



Progetto Sistemi Aperti e Distribuiti

Dipartimento di Matematica e Informatica Anno
accademico 2018/2019

Studenti

Mancinelli Alessio

Massetti Matteo

Professore:

Prof. Sergio Tasso

Indice

Analisi dei Requisiti	2
Realizzazione del servizio.....	3
Struttura del database.....	3
Costruzione del servizio REST.....	7
Ottenimento dati formato JSON.....	9
Scelta del dominio del servizio.....	9

Analisi dei Requisiti

La richiesta del committente consisteva nel creare un servizio web in grado di gestire le prenotazioni di una serie di Pizzerie. Analizzando le richieste ed i requisiti possiamo strutturare il Webservice come segue.

Il cliente può: eseguire il login e la registrazione al sito, visualizzare, modificare ed eliminare le prenotazioni effettuate, creare nuove prenotazioni ed eliminare il proprio account dal servizio.

La pizzeria può: Effettuare il login e la registrazione al sito, visualizzare modificare, eliminare i giorni in cui offre delle prenotazioni ed eliminare il proprio account dal servizio.

1.1 Abstract del Progetto

Pizzerie Riunate offre agli utenti un servizio di gestione delle prenotazioni dalla parte dei clienti, mentre alle Pizzerie offre la completa gestione dei giorni di apertura dell'attività. Per far questo il sistema si appoggia ad un database creato in MySQL grazie all'ausilio di Xampp dove verranno immagazzinati tutti i dati inseriti dagli utenti.

Realizzazione del Servizio

Il servizio è stato realizzato usando:

Xampp : Utilizzato per la gestione del server.

AngularJS : Utilizzato per realizzare i servizi REST.

MySQL : Linguaggio utilizzato per la gestione del Database.

Php : Utilizzato per effettuare le varie richieste al Database.

JSON: Tutti i dati vengono scambiati in formato JSON e sono inviati tra PHP e AngularJS.

HTML : Linguaggio utilizzato per la realizzazione delle pagine Web.

I file di implementazione sono stati suddivisi all'interno delle directory nel seguente modo:

index.html: File che viene caricato da AngularJS al suo avvio, attraverso di esso si effettuerà il display di tutti i singoli Templates implementati per il progetto.

Cartella css : Contiene il file bootstrap.css scaricato dal sito <https://bootswatch.com/> il quale ha enormemente facilitato l'implementazione e lo stile delle pagine HTML del sito.

Cartella js : Contiene tutti i file per l'utilizzo di AngularJS, inoltre il file app.js contiene tutti i metodi necessari al funzionamento del servizio.

Cartella Templates: Contiene tutti i file HTML per la visualizzazione del sito che verranno richiamati all'interno di index.html attraverso il tag <ng-view> di AngularJS.

Cartella webservice: Contiene tutti i file php utilizzati per interrogare il database. Nello specifico, il file config.php specifica anche come evitare la politica CORS che impedisce lo scambio di dati tra pagine. La seguente politica è stata disattivata per permettere il corretto funzionamento del servizio REST.

Struttura del Database

Il database è stato realizzato con tre tabelle fisse:

Tabella Clienti: All'interno vengono salvati tutti i dati dei clienti al momento della registrazione

```
CREATE TABLE clienti(  
    email VARCHAR(255) PRIMARY KEY,  
    password VARCHAR(255),  
    nome VARCHAR(255),  
    cognome VARCHAR(255),  
    indirizzo VARCHAR(255)  
);
```

Tabella Lista_pizzerie: All'interno vengono salvati tutti i dati delle pizzerie al momento della registrazione.

```
CREATE TABLE lista_pizzerie(  
    nome_pizzeria VARCHAR(255) PRIMARY KEY,  
    password VARCHAR(255) NOT NULL,  
    indirizzo VARCHAR(255)  
);
```

Tabella Prenotazioni: All'interno vengono salvati tutti i dati delle prenotazioni effettuate dai clienti

```
CREATE TABLE prenotazioni(  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    pizzeria VARCHAR(255) REFERENCES lista_pizzerie(nome_pizzeria),  
    utente VARCHAR(255) REFERENCES clienti(email),  
    giorno DATE,  
    numero_posti_prenotati INT NOT NULL,  
    ora INT NOT NULL,  
    minuti INT NOT NULL  
);
```

Viene poi implementata per ogni pizzeria registrata una tabella utilizzata per gestire i diversi giorni.

```
CREATE TABLE singola_pizzeria_prova(  
    giorno DATE,  
    nPostiLiberi INT,  
    nPostiTotali INT,  
    ora_apertura INT,  
    ora_chiusura INT,  
    numero_tavoli INT,  
    PRIMARY KEY (giorno)  
);
```

Costruzione del servizio REST

Il servizio REST all'interno del sito è stato implementato tramite l'ausilio di AngularJS. AngularJS gestisce il sito mediante dei controller definiti nel file `app.js` e richiamati nei diversi file `html`. Attraverso questi controller è possibile chiamare il file `php` interessato e passargli dei parametri tramite 'POST' e 'GET'. Inoltre si sono utilizzate alcune funzionalità di angular (come `ng-repeat`) per visualizzare nella pagina HTML i dati ottenuti dal Database.

I metodi per il passaggio dei parametri come già precedentemente accennato sono stati sia 'GET' che 'POST'. Il primo è stato principalmente utilizzato per tenere presente quale giorno l'utente intende modificare/cancellare poichè il giorno stesso (che è chiave primaria all'interno del database) è passato attraverso l'URL ed è quindi ottenibile tramite il metodo `Get`. Questo passaggio è stato necessario in quanto per esempio, in una lista di prenotazioni effettuata da un utente, saranno presenti molteplici giorni ed è necessario tenere traccia di quale giorno esso intenda modificare/cancellare.

Per utilizzare AngularJS si sono importati nell'head del file `index.html` i seguenti file:

`angular.js`

`angular-route.js`

In concomitanza con AngularJS, il progetto è stato implementato anche grazie all'ausilio dei Cookie. Essi sono stati utilizzati per memorizzare le credenziali di accesso degli utenti. Quest'informazione viene utilizzata per effettuare le interrogazioni all'interno del database, in modo da considerare solo i dati relativi all'utente loggato.

All'interno del file `app.js` è stato definito un modulo chiamato `pizzerieRiunite` di tipo `ngRoute`, esso fornisce servizi di routing per applicazioni che utilizzano `AngularJs`. Per utilizzare questi servizi di routing si utilizza la variabile `$routeProvider` che in base all'URL della pagina attuale gestisce i vari controller da applicare. Si è anche definito poi, con il metodo `.otherwise`, che qualora l'utente inserisse manualmente un URL incorretto allora esso sarà immediatamente rediretto alla pagina iniziale di Login.

Le altre variabili utilizzate all'interno delle funzioni di controller sono `$scope` e `$http`.

La variabile `$scope` permette di mettere in comunicazione il controller con l'html, all'interno del progetto è stata utilizzata per ottenere i singoli valori dagli oggetti JSON creati all'interno dei file PHP. Questi dati saranno poi mostrati all'interno del sito su una sezione apposita per la visualizzazione della risposta del sistema.

La variabile `$http` permette di effettuare chiamate Ajax al server e di gestire le risposte in maniera molto semplice, se la richiesta va a buon fine allora sarà eseguito il codice relativo alla parte `.then()`. Nel progetto è stata utilizzata la parte relativa a quest'ultimo metodo per aggiornare l'HTML affinché l'utente potesse avere una risposta dal servizio.

Molta attenzione è stata posta nell'effettuare tutti i controlli necessari affinché l'utente non inserisse valori scorretti che avrebbero portato inconsistenza nei dati. Infatti, prima che i file php vengono contattati dal servizio REST, ci si assicura che tutti i dati inseriti dall'utente siano sia del corretto tipo che nella giusta forma. Per aiutare l'utente a non effettuare un inserimento errato, è stato anche implementato il calendario sulle form di scelta del giorno sia per i clienti che effettuano una prenotazione sia per le pizzerie che scelgono il giorno da aggiungere/modificare nella loro programmazione.

All'interno della gestione degli orari si è assunto vero che una pizzeria non possa prendere prenotazioni dopo la mezzanotte, questo comporta che una pizzeria, per esempio, non può aprire alle 22:00 e chiudere alle 4:00 del giorno dopo.

Ottenimento dei dati formato JSON

Per lo scambio dati è stato utilizzato il formato JSON, sia trattato come singolo oggetto sia come array di dati. Quest'ultimo aspetto è stato utilizzato principalmente all'interno della visualizzazione della lista prenotazioni, da parte dell'utente, e della programmazione giornaliera di ogni pizzeria. I dati sono stati ottenuti all'interno del file `app.js` attraverso la variabile `$scope` e mostrati poi attraverso il meccanismo di `ng-repeat` proprietario di AngularJS.

Scelta del dominio del servizio

Per rendere più realistico il webservice Pizzerie Riunate, si è scelto di cambiare l'url del sito da `localhost` a `PizzerieRiunate.it`, questo permette ad un eventuale utente del servizio, di accedervi in maniera molto più semplice ed intuitiva.