: new FormControl(""),
21     contactPerson: new FormControl(""),
22     contactPersonContactNo: new FormControl(""),
23     totalEmpWorking: new FormControl(""),
24     projectCost: new FormControl(""),
25     projectDetails: new FormControl(""),
26     contactPersonEmailId: new FormControl(""),
27     clientId: new FormControl("")
28   })
29 }

67 <div class="col-6">
68   <label for=""> Contact
69   <input type="text"
70   </div>
71 <div class="col-6">
72   <label for=""> Contact Email</label>
73   <input type="text" formControlName="contactPersonEmailId" class="form-control" >
74 </div>
75 <div class="col-6">
76   <label for=""> Total Employee Count</label>
77   <input type="text" formControlName="totalEmpWorking" class="form-control" >
78 </div>
79 <div class="col-6">
80   <label for="">Project Costing</label>
81   <input type="text" formControlName="projectName" class="form-control" >
82 </div>
83 <div class="col-12">
84   <label for="">Project Description</label>
85   <textarea formControlName="projectDetails" class="form-control" > </textarea>
86 </div>
87 <div class="col-6">
88   <label for="">Project Costing</label>
89   <select formControlName="projectName" class="form-control" >
90     <option value="">Select Client</option>
91   </select>
92 </div>
```

understand what the project was about

```
export class ClientProjectComponent {
83
84   <div class="col-12">
85     <label for="">Project Description</label>
86     <textarea formControlName="projectDetails" class="form-control" > </textarea>
87   </div>
88   <div class="col-6">
89     <label for="">Select Client </label>
90     <select formControlName="projectName" class="form-control" >
91       <option value="">Select Client</option>
92     </select>
93   </div>
94   <div class="col-6">
95     <label for="">Lead By</label>
96     <select formControlName="projectName" class="form-control" >
97       <option value="">Select Employee</option>
98     </select>
99   </div>
100 }
```