

Progetto di esame

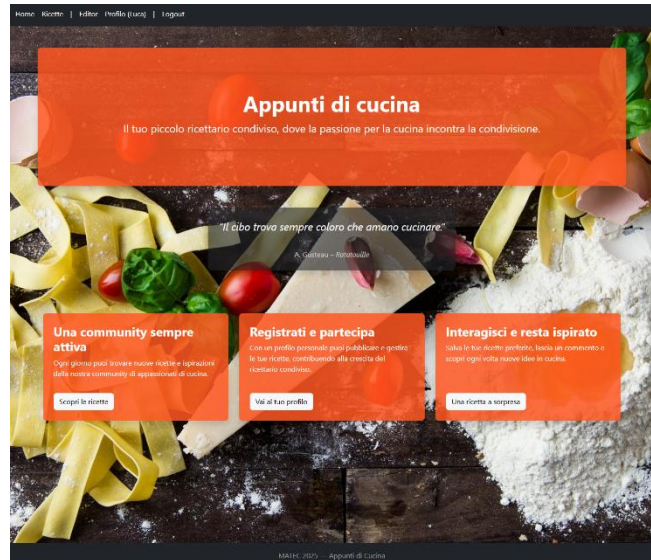
Appunti di cucina - piccolo ricettario condiviso

Concept

Il progetto consiste in un portale web per la gestione e la consultazione di ricette culinarie, basato su un modello di contenuto generato dagli utenti.

La piattaforma offre un archivio di ricette liberamente accessibile, che può essere esplorato e organizzato tramite funzionalità di ricerca e filtri specifici per tipo di piatto. Per ogni ricetta è disponibile una scheda di dettaglio completa.

Gli utenti registrati beneficiano di funzionalità avanzate, tra cui la possibilità di salvare le ricette preferite, interagire con la community tramite commenti e, in qualità di autori, di gestire in autonomia le proprie creazioni, con la facoltà di pubblicare, modificare ed eliminare le ricette.



Home Page

Struttura

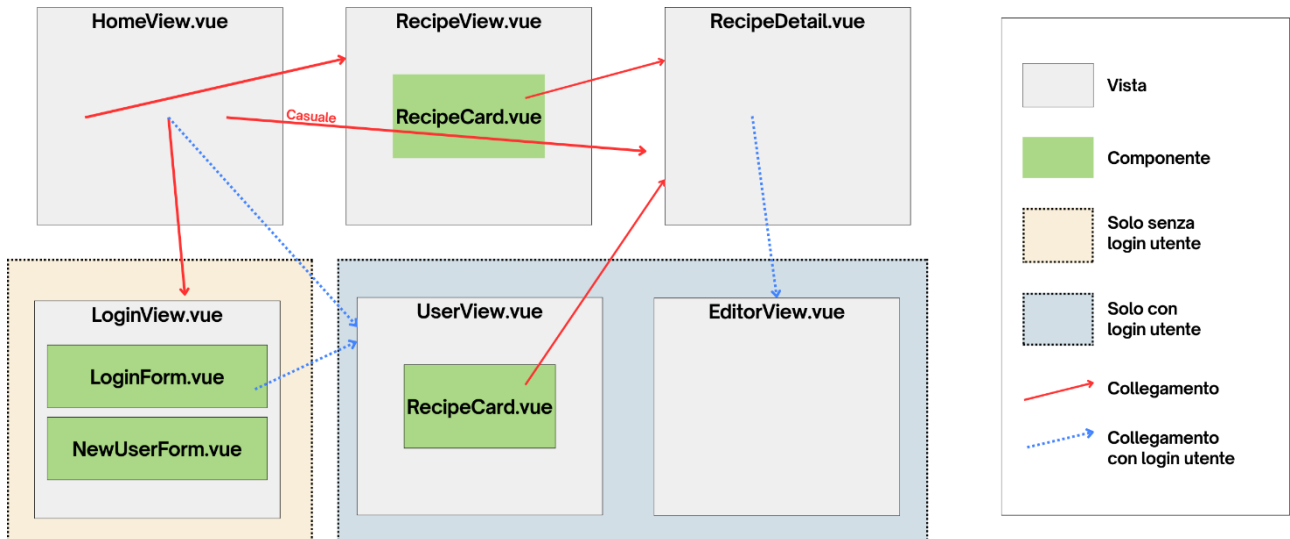
La struttura del sito, sviluppata con *Vue.js*, è modulare e organizzata per garantire una gestione chiara e dinamica delle diverse sezioni.

L'architettura del sito si basa su un'unica vista principale, `'App.vue'`, che funge da layout globale. Questa componente contiene tre elementi strutturali fondamentali:

| | |
|------------------------------|--|
| <NavBar /> | <ul style="list-style-type: none">• Header: Un'intestazione che ospita la componente della barra di navigazione, fornendo un accesso coerente e immediato alle principali sezioni del sito da qualsiasi pagina. |
| <router-view /> | <ul style="list-style-type: none">• Sezione Dinamica: Tra l'header e il footer, la sezione <code><router-view /></code> agisce come un "segnaposto". Il contenuto di questa sezione viene caricato dinamicamente dal <i>Vue Router</i>, che in base all'URL della pagina (la "route") seleziona e visualizza il componente corrispondente (ad esempio, la vista della lista delle ricette, la pagina di dettaglio di una singola ricetta o la pagina di login). |
| <FooterApp /> | <ul style="list-style-type: none">• Footer: Un'area a piè di pagina che rimane costante in tutte le viste. |

Questa configurazione, che sfrutta il concetto di *Single-Page Application (SPA)*, consente di aggiornare solo la parte centrale della pagina senza ricaricare l'intera struttura, migliorando la fluidità e l'esperienza d'uso.

Il sito si articola in 6 viste principali, che a loro volta possono implementare delle componenti modulari:



HomeView è una semplice landing page con una immagine di sfondo ed il titolo del sito. Da qui è possibile accedere direttamente all'elenco delle ricette, alla pagina di login o del profilo (se già loggati), o ad una ricetta scelta in modo casuale.

RecipeView ospita l'elenco completo delle ricette. Ognuna di esse è rappresentata da una componente modulare, **RecipeCard.vue** che ne riporta una descrizione sintetica.

Nella vista è presente anche un piccolo form che permette un filtro delle ricette per tipo di piatto (Antipasto, Primo, Secondo o Dolce) ed una ricerca per nome. Il binding avviene tramite l'uso di v-model.

Cliccando su una scheda si raggiunge la vista con tutti i dettagli delle ricette, **RecipeDetail**.

Home Ricette | Editor Profilo (Luca) | Logout

Home Ricette | Accedi

Dettaglio della barra di navigazione con utente loggato e non loggato.

Il sito offre delle funzioni disponibili esclusivamente per utenti registrati e loggati. In caso di nessun utente loggato, dalla barra di navigazione è possibile accedere alla vista **LoginView**, dove è possibile accedere con email e password, o in alternativa registrarsi come nuovo utente.

Nella vista **RecipeDetail**, per gli utenti registrati è possibile segnare una ricetta come preferita, e l'informazione apparirà anche sulla scheda di sintesi. Inoltre è possibile inserire dei commenti alle ricette.

La vista **EditorView** è accessibile solo agli utenti loggati, e permette di inserire nuove ricette. La stessa pagina è utilizzata per modificare una ricetta dell'utente. Per far questo è possibile accedere all'editor con un pulsante apposito nella pagina della ricetta.

La vista **UserView**, disponibile solo per utenti loggati, rappresenta la pagina del profilo dell'utente. In questa pagina sono visualizzate le schede delle ricette inserite dall'utente e delle ricette preferite. Da ogni scheda si può accedere alla pagina dei dettagli. Se si accede ad una propria ricetta, è possibile modificarla o eliminarla. L'eliminazione di una ricetta cancella anche i relativi commenti.

Nota: ai fini di test, sono presenti i seguenti utenti registrati

| Username | Email | password |
|----------|--|--------------|
| Luca | luca@mail.com | luca_123 |
| Marta | marta@mail.com | marta_123 |
| Giovanni | giovanni@mail.com | giovanni_123 |
| Anna | anna@mail.com | anna_123 |
| Marco | marco@mail.com | marco_123 |

Dettagli tecnici

Stile

Per lo stile del sito è stato utilizzato Bootstrap, con una personalizzazione del colore primario tramite il file `styles/custom.scss`.

A questo scopo sono stati aggiunti i moduli `"sass"` e `"sass-loader"`¹

Dati e backend

Lo store del progetto è gestito tramite il modulo `"vuex"`²

Il backend è simulato dal modulo `"json-server"`³ che simula una REST API fornendo le *route* per l'accesso ai dati salvati nel file `data.json`.

Le chiamate lato client sono effettuate tramite il modulo `"axios"`, che permette richieste di tipo `post`, `get`, `patch` e `delete` per creare, leggere, modificare o eliminare (CRUD) i dati dal backend.

Limiti dovuti all'uso di json-server

Non essendo stata sviluppata una API dedicata, la parte relativa al login è stata realizzata in modo totalmente non sicuro, di fatto con una semplice chiamata `get` che restituisce i dati dell'utente anziché un token di autorizzazione. Sarebbe possibile configurare del middleware per `json-server` o usare servizi di autenticazione di terze parti tipo Auth0⁴.

¹ <https://cli.vuejs.org/guide/css.html#pre-processors>

² <https://vuex.vuejs.org/>

³ <https://github.com/typicode/json-server>

⁴ <https://developer.auth0.com/resources/guides/spa/vue/basic-authentication>

Relativamente all'editor di ricette, senza un backend dedicato non è possibile caricare direttamente le immagini, ma solo inserire il nome del file. Diventa necessario inserire manualmente le immagini nella cartella `public/images`.

In alternativa sarebbe necessario sviluppare una piccola API dedicata, per esempio tramite Node/Express con middleware Multer⁵.

Altri moduli

Poiché è necessario far partire da terminale sia la componente del progetto (`npm run serve`) sia l'elemento di backend (`npm run back`, definito in `package.json`), è stato installato un il modulo *npm-run-all* ⁶(v2) che permette l'esecuzione in parallelo di più script dallo stesso terminale.

Si può quindi avviare l'intero progetto da terminale con

```
npm run test
```

Struttura di **data.json**:

```
{
  "recipes": [
    {
      "id":
      "nome": "
      "ingredienti": [ ]
      "descrizioneBreve":
      "procedimento":
      "immagine":
      "categoria":
      "authorId":
    },
  ],
  "comments": [
    {
      "id":
      "userId":
      "testo":
      "recipeId":
    },
  ],
  "users": [
    {
      "id":
      "email":
      "password":
      "username":
      "preferiti": [ ]
    },
  ],
}
```

Installazione

Per download e installazione fare riferimento alle istruzioni nel repository GitHub:

<https://github.com/Pi3rres/progettoVue>

⁵ <https://expressjs.com/en/resources/middleware/multer.html>

⁶ <https://github.com/bcomnes/npm-run-all2>