

# **RELAZIONE HOME\_GARDEN\_2025 PER IL CORSO DI PROGETTAZIONE E MANAGEMENT DEL MULTIMEDIA PER LA COMUNICAZIONE – MATEC 2024/25**

di Sara Rodigari

Per il progetto d'esame del corso di Progettazione e Management del Multimedia per la Comunicazione ho realizzato una single page web application dal titolo Home Garden. La piattaforma si presenta come un diario o un'enciclopedia digitale per gli appassionati di fiori e piante. Il sito non è un e-commerce, ma uno spazio che raccoglie diverse piante corredate dai consigli per la loro cura. Gli utenti di Home Garden possono consultare il catalogo digitale e selezionare le specie che possiedono per aggiungerle alla propria sezione personale. Da quest'area è possibile accedere al dettaglio delle singole piante e comprendere meglio le modalità di cura.

La struttura dell'applicazione si basa su un index.html che contiene un div con identificativo "app". Questo viene utilizzato nel file main.js per inizializzare l'applicazione, insieme al router e allo store. All'interno dell'index del router vengono importate le view, cioè le diverse sezioni del sito, con i rispettivi URL e path da visualizzare nella barra di ricerca del browser. Lo store, invece, contiene gli array e i contatori in cui immagazzinare le informazioni, le mutations e le actions per modificare lo stato dell'applicazione. Sempre nello store viene importato il file JSON che funge da database, fornendo i dati alle singole pagine tramite la mutation "SET\_PLANTS", eseguita attraverso l'action "dbPlants".

Il file App.vue definisce le singole pagine tramite il router-view, la barra di navigazione (NavBar.vue) e il footer, che risultano fissi e visibili in ogni sezione. Una volta installato il progetto, la prima pagina che appare è la Home (Home.vue), molto semplice e minimale. La sezione della barra di navigazione (che contiene i link alle varie sezioni e un contatore che indica il numero di piante inserite dall'utente nella propria area personale) e quella del footer sono uguali e costanti in ogni componente, poiché definiti e importati all'interno del file App.vue. Nella sezione del template della Home.vue, invece, sono presenti, oltre al titolo e al sottotitolo, i link alla sezione della Serra (PlantCatalog.vue) e del Giardino (GardenView.vue per accedere all'area personale).

Il file PlantCatalog.vue, che appare nel menù e nella Home sotto il nome di "Serra", permette agli utenti di consultare le piante già catalogate e le relative informazioni di cura. Le piante sono visualizzate tramite la computed property filterPlants, che restituisce l'elenco delle piante dell'arrayPlants (estratto dal file JSON). Una barra di ricerca in alto consente all'utente di cercare una specie in particolare, grazie alla searchQuery definita nello script, che applica un filtro a filterPlants. Successivamente appaiono le singole card contenenti una foto, il nome, la tipologia e il tasto "Aggiungi al tuo giardino" per ogni pianta. Ogni card è descritta nel componente PlantCards.vue, con cui comunica tramite un attribute binding (:plantProps="plant"). Questo componente, quando cliccato, rimanda alla pagina di dettaglio della pianta selezionata grazie al suo id, passato al router-link definito all'inizio del codice. Nel momento in cui l'utente clicca su "Aggiungi al tuo giardino" appare un alert che conferma l'avvenuta aggiunta. Il bottone "Ok" dell'alert cambia colore al passaggio del mouse grazie a uno style binding. L'aggiunta all'area personale utilizza il metodo addToGarden, che richiama l'action addGarden dello store, la quale tramite una mutation inserisce la pianta nell'array garden e aggiorna il contatore userGarden, incrementandolo di uno. In questa mutation viene anche definito un sistema che definisce se una specie viene inserita più volte e di conseguenza il numero di esemplari presenti nel giardino.

Nella Serra, quando l'utente clicca su una specifica card, accede alla ProductPage.vue tramite l'id, come descritto sopra. All'interno della pagina sono presenti diversi componenti:

- Carousel.vue: contiene il carosello che fa da sfondo al titolo e comunica con la view genitore tramite il parametro plantView, che recupera i dati dal database e li assegna alla props plantDetail.
- Title.vue: include il nome della specie e la sua tipologia, sempre tramite attribute binding.
- PlantDescription.vue: mostra alcune informazioni e curiosità sulla pianta, ricavate da plantView e passate come props plantDetail.
- PlantData.vue: include le informazioni sulla fascia di prezzo, la frequenza di innaffiatura, l'umidità richiesta e l'esposizione solare, cioè i dati necessari per prendersi cura della pianta. Prezzo, umidità ed esposizione vengono inseriti come valori numerici, ma sono visualizzati con etichette tramite conditional rendering (es. "Molto alta", "Media", "Bassa").
- AddToGarden.vue: contiene il testo che spiega come aggiungere la specie alla propria area personale, il bottone per farlo e gli eventi collegati. Il metodo addToGarden comunica con lo store tramite l'action addGarden, che a sua volta richiama la mutation ADD\_TO\_GARDEN per aggiungere la pianta all'array garden e incrementare di uno il contatore userGarden e contare quanti esemplari della stessa specie vengono aggiunti. Al click del bottone viene attivato anche l'evento showCustomAlert, che mostra un messaggio di conferma e ne gestisce la chiusura tramite closeAlert. In questo caso la view genitore passa il parametro plantView con selectedPlant come props, ricavato tramite attribute binding.
- SuggestionsBox.vue: riceve come parametro l'id di plantView, passato con il nome plantId, poiché i suggerimenti per la cura sono legati a una specifica pianta. Il box contiene un titolo, la sezione per la visualizzazione dei commenti lasciati dagli utenti e il messaggio predefinito "Ancora nessun consiglio dalla community". Tramite list rendering, i commenti inseriti vengono visualizzati a destra del form, mantenendo uno stile coerente. Per lasciare un suggerimento viene utilizzato un ulteriore componente: SuggestionForm.vue. Attraverso il form, l'utente può inserire il proprio nome, il livello di esperienza nella coltivazione e un racconto correlato. All'invio, il metodo sendSuggestion richiama l'action addSuggestion e la mutation ADD\_SUGGESTION: se non esistono recensioni, l'array suggestions rimane vuoto; altrimenti, con un push, viene aggiunto il nuovo commento in riferimento al plantId selezionato.

Infine sul fondo della pagina è presente un bottone per tornare alla sezione della serra.

La view successiva, AddPlant.vue, permette all'utente di aggiungere una nuova specie al catalogo, se non presente, grazie al binding bidirezionale offerto da Vue. Tramite il componente AddPlantForm.vue l'utente può compilare un form con il nome, la tipologia, la descrizione, l'umidità richiesta (selezionata da un menù a discesa), la frequenza di innaffiatura, l'esposizione solare (attivando o meno un checkbox) e la fascia di prezzo (sempre da menù a discesa). Al submit del form, le informazioni inserite vanno a comporre un nuovo elemento dell'arrayPlants, corredato di card e immagine predefinita. Al click del bottone "Aggiungi" compare anche un alert che conferma l'avvenuta aggiunta.

L'ultima view, GardenView.vue, si occupa di visualizzare l'area personale dell'utente, denominata "Giardino". Nel caso in cui l'utente non ha inserito alcuna pianta nello spazio personale, verrà visualizzato solo il titolo e la descrizione della pagina. Nel caso in cui l'utente avesse inserito delle specie nel proprio giardino, oltre alla descrizione viene mostrato il componente UserGarden.vue, che visualizza l'elemento plant e il suo index in riferimento all'array garden dello store. All'interno del componente appare una scheda con foto e nome della pianta aggiunta al giardino. La card è cliccabile e rimanda alla pagina con i dettagli della pianta. Sotto l'anteprima appare il contatore che definisce quanti esemplari di quella specie sono stati aggiunti e il bottone "Rimuovi", che consente di eliminare la specie dallo spazio personale, o di diminuirne la quantità. L'evento remove viene gestito nella pagina genitore: la mutation REMOVE\_FROM\_GARDEN modifica lo stato dell'array garden e il contatore userGarden, decrementandolo di uno ed escludendo la pianta appena rimossa.

Le modifiche relative allo stile e alla resa responsive dell'applicazione sono state realizzate con Bootstrap e con un foglio CSS importato in main.js.