

SalaryApp

Описание на проекта:

SalaryApp е уеб приложение, изградено с ASP.NET Core MVC. Предназначено е за управление на записи на служители и на заплати. Предоставя функционалности за преглед, добавяне, редактиране и изтриване на информация за служители, заплати и др. В приложението също така е добавена и много базова автентикация за достъп на потребител

Разработени Функционалности:

Управление на служители:

Създаване, преглеждане, редактиране и управление на записи на служители, включително лични данни, компания, длъжност, офис и адресна информация. Пример:

Create Employee

First Name	Middle Name	Last Name
<input type="text"/>	<input type="text"/>	<input type="text"/>
Company	Position	Office
-- Select Company --	-- Select Position --	-- Select Office --
Hire Date		
06 / 10 / 2025		
Address Information		
Street Name	Street Number	
<input type="text"/>	<input type="text"/>	
Neighborhood	Postal Code	
<input type="text"/>	<input type="text"/>	
City	Address Type	
-- Select City --	-- Select Type --	
<input type="button" value="Create"/> <input type="button" value="Back to List"/>		

Управление на Заплати:

Проследяване и управление на заплати на служители, включително бонуси и удържки, за конкретни периоди.

Базова Автентикация:

Проста функционалност за вход с потребител по подразбиране "admin".

Ръководство на потребителя

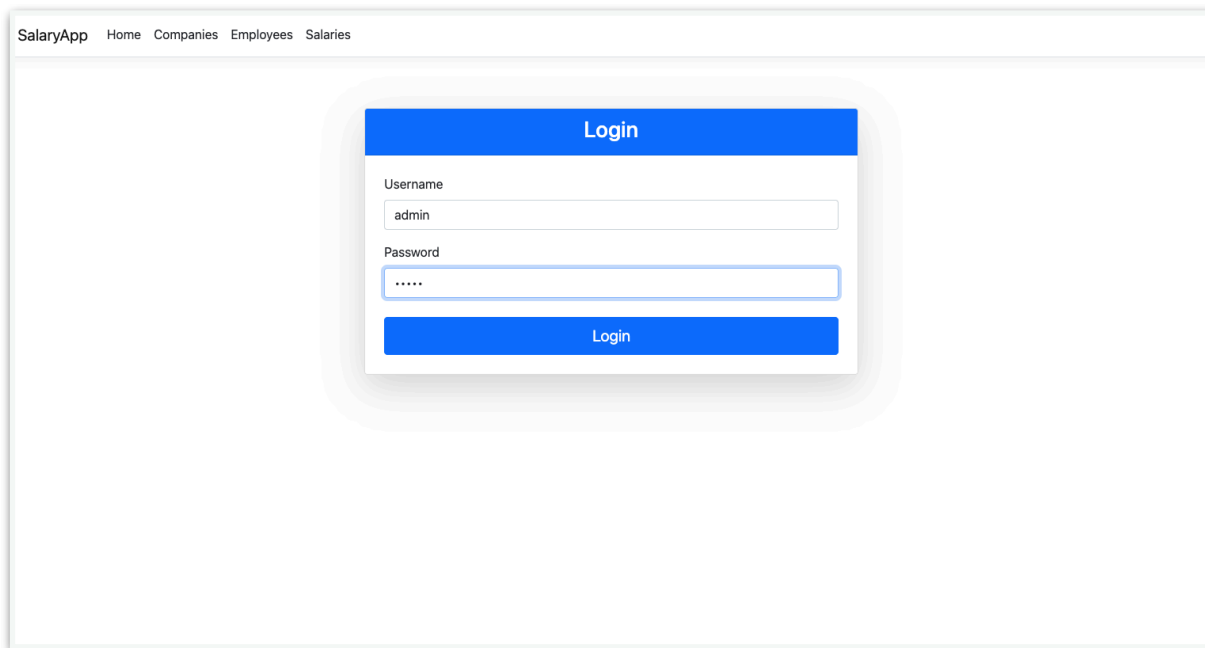
Вход в Системата

При стартиране на приложението ще бъдете пренасочени към страницата за вход (<https://localhost:7112/>) . Тя изглежда по следния начин:

Използвайте следните данни за достъп:

Потребителско име: admin

Парола: admin



SalaryApp Home Companies Employees Salaries

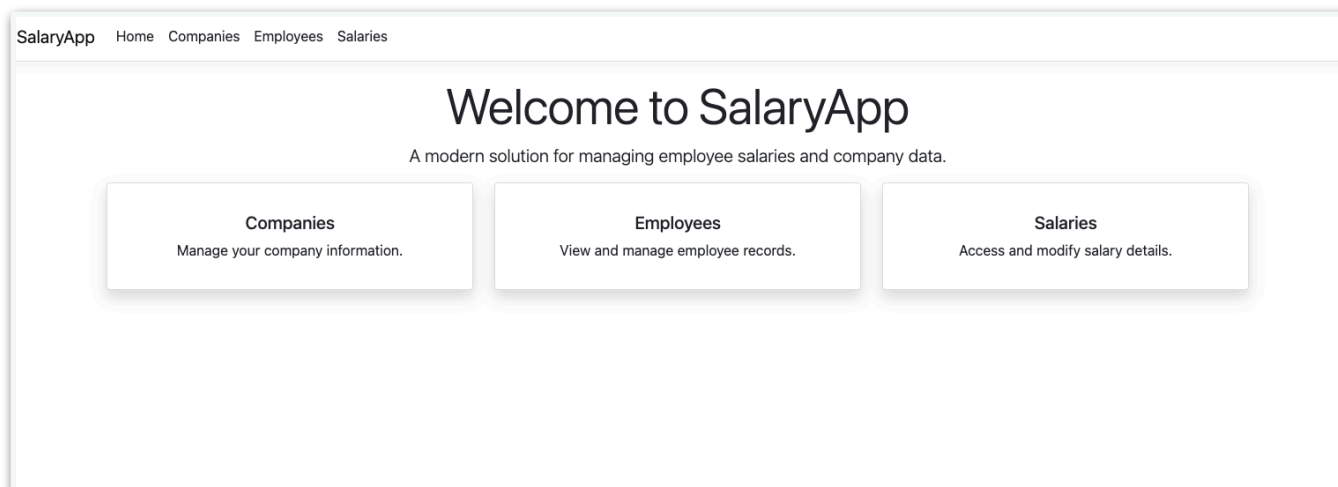
Login

Username
admin

Password
.....

Login

След успешен вход, ще видите началната страница с три основни секции: Компании, Служители и Заплати. Можете да навигирате между тях чрез кликане върху съответните контейнери или чрез навигационната лента в горната част на екрана.



SalaryApp Home Companies Employees Salaries

Welcome to SalaryApp

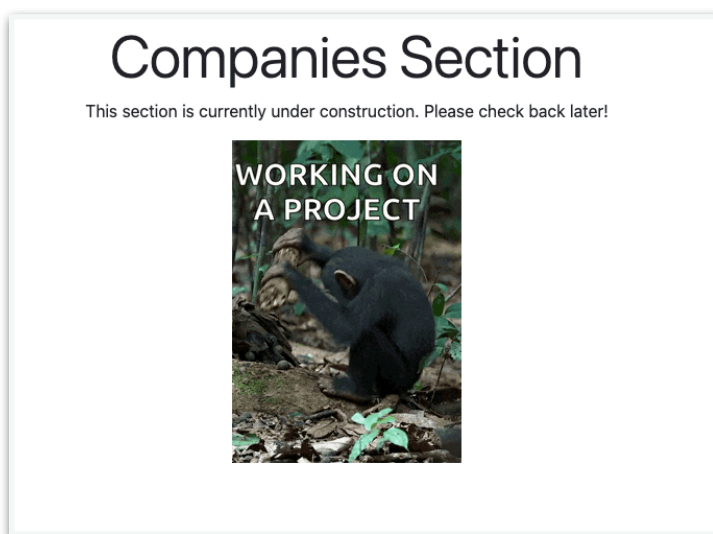
A modern solution for managing employee salaries and company data.

Companies
Manage your company information.

Employees
View and manage employee records.

Salaries
Access and modify salary details.

* **Компании:** Пренасочва към страница която е в разработка".



* **Служители:** Пренасочва към страницата за управление на служители.

Employees							
Create New							
FirstName	MiddleName	LastName	Company	Position	Office	HireDate	Actions
John		Doe	Tech Solutions Ltd.	Senior Developer	Sofia HQ	10.06.2023	Details Edit Delete
Jane		Smith	Tech Solutions Ltd.	HR Manager	Sofia HQ	10.06.2024	Details Edit Delete
Bob		Johnson	Digital Innovations Inc.	Junior Developer	Plovdiv Branch	10.12.2024	Details Edit Delete
Ivan		Todorov	Digital Innovations Inc.	Junior Developer	Plovdiv Branch	10.06.2025	Details Edit Delete
Penka		Penchova	Tech Solutions Ltd.	HR Manager	Sofia HQ	10.06.2025	Details Edit Delete

* **Заплати:** Пренасочва към страницата за управление на заплати.

Salary Records							
Employee	Position	Period	Base Salary	Bonus	Deduction	Total	Actions
John Doe	Senior Developer	1.05.2025 - 31.05.2025	5000,00 BGN	300,00 BGN	100,00 BGN	5200,00 BGN	Edit Delete
Ivan Todorov	Junior Developer	1.05.2025 - 31.05.2025	3000,00 BGN	300,00 BGN	100,00 BGN	3200,00 BGN	Edit Delete
Ivan Todorov	Junior Developer	1.03.2025 - 31.03.2025	3000,00 BGN	600,00 BGN	100,00 BGN	3500,00 BGN	Edit Delete
Penka Penchova	HR Manager	1.05.2025 - 31.05.2025	4000,00 BGN	500,00 BGN	200,00 BGN	4300,00 BGN	Edit Delete

База Данни:

Базата данни на проекта е проектирана с помощта на Entity Framework Core, с Code First подхода, което означава, че структурата на таблиците и техните релации пряко отразяват дефинициите на C# класовете в директорията `SalaryApp/Models`. Всеки клас представлява таблица, а неговите свойства (properties) са колоните на тази таблица, с допълнителни анотации, които дефинират изисквания за валидация, типове данни и релации.

По-долу е представено описание на всяка таблица, нейните атрибути (колони) и релации с други таблици.

1. Cities

Поле	Тип	Описание
Id	INTEGER (PK)	Уникален идентификатор
Name	NVARCHAR (100)	Име на града, задължително

Връзка: One-to-Many с Address- Един град може да има много Адреси

2. Addresses

Поле	Тип	Описание
Id	INTEGER (PK)	Уникален идентификатор
StreetName	NNVARCHAR (100)	Име на улицата
StreetNumber	NVARCHAR (15)	Номер на сграда
Neighborhood	NVARCHAR (30) NULL	Квартал (по избор)
PostalCode	NVARCHAR (20) NULL	Пощенски код
CityId	INTEGER (FK)	FK към таблицата Cities
AddressType	NVARCHAR (20)	Enum: Employee или Office

Връзки:

- Many-to-One с City- Много адреси в един Град
- One-to-Many с Office, Employee - На един Адрес може да има много Офиси и Работници

3. Companies

Поле	Тип	Описание
Id	INTEGER (PK)	Уникален идентификатор
Name	NVARCHAR (150)	Име на компанията (уникално)

Връзки:

- One-to-Many с Department, Office, Employees- Една компания може да има много Департаменти, Офиси и Работници

4. Departments

Поле	Тип	Описание
Id	INTEGER (PK)	Уникален идентификатор
Name	NVARCHAR (100)	Име на департамента
CompanyId	INTEGER (FK)	FK към таблицата Companies

Връзки:

- One-to-Many с Position - един Департамент може да има много Позиции

5. Positions

Поле	Тип	Описание
Id	INTEGER (PK)	Уникален идентификатор
Title	NVARCHAR (50)	Име на позицията
BaseSalary	DECIMAL (18 , 2)	Основна заплата за позицията
DepartmentId	INTEGER (FK)	FK към таблицата Departments

Връзки:

- Many-to-One с Department - Много Позиции могат да съществуват в един Департамент
- One-to-Many с Employees - Една Позиция може да принадлежи на много Работници
- BaseSalary се използва в Salary при изчисления

6. Offices

Поле	Тип	Описание
Id	INTEGER (PK)	Уникален идентификатор
Name	NVARCHAR (100)	Име на офиса
CompanyId	INTEGER (FK)	FK към таблицата Companies
AddressId	INTEGER (FK)	FK към таблицата Address тип Office

Връзки:

- One-to-Many с Employee - Един офис може да бъде свързан с много Служители
- Many-to-One с Address - Много офиси може да има на един Адрес
- Many-to-One с Company - Много офиси могат да принадлежат на една Компания

7. Employees

Поле	Тип	Описание
Id	INTEGER (PK)	Уникален идентификатор
FirstName	NVARCHAR (50)	Собствено име
MiddleName	NVARCHAR (50) NULL	Презиме (по избор)
LastName	NVARCHAR (50)	Фамилия
AddressId	INTEGER (FK)	FK към Address.Id (тип: Employee)
HireDate	DATE	Дата на наемане
TerminationDate	DATE NULL	Дата на освобождаване (NULL ако активен)
IsTerminated	BOOL	Указва дали е активен служител. Default false
PositionId	INTEGER (FK)	FK към таблицата Positions
OfficeId	INTEGER (FK)	FK към таблицата Offices тип Employee
CompanyId	INTEGER (FK)	FK към таблицата Companies

Връзки:

- One-to-Many с Salaries - Един Работник може да има много Заплати
- Many-to-One с Addresses, Positions, Offices, Companies - Много Работници може да има един Адрес (тип Employee), Позиция, Офис и Компания

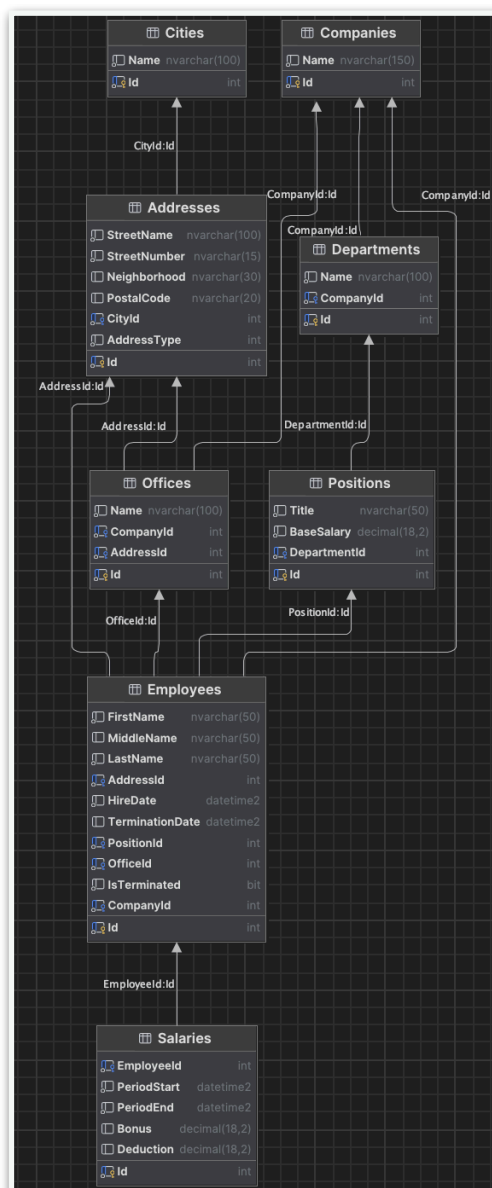
8. Salaries

Поле	Тип	Описание
Id	INTEGER (PK)	Уникален идентификатор
EmployeeId	INTEGER (FK)	FK към Employee.Id
PeriodStart	DATE	Начална дата на периода
PeriodEnd	DATE	Крайна дата на периода
Bonus	DECIMAL(18, 2)	Бонус за периода (по подразбиране 0)
Deduction	DECIMAL(18, 2)	Удръжки за периода (по подразбиране 0)

Връзки:

- Many-to-One с Employee - Много заплати принадлежат на един Работник

Диаграма на базата:



Технически Изисквания

* **.NET SDK:** Версия 6.0

* **SQL Server :** Или друга съвместима система за бази данни, ако промените връзката към базата данни.

* **IDE (По избор, но препоръчително):** Visual Studio, Visual Studio Code или JetBrains Rider. (Програмата е създадена в JetBrains Rider)

Описание на Външни Компоненти

Проектът SalaryApp използва няколко външни компонента (NuGet пакети), които са основни за неговата функционалност:

Microsoft.EntityFrameworkCore.SqlServer 6.0.36

Microsoft.EntityFrameworkCore.Tools 6.0.36

Бъдещи реализации към проекта

1. При създаване на работник, когато избереш компания да се филтрират само данните свързани с тази компания.
2. Търсачка на работници спрямо техните пропъртита.
3. Да може да се запази заплата към PDF по-добре.
4. Цялостен design overhaul - твърде е стерилно + dark theme.
5. За всеки един работник да се генерира парола, с която да може той да влезе и да си види издадени заплати.
6. Към даването на заплата да се оставя коментар- кое за какво е.
7. Подробно създаване на компания
8. Графики за дадени заплати на определен работник/департамент/компания за определен диапазон от време.
9. Оптимизация на код
10. Възможност да назначиш обратно уволнен работник (така или иначе не ги трия от базата)