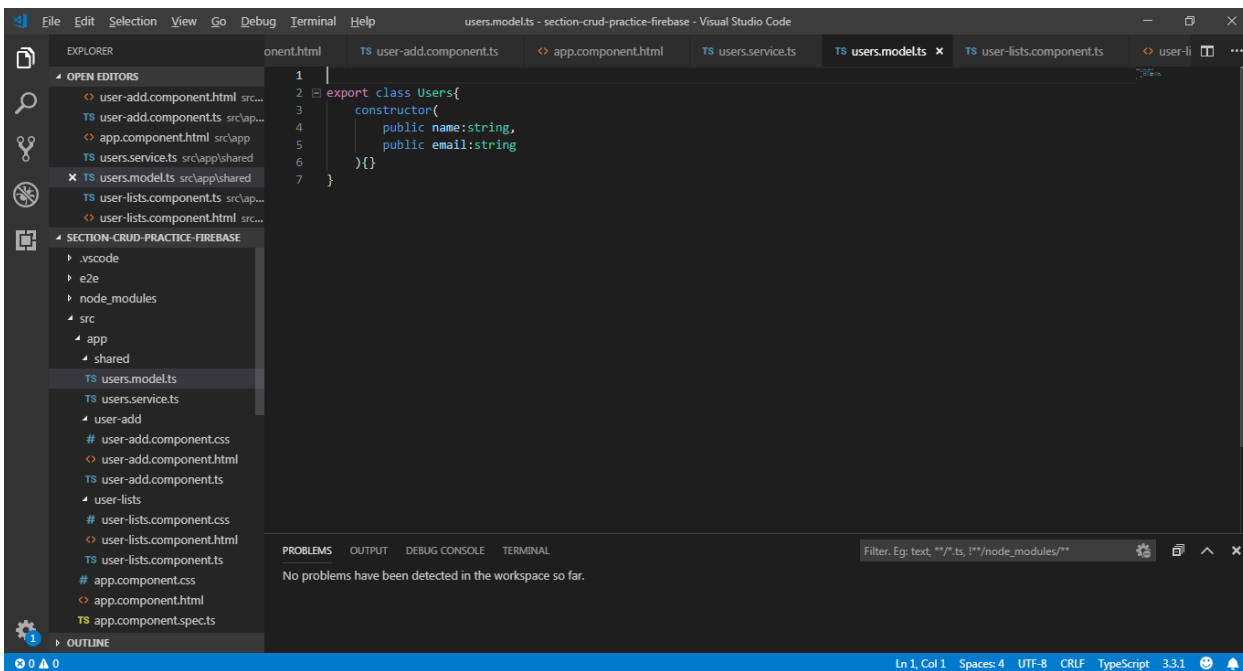


CRUD PRACTICE USING FIREBASE

1. Create a new project “ng new crud-practice”
2. Create a component “ng g c component-name”
3. Create a service “ng g s service-name”
4. Install firebase packages “npm install firebase angularfire2”

SHARED FILE

Users Model

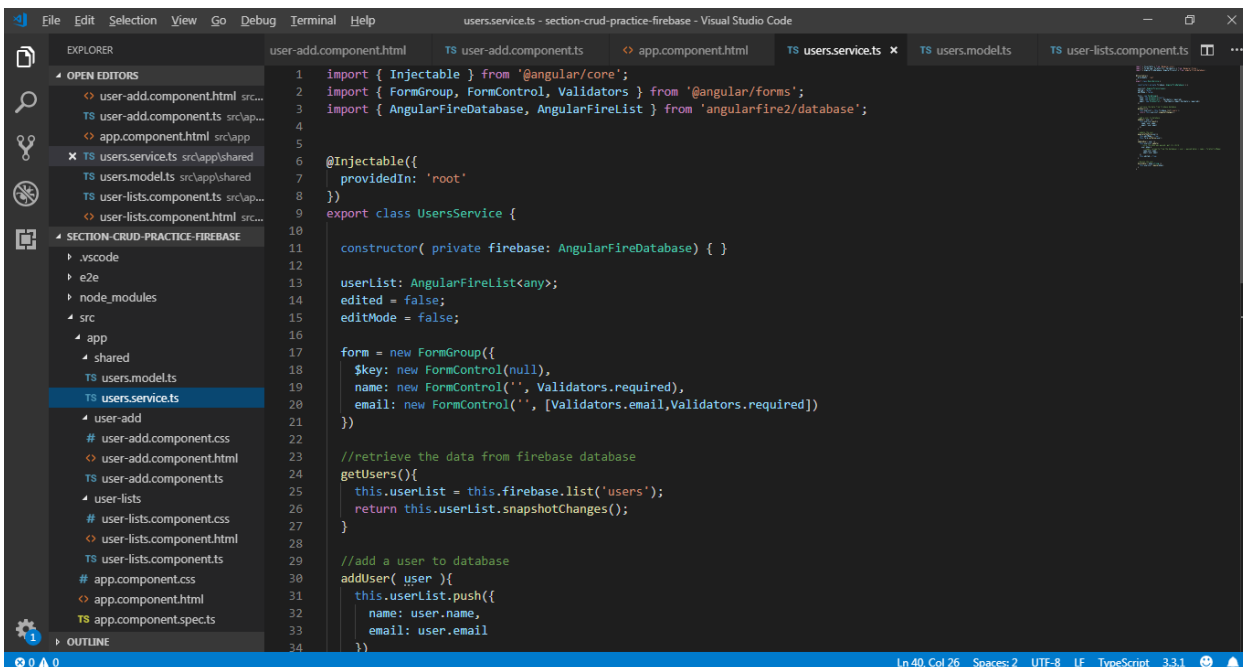


The screenshot shows the Visual Studio Code editor with the file `users.model.ts` open. The Explorer sidebar on the left shows the project structure for 'SECTION-CRUD-PRACTICE-FIREBASE', including `src/app/shared` where `users.model.ts` is located. The main editor displays the following TypeScript code:

```
1 export class Users {
2   constructor(
3     public name: string,
4     public email: string
5   ) {}
6 }
7
```

The bottom status bar indicates the cursor is at line 1, column 1, with 4 spaces, in UTF-8 encoding, CR/LF line endings, TypeScript 3.3.1.

Users Service



The screenshot shows the Visual Studio Code editor with the file `users.service.ts` open. The Explorer sidebar on the left shows the project structure, with `src/app/shared` containing `users.service.ts`. The main editor displays the following TypeScript code:

```
1 import { Injectable } from '@angular/core';
2 import { FormGroup, FormControl, Validators } from '@angular/forms';
3 import { AngularFireDatabase, AngularFireList } from 'angularfire2/database';
4
5 @Injectable({
6   providedIn: 'root'
7 })
8 export class UsersService {
9
10  constructor( private firebase: AngularFireDatabase ) { }
11
12  userList: AngularFireList<any>;
13  edited = false;
14  editMode = false;
15
16  form = new FormGroup({
17    $key: new FormControl(null),
18    name: new FormControl('', Validators.required),
19    email: new FormControl('', [Validators.email, Validators.required])
20  })
21
22  //retrieve the data from firebase database
23  getUsers(){
24    this.userList = this.firebase.list('users');
25    return this.userList.snapshotChanges();
26  }
27
28  //add a user to database
29  addUser( user ) {
30    this.userList.push({
31      name: user.name,
32      email: user.email
33    })
34  }
35}
```

The bottom status bar indicates the cursor is at line 40, column 26, with 2 spaces, in UTF-8 encoding, LF line endings, TypeScript 3.3.1.

```
28 //add a user to database
29 addUser( user ){
30     this.userList.push({
31         name: user.name,
32         email: user.email
33     })
34 }
35
36 //update the user
37 populateTheForm(user){
38     this.editMode = true;
39     this.form.setValue(user);
40 }
41
42 updateUser( user ){
43     this.userList.update(
44         //once the $key was passed, get its child
45         user.$key,{
46             //(name = property from the database):( user = passed data).( name = formControlName)
47             name:user.name,
48             email:user.email
49         })
50     this.edited = true;
51 }
52
53 //delete the user
54 deleteUser( $key:string ){
55     this.userList.remove($key);
56 }
57
58 }
```

COMPONENTS

User Add Component

```
1 <div class="p-5">
2   <h3 *ngIf="!editMode">Add User</h3>
3   <h3 *ngIf="editMode">Edit User Profile</h3>
4   <hr>
5   <form [formGroup]="this.userService.form" (ngSubmit)="onSubmit()">
6     <div class="form-group">
7       <input type="hidden" formControlName="$key">
8       <input class="form-control m-1" type="text" placeholder="Name" formControlName="name">
9       <small class="text-danger d-block" *ngIf="this.userService.form.get('name').invalid &&
10         this.userService.form.get('name').touched">Opp! We need your name!</small>
11       <input type="text" class="form-control m-1" placeholder="Email" formControlName="email">
12       <small class="text-danger d-block" *ngIf="this.userService.form.get('email').invalid &&
13         this.userService.form.get('email').touched">Sorry! But we also need your email!</small>
14     </div>
15     <button class="btn btn-success mt-2" type="submit" [disabled]="!this.userService.form.valid">Add User</button>
16   </form>
17   <div class="alert alert-info" *ngIf="success">
18     Submitted successfully
19   </div>
20 </div>
```

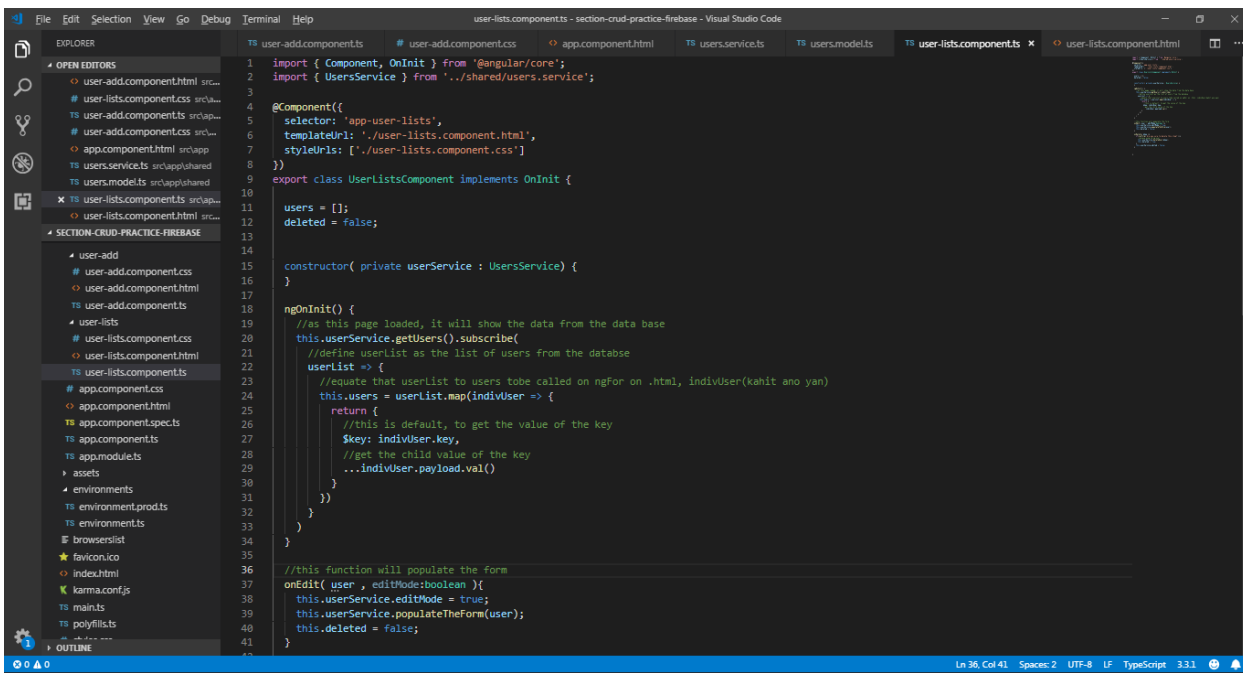
Style of the Input

```
input.ng-invalid.ng-touched{
  border: 2px solid red;
}
```

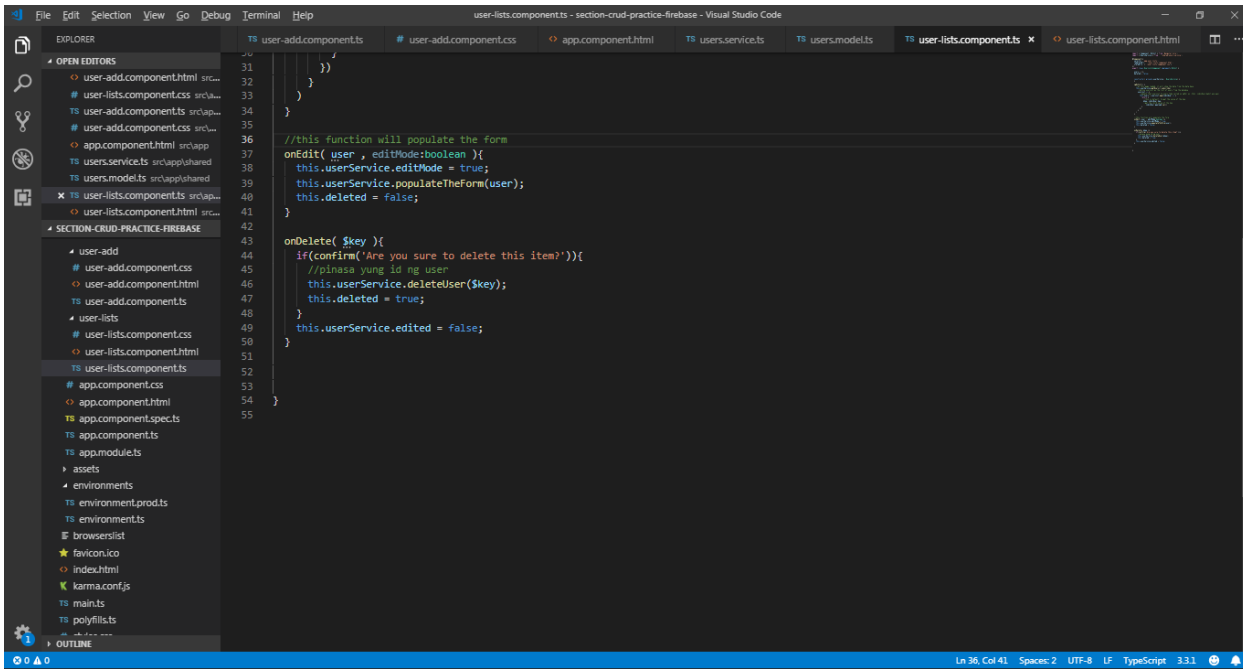
```
1 import { Component, OnInit } from '@angular/core';
2 import { UsersService } from '../shared/users.service';
3
4 @Component({
5   selector: 'app-user-add',
6   templateUrl: './user-add.component.html',
7   styleUrls: ['./user-add.component.css']
8 })
9 export class UserAddComponent implements OnInit {
10   success = false
11
12   editMode = this.userService.editMode
13
14   constructor( private userService:UsersService ) { }
15
16   ngOnInit() {
17   }
18
19   onSubmit(){
20     if(this.userService.form.valid){
21       if(this.userService.form.get('$key').value == null){
22         this.userService.addUser(this.userService.form.value);
23         this.success = true;
24       }else{
25         //this will edit the value
26         this.userService.updateUser(this.userService.form.value)
27       }
28       this.userService.form.reset()
29     }
30   }
31 }
32
33 }
```

User List Component

```
1 <div class="p-5">
2   <div class="row">
3     <div class="col-1g-10">
4       <input class="form-control" type="text" placeholder="Search">
5     </div>
6     <div class="col-1g-2">
7       <button class="btn btn-default btn-block">Clear</button>
8     </div>
9   </div>
10
11   <div class="alert alert-lime m-1 text-center">
12     <p *ngIf="deleted">Successfully Deleted.</p>
13     <p *ngIf="this.userService.edited"> Successfully Edited</p>
14   </div>
15
16   <table class="table table-responsive">
17     <thead>
18       <tr>
19         <td>Name</td>
20         <td>Email</td>
21         <td>Actions</td>
22       </tr>
23     </thead>
24     <tbody>
25       <tr *ngFor="let user of users">
26         <td>{{ user.name }}</td>
27         <td>{{ user.email }}</td>
28         <td>
29           <button class="btn btn-primary" (click)="onEdit( user )">Edit</button>
30           <button class="btn btn-danger" (click)="onDelete( user.$key )">Delete</button>
31         </td>
32       </tr>
33     </tbody>
34   </table>
35 </div>
```



```
1 import { Component, OnInit } from '@angular/core';
2 import { UsersService } from '../shared/users.service';
3
4 @Component({
5   selector: 'app-user-lists',
6   templateUrl: './user-lists.component.html',
7   styleUrls: ['./user-lists.component.css']
8 })
9 export class UserListsComponent implements OnInit {
10
11   users = [];
12   deleted = false;
13
14   constructor( private userService : UsersService ) {
15   }
16
17   ngOnInit() {
18     //as this page loaded, it will show the data from the data base
19     this.userService.getUsers().subscribe(
20       //define userList as the list of users from the database
21       userList => {
22         //equate that userList to users to be called on ngfor on .html, indivUser(kahit ano yan)
23         this.users = userList.map(indivUser => {
24           return {
25             //this is default, to get the value of the key
26             $key: indivUser.key,
27             //get the child value of the key
28             ...indivUser.payload.val()
29           }
30         })
31       }
32     )
33   }
34
35   //this function will populate the form
36   onEdit( user , editMode:boolean ){
37     this.userService.editMode = true;
38     this.userService.populateTheForm(user);
39     this.deleted = false;
40   }
41
42   onDelete( $key ){
43     if(confirm('Are you sure to delete this item?')){
44       //pinasa yung id ng user
45       this.userService.deleteUser($key);
46       this.deleted = true;
47     }
48     this.userService.edited = false;
49   }
50 }
51
52
53
54
55
```

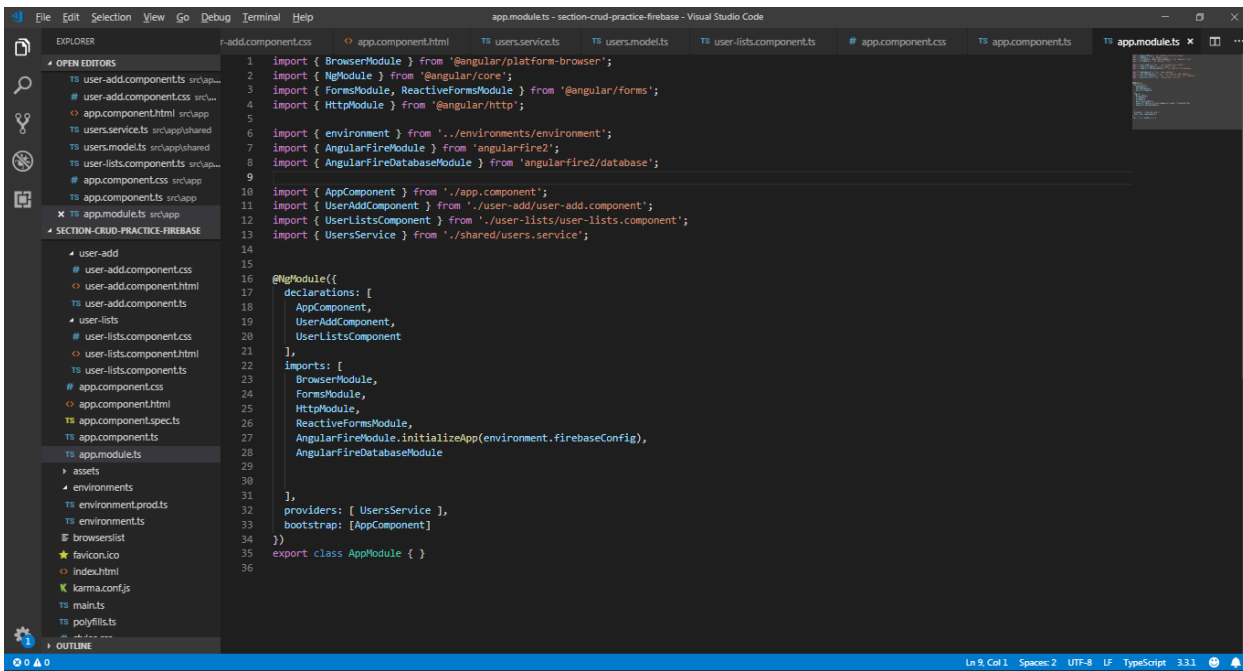


```
1 <h2>CRUD Practice</h2>
2
3 <div class="row">
4   <div class="col-lg-5">
5     <app-user-add></app-user-add>
6   </div>
7   <div class="col-lg-7">
8     <app-user-lists></app-user-lists>
9   </div>
10 </div>
11
12 </div>
13
```

App Component

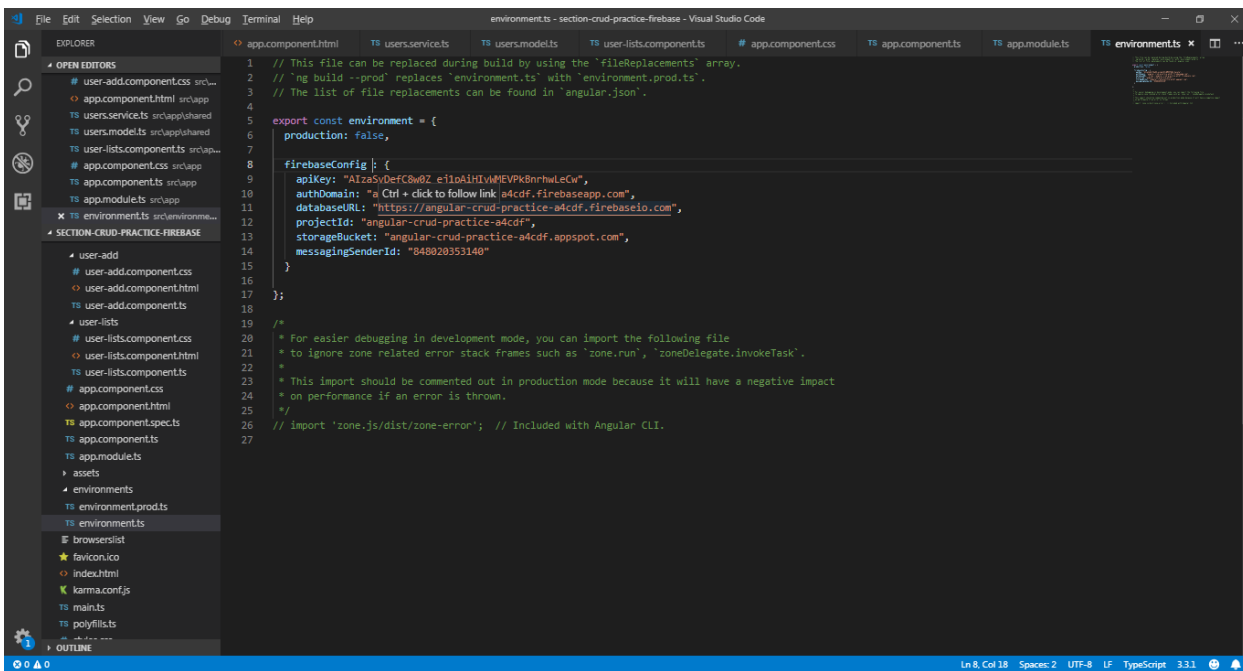
```
<div class="container-fluid">
  <h2>CRUD Practice</h2>
  <div class="row">
    <div class="col-lg-5">
      <app-user-add></app-user-add>
    </div>
    <div class="col-lg-7">
      <app-user-lists></app-user-lists>
    </div>
  </div>
</div>
```

App Module



```
1 import { BrowserModule } from '@angular/platform-browser';
2 import { NgModule } from '@angular/core';
3 import { FormsModule, ReactiveFormsModule } from '@angular/forms';
4 import { HttpClientModule } from '@angular/http';
5
6 import { environment } from '../environments/environment';
7 import { AngularFireModule } from 'angularfire2';
8 import { AngularFireDatabaseModule } from 'angularfire2/database';
9
10 import { AppComponent } from './app.component';
11 import { UserAddComponent } from './user-add/user-add.component';
12 import { UserListsComponent } from './user-lists/user-lists.component';
13 import { UsersService } from './shared/users.service';
14
15 @NgModule({
16   declarations: [
17     AppComponent,
18     UserAddComponent,
19     UserListsComponent
20   ],
21   imports: [
22     BrowserModule,
23     FormsModule,
24     HttpClientModule,
25     ReactiveFormsModule,
26     AngularFireModule.initializeApp(environment.firebaseConfig),
27     AngularFireDatabaseModule
28   ],
29   providers: [ UsersService ],
30   bootstrap: [AppComponent]
31 })
32 export class AppModule { }
```

Environment



```
1 // This file can be replaced during build by using the 'fileReplacements' array.
2 // 'ng build --prod' replaces 'environments.ts' with 'environment.prod.ts'.
3 // The list of file replacements can be found in 'angular.json'.
4
5 export const environment = {
6   production: false,
7
8   firebaseConfig: {
9     apiKey: "AIzaSyDefCw0Z_e1l0AihlVMEVPk8nrhwLeCw",
10     authDomain: "a Ctrl + click to follow link a4cdf.firebaseio.com",
11     databaseURL: "https://angular-crud-practice-a4cdf.firebaseio.com",
12     projectId: "angular-crud-practice-a4cdf",
13     storageBucket: "angular-crud-practice-a4cdf.appspot.com",
14     messagingSenderId: "848020353140"
15   }
16 };
17
18 /*
19  * For easier debugging in development mode, you can import the following file
20  * to ignore zone related error stack frames such as 'zone.run', 'zoneDelegate.invokeTask'.
21  * This import should be commented out in production mode because it will have a negative impact
22  * on performance if an error is thrown.
23  */
24 // import 'zone.js/dist/zone-error'; // Included with Angular CLI.
```