

CS673 Software Engineering
Team 5 - CVCoach
Project Proposal and Planning

Your project Logo
here if any

<u>Team Member</u>	<u>Role(s)</u>	<u>Signature</u>	<u>Date</u>
Linchen Xu	Team Leader, Design and Implementation Leader	<u>Linchen Xu</u>	<u>09/14/2024</u>
Lin Ma	Configuration Leader	<u>Lin Ma</u>	<u>09/14/2024</u>
Zihan Zhou	Security Leader	<u>Zihan Zhou</u>	<u>09/14/2024</u>
Zhen Cao	QA Leader	<u>Zhen Cao</u>	<u>09/14/2024</u>
Haochen Sun	Requirement Leader	<u>Haochen Sun</u>	<u>09/19/2024</u>

Revision history

<u>Version</u>	<u>Author</u>	<u>Date</u>	<u>Change</u>
<u>1.0</u>	Linchen Xu Lin Ma Zihan Zhou Zhen Cao Haochen Sun	<u>09/14/2024</u>	<u>Create Initial Document</u>
<u>2.0</u>	Linchen Xu Lin Ma Zihan Zhou	<u>10/16/2024</u>	<u>Update real time for iteration1 development</u>

	Zhen Cao Haochen Sun		
<u>3.0</u>			
<u>4.0</u>			

[Overview](#)

[Related Work](#)

[Proposed High level Requirements](#)

[Management Plan](#)

[Objectives and Priorities](#)

[Risk Management \(need to be updated constantly\)](#)

[Timeline \(need to be updated at the end of each iteration\)](#)

[Configuration Management Plan](#)

[Tools](#)

[Deployment Plan if applicable](#)

[Quality Assurance Plan](#)

[Metrics](#)

[Code Review Process](#)

[Testing](#)

[Defect Management](#)

[References](#)

[Glossary](#)

1. Overview Linchen Xu

(Please give an overview of your project. It should include the motivation, the purpose and the potential users of the proposed software system, the basic functionality of the proposed software system and the possible technology stack to be used.)

a. Motivation:

With the increasing need in the job market, both job seekers and recruiters face challenges in finding the right match. It is reported that recruiting a right candidate for a job position can cost \$4,700 (ExtensisHR 2023). Job seekers are struggling to tailor their resumes for specific roles, and hiring managers have to review numerous resumes to find a qualified candidate. Interviewers have to spend a lot of time preparing questions for the interview.

The motivation for this project is to use AI to automate and improve the process of resume evaluation and interview preparation. By providing personalized resume suggestions and interview question generation, the platform helps candidates present themselves better and makes the job search process more efficient. The

platform also helps hiring managers to filter resumes for better match and helps interviewers for preparing for questions for the interview.

b. Purpose:

The purpose of this AI resume analysis application is to assist job seekers in refining their resumes and preparing for job interviews, as well as to help hiring managers assess candidate resumes. By leveraging AI, the system will streamline the process of resume evaluation, matching job seekers with relevant job descriptions and providing valuable feedback for improvement. The system aims to reduce the time and effort spent by both job seekers and recruiters in the job application process.

c. Potential users:

- Job seekers who are seeking job opportunities.
- Hiring managers who are exhausted by filtering resumes.
- Interviewers who are troubled by preparing to interview candidates.

d. Basic functionality:

The AI resume analysis application will allow users to upload their resumes and interact with the system for several purposes:

- Resume evaluation against specific job descriptions.
- Suggestions for improving resumes to better match job requirements.
- Interview question generation based on the resume and job description.
- Recommendations for relevant projects or skills to enhance the resume.

e. Tech stack:

- **Front-end:** React, TypeScript
- **Back-end:** Python (with AI/ML libraries)
- **Database:** PostgreSQL or MongoDB
- **Cloud services:** AWS or Google Cloud for deployment and storage
- **Version control:** Git (GitHub/GitLab)

2. Related Work @All

(Please describe any similar software systems that you have found through the online research, and the differences between your software and those software systems.)

1. [Jobscan](#)

Jobscan offers resume optimization by analyzing a resume and comparing it to a specific job description, focusing on keyword matching to help job seekers improve their chances of passing applicant tracking systems (ATS).

Differences:

While Jobscan mainly focuses on keyword matching and ATS optimization, our application goes beyond that by also providing personalized suggestions on improving resumes, generating interview questions, and offering project recommendations to match job descriptions.

2. [resume.co](https://www.resume.co/)

ResumeCo uses a series of prompts to help users build their resumes and also evaluate it based on the completeness of the resume. Users can also upload their existing resume and modify it based on it.

Differences:

In addition to revising users' resumes, our application provides a more comprehensive evaluation of the resume including separate evaluations of each section of background, education, skills and work experience.

3. [Simplify](https://www.simplify.com/)

Simplify automates the job application process by auto-filling job applications and providing job matching recommendations for users.

Differences:

Simplify focuses on automating job applications, whereas our software focuses more on resume evaluation, interview preparation, and tailored suggestions for both job seekers and hiring managers.

4. <https://www.resumehelp.com/>

- a. Pons: Can view the entire resume and rate the resume's scores. Can beautify a single line in the resume.
- b. Can check the job description and give suggestions on what to add.
- c. Cons: Can't evaluate how well each part of the resume relates to the job description.
- d. Can't beautify the desired line in the context of the whole resume and job description.

3. Proposed High level Requirements @All

- Functional Requirements

(For each functional requirement, please give a feature title and a brief description using the following format: As (a role), I want to (action), so that (value).)

- Essential Features (the core features that you definitely need to finish):
(For each essential features, please give a rough estimation in terms of person hours or an range of person hours)
 - User login
 - Description: As a user, I want to login using my Google account so that I can track my historical analysis records.
 - Estimated person hour: 20.

- Resume analysis
 - Description: As a job seeker, I want to upload my resume and receive feedback on how well I match a specific job description, and how to increase my chances of getting the job.
 - Estimated person hour: 100.
- Interview question suggestion
 - Description:
 - Job seeker: As a job seeker, I want suggestions for potential interview questions so that I can prepare for the interview.
 - Interviewer: As an interviewer, I want suggestions for resume-related questions so that I can know the interviewee better.
 - Estimated person hour: 40.
- Answer users' questions
 - Description: As a user, I want to ask questions about the resume or the job description so that I can get insights on job match rate or what to improve or what interview questions to expect.
 - Estimated person hour: 40.
- Desirable Features (the nice features that you really want to have too):
 - Project recommendation
 - Description: As a job seeker, I want to get recommendations for relevant projects or skills I should pursue to better match the job description.
 - Estimated person hour: 80.
 - Optional Features (additional cool features that you want to have if there is time):
 - Mock interview
 - Description: As a user, I want to simulate an interview based on the resume and the job description so that I can practice my responses.
 - Estimated person hour: 40.
- Nonfunctional Requirements
 - Security requirements Zihan Zhou
 - Secure User Authentication
 - **Description: As a user, I want to log into the application securely, so that my personal information is protected from unauthorized access.**
 - Input Validation and Sanitization
 - **Description: As a developer, I want the application to validate and sanitize all user inputs, so that the system is protected from injection attacks and other input-based vulnerabilities.**
 - Secure Data Storage
 - **Description: As a user, I want my personal data to be stored securely, so that it is protected from unauthorized access or breaches.**

- Deploy the application on cloud services
 - Description: As a developer, I want to deploy the application on cloud services to make it accessible for all users.
 - Estimated person hour: 30.

4. Management Plan Linchen Xu

a. Objectives and Priorities

(Please describe your project objectives with highest priority first. Project Goals can include but not limited to complete all proposed (essential) features, deploy the software successfully, the software has no known bugs, maintain high quality, etc)

1. Complete all essential features (resumes analysis, answer users' questions, generate interview questions)
2. Successfully deploy the software.
3. Ensure the software runs without any known bugs and is secure.
4. Maintain a high standard of usability and performance.
 - a. Coding: code style, structure
 - b. Project management: control the whole procedure.
5. Develop desirable features and optional features to make the application more attractive.

b. Risk Management (need to be updated constantly)

(Please write a summary paragraph about the main risks your group identified and how you plan to manage these risks. Then **use the separate google sheet** for detailed risk management. The template is provided in the same folder with this file. **Please provide the link to the sheet.**)

Risk Management Sheet Link: [x CS673_SPPP_RiskManagement of Team5.xlsx](#)

c. Timeline (**this section should be filled in iteration 0 and updated at the end of each later iteration**)

Iteration	Functional Requirements(Essential/Disable/Option)	Tasks (Cross requirements tasks)	Estimated/real person hours
1	Essential: Resume Analysis	Build resume upload feature Build job description matcher Integrate analysis and feedback	Estimated: 100 Real: 200
2	Essential: Interview	Suggest potential	Estimated:

	question suggestion	interview questions based on resumes and job descriptions	40
2	Essential: Answer users' questions	Answer questions based on user's input	Estimated: 40
2	Essential: User Login	Login using Google account	Estimated: 20
2	Essential: Deploy the application on cloud services	Deploy the application on cloud services	Estimated: 30
3	Optional: Mock interview	Provide simulated interview	Estimated: 40
3	Desirable: Project Recommendation	Build project recommendation	Estimated: 80

5. Configuration Management Plan adamma@bu.edu

a. Tools

(In this project, we will use Git and Github as the version control tools. Please also specify any other tools to be used, e.g. IDE tools, CI/CD tools, container tools, SAST or DAST tools, and any other DevOps tools)

IDE: No limitation.

CI/CD(Done):

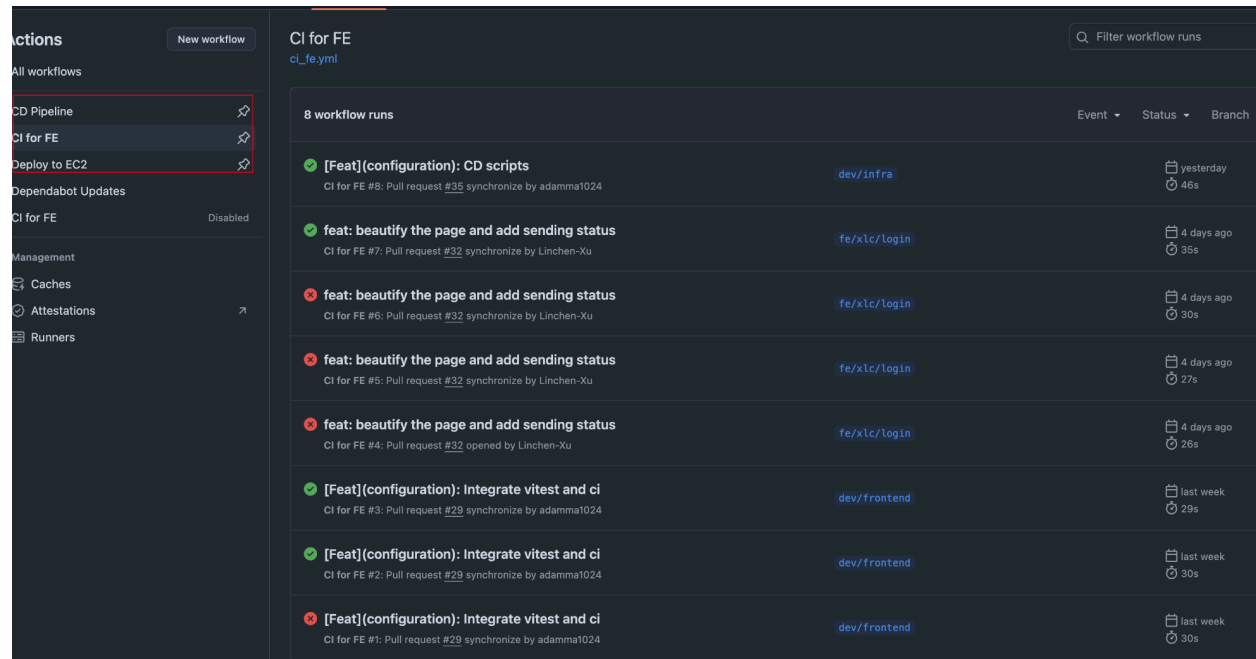
CI:

[CI for FE](#)

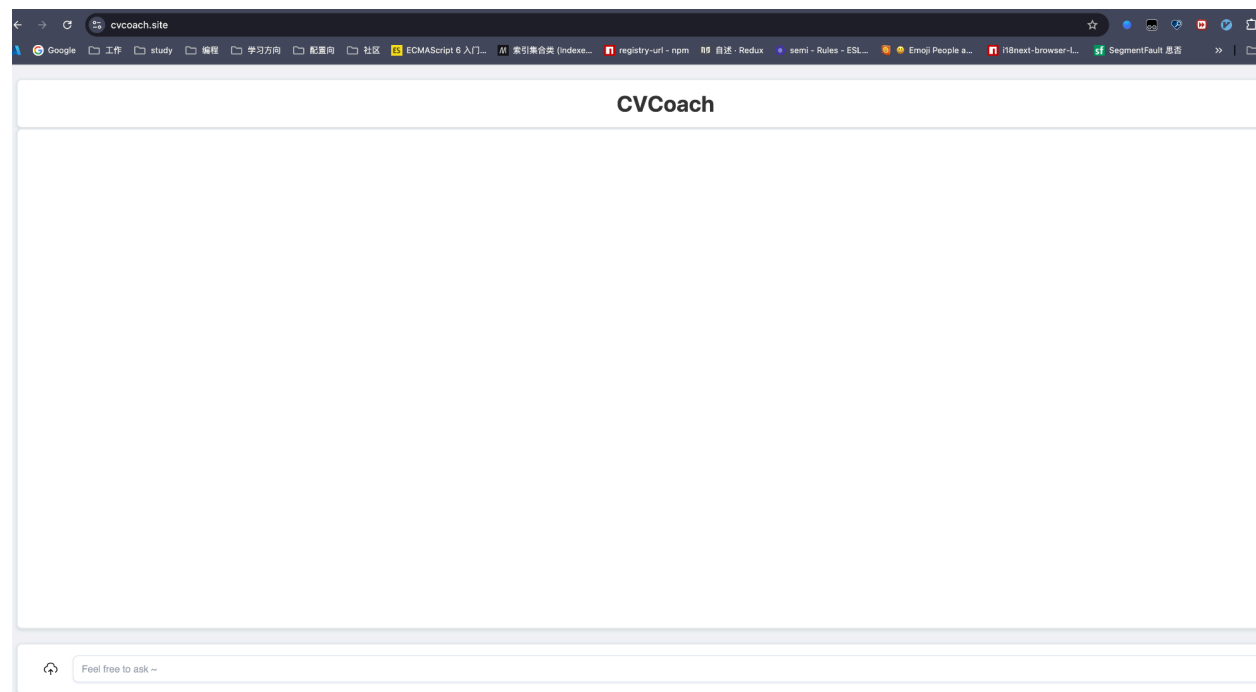
CD:

[Auto publish docker image](#)

[Deploy to EC2](#)



Container(Done): [Docker](#) + GCP EC2(instance) + Cloudflare (Security) + Domain([cvcoach.site](#))



Release Version and Changelog convention: [semantic-release](#)
<https://www.conventionalcommits.org/en/v1.0.0/> Lin Ma (Done)

- b. Code Commit Guideline and Git Branching Strategy
(Please briefly describe criteria for the code commitment and the branching strategy)

used, e.g. what are the branches to be used, how the pull request will be used etc. Here is an article to give you some basic knowledge about different git branching strategies: <https://www.flagship.io/git-branching-strategies/>

- i. [Github Ruleset: Main](#) (Done)
- ii. [Github Ruleset: Release](#) (Done)
- iii. Guided by [semantic-release](#) to maintain an CHANGELOG.md file (Done)
- iv. Branches:
 1. Dev branches: dev/frontend, dev/backend, dev/infra
 2. Release branches: releases/*
 3. Main: main
 4. fe/your_name/xxx, be/your_name/xxx for the building.
- v. [PR template](#) (Done)
- vi. [Issues templates](#) (Done)

c. Deployment Plan

(If you plan to deploy your application (e.g. your web application), briefly describe how you plan to deploy your application).

Github gist and github CI/CD.

1. Create a CD pipeline and manually trigger the Release deployment on the corresponding iteration branch. (Done)
2. **Projected day of release:** Generally, we'll check out a release branch on the last sunday before the end of iteration (Done)
3. [Docker](#) + GCP EC2(instance) + Cloudflare (Security) + Domain([cvcoach.site](#)) (Done)
4. We'll auto publish a docker image as we checkout the "releases/*" branch by the CD pipeline: [Auto publish docker image](#).
5. We'll auto deploy the latest docker image to the GCP EC2 instance as we publish the releases on Github by pipeline: [Deploy to EC2](#)

6. Quality Assurance Plan @Zhen Cao

a. Metrics

(Describe the metrics to be used in the project to measure the quality of your software. Each metric should be measurable and quantifiable. Examples of metrics include product complexity (LOC, # of files, # of classes, # methods, cyclomatic complexity, etc.) , defect rate (# of defect per KLOC), # of test cases, test case pass rate, cost (# of person hours used), # of user stories completed, etc. **The result of these metrics should be reported in the progress report/ iteration summary sheet.**)

Metric Name	Description
Lines of Code	Track the size and complexity of the product

# of Files # of Classes # of Methods	
Cyclomatic Complexity	Use the RADON tool which is related to Test Automation for python projects to make our code maintainable and readable
Defect Rate	Measure the number of defects per thousand lines of code to track the stability of our code
# of Test Cases Test Case Pass Rate	Ensure we uncover potential issues in our software development process
# of User Stories Completed	Make sure our application development keeps up with the timeline
# of Person Hours Used	Track the hours spent by each team member on specific tasks

b. Coding Standard

(Describe any coding standard to be used)

Python: [PEP 8](#)

React: [Preitter](#) adamma@bu.edu

c. Code Review Process

(Everyone should review all documents to be submitted. Here you will mainly describe how the code review will be done. Who will review the code, e.g. design or implementation leader will review all code or team members review each other's code. Do you use pull requests for the code review? Is there a checklist to help review? What feedback should the reviewer provide?)

- Everyone should review all documents to be submitted.
- All code will be reviewed by at least one other team member before being submitted
- Pull requests are used to initiate the process of integrating new code changes into the main branch
- Review should follow the checklist and write the feedback

Review Checklist:

- ☐ Obvious bugs
- ☐ Possible security issues

- ☐ Adequate unit tests
- ☐ Edge cases and exception handling are considered
- ☐ Possible performance improvements
- ☐ Code follows the coding standard
- ☐ Code is well commented

d. Testing

(Both manual testing and automated testing should be considered. Both unit testing and integration testing should be considered. Briefly describe the testing tools/framework to be used, the personnel involved (e.g. the QA leader will focus on the integration testing and each developer will unit test their own code), when and what types of testing will be performed, the testing objectives, etc)

Test	Person Involved	Tools	When to solve
Unit Test	Each team member are responsible for writing unit tests for their own code	Pytest	Before code reviews
Integration Test	QA Leader	Testify	After key integrations between components are implemented

e. Defect Management

(Describe the tool to be used to manage the defect (e.g github issues). The types of defects to look at. The actions or personnel for defect management.)

Notion Kanban board will be used for managing the defect.

Defect Types:

- Functional defects
- UI defects
- Performance defects
- Security

During development, any defects found during testing should be logged. Then defects will be assigned for resolution based on priority. At last each defect will be resolved before closure.

7. References @All

(For more details, please refer to the encounter example in the book or the software version of the documents posted on blackboard.)

1. Extensishr. 2023. "The Real Costs of Recruiting." Extensishr.

<https://extensishr.com/resource/blogs/the-real-costs-of-recruiting/>.

8. Glossary @All

(Any acronym used in the document should be explained here)

- **AI (Artificial Intelligence):**
A branch of computer science focused on creating systems capable of performing tasks that typically require human intelligence, such as learning, reasoning, and problem-solving.
- **ATS (Applicant Tracking System):**
Software used by employers to manage job applications and screen resumes. It filters resumes based on keyword matches with job descriptions.
- **Cloud Services:**
Cloud services are application and infrastructure resources that exist on the Internet. Third-party providers contract with subscribers for these services, allowing customers to leverage powerful computing resources without having to purchase or maintain hardware and software.
- **AWS (Amazon Web Services):**
A cloud computing platform provided by Amazon, offering a wide range of services such as storage, computing power, and machine learning models, often used for hosting and deploying applications.
- **UI (User Interface):**
The space where interactions between humans and machines occur. It includes the design and layout that make an application accessible and usable.