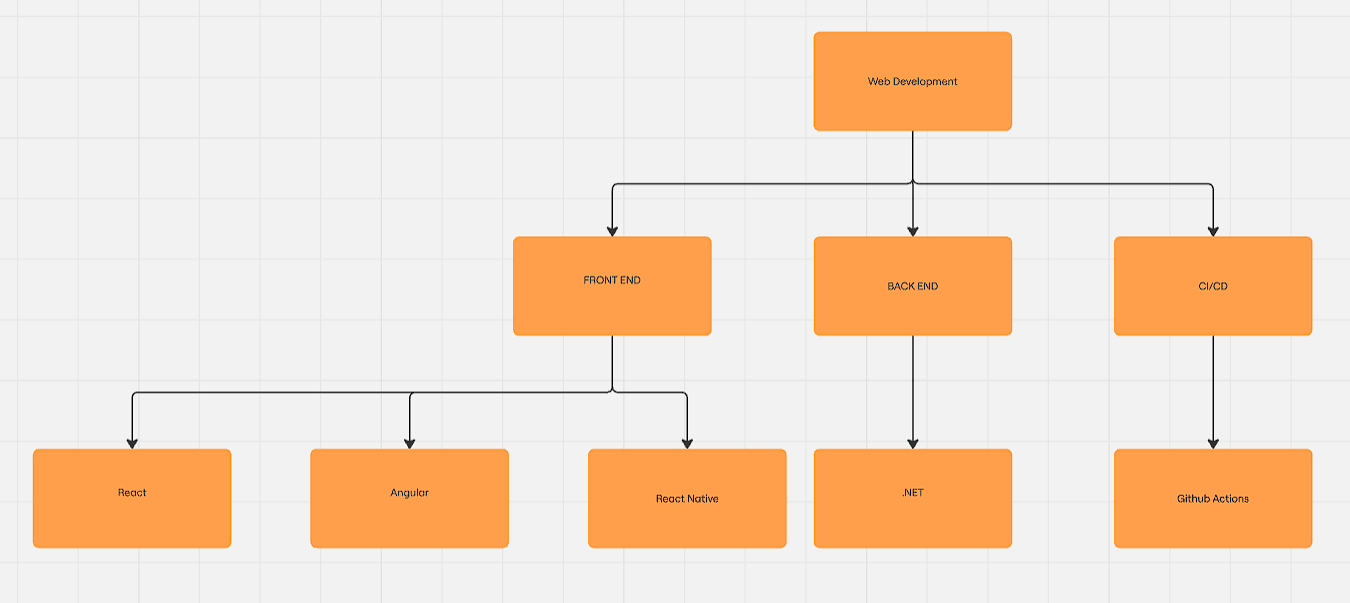
Keensight differentiates itself from competitors by offering personalized job matchings much more robust than simple keyword matching. This document describes how the matching feature is implemented.

A “skill tree” is a data structure used for every job posting and every user. It allows computation of compatibility between user and jobs based on it.

Below is an example of such a tree, the hierarchical relationship(and **ONLY** the relationship itself) between nodes are stored in tree.json, outside of the database. Additional info beside hierarchy are stored separately in database. They will reference the node in the hierarchy by name, and are referenced by specific user or jobs.



For each job posting, the additional info we store are a weight, a job requirement level, and a boolean indicating whether all child skills contribute to the parent skills, or if having any child skill is sufficient for the parent skill(for example, both front end and back end is important for web development, so comb=True for web dev, but having any framework can be considered good for front end, so comb=false for front end).

This info will be obtained with help from Large Language Model(LLM), we start with the topmost node, and ask these information based on its children and parent.

An example set of prompts:

-From a scale of 1 to 10, how much experience in <front end> does this job require?

-What percentage should <front end> contribute to the importance of <web development> in this role?

-The subskills of <front end> are <React> and <Angular>, based on the job description, is having either <React> or <Angular> sufficient or should a candidate have all?

