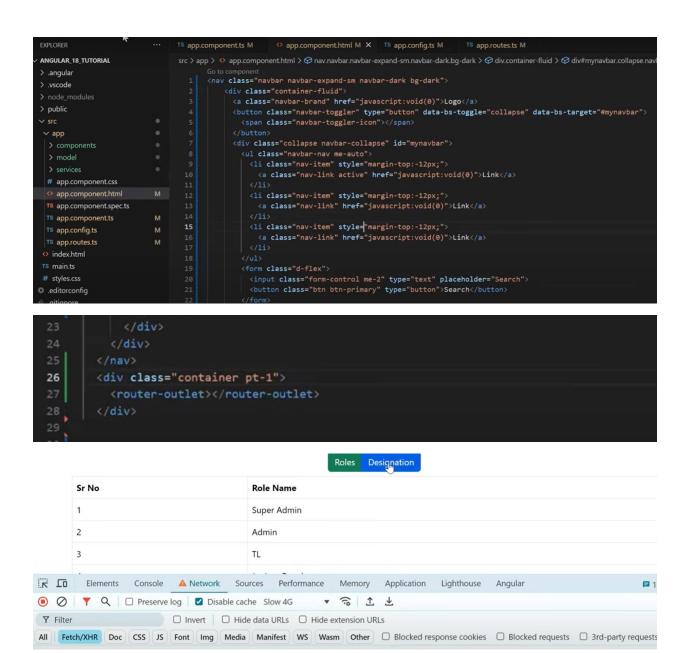
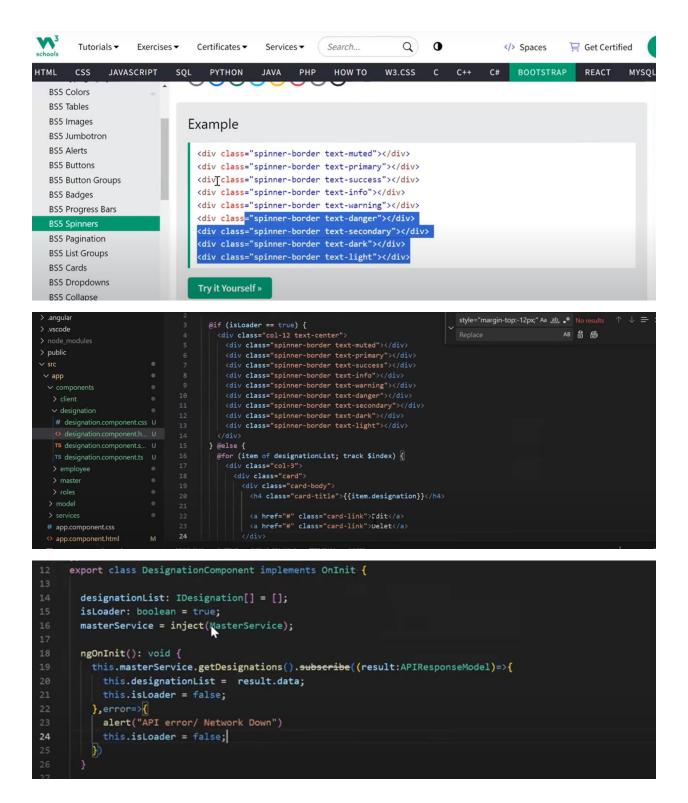
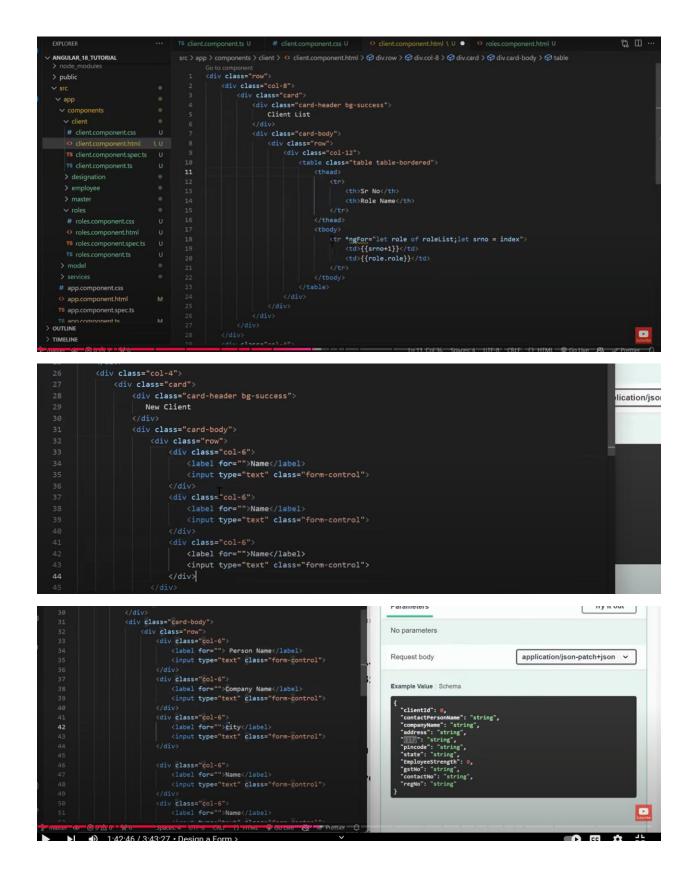
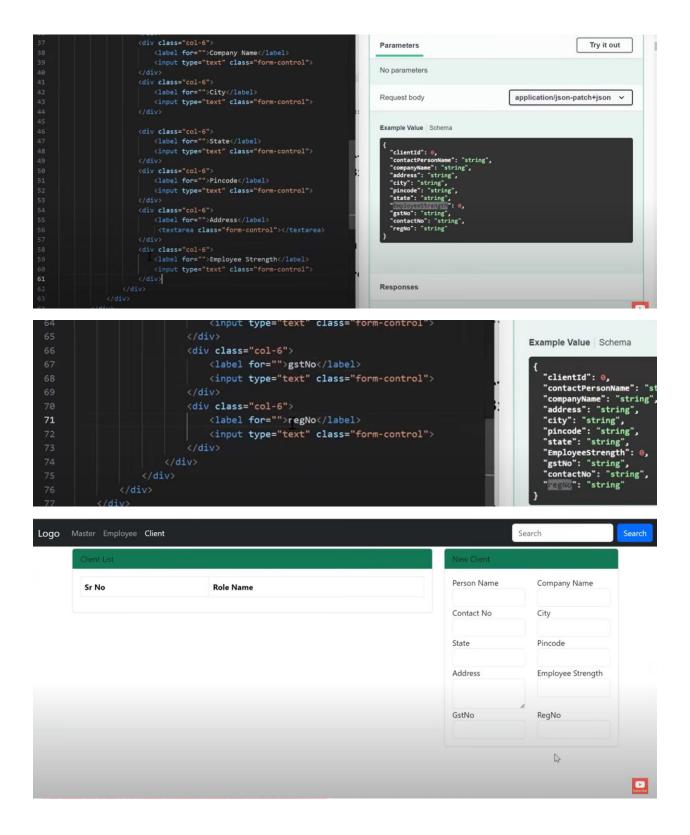
```
⇔ app.component.html M
                                             TS app.config.ts M
TS app.component.ts M
                                                                TS app.routes.ts X
     import { Routes } from '@angular/router';
     import { MasterComponent } from './components/master/master.component';
             path: '',
             redirectTo: 'master',
             pathMatch: 'full'
 10
             path: 'master',
              component:MasterComponent
 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
                component: MasterComponent
                path: 'employee',
                component: EmployeeComponent
           },
                path: 'client',
                component:ClientComponent
  BS5 Progress Bars
                            A navigation bar is a navigation header that is placed at the top of the page:
  BS5 Spinners
  BS5 Pagination
  BS5 List Groups
  BS5 Cards
  BS5 Dropdowns
                           Network
                                    Sources Performance Memory Application
                                                                             Lighthouse
                                                                                                            19 A 662
       ▶ <h1> ... </h1>
                                                                                            Styles Computed Layout
      ▶ <div class="w3-clear nextprev"> ··· </div>
                                                                                              Y Filter
        <hr>
        <h2>Navigation Bars</h2>
                                                                                             element.style {
        A navigation bar is a navigation header that is placed at the top of the page:
                                                                                             }
        <nav class="navbar navbar-expand-sm navbar-dark bg-dark">
                                                                                              .bg-dark {
          <div class="container-fluid">
                                                                                                background-color: ■ #212
            <a class="navbar-brand" href="javascript:void(0)">Logo</a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-</pre>
                                                                                             @media (min-width:
                                                                                                                  boot:
         target="#mynavbar">
                                                                                           .navbar-expand-sm {
```

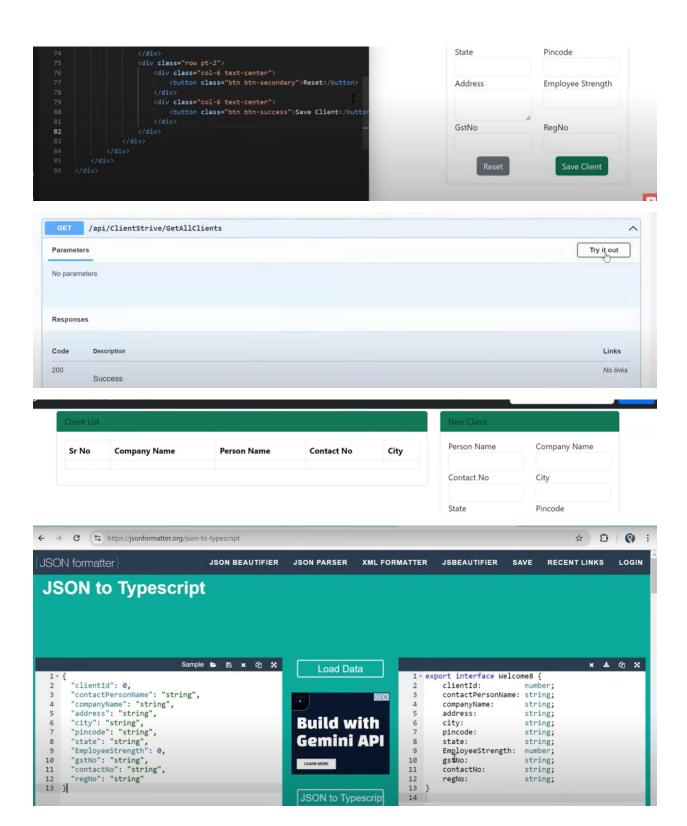


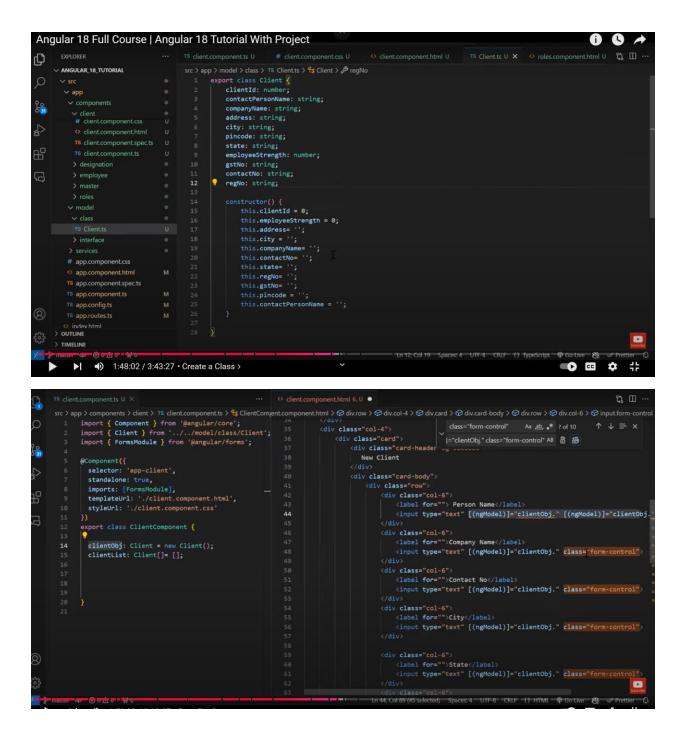


```
standalone: true,
9
       imports: [RouterOutlet, MasterComponent, RouterLink],
       templateUrl: './app.component.html',
      styleUrl: './app.component.css'
     export class AppComponent {
      title = 'angular_18_tutorial';
  designation
                                           <a class="nav-link active" routerLink="/master">Master</a>
                                           <a class="nav-link" [routerLink]="'/employee'">Employee</a>
  > master
                                             <a class="nav-link" routerLink="/client">Client</a>
 # app.component.css
                                         <form class="d-flex">
 app.component.html
                                           <input class="form-control me-2" type="text" placeholder="Search">
       @Component({
         selector: 'app-root',
        🕝 standalone: true,
 9
         imports: [RouterOutlet, MasterComponent, RouterLink, RouterLinkActive],
          templateUrl: './app.component.html',
        styleUrl: './app.component.css'
       })
       export class AppComponent {
            <a class="nav-link" routerLink="/master" routerLinkActive="active">Master</a>
            <a class="nav-link" [routerLink]="'/employee'" routerLinkActive="active">Employee</a>
            <a class="nav-link" routerLink="/client" routerLinkActive="active">Client</a>
            POST /api/ClientStrive/AddUpdateClient
  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",
"string",
"string",
"state": "string",
"gatNo": "string",
"contactNo": "string",
"regNo": "string",
"regNo": "string",
```









```
class="form-control" Aa ab, * 4 of 10
                           div class="card-body
                                   <label for=""> Person Name</label>
                                                   <input type="text" [(ngModel)]="clientObj.contactPersonName" class=</pre>
                                                <label for="">Company Name</label>
                                                   <input type="text" [(ngModel)]="clientObj.companyName" class="form-</pre>
                                           <div class="col-6">
                                                  <label for="">Contact No</label>
                                                   <input type="text" [(ngModel)]="clientObj.contactNo" class="form-contactNo" class="for
                                                   <label for="">City</label>
                                                   <input type="text" [(ngModel)]="clientObj.city" class="form-control</pre>
                                            <div class="col-6">
                                                    <input type="text" [(ngModel)]="clientObj.state" class="form-control</pre>
                                                                                                                                                                          cinput type="text" [(ngModel)]="clientObj.regNo" class="form-control
           export class ClientComponent {
             clientObj: Client = new Client();
                                                                                                                                                                 <div class="col-6 text-center">
                                                                                                                                                                        <button class="btn btn-secondary">Reset</button>
             onSaveClient() {
                                                                                                                          0
                                                                                                                                                                        89
                                                                               export class MasterService {
        TS client.service.spec.ts
       TS client.service.ts
TS master rvice.spec.ts
        TS master.service.ts
                                                                                 getDesignations():Observable<APIResponseModel> {
                                                                                     return this.http.get<APIResponseModel>("https://freeapi.miniprojectideas.com/api/ClientStrive/GetAllDesignation")
       # app.component.css
       o app.component.html
                                                      М
       TS app.component.ts
       TS app.config.ts
                                                              @ PS D:\v-18\angular18_tutorial\angular_18_tutorial\src\app\services> ng g s clinet
    PS D:\v-18\angular18_tutorial\angular_18_tutorial\src\app\services> ^C
                                                                                                                                                                                                                                                                              Powershell
                                                               o index.html
                                                                                                                                                                                                                                                                                    2 powershel.
                                                                               export class ClientService {
   index.html
 TS main.ts
                                                                                  constructor(private http: HttpClient) { }
                                                                                  getAllClients ():Observable<Client> {
gitignore
                                                                                   return this.http.get("https://freeapi.miniprojectideas.com/api/ClientStrive/GetAllDesignation")
{} angular.json
{} package-lock.json
{} package.json

    README.md

                                                                   PROBLEMS (1) OUTPUT DEBUG CONSOLE TERMINAL PORTS
{} tsconfig.app.json
stsconfig.json
                                                                                                                                                                                                                                                                                              ≥ nod
                                                               PS D:\v-18\angular18_tutorial\angular_18_tutorial> ng g environments
() tsconfig.spec.json
                                                                 CREATE src/environments/environment.ts (31 bytes)
CREATE src/environments/environment.development.ts (31 bytes)
                                                                                                                                                                                                                                                                                              2 pov
```

```
TS client.service.ts 1, U • {} angular.json M
                                                                                                                         TS environment.ts U •
                                                                                                                                                                ξβ □
src > environments > TS environment.ts > 101 environment > 125 API_URL
         export const environment = {
         API_URL: 'https://freeapi.miniprojectideas.com/api/ClientStrive/'
                               TS client.service.ts 1, U • {} angular.json M
                                                                                      TS environment.ts U ● TS environment.development.ts U ● 🗘 🛚 …
    src > environments > TS environment.development.ts > [0] environment > 1/2 API_URL
            export const environment = {
            API_URL: 'https://freeapi.miniprojectideas.com/api/ClientStrive/'
                                              providedIn: 'root'
    # app.component.css
    TS app.component.spec.ts
                                             getAllClients ():Observable<Client> {
    return this.http.get<Client>(environment.API_URL + "GetAllDesignation")
    TS app.component.ts
    TS app.config.ts
    TS app.routes.ts

    environments

(query)
                                                                                                                     Clear
Responses
Curl
 curl -X 'DELETE' \
  'https://freeapi.miniprojectideas.com/api/ClientStrive/DeleteClientByClientId?clientId=1' \
  -H 'accept: text/plain'
 https://freeapi.miniprojectideas.com/api/ClientStrive/DeleteClientByClientId?clientId=1
Server response
 200
                                           constructor(private http: HttpClient) { }
                                           getAllClients ():Observable<APIResponseModel> {
   return this.http.get<APIResponseModel>(environment.API_URL + "GetAllDesignation")
app.component.html
                                           addUpdate (obj:Client):Observable<APIResponseModel> {
TS app.component.spec.ts
                                             return this.http.post<APIResponseModel>(environment.API_URL + "GetAllDesignation",obj)
TS app.component.ts
                                           deletClientById (id:number):Observable<APIResponseModel> {
TS app.config.ts
                                             return this.http.get<APIResponseModel>(environment.API_URL + "DeleteClientByClientId?clientId="+id)
```

```
export class ClientComponent implements OnInit{
                                            clientObj: Client = new Client();
clientList: Client[]= [];
                                            clientService = inject(ClientService);
                                            ngOnInit(): void {
                                    24
25
                                            loadClient() {
                                              this.clientService.getAllClients().subscribe((res:APIResponseModel)=>{
ANGULAR_18_TUTORIAL
                                           clientList: Client[]= [];
v src
                                           ngOnInit(): void {
                                          this.loadClient();
}
                                           loadClient() {
                                           this.clientService.getAllClients().subscribe((res:APIResponseModel)=>{
                                           onSaveClient() {
                                            this.clientService.addUpdate(this.clientObj).subscribe((res:APIResponseModel)=>{
                                              if(res.result) {
   alert("Cleint Created Sucesss")
                                                 alert(res.message)
                                          ngOnInit(): void {
                                            this.loadClient();
                                          loadClient() {
                                            this.clientService.getAllClients().subscribe((res:APIResponseModel)=>{
                                             this.clientList = res.data;
                                          onSaveClient() {
                                            this.clientService.addUpdate(this.clientObj).subscribe((res:APIResponseModel)=>{
                                             if(res.result) {
    alert("Cleint Created Sucesss");
                                                this.loadClient();
                                               } else {
                                                alert(res.message)
# app.component.css
                                                                    TERMINAL
                                                                                     Company NamePerson Name
```

```
constructor(private http: HttpClient) { }

constructor(private http:
```

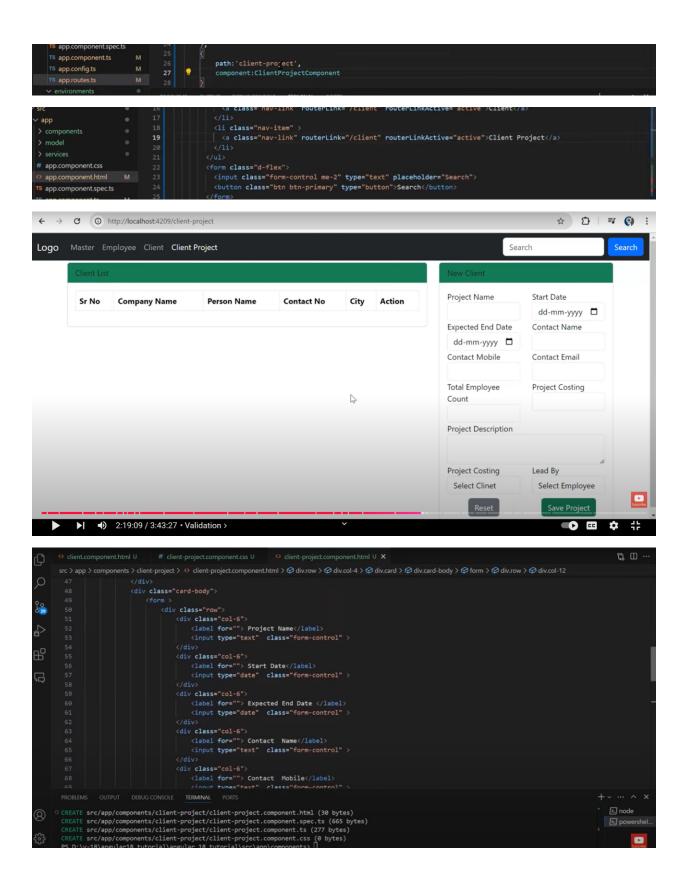




```
onDelete(id: number) {
   const isDelte = confirm("Are you sure want to delete");
                                      this.clientService.deletClientById(id).subscribe((res:APIResponseModel)=>{
                                       if(res.result) {
   alert("Cleint Delete Sucesss");
                                         alert(res.message)
 > employee
                                  deletClientById (id:number):Observable<APIResponseModel> {
                                   return this.http.delete APIResponseModel (environment.API_URL + "DeleteClientByClientId?clientId="+id)
                          PROBLEMS (1) OUTPUT DEBUG CONSOLE TERMINAL PORTS
                                                              @for (item of clientList; track $index) {
                                                                     {{item.companyName}}
{{item.contactPersonName}}
                                                                     {{item.contactNo}}
                                                                       0
                                 this.clientObj = data;
                                  onEdit(data: Client) {
                                  onDelete(id: number) {
TS clostcomponents
                                   if(isDelte) {
                                     this.clientService.deletClientById(id).subscribe((res:APIResponseModel)=>{
                                                    <button class="btn btn-success" (click)="onSaveClient()">
                                                       {{clientObj.clientId != 0 ?'Update Client': 'Save Client'}}
                    <input type="text" #personName="ngModel" required [(ngModel)]="clientObj.contactPersonName" class="form-control">
```

```
<div class="col-6";
               <label for=""> Person Name</label>
                <input type="text" #personName="ngModel" required [(ngModel)]="clientObj.contactPersonName" class="form-control">
               @if (personName.touched) {
                    <div class="text-danger" >
  @if(personName?.errors?.['required']) {
                             <span >This Is Required</span</pre>
            <div class="col-6":
                <label for=""> Person Name</label>
                <input type="text" #personName="ngModel" minlength="5" required [(ngModel)]="clientObj.contactPersonName" class="form-control"</pre>
                @if (personName.touched) {
                    <div class="text-danger" >
  @if(personName?.errors?.['required']) {
                         <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.contactPers</pre>
             @if (personName.touched) {
                  <div class="text-danger
                      @if(personName?.errors?.['required']) {
                           <span >This Is Required</span</pre>
                       @if(personName?.errors?.['minlength']){
                    <input type="text" [(ngModel)]="clientObj.companyName" class="for</pre>
                                                                                                                                             \uparrow \downarrow = \times
                                                                                                ntrol" name="contactPersonName" AB 🛱 👼
                    <input type="text" [(ngModel)]="clientObj.contactNo" class="form-control" name="contactNo">
                   <label for="">City</label>
<input type="text" [(ngModel)]="clientObj.city" class="form-control" name="city">
                   <button class="btn btn-secondary">Reset</button>
               <div class="col-6 text-center">
     <button class="btn btn-success" [disabled]="!form.valid" (click)="onSaveClient()">
          {clientObj.clientId != 0 ?'Update Client':' Save Client'}}
```





```
TS client-project.component.ts U
src > app > components > client-project > 18 client-project.component.ts > ts Client-projectComponent div.row > to div.col-4 > div.col-4 > div.cord > div.
                         ▼standalone: true,
imports: [ReactiveFormsModule],
                           styleUrl: './client-project.component.css'
                                                                                                                                                                                                                                                                             <div class="card-header bg-success">
                       export class ClientProjectComponent {
                                                                                                                                                                                                                                                                                 New Client
                                                                                                                                                                                                                                                                             <div class="card-body"
                            projectForm: FormGroup = new FormGroup({
                                                                                                                                                                                                                                                                                          <form [formGroup]="projectForm">
                                  projectName: new FormControl(""),
startDate: new FormControl(""),
                                                                                                                                                                                                                                                                                                                <div class="col-6">
     <label for=""> Project Name</label>
                                  expectedEndDate: new FormControl(""),
                                                                                                                                                                                                                                                                                                                           <input type="text" class="form-control" >
                                  leadByEmpId: new FormControl(""),
                                 completedDate: new FormControl(""),
contactPerson: new FormControl(""),
                                                                                                                                                                                                                                                                                                               <div class="col-6">
                                                                                                                                                                                                                                                                                                                       <label for=""> Start Date</label>
                                 totalEmpWorking: new FormControl(""),
projectCost: new FormControl(""),
projectDetails: new FormControl(""),
                                                                                                                                                                                                                                                                                                                <div class="col-6">
                                 clientId: new FormControl("")
                                                                                                                                                                                                                                                                                                               form model the model you have to add in
```

```
<div class="col-6">
                                                                                            class="form-control" Aa ab, * 6 of 11
selector: 'app-client-project',
                                                                      class= form-control AB @ 🗇
standalone: true,
 imports: [ReactiveFormsModule],
templateUrl: './client-project.component.html',
styleUrl: './client-project.component.css'
                                                                     <input type="text" formControlName="contactPersonEmailId" class="form-control</pre>
                                                                    <label for=""> Total Employee Count</label>
projectForm: FormGroup = new FormGroup({
                                                                     <input type="text" formControlName="totalEmpWorking" class="form-control"</pre>
  projectName: new FormControl(""),
                                                                  <div class="col-6">
  startDate: new FormControl(""),
                                                                     <label for="">Project Costing</label>
  expectedEndDate: new FormControl(""),
                                                                     <input type="text" formControlName="projectName" class="form-control">
  leadByEmpId: new FormControl(""),
  completedDate: new FormControl('
  contactPerson: new FormControl(""),
contactPersonContactNo: new FormControl(""),
                                                                     <textarea formControlName="projectName" class="form-control"> </textarea>
  totalEmpWorking: new FormControl(""),
  projectCost: new FormControl(""),
projectDetails: new FormControl(""),
                                                                     <label for="">Project Costing</label>
                                                                     contactPersonEmailId: new FormControl(""),
                                                       90
 Understand what the project was about
```

```
export class ClientProjectComponent {
                                                                        <label for="">Project Description</label>
                                                                        <textarea formControlName="projectDetails" class="form-control"> </text</pre>
  projectForm: FormGroup = new FormGroup({
   clientProjectId: new FormControl(0),
                                                                    <div class="col-6">
   projectName: new FormControl(""),
                                                                      <label for="">Select Client </label>
   startDate: new FormControl(""),
                                                                        <select formControlName="projectName" class="form-control">
   expectedEndDate: new FormControl(""),
                                                          90
                                                                          <option value="">Select Client</option>
   leadByEmpId: new FormControl(""),
   completedDate: new FormControl("
   contactPerson: new FormControl(""),
                                                                    <div class="col-6">
   contactPersonContactNo: new FormControl(""),
                                                                      <label for="">Lead By</label>
   totalEmpWorking: new FormControl(""),
                                                                        <select formControlName="projectName" class="form-control">
   projectCost: new FormControl(""),
                                                                            <option value="">Select Employee</option>
   projectDetails: new FormControl(""),
   contactPersonEmailId: new FormControl(""),
   clientId: new FormControl("")
```