Skill Map Engine

**This is a Skill Mapping Engine.**

1. Core functionality –
   1. Employees will register themselves in the application and fill in the professional details along with their core competencies.
   2. The application should add the competencies to the skill map.
   3. The Employers should be able to search for relevant profiles for their openings(Permanent/Temporary) using the keyword(s) which should match the profiles entered by the users in the engine.
   4. Based on the match, the relevant profiles should be shown to the employers with their contact details and IBU name.
   5. Users should be shown their profile performance and should be shown the no. of times their profiles were visited in a specific period (for e.g. 30 days).

1. Stories:
   1. Front end functionality
      1. Employee
         1. Employee Profile creation
         2. Authentication(through Email ID and password)
         3. Update profile
         4. Archive Profile
         5. View Profile Status and Performance
         6. View Profile completeness%
      2. Employer
         1. Employer Registration
         2. Authentication
         3. Search through keyword for skills
         4. View the result with match%, contact details, supervisor name
         5. Subscribe to certain keywords.
      3. HR
         1. Profile Validation
   2. API layer
      1. user auth
      2. create user, delete user, suspend user
      3. create profile, update profile, archive profile, show profile completeness, show profile performance
      4. search using one or multiple keywords(Skills) separated by comma. It should return matching profiles with profile match%, contact details, IBU details and supervisor name.
      5. Subscribe to keywords so that new profiles matching the keywords are notified to the subscribed employer.
      6. Profile validation
   3. Business logic layer
      1. Skill Mapping algorithm
      2. Calculate profile matching%
   4. Persistence layer
      1. DB
2. Design elements to focus on:
   1. Architecture components:
      1. Client side app connecting to a server-side set of Micro services using REST APIs
      2. Micro service based separation of the persistence model. The cross cutting services are for routing and multiplexing.
   2. server side:
      1. what is the data model for profiles. is extensibility for new profile types taken care of payload.
      2. what is the data model for the subscriptions of profiles.
3. Advanced concepts that can be introduced as variants:
   1. Pluggable client side components to act as templates for various types of profiles.
   2. Use REST - HATEAOS based APIs
   3. State management on the server to a multi-server environment.
4. Potential tech stacks
   1. Complete MEAN or MERN – Angular or React based front end – uses Material or SemanticUI, JSON as well as REST and JSON based communication with server. API layer written using NodeJS and Express. Middleware layer written as micro services in NodeJS. Persistence in Cassandra
   2. Java Spring + Angular/React – Angular or React based front-end – uses Material or SemanticUI. Spring boot based micro services. Tomcat and Spring based API gateway. Persistence in Cassandra