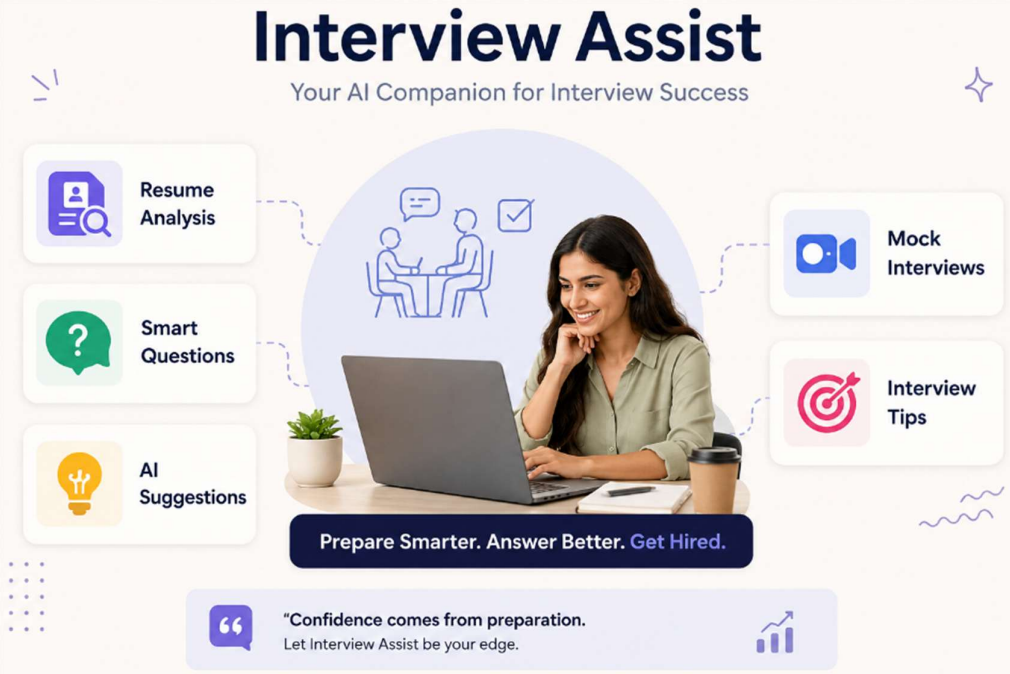


Contents

Context C1 1
Container C2 2
Component C3 3
Code C4 3
Upload Resume Flow 4
Job Extraction 4
Display Results 5
General , Settings 6
Technical Constraints 6
Future Phase 7

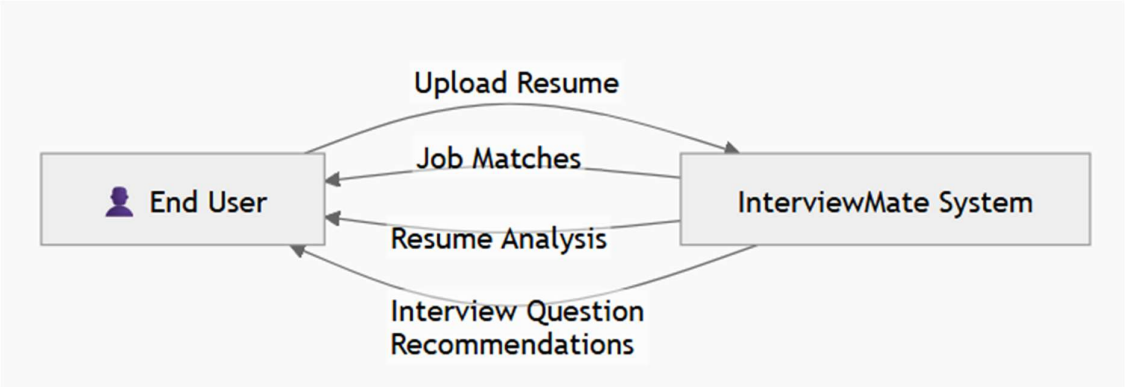
Context C1

Interview Assist is an AI-powered interview preparation and job matching platform for C# and ASP.NET developers.

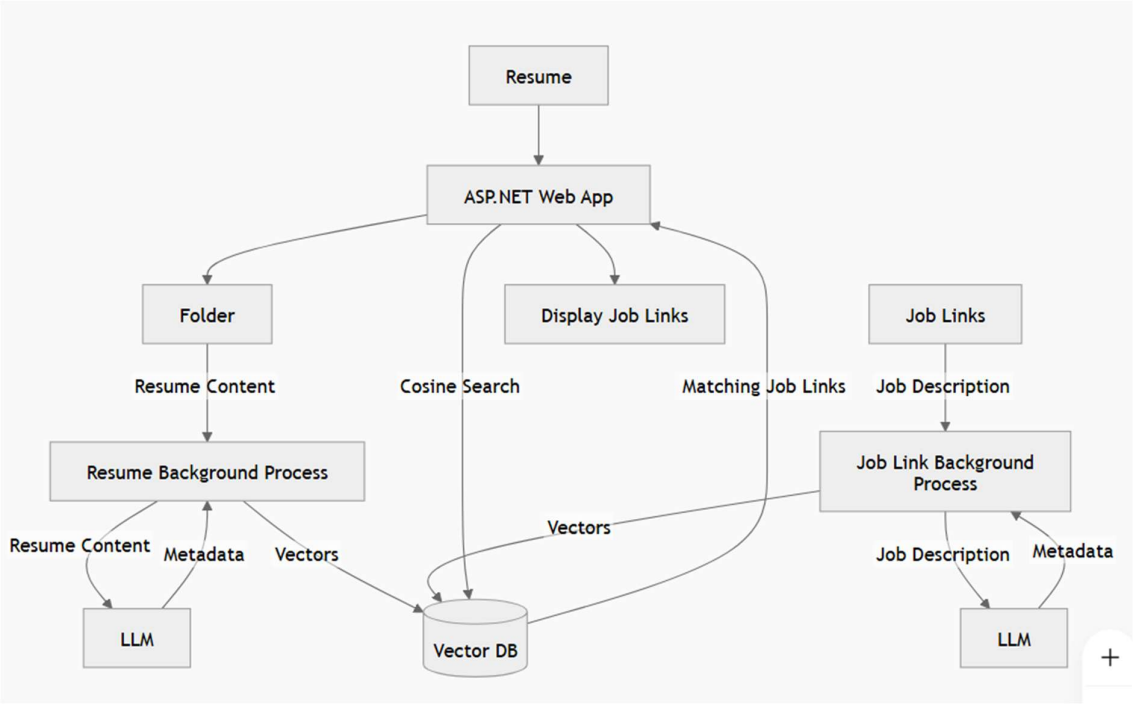
The application allows users to upload their resumes and performs the following functions:

1. **Job Matching**

- Compares the uploaded resume with available job postings.
- Uses vector embeddings and similarity search to identify the most relevant job opportunities.
- Displays matching job links and job details to the candidate.



Container C2



Resume upload Flow :-

- The End User uploads a resume through the ASP.NET Web App.
- The ASP.NET Web App stores the uploaded resume in a folder.

- The Background Process reads the resume content from the folder.
- The resume content is sent to the LLM.
- The LLM extracts metadata from the resume.
- The Background Process converts the extracted metadata into vectors/embeddings.
- The generated vectors are stored in the Vector Database (Vector DB).

Job link processing Flow :-

- The Job Link Background Process reads the job links.
- The Job Link Background Process extracts the current job descriptions from the job links.
- The job descriptions are sent to the LLM.
- The LLM extracts metadata from the job descriptions.
- The Job Link Background Process converts the extracted metadata into vectors/embeddings.
- The generated vectors are stored in the Vector Database (Vector DB).

Job Display Flow :-

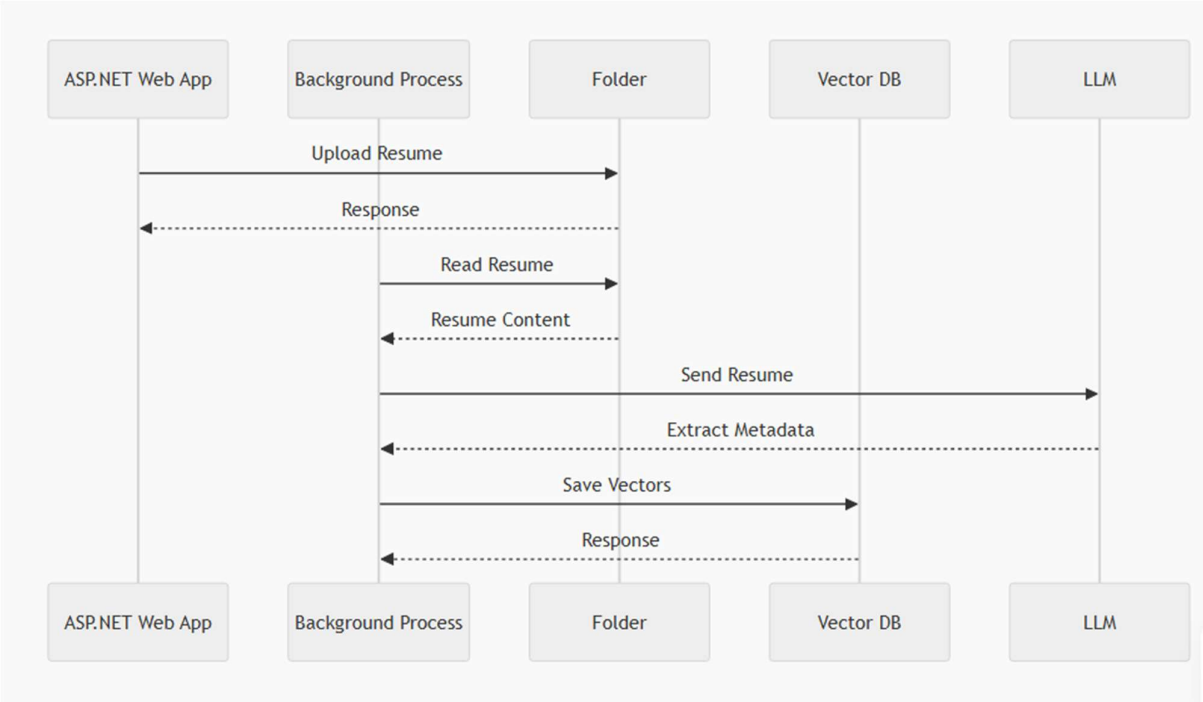
- ASP.NET app queries the job listing.
- Best Cosine JOB searches are displayed to the end user.

Component C3

Components	**Description**
Embedding	nomic embed text
Vector Db	Qdrant
Service Layer.	This will do the interactions between the UI and the Other layers (Repository Layer , LLM and Vector Layer).
Repository Layer.	This will talk with QDrant for Insert update and Delete.
LLM Layer.	This layer will talk with the LLM and extract the Meta data.
Vector Layer.	This Layer will help to convert and compare vector.
Console Application for Resume	This is the background process which will read the resume links , call repository and make it vectorized.
Console Application for Job links	This is the background process which will read the resume links , call repository and make it vectorized.

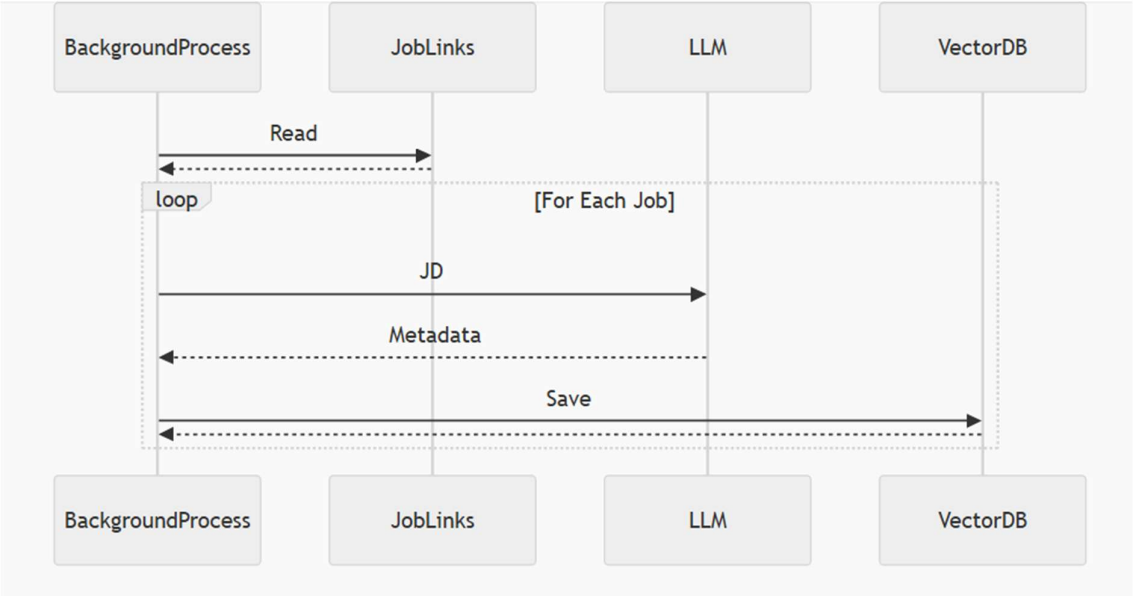
Code C4

Upload Resume Flow



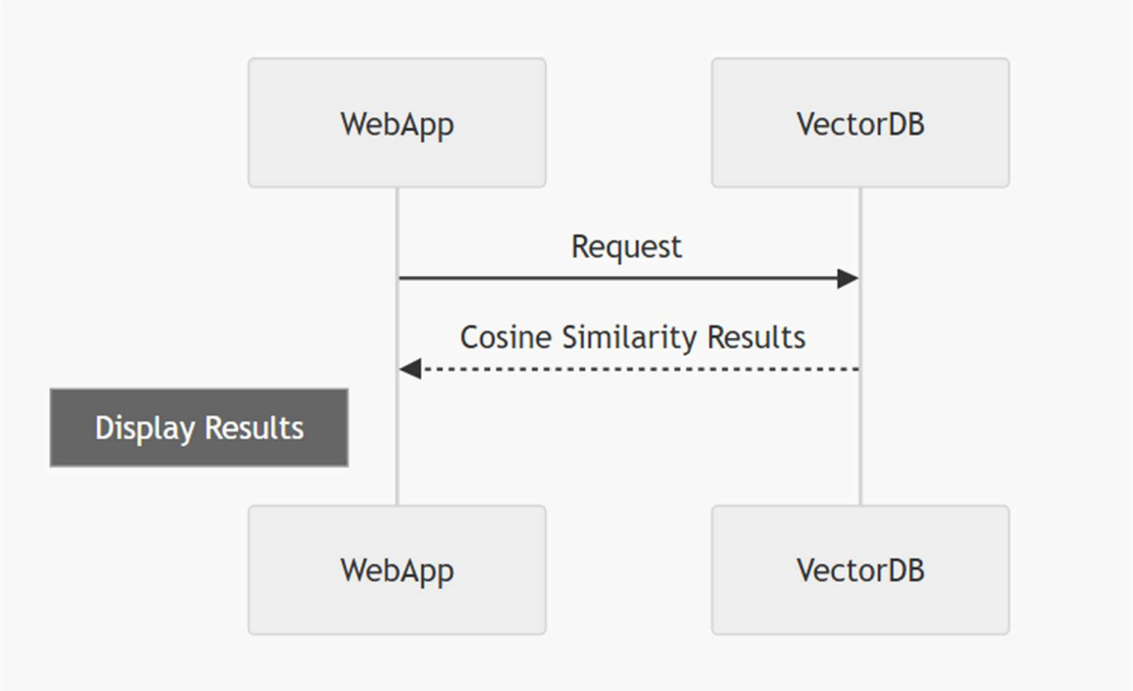
- ASP.NET uploads the resume to a folder.
- The Background Process reads the resume content from the folder.
- The resume content is sent to the LLM.
- The LLM extracts metadata from the resume.
- The Background Process creates embeddings/vectors from the extracted data.
- The vectors are stored in the Vector Database (VectorDB).

Job Extraction



- The Background Process reads the job links.
- For each job link, the job description (JD) is sent to the LLM.
- The LLM extracts metadata from the job description.
- The extracted metadata is converted into vectors/embeddings.
- The vectors are stored in the Vector Database (VectorDB).

Display Results



- The Web Application sends a search request to the Vector Database.
- The Vector Database performs a cosine similarity search between the resume vectors and job vectors.
- The job links with the highest cosine similarity scores are retrieved.
- The matching job links are returned to the Web Application.
- The Web Application displays the most relevant job links to the user.

General , Settings

Appsettings.json :-

- Folder path where the resume will be stored.
- Qdrant server name and Qdrant port number.

General

- **ASP.NET Web Application** for resume upload and result display.
- **Background Processing Service** for extracting metadata from resumes and job descriptions.
- **Vector Database** for storing and comparing embeddings.
- **Ollama** for local LLM processing.
- Embedding proposed:- Nomic Embed Text.
- Metadata extraction from both resumes and job descriptions into a common structured format:
    - Years of Experience
    - Technical Skills
    - Education
    - Company Name

Job Link

Technical Constraints

- Must use **free/open-source models and embeddings**.
- No paid services such as OpenAI/ Azure AI APIs should be used.
- LLM’s and Embeddings should be replaceable in future with minimal code changes.
- Security should be a primary consideration.
- Solution should run locally wherever possible.

Future Phase

1. **Resume Gap Analysis**
    - Analyzes the candidate's experience, skills, and profile.
    - Compares them against predefined expectations for different experience levels.
    - Suggests missing skills, technologies, or certifications (e.g., Azure, Microservices, etc.) that should be learned to improve employability.
2. **Interview Question Recommendation**
    - Maps the resume against a curated database of real interview questions collected from industry professionals.
    - Generates a personalized list of the most relevant interview questions based on the candidate's skills and experience.