

Event Management App:

Source Code:

db.json:

```
{
  "employees": [
    {
      "id": 1,
      "first_name": "Sebastian",
      "last_name": "Eschweiler",
      "email": "sebastian@codingthesmartway.com"
    },
    {
      "id": 2,
      "first_name": "Steve",
      "last_name": "Palmer",
      "email": "steve@codingthesmartway.com"
    },
    {
      "id": 3,
      "first_name": "Ann",
      "last_name": "Smith",
      "email": "ann@codingthesmartway.com"
    }
  ]
}
```

Package.json:

```
{
  "name": "event-management",
  "version": "0.0.0",
  "scripts": {
    "ng": "ng",
    "start": "ng serve",
    "build": "ng build",
    "json:server": "json-server --watch db.json",
    "watch": "ng build --watch --configuration development",
    "test": "ng test"
  },
  "private": true,
  "dependencies": {
    "@angular/animations": "^14.2.0",
    "@angular/common": "^14.2.0",
    "@angular/compiler": "^14.2.0",
    "@angular/core": "^14.2.0",
    "@angular/forms": "^14.2.0",
    "@angular/platform-browser": "^14.2.0",
    "@angular/platform-browser-dynamic": "^14.2.0",
    "@angular/router": "^14.2.0",

```

```

    "bootstrap": "^5.2.1",
    "jquery": "^3.6.1",
    "rxjs": "~7.5.0",
    "tslib": "^2.3.0",
    "zone.js": "~0.11.4"
  },
  "devDependencies": {
    "@angular-devkit/build-angular": "^14.2.3",
    "@angular/cli": "~14.2.3",
    "@angular/compiler-cli": "^14.2.0",
    "@types/jasmine": "~4.0.0",
    "jasmine-core": "~4.3.0",
    "karma": "~6.4.0",
    "karma-chrome-launcher": "~3.1.0",
    "karma-coverage": "~2.2.0",
    "karma-jasmine": "~5.1.0",
    "karma-jasmine-html-reporter": "~2.0.0",
    "typescript": "~4.7.2"
  }
}

```

Index.html:

```

<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>EventManagerApp</title>
<base href="/">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="icon" type="image/x-icon" href="favicon.ico">
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css"
integrity="sha384-
x0OlHFLEh07PJGoPkLv1IbcEPTNtaed2xpHsD9ESMhqIYd0nLMwNLD69Npy4HI+N"
crossorigin="anonymous">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
</head>
<body>
<script
src="https://cdn.jsdelivr.net/npm/jquery@3.5.1/dist/jquery.slim.min.js"
integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
crossorigin="anonymous"></script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootstrap.bundle.min
.js" integrity="sha384-
Event Management App (Source Code)
Fy6S3B9q64WdZWQUiU+q4/2Lc9npb8tCaSX9FK7E8HnRr0Jz8D60P9d05Vg3Q9ct"
crossorigin="anonymous"></script>
<app-root></app-root>

```

```
</body>
</html>
```

App.module.ts:

```
import { NgModule } from '@angular/core';
import { FormsModule,ReactiveFormsModule } from '@angular/forms';
import { BrowserModule } from '@angular/platform-browser';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { HttpClientModule } from '@angular/common/http';
import { AdminComponent } from './admin/admin.component';
import { EmployeesComponent } from './employees/employees.component';
import { HomeComponent } from './home/home.component';
@NgModule({
  declarations: [
    AppComponent,
    AdminComponent,
    EmployeesComponent,
    HomeComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    HttpClientModule,
    ReactiveFormsModule,
    FormsModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

crud-http.service.ts:

```
import { Injectable } from '@angular/core';
import { HttpClient, HttpClientModule } from '@angular/common/http';
import { map } from 'rxjs';
@Injectable({
  providedIn: 'root'
})
export class CrudHttpService {
  constructor(private http:HttpClient) { }
  postEmployee (data : any){
    return
    this.http.post<any>("http://localhost:3000/employees",data).pipe(map((res: any)=>{
    return res;
    })))
  }
  getEmployee(){
    return
    this.http.get<any>("http://localhost:3000/employees").pipe(map((res:any)=>{
```

```

return res;
}))
}
updateEmployee(data : any,id:number){
return
this.http.put<any>("http://localhost:3000/employees/"+id,data).pipe(map((res:any)=>{
return res;
}))
}
deleteEmployee(id:number){
return
this.http.delete<any>("http://localhost:3000/employees/"+id).pipe(map((res:any)=>{
return res;
}))
}
}
}

```

UserClass.ts:

```

export class UserClass{
  id!: number;
  first_name!: string;
  last_name!: string;
  email!: string;
}

```

App.component.html:

```

<!-- <app-signin></app-signin> -->
<router-outlet></router-outlet>
<!-- <app-dashboard></app-dashboard> -->

```

Home.component.html:

```

<nav class="navbar navbar-expand-sm">
  <h2>Event Management</h2>
  <ul class="navbar-nav ml-auto">
    <li class="nav-item">
      <a href="#" class="nav-link" routerLink="admin">Admin Login</a>
    </li>
  </ul>
</nav>
<div style="padding: 10px;align-self: center;" >
  <div style="background-image: url(https://images.unsplash.com/photo-1508780709619-79562169bc64?ixlib=rb-1.2.1&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&auto=format&fit=crop&w=1470&q=80);
width: 100%; margin-top: 30px; padding: 280px;background-repeat:
no-repeat; background-size: cover; border-radius: 8px;">

```

```

<h1 style="color:white;margin-top: 30px;text-align: right">Hello
There!</h1>
<h2 style="color: white;text-align: right">Welcome to Event Management
App</h2>
</div>
</div>

```

Home.component.css:

```

.navbar{
  background-color: #333333;
}
a{
  color: white;
  font-size: 13px;
}
h2{
  color: white;
}

```

Admin.component.html:

```

<div class="container mt-5 w-50">
  <h1 class="text-center" style="margin-top: 50px;">Admin Login</h1>
  <form [formGroup]="registerForm" (ngSubmit)="LoginAdmin()">
    <div class="form-group pt-5"><i class="fa fa-envelope"></i>
    Email
    <input type="email" formControlName="email" class="form-control"
    placeholder="Enter Admin Email">
    </div>
    <div class="form-group pt-5">
    <i class="fa fa-lock"></i>
    Password
    <input type="password" formControlName="password" class="form-control"
    placeholder="Enter Admin Password">
    </div>
    <div class="text-center">
    <button type="submit" class="btn btn-primary">Submit</button>
    </div>
  </form>
</div>

```

Admin.component.css:

```
.navbar{
  background-color: #333333;
}
a{
  color: white;
  font-size: 13px;
}
h2{
  color: white;
}
```

Admin.component.ts:

```
import { Component, OnInit } from '@angular/core';
import { FormBuilder, FormGroup } from '@angular/forms';
import { Router } from '@angular/router';
@Component({
  selector: 'app-admin',
  templateUrl: './admin.component.html',
  styleUrls: ['./admin.component.css']
})
export class AdminComponent implements OnInit {
  registerForm !: FormGroup;
  email: string="";
  password : string="";
  submitted:boolean=false;
  constructor(private builder:FormBuilder,private router:Router) { }
  ngOnInit(): void {
    this.registerForm=this.builder.group({
      email:[''],
      password:['']
    });
  }
  LoginAdmin(){
    this.submitted = true;
    this.email=this.registerForm.get('email')?.value;
    this.password=this.registerForm.get('password')?.value;
    if(this.email=='boobalan@gmail.com' && this.password=='password'){
      alert("Admin Login successfully");
      this.router.navigate(['employees']);
    }
    else{
      alert("Wrong Credential");
    }
  }
}
```

```
}  
}
```

Employees.component.html:

```
<nav class="navbar navbar-light">  
  <div class="container-fluid">  
    <h1 style="color: white;">Admin Dashboard</h1>  
    <div>  
      <!-- reset password -->  
      <button (click)="resetpassword()" type="button" class="btn btn-primary" data-  
toggle="modal" data-target="#Modal" class="btn btn-primary"><i  
class="fa fa-refresh"></i> Reset Password</button>  
      &nbsp;  
      <!-- log out -->  
      <button class="btn btn-danger" (click)="logout()"><i class="fa fa-sign-out"></i> Log  
Out</button>  
    </div>  
  </div>  
</nav>  
  
<div class="btn-add">  
  <button (click)="showAddEmployee()" type="button" data-toggle="modal"  
data-target="#exampleModal" class="btn btn-success"><i class="fa fa-plus"></i>  
Add Employee</button>  
</div>  
  
<table class="table table-border table-striped" style="margin-top:30px;">  
  <thead>  
    <tr>  
      <th>ID</th>  
      <th>First Name</th>  
      <th>Last Name</th>  
      <th>Email</th>  
      <th>Action</th>  
    </tr>  
  </thead>  
  <tbody>  
    <tr *ngFor="let employee of employeeDetails">  
      <td>{{employee.id}}</td>  
      <td>{{employee.first_name}}</td>  
      <td>{{employee.last_name}}</td>  
      <td>{{employee.email}}</td>  
      <td>  
        <button (click)="updateEmployee(employee)" class="btn btn-info" type="button" data-  
toggle="modal" data-target="#exampleModal"><i  
class="fa fa-pencil-square-o"></i> Update</button>  
        <button (click)="deleteEmployee(employee)" class="btn btn-danger mx-3"><i class="fa fa-  
trash"></i> Delete</button>  
      </td>  
    </tr>  
  </tbody>
```

```

</table>
<!-- Modal -->
<div class="modal fade" id="exampleModal" tabindex="-1" role="dialog" aria-
labelledby="exampleModalLabel" aria-hidden="true">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModalLabel">Employee Details</h5>
        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
<span aria-hidden="true">&times;</span>
        </button>
      </div>
      <div class="modal-body">
        <form [formGroup]="formValue">
          <div class="form-group">
            <label for="exampleInputEmail1">Id</label>
            <input type="number" class="form-control"
formControlName="id" id="exampleInputEmail1" placeholder="Employee Id">
          </div>
          <div class="form-group">
            <label for="exampleInputEmail1">First Name</label>
            <input type="text" class="form-control"
formControlName="firstname" id="exampleInputEmail1" placeholder="Enter First
Name">
          </div>
          <div class="form-group">
            <label for="exampleInputPassword1">Last Name</label>
            <input type="text" class="form-control"
formControlName="lastname" id="exampleInputPassword1" placeholder="Employee
Last Name">
          </div>
          <div class="form-group">
            <label for="exampleInputPassword1">Email</label>
            <input type="email" class="form-control"
formControlName="email" id="exampleInputPassword1" placeholder="Employee
Email">
          </div>
        </form>
      </div>
      <div class="modal-footer">
        <button type="button" id="close" class="btn btn-secondary" data-
dismiss="modal">Close</button>
        <button type="button" *ngIf="showAdd" (click)="AddEmployeeDetails()"
class="btn btn-primary">Add</button>
        <button type="button" *ngIf="showUpdate"
(click)="saveEmployeeDetails()" class="btn btn-primary">Update
Changes</button>
      </div>
    </div>
  </div>
</div>
<!-- Modal -->
<div class="modal fade" id="Modal" tabindex="-1" role="dialog" aria-
labelledby="exampleModalLabel" aria-hidden="true">

```


Event Management App (Source Code)

```
<div class="modal-dialog" role="document">
  <div class="modal-content">
    <div class="modal-header">
      <h5 class="modal-title" id="exampleModalLabel">Reset Password</h5>
      <button type="button" class="close" data-dismiss="modal" aria-label="Close">
        <span aria-hidden="true">&times;</span></button>
    </div>
    <div class="modal-body" *ngIf="showReset">
      <form [formGroup]="formValue">
        <div class="form-group">
          <label for="exampleInputEmail1">Old Password</label>
          <input type="password" class="form-control"
            formControlName="opassword" id="exampleInputEmail1" placeholder="Old
            Password">
        </div>
        <div class="form-group">
          <label for="exampleInputEmail1">New Password</label>
          <input type="password" class="form-control"
            formControlName="cpassword" id="exampleInputEmail1" placeholder="New
            Password">
        </div>
      </form>
    </div>
    <div class="modal-footer">
      <button type="button" id="end" class="btn btn-secondary" data-
dismiss="modal">Close</button>
      <button type="button" (click)="changePassword()" class="btn btn-primary">Change
Password</button>
    </div>
  </div>
</div>
</div>
</div>
</div>
```

Employees.component.css:

```
.navbar{
  background-color: #333333;
}
.btn-add{
  margin-top: 30px;
  margin-left: 50px;
}
```

Employee.component.ts:

```
import { Component, OnInit } from '@angular/core';
import { FormBuilder, FormGroup } from '@angular/forms';
import { CrudHttpService } from '../crud-http.service';
import { UserClass } from '../UserClass';
import { Router } from '@angular/router';
@Component({
  selector: 'app-employees',
  templateUrl: './employees.component.html',
  styleUrls: ['./employees.component.css']
})
export class EmployeesComponent implements OnInit {
  formValue ! : FormGroup;
  UserClassModelObject : UserClass = new UserClass();
  employeeedetails ! : any;
  showAdd !: boolean;
  showUpdate !:boolean;
  showReset ! :boolean;
  router: any;
  constructor(private builder:FormBuilder, private api:CrudHttpService,private router:Router)
  { }
  ngOnInit(): void {
    this.formValue = this.builder.group({
      id:[''],
      firstname : [''],
      lastname : [''],
      email:[''],
    })
    this.getAllEmployee();
  }
  showAddEmployee(){
    this.formValue.reset();
    this.showAdd=true;
    this.showUpdate=false;
    this.showReset=false;
  }
  AddEmployeeDetails(){
    this.UserClassModelObject.id = this.formValue.value.id;
    this.UserClassModelObject.first_name = this.formValue.value.firstname;
    this.UserClassModelObject.last_name = this.formValue.value.lastname;
    this.UserClassModelObject.email = this.formValue.value.email;
    this.api.postEmployee(this.UserClassModelObject).subscribe((res: any)=>{
      alert("Employee Added Successfully");
      this.formValue.reset();
      let ref =document.getElementById('close')
      ref?.click();
      this.getAllEmployee();
    },
    (err: any)=>{
      alert("Employee Not Added");
    })
  }
```

```

}
getAllEmployee(){
this.api.getEmployee().subscribe((res: any) => {
this.employeeDetails = res;
})
}
deleteEmployee(employee : any){
this.api.deleteEmployee(employee.id).subscribe( (res: any) =>{
alert("Employee Deleted");
this.getAllEmployee();
})
}
updateEmployee(employee : any){
this.showAdd=false;
this.showUpdate=true;
this.showReset=false;
this.UserClassModelObject.id =employee.id;
this.formValue.controls['id'].setValue(employee.id);
this.formValue.controls['firstname'].setValue(employee.first_name);
this.formValue.controls['lastname'].setValue(employee.last_name);
this.formValue.controls['email'].setValue(employee.email);
}
saveEmployeeDetails(){
this.UserClassModelObject.id = this.formValue.value.id;
this.UserClassModelObject.first_name = this.formValue.value.firstname;
this.UserClassModelObject.last_name = this.formValue.value.lastname;
this.UserClassModelObject.email = this.formValue.value.email;
this.api.updateEmployee(this.UserClassModelObject,this.UserClassModelObject.id).subscribe(
(res: any)=>{
alert("Updated Successfully");
this.formValue.reset();
let ref =document.getElementById('close')
ref?.click();
this.getAllEmployee();
})
}
logout(){
alert("Admin Logout Successfully");
this.router.navigate(['']);
}
resetpassword(){
this.showReset=true;
}
changePassword(){
alert("Change Password Successfully");
this.formValue.reset();
let ref =document.getElementById('end')
ref?.click();
}
}

```

App-routing.module.ts:

```

import { NgModule } from '@angular/core';

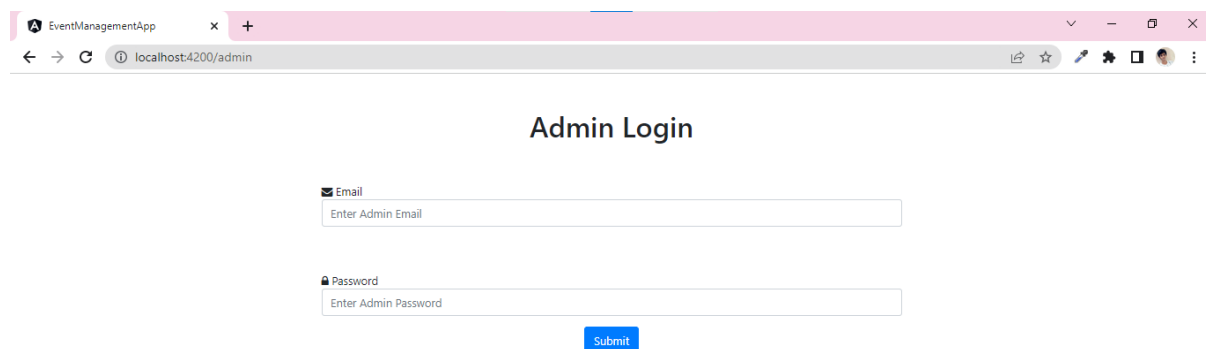
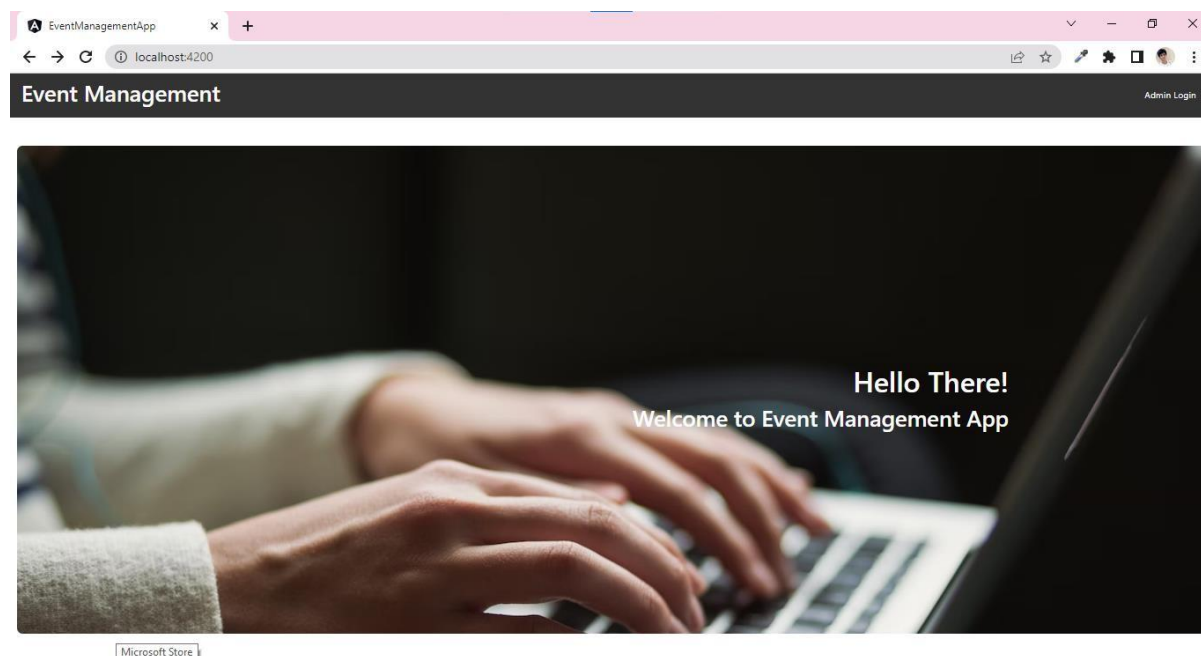
```

```
import { RouterModule, Routes } from '@angular/router';
import { DashboardComponent } from '../dashboard/dashboard.component';
import { SigninComponent } from '../signin/signin.component';

const routes: Routes = [
  {path:'signin', component:SigninComponent},
  {path:'dashboard', component:DashboardComponent}
];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
export const routingComponents=[SigninComponent,DashboardComponent]
```

OUTPUT:



localhost:4200 says
Wrong Credential

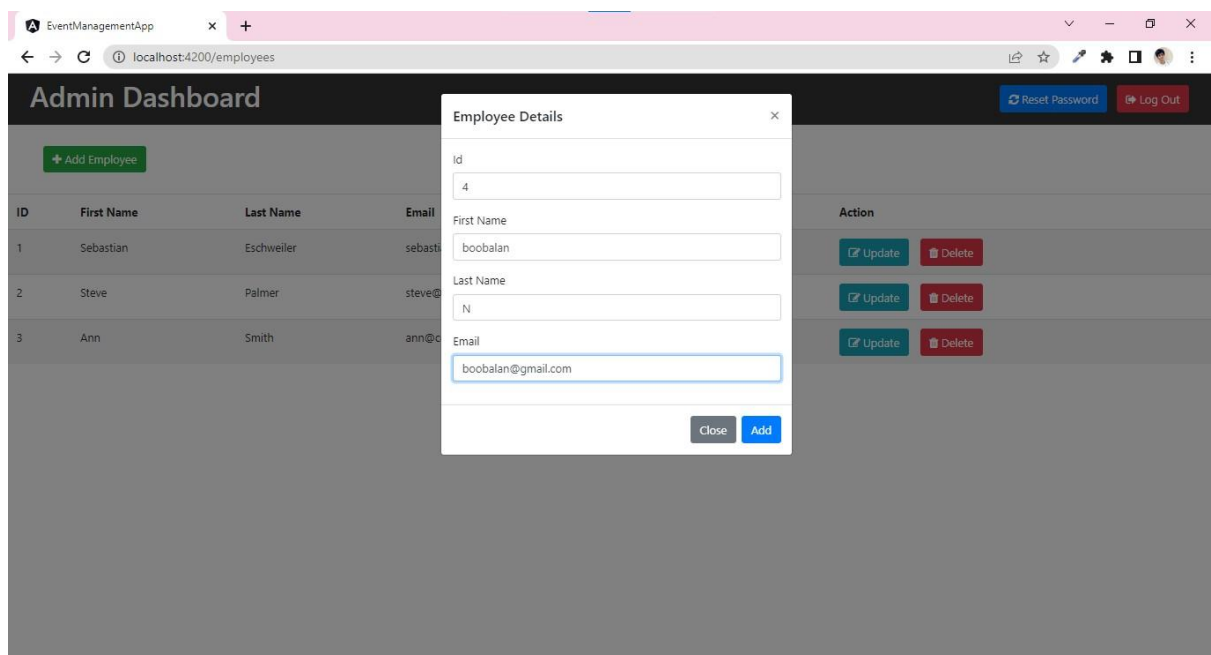
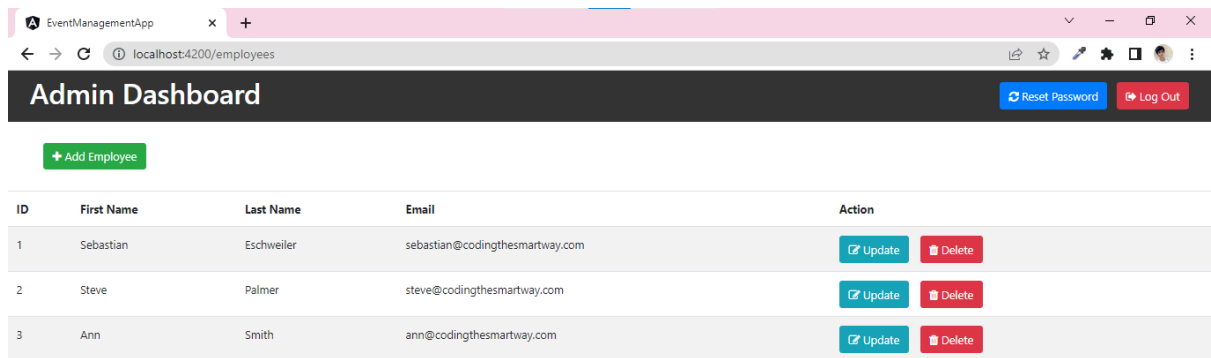
OK

Email

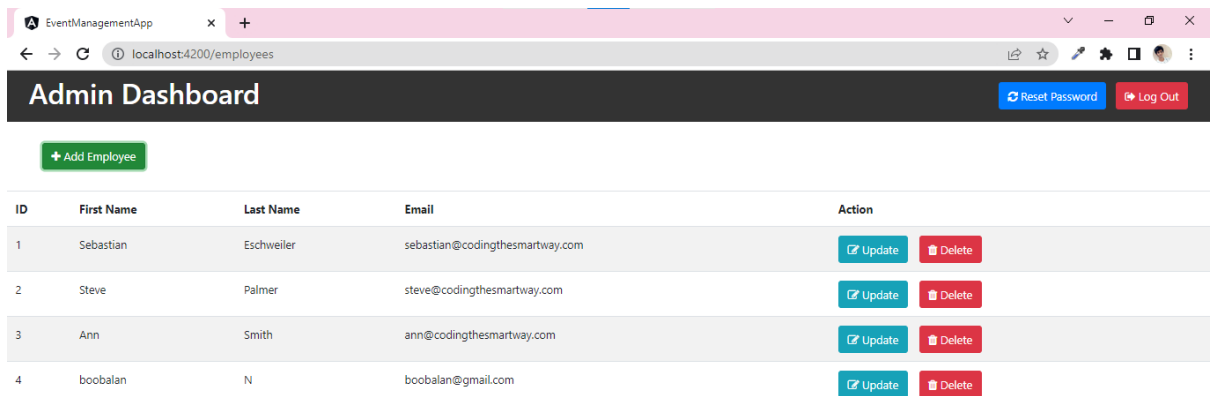
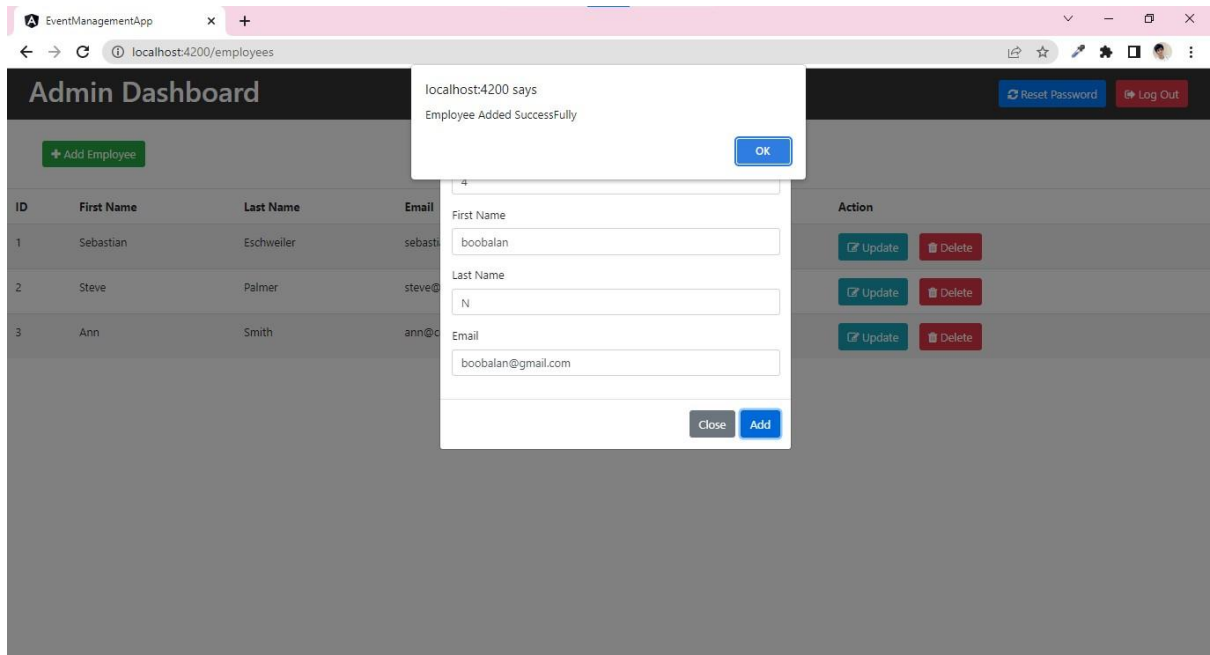
boobalan@gmail.com

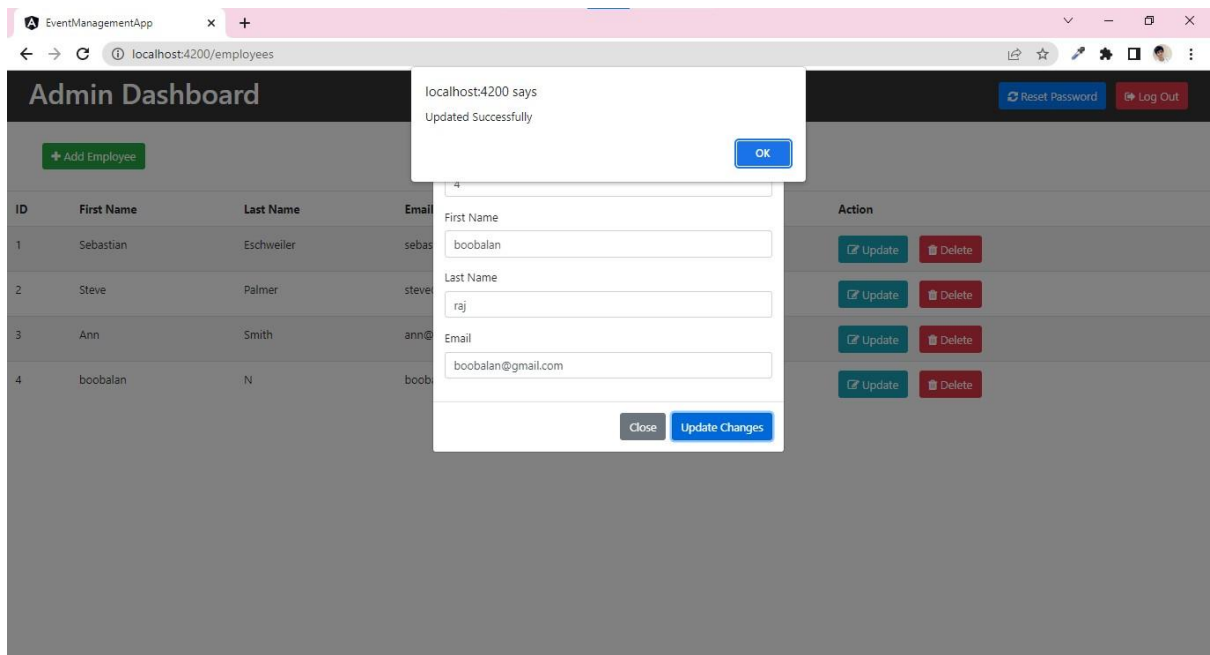
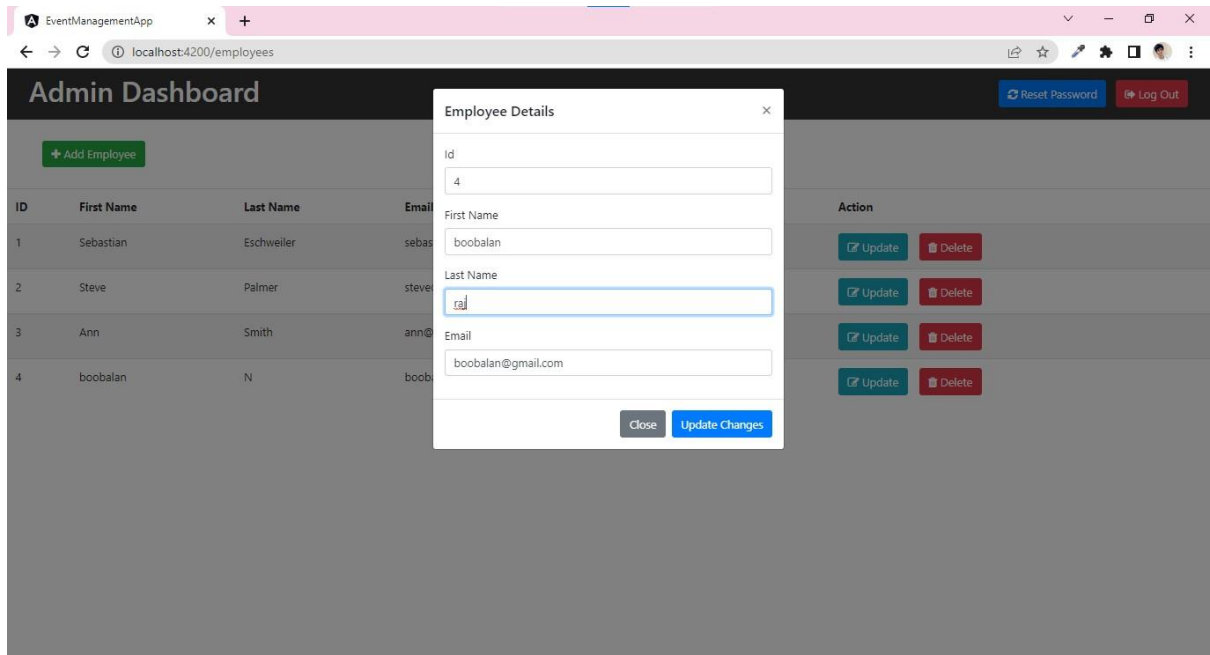
Password

Submit



Adding Employee details





EventManagementApp x +

localhost:4200/employees

Admin Dashboard

Reset Password Log Out

+ Add Employee

ID	First Name	Last Name	Email	Action
1	Sebastian	Eschweiler	sebastian@codingthesmartway.com	Update Delete
2	Steve	Palmer	steve@codingthesmartway.com	Update Delete
3	Ann	Smith	ann@codingthesmartway.com	Update Delete
4	boobalan	raj	boobalan@gmail.com	Update Delete

EventManagementApp x +

localhost:4200/employees

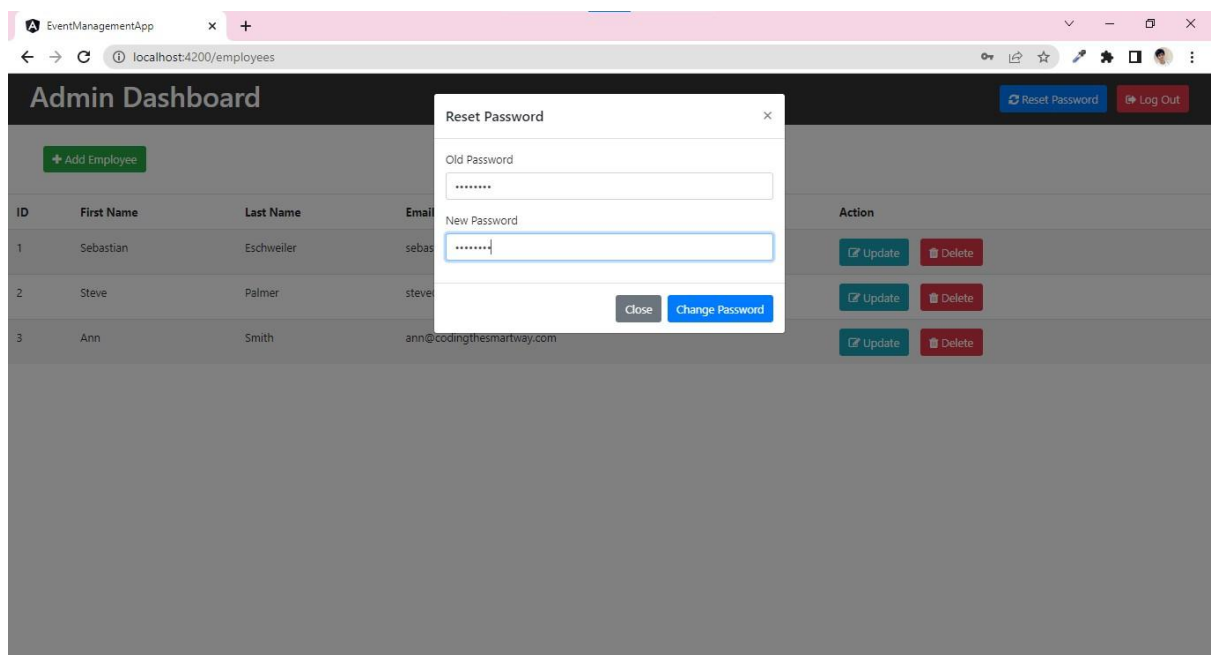
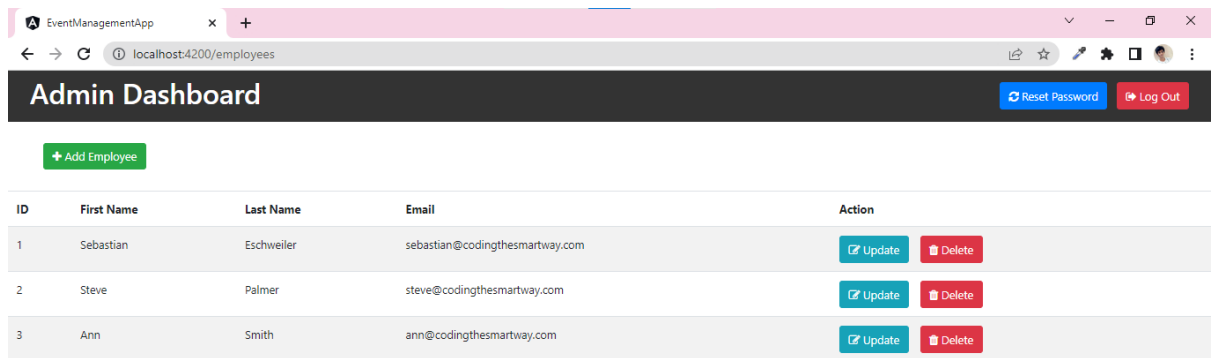
Admin Dashboard

localhost:4200 says
Employee Deleted

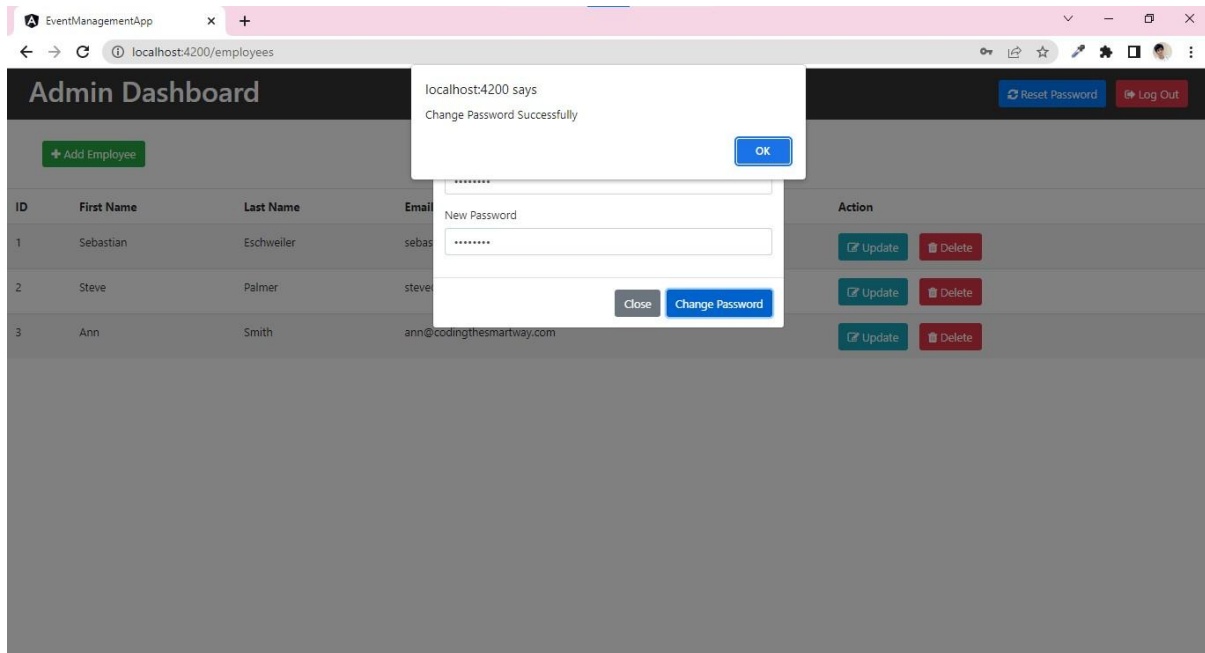
Reset Password Log Out

+ Add Employee

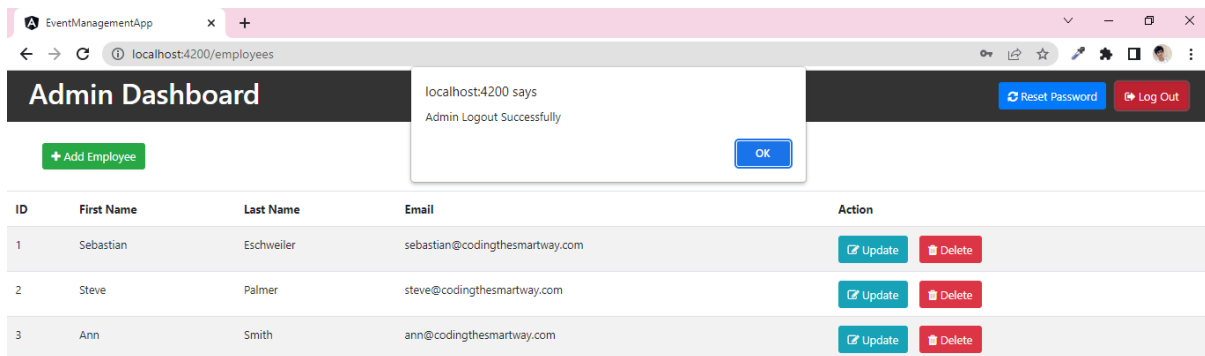
ID	First Name	Last Name	Email	Action
1	Sebastian	Eschweiler	sebastian@codingthesmartway.com	Update Delete
2	Steve	Palmer	steve@codingthesmartway.com	Update Delete
3	Ann	Smith	ann@codingthesmartway.com	Update Delete
4	boobalan	raj	boobalan@gmail.com	Update Delete



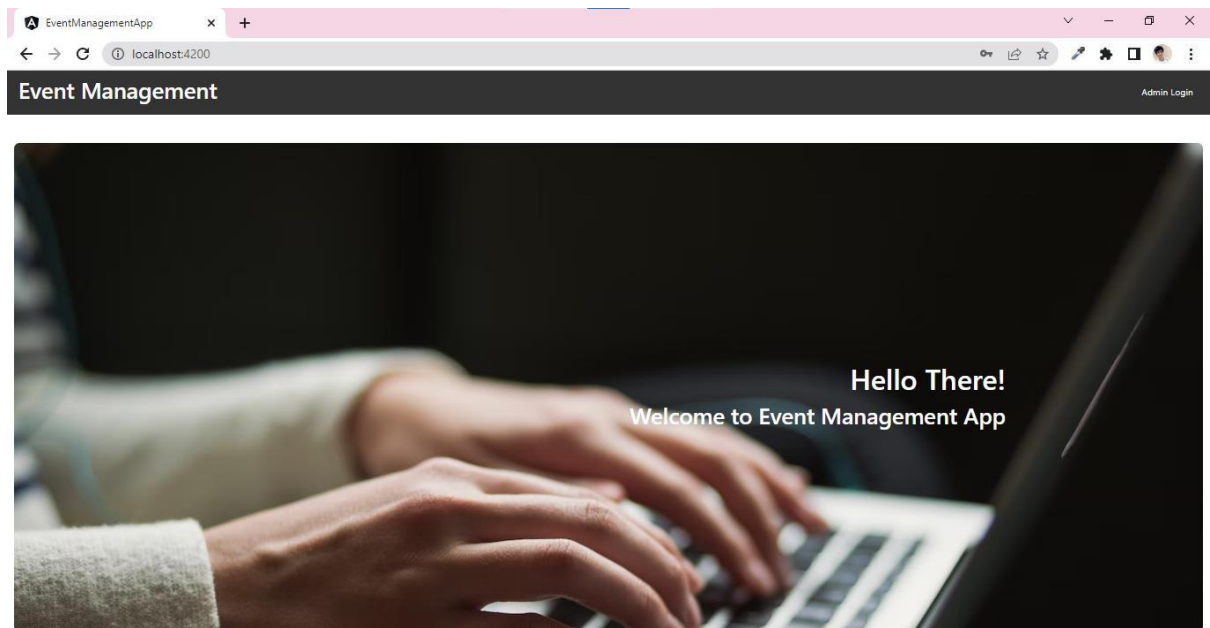
Reset Password



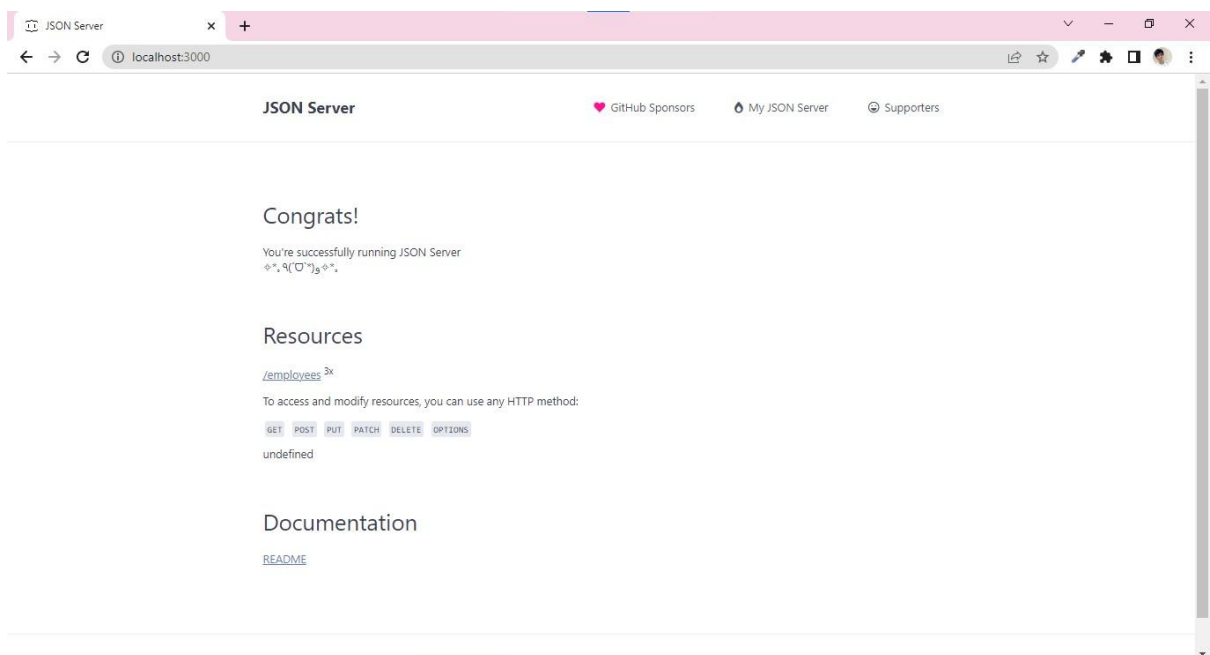
Password Reset successfully



Log out admin alert box



After logout from admin



Json server running page

```
localhost:3000/employees x +
localhost:3000/employees
[
  {
    "id": 1,
    "first_name": "Sebastian",
    "last_name": "Eschweiler",
    "email": "sebastian@codingthesmartway.com"
  },
  {
    "id": 2,
    "first_name": "Steve",
    "last_name": "Palmer",
    "email": "steve@codingthesmartway.com"
  },
  {
    "id": 3,
    "first_name": "Ann",
    "last_name": "Smith",
    "email": "ann@codingthesmartway.com"
  }
]
```