



Instytut Informatyki  
Kolegium Nauk Przyrodniczych

Przedmiot:  
**Aplikacje internetowe 2**

**Dokumentacja projektu:**  
Aplikacja internetowa do zarządzania pracownikami z  
funkcjonalnością wysyłania wiadomości e-mail

**Wykonał:**  
Wojtowicz Aleksander  
Wrona Wojciech  
Szot Adam  
Stec Piotr

**Prowadzący:** mgr inż. Jaromir Sarzyński  
**Rzeszów 2021**

## 1. Zespół projektowy

Skład zespołu projektowego:

Imię i nazwisko	Pełna rola	Zakres czynności / odpowiedzialności
Piotr Stec	Programista	Programowanie części frontendowej aplikacji
Adam Szot	Programista	Programowanie części backendowej aplikacji opracowanie dokumentacji
Aleksander Wojtowicz	Bazy danych, programista	Programowanie części frontendowej aplikacji, stylizacja aplikacji, generowanie danych
Wojciech Wrona	Lider zespołu, programista	Programowanie części backendowej aplikacji, testowanie API

## 2. Specyfikacja projektu

### a. Opis programu / systemu

Projekt składa się z aplikacji internetowej utworzonej w architekturze mikroserwisowej. Część backendowa została napisana w języku C# jak i część frontendowa. W projekcie została wykorzystana również relacyjna baza danych PostgreSQL. Aplikacja ma pozwalać na zarządzanie pracownikami jak i również wysyłanie wiadomości e-mail dla jednego lub wielu pracowników. Wiadomości e-mail można podać do innego pracownika (DW lub z ang. CC), można dodać również pliki w postaci załączników.

### b. Cel projektu

Celem projektu jest stworzenie aplikacji webowej zajmującej się obsługą pracowników. Aplikacja składa się z części backendowej, które ma obsługiwać takie czynności jak dodanie, edycja, usuwanie i wyświetlanie danych każdego z pracowników. Dodatkowo również aplikacja będzie zawierać część frontendową, która umożliwi użytkownikowi wykonywanie wcześniej wymienionych czynności. Dodatkową funkcjonalnością będzie również obsługa wysyłania maili do dowolnej liczby odbiorców.

### c. Wymagania projektowe

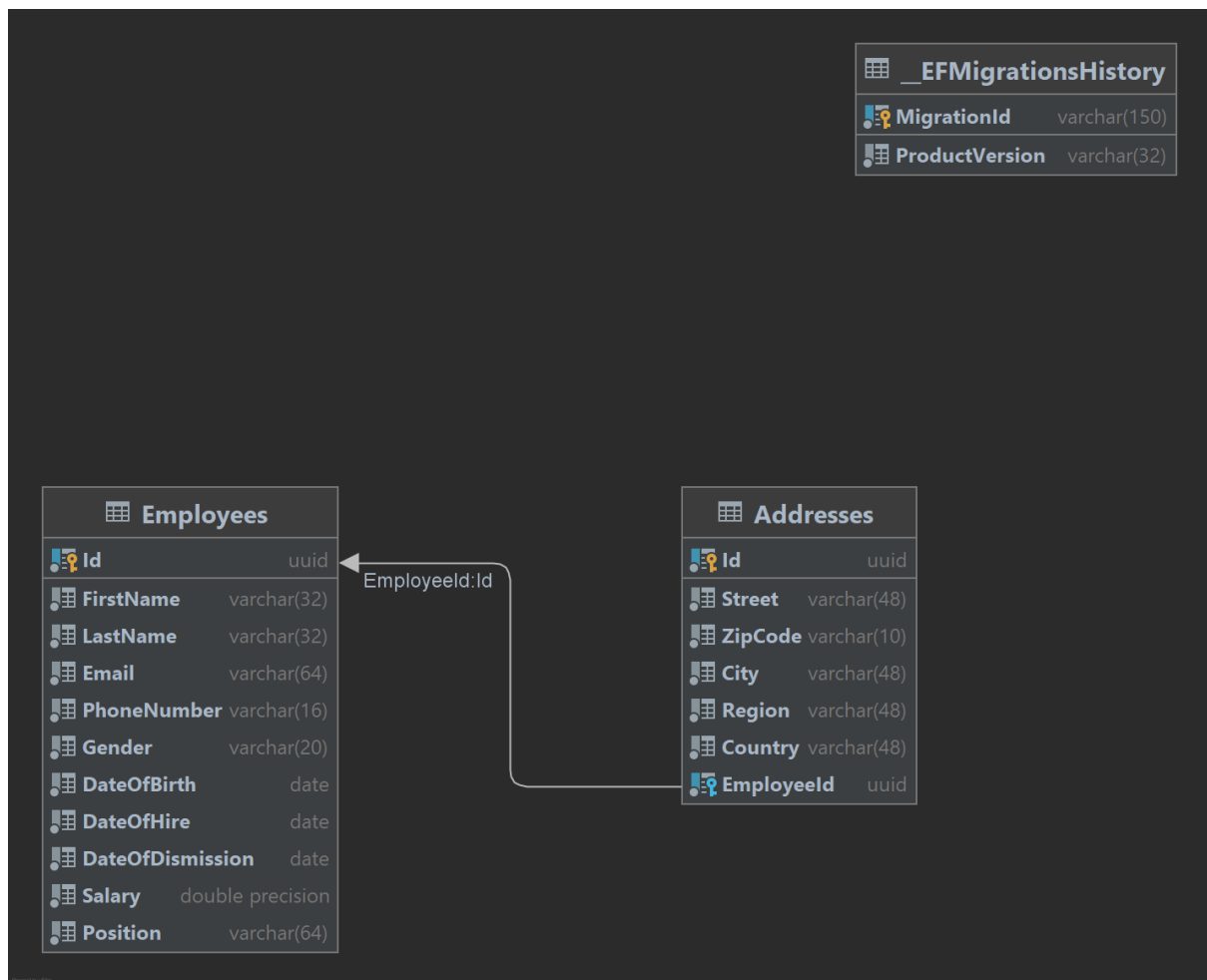
- Przygotowanie dowolnej bazy danych, tabeli do przechowywania pracowników, seed bazy przykładowymi danymi
- Przygotowanie części backendowej aplikacji umożliwiającej obsługę pracowników

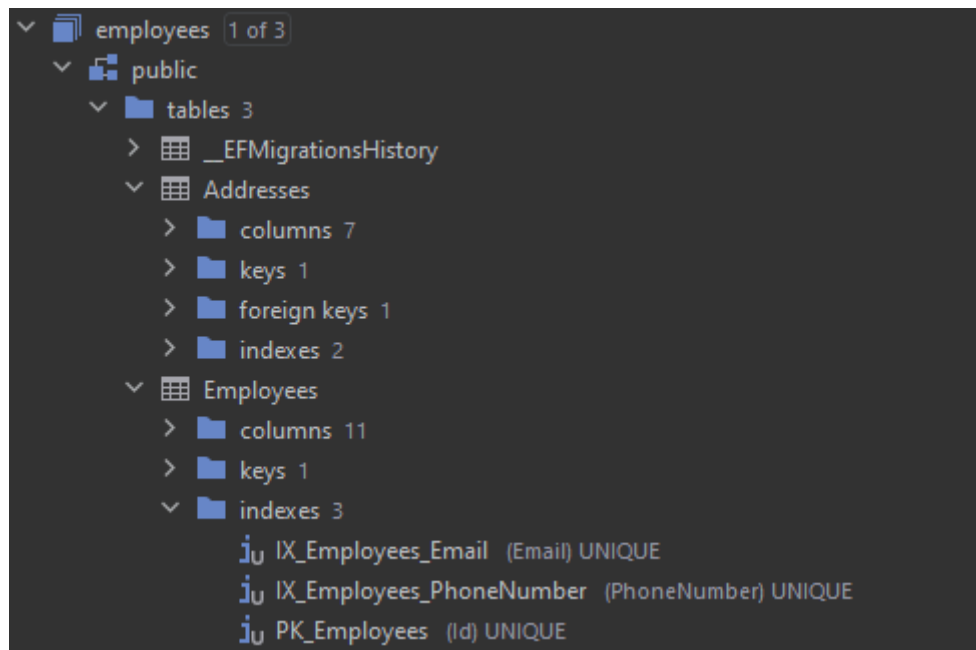
- Dodanie funkcjonalności filtrowania oraz sortowania pracowników po dowolnej kolumnie
- Przygotowanie części frontendowej aplikacji
- Wykonanie przez zespół trudniejszego zadania aplikacji internetowej
- Przygotowanie prezentacji dotyczącej realizacji tego zadania oraz jej zaprezentowanie
- Napisanie krótkiej dokumentacji projektu

d. Przepływ informacji w środowisku systemu  
Scentralizowany oparty na bazie danych.

### 3. Baza danych

#### a. Diagram ERD





#### b. Opis bazy danych

Tabela „Employees” zawiera kolumny FirstName i LastName, odpowiadające imieniu i nazwisku pracownika, e-mail, numer telefonu, płeć, datę urodzenia, zatrudnienie oraz datę zwolnienia pracownika, pensję oraz jego pozycję w firmie. Tabela „Addresses” zawiera kolumny przechowujące informacje o kraju, regionie, mieście, kodzie pocztowym, oraz ulicy adresu zamieszkania pracownika wraz z odwołaniem do tabeli „Employees” za pomocą „uuid”<sup>1</sup> danego pracownika.

### 4. Wykorzystane technologie

#### a. .Net 5 C#

- Entity Framework Core (EF Core) — jest to biblioteka, która umożliwia dostęp do danych. EF Core może być wykorzystane jako ORM (Mapowanie relacyjno-obiektowe), wtedy możliwy jest dostęp do relacyjnej bazy danych w prostszy i krótszy sposób.
- MailKit — biblioteka pozwalająca na obsługę e-maili w C#. Dzięki niej można wysyłać i odczytywać wiadomości e-mail.

b. Blazor — platforma do tworzenia interaktywnego internetowego interfejsu użytkownika po stronie klienta.

c. Bootstrap — Pozwala na szybkie tworzenie responsywnych stron, implementując styl.

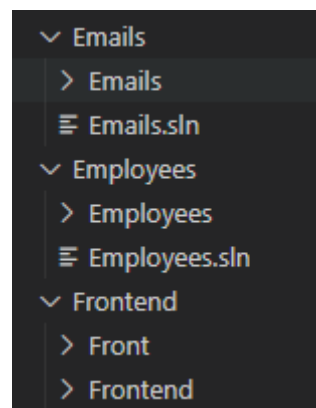
d. PostgreSQL — obiektowo-relacyjna baza danych, która pozwala korzystać z abstrakcji takich jak dziedziczenie tabel i przeciążanie funkcji.

<sup>1</sup> [1] - Uniwersalnie unikalny identyfikator (UUID) to 128-bitowa liczba, która identyfikuje informacje w systemie komputerowym. Jest również znany jako globalnie unikatowy identyfikator (GUID). Standardowa reprezentacja UUID składa się zazwyczaj z cyfr szesnastkowych

## 5. Podział aplikacji

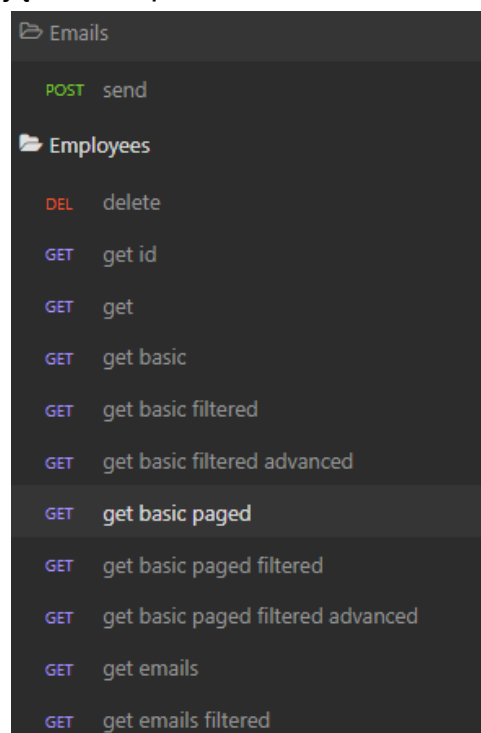
Aplikacja została napisana w języku C#. Projekt składa się z 3 rozwiązań:

- Emails — zawiera projekt, który zarządza funkcjonalnością wysyłania maili
- Employees — zawiera kontrolery, ORM, migracje oraz konfigurację aplikacji
- Frontend — zawiera projekt, którym jest graficzna część aplikacji
- insomnia-api-tests - zawiera eksport projektu z programu Insomnia, który został wykorzystany do testów API



### a. Web API

Wszystkie endpointy zwracają dane w postaci JSON.



```

POST create

PUT put address

PUT put name

PUT put dob

PUT put hire

PUT put dismissal

PUT put phone number

PUT put email

PUT put position

PUT put salary

PUT put gender

```

- Endpoint „send”

Służy do wysyłania maili do wybranych osób. Elementem, który zostanie wysłany to parametr, który ma postać JSON.

- Endpoint „delete”

Służy do usuwania użytkownika z aplikacji. Jako parametr przyjmuje numer id. Numer id jest pobierany od administratora z frontendu.

- Endpoint “get id”

Endpoint zwraca dane pracownika o podanym numerze.

```

1 {
2   "id": "c4285138-b794-44de-b6d7-45427a0b2f14",
3   "firstName": "Gaby",
4   "lastName": "Elderkin",
5   "email": "gelderkin0@ameblo.jp",
6   "phoneNumber": "985-884-3734",
7   "gender": "Transsexual Woman",
8   "dateOfBirth": "1959-07-16T00:00:00",
9   "dateOfHire": "2011-08-13T00:00:00",
10  "dateOfDismission": "0001-01-01T00:00:00",
11  "salary": 1335.0,
12  "position": "Project Manager",
13  "address": {
14    "id": "c5c16c6e-b49a-47a9-9d1e-55363b7a5e32",
15    "street": "7893 Anzinger Road",
16    "zipCode": "",
17    "city": "Nagrog",
18    "region": "",
19    "country": "Indonesia",
20    "employeeId": "c4285138-b794-44de-b6d7-45427a0b2f14"
21  }
22 }

```

- Endpoint “get”

Endpoint zwraca wszystkich pracowników z wszystkimi ich danymi z bazy danych w postaci listy.

```
1  [
2  {
3    "id": "a8fbe334-6cc1-4575-82db-ebd6505fd026",
4    "firstName": "Gaby",
5    "lastName": "Elderkin",
6    "email": "gelderkin0@ameblo.jp",
7    "phoneNumber": "985-884-3734",
8    "gender": "Transsexual Woman",
9    "dateOfBirth": "1959-07-16T00:00:00",
10   "dateOfHire": "2011-08-13T00:00:00",
11   "dateOfDismission": "0001-01-01T00:00:00",
12   "salary": 1335.0,
13   "position": "Project Manager",
14   "address": {
15     "id": "440d0791-8a33-47f0-afa9-ebe487662f63",
16     "street": "7893 Anzinger Road",
17     "zipCode": "amp;",
18     "city": "amp;Nagrog",
19     "region": "amp;Region name",
20     "country": "amp;Indonesia",
21     "employeeId": "a8fbe334-6cc1-4575-82db-ebd6505fd026"
22   }
23 },
24 {
25   "id": "33c3ea52-85fd-4ec8-85df-c3977c434375",
26   "firstName": "Alick",
27   "lastName": "Davidzon",
28   "email": "adavidzon9h@reverbnation.com",
29   "phoneNumber": "306-386-9828",
30   "gender": "Androgyne",
31   "dateOfBirth": "1957-06-23T00:00:00",
32   "dateOfHire": "2012-01-16T00:00:00",
33   "dateOfDismission": "2022-11-14T00:00:00",
34   "salary": 552.0,
35   "position": "Analyst Programmer",
36   "address": {
37     "id": "f8fbb123-0316-4e3f-bdcf-cb684d791783",
38     "street": "92 Canary Circle",
39     "zipCode": "2342",
40     "city": "Ruše",
41     "region": "",
42     "country": "Slovenia",
43     "employeeId": "33c3ea52-85fd-4ec8-85df-c3977c434375"
44   }
45 },
```

- Endpoint "get basic"

Endpoint zwracający podstawowe informacje o wszystkich użytkownikach w bazie danych.

- get basic filtered

Służy do przekazania alfabetycznie poukładanej listy pracowników według imienia. Endpoint przyjmuje dwa parametry: ciąg znaków, po którym sortuje oraz kierunek wyświetlania listy A-Z lub Z-A.

```
1  [
2  {
3    "id": "3791b1fc-750e-456a-b01d-d4d691f73b24",
4    "firstName": "Dar",
5    "lastName": "Keilloh",
6    "email": "dkeillohz@bloglovin.com",
7    "phoneNumber": "354-351-5220",
8    "position": "Paralegal"
9  },
10 {
11   "id": "d11cd587-a9ff-42fb-8d7d-ea8013b6dd6d",
12   "firstName": "Dara",
13   "lastName": "Lonsbrough",
14   "email": "dlonsbroughbq@census.gov",
15   "phoneNumber": "631-718-5703",
16   "position": "Mechanical Systems Engineer"
17 },
18 {
19   "id": "834d66ca-a2bb-45ca-95e3-576b57a81d14",
20   "firstName": "Darb",
21   "lastName": "Wessing",
22   "email": "dwessingey@hhs.gov",
23   "phoneNumber": "698-532-8917",
24   "position": "Senior Cost Accountant"
25 },
26 {
27   "id": "4e4d6582-9525-4c9f-9bf2-ce06b0b9caab",
28   "firstName": "Darbee",
29   "lastName": "McErlaine",
30   "email": "dmcerlainebr@t-online.de",
31   "phoneNumber": "114-563-1071",
32   "position": "Project Manager"
33 },
34 ]
```

- get basic advanced

Służy do zwracania alfabetycznie poukładanej listy pracowników, gdzie funkcja przyjmuje kilka parametrów sortujących : imię, nazwisko, pozycja w firmie, numer telefonu, email.



```
[
  {
    "id": "f728ee3f-075a-422f-bff8-52b7b627356f",
    "firstName": "Frasquito",
    "lastName": "Simacek",
    "email": "fsimacekmb@reuters.com",
    "phoneNumber": "355-123-1775",
    "position": "Environmental Tech"
  },
  {
    "id": "e968a991-89c5-46a1-9ead-0e8d82ff1882",
    "firstName": "Frederigo",
    "lastName": "Ream",
    "email": "fream5c@cocolog-nifty.com",
    "phoneNumber": "192-976-1234",
    "position": "Administrative Officer"
  },
  {
    "id": "5845c563-193e-4301-a71c-f1572a548f94",
    "firstName": "Garrek",
    "lastName": "Mundwell",
    "email": "gmundwell1dl@prnewswire.com",
    "phoneNumber": "144-401-1230",
    "position": "Biostatistician II"
  }
],
```

- Endpoint „get basic paged”

Służy do zwracania informacji, które są wyświetlane w menu głównym aplikacji. Zwracany JSON zawiera podstawowe informacje o użytkownikach oraz link do następnej strony z informacjami.

```
1 {
2   "pageNumber": 1,
3   "pageSize": 25,
4   "firstPage": "https://localhost:5001/api/EmployeeGet/basic-info/paged?
5     pageNumber=1&pageSize=25&orderBy=&orderByDesc=False",
6   "lastPage": "https://localhost:5001/api/EmployeeGet/basic-info/paged?
7     pageNumber=80&pageSize=25&orderBy=&orderByDesc=False",
8   "totalPages": 80,
9   "totalRecords": 2000,
10  "nextPage": "https://localhost:5001/api/EmployeeGet/basic-info/paged?
11    pageNumber=2&pageSize=25&orderBy=&orderByDesc=False",
12  "previousPage": null,
13  "data": [
14    {
15      "id": "d8e2163d-8462-4cf4-ab23-7bcdcf6d5f06",
16      "firstName": "Gaby",
17      "lastName": "Elderkin",
18      "email": "gelderkin0@ameblo.jp",
19      "phoneNumber": "985-884-3734",
20      "position": "Project Manager"
21    },
22    {
23      "id": "8f0562dc-e7c6-452f-ab0f-e011aa3ee069",
24      "firstName": "Alick",
25      "lastName": "Davidzon",
26      "email": "adavidzon9h@reverbNation.com",
27      "phoneNumber": "306-386-9828",
28      "position": "Analyst Programmer"
29    }
30  ]
31 }
```

- Endpoint „get basic paged filtered”

Służy do zwracania informacji, które są wyświetlane w menu głównym aplikacji. Zwracany JSON zawiera podstawowe informacje o użytkownikach posortowanych według alfabetu.

```
1 {
2   "pageNumber": 1,
3   "pageSize": 25,
4   "firstPage": "https://localhost:5001/api/EmployeeGet/basic-info/paged/filter/basic?
   pageNumber=1&pageSize=25&orderBy=2&orderByDesc=False&search=ga",
5   "lastPage": "https://localhost:5001/api/EmployeeGet/basic-info/paged/filter/basic?
   pageNumber=6&pageSize=25&orderBy=2&orderByDesc=False&search=ga",
6   "totalPages": 6,
7   "totalRecords": 137,
8   "nextPage": "https://localhost:5001/api/EmployeeGet/basic-info/paged/filter/basic?
   pageNumber=2&pageSize=25&orderBy=2&orderByDesc=False&search=ga",
9   "previousPage": null,
10  "data": [
11    {
12      "id": "9c6740b4-05a4-4657-a9d0-ec77debd0d77",
13      "firstName": "Gale",
14      "lastName": "Alesbrook",
15      "email": "galesbrookck@sitemeter.com",
16      "phoneNumber": "653-166-2111",
17      "position": "Analyst Programmer"
18    },
19    {
20      "id": "2fb477b0-83bb-4fd6-a638-a0b75123c073",
21      "firstName": "Reinaldos",
22      "lastName": "Beste",
23      "email": "rbestead@mozilla.org",
24      "phoneNumber": "713-314-9387",
25      "position": "Paralegal"
26    },
27    {
28      "id": "f5f4df38-b230-41c9-9184-e97c6b276518",
29      "firstName": "Jobina",
30      "lastName": "Bogaert",
31      "email": "jbogaert2b@mashable.com",
32      "phoneNumber": "449-989-7608",
33      "position": "Data Coordinator"
34    },
35  ]
36 }
```

- Endpoint „get basic paged filtered advanced”

Służy do zwracania informacji, które są wyświetlane w menu głównym aplikacji. Zwracany JSON zawiera podstawowe informacje o użytkownikach posortowanych według imienia, nazwiska, stanowiska w firmie, numeru telefonu, maila.

- Endpoint „get emails”

Służy do zwracania maili wszystkich użytkowników.

```

1  [
2  {
3    "email": "gelderkin@ameblo.jp",
4    "firstName": "Gaby",
5    "lastName": "Elderkin"
6  },
7  {
8    "email": "adavidzon9h@reverbnation.com",
9    "firstName": "Alick",
10   "lastName": "Davidzon"
11  },
12  {
13    "email": "eghidoli9g@behance.net",
14    "firstName": "Ezequiel",
15    "lastName": "Ghidoli"
16  },
17  {
18    "email": "gilyukhov9f@dailymotion.com",
19    "firstName": "Gerhard",
20    "lastName": "Ilyukhov"
21  },

```

- Endpoint „get emails filtered”

Służy do pobrania listy maili z bazy posortowanej według imienia, które jest podawane przez użytkownika.

```

1  [
2  {
3    "email": "akingabyas@scientificamerican.com",
4    "firstName": "Alyda",
5    "lastName": "Kingaby"
6  },
7  {
8    "email": "gridger5g@goo.gl",
9    "firstName": "Gabbie",
10   "lastName": "Ridger"
11  },
12  {
13    "email": "gsaleiz@cdc.gov",
14    "firstName": "Gabi",
15    "lastName": "Sale"
16  },

```

- Endpoint „create”

Służy do tworzenia użytkownika, w celu jego wysłania są wymagane wszystkie dane z pól frontendowych, jest to metoda typu POST, jako element zwracany otrzymujemy utworzonego użytkownika.

```

1 {
2   "id": "ee0fdcdd-9a67-41bb-b23f-e4e57d82fb1d",
3   "firstName": "string",
4   "lastName": "string",
5   "email": "user117531@example.com",
6   "phoneNumber": "123 182-1255",
7   "gender": "string",
8   "dateOfBirth": "2021-10-15T00:00:00",
9   "dateOfHire": "2021-10-15T00:00:00",
10  "dateOfDismission": null,
11  "salary": 1000.0,
12  "position": "string",
13  "address": {
14    "id": "ca0c70a4-bd4f-4608-bf8d-1e319ed6df6d",
15    "street": "string",
16    "zipCode": "string",
17    "city": "string",
18    "region": "string",
19    "country": "string",
20    "employeeId": "ee0fdcdd-9a67-41bb-b23f-e4e57d82fb1d"
21  }
22 }

```

- Endpoint „put address”

Służy do edycji kolumn w bazie: city, region, country, street, zipCode.

- Endpoint „put name”

Służy do edycji kolumn w bazie: firstname, lastname.

- Endpoint „put dob”

Służy do edycji kolumn w bazie: date of birth.

- Endpoint „put hire”

Służy do edycji kolumny w bazie: date of hire.

- Endpoint „put dismissal”

Służy do edycji kolumny w bazie: date of dismissal.

- Endpoint „put phone number”

Służy do edycji kolumny w bazie: phone number.

- Endpoint „put email”

Służy do edycji kolumny w bazie: email.

- Endpoint „put position”

Służy do edycji kolumny w bazie: position.

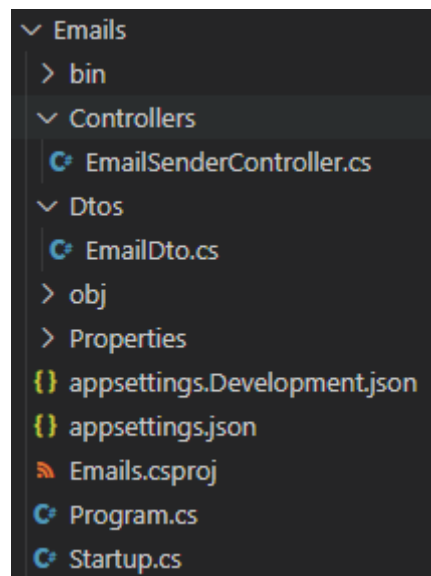
- Endpoint „put salary”

Służy do edycji kolumny w bazie: salary.

- Endpoint „put gender”

Służy do edycji kolumny w bazie: gender.

## a. Emails



Rozwiązanie to składa się z dwóch plików. Pierwszy plik “EmailSenderController.cs” ma w sobie metody służące do złożenia maila z odpowiednich obiektów oraz wysłania go do listy osób.

```
[RequireHttps]
[HttpPost]
0 references
public async Task<IActionResult> SendEmail([ModelBinder(BinderType = typeof(JsonModelBinder))] EmailDto dto)
{
    [loginData]

    try
    {
        if (dto is null)
        {
            throw new ArgumentNullException(nameof(dto));
        }
    }
    catch (ArgumentNullException e)
    {
        return StatusCode(500);
    }

    var recipients = new InternetAddressList();
    foreach (var rcpt in dto.Recipients)
    {
        recipients.Add(MailboxAddress.Parse(rcpt));
    }

    var ccs = new InternetAddressList();
    foreach (var rcpt in dto.Ccs)
    {
        ccs.Add(MailboxAddress.Parse(rcpt));
    }

    var bccs = new InternetAddressList();
    foreach (var rcpt in dto.Bccs)
    {
        bccs.Add(MailboxAddress.Parse(rcpt));
    }
}
```

```

var bodyBuilder = new BodyBuilder();
bodyBuilder.HtmlBody = "<p>" + dto.Body + "</p><br/><p>" + dto.Signature + "</p>";

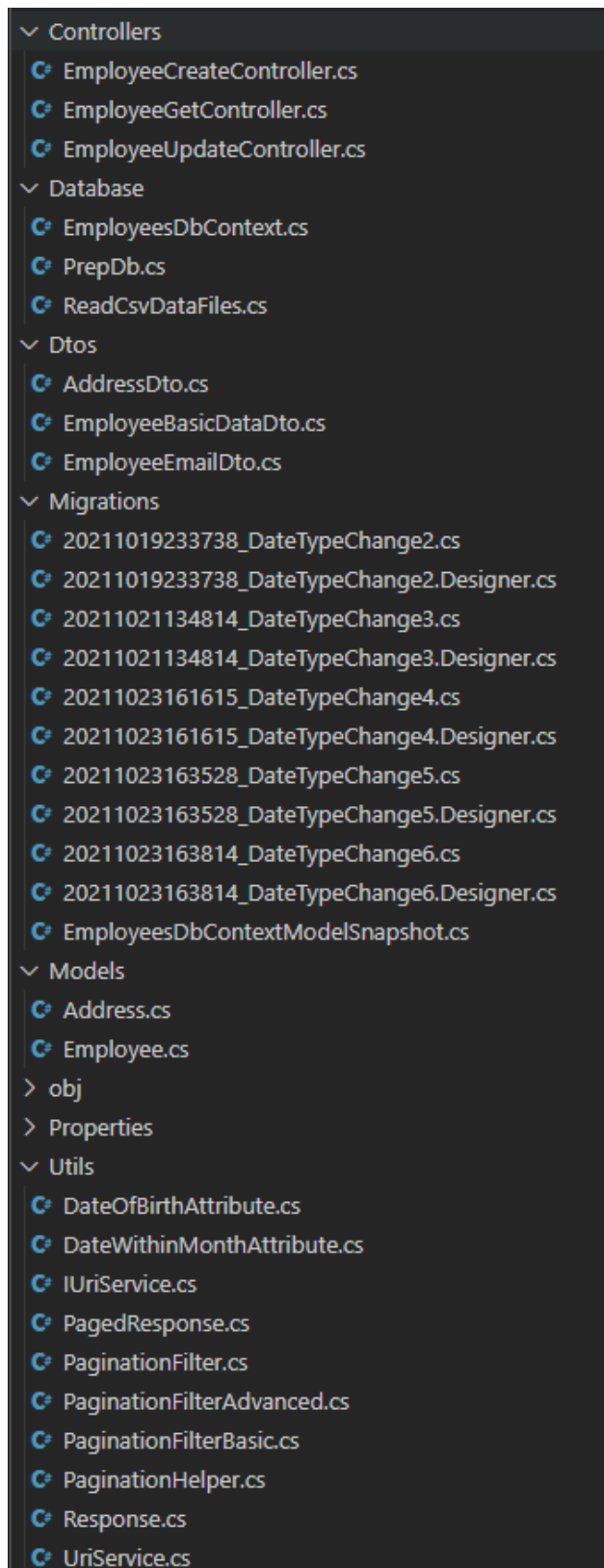
var email = new MimeMessage();
email.From.Add(MailboxAddress.Parse(mailAddress));
email.To.AddRange(recipients);
email.Cc.AddRange(ccs);
email.Bcc.AddRange(bccs);
email.Subject = dto.Subject;
email.Body = bodyBuilder.ToMessageBody();

try
{
    using (var smtp = new SmtpClient())
    {
        await smtp.ConnectAsync("smtp.gmail.com", 587, SecureSocketOptions.StartTls);
        await smtp.AuthenticateAsync(mailAddress, mailPassword);
        await smtp.SendAsync(email);
        await smtp.DisconnectAsync(true);
    }
}
catch (Exception e)
{
    return StatusCode(500);
}

return Ok();

```

## b. Employees



Projekt dzieli się na katalogi:

- Controllers

Zawiera implementacje metod, które są wywoływane przez endpointy. Katalog składa się z trzech klas. Pierwsza o nazwie „EmployeeCreateController”, która zawiera dwie metody. PostEmployee, która odpowiada za utworzenie nowego pracownika oraz DeleteEmployee, która usuwa pracownika o wskazanym + ID. Klasa o nazwie EmployeeGetController przechowuje metody: GetEmployee, służącą do pobierania danych o pracowniku ze wskazanym ID, GetAllEmployeesBasic, pobierającą podstawowe informacje o pracownikach (w obu wersjach, jedna bez paginacji, druga razem z paginacją). Dodatkowo zawarte są również metody odpowiadające za zwrócenie przefiltrowanej listy pracowników.

Klasa EmployeeUpdateController zawiera metody pozwalające na zmianę wszystkich pól dotyczących pracownika.

#### Database

Zawiera dane metody wykonujące operacje związane z bazą danych oraz powiązanych z nią modeli. Składa się z trzech klas. Klasa EmployeesDbContext odpowiada za utworzenie kontekstu. Klasa PrepDb zawiera metody umożliwiające wykonanie migracji na bazie danych oraz wypełnienie tabel danymi. Natomiast klasa ReadCsvDataFiles zawiera metodę umożliwiającą odczyt danych z pliku .csv, które są później ładowane do bazy danych.

- Dtos

Zawiera obiekty transferu danych wykorzystywane w aplikacji. Wszystkie klasy zawierają podstawową konfigurację pól oraz gettery i settery.

- Migration

Zawiera pliki umożliwiające utworzenie tabel wraz z kolumnami i relacjami.

- Models

Zawiera klasy odpowiadające za tworzenie modeli.

- Properties

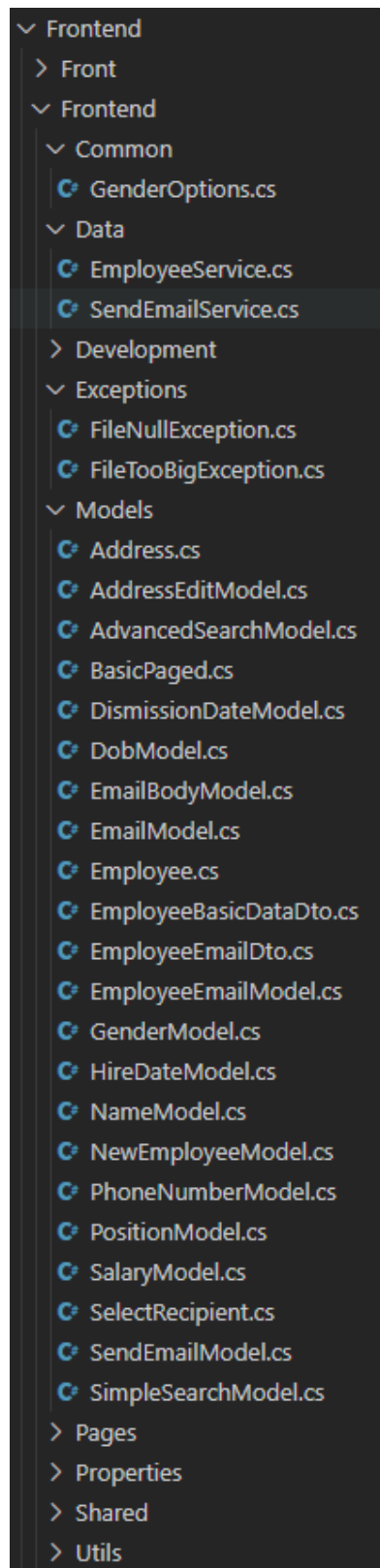
Katalog zawiera plik launchSettings.json, który przechowuje podstawową konfigurację potrzebną do uruchomienia projektu.

- Utils

Zawiera klasy wspierające działanie aplikacji, głównie są to metody ułatwiające walidację danych. Ponadto znajdują się klasy umożliwiające paginację danych oraz kontrolę nad URI.



b. Frontend



Zawiera wszystkie pliki związane z frontendem aplikacji, został napisany przy użyciu Blazora. Szczegółowe informacje na temat wyglądu aplikacji zostaną przedstawione w następnym rozdziale.

c. insomnia-api-tests

Zawiera eksport testów wykonanych w aplikacji Insomnia (<https://insomnia.rest/>). Jest to alternatywa dla Postman'a.

## 6. Interfejs aplikacji / systemu

Screen przedstawia widok strony głównej. Znajdują się na niej odnośniki do edycji danych, dodania osoby do listy wysłanych maili, usuwania użytkownika. Z lewej strony znajduje się sidebar z opcjami dodania, wyszukania i wysyłania maili.

MANAGE EMPLOYEES

+ Add new employee

Search

Send an e-mail

ACTION	SELECT	FIRST NAME	LAST NAME	EMAIL	PHONE NUMBER	POSITION
<div>See details</div> <div>Edit</div> <div>Delete</div> <div>ADD</div>		Alick	Davidzon	adavidzon9h@reverbnation.com	306-386-9828	Analyst Programmer
<div>See details</div> <div>Edit</div> <div>Delete</div> <div>ADD</div>		Ezequiel	Ghidoli	eghidoli9g@behance.net	286-563-1624	Help Desk Technician
<div>See details</div> <div>Edit</div> <div>Delete</div> <div>ADD</div>		Gerhard	Ilyukhov	gilyukhov9f@dailymotion.com	470-467-9278	Desktop Support Technician
<div>See details</div> <div>Edit</div> <div>Delete</div> <div>ADD</div>		Skyler	Scorer	sscorer9e@eventbrite.com	650-558-8724	Automation Specialist IV
<div>See details</div> <div>Edit</div> <div>Delete</div> <div>ADD</div>		Mag	Dutch	mdutch9d@mashable.com	427-501-2608	Chief Design Engineer
<div>See details</div> <div>Edit</div> <div>Delete</div> <div>ADD</div>		Arney	Tippell	atippell9c@rambler.ru	986-637-4545	Systems Administrator IV
<div>See details</div> <div>Edit</div> <div>Delete</div> <div>ADD</div>		Marje	Libbey	mllibbey9b@rediff.com	524-301-4339	Actuary
<div>See details</div> <div>Edit</div> <div>Delete</div> <div>ADD</div>		Stacey	Horbart	shorbart9a@apache.org	233-727-3926	Administrative Assistant II
<div>See details</div> <div>Edit</div> <div>Delete</div> <div>ADD</div>		Alyda	Blanque	ablanque99@noaa.gov	295-222-7393	Junior Executive
<div>See details</div> <div>Edit</div> <div>Delete</div> <div>ADD</div>		Frederico	Idenden	fidenden98@noaa.gov	711-539-6205	Software Consultant
<div>See details</div> <div>Edit</div> <div>Delete</div> <div>ADD</div>		Bettina	Tarborn	btarborn97@umn.edu	634-811-5228	Analyst Programmer
<div>See details</div> <div>Edit</div> <div>Delete</div> <div>ADD</div>		Nilson	Milksop	nmilksop96@elegantthemes.com	958-494-7783	Account Representative IV

Główne okno aplikacji wyświetla podstawowe informacje o użytkowniku oraz każdy wiersz posiada opcję edycji, usunięcia oraz wyświetlenia większej ilości informacji.

See details	Edit	Delete	ADD	Vidovik	Marias	vmarias8y@earthlink.net	475-446-9014	Automation Specialist I
See details	Edit	Delete	ADD	Montague	Willatt	mwillatt8x@mozilla.org	531-598-0861	Design Engineer
See details	Edit	Delete	ADD	Ernest	Gosforth	egosforth8w@hostgator.com	902-575-6371	Assistant Professor
See details	Edit	Delete	ADD	Shelagh	Fouch	sfouch8v@examiner.com	958-519-0599	Design Engineer
See details	Edit	Delete	ADD	Wakefield	Hartnell	whartnell8u@devhub.com	464-825-6314	Computer Systems Analyst I
See details	Edit	Delete	ADD	Clo	Goggins	cgoggins8t@zimbio.com	823-226-6314	Structural Analysis Engineer
<div>PREVIOUS</div> <div>1</div> <div>2</div> <div>80</div> <div>NEXT</div>								

Po wciśnięciu przycisku “Add” kliknięty wiersz zostanie dodany do listy przeznaczonej do wysłania maili. Gdy dany wiersz znajduje się w liście przycisk zamieniany jest na “Remove” i pozwala na usunięcie użytkownika z listy.

Press to send email			
<div>adavidzon9h@reverbnation.com X</div> <div>eghidoli9g@behance.net X</div> <div>gilyukhov9f@dailyemotion.com X</div> <div>sscorer9e@eventbrite.com X</div>			
ACTION	SELECT	FIRST NAME	LAST NAME
See details Edit Delete	REMOVE	Alick	Davidzon
See details Edit Delete	REMOVE	Ezequiel	Ghidoli
See details Edit Delete	REMOVE	Gerhard	Ilyukhov
See details Edit Delete	REMOVE	Skyler	Scorer
See details Edit Delete	ADD	Mag	Dutch
See details Edit Delete	ADD	Arney	Tippell

Po naciśnięciu przycisku “See details” otrzymujemy dokładne informacje o pracowniku — imię, nazwisko, e-mail, numer telefonu, płeć, datę urodzenia, zatrudnienia i zwolnienia, pensję, pozycję oraz dokładny adres zamieszkania.

MANAGE EMPLOYEES

+ Add new employee

↺ Search

📧 Send an e-mail

DETAILS

FIRST NAME

Alick

LAST NAME

Davidzon

EMAIL

adavidzon9h@reverbnation.com

PHONE NUMBER

306-386-9828

GENDER

Androgyne

DATE OF BIRTH

23.06.1957

DATE OF HIRE

16.01.2012

DATE OF DISMISSION

14.11.2022

SALARY

\$ 552

POSITION

Analyst Programmer

ADDRESS

92 Canary Circle

2342 Ruše

Slovenia

EDIT DATA

REMOVE EMPLOYEE

Formularz dodania nowego pracownika, posiada on wymaganą walidację dla każdego pola. Gdy wprowadzony tekst nie spełni warunków, wyświetlany jest stosowny komunikat odnośnie nieprawidłowości.

**ADD NEW EMPLOYEE**

FIRST NAME	STREET ADDRESS
<input type="text"/>	<input type="text"/>
LAST NAME	ZIP CODE
<input type="text"/>	<input type="text"/>
PHONE NUMBER	CITY NAME
<input type="text"/>	<input type="text"/>
EMAIL ADDRESS	REGION NAME
<input type="text"/>	<input type="text"/>
DATE OF BIRTH	COUNTRY NAME
<input type="text" value="01.01.0001"/>	<input type="text"/>
GENDER	
<input type="text" value="Agender"/>	
POSITION AT THE OFFICE	HIRING DATA
<input type="text"/>	<input type="text" value="01.01.0001"/>
SALARY	
<input type="text" value="0"/>	

CREATE NEW EMPLOYEE

Przy edycji danych pracownika możemy zmienić wszystkie dane, które są wyświetlane w dokładnych informacjach o pracowniku. Dane znajdujące się w polach są pobierane z bazy danych.

### EDIT EMPLOYEE DETAILS

FIRST NAME	STREET ADDRESS
<input type="text" value="Alick"/>	<input type="text" value="92 Canary Circle"/>
LAST NAME	ZIP CODE
<input type="text" value="Davidzon"/>	<input type="text" value="2342"/>
<input type="button" value="CHANGE NAME"/>	CITY NAME
PHONE NUMBER	<input type="text" value="Ruše"/>
<input type="text" value="306-386-9828"/>	REGION NAME
<input type="button" value="CHANGE PHONE NUMBER"/>	<input type="text"/>
EMAIL ADDRESS	COUNTRY NAME
<input type="text" value="adavidzon9h@reverbnation.com"/>	<input type="text" value="Slovenia"/>
<input type="button" value="CHANGE EMAIL ADDRESS"/>	<input type="button" value="CHANGE ADDRESS"/>
DATE OF BIRTH	
<input type="text" value="23.06.1957"/>	
<input type="button" value="CHANGE DATE OF BIRTH"/>	
GENDER	
<input type="text" value="Androgyne"/>	
<input type="button" value="CHANGE GENDER"/>	
POSITION AT THE OFFICE	HIRING DATA
<input type="text" value="Analyst Programmer"/>	<input type="text" value="16.01.2012"/>
<input type="button" value="CHANGE POSITION"/>	<input type="button" value="CHANGE HIRING DATE"/>
SALARY	DISMISSION DATA
<input type="text" value="552"/>	<input type="text" value="14.11.2022"/>
<input type="button" value="CHANGE SALARY"/>	<input type="button" value="CHANGE DISMISSION DATE"/>

Przy szukaniu pracowników, mamy do wyboru dwie opcje, szukanie podstawowe i zaawansowane. Szukanie podstawowe pozwala na wpisanie tekstu, którego chcemy znaleźć np. imienia, wybór kolumny, po której sortujemy, czy sposób sortowania ma być malejący oraz, czy wyniki mają zostać paginowane.

### SEARCH FOR EMPLOYEES

SELECT IF YOU WANT ADVANCED SEARCH ☐ INPUT TEXT THAT YOU WANT TO SEARCH FOR

ORDER BY COLUMN

▼

SHOULD THE ORDER BE DESCENDING ☐

SHOULD THE RESULT BE PAGED ☐

SEARCH

W wyszukiwaniu zaawansowanym pracowników możemy wyszukać po wielu informacjach naraz lub tych przez nas wybranych — imieniu i nazwisku, pozycji w firmie, numeru telefonu, adresie e-mail, wybranej kolumnie, za pomocą której będziemy sortować wyniki, sposób sortowania wyniku — malejący czy rosnący oraz, czy wyniki mają być paginowane.

### SEARCH FOR EMPLOYEES

SELECT IF YOU WANT ADVANCED SEARCH ☒ SEARCH BY FIRST NAME

SEARCH BY LAST NAME

SEARCH BY POSITION

SEARCH BY PHONE NUMBER

SEARCH BY EMAIL

ORDER BY COLUMN

▼

SHOULD THE ORDER BE DESCENDING ☐

SHOULD THE RESULT BE PAGED ☐

SEARCH

Zakładka “Send an email” na samej górze posiada zaznaczoną wcześniej listę wybranych osób. Kolejne pola służą do dostosowania informacji jakie ma zawierać email.

**SEND AN EMAIL**

eghidoli9g@behance.net X

ablanque99@noaa.gov X

TO

Alyda Blanque - ablanque99@noaa.gov

**ADD RECIPIANT**

shorbart9a@apache.org X

CC

Stacey Horbart - shorbart9a@apache.org

**ADD CC**

fidenden98@noaa.gov X

BCC

Frederico Idenden - fidenden98@noaa.gov

**ADD BCC**

SUBJECT

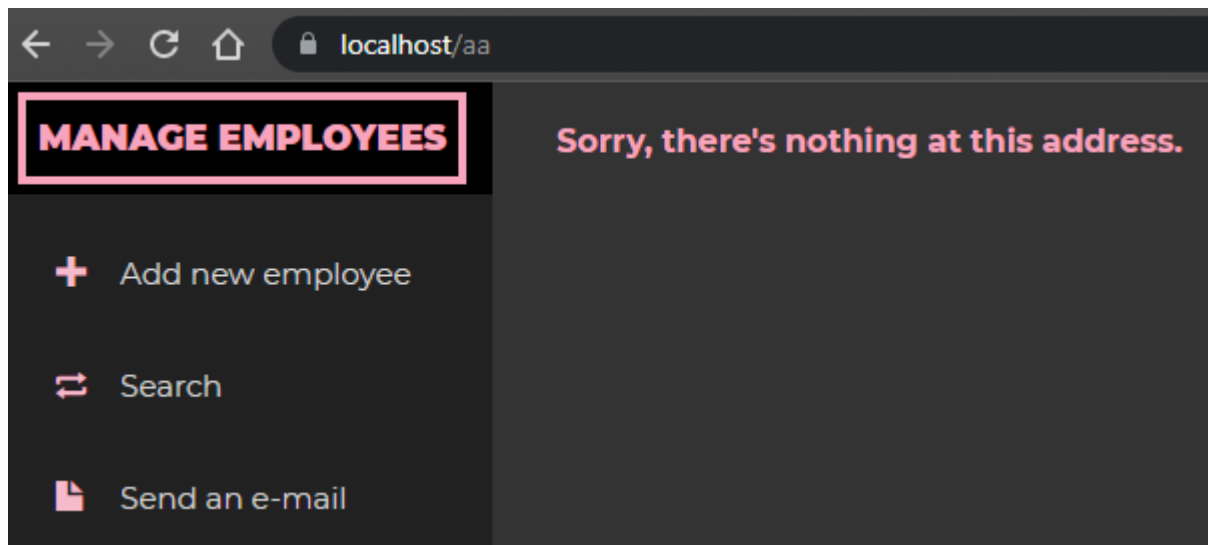
MESSAGE

SIGNATURE

**SEND EMAIL**



W przypadku gdy dana strona jest niedostępna lub backend nie zwrócił żadnych informacji, zostanie wyświetlony komunikat.



Jeżeli nie nastąpi połączenie z serwerem, zostanie wylosowany jeden z dwóch motywów informujący nas o braku połączenia z serwerem.



**NO SERVER IN  
YOUR AREA**



**NO SERVER IN  
YOUR AREA**