Technology: Javascript, Vue 3.2 js, Pinia js, Vite js, Tailwind css, Swiffy Slider js 1.5.3

check code: GITHUB
check live demo: LIVE DEMO

Project description

Follow up 'Pinia: The Enjoyable Vue Store' from 'vueschool.io' course.

Exploration into using Vue Pinia to create a dynamic store front and a shopping cart that display products and prices, store user product choices, add product price to total sum and provide a cart checkout method.

Approach and technology brief

The project is built on Vite for Vue 3.2 that among other features, allows developers to use the Vue 3.2 script setup, offering a more efficient way to work with the Vue composition API.

Following the course 'Pinia: The Enjoyable Vue Store' from 'vueschool.io', I have covered important state management topics such:

-Creating modular stores
-Storing state -
-Mutating the state with actions
-Sync and async actions
-Computed data with Pinia getters
-Pinia and the Composition API

Subsequently, I have implemented features such:

-A Pinia plug-in that stores user cart information in local storage and prevent loss of data if the app is reloaded (covered in the advance section of the above course)
-A connection to a real database (initially the source was a Jason file)
-A category product filter based on a computed property from the original data.
-A new design using Tailwind style library for Vue j with dark mode available.

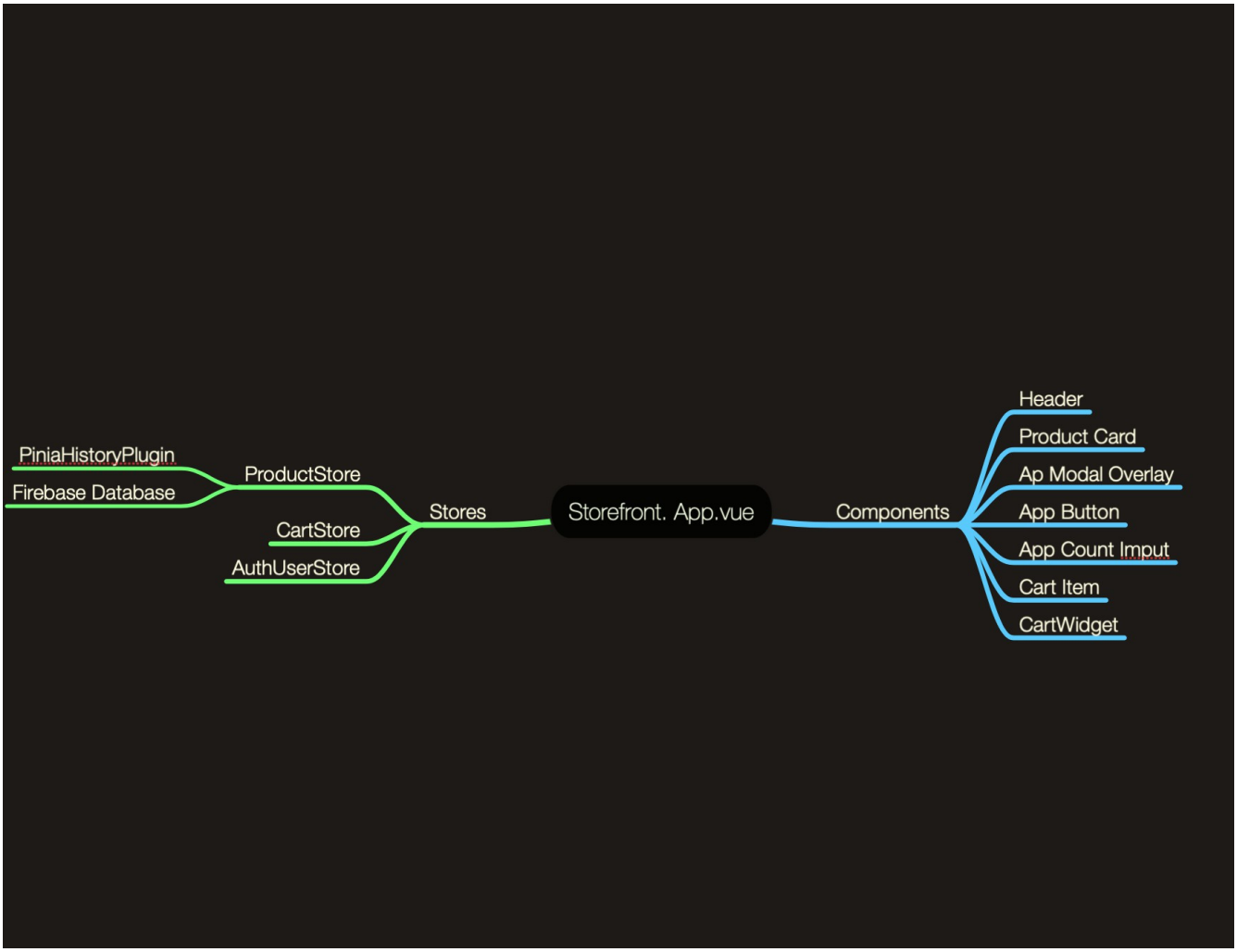
Project Structure

In contrast to other state management libraries, Pinia is essentially modular. This allows to create several shops adapted to your application in a single project.

The stores are split into three departments, each with its own set of tasks.
-Product store: store and deliver actions with data extracted from a database.
-Cart Store: storage of users' product selections, addition of total prices and method of payment.
-User Auth: a third store that allows user identification via gmail (yet to be implemented, coming soon)

The application is divided into a number components according to best practices on building single file Vue components. They are re-usable and customizables in essence

I have concluded by re-designing the app style using Tailwind library for Vue jails' (installed as a dependency) and making use at some cases of third party Tailwind blocks.
The products are shown into a horizontal carousel slider. For this tasks I have used a lightweight javascript library (swiffy-slider 1.5.3)that provides a carousel functionality component.
Two modes (dark, light) are available, providing two contrast designs for night and day time.



Conclusion

Through this project, I have acquired all the necessary skills to build a simple front shop structure that is transferable to different shop approaches. The code is maintainable and scalable, and its web performance is optimized for all kinds of screens and browsers.

Next Steps

- Enable firebase authentication via Gmail
- Implement user information storage(cart, wish list, profile)
- Improve security rules