**Relazione Sito Web Chalet**

Questo documento fornisce una relazione dettagliata riguardo a un servizio di prenotazione di ombrelloni da spiaggia per uno chalet. L’obiettivo principale è stato quello di sviluppare un sito web che consenta agli utenti di registrarsi e prenotare il loro ombrellone.

La mia applicazione è di tipo Single Page Application.

**Struttura del Sito**

Il sito è stato strutturato nel seguente modo:

* **Frontend**
* **Backend**
* **Database** (utilizzando il programma **POSTGRES**)

Prima di arrivare al frontend, è stato creato un prototipo su carta per definire la sua struttura.

**Frontend**

Il frontend è stato sviluppato in **React** utilizzando **Vite** e **Typescript**. Questa parte del progetto include diverse pagine:

* **Landing Page**
* **Visualizzazione delle Prenotazioni**
* **Creazione delle Prenotazioni**

Le pagine sono gestite attraverso la libreria “react-router-dom”, che consente di collegare le pagine a specifici percorsi (main.tsx).

Le percorsi sono suddivisi in due categorie:

* **Percorsi protetti**: accessibili solo dopo l’effettuazione del login.
* **Percorsi pubblici**: accessibili da chiunque (ad esempio, la Landing Page).

I percorsi pubblici non includono la barra di navigazione superiore (AppBar), a differenza dei percorsi protetti, che la contengono, poiché la AppBar include elementi di autenticazione.

**Layout**

Il Layout include i seguenti componenti:

* **CustomThemeProvider**: contiene il ThemeProvider della libreria MUI, che imposta il tema dei componenti in base alle preferenze dell’utente (chiaro o scuro).
* **AuthProvider**: fornisce ai componenti figli le funzionalità di autenticazione tramite il context provider.
* **ProtectedRoute**: visualizza il componente figlio solo se l’utente è autenticato; altrimenti, reindirizza alla pagina di errore 403.
* **CustomAppBar**: contiene l’AppBar di MUI con un pulsante di logout.
* **Outlet**: la pagina da visualizzare in base al percorso specificato.

**Visualizzazione Prenotazioni**

Questa pagina mostra un elenco di immagini rappresentanti gli ombrelloni, organizzati in una griglia. Ciascun ombrellone ha un “event listener onClick” che reindirizza alla pagina “Crea Prenotazione” con i dati dell’ombrellone selezionato. Gli ombrelloni prenotati mostrano un’icona di divieto e non possono essere selezionati.

**Crea Prenotazioni**

Questa pagina consente agli utenti di effettuare una prenotazione per l’ombrellone selezionato. I dati seguenti vengono inviati al server:

* Numero di fila e colonna
* Data di inizio e fine della prenotazione
* Numero di sdraio
* Numero di lettini

**Autenticazione**

L’autenticazione è gestita tramite il componente **AuthProvider.tsx**, che fornisce alle componenti figlie le funzionalità di autenticazione, come login, logout, checkAuth e navigateProtected, utilizzando il context hook.

Tutte le funzioni di autenticazione inviano richieste HTTP tramite la funzione **SendRequest**, definita nel componente **Api.tsx**. Questa funzione utilizza la fetch API per inviare le richieste al backend. Il componente è progettato per essere riutilizzabile e accetta parametri per configurare le richieste.

**Backend**

Il backend è stato sviluppato in **Node.js** utilizzando **Express** e **TypeScript** nell’**index.js**. L’index.ts è suddiviso in vari endpoint:

* **GET** /api/sunshades
* **POST** /api/auth
* **POST** /api/protected
* **POST** /api/logout
* **POST** /api/booking
* **DELETE** /api/booking
* **GET** /api/booking/owner/:booking\_id

Il file **Connect.tsx** contiene le funzioni per inviare query al database tramite la libreria **node-postgres** (pg). Gestisce la connessione al database utilizzando variabili d’ambiente caricate da **dotenv** dal file **.env**.

**Middleware**

Il middleware è situato tra il frontend e il backend.

Il middleware contiene delle funzioni per gestire il JWT e la password dell’utente.

Utilizza la libreria ‘jsonwebtoken’ per creare e verificare un token.

Contiene le seguenti funzioni:

* Authenticate token
* Generate access token
* hashPassword
* compare password

**DATABASE**

Il database (dbDiagram.png) è strutturato in 4 tabelle:

* chalet: contiene nome, indirizzo e città.
* sunshade: contiene numero di riga, numero di colonna, numero di sdraio, numero di lettini, il prezzo dello sdraio e il prezzo del lettino.
* booking: contiene la data inizio e la data di fine
* users: contiene email e password.

Le relazioni tra le varie tabelle hanno le seguenti cardinalità:

* Chalet -> Sunshade 1 a N
* Sunshade -> Booking N a 1
* Booking -> Users N a 1

