

# Progetto Vue.js 2

## START 2 IMPACT

Questo progetto, mi ha messo davvero alla prova per la prima volta durante il mio percorso di studente Front-End, sono state 3 settimane piene di scrittura, cancellature, innumerevoli 'npm' nel terminale e momenti di sconforto, ma sono contento che come sempre, la teoria insieme ad una generosa dose di pratica, portano sempre ottimi risultati, la mia difficoltà più grande? Certamente è stato l'utilizzo e la comprensione di **Vuex**.

# Components

## NAVBAR.VUE

Questo è il primo componente visualizzato in ordine discendente, è molto semplice ha il compito di mostrare il nome dello store (kreas) e da esso rimandare alla schermata **Home** tramite **router-link**, e l'icona del carrello che rimanda alla pagina **Cart** tramite **router-link**, con accanto un counter che mostra il numero di elementi presenti nel carrello grazie alla funzione **cartItemCount()**

# Components

/PRODUCT/ALLPRODUCTS.VUE

Il componente `AllProducts`, consiste nella pagina in cui vengono mandati a schermo tutti i prodotti presenti nel json messo a disposizione per il progetto, tramite la chiamata alle api con `axios` e con un `v-for`, mostra sotto forma di cards i dettagli principali del prodotto. Cliccando sull'immagine dell'elemento, si passa alla visualizzazione singola dello stesso, grazie alla funzione `handleImageClick(product)`

# Components

/PRODUCT/PRODUCTPAGE.VUE

Questa è la pagina specifica del singolo prodotto, vengono mostrate tutte le caratteristiche contenute nel suo array.

## Button :

- **Add to Cart** = cliccando, si aggiunge il prodotto al carrello grazie ad `handleAddToCartButton(product)`, se il prodotto è già stato aggiunto, grazie al `v-if="checkInCart(product)`, verrà mostrato il button disabilitato che avvisa "Added to Cart".
- **Close** = chiude la pagina e riporta alla schermata con tutti gli items con `@click="handleClose"`

# Axios

`SRC/AXIOS/AXIOS.JS`

In questo file js, ho dichiarato nella function `getProducts()`, la chiamata axios che contatta le api fornite per il progetto, inizialmente la ripeteva in ogni componente in cui mi serviva contattare le api, poi ho snellito un bel po' il codice con questa funzione

# Router

## INDEX.JS

In questo index ho dichiarato tutte le routes per sfruttare tutti i router-link inseriti nei componenti, buona svolta per un principiante come me che viene dalle href di Html..!

# Views

## CART.VUE

In questa sezione, si visualizzano gli elementi aggiunti al carrello

### Functions

- `handleIncButton` : incrementa di +1 la quantità di un prodotto
- `handleDecButton` : decrementa di -1
- `handleRemoveButton` : rimuove dal carrello il prodotto
- `const singleItemTotal` : il prezzo totale relativo ad determinato item
- `total = computed()=>` : il totale del carrello
- `discount = () =>` : 10% di sconto sul totale per 3 o più prodotti nel carrello

# Views

HOME.VUE

```
<Products v-if="!showProduct()" />
```

```
<SingleProduct v-if="showProduct()" />
```

Queste, le due funzioni cardine, una esclude l'altra permettono di visualizzare nella Home, l'intera lista di prodotti, altrimenti, un solo prodotto selezionato. Questo, seppur banale è stato il passaggio che più mi ha messo in difficoltà, poichè confrontando tante guide online non riuscivo a far sì che venisse richiamato un solo elemento del json, allora ho deciso di ragionare al contrario e mascherare tutti gli altri dopo aver selezionato l'elemento desiderato.



# Store

## INDEX.JS

Vuex è stata la svolta per la realizzazione e la gestione di questo progetto, ho dovuto approfondirlo bene teoricamente e vedere molti esempi pratici prima di comprendere la sua potenzialità, rendendomi conto che si tratta di un plugin fondamentale.

Nell index, ho semplicemente importato vuex, lo store e i moduli, dichiarando il tutto nella funzione `createStore()`

# Store

`MODULES/CART.JS + MODULES/PRODUCTS.JS`

Tramite svariate (innumerevoli) guide online in cui venivano realizzati progetti simili relativi ad e-commerce, mi sono fatto guidare su come e quali actions, mutations, getters, state dichiarare, e cosa scrivere al loro interno, mi ha aiutato un po' di sano cut & paste con le dovute modifiche affinché il tutto funzionasse, complesso ma non impossibile!

# Link utili

EMANUELE FOTIA - START2IMPACT

Li

- Link alla repository : <https://github.com/emanuelefotia/VueJS2-S2I-Emanuele-Fotia>
- Link al progetto : <https://kreas-shop.netlify.app/>