

## Applicazioni Internet 2016-17

### Esercitazione 4

Si vuole realizzare, utilizzando Spring Boot, un piattaforma social per la condivisione e lo scambio di opinioni tra utenti riguardanti i trasporti della città di Torino.

La piattaforma dovrà essere organizzata come segue:

1. Home page – punto di ingresso della piattaforma nella quale sono elencati i diversi argomenti di discussione (implementare almeno tre categorie differenti – es. Bus&Metro, Traffico, Giro in bici). Questa pagina deve inoltre consentire l'accesso alla pagina di login/registrazione e alla pagina del profilo utente. Per partecipare alla discussione è infatti necessario autenticarsi alla piattaforma<sup>1</sup>.
  - a. L'aspetto di questa pagina deve quindi variare in funzione del fatto che l'utente sia loggato o meno: login e registrazione nel caso di utenti anonimi, modifica profilo e logout per utenti registrati.
2. Login/Registrazione – ogni utente deve registrarsi alla piattaforma inserendo la propria mail, un nickname e una password di almeno 8 caratteri. All'atto del login l'utente deve inserire email e password; il nickname è invece visualizzato dagli altri utenti all'interno delle discussioni.
3. Profilo utente – questa pagina ha lo scopo di raccogliere informazioni riguardanti l'utente e le sue abitudini di mobilità. Tale pagina viene visualizzata in seguito al primo login o tramite un apposito pulsante presente nell'home page. Consente la modifica del nickname e della password e raccoglie le seguenti informazioni:
  - a. Genere
  - b. Età
  - c. Livello di istruzione (scegliere da un insieme predefinito)
  - d. Occupazione (scegliere da un insieme predefinito)
  - e. Possesso di un'auto privata; nel caso di risposta affermativa aggiungere i campi:
    - i. Anno di immatricolazione
    - ii. Tipologia di carburante (scegliere da un insieme predefinito)
  - f. Utilizzo di un servizio di car sharing; nel caso di risposta affermativa scegliere il fornitore del servizio (scegliere da un insieme predefinito)
  - g. Utilizzo della bicicletta; nel caso di risposta affermativa indicare se si utilizza una bici privata e/o il servizio di bike sharing
  - h. Utilizzo dei mezzi pubblici; nel caso di risposta affermativa indicare il titolo di viaggio utilizzato più frequentemente (scegliere da un insieme predefinito)

**Facoltativo** – consentire all'utente di caricare una foto profilo visualizzata nella chat.
4. Chat per argomento – per ogni argomento selezionato nella home page, si accede ad una pagina dedicata nella quale è possibile:
  - a. Visualizzare i messaggi, in tempo reale, scritti dagli altri partecipanti alla discussione
  - b. Scrivere un proprio messaggio

Ogni messaggio deve riportare il nickname dell'autore, la foto profilo se presente (utilizzare un'immagine di default in caso questa sia assente) e la data e ora di inserimento.

Si rendano i messaggi persistenti, memorizzandoli per poi analizzarli in seguito. Al momento dell'ingresso in una chat, popolare quest'ultima con i 10 messaggi più recenti.

**Facoltativo** – consentire l'upload di immagini all'interno dei messaggi in chat.

Per l'implementazione delle suddette funzionalità, realizzare delle pagine dinamiche con Thymeleaf<sup>2,3</sup>. Utilizzare websocket<sup>4</sup> per la realizzazione delle chat (vedi slide 14-applicazioniintemporeale.pptx).

Inoltre, al fine di rendere queste informazioni accessibili a terzi (agenzie trasporti, amministrazione pubbliche, etc.), esporre i seguenti endpoint REST:

1. Elenco dei profili utente registrati  
Deve ritornare una lista di profili anonimizzati (eliminare nickname, email e password, foto se presente). Tale endpoint deve prevedere dei meccanismi di paginazione dei risultati.
2. Elenco degli argomenti di discussione
3. Elenco dei messaggi relativi ad uno specifico argomento di discussione.  
I risultati dovranno essere paginati.

Tutti questi endpoint devono restituire i dati in formato JSON. Si ponga attenzione alle convenzioni sui nomi e i parametri descritti dal modello REST (vedi slide 12-restspring.pptx).

#### NOTE

1. <https://spring.io/guides/gs/securing-web/>
2. <https://spring.io/guides/gs/serving-web-content/>
3. <http://www.thymeleaf.org/doc/tutorials/3.0/usingthymeleaf.html>
4. <https://spring.io/guides/gs/messaging-stomp-websocket/>