

GoCalendar

Applicazioni e Servizi Web

Luigi Incarnato - 0001076053 {luigi.incarnato@studio.unibo.it}

06 Novembre 2023

Contents

1	Introduzione	3
2	Requisiti	4
3	Design	5
3.1	Analisi target user	5
3.2	Mockup	6
4	Tecnologie	14
4.1	Frontend	14
4.2	Backend	14
4.3	Ulteriori Tecnologie	15
4.4	Linguaggi utilizzati	15
5	Codice	16
6	Test	16
7	Deployment	16

1 Introduzione

GoCalendar è una web app nata dallo scopo di organizzare in modo migliore il proprio tempo, ma gestendo comunque la funzione di calendario per poter avere comunque le funzioni basilari. Un utente può registrarsi tramite mail, con un suo nickname e una password conforme alla sicurezza con, cioè, almeno una lettera minuscola, una maiuscola, numeri e caratteri speciali. Il nucleo dell'app è la possibilità di creare proprio eventi e oggetti collegati ad essi di default, tramite cui assegnare un inizio, una fine per poterli successivamente visualizzare all'interno di un grafico. Il grafico per questo ha una visualizzazione mese per mese degli eventi a cui è stato dato un "lasso di tempo", inoltre saranno visualizzati solo le etichette a cui si è deciso di voler assegnare la possibilità di comparire all'interno della visualizzazione dei dati. L'utente potrà modificare in qualsiasi momento il suo nickname, la sua password e le impostazioni riguardanti al calendario, più precisamente di dare la possibilità all'app di ricevere notifiche degli eventi "scaduti" oppure la visualizzazione del tempo in Ore, Minuti o Secondi all'interno del grafico.

2 Requisiti

Descrizione delle caratteristiche e funzionalità che il sistema prevede.

Le principali requisiti funzionali richieste dall'app sono:

- **Registrazione e gestione del proprio utente:** possibilità di creare ed eventualmente andare a modificare un certo utente. Modificando anche le importazioni del proprio calendario.
- **Creazione e gestione di eventi:** possibilità di creare e modificare a proprio piacimento i propri eventi personali, assegnandogli l'etichetta che si vuole, modificando l'ora di inizio per poi andare ad inserire l'ora di fine per la visualizzazione del grafico.
- **Creazione e gestione di etichette:** possibilità di creare e modificare etichette da assegnare ad un certo evento creato, con l'ulteriore possibilità di modificare a proprio piacimento il colore, il titolo e la visualizzazione all'interno del grafico.
- **Creazione e gestione di oggetti:** creazione e modifica di oggetti collegati a particolari etichette di hobby, presenti di default, tra cui lettura, serie tv e film, modificandone il titolo o l'immagine.
- **Upload di file e link da collegare ad un evento:** possibilità di fare l'upload di un qualsiasi tipo di file all'evento con la possibilità di modificarlo in seguito oppure scaricarlo. Inserimento di un link tramite cui si potrà andare alla pagina corrispondente.
- **Notifiche:** visualizzazione di notizie pop-up dell'applicativo rappresentanti un evento scaduto da più di 24 ore.

I requisiti non funzionali sono:

- **Navigazione dell'applicazione:** navigazione user friendly per qualsiasi tipologia di utente.
- **Meccaniche di sicurezza:** garantire la sicurezza del proprio account tramite un controllo attivo sulle password e la creazione di token obbligatori per la navigazione.
- **Rendere il sistema estendibile:** dare la possibilità al sistema di poter essere esteso o modificato successivamente senza problemi.

3 Design

3.1 Analisi target user

Personas: Didato Donato

Donato è un uomo di 45 anni, sposato e padre di due bambini, che ha comprato un nuovo cellulare da poco a cui stranamente manca un'applicazione che gli permetta di organizzare la propria agenda, i propri impegni e quelli dei suoi familiari. Tutto questo possibilmente presente anche come pagina web e sempre disponibile anche da computer fisso.

- Donato entra nella web app per controllare giorno per giorno le sue attività in modo da poter organizzarsi con gli eventuali impegni dei figli.
- Donato cerca sempre di ottimizzare il suo tempo, quindi fa molto spesso uso del grafico per visualizzare il tempo trascorso per ciascuna attività.
- Donato accede per organizzare meglio il tempo da poter dedicare ai suoi hobby e alle uscite con gli amici.

Personas: Gianburrasca Maria

Maria è studentessa lavoratrice di 25 anni che ha iniziato ad utilizzare questa web app tramite consiglio di un amico, per poter gestire meglio il poco tempo a disposizione che gli resta per studiare, per i suoi hobby e la palestra.

- Maria controlla spesso quando tempo ha dedicato alle singole lezioni di fitness che ha fatto in palestra.
- Maria accede per controllare quanto tempo ha trascorso rispetto al mese precedente allo studio di una certa materia per metterlo a confronto con il suo andamento giornaliero.
- Maria effettua spesso il login per visualizzare le notiche per compilare il tempo trascorso di un evento, per poter vedere il tempo utilizzato in una certa giornata.

Personas: Delcaso Tiziana

Tiziana è una giovane manager di una start up, non riesce mai ad organizzare settimanalmente il suo lavoro, tra cui appuntamenti con clienti in sede e fuori sede, avendo sempre a disposizione il link della pagina principale dell'azienda e il file di presentazione che dovrà far vedere al cliente.

- Tiziana utilizza molto spesso la funzione di upload dei file e il salvataggio del

link, in modo tale da avere dati in modo immediato

- Tiziana in quanto manager deve organizzare i tempi da dedicare ai singoli clienti, dipendenti ed eventuali stagisti.

3.2 Mockup

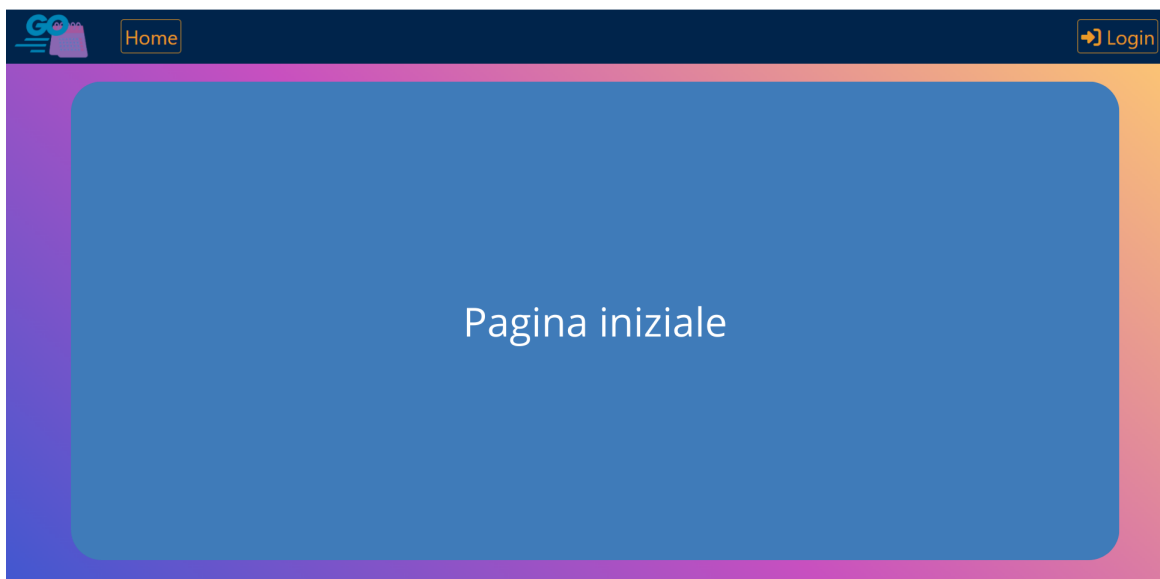
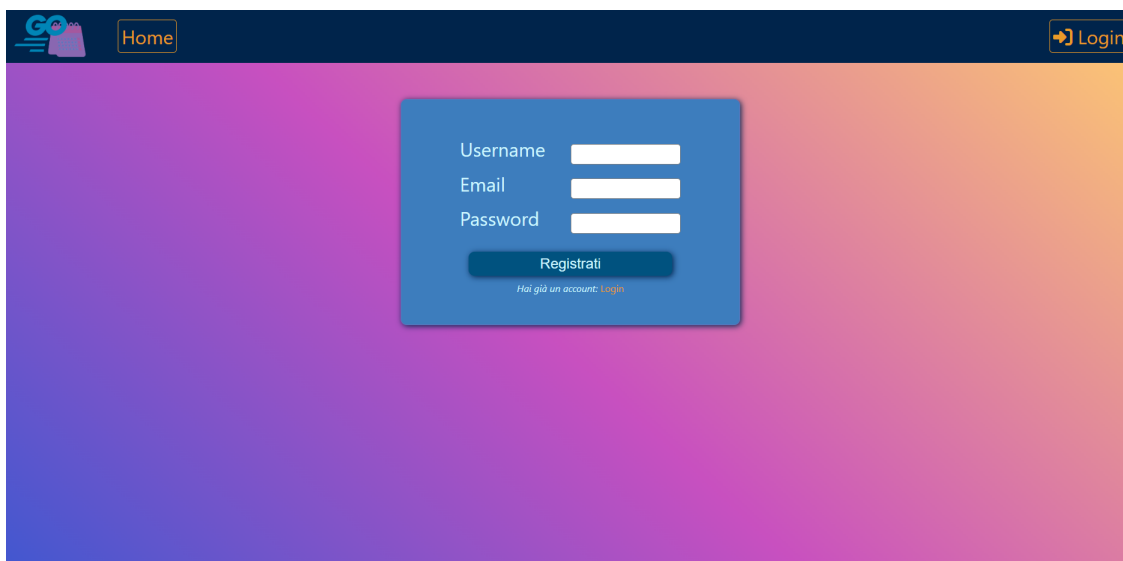


Figure 1: Home page



The image shows the registration page of the GoCalendar application. The page has a dark blue header with the GoCalendar logo on the left, a 'Home' button in the center, and a 'Login' button on the right. The main content area has a vibrant purple-to-orange gradient background. In the center, there is a blue registration form with three input fields for 'Username', 'Email', and 'Password'. Below these fields is a dark blue 'Registrati' button. Underneath the button, there is a link that says 'Hai già un account: [Login](#)'.

Figure 2: Page di registrazione

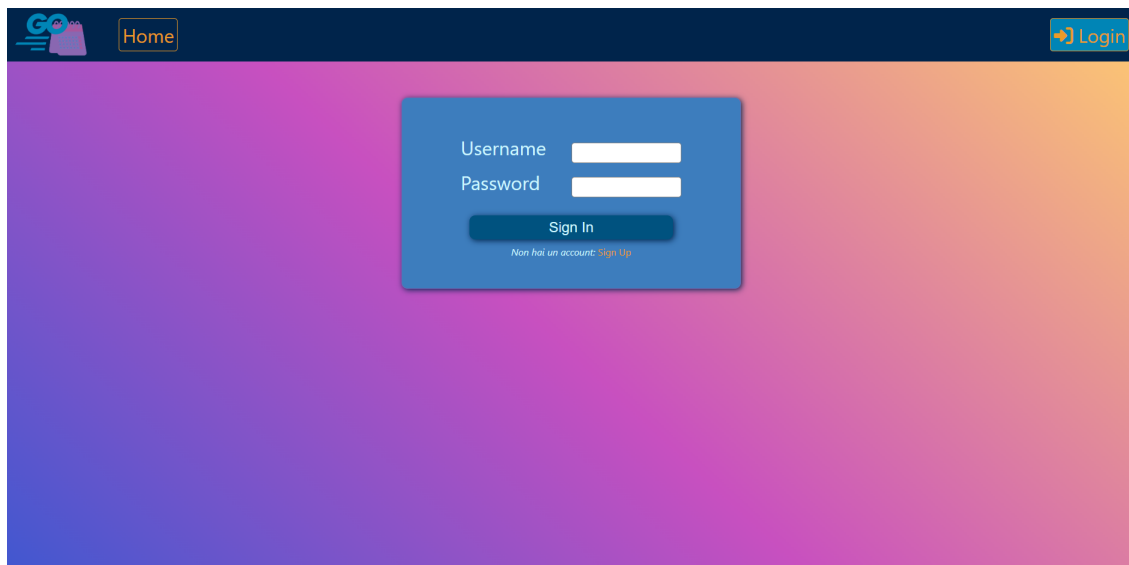


Figure 3: Page di login

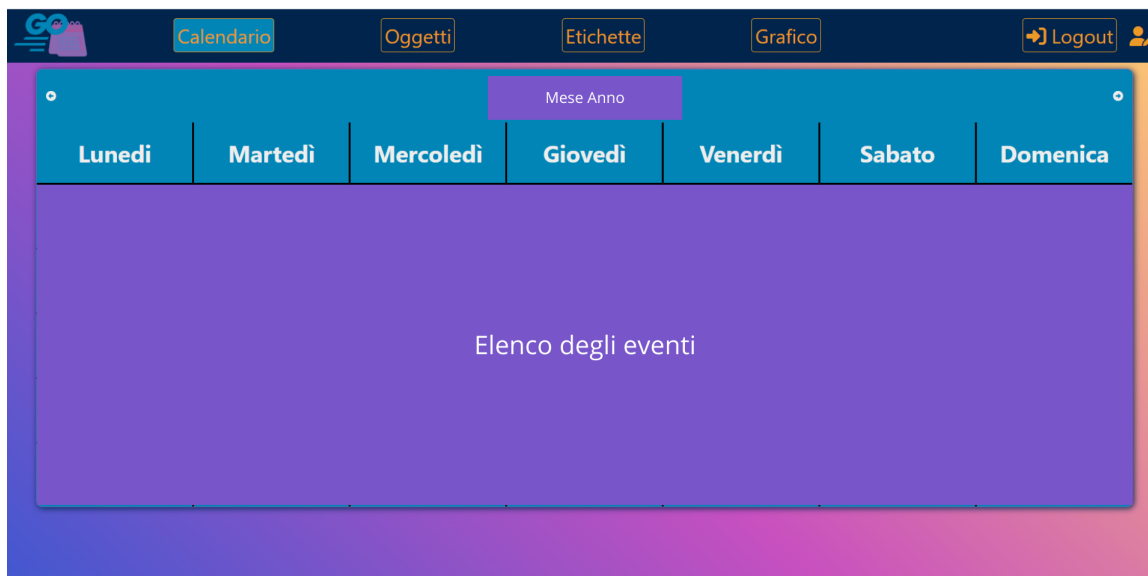


Figure 4: Page del calendario



Figure 5: Aggiunta di un evento



Figure 6: Page degli oggetti



Figure 7: Page delle etichette

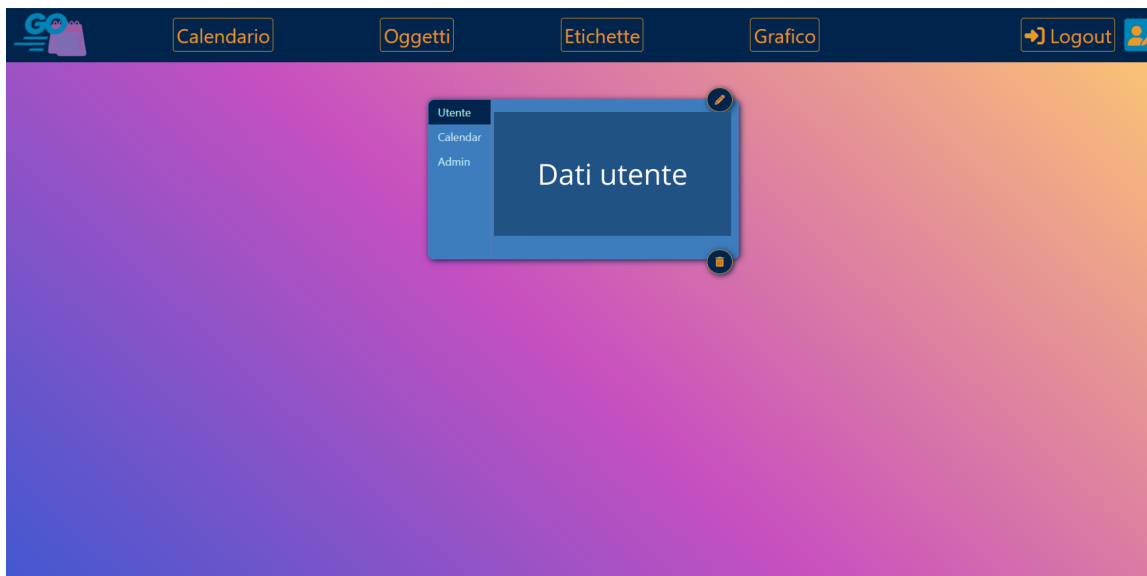


Figure 8: Page dell'utente

4 Tecnologie

L'architettura su cui è basata è quella dello stack MEVN ovvero un sistema composto dalle seguenti tecnologie:

- **MongoDB:**[7] per creazione di database non relazionali (NoSQL).
- **Express.js:**[5] framework che utilizza il linguaggio Javascript utilizzato per la scrittura del lato backend.
- **Vue.js:**[13] framework che utilizza il linguaggio Javascript utilizzato per la scrittura del lato frontend.
- **Node.js:**[10] ambiente per l'esecuzione di applicativi server-side senza avere bisogno di un browser.

4.1 Frontend

Di seguito elencate ulteriori tecnologie utilizzate per il Frontend:

- **Electron:**[4] per la creazione di applicazioni desktop utilizzando comunque le tecnologie web.
- **Vue CLI:**[12] permette di avere comandi di creazione Vue immediati e di semplificare la complessità del codice.
- **ChartJs:**[3] utilizzato per la creazione di diverse tipologie di grafici interattivi.

4.2 Backend

Di seguito elencate ulteriori tecnologie utilizzate per il Backend:

- **Bcrypt:**[2] per il salvataggio a db di una password criptata tramite un valore di salt rendendo più sicura la password.
- **Nager.Date:**[9] API che permette di ottenere la lista delle festività di un certo Paese di un certo anno.
- **Jsonwebtoken:**[6] per la creazione di acces token e refresh token che vengono utilizzati ad ogni chiamata.
- **Mongoose:**[8] utilizzato per l'interfacciamento del serve al database Mongo.

4.3 Ulteriori Tecnologie

Di seguito le rimanenti tecnologie utilizzate:

- **Socket.io:**[11] libreria utilizzata per la creazione di socket interattivi con possibilità di comunicazione veloce tra eventuale lato cliente e server, utilizzata per l'invio delle notifiche tra server a client.
- **Axios:**[1] permette di semplificare la creazione di richieste http asincrone, ovvero che non appena sarà disponibile, il dato verrà inviato.

4.4 Linguaggi utilizzati

Di seguito le rimanenti tecnologie utilizzate:

- **HTML**
- **Javascript**
- **CSS**
- **SCSS**

5 Codice

Il codice è stato pensato per poter essere scalabile e dare la possibilità di poter aggiungere facilmente nuovi componenti che siano sia lato server che lato client.

Nella parte backend si è utilizzato una modellazione in pipeline delle chiamate API RESTful, in modo tale da poter permettere una migliore logica di implementazione di api future e rendere ben leggibile il passaggio che il server effettua tramite il middleware, che è importante anche per l'autenticazione dell'utente stesso; infatti il middleware controlla che il token passato alla chiamata api sia corretta e in caso contrario comunica che non sia corretto il token passato.

Nel frontend è stata utilizzato il salvataggio del token e id dello user dato che sono richiesti per qualsiasi chiamata effettuata al server, tranne ovviamente quelle di login e di registrazione.

6 Test

Le funzioni richieste dall'applicativo sono state testate su diversi dispositivi fisici, sia un pc portatile, desktop e smartphone, dando priorità a quelli desktop inizialmente. Si è fatto principalmente caso che il funzionamento dell'applicativo non cambiasse a seconda del dispositivo utilizzato, e che avesse una visualizzazione conforme per tutti i dispositivi.

Le API sono state testate tramite lo strumento PostMan che permette la visualizzazione e le chiamate ad API in modo tale da confermarne il contenuto, sia in invio che in ricezione.

7 Deployment

Il deployment dell'applicativo può essere effettuato tramite i seguenti passaggi:

- **Clone del seguente repository:** link repo.
- Recarsi nella cartella del repository ed eseguire il comando **npm install** e successivamente **npm start**, in modo tale da avviare il server.
- Il server popolerà, in caso di necessità il database per la prima volta.
- Scaricare la release contenuta nel repo e estrarre il contenuto dello zip.
- All'interno della cartella appena estratta avviare l'eseguibile **gocalendar.exe**, in questo modo si avvierà il lato client dell'applicativo

References

- [1] Axios.
- [2] Bcrypt.
- [3] Chartjs.
- [4] Electron.
- [5] Express.
- [6] Jsonwebtoken.
- [7] Mongoddb.
- [8] Mongoose.
- [9] Nager.date holiday api.
- [10] Nodejs.
- [11] Socket.io.
- [12] Vue cli.
- [13] Vuejs.