

### 3.1.3 เว็บไซต์ที่พัฒนาโดยใช้ Angular, Bootstrap, Spring Boot, PostgreSQL

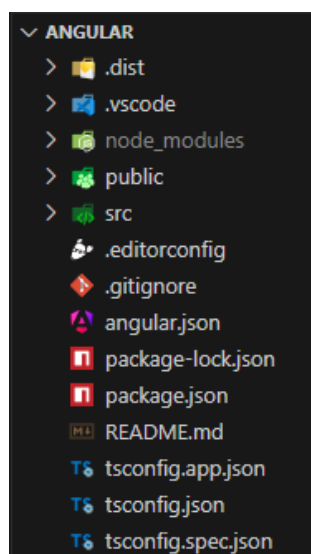
#### ขั้นตอนการสร้างโปรเจกต์

```
PS C:\Users\ITCORN> cd desktop
PS C:\Users\ITCORN\desktop> ng new angular
✓Which stylesheet format would you like to use? CSS [ https://developer.mozilla.org/docs/Web/CSS ]
✓Do you want to enable Server-Side Rendering (SSR) and Static Site Generation (SSG/Prerendering)? No
CREATE angular/angular.json (2690 bytes)
CREATE angular/package.json (1039 bytes)
CREATE angular/README.md (1530 bytes)
CREATE angular/tsconfig.json (942 bytes)
CREATE angular/.editorconfig (331 bytes)
CREATE angular/.gitignore (629 bytes)
CREATE angular/tsconfig.app.json (439 bytes)
CREATE angular/tsconfig.spec.json (449 bytes)
CREATE angular/.vscode/extensions.json (134 bytes)
CREATE angular/.vscode/launch.json (490 bytes)
CREATE angular/.vscode/tasks.json (980 bytes)
CREATE angular/src/main.ts (256 bytes)
CREATE angular/src/index.html (306 bytes)
CREATE angular/src/styles.css (81 bytes)
CREATE angular/src/app/app.component.html (20239 bytes)
CREATE angular/src/app/app.component.spec.ts (948 bytes)
CREATE angular/src/app/app.component.ts (295 bytes)
CREATE angular/src/app/app.component.css (0 bytes)
CREATE angular/src/app/app.config.ts (318 bytes)
CREATE angular/src/app/app.routes.ts (80 bytes)
CREATE angular/public/favicon.ico (15086 bytes)
✓Packages installed successfully.
Successfully initialized git.
```

รูปภาพที่ 19 ขั้นตอนการสร้างโปรเจกต์

#### โครงสร้างของโปรเจกต์

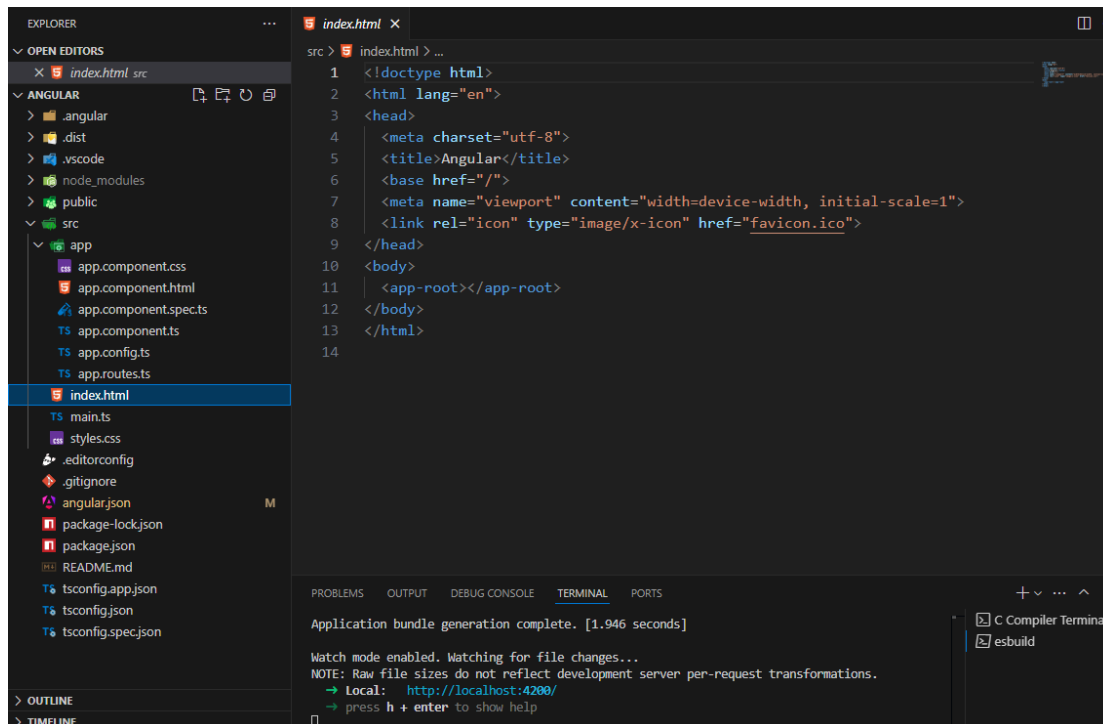
เมื่อสร้างโปรเจกต์เสร็จแล้ว จะมีโครงสร้างเริ่มต้นของโปรเจกต์แบบนี้



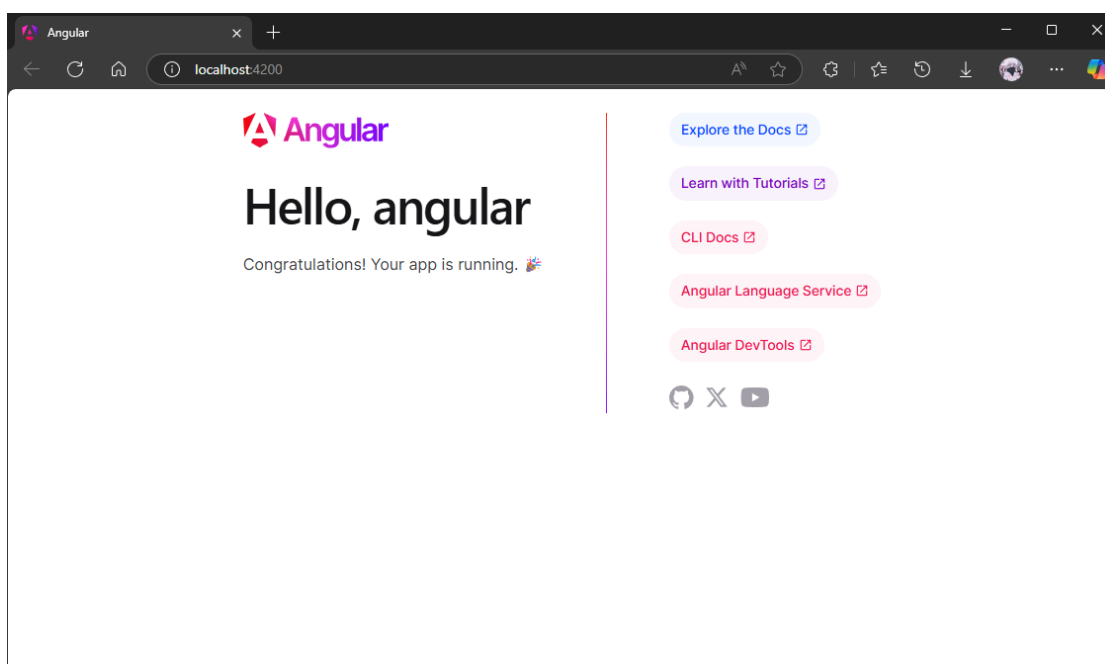
รูปภาพที่ 20 โครงสร้างของโปรเจกต์

## เริ่มการทำงานโปรเจกต์

ใช้คำสั่ง npm start แล้วไปที่ <http://localhost:4200/> เพื่อดูหน้าเว็บไซต์ที่กำลังพัฒนาของโปรเจกต์



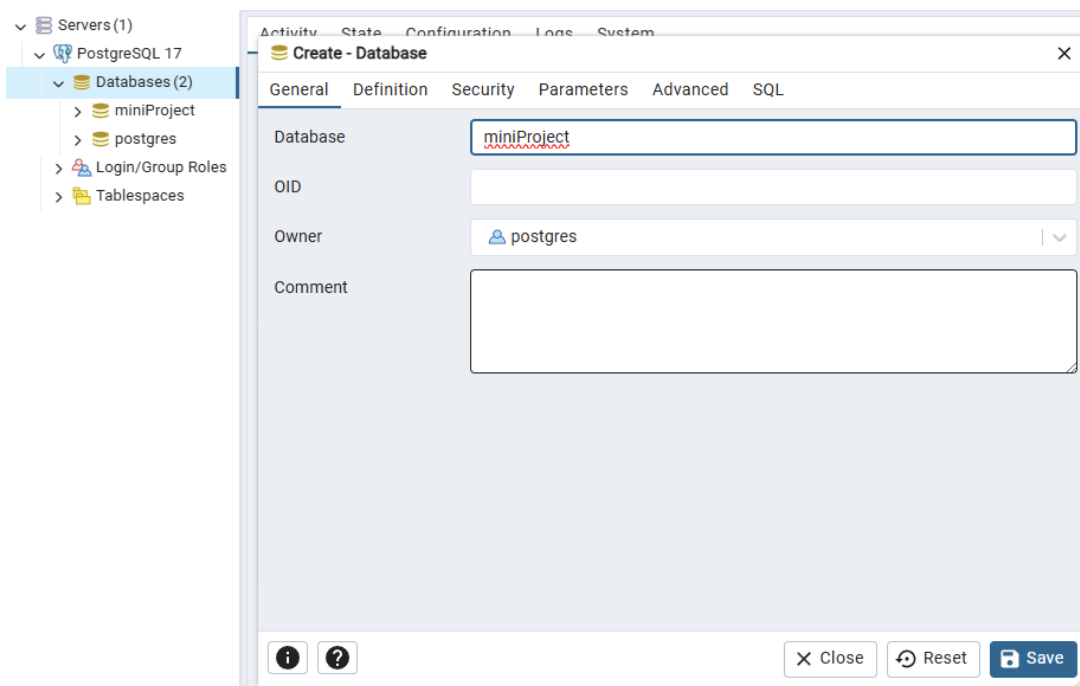
รูปภาพที่ 21 การเริ่มการทำงานโปรเจกต์



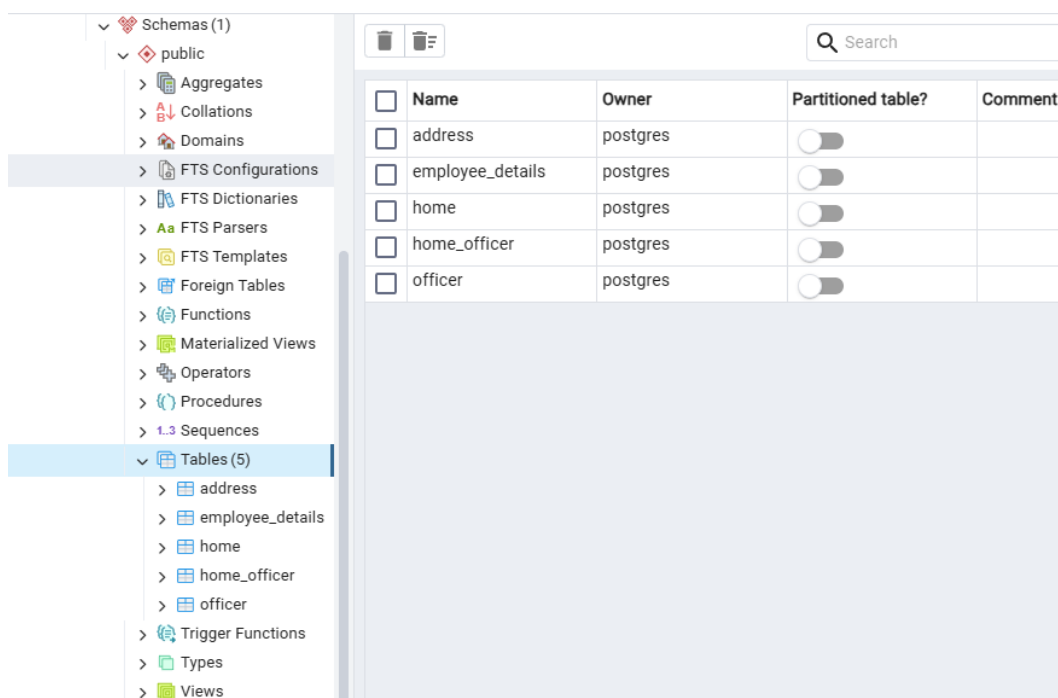
รูปภาพที่ 22 หน้าเว็บไซต์ที่กำลังพัฒนาของโปรเจกต์

## สร้าง Database

เข้าไปที่โปรแกรม PostgreSQL เข้าไป Servers ที่ต้องการแล้วคลิกขวาที่ Databases --> Create --> Database.. จากนั้นตั้งชื่อ Database ตามต้องการ



รูปภาพที่ 23 การสร้าง Database



รูปภาพที่ 24 เพิ่ม Table ลงไปใน Database

## ตั้งค่า Database ในโปรเจกต์

ไปที่ตัว Backend เราจะทำการเชื่อม Backend (Spring Boot) กับ Database (PostgreSQL) โดยตั้งค่าใน `application.yml` ของ Spring Boot ซึ่งใช้สำหรับกำหนดค่าต่างๆ ของโปรเจกต์ โดยเฉพาะส่วนที่เกี่ยวข้องกับ การเชื่อมต่อฐานข้อมูล PostgreSQL และการตั้งค่า Spring MVC สำหรับการอัปโหลดไฟล์

```

1  spring:
2      jpa:
3          hibernate:
4              ddl-auto: update
5      datasource:
6          url: jdbc:postgresql://localhost:5432/miniProject
7          username: postgres
8          password: 
9      servlet:
10         multipart:
11             enabled: true
12             max-file-size: 10MB
13             max-request-size: 10MB
14

```

รูปภาพที่ 25 เชื่อม Backend กับ Database

## ดึงข้อมูลจาก Backend

Frontend (Angular) ส่ง HTTP request ไปยัง Backend (Spring Boot) โดยจะให้ Backend API ดึงข้อมูลจาก PostgreSQL (Database) เพื่อ query ข้อมูลและส่งข้อมูลกลับมา ผ่าน JSON format เพื่อนำข้อมูลมาใช้

```

// บอก Spring Boot ว่า http://localhost:4200 (Angular app) ได้รับอนุญาต ให้เรียก API
@Bean no usages
public WebMvcConfigurer corsConfigurer() {
    return new WebMvcConfigurer() {
        @Override no usages
        public void addCorsMappings(CorsRegistry registry) {
            registry.addMapping(pathPattern: "**")
                .allowedOrigins("http://localhost:4200")
                .allowedMethods("GET", "PUT", "POST", "DELETE", "OPTIONS")
                .allowCredentials(false)
                .allowedHeaders("*");
        }
    };
}

```

รูปภาพที่ 26 เชื่อม Frontend กับ Backend

```

import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs';
import { Home } from '../model/home';

@Injectable({
  providedIn: 'root',
})
export class HomeService {
  private apiUrl = 'http://localhost:8080'; // URL ของ API ที่ใช้ในการดึงข้อมูล
  // อย่างลึบ / ก่อนใส่ path

  constructor(private http: HttpClient) {}

  getAllHomes(): Observable<Home[]> {
    return this.http.get<Home[]>(this.apiUrl + '/home');
  }

  getHomeById(homeId: number): Observable<Home> {
    return this.http.get<Home>(this.apiUrl + '/home/id/' + homeId);
  }

  deleteHome(homeId: number): Observable<void> {
    return this.http.delete<void>(this.apiUrl + '/home/' + homeId);
  }

  updateHome(home: Home): Observable<Home> {
    return this.http.put<Home>(this.apiUrl + '/home/update/', home);
  }

  addNewHome(home: Home): Observable<Home> {
    return this.http.post<Home>(
      this.apiUrl + '/home/addHomeWithOfficers/',
      home
    );
  }

  selectOfficerToHome(homeId: number, officerId: number[]): Observable<Home>
  > {
    return this.http.post<Home>(
      this.apiUrl + `/home/selectOfficerToHome/${homeId}`,
      officerId.map((id) => ({ officerId: id }))
    );
  }
}

```

รูปภาพที่ 27 โค้ดดึงข้อมูลมาแสดงผลบน Frontend

## หน้าเว็บที่ได้ทำการตกแต่งด้วย Bootstrap

### การติดตั้ง Bootstrap

ในโฟลเดอร์โปรเจกต์ Angular ให้รันคำสั่งนี้ใน Terminal

```
npm install bootstrap
```

จากนั้นทำการเชื่อม Bootstrap กับ Angular เปิดไฟล์ angular.json แล้วเพิ่ม path ของ Bootstrap CSS ในส่วน "styles" และ "scripts" ตามนี้

```
"styles": [
  "node_modules/bootstrap/dist/css/bootstrap.min.css",
  "src/styles.css"
],
"scripts": [
  "node_modules/bootstrap/dist/js/bootstrap.bundle.min.js"
]
```

```
<div class="container">
  <div class="row">
    <div class="col-12 py-4">
      <h1 class="text-center mb-3">Home List</h1>
      <div class=
"d-grid d-md-flex justify-content-md-end mb-3 mx-3">
        <button
          class="btn btn-success"
          type="button"
          title="เพิ่มข้อมูลHomeใหม่"
          (click)="pageAddHome('home')"
        >
          เพิ่มข้อมูลHomeใหม่
        </button>
      </div>
      <table class="table table-bordered table-striped">
        <thead>
          <tr class="text-center align-middle">
            <th>Home ID</th>
            <th>Home Name</th>
            <th>แก้ไขข้อมูล</th>
            <th>Officer ที่เกี่ยวข้อง</th>
            <th>ลบข้อมูล</th>
          </tr>
        </thead>
        <tbody>
          <tr class="text-center align-middle" *ngFor="let home
of datahomes">
            <td>{{ home.homeId }}</td>
            <td>{{ home.homeName }}</td>
```

รูปภาพที่ 28 โค้ดในส่วนตกแต่งหน้าเว็บ