Relazione progetto sito web Pixel Craft

Daisy Romaniello, Giulia Sacco, Pietro Salciarini, Jessica Scano

Concept

Pixelcraft Ecommerce è un portale web realizzato dall'agenzia di comunicazione Pixelcraft, che offre una piattaforma moderna e interattiva con portfolio e articoli dell'agenzia e shop per la vendita online di merchandising personalizzabile.
Il progetto mira a garantire agli utenti un'esperienza fluida, responsiva e coinvolgente con funzionalità avanzate di esplorazione, filtraggio e acquisto prodotti, senza continui ricaricamenti di pagina. Il target principale è costituito da appassionati di design, che apprezzano l'accesso semplice e diretto a prodotti unici.

Fase di progettazione e sviluppo

Il processo di sviluppo ha previsto un'analisi approfondita delle funzionalità chiave del sistema:

- visualizzazione del catalogo prodotti, con possibilità di filtraggio per categoria e ricerca testuale
- visualizzazione dettagliata di ogni prodotto con opzioni di personalizzazione (taglia, colore, design)
- gestione completa del carrello con inserimento, modifica quantità e rimozione articoli

Nell'ottica di offrire un'esperienza utente fluida, tutte le pagine statiche portfolio con due case history, journal con articolo, about e contatti con moduli interattivi, sono state riviste con architettura Vue e ottimizzazione Bootstrap.

Struttura

L'applicazione è una Single Page Application (SPA) sviluppata con Vue.js 3, che utilizza Vue Router per la navigazione e Vuex per la gestione dello stato globale.
La componente principale App.vue funge da layout globale e include AppNavBar, AppFooter e RouterView per il caricamento dinamico dei contenuti.
Gli stati relativi ai prodotti, carrello e opzioni personalizzate, così come le recensioni e i preferiti, sono gestiti tramite Vuex con un file JSON esterno, e vengono preservati nel localStorage via plugin vuex-persistedstate per un'esperienza persistente.
Il progetto integra Bootstrap 5 per il design responsive e FontAwesome. È stata curata la compatibilità CSS/JS tra Vue e Bootstrap per evitare conflitti.

Viste

La **Home** integra i componenti HeroBanner, per il banner principale, ProjectList per mostrare i progetti selezionati, Journal per articoli di approfondimento, e AppNewsletter per l'iscrizione alla newsletter.

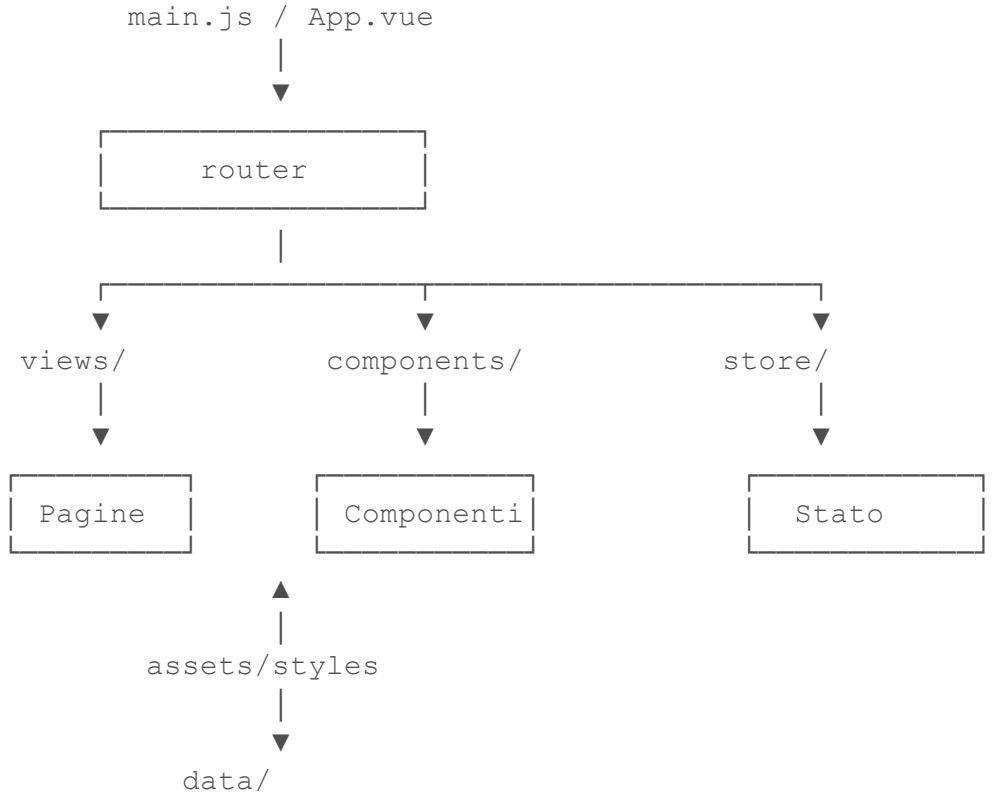
La pagina **About** mostra la componente Hero, i profili del team tramite il componente TeamMemberCard, elencando nome, ruolo, immagine e link ai social per ciascun membro. La vista si completa con il componente AppNewsletter per l'iscrizione alla newsletter, mantenendo il contatto con gli utenti.
La **Portfolio View** mostra la componente Hero con descrizione dei progetti, integra il componente ProjectList per visualizzare una selezione del portfolio, e include il componente AppNewsletter per l'iscrizione alla newsletter.
La vista **CaseIsiFoundation** presenta un case study dettagliato, integra visualizza progetti correlati tramite il componente ProjectList, e il componente AppNewsletter.
La vista **CaseMusicSound** racconta il progetto Open Sound Festival e altri progetti correlati tramite il componente ProjectList, chiudendosi con il componente AppNewsletter per l'iscrizione alla newsletter.
La vista **Journal** presenta la componente Hero con descrizione degli articoli, che sono rappresentati tramite il componente ArticleCard, consentendo una navigazione semplice e modulare. È presente la componente AppNewsletter.
ArticoloMaterieprime mostra un articolo con una sezione con articoli correlati tramite il componente ArticleList. È presente la componente AppNewsletter.
La pagina **Contatti** presenta una componente Hero con immagine e messaggio di benvenuto. Include un modulo di contatto con validazioni, informazioni sulle sedi operative e una mappa interattiva per la sede principale.
La vista **CatalogView** integra i componenti CatalogNavBar per la navigazione con accesso a preferiti e carrello, HeroCatalog come elemento visivo principale, e ProductList per mostrare i nuovi prodotti del mese. Include una sezione per visualizzare tutte le categorie e un banner promozionale. Le funzioni di apertura/chiusura dei drawer per preferiti e carrello sono gestite con metodi Vue.
CatalogView presenta una barra di navigazione tramite CatalogNavBar con funzioni per aprire preferiti e carrello. Mostra il titolo della categoria filtrata attiva e utilizza ProductList per visualizzare i prodotti filtrati in base alla categoria selezionata.
ProductDetailView mostra il dettaglio di un prodotto con immagini principali e anteprime, badge di disponibilità stock con tooltip informativi, prezzo con eventuali sconti, e descrizione. Offre opzioni di personalizzazione specifiche per categoria (dimensioni, cornici, taglie, colori), gestione quantità e pulsanti per aggiungere al carrello o ai preferiti. Include una sezione recensioni con componenti dedicato ReviewsSection.
FavoritesPage mostra una pagina "Preferiti" per un catalogo prodotti, dove si visualizzano i prodotti aggiunti come preferiti dall'utente. Usa Vuex per accedere allo stato globale dei prodotti e degli ID preferiti, filtrando i prodotti preferiti per mostrarli in lista.
CartPage rappresenta la pagina del carrello della spesa in un e-commerce. Mostra i prodotti aggiunti al carrello con quantità modificabili tramite pulsanti "+" e "−", e consente di rimuovere singoli prodotti.

Componenti

AppNavBar: barra di navigazione responsive con icone social e menu di navigazione principale.
AppNewsletter: invita gli utenti a iscriversi alla newsletter tramite un form con campi obbligatori e messaggio di conferma al submit.
AppFooter: footer responsive con dati aziendali e icone social, con contatto supporto via email.
TeamMemberCard: mostra un membro del team con immagine, nome, ruolo e icone sociali cliccabili.
ProjectList: griglia responsive di progetti con immagine, titolo e descrizione in overlay visibile al passaggio del mouse che impiega Intersection Observer per animare il fade-in dei progetti quando entrano nel viewport.
ArticleCard: componente che visualizza sinteticamente un articolo con immagine, titolo, autore, data e categoria, ideato per liste di articoli dinamiche e responsive.
ArticleList: sezione che mostra una griglia responsive di articoli con immagine, titoli cliccabili e categorie in overlay al passaggio del mouse.
CatalogNavBar: barra di navigazione per cataloghi prodotti con barra di ricerca, pulsanti categoria filtrabili, icone interattive per preferiti e carrello con badge numerici dinamici.
CategoriesSection: sezione mostra una lista di categorie prodotto con immagine, etichetta e numero di prodotti per categoria.
HeroBanner: blocco visivo Vue.js che mostra un'immagine di sfondo personalizzabile.
HeroCatalog: sezione Vue.js con layout a griglia con un'immagine principale con didascalia testuale sovrapposta e immagini laterali.
BannerShop: contenitore banner fullscreen che mostra un'immagine centrata e contenuta nel viewport, adattandosi alla larghezza e altezza della finestra.
ProductGallery: visualizzatore di immagini prodotto che mostra un'immagine principale valorizzata e una serie di miniature cliccabili sottostanti per cambiare l'immagine principale.
ProductList: griglia responsive che visualizza una lista di prodotti usando il componente ProductCard. Gestisce eventi di aggiunta al carrello e toggle preferiti propagati dai ProductCard, aggiornando lo stato globale Vuex tramite commit delle relative mutazioni.
ProductCard: card che mostra sinteticamente un prodotto con immagine, nome, prezzo e eventuale prezzo scontato barrato. Include badge dinamici per "New" e sconto percentuale, pulsanti interattivi per aggiungere ai preferiti e carrello, integrati con Vuex per la gestione dello stato preferiti.
ReviewsSection: mostra statistiche riassuntive delle recensioni, ha un form per aggiungere nuove recensioni con validazione dei campi obbligatori e visualizza la lista delle recensioni.
StarRating visualizza una valutazione a stelle, personalizzabile nel numero massimo di stelle, nell'incremento (es. 0.5 o 1), e nei colori attivi/inattivi. Supporta

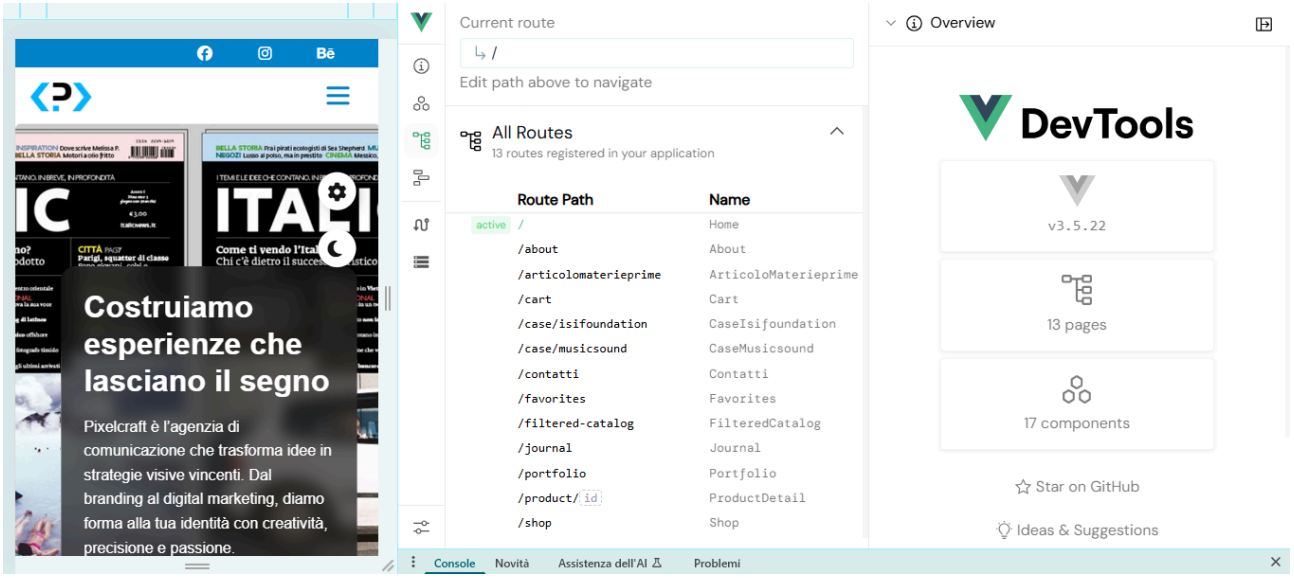
selezione interattiva del voto con evento di aggiornamento o può essere reso in sola lettura.
StyleSwitcher permette all'utente di scegliere un tema colore e di attivare/disattivare la modalità scura del sito.

Flowchart della struttura reale del progetto



- assets/styles: stili CSS e file statici utilizzati globalmente
- components: componenti riutilizzabili (es. navbar, card prodotto)
- data: file con prodotti demo in formato JSON)
- router: gestione delle rotte
- store: stato centralizzato
- views: le pagine principali dell'app

La mappa mostra come App.vue richiama il router che decide quale pagina (view) deve essere mostrata. Le view usano i componenti riutilizzabili. Sia le view che i componenti possono utilizzare dati dalla cartella data/, stili da assets/styles e lo stato condiviso tramite store/.



Flowchart del flusso utente

Il percorso tipico seguito dall'utente dal momento in cui entra nel sito fino alla gestione del carrello è:

- L'utente arriva alla pagina catalogo dove può visualizzare i prodotti disponibili, filtrare per categoria e selezionare articoli di interesse.
- Può aprire sezioni specifiche come i preferiti o il carrello tramite pulsanti dedicati.
- Dalla lista prodotti, l'utente può cliccare su un singolo prodotto per visualizzare la pagina di dettaglio con immagini, descrizioni, opzioni di personalizzazione (taglia, colore, dimensione, ecc.) e selezione quantità.
- Sulla pagina del prodotto può aggiungere l'articolo al carrello o ai preferiti e visualizzare e inserire recensioni.
- Nel carrello, l'utente può modificare la quantità degli articoli o rimuoverli; il sistema calcola i totali aggiornati in tempo reale.
- Il sito prevede anche sezioni informative, portfolio, articoli, e newsletter, che costituiscono contenuti aggiuntivi a supporto dell'esperienza utente.

Comportamenti dinamici e validazioni

- Gestione di quantità e aggiornamento prezzi nel carrello.
- Validazione dei form di contatto (es. email, nome, accettazione privacy) con feedback per l'utente.
- Filtri e reset dei filtri per la ricerca prodotti.
- Supporto per preferiti e carrello persistente tramite Vuex per la gestione dello stato.
- Visualizzazione responsive con adattamenti per mobile e desktop.
- Interazioni di navigazione fluida tra sezioni multiple tramite router Vue.js.

In sintesi, il flusso utente è quello canonico di un sito e-commerce con aggiunta di contenuti editoriali e multimediali per arricchire l'esperienza, gestione dello stato

applicativo con Vuex e uso di componenti personalizzati per navigazione, visualizzazione prodotti, carrello e form di contatto. L'esperienza è completata da messaggi di feedback, animazioni per il caricamento e stili responsivi per garantire usabilità su vari dispositivi.

Flowchart del flusso di interazione interna

Rappresenta come le azioni dell'utente in una view aggiornano lo stato globale, influenzano i componenti e provocano cambi di pagina tramite il router nel progetto Pixelcraft-Ecommerce.

L'utente si trova sulla pagina di dettaglio di un prodotto e decide di compiere un'azione, come aggiungere il prodotto al carrello. Questa azione attiva la view corrispondente, che provvede ad aggiornare lo stato globale dell'applicazione, modificando i dati relativi al carrello. Questo aggiornamento dello store fa sì che i componenti dell'interfaccia, come l'icona del carrello nella barra di navigazione, si modifichino in tempo reale, riflettendo il nuovo stato.

Questi due flowchart, flusso utente e flusso di interazione interna, mostrano rispettivamente la prospettiva dell'utente (cosa fa) e quella tecnica del sistema (come reagisce). Insieme, aiutano a comprendere sia l'esperienza che il funzionamento interno.

Istruzioni per l'esecuzione

Per visualizzare il progetto è necessario effettuare il download al link: https://github.com/gsacco2000/Pixelcraft-Ecommerce.git

I passaggi chiave da effettuare sono poi:

- installare Node.js
- clonare o scaricare il progetto
- aprire terminale e entrare nella cartella del progetto
- eseguire npm install per installare librerie
- lanciare npm run serve
- aprire il browser all'indirizzo locale mostrato