Project Analysis

# Project Goal

The objective of the system is to be a complete employment management solution, that would allow both employers and employees to interact with each other be it through posted jobs or through visiting each other’s profile.

## Target Users (Functionality Varied)

* Employee: People seeking jobs.
* Employer: Business Owners / Mangers seeking Employees.

## Functionality Presented to Users

**Employees:**

1. Register personal and technical information to a public profile.
2. Edit the previously mentioned profile.
3. View how many visitors checked his profiles.
4. Browse jobs that match his requirements to his profile.
5. Apply to jobs he is interested in. (A single button press applying).
6. Check the status on the jobs he applied to. (Accepted/Rejected/Pending)

**Employees:**

1. Register basic details to declare his status as employer on the site.
2. Edit the previously mentioned details.
3. Add job posts.
4. Edit his own job posts.
5. Delete his own job posts.
6. View the employees that applied to their job posts.
7. Change the status of employee application, informing if they were accepted or rejected.
8. Browse employee candidate that applied to his job post.
9. Browse possible employee candidates matching his requirements to his profile.

# Technologies to be used

## Data Storage

Data storage will be handled with some type of relational database. in the case of this prototype the DBMS that will be used is SQLite for rapid prototyping, and an ORM will be used to allow for future flexibility to change the database source to another type

The tools, in summery:

* The relational database SQLite
* Django ORM (built into Django)

## Back-End

The code running this project will be built using python, and I think a framework like django will be needed to show result data to webpages, and a similarity algorithm of sort to compare the post description and employee. All functionality will be presented to REST API end points, which any application can connect to.

The tools, in summery:

* Python Version 3.7
* Django Framework
* Django Rest API package
* Token based authentication
* Similarity Comparison Algorithm like spacy….

## Front-End

A web-based UI is the ideal way to go with this type of project, as it will allow to use some UI library for quickly getting a good interface together. There is of course many routes to take, but in this project I think react would prove to very useful. Using Fetch API to call to the Backend Django server through rest API calls.

The tools, in summery:

* React
* React Dom Router (to manage navigation)
* Material UI library (Clean and elegant react component to build an application with)

# Database ERD Model

