Lucia Ferrari

Relazione Laboratorio di Industria 4.0

Requisiti

Sviluppare un'applicazione Web che permetta ad utenti ed organizzatori di gestire la prenotazione ad eventi.
L'applicazione dovrà implementare le seguenti feature:

- Registrazione di utenti
- Registrazione di organizzatori
- Permettere ad organizzatori di creare eventi caratterizzati da:
  - Nome
  - Orario Inizio (anche data)
  - Orario Fine (anche data)
  - Luogo evento
  - Posizione geografica
  - Massimo numero di partecipanti

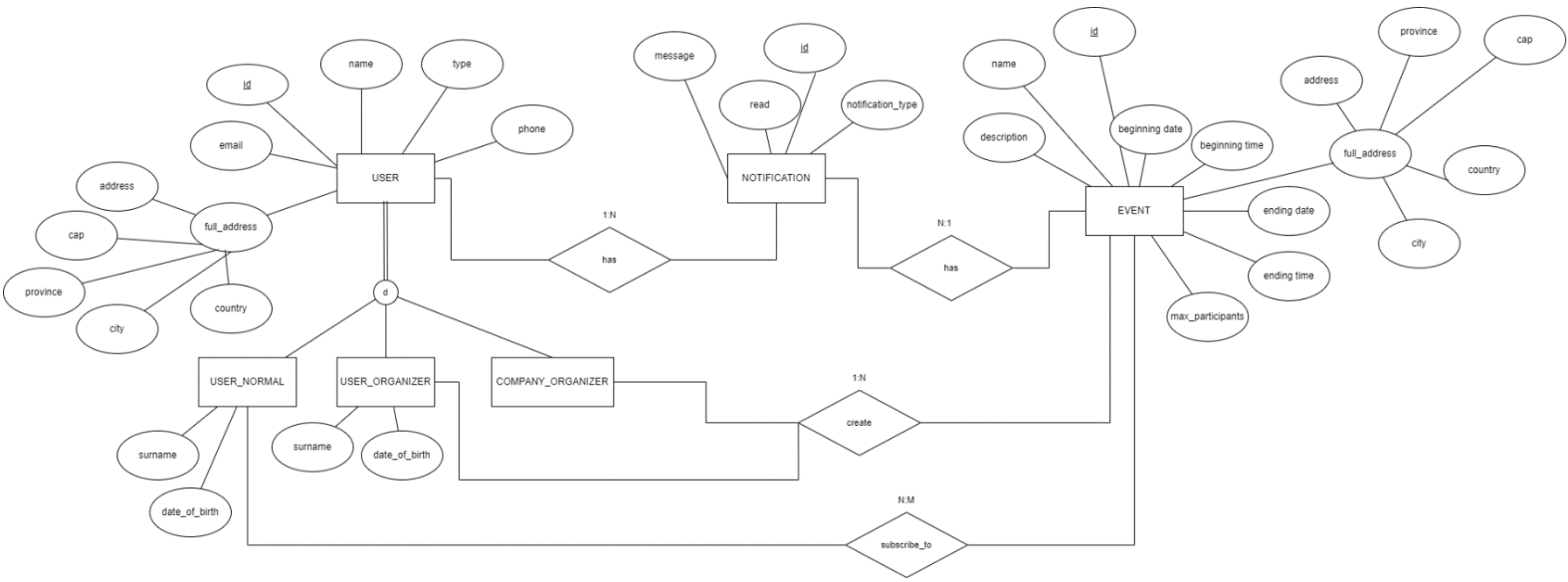
- Gli utenti invece saranno in grado di
  - Ricercare eventi (anche filtrando per posizione geografica)
- Registrarsi e deregistrarsi ad eventi
- Impedire ad utenti di registrarsi ad un evento se concomitante ad un altro ai quali sono già iscritti.
- Permettere agli organizzatori di consultare la lista dei partecipanti ai propri eventi ed eventualmente rimuovere utenti iscritti indesiderati dai propri eventi

Inoltre, l'applicazione dovrà implementare un sistema di notifiche per permettere ad utenti registrati ad un evento se esso ha avuto variazioni, se esso viene annullato dall'organizzatore, o se essi sono stati rimossi. Allo stesso modo, gli organizzatori dovranno essere notificati in caso di vari scenari legati ai loro eventi (es massima capienza raggiunta).

Realizzazione del progetto

Tutto il codice del seguente progetto è disponibile su Github:
https://github.com/dream-19/Ruby_On_Rails_Web_App

Base dati – Modello EER



La base dati è stata poi implementata su MySql, ed è stata creata automaticamente tramite la creazioni di migrazioni su Ruby on Rails.

Le tabelle create sono quindi le seguenti, con campi e relativi tipi:

- Users
  - id
  - name varchar(255)
  - surname varchar(255)
  - type varchar(255)
  - email varchar(255), unique
  - phone varchar(255)
  - date_of_birth date
  - address varchar(255)
  - cap varchar(255)
  - province varchar(255)
  - city varchar(255)
  - country varchar(255)
  - created_at datetime(6)
  - updated_at datetime(6)

In aggiunta a questi campi sono poi aggiunti quelli relativi al salvataggio delle password, gestiti automaticamente dalla gemma devise (encrypted_password varchar(255), reset_password_token varchar(255) , reset_password_sent_at datetime(6), remember_created_at datetime(6))

- Events
  - id
  - name varchar(255)
  - description text
  - beginning_time time
  - beginning_date date
  - ending_time time
  - ending_date date
  - max_participants int
  - address varchar(255)
  - cap varchar(255)
  - province varchar(255)
  - country varchar(255)
  - city varchar(255)
  - user_id bigint
  - created_at datetime(6)
  - updated_at datetime(6)

- Subscriptions
  - id
  - user_id bigint
  - event_id bigint
  - created_at datetime(6)
  - updated_at datetime(6)

- Notifications
  - id
  - user_id bigint
  - event_id bigint
  - message text
  - read tinyint(1)
  - notification_type varchar(255)
  - created_at datetime(6)
  - updated_at datetime(6)

Eliminazioni in cascata

Le eliminazioni sono quasi sempre gestite in cascata. L'eliminazione di un utente scatena l'eliminazione di tutte le sue iscrizioni ad eventi o di tutti i suoi eventi creati (che a loro volta eliminano anche tutte le iscrizioni ricevute).
Le notifiche vengono eliminate quando si elimina l'utente ad esso associate, mentre nel caso dell'eliminazione di un evento vengono comunque mantenute ed il campo event_id è posto a null.

Implementazione tramite Ruby On Rails

La Web App è stata realizzata tramite Ruby on Rails (versione: 7.1.3.2), per la grafica invece è stato utilizzato Bootstrap 5.
L'applicazione si basa su quattro modelli: user, event, subscription e notification.

Gestione degli utenti (users)

- **Suddivisione degli utenti**

Gli utenti sono gestiti tramite STI (Single Table Inheritance), sono infatti salvati in un'unica tabella nel database ma sono suddivisi in tre possibili sottoclassi:

- USER_NORMAL: rappresenta un utente normale, si può iscrivere/disiscrivere da eventi ma non può crearli. Per lui i campi 'phone' e tutti quelli relativi all'indirizzo sono opzionali.
- USER_ORGANIZER: rappresenta un utente organizzatore, può creare/eliminare eventi ma non si può iscrivere a nulla. Rispetto all'utente normale il campo 'phone' diventa obbligatorio.
- COMPANY_ORGANIZER: rappresenta un'azienda organizzatrice di eventi, come l'utente organizzatore gli è permesso creare/eliminare eventi ma non può iscriversi a nulla. Rispetto all'utente normale il campo 'phone' diventa obbligatorio. A differenza delle due precedenti classi non sono presenti i campi 'date_of_birth' e 'surname'.

La differenziazione tra le varie sottoclassi è gestita dal campo 'type' che ha i tre possibili valori: USER_NORMAL, USER_ORGANIZER, COMPANY_ORGANIZER

- **Validazione degli utenti**

Per gestire la diversa validazione esistono quindi 3 sotto-classi create a partire dal modello generale User. Ogni campo del modello utente è validato, controllando se deve essere presente (o se non deve essere presente), il suo tipo e la relativa lunghezza massima.

Esempio di validazione:

```
validates :name, :surname, :email, :phone, :address, :cap, :province, :city, :country,
length: { maximum: 255, too_long: "must be at most %{count} characters" }
validates :email, uniqueness: true, format: { with: URI::MailTo::EMAIL_REGEXP, message:
"must be a valid email address" }
```

Il campo email è inoltre impostato come campo unico, non possono quindi esistere più utenti con la stessa email.

- **Autenticazione degli utenti**

L'autenticazione degli utenti (login, logout, registrazione, gestione delle password) è realizzato tramite l'utilizzo della gemma Devise, che permette anche di mantenere una sessione attiva. Alla tabella dell'utente è quindi aggiunto il campo 'encrypted_password', che salva la password dell'utente tramite algoritmo Bcrypt.

- **Interfacce accessibili agli utenti**

Tutti gli utenti hanno accesso alla stessa pagina per il login mentre la registrazione si differenzia in tre possibili form, in base al tipo di utente. A livello applicativo questo è realizzato tramite più partials, che permettono il riuso dei campi uguali tra i vari form senza doverli ridefinire ogni volta.
A seguito del login gli utenti normali vengono rimandati alla homepage (da cui possono iscriversi ad eventi), mentre gli utenti organizzatori sono mandati all'elenco dei loro eventi. Gli utenti normali hanno a disposizione anche una pagina che elenca gli eventi a cui sono iscritti (eventi correnti, passati, futuri).

Figure 1: pagine di elenco delle iscrizioni per un utente normale

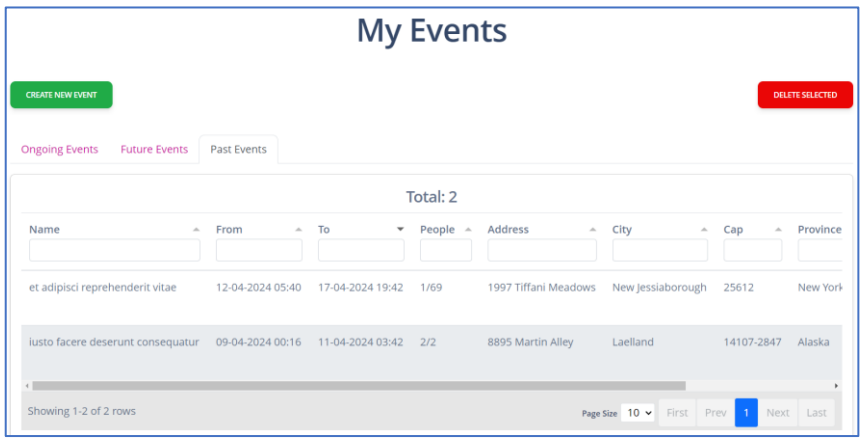


Figure 2: pagina che elenca gli eventi creati, per gli utenti organizzatori

Tutti gli utenti hanno poi a disposizione una pagina per modificare le proprie informazioni (da cui è anche possibile eliminare il proprio account) e una pagina di riepilogo delle notifiche ricevute.
Se un utente organizzatore decide di eliminare il proprio account anche gli eventi da lui creati saranno eliminati, lo stesso vale per gli utenti normali con le iscrizioni agli eventi.

Gestione degli eventi (events)

Gli eventi possono essere creati solamente da utenti organizzatori. Gli eventi sono modificabili fino a che non risultano passati. A quel punto non possono più essere né editati né cancellati e gli utenti non possono più iscriversi.

- **Validazione eventi**

Gli eventi presentano una validazione molto stretta. Tutti i campi devono essere obbligatoriamente presenti (ad esclusione della descrizione e delle foto, che possono essere opzionali).
Inoltre:

- Le date devono essere sempre nel formato yyyy-mm-dd, agli utenti vengono presentate poi come dd/mm/yyyy
- Gli orari sono invece nella forma hh:mm o hh:mm:ss
- Non è possibile creare eventi nel passato
- Non è possibile creare eventi con un data di inizio successiva a quella di fine (o nel caso siano eventi giornalieri è presente il controllo dell'orario di fine e di inizio)

- **Aggiunta di immagini agli eventi**

Agli eventi è possibile associare fino ad un massimo di 3 immagini (formati possibili: jpg, png, gif, webp). Le immagini sono gestite localmente tramite Active Storage di Rails, che si occupa autonomamente della loro gestione. Gli eventi senza immagini associate mostrano comunque un'immagine di default nella homepage e nella loro pagina di presentazione.

- **Interfacce per gli eventi**

Gli eventi sono il punto principale dell'applicazione, di conseguenza sono visualizzati in più modi.

- Nella homepage è presente l'elenco di tutti gli eventi correnti e futuri, con un menù che permette di filtrarli e ordinarli in base alle loro principali caratteristiche. Gli eventi inoltre sono mostrati tramite una paginazione (non più di 18 per pagina), realizzata tramite l'uso della gemma 'kaminari'

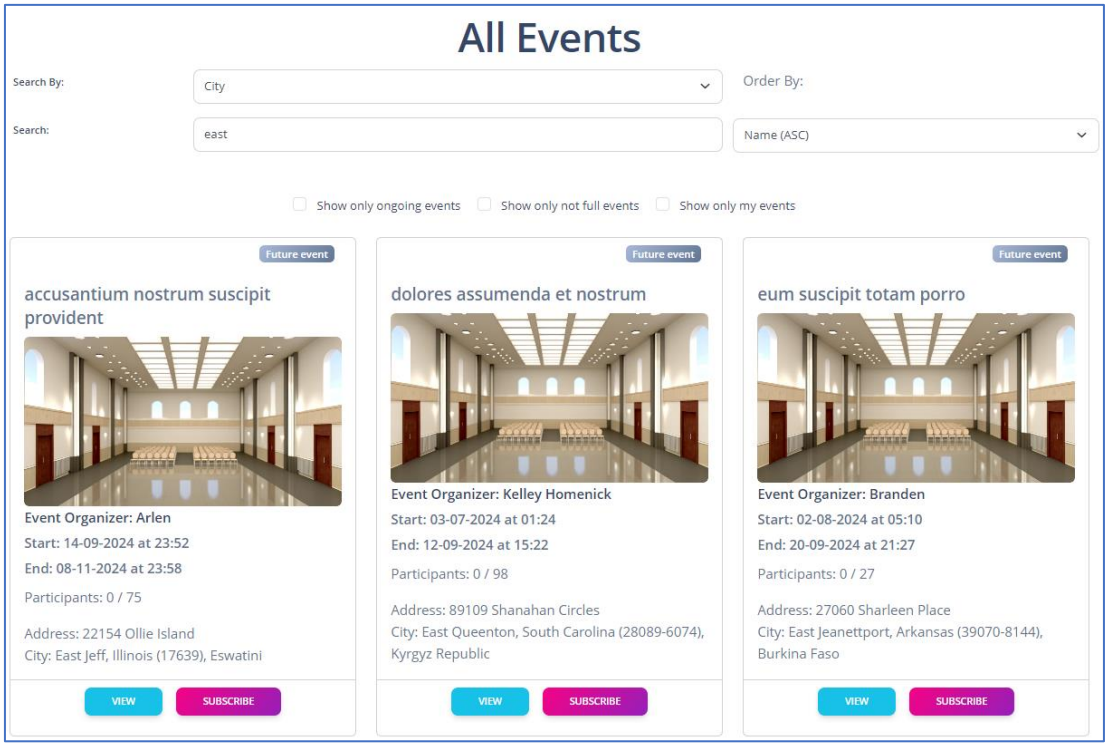


Figure 3: homepage

- Per ogni evento è possibile accedere alla sua pagina, con tutte le informazioni presenti. Se la pagina è visualizzata dal proprietario dell'evento è presente anche l'elenco degli utenti iscritti (con la possibilità di rimuoverli).

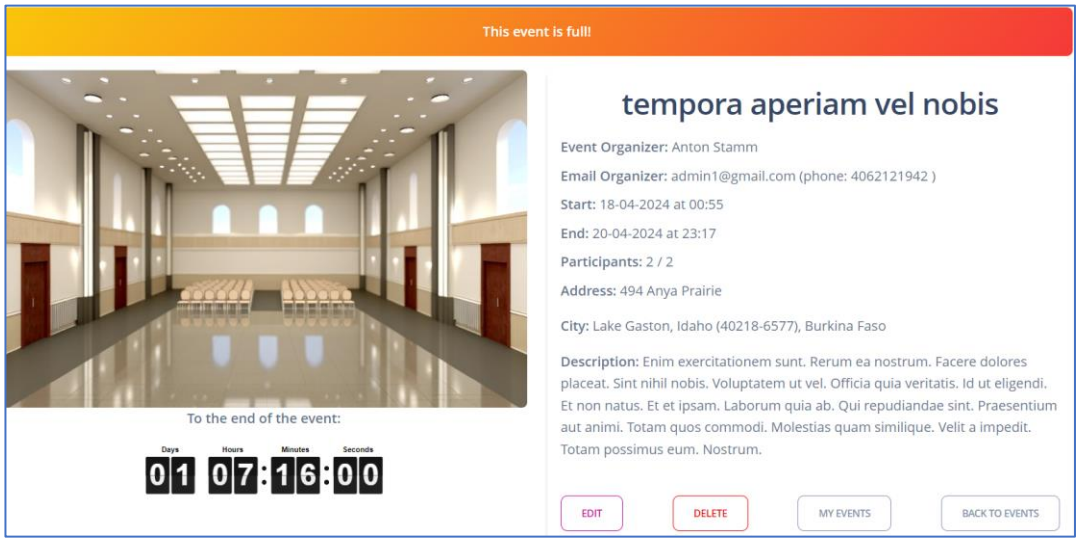


Figure 4: dettagli di un evento

- Ogni evento ha poi il suo form di creazione e relativa modifica, da cui è anche possibile aggiungere/rimuovere foto.

Gestione delle iscrizioni (subscriptions)

Le iscrizioni ad eventi sono effettuate seguendo determinate condizioni:

- L'utente deve essere un utente normale (quindi non un organizzatore)
- L'evento deve essere in corso o futuro (non è possibile iscriversi ad eventi già passati)
- Non è possibile iscriversi due volte ad uno stesso evento
- Non è possibile iscriversi ad un evento che ha già raggiunto il massimo numero di partecipanti
- Non è possibile iscriversi ad eventi concomitanti ad altri eventi a cui si è iscritti. Per quest'ultimo caso vengono controllate quattro condizioni:
    - Se la data d'inizio del nuovo evento a cui ci si vuole iscrivere è interna al periodo di svolgimento di qualche altro evento a cui si è iscritti
    - Se la data di fine del nuovo evento a cui ci si vuole iscrivere è interna al periodo di svolgimento di qualche altro evento a cui si è iscritti
    - Se il periodo di svolgimento del nuovo evento a cui ci si vuole iscrivere è interno al periodo di svolgimento di qualche altro evento a cui si è iscritti
    - Se il periodo di svolgimento del nuovo evento a cui ci si vuole iscrivere include il periodo di svolgimento di qualche altro evento a cui si è iscritti

Gli utenti possono poi eliminare le proprie iscrizioni ad eventi (a meno che questi non siano passati), allo stesso modo gli organizzatori di eventi possono rimuovere le iscrizioni non gradite.

Gestione delle notifiche (notifications)

Le notifiche vengono usate come storico delle azioni eseguite e sono associate ad un utente ed un evento (l'evento può anche essere nullo, ad esempio nel caso in cui venga eliminato).

Vengono generate automaticamente in base alle seguenti azioni:

1. Utente si iscrive ad un evento: la notifica viene mandata sia all'utente stesso che al proprietario dell'evento
2. Utente si disiscrive da un evento: la notifica viene mandata sia all'utente stesso che al proprietario dell'evento

3. Utente normale elimina il suo account: viene inviata una notifica di disiscrizione da tutti gli eventi a cui era iscritto, per gli organizzatori degli eventi
4. Organizzatore rimuove un utente da un evento: la notifica viene mandata sia all'organizzatore stesso che all'utente rimosso
5. Organizzatore crea un evento: creazione di una notifica per l'organizzatore stesso, con il nome dell'evento
6. Organizzatore aggiorna un evento: creazione di una notifica per l'organizzatore e tutti gli iscritti all'evento. La notifica riepiloga anche tutte le modifiche fatte
7. Organizzatore elimina un evento: creazione di una notifica per l'organizzatore stesso e per tutti gli utenti che erano iscritti. Questa notifica viene generata anche nel caso in cui un organizzatore elimini il proprio account (con conseguente eliminazione in cascata di tutti i suoi eventi)
8. Evento raggiunge la massima capacità: creazione di una notifica per avvertire l'organizzatore dell'evento

Dal punto di vista implementativo le notifiche sono gestite da un Service.

- **Interfaccia delle notifiche**

Le notifiche sono accessibili in due diversi punti: direttamente dal menù in una visione compatta oppure in una pagina che le contiene tutte. In entrambi i casi è possibile marcarle tutte come lette con un solo click.

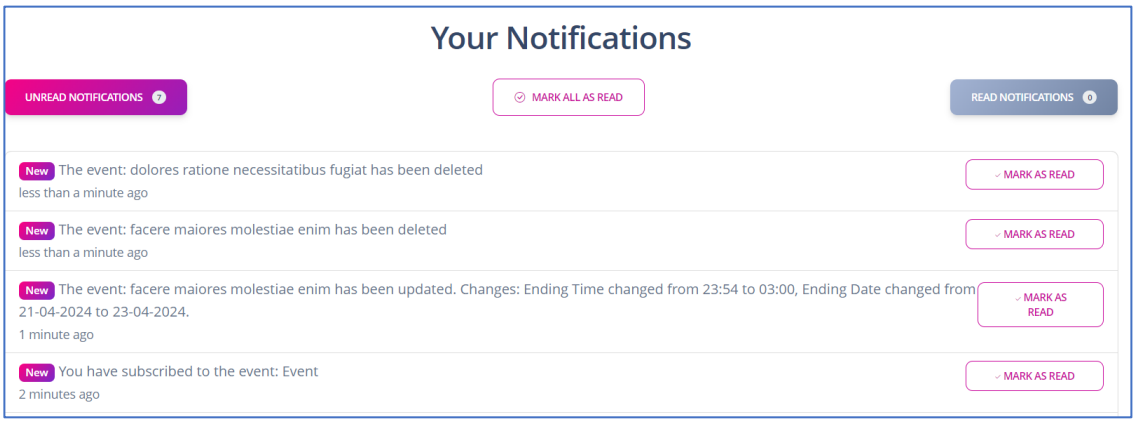


Figure 5: pagina di gestione delle notifiche

Altre caratteristiche presenti

- **Modali**: per tutte le azioni ritenute pericolose, principalmente le eliminazioni, sono presenti dei modali che richiedono la conferma definitiva dell'utente per poter procedere.

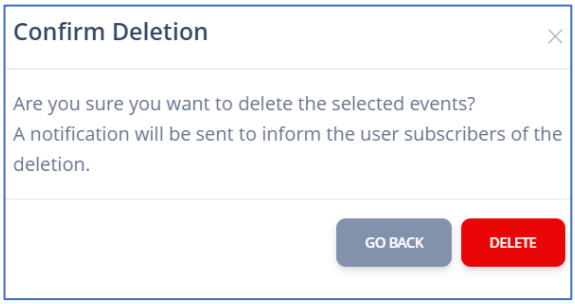


Figure 6: esempio di modale

- **Autocompletamento per stati e città**: nei form in cui sono richiesti degli indirizzi è presente l'autocompletamento per il campo dello stato e della città, inoltre una volta inseriti, cap e provincia sono calcolati automaticamente. A livello implementativo questo è realizzato tramite l'utilizzo dell'api Geonames, servizio gratuito che consente di eseguire fino a 1000 query ogni ora.

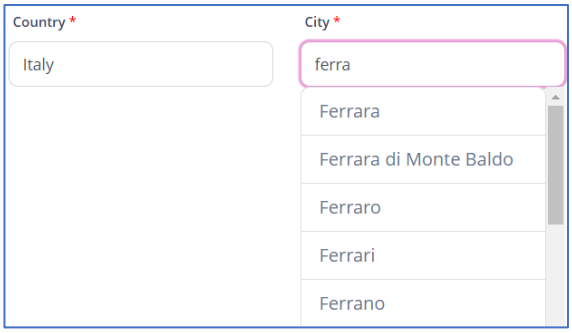


Figure 7: esempio di autocompletamento per le città

- **Gestione delle tabelle**: tutte le tabelle presenti, ovvero quelle di riepilogo degli eventi passati, futuri e correnti a cui si è iscritti o che sono stati creati, sono gestite tramite il plugin Tabulator. Questo permette di ordinare la tabella in base ai campi che contiene e permette la ricerca di valori interni alla tabella. Inoltre gli eventi sono paginati in pagine di 25 eventi ciascuna.

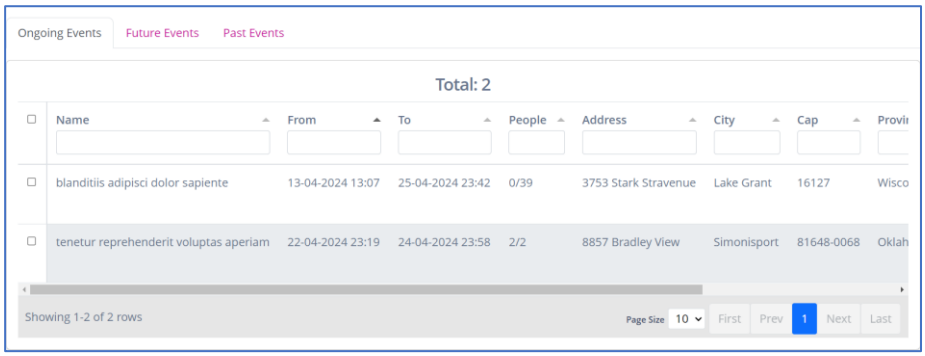


Figure 8: esempio di tabella con Tabulator

- **Gestione delle richieste asincrone**: tutte le azioni che non necessitano del ricaricamento dell'intera pagina e che preferiscono mantenerla responsive durante una richiesta sono gestite in modo asincrono tramite *fetch* di javascript.
  Un esempio di queste azioni sono: eliminazione di una foto da un evento, tutte le richieste fatte all'api geonames ed eliminazione di multipli eventi.

Conclusioni

Durante la realizzazione di questo progetto è stato possibile mettere in pratica le conoscenze teoriche apprese durante il corso di studio, affrontando sfide reali tipiche dello sviluppo software in contesti professionali.
Il risultato finale è un'applicazione web di semplice uso per la gestione di eventi, che non solo facilita l'organizzazione degli stessi ma assicura anche un alto livello di sicurezza per tutte le operazioni eseguite dagli utenti.
In conclusione, questo progetto ha raggiunto gli obiettivi proposti ed ha aperto la strada a future implementazioni e ulteriori miglioramenti.