Event Hub

Applicazioni e Servizi Web

Elena Morelli - 000928594 {elena.morelli3@studio.unibo.it}

12 Settembre 2022

Indice

Capitolo 1

Introduzione

L'elaborato consiste nella realizzazione di un'applicazione che abbia lo scopo di fornire un servizio di gestione di eventi, sagre e spettacoli presenti nel territorio emiliano-romagnolo. La piattaforma prende il nome di EventHub e consente agli utenti di visualizzare appunto eventi e gestirne gli interessi e le prenotazioni. L'applicativo si compone di una homepage che consente una visione libera a tutti e di un'area personale accessibile solo dopo aver effettuato il login.

Capitolo 2

Requisiti

Il sistema consente a qualsiasi utente di accedere gli eventi presenti e futuri. Permette di filtrarli per:

- Parole chiavi presenti all'interno del nome e della descrizione
- Provincia
- Tipologia dell'evento.

L'utente ha inoltre la possibilità di creare un proprio account e, una volta eseguito l'accesso, di effettuare prenotazioni o esprimere un interesse particolare rispetto ad un evento. I risultati delle preferenze espresse saranno poi visibili all'interno dell'area personale.

L'autenticazione consentirà anche di ricevere notifiche, rispetto agli eventi salvati, generate automaticamente rispetto ad una variazione dello stato attuale.

Per un utente amministratore il servizio web autorizza l'inserimento di nuovi eventi, oltre alla possibilità di modificare o cancellare quelli di cui risulta essere il proprietario.

Capitolo 3

Design

In questo capitolo verrà illustrata la realizzazione dell'applicativo, mostrando lo schema della base di dati e il confronto tra i mockup con le effettive interfacce utente.

3.1 Architettura del sistema

L'architettura del sistema prevede una suddivisione del progetto in Backend e Frontend. Il Backend organizza le API necessarie al Frontend per visualizzare i dati mediante le rotte. A ciascuna di esse vengono assegnati una serie di metodi che, una volta interfacciati con il database documentale, eseguiranno le query necessarie. Ogni documento ha un proprio modello di riferimento e un proprio controller, entrambi contenuti nelle apposite cartelle.

Per quanto riguarda il Frontend, il progetto è stato costruito in maniera tale che ciascuna pagina, raggiungibile dal Router, sia rappresentata da un componente. Quest'ultimo consente di visualizzare correttamente l'interfaccia, oltre a facilitare l'interazione con l'utente e il passaggio di dati con altre componenti.

3.2 Interfacce utente

Il design del Frontend è stato dapprima abbozzato mediante mockup, accessibili tramite la release del progetto. Tutte Le pagine contengono la barra di navigazione, figura 3.1, che permette la visualizzazione del menù notifiche, oltre alla navigazione dalla home all'area personale e viceversa. In questo modo l'utente ha sempre accesso a tutte le pagine principali con un solo click.

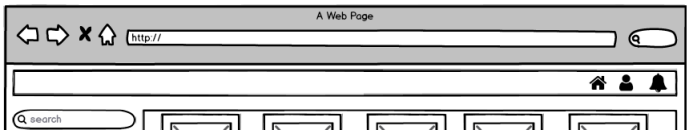

Figura 3.1: Mockup barra di navigazione

Nell'applicazione è presente inoltre una barra laterale, detta sidebar, che si occupa delle diverse modalità di filtraggio degli eventi, così da permettere una ricerca mirata ai propri interessi. Ogni evento viene visualizzato tramite una *card*, la quale una volta cliccata, mostrerà nel dettaglio tutte le informazioni, consentendo di effettuare un'eventuale prenotazione. Facendo "Click" sulla stella, l'evento verrà salvato all'interno dell'area personale, con la possibilità di ricevere notifiche a riguardo.

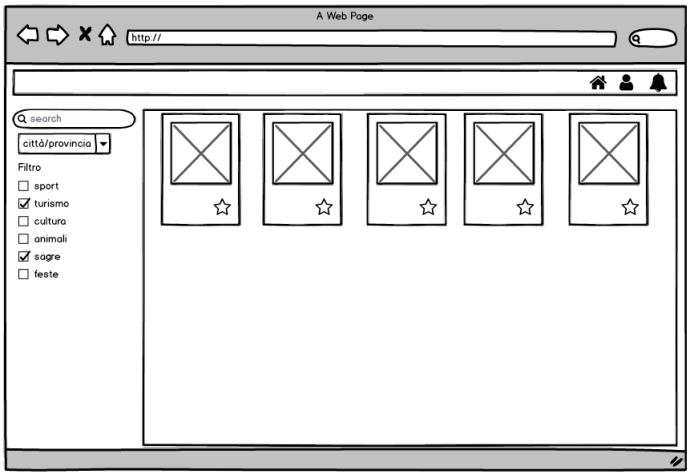

Figura 3.2: Mockup pagina iniziale

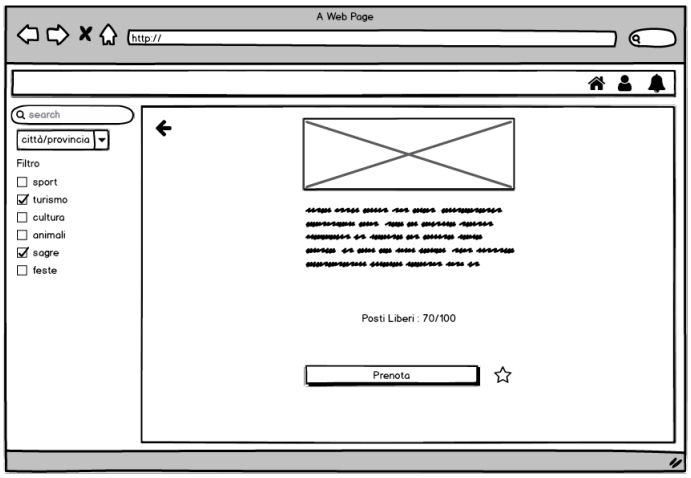

Figura 3.3: Mockup dettaglio evento

Entrando nella propria area personale, l'utente sarà in grado di visualizzare tutte le informazioni che lo riguardano e di modificarle a piacimento, oltre a gestire gli eventi d'interesse.

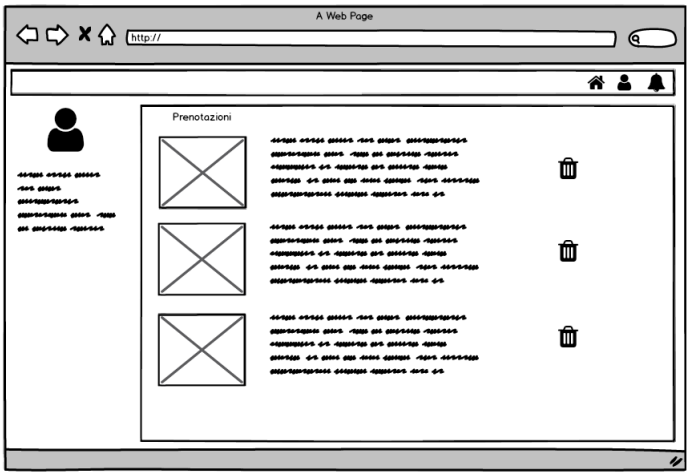

Figura 3.4: Mockup area personale

Come mostra la figura 3.5, l'utente con privilegi d'amministratore avrà invece la possibilità di modificare, cancellare o creare nuovi eventi.

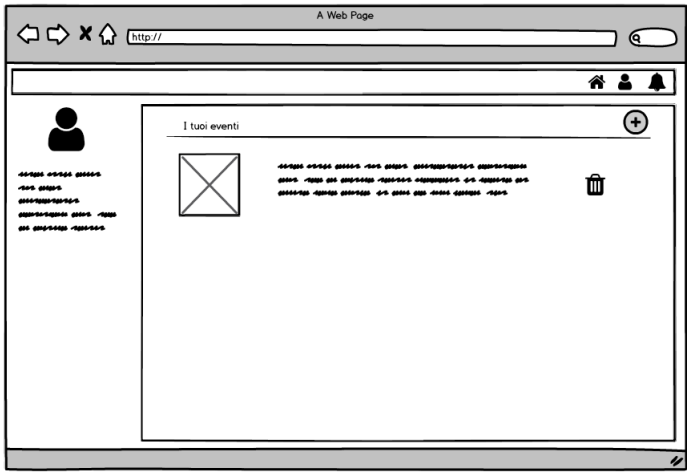

Figura 3.5: Mockup area amministratore

A differenza di quanto visto nel mockup, come mostra la figura 3.6, è stato preferito non mettere la stella all'interno delle card, per non appesantire ulteriormente la vista, ma renderla visibile solamente all'interno della schermata di dettaglio.

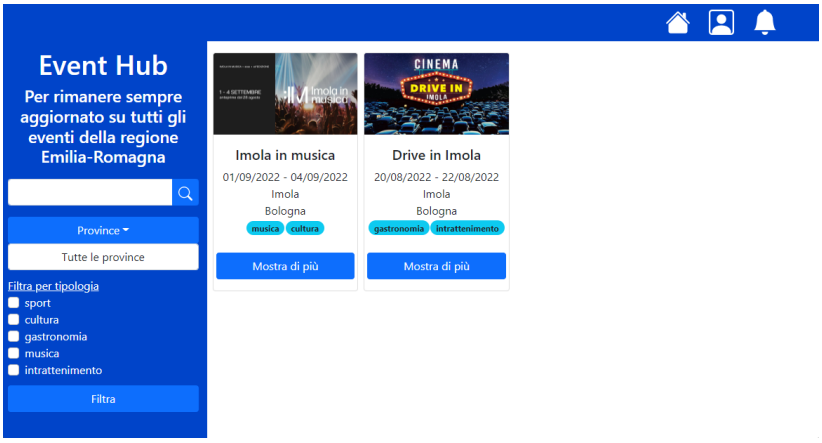

Figura 3.6: Schermata homepage

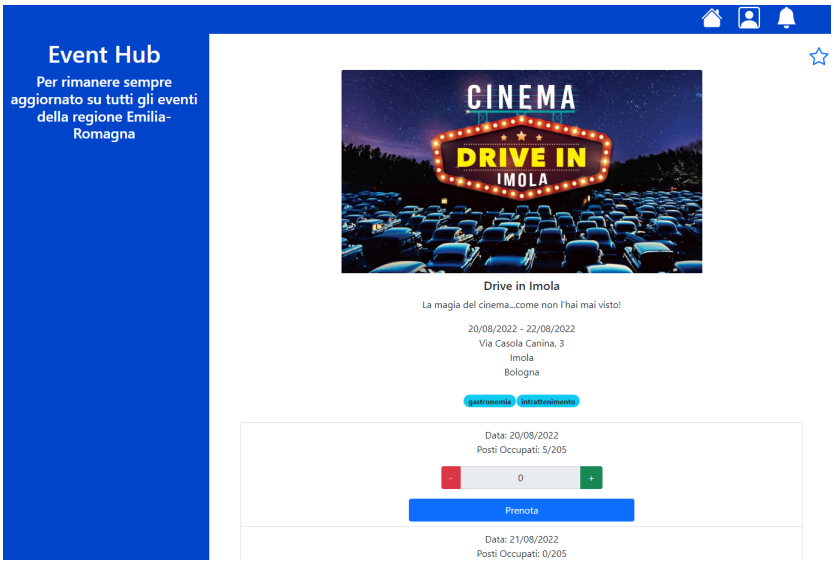

Figura 3.7: Schermata dettagli evento

Nella seguente figura 3.7 si è deciso di nascondere i filtri nella barra laterale, non necessari all'utente ad interagire con la pagina.

Per l'area personale, si è optato per una schermata separata con relativo menu a lato. Scelta valida anche per la schermata dell'utente amministratore.

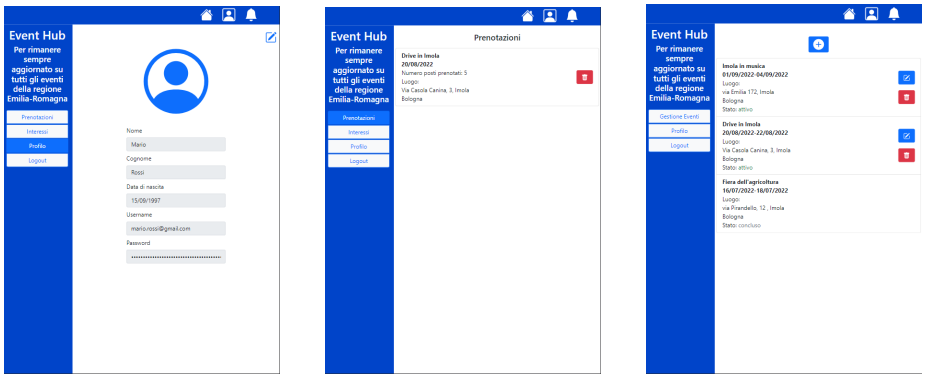

Figura 3.8: Schermate area personale

3.3 Database

Il modello del database è raffigurato in figura 3.9. L'applicazione può avere un numero variabile di eventi. Ad ognuno di essi sono associate le proprie notifiche,

i propri follower (utenti che hanno espresso un interesse verso l'evento), con i relativi tag e provincia. Un singolo evento racchiude tutte le informazioni che lo riguardano, oltre ad un un array (booking) contenente i posti disponibili per ogni giornata. Ogni utente si distingue in amministratore o utente standard e presenta due array, uno con le prenotazioni effettuate e uno con gli eventi d'interesse.

In caso di modifica o cancellazione di un evento viene creata una notifica per ogni follower.

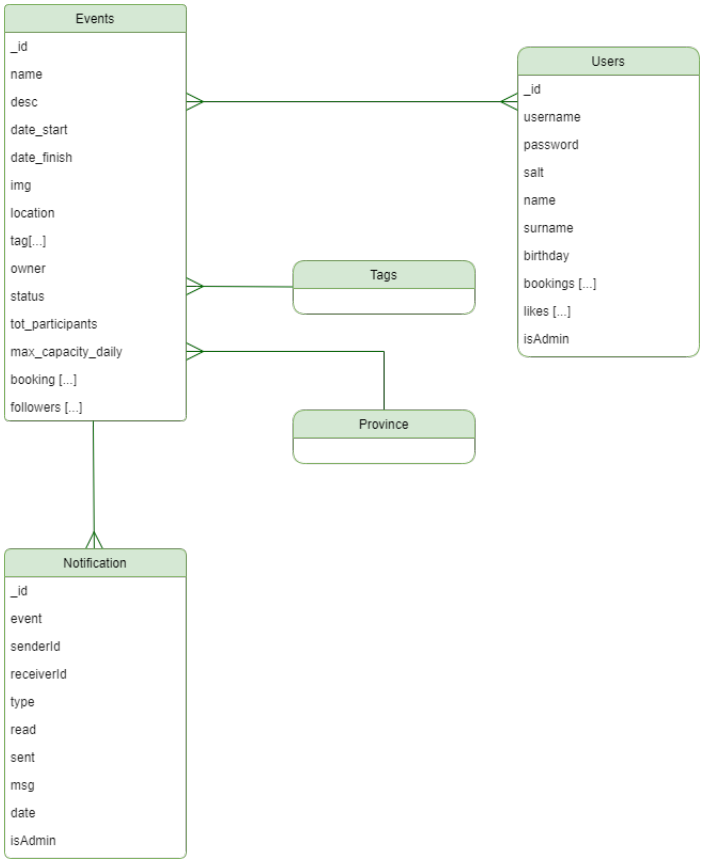

Figura 3.9: Schema del database

Capitolo 4

Tecnologie

L'applicativo opera sfruttando lo stack MERN, variante dello stack MEAN, nel quale il servizio web utilizza MongoDB come database, Express come framework server-side di supporto per Node, React come framework client-side e Node come web server.

Il motivo principale per cui è stato scelto React è un interesse personale rispetto all'apprendimento di questa libreria, oltre al fatto di avere una buona curva di apprendimento e di essere improntato sulle UI.

Per effettuare le richieste HTTP, sia per il server che per il servizio web esterno da cui vengono ottenuti i dati, viene utilizzato Axios. Dal punto di vista dello stile è stato ampiamente utilizzato il framework *Bootstrap*, insieme alla libreria react-icons e react-bootstrap. Quest'ultimo ha semplificato notevolmente la grafica delle pagine web realizzate, poichè permette di modificare determinate variabili per personalizzare il risultato finale.

Per poter gestire le notifiche push in tempo reale si è utilizzato la libreria Socket IO, sia lato client che lato server.

Nel login e nella registrazione si è usufruito di *bcrypt*: una funzione di hashing per salvare le password in maniera sicura nel database; la stessa viene utilizzata al momento della registrazione di un utente, per il salvataggio della password nel db (viene salvato il suo hash con il relativo valore di sale), e al login quando bisogna confrontare la password ricevuta dall'utente con quella salvata.

Jest e Supertest sono stati impiegati nei test di backend.

Capitolo 5

Test

Nella parte di backend sono stati sviluppati alcuni JavaScript Unit test, per verificare il corretto funzionamento delle API.

Terminata la realizzazione del progetto, il tutto è stato testato in modalità Cognitive Walkthrough con i seguenti task:

- Visualizzazione e comprensione della propria dashboard
  - Comprensione:
    - Filtro tramite barra di ricerca
    - Filtro tramite tag
    - Filtro tramite provincia
- Login
- Visualizzazione e comprensione pagina di dettaglio
  - Comprensione:
    - Effettuazione prenotazione
    - Espressione d'interesse
- Visualizzazione e comprensione area personale
  - Comprensione:
    - Individuazione prenotazioni
    - Individuazione interessi
    - Individuazione dati personali
  - Eliminazione prenotazione/interesse
- Ricevuta notifica a seguito di una cancellazione o modifica
- Visualizzazione notifica con segnalazione di lettura e cancellazione
- Logout

- Visualizzazione e comprensione dashboard per amministratori
  - Individuazione lista eventi
  - Individuazione azioni di creazione, modifica ed eliminazione
- Creazione di un nuovo utente

Per ultimo, l'applicazione è stata fatta testare ad utenti che non hanno seguito lo sviluppo. Dai loro feedback si è potuto capire come alcuni aspetti dell'interfaccia non fossero chiari e non aiutassero gli stessi a comprendere appieno come utilizzarla, ad esempio: nella barra di ricerca non risultava chiaro come eliminare una ricerca già conclusa, quindi è stato aggiunto un pulsante per l'apposita cancellazione del campo. Stessa cosa nella ricerca per tag, anche lì è stato aggiunto un nuovo pulsante per "azzerare" le scelte. Inoltre, seguendo l'utente durante l'utilizzo della piattaforma, è stato possibile individuare alcuni errori di programmazione, sfuggiti precedentemente. La registrazione dell'utente e la creazione di un evento andavano a buon fine anche se erano presenti degli errori nell'inserimento dei dati.

Capitolo 6

Deployment

Per installare l'applicativo è necessario aver già installato Npm e MongoDB.
Una volta scaricato il progetto al seguente link:

`https://github.com/MorelliElena/AWS-EventManager`

è' necessario eseguire i seguenti passi:

1. Aprire una console nella cartella del progetto e digitare `npm install`
2. Importare tutte le collections presenti all'interno della cartella *./mongodb* su MongoDB attraverso il comando:

   ```
   mongoimport --db EventsDB --collection nomeFile nomeFile.json --jsonArray
   ```

3. Spostarsi all'interno della cartella *./server*. Eseguire il comando `node index.js` per avviare il server
4. Aprire un'altra console all'interno della cartella *./client* del progetto e digitare `npm install`
5. In quest'ultima lanciare l'applicazione con il comando `npm start` e aprire il proprio browser all'indirizzo `http://localhost:3000/`.

Il progetto, di base, parte configurato con tre account:

- email: `mario.rossi@gmail.com` password: `mariorossi`
- email: `luca.bianchi@gmail.com` password: `lucabianchi`
- email: `andrea.verdi@gmail.com` password: `andreaverdi` (amministratore)

Capitolo 7

Conclusioni

Questo progetto mi ha dato la possibilità di approfondire le mie conoscenze del mondo Web, sopratutto quelle riguardanti l'utilizzo dello stack MERN, che, fino ad ora, non avevo mai utilizzato. L'apprendimento di React non mi è risultato subito facile, ha richiesto più tempo di quello che avevo previsto. Sono comunque soddisfatta di aver impiegato questa libreria, sicuramente perché mi sarà utile in futuro in ambito lavorativo.