SYMPLICITY careerhub

Developer Evaluation Project

Task

Create a web application that allows a user to request a random image that the user can like or dislike. Read this document in its entirety before you begin.

Ideal user story

1. A random image is displayed to the user
2. User likes or dislikes the image
3. Preference is sent to the server to be persisted
4. The user can request a new random image or view the history of previously liked or disliked images

Requirements

- The back-end must be built using C# ASP.NET 4.*, .NET Core or 5+
- The front-end can be built using any framework and libraries you choose (ASP.NET MVC views, Razor Pages, AngularJS, Angular, React, jQuery, Blazor, etc)
- A user can be identified however you see fit, however authenticating and verifying the user identity is not necessary
- The user's preferences must be persisted in back-end storage (SQL Server, CSV, SQLite, etc)
- Images can be stored locally, retrieved from external services, or generated dynamically. We have prepared a zip file of images you can use here: https://drive.google.com/file/d/1Uvn2GCYgzGjPkEd_HjAffuiR3u5TB8dD/view?usp=sharing
- When you submit your solution, send the entire solution's source code

Notes

The requirements are deliberately vague as we are interested in how you choose to solve and implement this task.

The task does not have to be finished to completion or even build/run. Its purpose is to demonstrate your technical skills and knowledge. Complete as much as you can in the given timeframe.

While there is no time limit, we do value your time and suggest you spend up to 3 hours on this task. Focus on demonstrating a variety of your skills rather than getting everything done and working properly. If you don't finish in the 3 hours don't worry, just submit what you have and provide notes on what further enhancements you would have implemented if you had more time.

The provided wireframe diagrams are suggestions on how the frontend might look. These are not strict requirements, just suggestions.

The below steps are intended to serve as a guide for implementing the above user stories. These are not strict requirements, and you are free to change the implementation as long as the user stories are fulfilled.

Steps

Setup

1. In Visual Studio, create a new solution. If you've chosen to use MVC, Angular, React or Blazor, there are templates available in Visual Studio to set these up quickly
2. Ensure your solution contains a project with API Controllers
3. Decide on how you are going to persist the data, and set up that data store

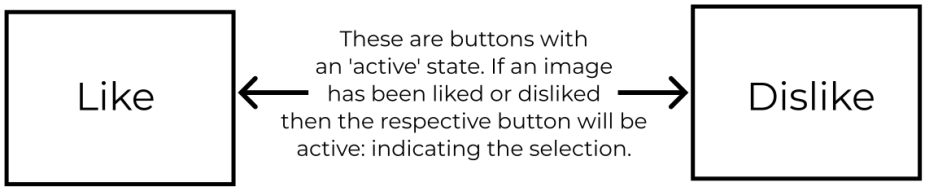
Backend

1. Create a **GET** API endpoint to randomly load an image from your chosen source
    - This endpoint should return either the image file itself or a URL it can be loaded from depending on your approach
2. Create a **POST** API endpoint to like or dislike a given image (👍 / 👎)
    - Implement this method so that it saves the vote in your data storage
3. Create a **GET** API endpoint that returns whether an image has been liked or disliked previously

Frontend

1. Create a page that uses the first backend endpoint to load a random image:

Image 1



2. Implement the like and dislike buttons so that they:
    a. Make a call to the second backend endpoint to save the like or dislike
    b. Trigger the next random image to be loaded
3. Implement logic that marks the appropriate button as active if a vote exists when an image loads. Use the third backend endpoint for this.