



"Ss. Cyril and Methodius" University in Skopje  
**FACULTY OF COMPUTER  
SCIENCE AND ENGINEERING**

Seminar work for the subject  
**Advanced Web Design**

Topic: Infotainment UI

Mentor:  
Dr. Boban Joksimoski

Written by:  
Aleksandar Jovanovski - 223135

Skopje, June 2025

## Abstract

As the digital transformation of vehicles continues to accelerate, the importance of intuitive and functional user interfaces within infotainment systems has become increasingly evident. The “Infotainment UI” project was developed as a lightweight, responsive frontend interface aimed at addressing common shortcomings found in modern in-car UI design. Built using Vue.js and pure CSS, the project focuses on delivering a seamless and immersive experience that prioritizes usability without causing distraction. With carefully designed, always-visible components for navigation, climate control, and system status, as well as specialized screens for applications, driving dynamics, vehicle performance, and connectivity options, the project explores an approach to infotainment that balances functionality with simplicity. It serves as a practical example of how modern web technologies can be applied to improve user interaction in vehicle environments.

## Summary

Abstract.....	2
Summary .....	3
Introduction .....	4
Technologies Used .....	4
Vue.js – Key Features and UI Relevance .....	4
Pure CSS – Key Features and UI Relevance.....	5
Application features and functionality .....	5
Static and always visible components: .....	5
Apps screen.....	6
Driving dynamics screen .....	7
Performance screen .....	7
Options screen .....	8
Conclusion.....	9

## Introduction

As technology advances, car manufacturers have no other choice but to adapt to the impact that it has and implement it in their products. While technology makes every-day life easier it only manages it if implemented and designed correctly. Unfortunately, many of the car manufacturers choose not to put the required efforts into designing good and practical User Interfaces for the infotainment systems in the cars they produce. This is a very big mistake, especially considering the fact that the owners and all the people correlated to the cars are mostly in touch with exactly this part of the technology that is inside the cars. A bad UI can shift the whole perspective of a user for the quality of that infotainment and also the whole car. This is why it has become a trend for the car enthusiasts to avoid cars with infotainment systems and instead go back to old cars that have a more analog approach to the in-car commodities.

For this, the “Infotainment UI” project was created serving as a lightweight frontend application that can easily be incorporated into almost all car infotainment systems. The project focuses on usability and immersive user experience giving the user the right amount of information and control over their car while being light and non-distractive.

## Technologies Used

This project employs Vue.js as the JavaScript framework and pure CSS for styling the user interface. These technologies were selected for their efficiency, modularity, and support for building responsive, maintainable, and interactive web applications. The goal was to develop a user interface that is both functionally robust and visually coherent, adhering to modern web development standards.

### Vue.js – Key Features and UI Relevance

- Component-Based Architecture
  - Enables the decomposition of the UI into reusable components.
  - Facilitates organized and scalable front-end development.
- Reactive Data Binding
  - Automatically reflects data changes in the UI.
  - Improves responsiveness and reduces manual DOM manipulation.
- Declarative Syntax
  - Simplifies the creation of dynamic interfaces.
  - Enhances readability and reduces development complexity.
- Single File Components (SFCs)
  - Consolidates template, script, and style into one file.
  - Promotes better code structure and maintainability.

- Transition Support
  - Provides built-in tools for animating elements.
  - Enhances user experience through smooth visual feedback.

## Pure CSS – Key Features and UI Relevance

- Performance and Simplicity
  - Eliminates the need for additional libraries or frameworks.
  - Ensures fast load times and straightforward maintenance.
- Responsive Layouts (Flexbox/Grid)
  - Supports adaptive design for various screen sizes.
  - Essential for creating mobile-friendly interfaces.
- Custom Styling
  - Allows complete control over visual design.
  - Facilitates consistent branding and UI aesthetics.
- Media Queries and Interaction Styling
  - Enables style changes based on screen size and user actions.
  - Improves accessibility and interactivity.
- Separation of Concerns
  - Keeps styling independent of application logic.
  - Encourages cleaner code and collaborative development.

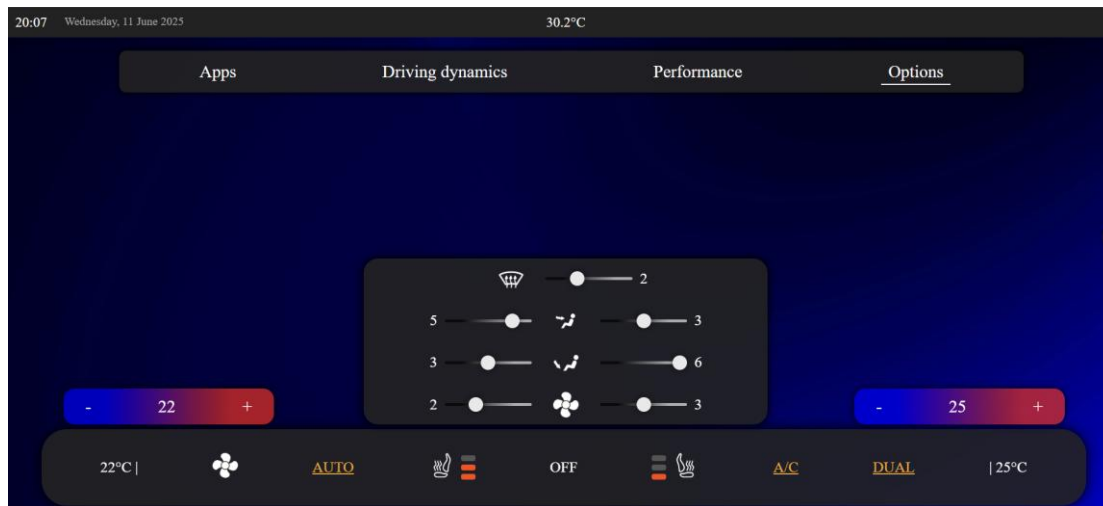
## Application features and functionality

### Static and always visible components:

The top view component serves as the persistent header of the application, providing essential real-time information and status indicators that enhance usability and user awareness. The header includes a real-time clock and date display which is used to show the current date and time, and is updated every second to ensure real-time accuracy. Another key feature of this component is the live temperature display which asynchronously fetches the current temperature using the WeatherAPI. The final feature of this component is the Options panel which dynamically displays the currently active options such as wi-fi, Bluetooth, location and Mobile Data. The component is always visible ensuring constant user awareness of the critical environmental and system information. While being always on screen, this component has a minimalistic design ensuring that the user is not distracted by it.

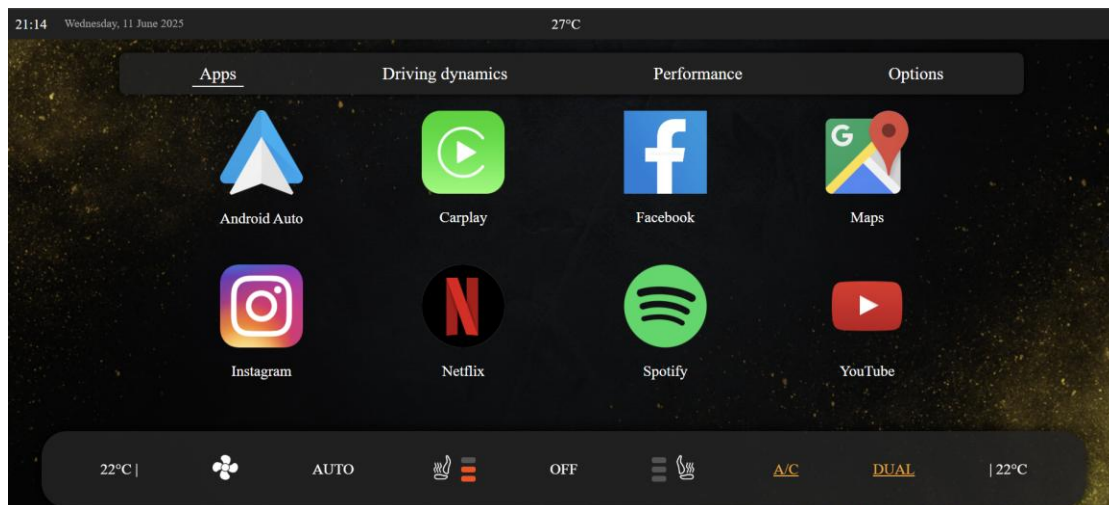
The navigation bar is another always visible component that controls which of the main screens is showing on the display. For a clearer perspective the active screen is underlined with an interesting visual effect that enhances the visual appearance of the system.

The climate control component provides an intuitive and visually responsive experience for managing the cabin environment. It includes features such as: dual-zone climate control, seat heating control, heating and airflow management. It is divided into two parts, one of which is the main control panel that shows all the modes and serves as an information panel, the other part are the three additional panels that control the temperature for each side of the dual-zone climate control and the air-flow distribution. This division is done so that the driver the key information available at any time while being able to adjust the cabin environment with minimal effort.



## Apps screen

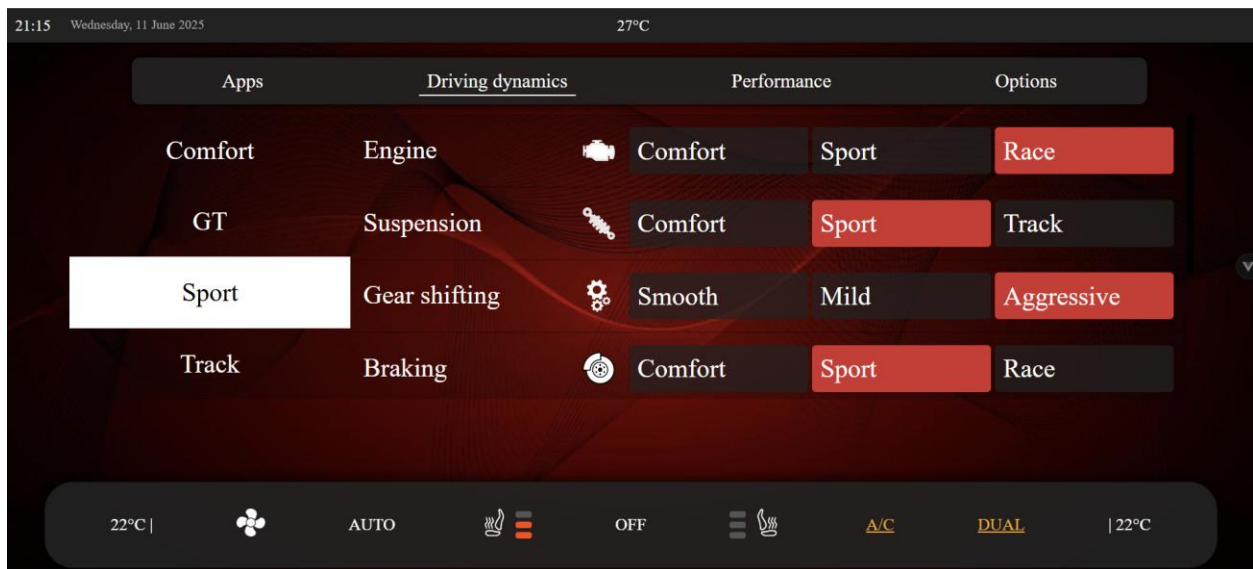
This component consists of some of the most used applications when it comes to infotainment systems. It again follows the light nature of the application showing only the applications that a user would need.



## Driving dynamics screen

This component provides an interactive customization interface that allows the user to adjust the behavior of their car to their liking. The component consists of a selection of presets that control various aspects of the car's driving characteristics such as: engine aggressiveness, ride and stability control, distribution of power from the engine to the wheels and traction and stability control. It offers the user the convenience of pre-selected modes or the freedom to further fine tune the car, for all of the enthusiasts who think they now better than the manufacturer. Another key aspect of this component that is more visually focused is that depending on the selected mode the whole appearance of the application changes further increasing the emotion that the car gives the user.

The appearance change that follows throughout the whole application is made possible using the state management library Pinia. Once the user chooses one of the driving presets, that state is stored and then passed to the other components which then use it for their appearance.



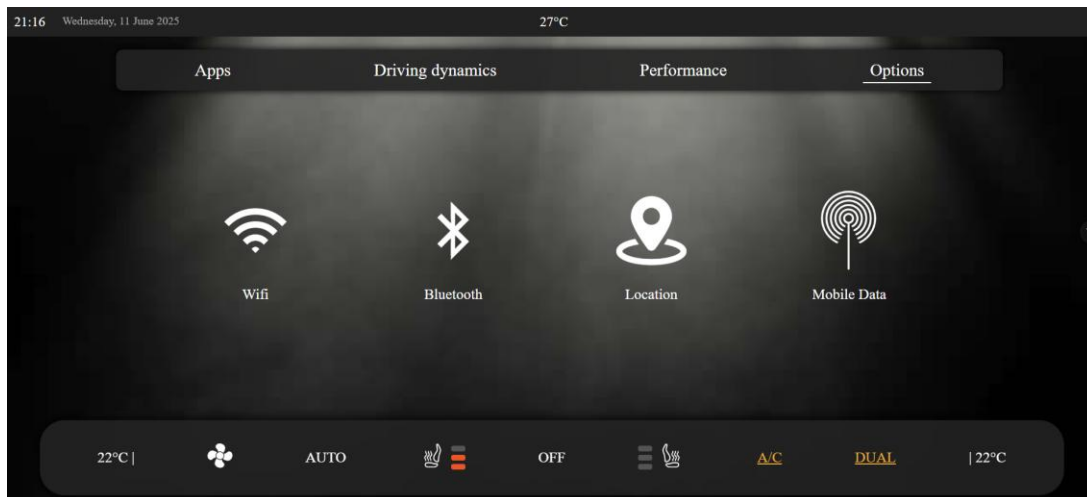
## Performance screen

The performance component visually represents the live car telemetry through animated progress bars. This gives the user the inside of what the car is doing at any given moment. While providing an interesting visual effect for the driver and passengers it also gives crucial information about the car's health. It is very beneficial for the user to know the car's behavior for diagnosing problems and preventing them in the future. For the purpose of this project the component simulates a working petrol-powered car engine with realistic acceleration and deceleration animations. These animations display calculated engine performance values such as: engine revolutions per minute, power, torque, boost and oil temperature.



## Options screen

This component is a toggleable settings panel which visually presents the connectivity features like: Wi-Fi, Bluetooth, Location and mobile data. It is meant to function like a quick settings panel using icons and labels for greater visual appearance and easier usability.





## Conclusion

The “Infotainment UI” project successfully demonstrates how a lightweight and thoughtfully designed interface can significantly enhance the overall experience of modern car infotainment systems. By addressing the common issues found in current implementations—such as cluttered layouts, inconsistent interactions, and limited feedback—this project offers a practical and aesthetically pleasing alternative. Through the use of Vue.js and CSS, the interface remains both maintainable and scalable, allowing for easy customization and future feature expansion. The integration of state management with Pinia further elevates the user experience by ensuring visual consistency and reactive control across the application. Whether for everyday drivers or automotive enthusiasts, this infotainment system combines functionality and design to deliver a product that aligns with both user expectations and evolving technological standards in the automotive industry.