Suggestion Board

Project Documentation

Table of Contents

1 Abstract

This document describes the purpose, user interface, project structure, code blocks, and using the application.

2 Keywords

C#, .Net Framework, Dependency Injection, ASP.NET Core, MVC, Entity Framework, ORM

3 Introduction

The project developed for the Web Programming lesson to demonstrate the ASPNET, HTML, CSS, JavaScript, and an ORM framework, how they work together as a monolith application.

3.1 The Purpose of the Project

Collect suggestions from the users on different topics and categories, then make them to visible to the other users for voting and commenting. Finally, to be able to list the most liked and reacted suggestions to the authorities. Thus authorities can able to take action to the most important suggestions.

3.2 Project Structure and Used Technologies

The project structure is N-Layer Monolith Application. N-Layer means the solution contains more than one project, and Monolith means there is only one web service in our situation. We are not using services like Authentication Web API and Suggestion-Board Web API separately. We have only one web service which contains all of the things.

C# used as programing language with ASPNET Core framework, which is to develop web services.

Entity-Framework Core is used to manage all database operations as ORM. Additionally, EF Core configured for MSSQL-Server.

Dotnet built-in Dependency Injection is used to solve object creations and injection into constructors.

Bootstrap CSS and JavaScript library is used to create web page designs.

Elasticsearch and Kibana are used to store application logs and visualize the logs.

4 Versions

Name	Features	Release Date
V0.5 (Pre-release)	• Suggestion Listing • Suggestion Adding • User Registration • User Login • Reaction to a Suggestion • Commenting to a Suggestion • Contact Form	16/11/2020
V1.0	• Listing All Users • Viewing User's Profiles • Editing Current User Data • Listing, Adding, Editing, Deleting Categories • Sort and Filtering options for Suggestion List • Editing and Deleting for Suggestion • Error (404) page	19/11/2020
V1.1	• Multi Language Support • Using Fluent-API on Domain classes instead of Data Annotations • Unit Testing • Better Image Visualization • Fix for Contact Form • About page • Fix for the deleting records when it's have connection via foreign keys	In Development

5 User Interface

User Interface is built by CSS, JS, HTML, and Razor codes. I used the Bootstrap library while I am creating the UI. Bootstrap is the most popular HTML, CSS, and JS library.

Each page occurs from a Navbar on the top, a Footer, and the body of the page. Footer and Navbar are the same for every page, but the body depends on the current page.

Users can navigate between the pages via Navbar and some redirection buttons which are located on a few pages.

5.1 Screen List

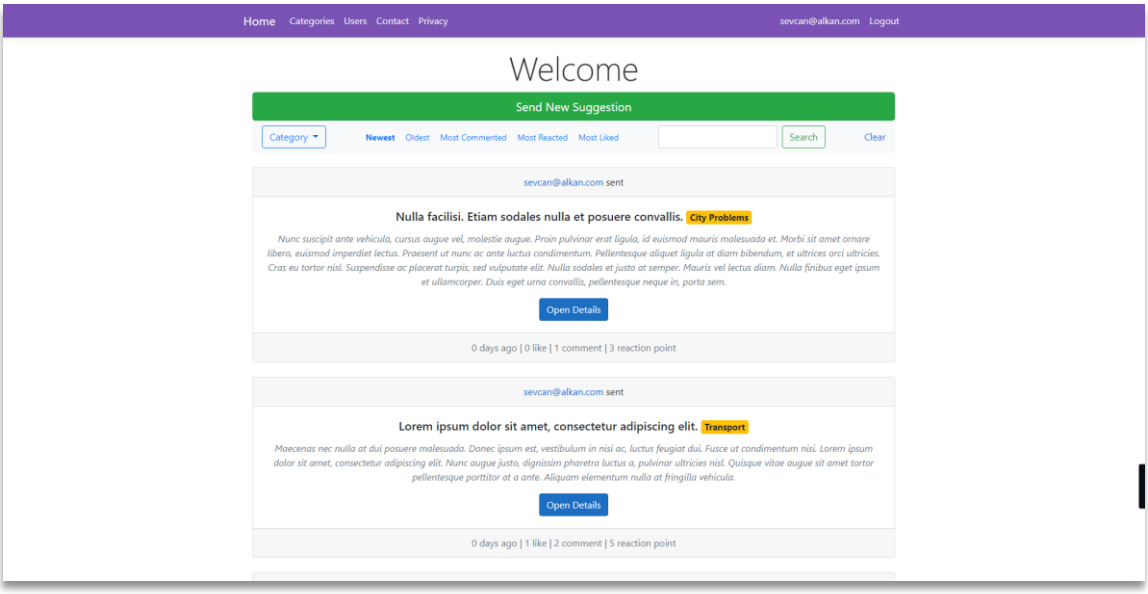
- Home
- Contact
- Privacy
- Category List
- Category Detail
- Suggestion Detail
- User List
- User Profile
- User Edit
- Login
- Register
- Error (404)

5.2 Home Page

Controller	**HomeController**
URL Path	/ **OR** /Home/Index

The Home page is containing a suggestion list, also filtering and sorting options for the suggestion list. Users can see suggestions, filter, and sort them via the Home page. Also, users can be redirected to the Suggestion-Detail page via the “Send New Suggestion” button if the user wants to send a new suggestion.

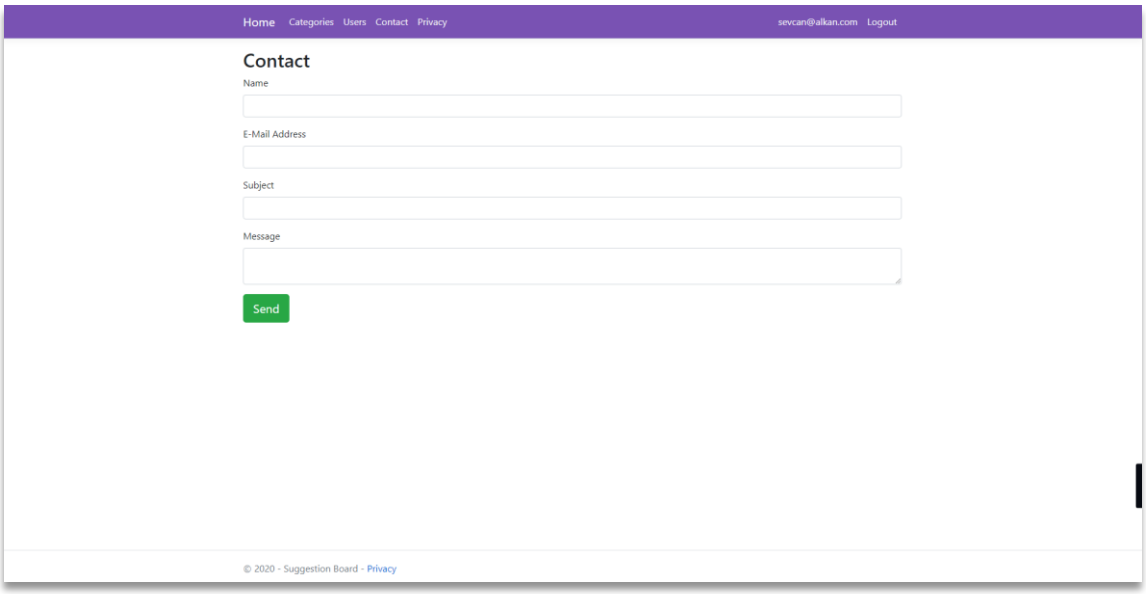
An addition, HomeController uses an ISuggestionService instance to get Suggestions for the list. SuggestionController not using in this situation.



5.3 Contact Page

Controller	ContactController
URL Path	/Contact/Index

The Contact page is designed for the user to be able to send feedback to the developer of the website. The page contains a form that has the message sender's information and the message fields.



5.4 Privacy Page

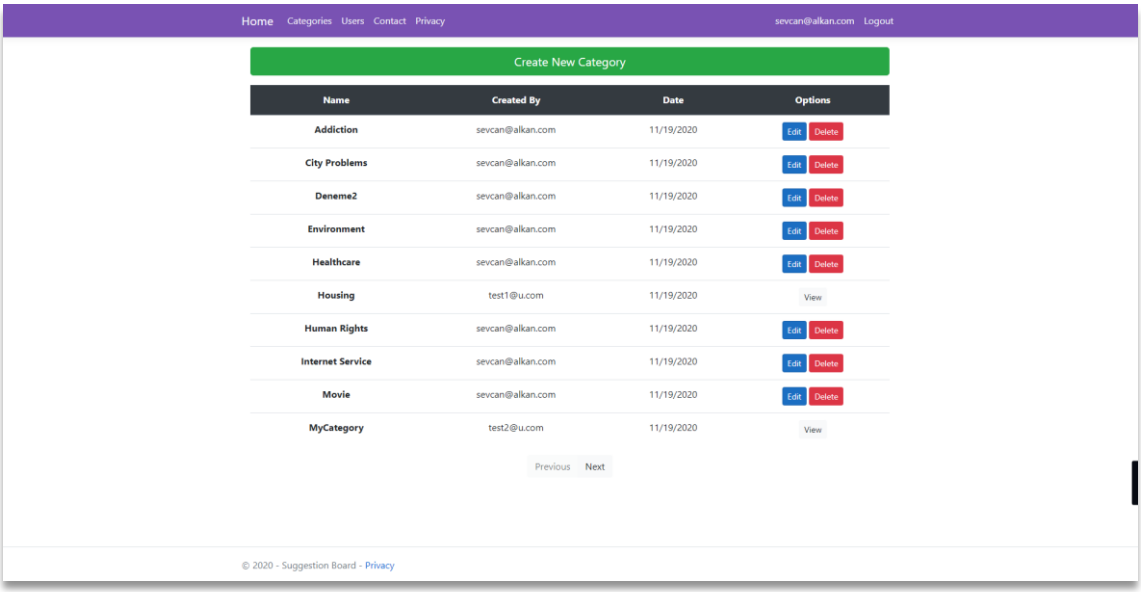
Controller	HomeController
URL Path	/Home/Privacy

The page idea to present the site's privacy policy to the users. But this page does not contain a privacy policy right now because the project is not planned to publish for the end-users.

5.5 Category List Page

Controller	CategoryController
URL Path	/Category/Index

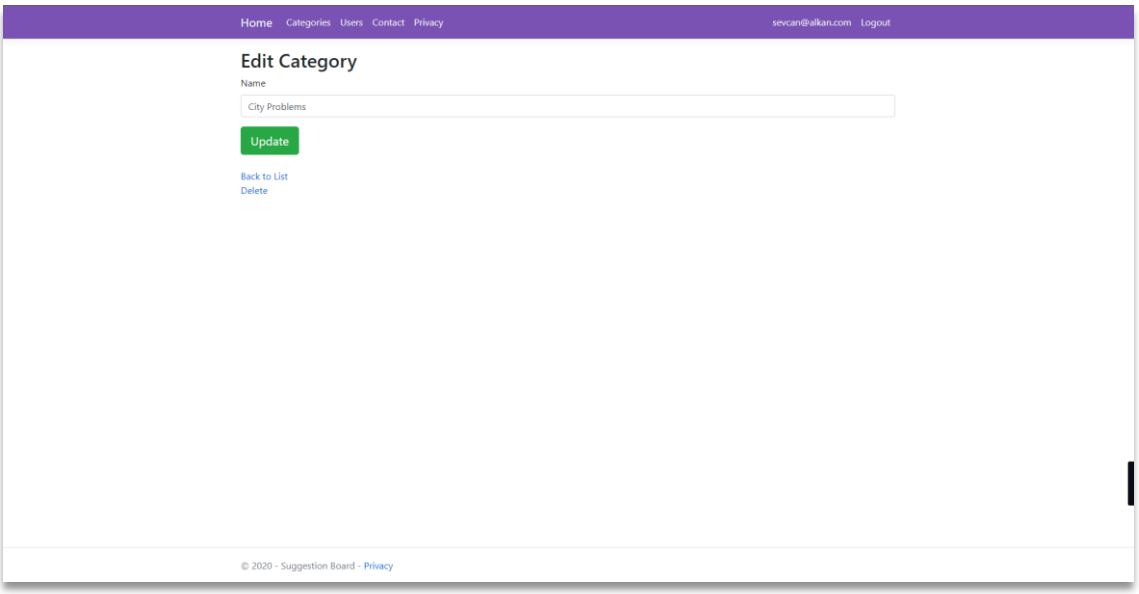
Categories are used in Suggestions, to make them much more groupable, and to associate with main topics. Each Suggestion must have a Category. The list page is making it possible to find all categories. Users can edit or view a category via the selection on the list. Also, the list page has a button which is to add a new category, it's redirecting to the Category-Detail page.



5.6 Category Detail Page

Controller	CategoryController
URL Path	/Category/Detail

This page using for adding a new category, editing an existing one, and viewing a category. The thing is a user can edit and delete only their own categories. If the category didn't add by the current user, the user only can view it.



5.7 Suggestion Detail Page

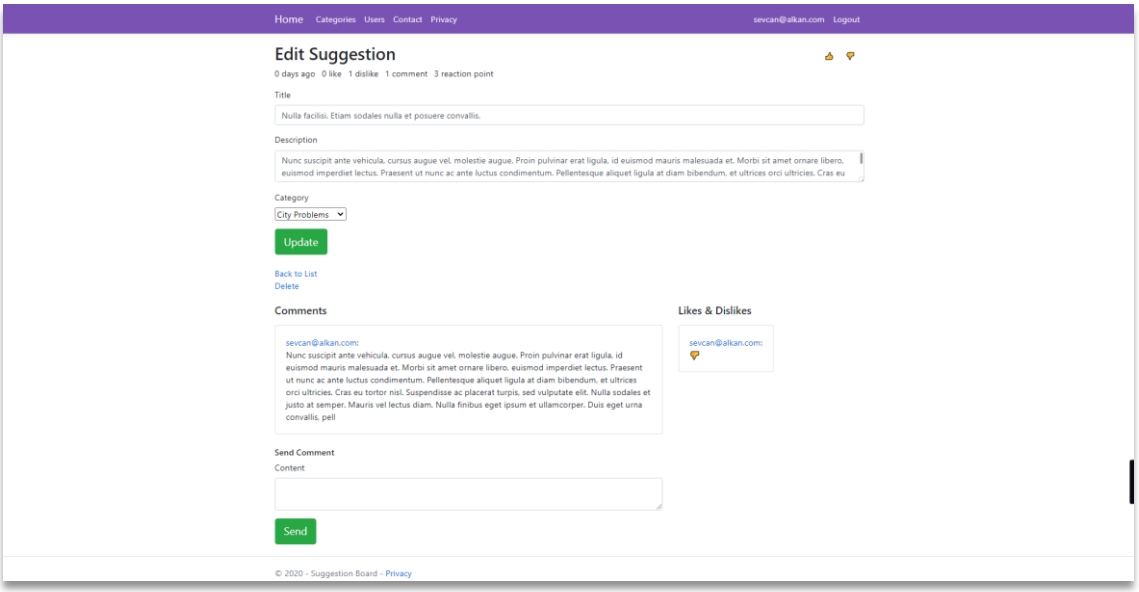
Controller	SuggestionController
URL Path	/Suggestion/Detail

The detail page is using for adding a new suggestion, editing the existing one, and viewing suggestion details. If the endpoint called with an ID parameter and is the ID value matching a record from the database, the page loads with suggestion record data. Also, if the current user is the creator of the record, editing options will be loaded. Otherwise, the record loads as only viewable. If there don't have an ID parameter, an empty page will be loaded to add a new suggestion record.

The reaction buttons are located on the top right section of the page. They are using to giving a “like” or “dislike” reaction to the suggestion. A user cannot give the reaction more than one.

Comments and given reactions are located on the bottom section of the page. Comments are listed on the left side, and reactions are listed on the right side. Also, users can use the comment form to send a new comment. The comment form is located at the bottom of the comment list.

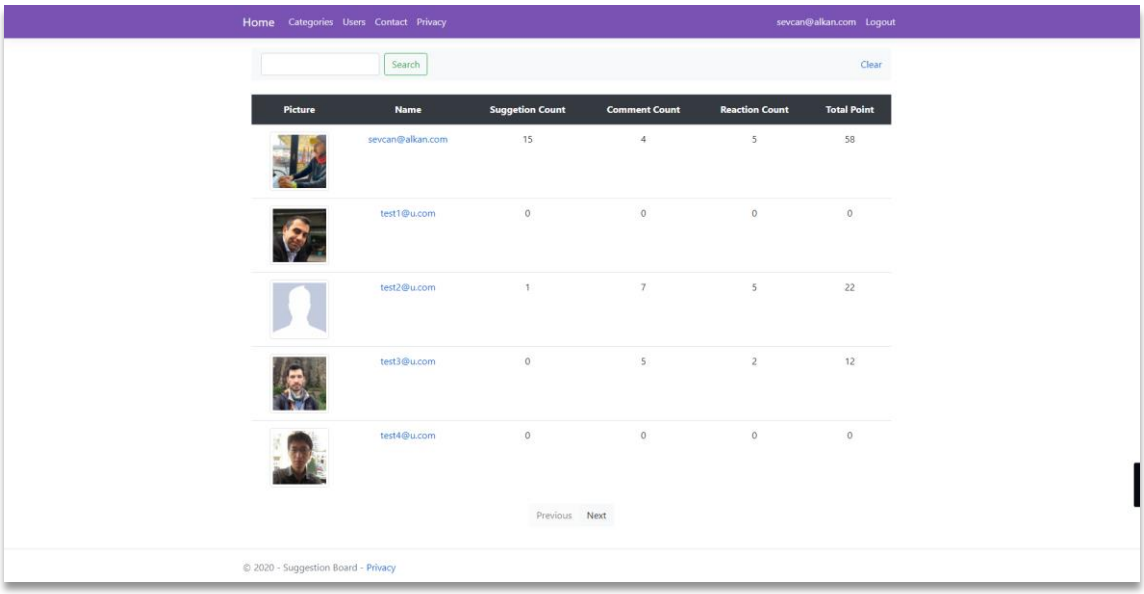
Reaction buttons, reactions list, comments are not visible and useable on new record creation. That means when the page loaded without a suggestion record data.



5.8 User List Page

Controller	UserController
URL Path	/User/Index

The page is using to list all users and their main stats. Thus, users can explore other user's profiles. The list has only a search option. There do not have sorting options, the list sorted by username filed as default.

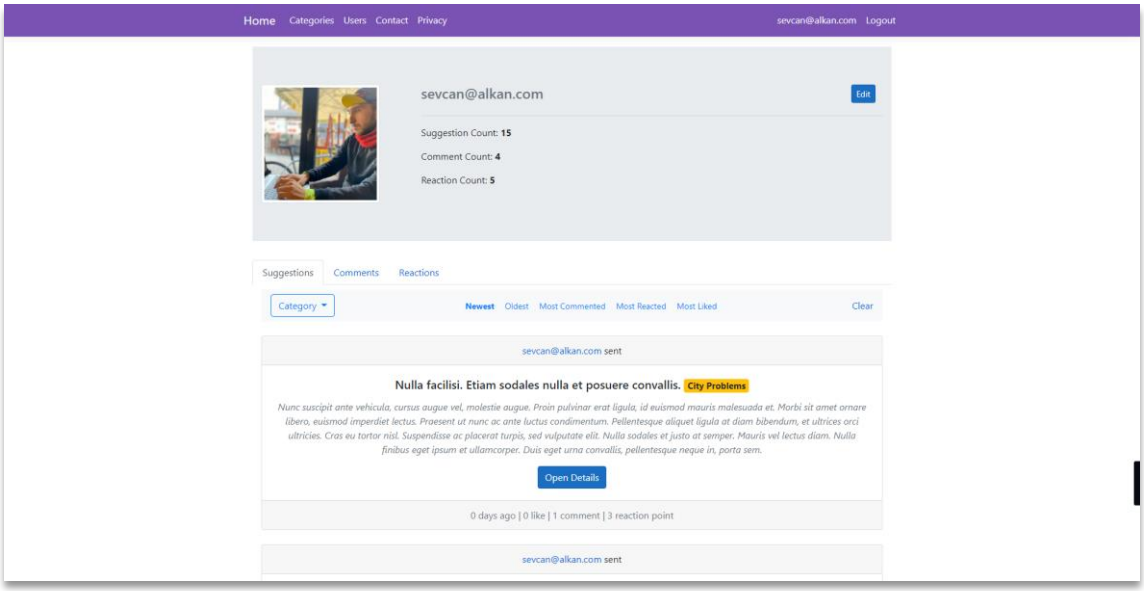


5.9 User Profile Page

Controller	UserController
URL Path	/User/Profile

The profile page shows the user's main stats, suggestions, comments, and reactions. If the profile page loaded with another user's ID, the edit button will be hidden. But if the page called without an ID parameter, the page loads for the current user. When a user clicks their username from the navbar, the page loads for the current user. Also, when a user clicks another user's username (from suggestion/comment/reaction/user list), the profile will be loaded for this specific user.

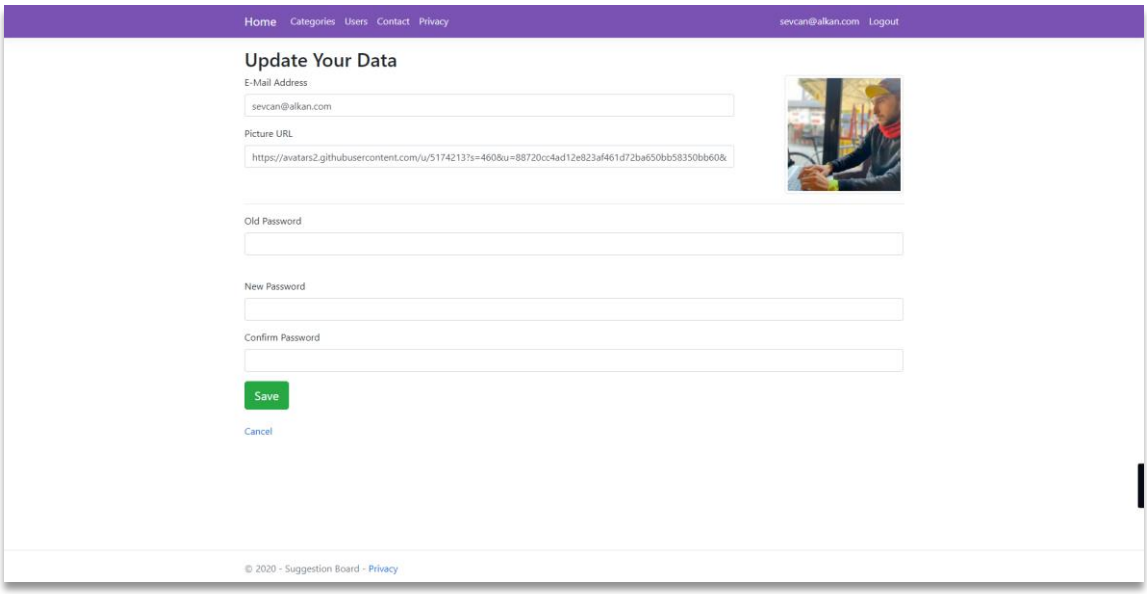
User picture, name, main stats, and edit button are located on the top section. A tab view is located at the bottom, it contains a suggestion list, comment list, and reaction list for the user.



5.10 User Edit Page

Controller	UserController
URL Path	/User/Edit

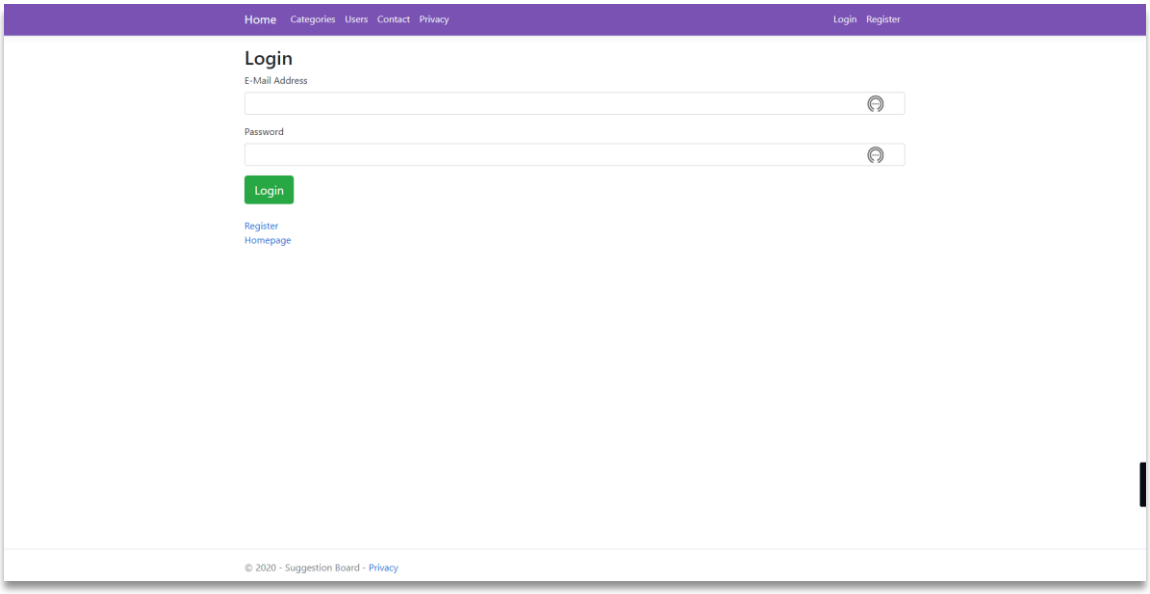
The page is very important because in some cases users can want to change their usernames or passwords. This page almost like the registration page, it makes able to change password, username, and user-picture.



5.11 Login Page

Controller	AuthenticationController
URL Path	/Authentication/Login

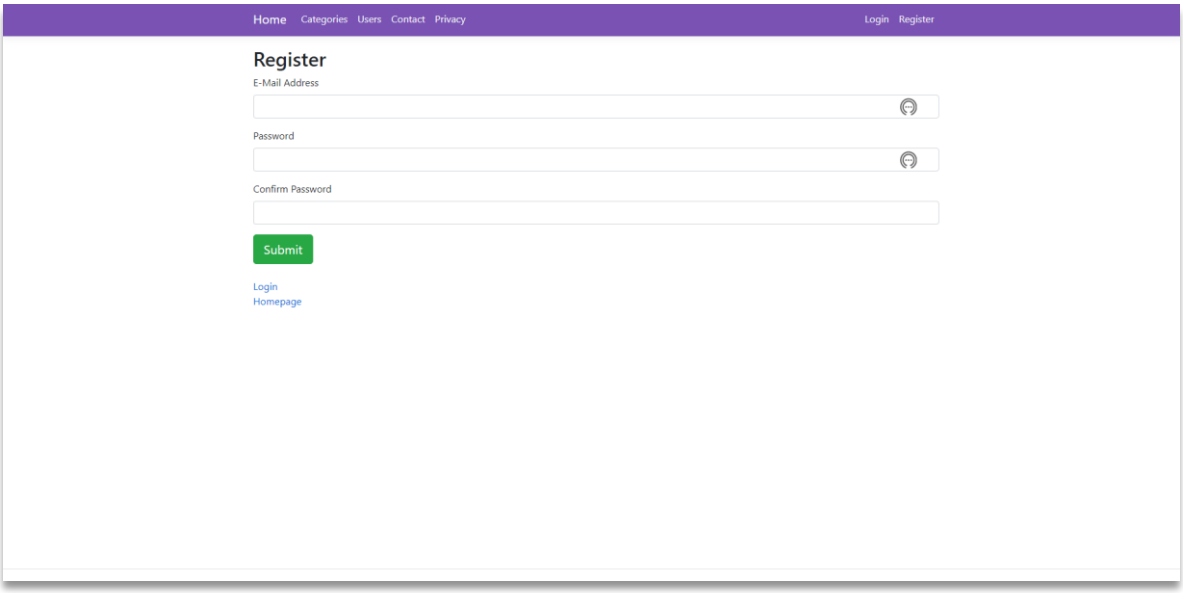
The page makes it able to do the login operations by the user. Also, the page has a redirection link for the registration page.



5.12 Register Page

Controller	AuthenticationController
URL Path	/Authentication/Register

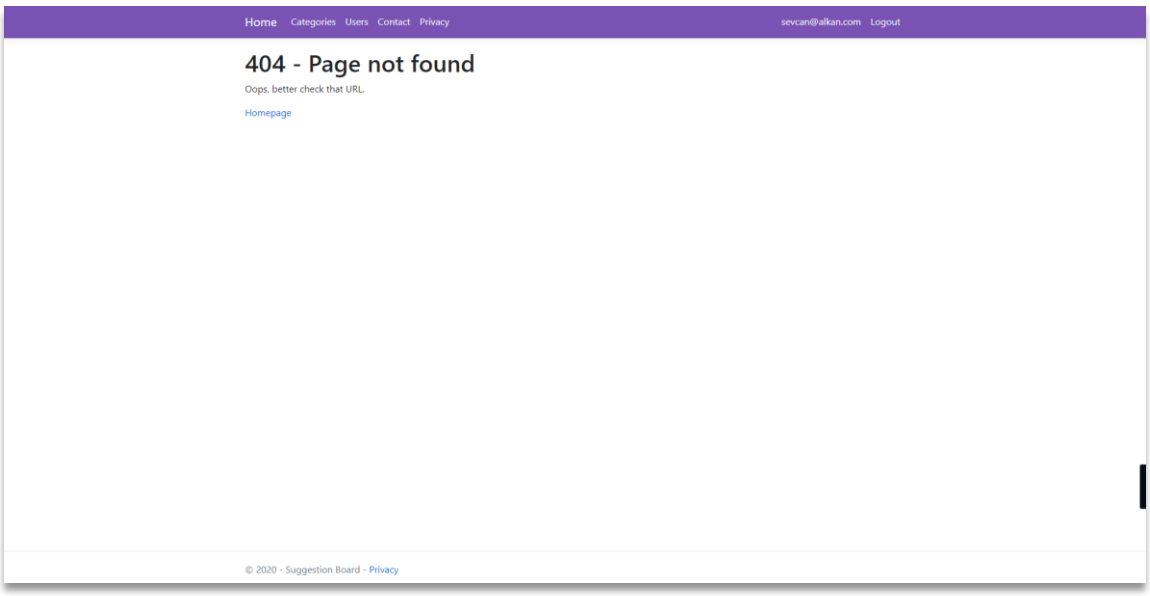
The page has a registration form that is using to create new user records on the database. Any visitor can register via this page. Login, Register, and Home page are allowing visiting as a guest, but the other pages need an authenticated user to be able for the visit.



5.13 Error (404) Page

Controller	ErrorController
URL Path	

When the URL doesn't match any existing endpoint, users redirection to the this page.



6 Using of the Project

This section describing getting ready to run the project, then using scenarios one by one.

6.1 Getting Ready to Run and Starting Project

The project uses the Docker to host MS-SQL database and Elasticsearch. Docker must be installed and ready to run the project.

6.1.1 Installing Docker, Dotnet and Git SCM

The GitHub repository should be pulled or downloaded via GitHub website UI. If we pull via Git SCM, follow this: Git - Installing Git (git-scm.com) link. To pull project files: Cloning a repository - GitHub Docs. Or just download as ZIP file and then export the files. The project files are located in this repository: GitHub - SevcanAlkan/SuggestionBoard

Install Dotnet Core 3.1 SDK: Download .NET Core 3.1 (Linux, macOS, and Windows) (microsoft.com). Also, install Entity Framework Core Tools via "dotnet tool install --global dotnet-ef" command. You can find more information here EF Core tools reference (.NET CLI) - EF Core | Microsoft Docs

After the pulling project files, we should install docker. To do that follow this: Docker Desktop overview | Docker Documentation link and install Docker Desktop.

6.1.2 Starting Project and Services

Firstly, open the command-line and then navigate to directory of the project files. The "docker-compose.yml" file must be located into this directory, be sure about that. Execute "docker-compose up" command into command-line. That process can take a while. After that MS-SQL, Elasticsearch, and Kibana will be started running on Docker.

Navigate to Suggestionboard.Data folder. Then execute "dotnet ef database update –startup-project ..\SuggestionBoard.Web\SuggestionBoard.Web.csproj" to create database on the MS-SQL Server.

Lastly, navigate to the "SuggestionBoard.Web" directory, then execute "dotnet run" command. The ASPNET project will be started. Open the web browser and enter https://localhost:5001 URL into the address bar.

6.2 Register New User and Login

Click on the Register button from the Navbar, the application will redirect to the registration page. To create a new user an email address and password must be specified. When the user clicks the submit button on the registration page, the new will be created and the user going to redirected to the login page. Fill in the login form via the created user's email address and password. Then send the form, login request going to be procced and the user will be redirected to the home page.

6.3 Listing, Adding, Editing and Deleting Categories

Before creating a suggestion, we must create some categories. Navigate to the category list via Navbar. The first time, the list will be empty, click on the "Create New Category" button to solve this. When the category detail page opened to enter the name of the new category, and then send the form. The first category added to the database!

The user is able to see existing categories on the list page now. There are edit and delete buttons for each entry. If another user created the category record, only the view button will be visible. Users can edit via the "Edit" button, like creating a new record. Also, can delete via the "Delete" button.

6.4 Listing Suggestions and Sorting, Filtering options

The suggestion list is located on the home page. The home page has the "Send New Suggestion" button, sorting and filtering options, and a list of suggestions.

The list has some filtering and sorting options, there are listed below.

- **Category selection:** if it's not empty, only gets records of the selected category.
- **Search box:** When the users enter a text, suggestions are shown which are contain this text on their Title or Description filed.
- **Sorting options:** Users can select only one of them, and suggestions sorting by this selection.

6.5 Creating New Suggestion and Editing Existing Suggestion

To create a new suggestion record, the user should click on the "Send New Suggestion" button which is located on the home page. When the user clicks on this button, the suggestion-detail page load. Title, Description, and Category fields are required fields to create a new record. Users can create a new record via this form. After the creation, the user going to redirect back to the home page.

Editing also same with creating new record. To edit your suggestion, find the suggestion on the list. Then click to the “Open Details” button. You going to redirected to the detail page and the record going to be editable. But users cannot edit the comments and reactions!

6.6 Commenting and Giving a Reaction to the Suggestions

Open detail of an existing suggestion. To commenting users can use the comment form which is located on left bottom. To giving like or dislike, users can use reaction buttons which are located on top right.

6.7 Exploring Other Users

Firstly, the user should open the Users page, the user can use the navigation bar to open it. The Users page has a search box for filtering the users, also the page has a list that contains the users.

The users can use that search box to apply filtering to listed users. When any text entered into the search box, only which users have this text on their username that users showing on the list

page. Each user is listing with the user's picture, username, suggestion count, comment count, reaction count(dislike & like count), total point(calculated value from all stats) on the list.

To open the profile page of a registered-user, the current user should click on the username from the list. The user picture and stats also seeable on the profile page. Also, all suggestions, comments, and reactions of the user are listing as more detailed.

7 Solution Structure

The SuggestionBoard project has one solution and several projects into this solution. That means the application is using the N-Tier structure.

Firstly, the solution has projects which are listed below.

- SuggestionBoard.**Core**
- SuggestionBoard.**Data**
- SuggestionBoard.**Domain**
- SuggestionBoard.**Test**
- SuggestionBoard.**Web**

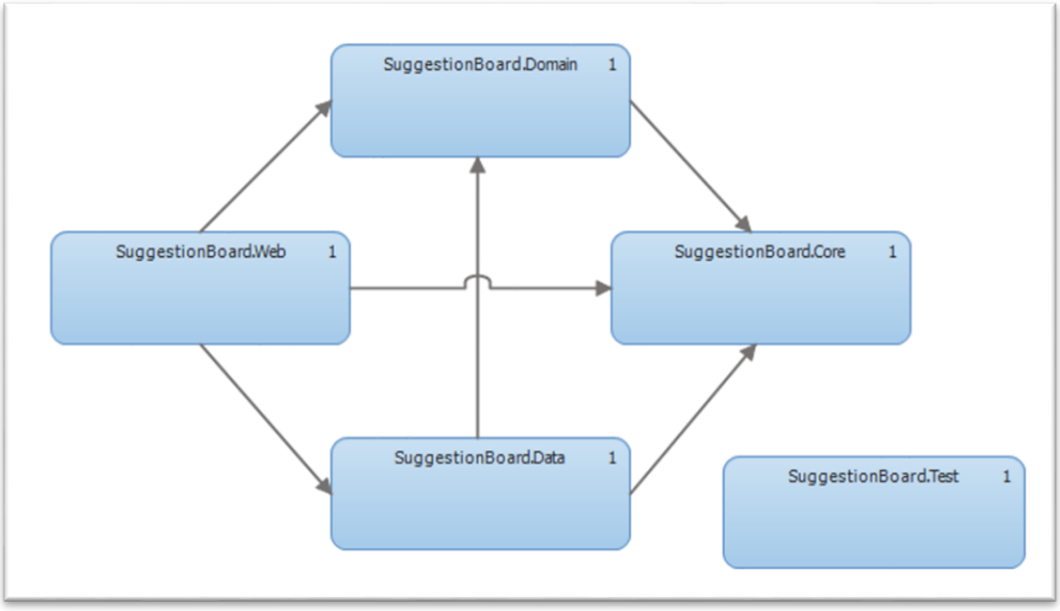
The Core project contains useful classes which are don't need to dependencies of the other projects and which have more than one implementation. Also, this project is can store in an independent repository, which makes it available to use the other development projects in the future.

The Domain project contains only the EntityFramework(ORM Framework) domain(poco) classes. These classes tracking by the EF, and the EF using them to manage the database. Each class is a definition for a database table.

The Data project conations all business logic. These are DBContext, Repository, Service, UnitOfWork, ViewModel classes. Also, generated migration files by EF.

The Test project created but not developed yet, it will be completed in the future release(V1.1).

The Web project is the ASPNET MVC project which contains Views, Controllers, Configuration. Also, it is the startup project of the solution.



The diagram shows the dependencies between the projects. Each project has a dependency on the Core project. Also, the Data project has a dependency on the Domain project. And the Web project has dependencies for Data, Domain projects as additionally.