Setup Guide

1. Follow the link https://github.com/TeodorVasilev/TaskManagementApp.git and download the project
2. Open the solution with **VisualStudio** and locate the **appsettings.json** file in **TaskManagmentApp** project
3. Locate the **ConnectionStrings** section and change the value of the **DefaultConnection** to the connection string you want to use. Database used: **MSSQL Server**.
4. Save the **appsettings.json** file.
5. Set the **TaskManagmentApp.DAL** as **StartUp** project
6. Open the **Package Manager Console** and set your **default project** to **TaskManagmentApp.DAL**
7. Enter **Update-Database** command
8. Set the **TaskManagmentApp** as **StartUp** project and run it.

User Guide

1. Adding employee:
   1.1. Login with email **admin@admin.com** and password **Test123!**
   1.2. Select **Users** from the menu
   1.3. Click **Add new user** and fill in the user details and select **role: Employee**
   1.4. After a new user is created **be sure to write down the password that you will receive as it cannot be recovered if lost.**
2. Creating assignment:
   2.1. Select **Assignments** form the menu
   2.2. Click **Add new assignment** and fill in the assignment details and assign it to a specific user.
3. Log in as employee and you can view your tasks and mark them as completed when finished

Functionality

- ☑ TaskManagementApp allows users to perform CRUD operations on users and assignments based on role.
- ☑ If user is logged in as admin, they can perform CRUD operations on both users and assignments, can assign assignments to specific employee, can change the user role.
- ☑ If user is logged in as employee, they can only view their tasks and mark them as completed when finished. They cannot edit or update user information or assignments.
- ☑ Displays statistics about the 5 best employees in the last month, the number of employees,the number of tasks, the number of tasks in progress.

Architecture

1. **TaskManagementApp** - Presentation layer that is responsible for handling user interaction and displaying data. The Controllers folder contains the controllers that handle Http requests. The Views folder contains the Razor views that display data.
2. **TaskManagementApp.DAL -** Data access layer. The Data folder contains the AppDbContext class which is the connection to the database. The Models folder contains the classes that represent the tables in the database. The Configuration folder contains configuration classes for EntityFramework. The ViewModels folder contains the models that are used to pass the data between layers.
3. **TaskManagementApp.Service -** The service layer is responsible for passing data between the presentation layer and the data layer. It includes service classes that are responsible for implementing the logic and ensuring the app is working correctly.