Scopus App

Presentato da: Oleksandr Melnychuk

Scopus App

8 giugno 2018

Tutor didattico: Silvia Mirri

Introduzione 3

Tecnologie 4

Attività 4

Conclusioni 7

Bibliografia 7

Nome società

Introduzione

Scopo

Lo scopo del tirocinio è quello di progettare e implementare un prototipo di web app basandosi sui dati e sulle APIs offerte da Scopus (<https://www.scopus.com/home.uri>), uno dei principali servizi della casa editrice Elsevier (<https://www.elsevier.com/>). In particolare, Scopus è un database bibliografico che raccoglie articoli scientifici relativi a diversi ambiti di ricerca e offre agli utenti diversi informazioni e dati bibliometrici relativi a singole pubblicazioni scientifiche, libri ed enciclopedie, riviste scientifiche, atti di conferenze internazionali, docenti e ricercatori. Attualmente, Scopus si basa su un database contente oltre 22.000 titoli forniti da più di 5.000 editori internazionali.

Obiettivi

L’obiettivo principale del tirocinio è stato quello di acquisire, e comprendere a pieno, tutte le tecnologie atte a progettare e sviluppare una Web App con un design di tipo responsive. La Web App deve inoltre fornire agli utenti elementi di visualization interattivi per permettere la navigazione di dati bibliometrici, sia aggregati che non aggregati.

Tecnologie

Angular 2

Come tecnologia principale, si è scelto di utilizzare Angular, in quanto al momento risulta essere un promettente framework emergente, in continuo aggiornamento. Come versione, è stata scelta la numero 2, preferendola a versioni più recenti, in modo da apprendere tale linguaggio partendo dalle basi e poter avere a disposizione più documentazione facilmente reperibile.

Angular 2 è una piattaforma open source per lo sviluppo di applicazioni web e risulta essere la naturale evoluzione, non a livello sintattico ma concettuale, di AngularJS. A livello di codice, le due tecnologie hanno poco in comune, in quanto Angular 2 è stato completamente riscritto e, di conseguenza, i due framework non sono compatibili. Ciò nonostante, a livello concettuale, i due linguaggi risultano fortemente connessi. Il linguaggio di programmazione utilizzato per implementare AngularJS è JavaScript invece, per Angular 2, è stato utilizzato TypeScript. Al tempo stesso la maggior parte del codice può essere scritta utilizzando l’ultima versione di JavaScript sfruttando i types per iniezione e i decoratori per i metadati.

Scopus APIs

Le API Scopus forniscono un’interfaccia per accedere agli abstract e ai dati delle citazioni, forniti dai molteplici periodici accademici indicizzati da Scopus. Inoltre, è possibile accedere al database estratto dalla casa editrice Elsevier.

Attività

L’attività del tirocinio consiste, come precedentemente accennato, nella creazione di una Web App di tipo responsive che permetta, tramite l’utilizzo delle API fornite da Scopus, di interfacciarsi con il servizio in questione al fine di permettere all’utente di interrogare il Database per ottenere le informazioni desiderate.

Nello specifico, la Web App mette a disposizione due funzionalità principali: La prima consiste nell’interrogazione del Database al fine di ottenere tutti gli abstract e le citazioni relative al singolo autore. La seconda, invece, consente di effettuare un confronto tra più autori.

Come punto di partenza è stata sviluppata una versione dummy dell’applicativo, in modo da prendere confidenza con le tecnologie utilizzate. Inoltre, tale applicativo fittizio si è rivelato essere utile per comprendere la struttura dei dati restituiti dal database. Per sviluppare tale applicativo, è stata consultata la pagina <https://dev.elsevier.com/interactive.html>, dove vengono messe a disposizione degli sviluppatori alcuni esempi di utilizzo delle API per effettuare diverse tipologie di interrogazioni al Database. Inoltre, per testare il corretto funzionamento di tale versione dummy, è stato creato un file db.json, contenente una risposta, in formato json, prelevata dalla pagina indicata precedentemente. Per simulare in manie efficace e, da un punto di vista funzionale, totalmente simile ad un utilizzo reale, è stato attivato un server in localhost, in ascolto sulla porta 3000. Tale server è totalmente conforme allo standard REST, quindi il suo comportamento risulta totalmente conforme a quello del server, se l’esempio fornito nella pagina precedentemente indicata risulta efficace.

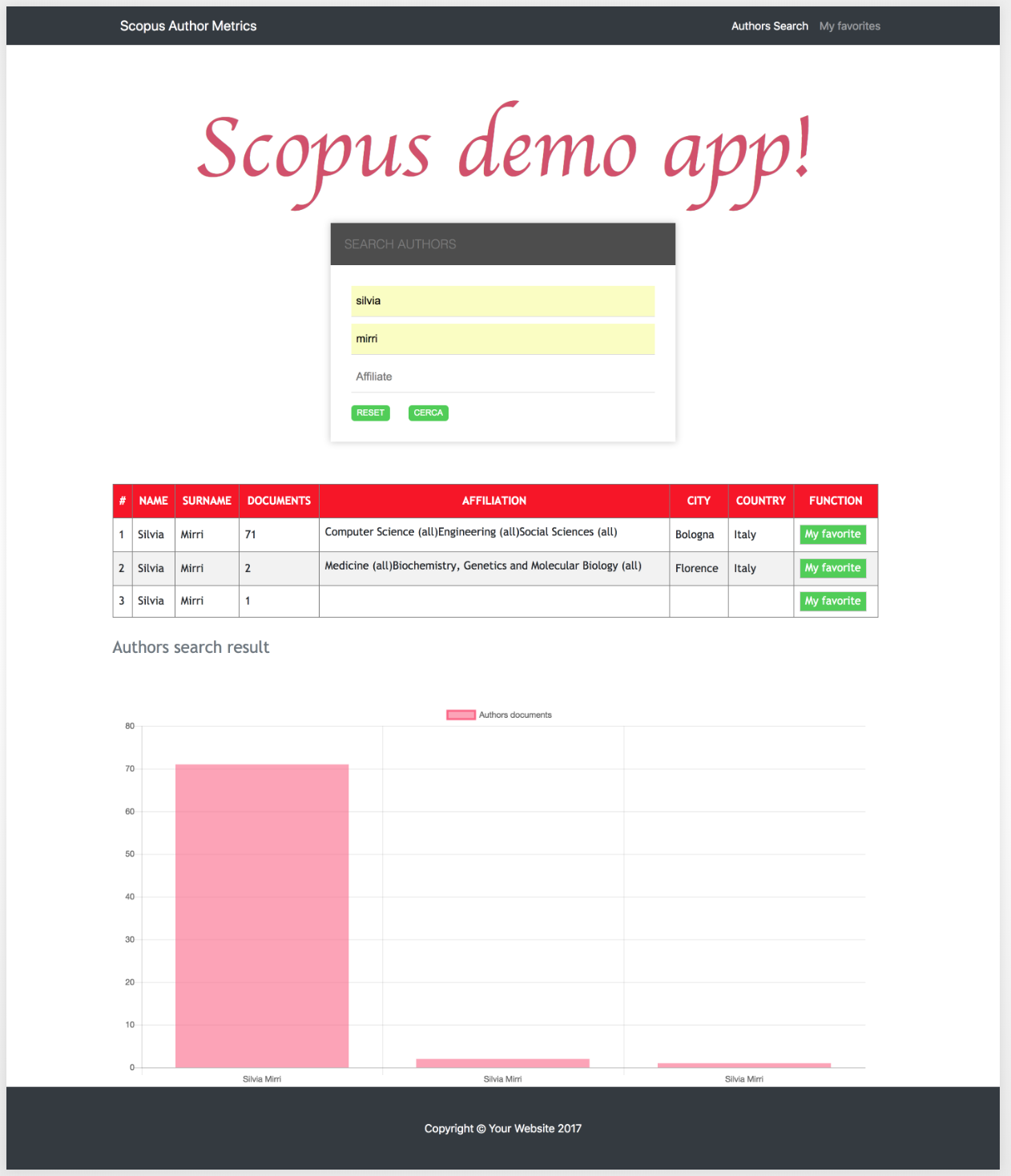
Nello specifico, per lanciare il server, è necessario lanciare il comando npm run json-server il quale, caricato il file db.json dalla cartella di progetto, attiva un database RESTful, con tutti i percorsi corretti.  
Una volta attivato il server rimane in ascolto per eventuali richieste e, quando pervengono, analizza il file .json e fornisce al richiedente le informazioni richieste.

In seguito, l’applicazione è stata ulteriormente sviluppata, in modo da supportare la funzionalità “Preferiti”. Tramite il pulsante “MyFavorite” l’utente, dopo aver effettuato una ricerca, può aggiungere i risultati ottenuti alla lista dei preferiti.

Per semplicità, si è scelto di utilizzare il Database interno al browser (indexes.db).

A causa di alcune problematiche riscontrate nell’utilizzo dei *CORS (Cross Origin Source Sharing)*, è stato necessario lanciare il browser Chrome con permessi d’amministratore, in modo da poter disabilitare la *Web Security.* Per effettuare tale operazione è stato lanciato il comando: “**sudo /Applications/Google\ Chrome.app/Contents/MacOS/Google\ Chrome --disable-web-security --user-data-dir "/Users/OleksandrMelnychuk/Library/Application Support/Google/Chrome/DISABLE"** “

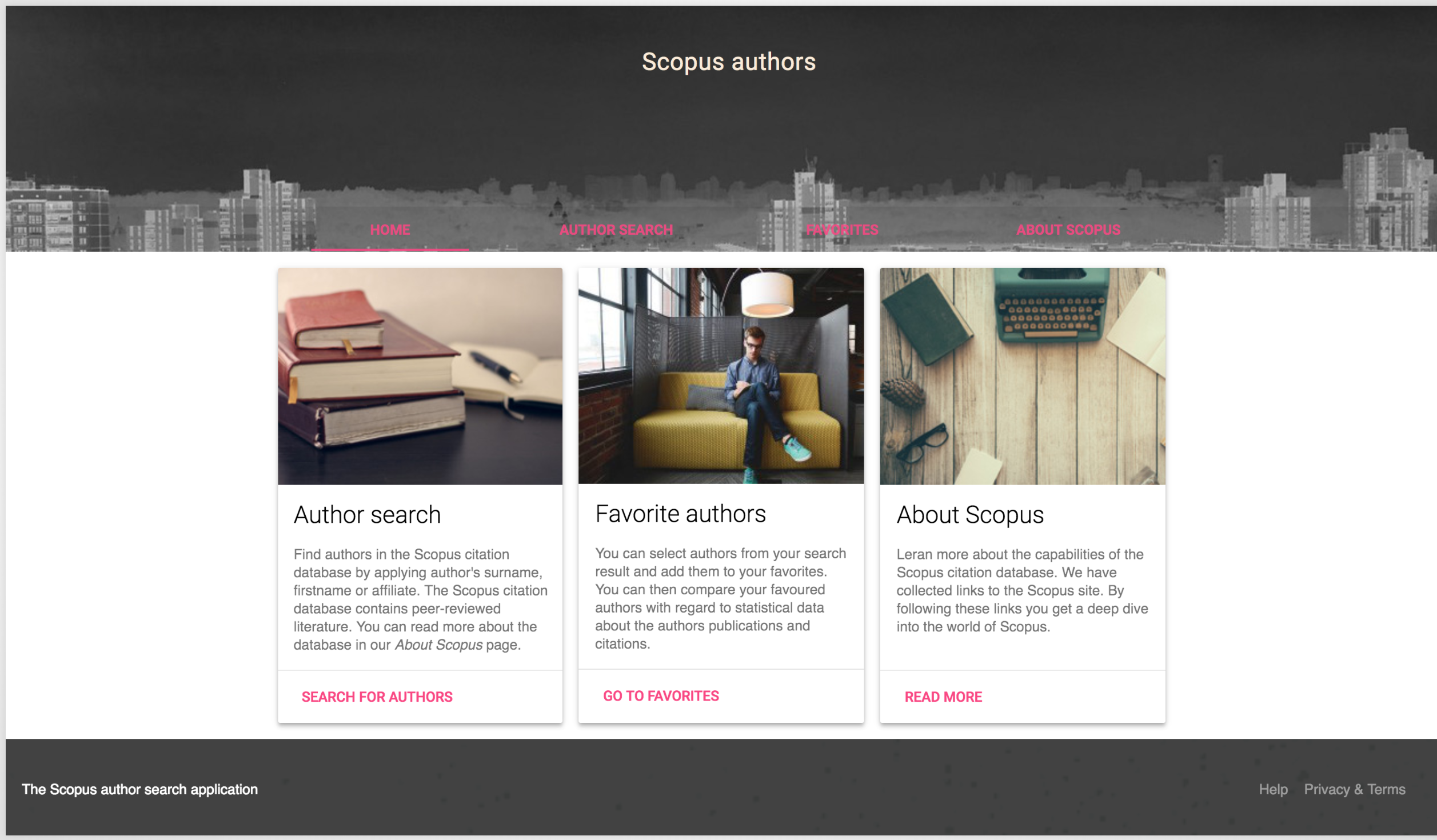
Benchè tale procedura sia pericolosa e sconsigliabile, al momento risulta essere l’unico *Work Around* efficace in tale contesto.

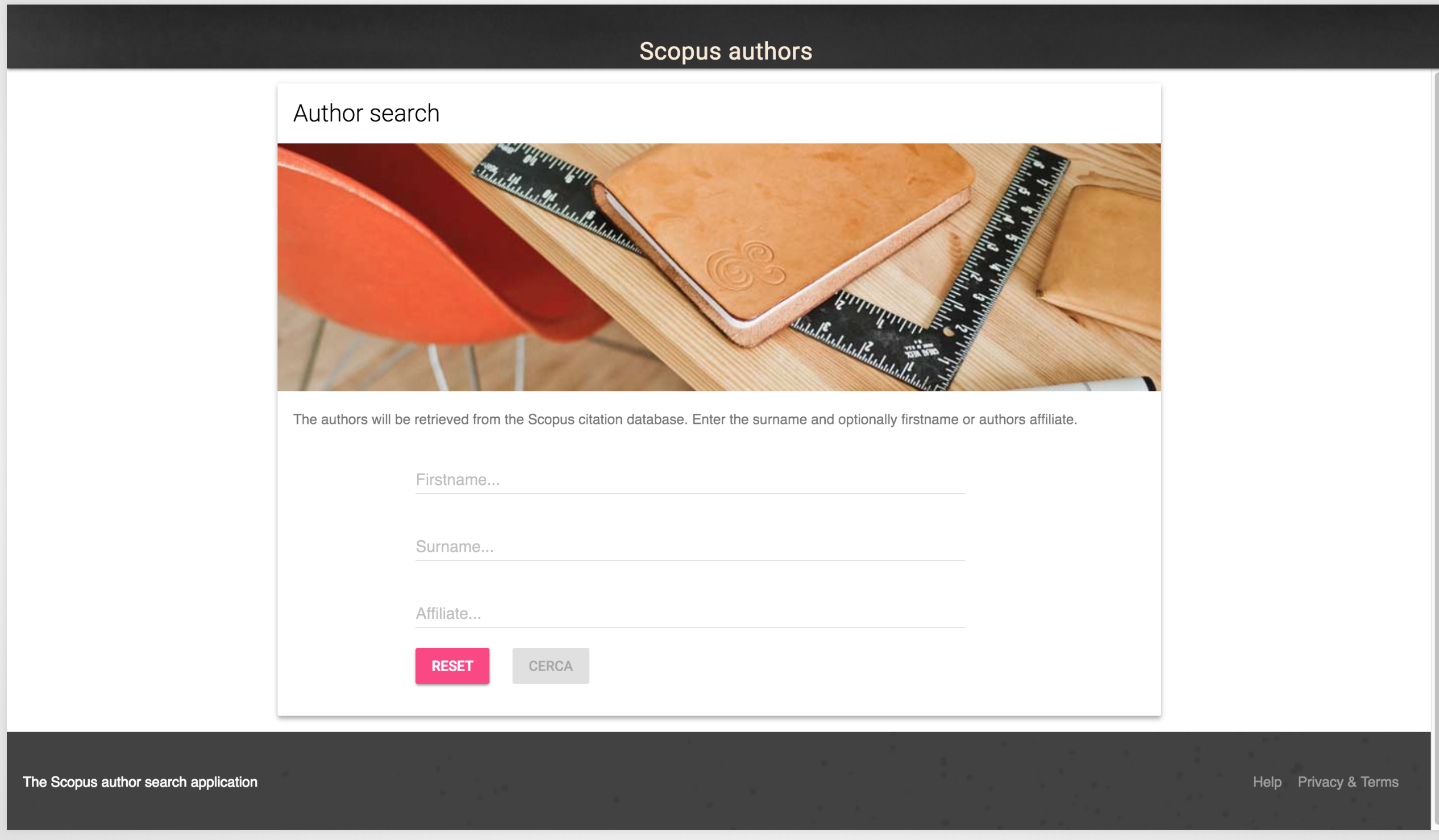
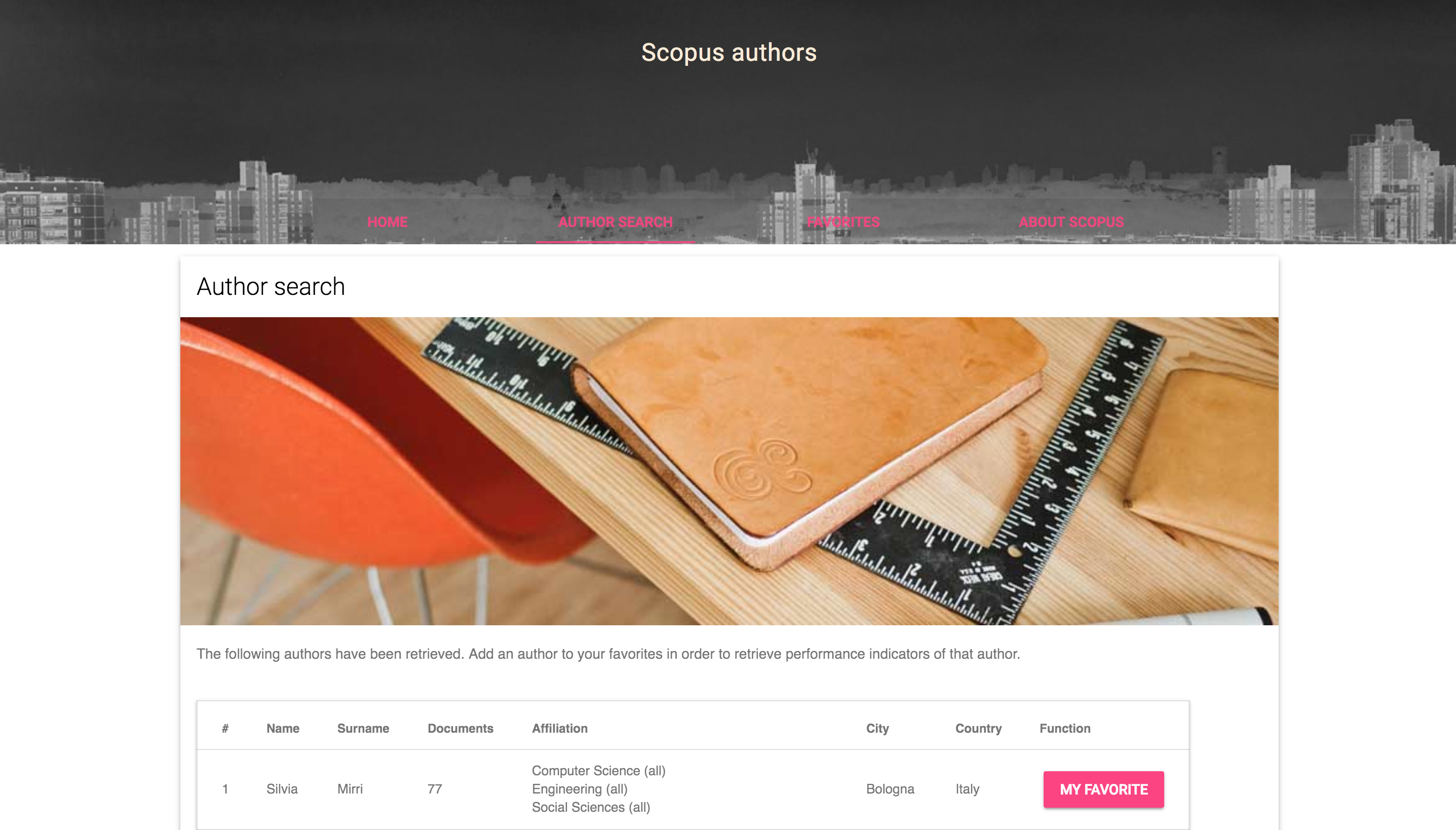


Nell’immagine precedente, a sinistra, viene mostrata la versione Dummy sviluppata inizialmente. Nella parte superiore è presente il form per la ricerca, contenente i due pulsanti di ricerca e di reset. Nella parte centrale è presente una tabella strutturata per mostrare i dati estratti dal Database. Nella parte inferiore è presente un grafico generato sui dati fittizi a disposizione che mostra statistiche relative ai documenti pubblicati.  
Nell’immagine a destra, invece, viene mostrato il funzionamento della pagina “Preferiti”. Selezionando i vari autori è teoricamente possibile confrontarli tra di loro.

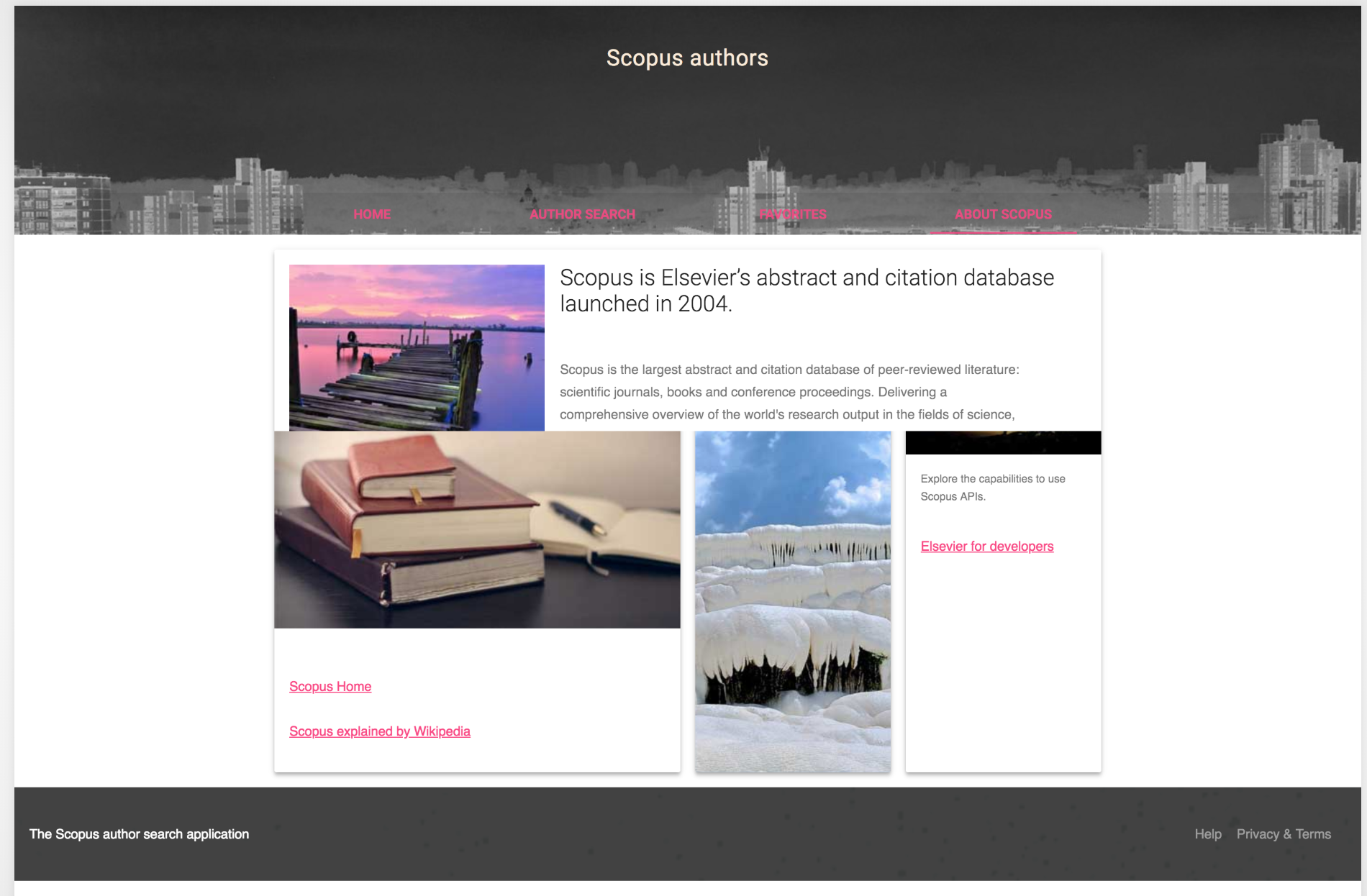
Dopo aver effettuato diverse prove e prototipi basandosi sulla versione Dummy precedentemente illustrata, si è cercato di sviluppare una Web App realmente funzionante.  
La prima problematica riscontrata è relativa alla non coerenza su quanto riportato dal servizio nella pagina di presentazione delle API.

Dopo vari tentativi sono state risolte le problematiche relative alla ricerca per autore. Le modifiche effettuate hanno però reso inutilizzabile la funzionalità dei “Preferiti”.



Nell’immagine precedente viene mostrata la home page dell’applicativo sviluppato, dove è possibile accedere alle funzionalità di ricerca per autore e di gestione dei favoriti.

Nelle due immagini precedenti viene mostrato la sezione relativa alla ricerca per autore.

Pagina contenente le informazioni relative a Scopus.

Conclusioni

Ritengo l’attività particolarmente formativa sotto diversi punti di vista.

Il primo consiste nell’apprendimento di tecnologie quali Angular, utili per sviluppare Web App interattive.  
Secondariamente ritengo l’esperienza formativa, in quanto mi ha richiesto un notevole sforzo, spronandomi ad effettuare numerose ricerche per comprendere tecnologie precedentemente sconosciute.

Bibliografia

<https://dev.elsevier.com/>

<https://dev.elsevier.com/api_docs.html>

<https://www.npmjs.com/package/json-server>