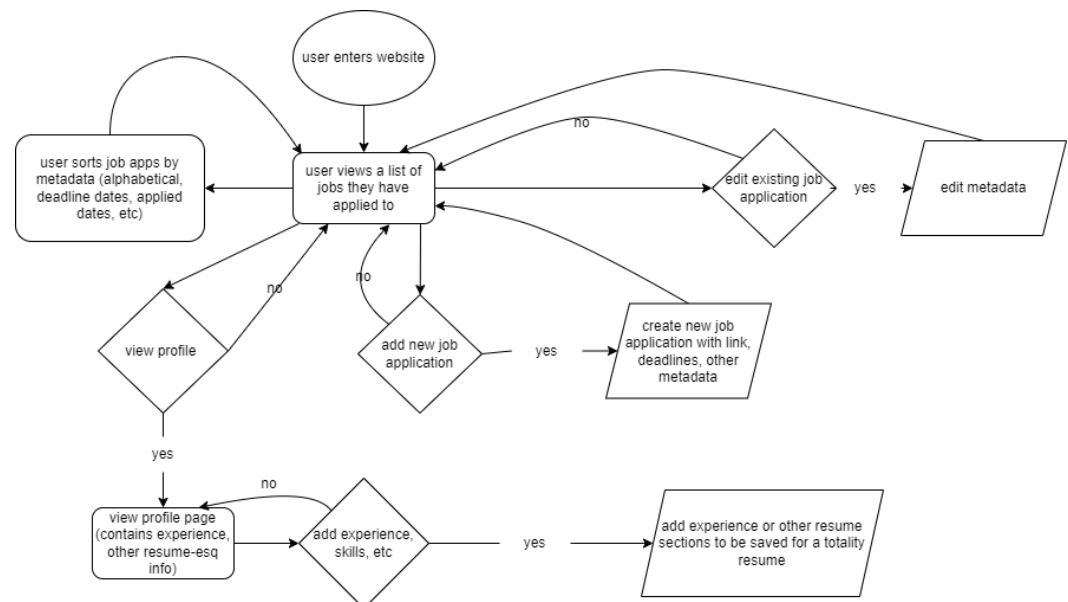Team 24 Brainstorm

Division of roles:
- frontend: Isabelle, Eric, Jin, Pierre
- backend: Jessalyn, Alex, Hayk, Estella
- design: Srivasta, Isabelle, Eric

1. Job application management system
   a. Problem: it is hard to keep track of what jobs one has applied to, and the different results that were received from each application. For example, multiple job application websites such as LinkedIn route to a third party application portal, and it can be hard to keep track of them all.
   b. Personas: people who are looking for jobs, specifically those who do not have a lot of experience and may need to apply to a large volume of jobs before they are able to get a couple offers.
   c. User flow

      i.
   d. User interview
      i. Pain points of job application process
         1. Some jobs have third party job app portal, annoying there is no quick apply
         2. Saving the job listing would be nice, in case companies delete it later
         3. Job application search is exhausting, application manager would help
         4. Hard to manage volume of applications
   e. Features:
      i. Contact information: Recruiters/Hiring Managers

1. This is useful to users who like to reach out to recruiters and apply. They can document the contact info and when they last reached out to decide whether to reach out again
   ii. Link to third part app portal + username and password used for the portal
      1. This is useful for keeping track of all the third party portals and the accounts made for them
   iii. Deadlines for applications
      1. If a user has not yet applied to a job but wants to, listing deadlines for applications will allow users to sort by deadline, and prioritize apps that have a closer deadline
   iv. Interview dates and times and topics, company principles
      1. Reminder of upcoming interview dates or other deadlines, when dealing with a lot of assessments at once and other obligations it can be difficult to keep track of all the deadlines
   v. Job description and title
      1. Useful for distinction between jobs
   vi. Offer information for comparison/leveraging + deadline
      1. Useful for managing offer numbers and offer deadlines to help plan courses of action for when a user is in progress with some companies but has pending deadlines
   vii. Ability to tag applications in the system so can filter by target companies, application type (needs resume, needs short answer questions, etc)
      1. Useful for when wanting to view a specific subset of applications
   viii. The ability to save information on your entire experience (totality resume)
      1. Many people have to pick and choose what to put on their resume and depending on the job application they will choose to showcase different skills. The ability to save the bullet points of information on past experiences to construct a job specific one page resume is nice
      2. In the future can have experience recommendation system (recommending what experience to choose for the one page resume) based on job description
   ix. Personal notes
      1. For example what went well, went poorly for each app
      2. Another example is notes on personal anecdotes that match company principles or what skills the company is looking for
      3. Useful for keeping track of how applications went for future improvements
   x. Leetcode integration
      1. Stretch goal
      2. Specifically for CS major related users, different leetcode questions can be recommended based on company or student needs (perhaps a student needs help with dynamic programming questions)

2. Alt idea 1: Recipe management system
    a. Problem: it is hard to keep track of recipes, when doing recipe experimentation it is nice to have a permutation of recipes, figure out what recipes exist based on contents of fridge
    b. Users: people cooking for themselves and others, avoiding food waste
    c. Features:
        i. Create a recipe with ingredients and cooking steps (and maybe photos)
        ii. Add multiple versions of recipes which is good for recipe experimentation and alternatives
        iii. Determine if a recipe is feasible based on fridge ingredients
        iv. Set ingredients which can be substituted by other ingredients
        v. Easily convert measurements between different units
        vi. Convert between volume and mass
        vii. Easily scale recipe to different final yields (e.g increase from 5 to 7 servings)
        viii. Smart fridge integration :0
        ix. Timer for when someone is cooking so we can send reminders that is triggered based on pressing done only so ppl don't have to input the numbers if they don't want to
        x. Share recipes with others
        xi. Better collaboration for multiple people who cook in the same kitchen
3. Alt idea 2: school schedule management system
    a. Problem: it is annoying to keep making school schedules every quarter and keep track of assignments
    b. Users: students
    c. Features:
        i. Prof or course staff create and schedules with oh, lecture, tutoring, exam dates, assignment dates
        ii. Prof or course staff update or delete dates if they change
        iii. Students create accounts and select what classes they are taking, and an aggregate class schedule of info across all courses is made
            1. Just add all things to a list and sort by date
        iv. Students can choose to delete things off their schedule such as extraneous oh
        v. Students can choose notification preferences
4. Alt idea 3: Syntax aggregator
    a. Problem: it's annoying to look up syntax all the time esp when using multiple frameworks and or languages since they are not in one place
    b. Users: developers/coders
    c. Features
        i. Upload code blocks with tags so they are easily searchable to app
            1. So it is easy to find again in the future, nice for ppl who work in companies with very specific coding practices
        ii. Copy and paste code blocks

iii.   Maybe have a page for the collective to upload syntax blocks so ppl can search without going to google

iv.   Note this is syntax not algorithm/heavy implementation so things like how to use the state hook in react in general rather than how to use for one specific purpose

v.   Can have notes on best practices such as when to use a feature or when not

5.   Alt idea 4: Tree-based notetaking

    a. Problem: It can be difficult to take notes in a single linear document

    b. Users: students who want class notes that are easier to study off of and search through

    c. Features

        i.   Add topics in a hierarchy format

        ii.   Add written notes, images, sketches, clips from videos to specific topics

        iii.   Quickly search through notes to find what you are looking for

        iv.   Link to tree nodes from other classes with similar material