MOMENTON

Coding Challenge – Preface
(Thing to bear in mind) - Full Stack and Front End Developers

Besides clear comprehensive documentation and a clear outline of assumptions made,

- The code must be clean, following the best practices and standards (e.g. TDD)
- Should be well formatted, should be easy to follow the logic (e.g. proper choice of variable/function names, inline comments etc.),
- Should have README.md, proper test coverage etc., must have good choice of data structure, efficient use of memory, no unnecessary looping or inefficient code.
  (These are bare minimum.)

Candidates are encouraged to showcase everything they know e.g.:

- Utilisation of Design Patterns
- Frameworks – e.g. Angular
- Data structure knowledge
- Data access frameworks (Hibernate, JPA etc using in-memory databases), REST APIs, Spring Boot, Testing (JUnit, Mockito, PowerMock etc).

For the Front End Part..things of key interest are:

- Front-end communicating to backend service using Angular, JavaScript or any other framework;
- Unit testing using Karma, Jasmine, automated testing using protractor etc.

****

All candidates are expected to load share their code on:

(GitHub or Bitbucket). It is implicit that every developer is familiar with Git

****

Below is employee data of a small company.
It represents the hierarchical relationship among employees. CEO of the company doesn't have a manager.

Employee Name	**id**	**Manager id**
Alan	100	150
Martin	220	100
Jamie	150	
Alex	275	100
Steve	400	150
David	190	400

Design a suitable representation of this data. Feel free to choose any database (RDBMS, in-memory database etc), file system or even a data structure like List or Map. Then write code (in any language and framework) that displays the organisation hierarchy as below:

Jamie		
	Alan	
		Matin
		Alex
	Steve	
		David

The result can be simply displayed on the console, or HTML page or even a file; whatever suits you.
Try to cover all the possible scenarios, for example an employee with no manager, a manager who is not valid employee; etc.
Pay more attention on writing the actual logic of representing the employee tabular data into the hierarchical format.