|  |  |
| --- | --- |
| DESIGN NOTE  ON  SKILLS TRACKER  .Net FSD | ABSTRACT  The Skills Tracker Design note summarizes the Project Structure and includes the screenshots of the working application  Developer  Nakash Sheikh (295380) |

Contents

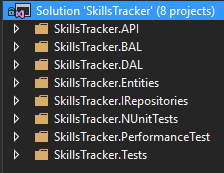
[Project Structure: 3](#_Toc514018029)

[Skills Tracker – SPA: 6](#_Toc514018030)

[Working Screenshots of Skills Tracker: 7](#_Toc514018031)

Project Structure:

The Skills Tracker Project implements a “Separation of Concerns” pattern for maintainability. The following sections details each concern’s functionality.



1. **SkillsTracker.API:**

This is a ASP.NET Web API project which acts as a bridge between the Single Page Application (SPA - built with Angular) and the database. The request is in the form of Data Transfer Objects (DTO) which is sent to the next concern which is the SkillsTracker.BAL.

1. **SkillsTracker.BAL (Business Access Layer):**

This is a class library which contains all the business logic of the SkillsTracker . The incoming data from the API request is tranformed into the Database entities and sent to the next concern which is the SkillsTracker.DAL.

1. **SkillsTracker.DAL (Data Access Layer):**

This is a class library which contains all the data access logic of the SkillsTracker. The layer implements Repository pattern to access the data via the Entity Framework.

1. **SkillsTracker.Entites:**

This is a class library containing POCOs which are used as entities (for Entity Framework Code First approach) in the Data Access Layer and the Data Transfer Objects for APIs.

1. **SkillsTracker.IRepositories:**

This is a class library containing the contracts for the Repository Framework.

1. **SkillsTracker.NUnitTests:**

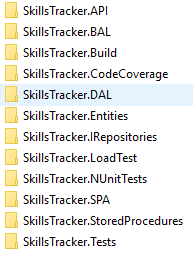
This is a class library containing unit tests implemented using the NUnit Framework.

1. **SkillsTracker.PerformanceTest:**

This is a class library containing preformance tests implemented using the NBench. The folder contains a Readme.txt file containing the guidelines. Reports are copied in the folder.

1. **SkillsTracker.Tests:**

This is a class library containing unit tests implemented using MSTest.



1. **SkillsTracker.SPA:**

This is an Angular project built with Angular CLI.

1. **SkillsTracker.StoredProcedures:**

This contains the database scripts for creation of the database, tables, stored procedures and also seeding the tables.

1. **SkillsTracker.Build:**

This contains the build scripts written in NAnt which integrates with CruiseControl.NET Continuous Integration server. The folder contains a Readme.txt file containing the guidelines.

1. **SkillsTracker.CodeCoverage:**

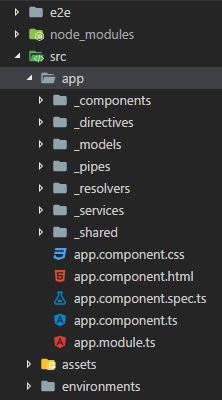
This contains the script for running Code Coverage. The Code Coverage report for Skills Tracker has been copied in the folder. The folder also contains a Readme.txt file containing the guidelines.

1. **SkillsTracker.LoadTest:**

This contains the Load Test Plan configured using JMeter whose instructions are given in the Readme.txt available in the folder.

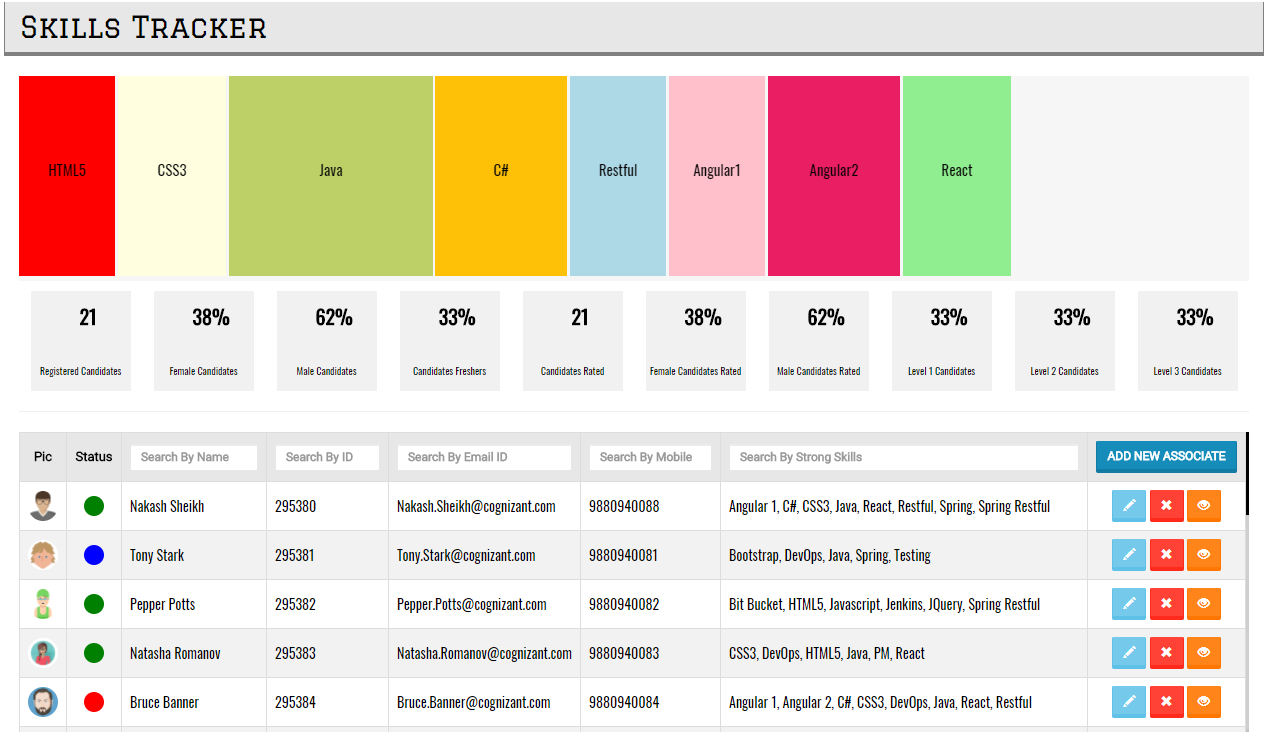
Skills Tracker – SPA:

Skills Tracker is a Single Page Application built with Angular 5 and Bootstrap. Prominence have been given to adhere to the coding standards, structuring it into different modules (shown below) which is quite self-explanatory.

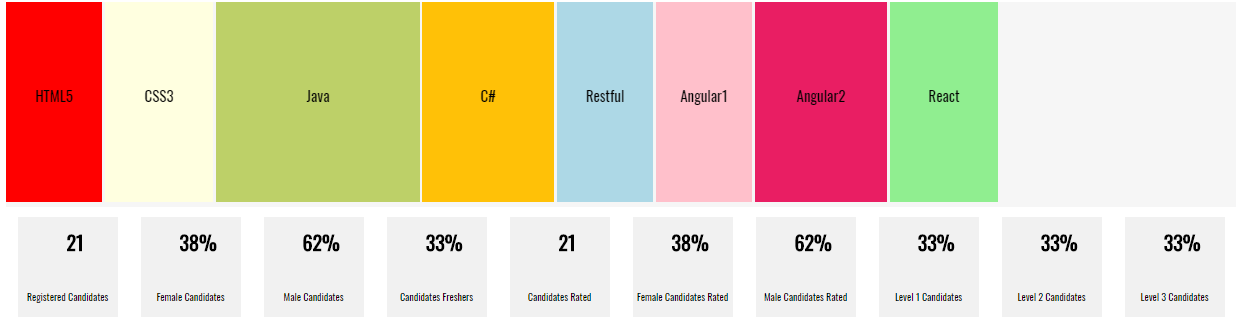


Working Screenshots of Skills Tracker:

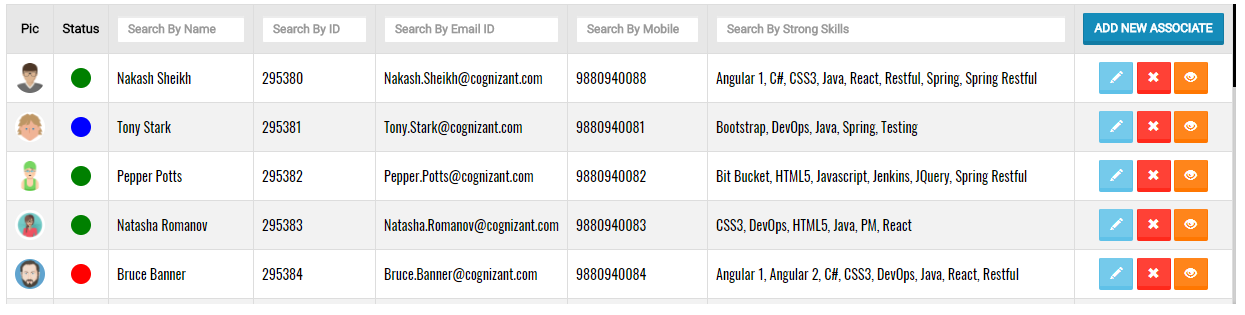
**Landing/Home Page**



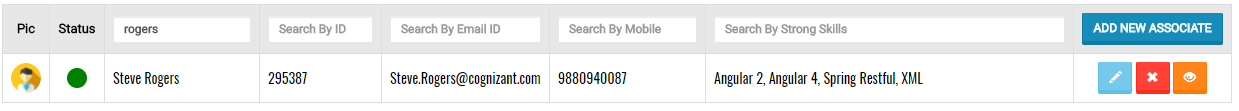
**Dashboard: (Built with HTML5 Canvas elements without using any 3rd party charts)**



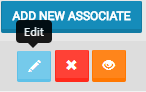
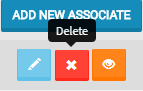
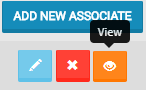
**Candidates Data Table: (Scrollable table with Picture and Status)**



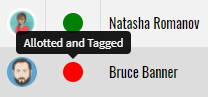
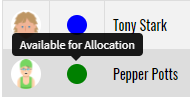
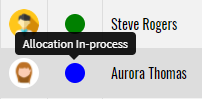
**Candidates Data Table: (Filter by each column)**



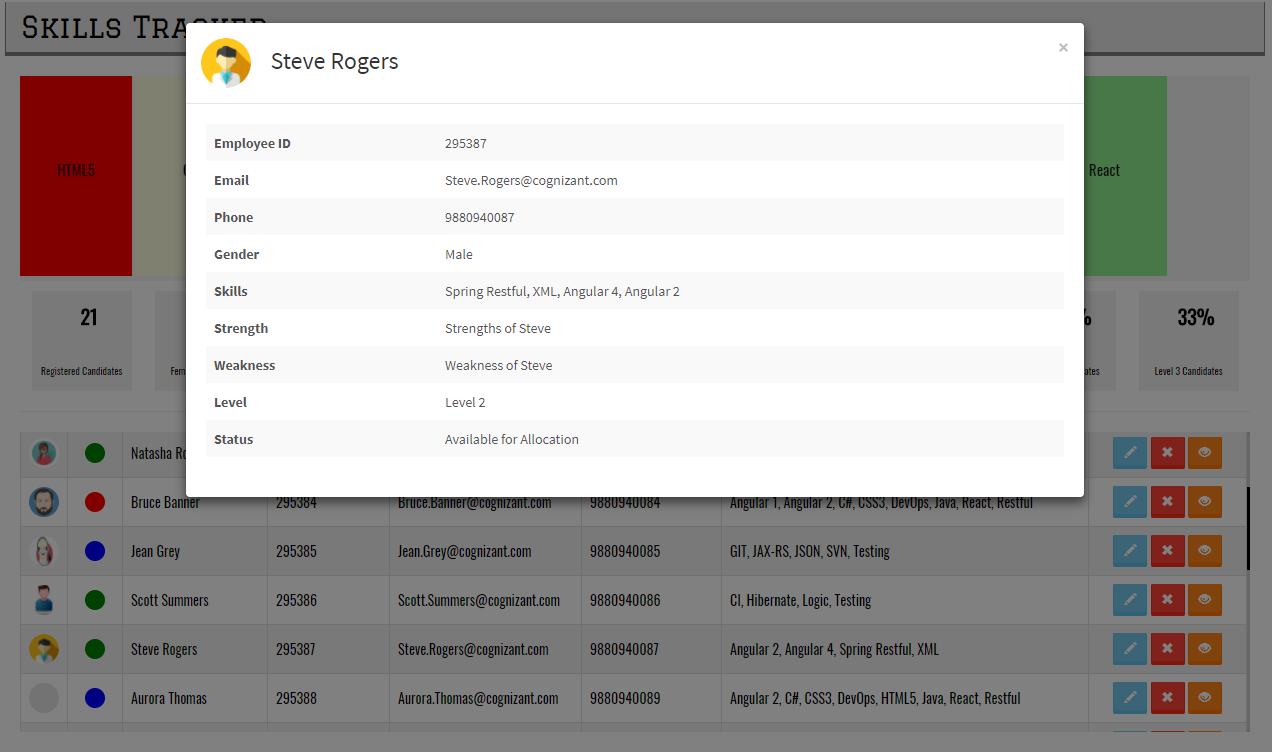
**Candidates Data Table: (Navigation buttons with tooltip)**

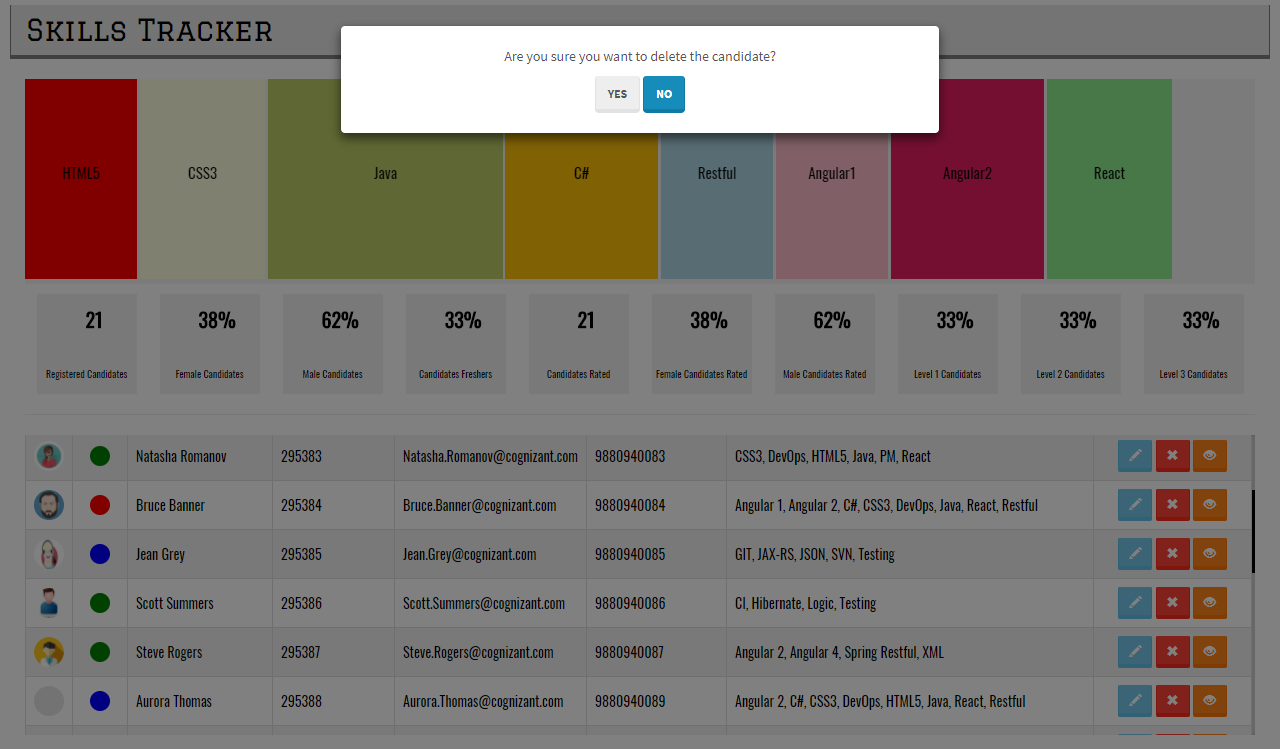
**Candidates Data Table: (Different status tooltips)**

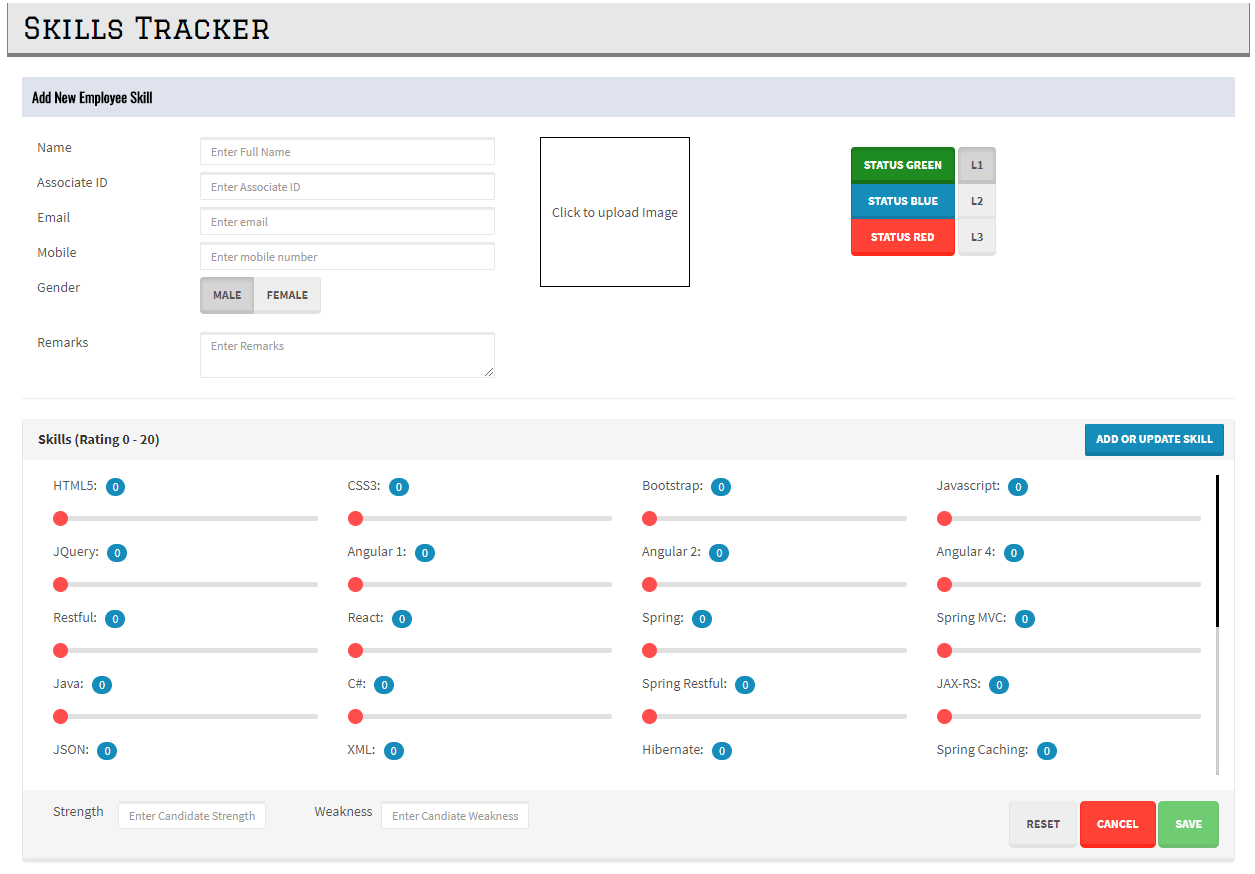
**Candidates Data Table: View (Candidate Data via Modal Pop-up)**



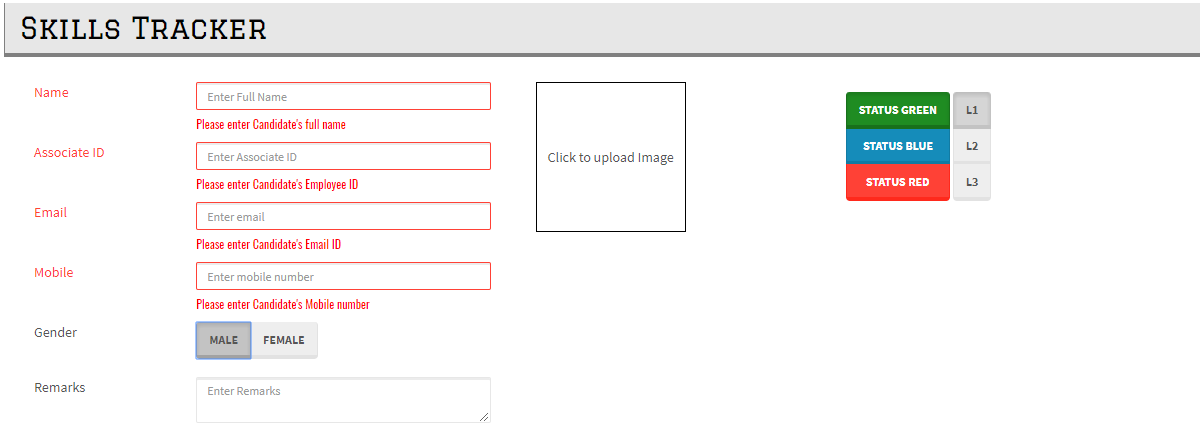
**Candidates Data Table: Delete with confirmation**



**Add New Employee Screen**



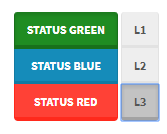
**Add New Employee Screen (Validation)**



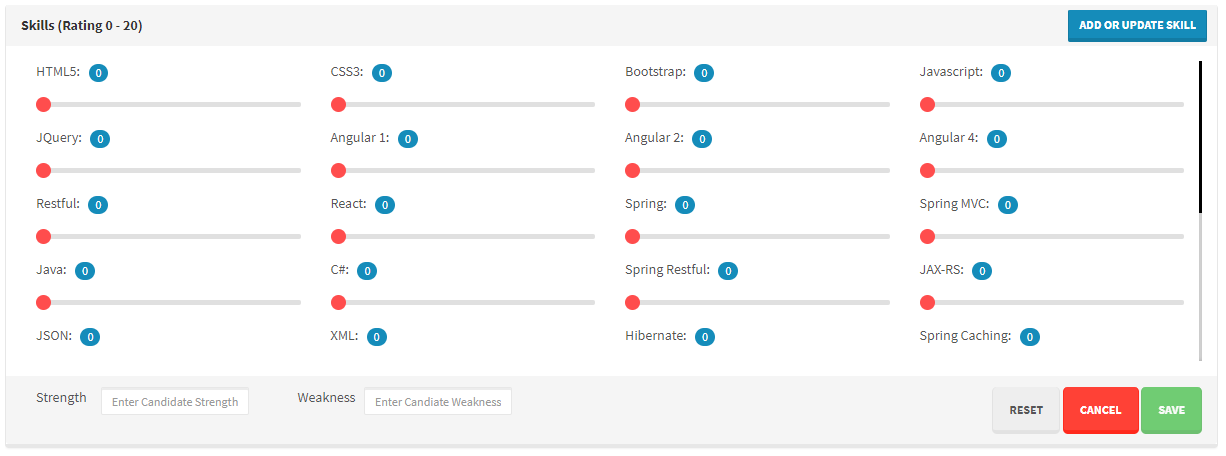
**Add New Employee Screen (Photo Upload)**



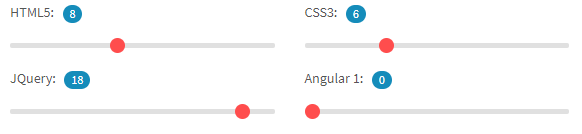
**Add New Employee Screen (Choose status and level)**



**Add New Employee Screen (Skill Selection)**



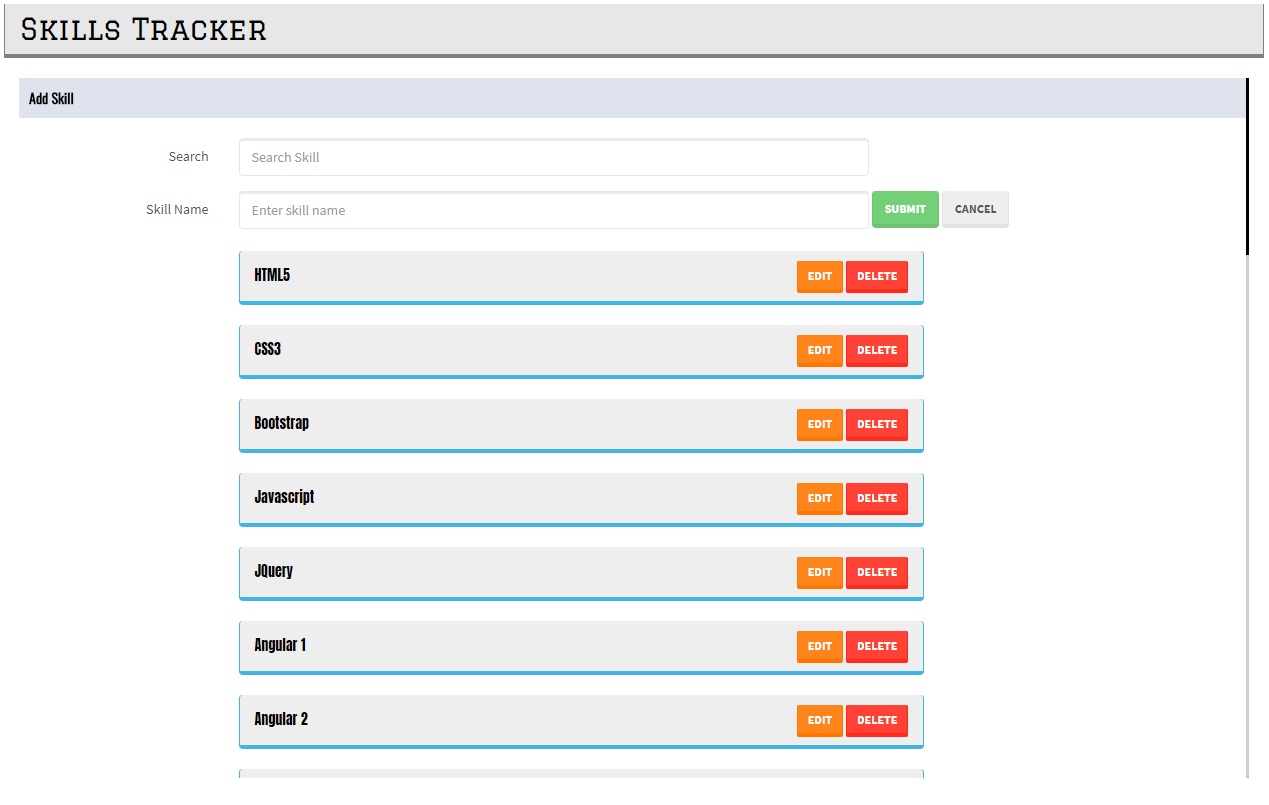
**Add New Employee Screen (Skill rating using HTML5 range slider with custom CSS)**



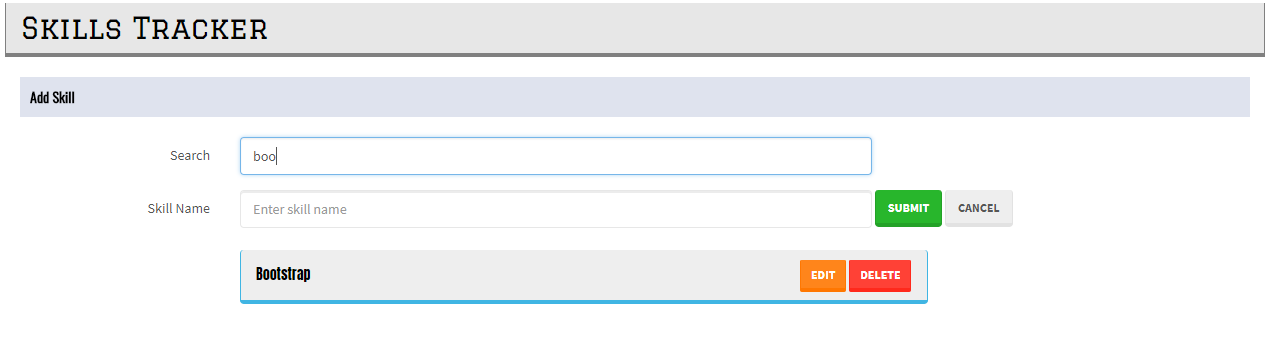
**Add New Employee Screen (Add or Update Skill button)**



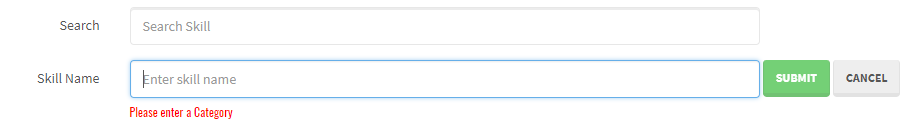
**Add New Skill Screen**



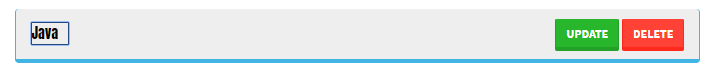
**Add New Skill Screen (Skill filter)**



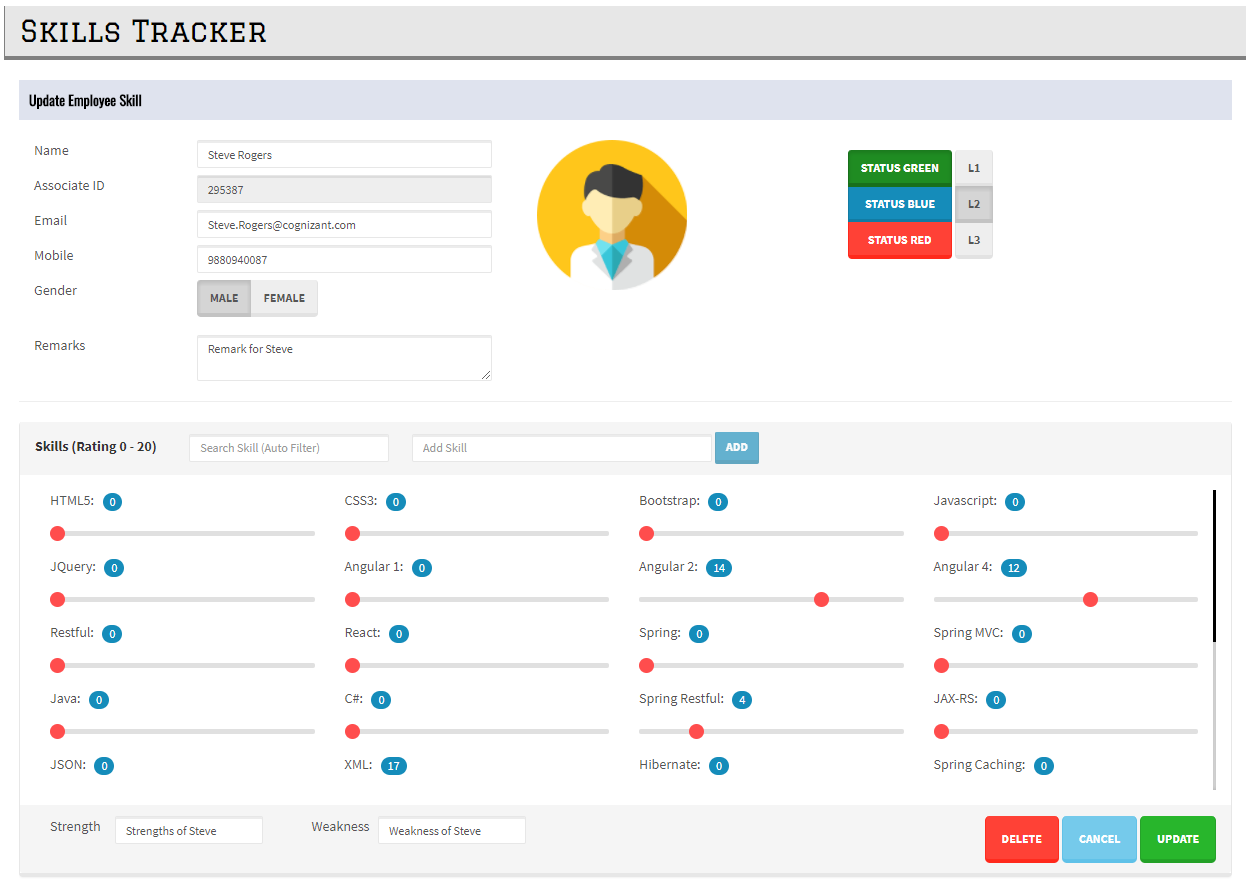
**Add New Skill Screen (Skill Name validation)**



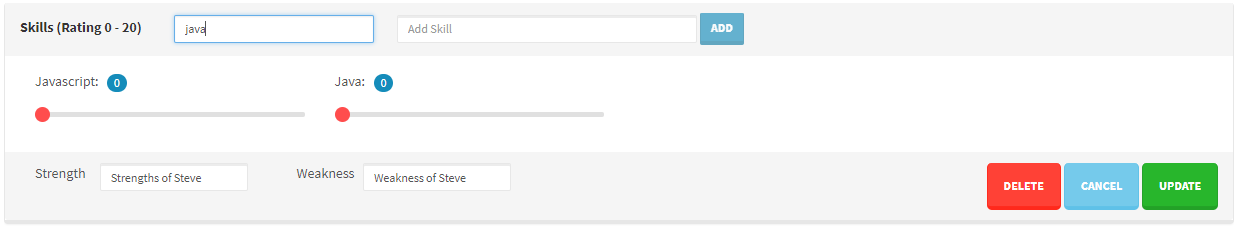
**Add New Skill Screen (Update skill via “contenteditable” attribute. Delete skill)**



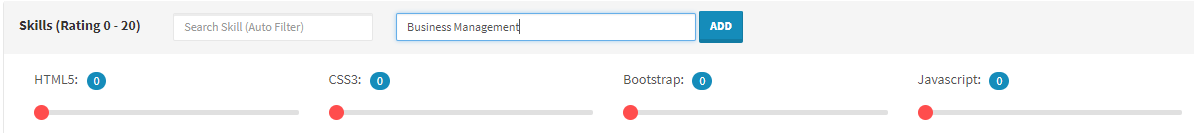
**Update Employee Skill Screen**

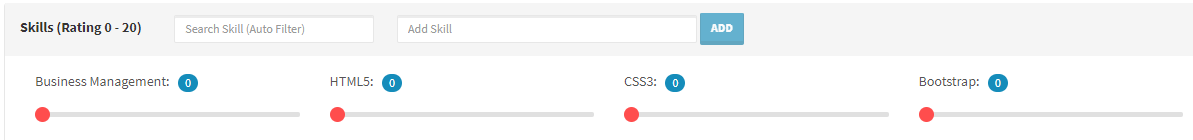


**Update Employee Skill Screen (with Skill filter)**



**Update Employee Skill Screen (adding new Skill)**





**Skills Tracker - Progress Bar (for each HTTP request made)**

