Components of the application:

1. Frontend

2. Database

3. API

Frontend Work done till now:

Screens created so far:

Database Work done till now:

1. USER\_LOGIN\_INFO: This table gets created whenever user signs up for the first time and stores his/her login credentials.

user\_id

password

login\_type

Notes:

● Password is the encrypted field

● login\_type is enum (mobile, facebook, gmail or linkedin)

● This table is same for all the candidates or referrers

2. USER\_PROFILES: This stores all the singular data about the user, and gets updated as and when we get details during the onboarding process.

user\_id

name

referrer/candidate

company\_id

email\_id

mobile

current\_location

gender

age

current\_role

total\_experience

minimum\_pay

availability

Note:

● This table is same for all the users (both the candidates and referrers)

3. USER\_PREFERRED\_LOCATIONS

4. USER\_SKILLS

5. LOCATIONS: This is a hardcoded table of all the locations that we provide our service in.

6. ROLES: Hardcoded table of all the roles that we are aware of. Role is the same thing as position or work profile in the company and is specific. For example, DevOps Engineer, Backend Engineer-1, Backend Engineer-2 are separate roles.

role

7. ROLES\_BY\_ROLE\_CLASS: Hardcoded roles to role class mapping

Note: There is no need to provide ‘id’ instead of strings in the map tables. These are hardcoded

values and we want tables to be readable as well. Using foreign integer keys instead of string foreign keys provide no additional value and make the entire db look encrypted for no reason. That is why I keep ‘location’ string in the USER\_PROFILE table instead of location and so on There are only finite set of locations and roles anyway.

10: SKILL\_BY\_SKILL\_CLASS

11. USER\_WORK\_EXP:

user\_id

company\_id

from\_date

to\_date

role

12. USER\_EDU\_EXP

user\_id

college\_name

from\_date

to\_date

degree

13. COLLEGES

college\_id

college\_name

tier\_class

14. DEGREES

15. COMPANIES: To be prefilled and hardcoded.

company\_id

company\_name

location

website

logo

founded\_date

business\_model

company\_size

company\_type

industry

16. COMPANY\_COMPETITORS

17. JOBS

job\_id

referrer\_id

company\_id

min\_experience

max\_eperience

max\_expectation

current\_role

gender

availability

skills

location

qualification

Note: referrer\_id is same as the user\_id in the USER\_PROFILES table

18. JOBS\_ACCEPTED\_COLLEGES: This is created when the filter explicitly mentions the colleges.

19. JOBS\_ACCEPTED\_PREVIOUS\_EMPLOYERS: This is created when the filter explicitly mentions the previous employers.

20. JOBS\_ACCEPTED\_SKILLS: What are the skills that this job or referral requires?

21. JOBS\_ACCEPTED\_LOCATIONS: A job could be applicable for multiple locations.

22. JOBS\_ACCEPTED\_QUALIFICATIONS. As again, degree value must match from the DEGREES table.

job\_id

degree\_id

23. JOBS\_ACCEPTED\_AVAILABILITY

job\_id

availability

24. MATCHES: This indicates what companies and then referrers are shown to the candidate:

candidate\_id

job\_id

referrer\_id

company\_id

match\_score

location

25. MATCHES\_PER\_COMPANY: This is derived from the above table. Could be a VIEW for quick access. (select candidate\_id, company\_id, sum(match\_score) from MATCHES group by candidate\_id, company\_id order by sum(match\_score) desc). Score is something which will be from job\_id and user\_id on the basis of availability, experience, college, skills and all other criteria. Score will be more if referrer is the alumni.

candidate\_id

company\_id

total\_score

26. REFERRAL\_REQUESTS:

referrer\_id

candidate\_id

job\_id

applied\_date

company\_id

role

current\_status

27. REFERRAL\_STATUS\_LOG

referral\_id

date

status

28. REFERRAL\_STATUSES: Hardcoded value of what can be the value for referral status

referral\_status\_id

referral\_status

29. REFERRER\_SCORE: How good is a referrer for a particular role class? Star rating.

referrer\_id

role\_class

score

30. USER\_SEARCH\_PARAMS: This describes what kind of matches we have been looking for the candidate.

user\_id

role\_class

location

API Work done till now:

Work needs to be done.

Components of the application:

1. Frontend

2. Database

3. API

Frontend Work done till now:

Screens created so far:

Database Work done till now:

1. USER\_LOGIN\_INFO: This table gets created whenever user signs up for the first time and stores his/her login credentials.

user\_id

password

login\_type

Notes:

● Password is the encrypted field

● login\_type is enum (mobile, facebook, gmail or linkedin)

● This table is same for all the candidates or referrers

2. USER\_PROFILES: This stores all the singular data about the user, and gets updated as and when we get details during the onboarding process.

user\_id

name

referrer/candidate

company\_id

email\_id

mobile

current\_location

gender

age

current\_role

total\_experience

minimum\_pay

availability

Note:

● This table is same for all the users (both the candidates and referrers)

3. USER\_PREFERRED\_LOCATIONS

4. USER\_SKILLS

5. LOCATIONS: This is a hardcoded table of all the locations that we provide our service in.

6. ROLES: Hardcoded table of all the roles that we are aware of. Role is the same thing as position or work profile in the company and is specific. For example, DevOps Engineer, Backend Engineer-1, Backend Engineer-2 are separate roles.

role

7. ROLES\_BY\_ROLE\_CLASS: Hardcoded roles to role class mapping

Note: There is no need to provide ‘id’ instead of strings in the map tables. These are hardcoded

values and we want tables to be readable as well. Using foreign integer keys instead of string foreign keys provide no additional value and make the entire db look encrypted for no reason. That is why I keep ‘location’ string in the USER\_PROFILE table instead of location and so on There are only finite set of locations and roles anyway.

10: SKILL\_BY\_SKILL\_CLASS

11. USER\_WORK\_EXP:

user\_id

company\_id

from\_date

to\_date

role

12. USER\_EDU\_EXP

user\_id

college\_name

from\_date

to\_date

degree

13. COLLEGES

college\_id

college\_name

tier\_class

14. DEGREES

15. COMPANIES: To be prefilled and hardcoded.

company\_id

company\_name

location

website

logo

founded\_date

business\_model

company\_size

company\_type

industry

16. COMPANY\_COMPETITORS

17. JOBS

job\_id

referrer\_id

company\_id

min\_experience

max\_eperience

max\_expectation

current\_role

gender

availability

skills

location

qualification

Note: referrer\_id is same as the user\_id in the USER\_PROFILES table

18. JOBS\_ACCEPTED\_COLLEGES: This is created when the filter explicitly mentions the colleges.

19. JOBS\_ACCEPTED\_PREVIOUS\_EMPLOYERS: This is created when the filter explicitly mentions the previous employers.

20. JOBS\_ACCEPTED\_SKILLS: What are the skills that this job or referral requires?

21. JOBS\_ACCEPTED\_LOCATIONS: A job could be applicable for multiple locations.

22. JOBS\_ACCEPTED\_QUALIFICATIONS. As again, degree value must match from the DEGREES table.

job\_id

degree\_id

23. JOBS\_ACCEPTED\_AVAILABILITY

job\_id

availability

24. MATCHES: This indicates what companies and then referrers are shown to the candidate:

candidate\_id

job\_id

referrer\_id

company\_id

match\_score

location

25. MATCHES\_PER\_COMPANY: This is derived from the above table. Could be a VIEW for quick access. (select candidate\_id, company\_id, sum(match\_score) from MATCHES group by candidate\_id, company\_id order by sum(match\_score) desc). Score is something which will be from job\_id and user\_id on the basis of availability, experience, college, skills and all other criteria. Score will be more if referrer is the alumni.

candidate\_id

company\_id

total\_score

26. REFERRAL\_REQUESTS:

referrer\_id

candidate\_id

job\_id

applied\_date

company\_id

role

current\_status

27. REFERRAL\_STATUS\_LOG

referral\_id

date

status

28. REFERRAL\_STATUSES: Hardcoded value of what can be the value for referral status

referral\_status\_id

referral\_status

29. REFERRER\_SCORE: How good is a referrer for a particular role class? Star rating.

referrer\_id

role\_class

score

30. USER\_SEARCH\_PARAMS: This describes what kind of matches we have been looking for the candidate.

user\_id

role\_class

location

API Work done till now:

Work needs to be done.