**Relazione progetto Ingegneria dei Sistemi Web**

**Componenti del gruppo:**

* Tommaso Stefani - [tommaso.stefani5@studio.unibo.it](mailto:tommaso.stefani5@studio.unibo.it)
* Gabriele Broccoli – [gabriele.broccoli@studio.unibo.it](mailto:gabriele.broccoli@studio.unibo.it)

**Docenti del corso:**

* Giovanni Delnevo
* Chiara Ceccarini

**Descrizione progetto:**

l gruppo si pone come obiettivo quello di realizzare una single page application per la gestione interna ed esterna di un hotel di montagna, con la possibilità da parte di un cliente di visionare le camere e fare prenotazioni nel caso siano disponibili. La prenotazione delle camere è riservata agli utenti registrati che hanno effettuato l’accesso. La registrazione di un utente è compresa di codifica delle password, che non vengono mai conservate in chiaro.

Il sito ha inoltre una sezione dedicata agli amministratori, a cui gli utenti normali non possono accedere tramite una serie di controlli.

Dalla sezione Admin è possibile eliminare camere libere da prenotazioni o le singole prenotazioni dei clienti, è possibile modificare le camere esistenti o aggiungerne di nuove con massimo due immagini ciascuna.

Il sito è stato inoltre testato per l’accessibilità del codice ACCESSIBILITY PERCENTAGE con risultato medio 86% e per EVALUATION COMPLETENESS con risultato medio 91% da <https://mauve.isti.cnr.it/> e per il contrasto colori al 100%.

**Tecnologie utilizzate:**

**back-end:**

Node.js – ambiente di runtime dell’applicazione.

Express.js – framework per l’utilizzo di middleware per semplificare lo sviluppo del server.

Vite.js – inizializzazione facilitata del progetto, aggiornamento del codice in runtime, proxy e supporto typescript.

MySql2 – modulo node che permette la di interagire con un database mysql e di eseguire query usando javascript, include altre interfacce.

XAMPP – web hosting in locale per il database.

Typescript – Superset di javascript, creazione di interfacce riutilizzabili, individuazione più facile degli errori tramite tipizzazione, import ed export facilitati, migliore lettura dei path.

**front-end:**

Vue.js – framework per la creazione della single page application e di tutte le sue funzionalità interattive lato front-end, e comunicazione da/verso il back-end.

Axios – permette di effettuare richieste http al back-end e di interpretare richieste e risposte in formato JSON.

SCSS – utilizzo di variabili per i colori, possibilità di innestare elementi, riduzione ridondanza di codice.

Crypto.js – libreria per la codifica delle password.

Fs – modulo node che permette di interfacciarsi con il file system.

Path – Libreria per rendere dinamici i percorsi dei file.

Multer – middleware che permette la gestione di dati multipart/form-data, include DiskStorage e upload per il salvataggio dei file in locale. Utilizzato nel caricamento delle immagini.

**Avvio del progetto:**

avviare XAMPP -> Apache + MySql

sostituire il path di public/img in

cd ./frontend

npm install

npm run build

cd ../backend

npm install

npm run build

npm run dev (npm run start)

**Funzionamento del progetto:**

Ogni richiesta http effettuata dal front-end tramite axios contiene un oggetto JSON ed è indirizzata a una specifica route.

Le richieste raggiungono il back-end su app.ts, che tramite il router corretto, reindirizza la richiesta alla funzione/query corrispondente nel controller, che contatta il db. La risposta del server contenente i dati della query viene poi processata e restituita al front-end sempre in formato JSON, poi riprocessata e mostrata all’occorrenza su schermo.

**Ripartizione del lavoro:**

Equamente suddiviso