Perfect Fit Resumes/CVs

Summary

Perfect Fit Resumes/CVs is a web application designed to help job seekers create highly customized resumes and CVs that perfectly align their unique experiences and skills with the specific requirements of each job application. Using Hugging Face Meta Llama 3 API, the platform takes in personal career history, relevant skills, accomplishments, and qualifications to generate tailored documents. These customized resumes and CVs are optimized to successfully pass through automated resume parsers, stand out to hiring managers, and maximize the chances of securing job interviews.

Organization

The frontend files are located in the "react-app" folder. Within this folder, the "public" directory holds the favicon, while the "src" directory contains the rest of the pages and pictures. The "assets" folder within "src" stores all the pictures needed for the frontend, and the "pages" folder includes all the different website pages for the frontend. Each page has its own corresponding CSS file named after the page itself. For example, the CSS elements for "Favorites.jsx" are found in "Favorites.css." Additionally, "index.css" defines the general theme of the website, including the color scheme and fonts. The "main.jsx" file serves as the core page that integrates and routes all the other pages, making the website function properly.

The backend files are located in the "src" folder. The "test" directory contains a variety of text files, pictures, tasks, and test cases for each sprint. The main logic for handling requests between the frontend and backend is found in "main.py," allowing you to run the code with a single "python main.py" command instead of managing multiple files. Each function in this file is separated based on the different requests users might make, and they are all properly commented. The database used by the backend is "test.db," and the "testDB.py" file is responsible for creating the database tables initially, so there's no need to recreate them each time.

The "Burndown" folder contains images that are needed for the "README.md" file.

Maybe this layout is clearer:
Frontend is in “react-app” folder

- “public” holds the favicon
- “src” holds the rest of the pages and pictures
    - “assets” holds all the pictures needed for the frontend
    - “pages” holds all the different website pages needed for the frontend
        - Each page’s css element is under the same name as the page
        - Ex: Favorites.jsx CSS’ elements will be under Favorites.css
- “index.css” holds the general theme of the website (color scheme, fonts, etc)
- “main.jsx” is the main core page that combines and routes all the pages together to make it work

Backend is in “src” folder:

- “test” contains a random assortment of text files and pictures, and tasks and test cases for each sprint
- main.py is where the code to handle all the requests between the frontend and backend is
    - So you can just do “python main.py” instead of having to do multiple files
    - Here, each function is separated for the different requests that the user may need. They are properly commented
- test.db is the database we are accessing and using
- testDB.py is is used to create the database tables in the first place
    - So we won't have to meddle with creating the tables or recreating again each time

Others:

- “Burndown” folder contains images needed for the README.md

Technologies

- Python
- ReactJS
- SQLite
- Hugging Face Meta Llama 3 API
- HTML/CSS
- Axios/Flasks

Requirements:

- Git
- Node.js
- Npm

Technologies' reasoning:
We thought that Python, ReactJS, SQLite, Axios/Flasks, Git, Node.JS, npm are all common tools that are widely used in the industry. We are quite familiar with those technologies and believe that they would be easy to use and simple enough to make the project work.

We explored several different AI APIs but ended up choosing Hugging Face Meta Llama 3 API because they are a free open source API. We tested with the Hugging Face Meta Llama 3 API first and it seems good enough and so we went with it without testing other options further.

Instructions:
Get your own AI API key:
Get the AI API key by creating an account and request to use it from:
https://huggingface.co/meta-llama/Meta-Llama-3-8B

Instructions to apply the API: https://huggingface.co/docs/huggingface_hub/v0.14.1/en/guides/inference.
At line 12 in main.py, HG_NAME is replaced with the API name and HG_API_KEY is the key. We're not granting our API freely for another client.

Client setting up after obtaining the API KEY:

1. Install dependencies:	Frontend: cd ../react-app npm install Backend: pip install inference-client pip install python-dotenv
2. Start backend:	cd ../src Windows: python main.py Mac: python3 main.py
3. Start frontend:	cd ../react-app npm run dev
4. Access the website on:	http://localhost:5173/

User instructions:
To get a generated resume or cover letter, go to the home page and do the following:

1. Go to home page by pressing home or the top left "logo"
2. Specify what you want from the dropdown menu then press ok
3. Either paste or upload your resume/cover letter and press ok
4. Either paste the URL of the job description or the job description itself
5. Enjoy your tailored resume/cover letter!

Evaluation:

1. How to perform tests:
    a. Start server
    b. Make a new account
        i. Try weak passwords to see if password strength is enforced
    c. Login
        i. Test wrong passwords
    d. Edit user information and filled out all the necessary fields
    e. Follow the user instructions above and try every option in dropdown menu
    f. Favorite a sample resume and cover letter
    g. Next visit the favorites page and view favorites and try to export from there
    h. Log out
2. Results from tests:
    a. Should load no issue
    b. A new account should be made with the profile page being able to verify that
    c. Login should work for that account and wrong passwords should be denied
    d. Every option in dropdown menu should function as intended and lead to a tailored product
    e. Favoriting should save the current product in the favorites page
    f. Visiting favorites page and exporting a product should be available
    g. Logging out should properly sign out the profile.
3. Strengths and weaknesses:
    a. Strengths:
        i. Simple interface with effective results
        ii. Successfully incorporate an AI API and a database with little issues
        iii. Relatively well organized and structured code for further developments
    b. Weaknesses:
        i. Exporting can be unintuitive. The criteria for it is not clear to a brand new user and can be something to work on
        ii. Only PDF files can be uploaded and exported
        iii. UI could definitely be improved still. We did what we thought was the best but we're just a bunch of overworked and underpaid "software developers"
        iv. There could be more comments of all the functions of the code
        v. The naming convention could still be better
        vi. Most input fields aren't strict enough and can leave room for users to mess around and create unplanned results. Present in most features (AI generating resume/cover letter, profile information)
        vii. Frontend programming could still be improved to be more adaptive. Some were brute forced, i.e. using "absolute" and hard coding the location.
4. Next Steps:
    a. Upload and export multiple different types of files
    b. Be able to export resumes
    c. A lot more restrictions with each field inputs to better guide the user. For example, not having "123" as the phone number or inputting nonsense for resume and job description to generate a resume/cover letter.
    d. Improve frontend code for less brute force
    e. Figure out a better organization or structure. There's always room for improvement.
    f. New features such as a job board that can show available jobs and allow the AI to generate a cover letter with each job based on profile's information. That would speed up the job application process significantly. This would then open more opportunities for more features.

User Stories Demo Video:
User Story 1: https://www.youtube.com/watch?v=-VNeujB2NqM
https://docs.google.com/presentation/d/1Vcs1l0Zy-c9IGaobR30qeIcbFNIne5ml-57R-PVzHCo/edit#slide=id.p

User Stories 2-4: https://youtu.be/HUee8ud0N0Q
https://docs.google.com/presentation/d/126BASXZ-Qi_w4PdXGO6X_Olu04vtTevV8Vy-oi-KpQM/edit#slide=id.p

User Story 5: https://youtu.be/SAtNrb5cmEw
https://docs.google.com/presentation/d/1WZC-5HbBXLuJpmUi9wr3irVmaCwATPkDKoen6nYbqEk/edit?usp=sharing

User Story 6: https://youtu.be/ctOKkVn-M_8
https://docs.google.com/presentation/d/1qea4qjR_yBEIL_s49PNmNxYBt15IB5r09KcBsRjjb3A/edit?usp=sharing

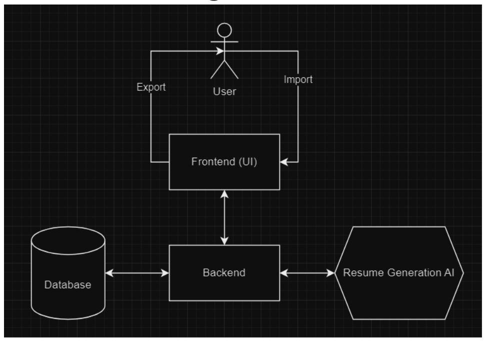
User Story 8: https://www.youtube.com/watch?v=xgNvV0sGZAs&t
https://docs.google.com/presentation/d/1b2H2twQWx2uUVGl3yoK7L4rpwahbzYRHb6RtNDqsfig/edit?usp=sharing

User Story 9: https://www.youtube.com/watch?v=EpQBcx6nado
https://docs.google.com/presentation/d/1nPdHOXk5KBnLMe0RbyF8esSt06oqULPW8noBOFQIJD8/edit?usp=sharing

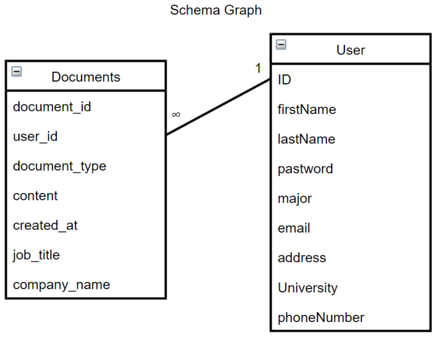
User Story 10: https://www.youtube.com/watch?v=WtFugzIVRYk
https://docs.google.com/presentation/d/1TpWReWOJ5aqkpIcQCOCAEhkmsNZ4AWIsgdTgoC-6mgc/edit?usp=sharing

User Story 11-15: https://youtu.be/cpvHa5CWx6k

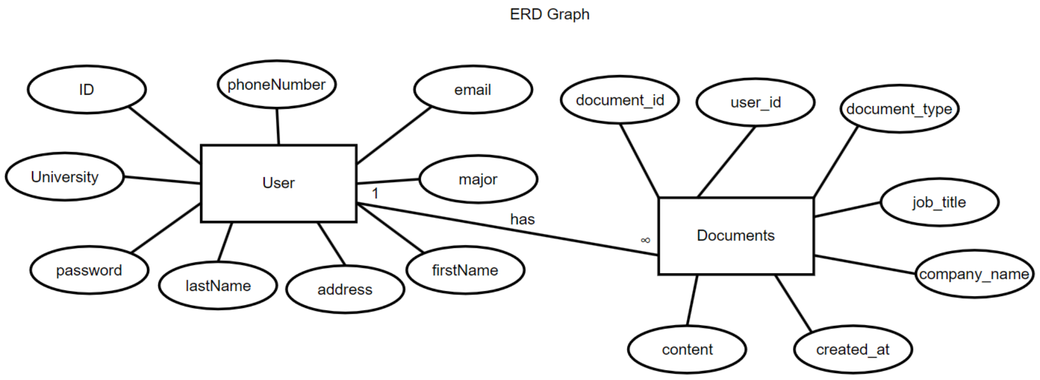
Architecture Diagram



Schema Graph



Database Schema


ERD Graph
ID
phoneNumber
email
University
User
major
password
lastName
address
firstName
1
has
∞
Documents
document_id
user_id
document_type
job_title
company_name
content
created_at