

Relazione Progetto “Portfolio” - realizzato da Sara Casarano

Il progetto che ho sviluppato consiste in una **Single Page Web Application** creata con **Vue.js 3**, **Vue Router**, **Vuex** e **Bootstrap 5**.

Il tema scelto è quello di un **portfolio professionale personale**, pensato come se fossi già una designer, con l’obiettivo di presentare le mie grafiche e i miei font, raccontare i progetti realizzati, offrire un modulo per richiedere preventivi e creare uno spazio di confronto aperto tramite un piccolo forum in cui gli utenti possono pubblicare dei post con le loro idee, dubbi o curiosità.

Tutta l’applicazione è progettata secondo i principi del **responsive design** grazie all’utilizzo di Bootstrap per la maggior parte degli elementi integrati, così da adattarsi automaticamente a smartphone, tablet e desktop.

La struttura del sito è suddivisa in più **viste (pagine) gestite da Vue Router**:

1) La **HomeView.vue** accoglie l’utente con una presentazione personale, una foto profilo, una sezione “Chi sono” e l’elenco delle “Skills Visive” con icone.

2) La **pagina GraficheView.vue** mostra un elenco di card prodotto su grafiche e font realizzati da me (rese dinamiche tramite un hover zoom in): qui l’utente può cercare con un campo filtro, aprire una modale per vedere i dettagli di ogni elemento e aggiungerlo al carrello. Dopo ogni aggiunta compare una notifica toast personalizzata in basso a destra.

3) La **pagina ProgettiView.vue** utilizza un **carosello Bootstrap** per mostrare delle anteprima di alcuni lavori svolti, controlli “prev/next” e un box sottostante che evidenzia il progetto attivo. Un pulsante “Dettagli” apre una modale che riporta descrizione completa, data, tempo di realizzazione, funzionalità principali e tecnologie utilizzate.

4) La **pagina ContattiView.vue** ospita un form per richiedere un preventivo personalizzato: l’utente sceglie il tipo di progetto da un menu a tendina, indica un budget, descrive l’idea e inserisce la propria email. Tutti i campi sono stati resi obbligatori da compilare tramite un “required”. Alla conferma compare un messaggio di successo e i campi vengono resettati.

5) Il **ForumView.vue** è un piccolo spazio di discussione in cui gli utenti possono pubblicare post e rispondere ai messaggi esistenti. Tutti i contenuti sono salvati in **localStorage** così da mantenere i dati anche dopo un refresh.

6) Infine, la **pagina CarrelloView.vue** mostra gli articoli selezionati con quantità e prezzo, permettendo di rimuovere singoli prodotti o svuotare l’intero carrello.

Nella **Navbar**, sempre visibile insieme al **Footer** (completa di diritti riservati), compare un contatore live degli articoli presenti nel carrello.

Ho integrato due **componenti Vue personalizzati** per incapsulare funzionalità riutilizzabili.

In particolare:

- **ProductCard.vue**: gestisce la visualizzazione delle singole grafiche/font in formato card, con pulsanti per aggiungere l'articolo al carrello e aprire il dettaglio.
- **ProductDetailModal.vue**: mostra le informazioni complete di un prodotto in una finestra modale Bootstrap, integrandosi con Vuex per l'aggiunta diretta al carrello.

Lo **stato globale** dell'applicazione è gestito con **Vuex**. Lo stato "grafiche" simula un piccolo database di prodotti, mentre "carrello" contiene gli articoli aggiunti dall'utente.

Le mutations aggiornano questi dati e li sincronizzano automaticamente con **localStorage**, così da garantire la persistenza anche dopo un refresh. Il forum sfrutta lo stesso meccanismo: i post iniziali vengono caricati in localStorage e ogni nuova modifica aggiorna i dati salvati.

Dal punto di vista tecnico, l'app implementa diverse funzionalità chiave:

- layout responsive basato sulla griglia Bootstrap e media queries personalizzate (ad esempio card e icone più piccole su mobile);
- pattern **master/detail** nella sezione grafiche con lista e modale dettaglio;
- form contatti con validazione HTML5 e messaggio di conferma;
- forum con inserimento post/risposte e persistenza locale;
- carrello con contatore live e gestione quantità;
- campo di ricerca nelle grafiche per filtrare i contenuti;
- toast personalizzato in basso a destra per confermare l'aggiunta di un prodotto;
- carosello Bootstrap nella sezione progetti;
- routing SPA per navigare tra le sezioni senza ricaricare la pagina.

Le tecnologie utilizzate comprendono **Vue.js 3** per la struttura SPA e i componenti, **Vue Router** per la navigazione, **Vuex** per la gestione dello stato globale e la persistenza, **Bootstrap 5** per layout e componenti (carosello, form, pulsanti), **Lucide Icons** per le icone nella sezione "Skills Visive" e **localStorage** per la memorizzazione dei dati di forum e carrello.

Istruzioni per l'installazione del progetto

Per eseguire e testare il progetto basta:

- clonare o scaricare la repository: **git clone <https://github.com/saracasarano/portfolio-sara.git>**

- aprire il terminale e scrivere **cd portfolio-sara** e premere invio
- eseguire all'interno della cartella del progetto **npm install** per installare le dipendenze
- avviare il server di sviluppo con **npm run serve**.
- Una volta avviato il server si può aprire il browser all'indirizzo indicato **http://localhost:8080/**

In conclusione, l'applicazione soddisfa i requisiti dell'esame: è una SPA sviluppata con Vue.js, Bootstrap, HTML, CSS e JavaScript, ha un layout responsive con più colonne, utilizza componenti Bootstrap, routing, Vuex e componenti a file singolo, implementa un pattern master/detail tra lista grafiche e modale, offre un form di ricerca e un form di aggiunta contenuti (forum) e garantisce persistenza dei dati tramite localStorage per simulare un database.

Il risultato è un portfolio completo, coerente con il tema scelto e facilmente estendibile in futuro.

Flowchart

