

SafeSteer Documentation

David Pérez Lorenzo | <https://www.linkedin.com/in/corium/>

Content

SafeSteer Documentation	1
Summary	2
Objective and Utilization	2
Web Application	2
API Service	2
Design	2
Organization	3
Pages	3
Home	3
Demo	3
About	4
Components	4
Skill Card	4
Learning path	4
Chips filter	5
Skill creation	5
Detail PopUp	5
Comments	6

Summary

Objective and Utilization

The objective of the application is to be part of a driving skills learning system. The system could be connected to the user's car's central unit to provide real-time feedback on the rewards obtained for each skill.

If a user wanted to acquire specific skills, such as the perfect heel-toe technique, the car could report the RPM differential between gears when releasing the clutch. Another example could be improving battery efficiency and attention through gamification, where the user could receive a reward for lifting off the accelerator at the right moment to stop the car as close as possible to the traffic light.

To determine the skills the user wants to acquire, they would need to subscribe to a Learning Path. These Learning Paths could be created by the community through a Skills and Learning Paths generator. In this case, the portion of the application I am presenting would fulfill this purpose.

A view is provided where it is possible to filter the Skills, add new Skills, view their details, or delete them. To create the Learning Path, you simply need to add or remove the desired activities from the button available on the card.

Web Application

- **Test environment:** The web application can be tested at <https://polite-mushroom-0c65cab10.3.azurestaticapps.net>
- **Source code:** The source code for the web application is hosted on GitHub at <https://github.com/coriumalpha/Safe-Steer>
- **Deployment:** The web application is deployed using Azure Static Web Application.
- **CI/CD Pipeline:** The project utilizes GitHub Actions for continuous integration and deployment.
- **Technologies:** The web application is built using Angular and TypeScript. It also uses SCSS for styling and incorporates angular core testing.
- **Akita:** The web application utilizes Akita as its state management solution.

API Service

- **Test environment:** A test environment for the API service is available at <https://steersafeapiserver.azurewebsites.net>
- **Deployment:** The API service is hosted on Azure Web App, and the deployment process is described in this repository at <https://github.com/jesperorb/json-server-heroku>
- **Source code:** The source code for the API service can be found at <https://github.com/coriumalpha/SteerSafeApi>

Design

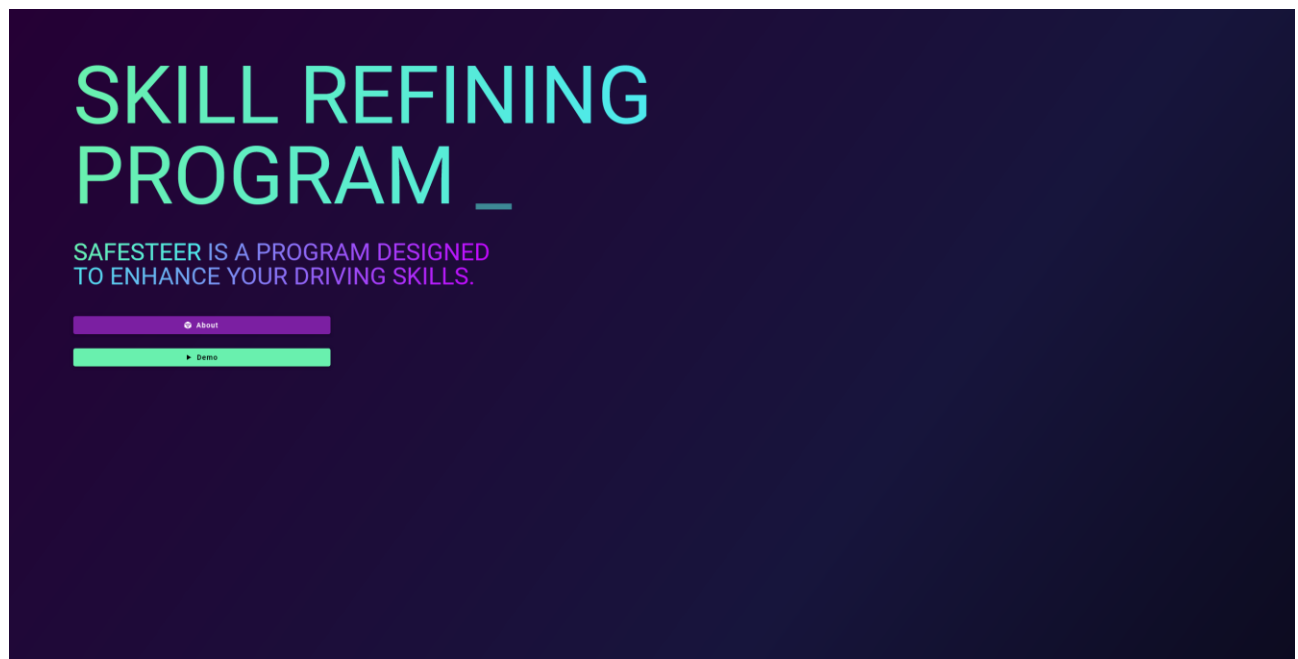
- **Material Design:** The web application follows the principles of Material Design for its user interface.
- **Custom style:** The design incorporates a custom style inspired by vaporwave art, giving it a unique aesthetic.

Organization

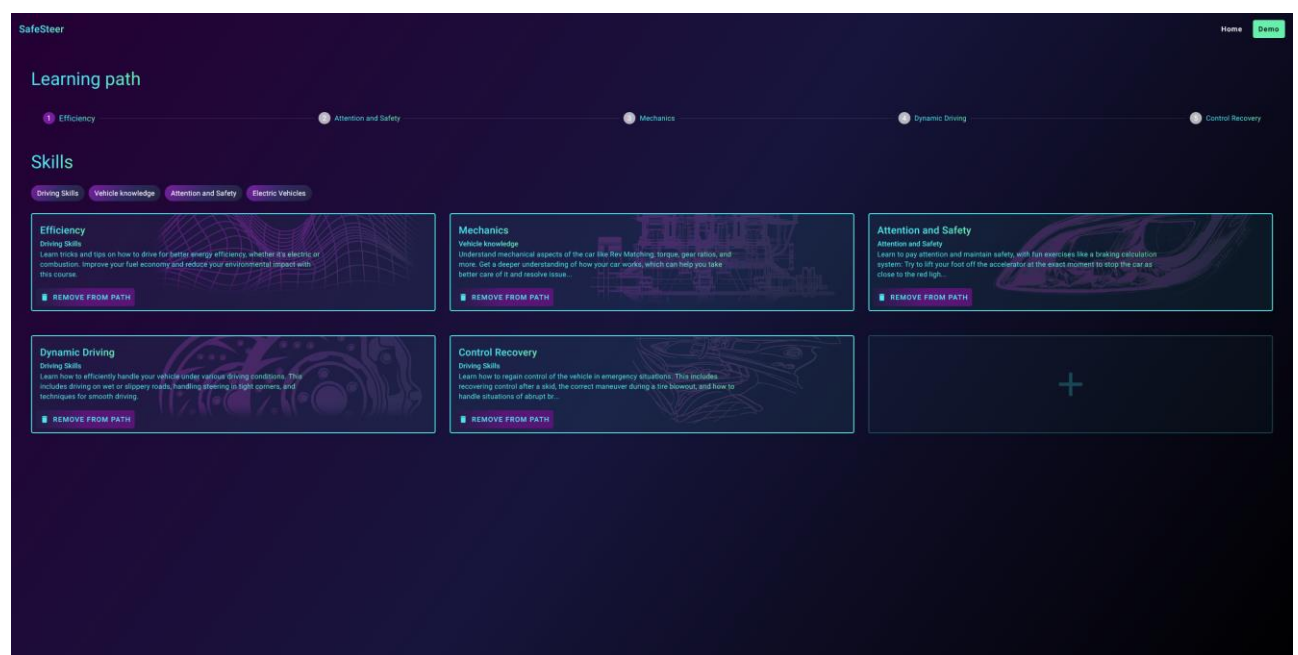
- Notion Projects: The organization and management of the project's tasks and resources are handled using Notion Project

Pages

Home



Demo

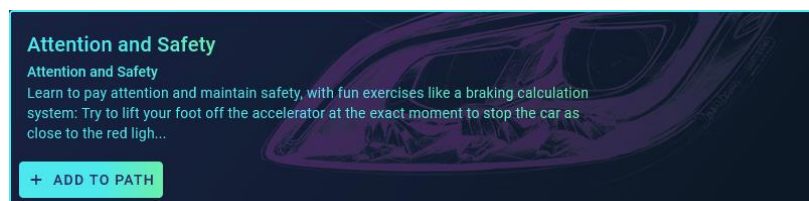


About



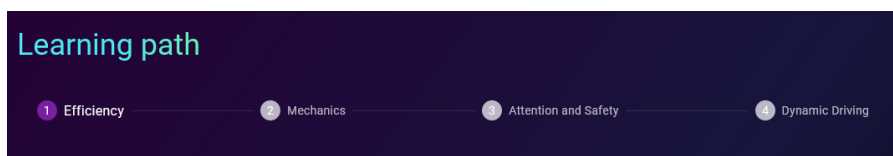
Components

Skill Card



Component aimed at displaying the details of each available skill. It allows triggering a PopUp with the details and adding or removing the skill from the learning path. It is built on the Material Card component, with a layer of SCSS customization.

Learning path



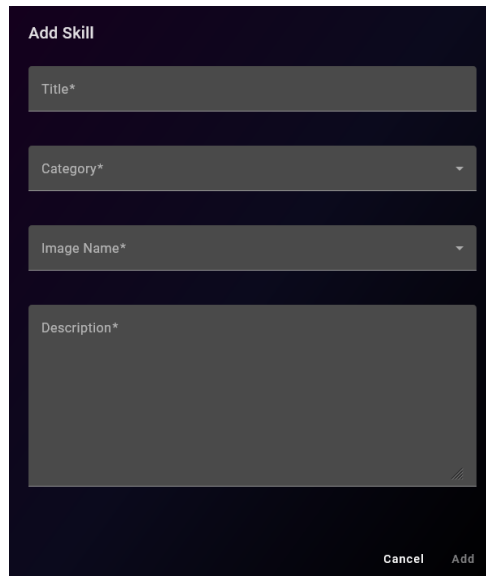
A component dedicated to displaying which skills are being configured in the Learning Path, so that the view allows the creation of a learning plan with the selected skills. It is built on a slightly modified Material Stepper to adapt its functionality to this case.

Chips filter



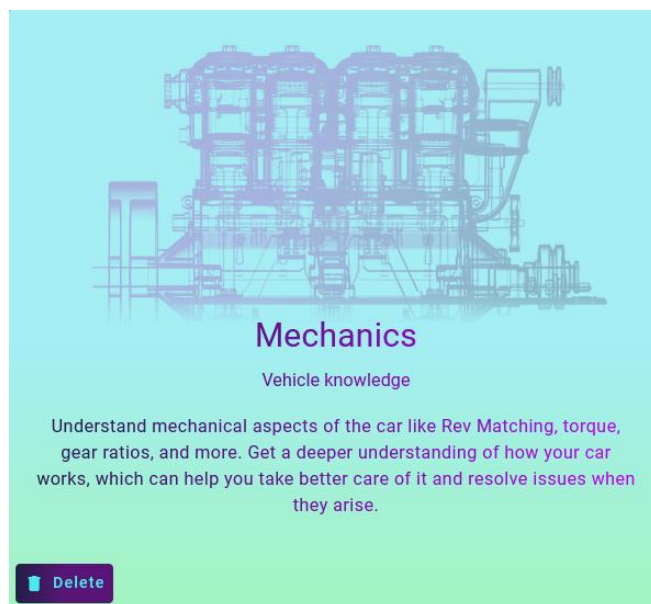
Allows filtering the view of skills by categories. It is built on the Material Chips component.

Skill creation

A dark-themed 'Add Skill' form. It has four input fields: 'Title*' (text), 'Category*' (dropdown), 'Image Name*' (dropdown), and 'Description*' (text area). At the bottom right are 'Cancel' and 'Add' buttons.

Enables the creation of new skills, with the option to select categories or images. Built on a Material Form. It probably wouldn't be so ugly if I had had more time 😊

Detail PopUp



Allows for an extended consultation of the complete description of the skill, as well as its deletion. Built on a Material Dialog.

Comments

Creating this Angular application in one week has been a rewarding and intense challenge. I've tried to focus on all areas of Front-End development to create a user-friendly application concept that also serves as a showcase for the utility of reactive components.

I decided against creating a table because I didn't think it would provide a differentiable value in terms of the user experience that can be achieved through the use of components and reactivity. I've enjoyed putting into practice the use of state managers and reviewing what I've learned about unit testing, while being aware that I started with limited experience.

I had been wanting to personally try out Azure services because the limitations imposed by my current job are quite restrictive, and I am very satisfied with the results. Thanks to setting up a CI/CD pipeline, there is always an updated version of the application on the test server.

I underestimated the scope of the proposed task and couldn't dedicate the necessary effort to each aspect of the application. I would have liked to finish styling some components, conduct a thorough overall review of the application, cover more cases in unit tests, seek advice from someone with more knowledge to assess the overall state of the application, and perform a refactoring to ensure a robust and aesthetically pleasing product.

In any case, I appreciate being given the opportunity to participate in this selection process with this interesting yet challenging task, and I'm excited to contribute my maximum effort and value to the Volkswagen Hub team! 🤓