

CV Analysis API Documentation

This document provides comprehensive documentation for the CV Analysis API endpoints.

Base Information

- **Base URL:** `http://localhost:8000` (or your configured host)
- **API Version:** 1.0
- **Content-Type:** `application/json` (except for file uploads)

Authentication

Currently, no authentication is required for API endpoints.

Endpoints

CV Management

1. Upload CV Files

POST `/cv/upload`

Upload CV files or provide a Google Drive link for processing.

Content-Type: `multipart/form-data`

Parameters:

- `googleDriveUrl` (string, optional): Google Drive URL for CV files
- `files` (array of files, optional): CV files to upload (PDF, DOCX, ODT)

Request Example:

```
curl -X POST "http://localhost:8000/cv/upload" \  
  -F "files=@resume1.pdf" \  
  -F "files=@resume2.docx" \  
  -F "googleDriveUrl=https://drive.google.com/file/d/111JgIe-990mf4gQhE98G3dhkGiPDx4mf/view?usp=sharing"
```

Response:

Will return a list of the corresponding

```
{  
  "application_ids": [1, 2, 3],  
  "message": "Successfully added 3 applications."  
}
```

Error Responses:

- 400: No CV files or Google Drive link provided
 - 400: No valid CV files found
 - 500: Internal server error
-

2. Update CV Files**PUT** /cv/update/{id}

Update CV file for an existing application.

Path Parameters:

- id (integer, required): Application ID to update

Content-Type: multipart/form-data

Parameters:

- googleDriveUrl (string, optional): Google Drive URL for CV file (must be a shared link with public view, can be only a link to a file, as this API point does not support folder).
- file (array of files, optional): CV file to upload (PDF, DOCX, ODT)
- There should not be both

Request Example:

```
curl -X PUT "http://localhost:8000/cv/update/9" \  
-F "file=@updated_resume.pdf"
```

Response:

```
{  
  "application_ids": 9,  
  "message": "Successfully updated application with id 9."  
}
```

Error Responses:

- 400: No CV files or Google Drive link provided
 - 400: Google Drive folders are not supported for updates
 - 400: Invalid Google Drive link provided
 - 400: No valid CV files found
 - 404: Application not found. Cannot update application.
 - 500: Internal server error
-

3. Search CVs

POST /cv/search

Search for CVs based on various criteria using both structured and semantic search.

Search parameters is a SearchCVQuery object (JSON) with the following fields

For name, email, phone, linkedIn, gitRepo, experiencedSkills: Will match any application containing the keyword.

For skills, jobTitles, location, requirementDescription: Will search for applications that match the best to the keywords in the database.

- **name**: String (Applicant name in the CV, optional)
- **email**: String (Email address, optional)
- **phone**: String (Phone number, optional)
- **linkedIn**: String (LinkedIn URL, optional)
- **gitRepo**: String (Git repository URL, optional)
- **experiencedSkills**: A JSON object with the key is the skill or role (with real experience) to search (String) with value is a float number indicating minimum years for applications (Float) (optional)
- **keywords**: Array of strings (Keywords to search, optional)
- **skills**: Array of strings (Skills to search, optional)
- **jobTitles**: Array of strings (Job titles to search, optional)
- **location**: String (Address or current work location of the , optional)
- **requirementDescription**: String (Job requirements description, optional)

Content-Type: application/json

Request Body (SearchCVQuery):

```
{
  "name": "Josh",
  "email": "josh@jf.us",
  "phone": "(860) 716-5996",
  "linkedIn": "/josh",
  "gitRepo": "/josh",
  "experiencedSkills": {
    "Front-end": 2.5
  },
  "keywords": [
    "University of Maryland"
  ],
  "skills": [
    "Python", "ASP.NET", "AWS"
  ],
  "jobTitles": ["Senior Software Engineer"],
  "location": "United States",
  "requirementDescription": "Having experience in cloud, ASP.NET, Amazon Web
```

```
Services and databases such as MongoDB and MySQL"
}
```

Note you can omit some of the parameters above to exclude these in searching, for example:

```
{
  "gitRepo": "/josh",
  "experiencedSkills": {
    "Front-end": 2.5
  },
  "skills": [
    "Python", "ASP.NET", "AWS"
  ],
  "requirementDescription": "Having experience in cloud, ASP.NET, Amazon Web
Services and databases such as MongoDB and MySQL"
}
```

Response Example:

```
[
  {
    "application": {
      "id": 5,
      "name": "Josh Finn",
      "gitRepo": "/josh",
      "lastUpdated": "2025-06-17T18:23:33",
      "vectorDbUuid": "c52753c0-297b-4bdf-b2a3-7a6fca7fee32",
      "email": "josh@jf.us",
      "phone": "(860) 716-5996",
      "linkedIn": "/josh",
      "yearsOfExperience": null
    },
    "education": [
      {
        "institution": "University of Maryland",
        "id": 8,
        "degree": "Master of Science in Data Analytics",
        "application_id": 5,
        "gpa": null,
        "year": "Expected 2027"
      },
      {
        "institution": "Central Connecticut State University",
        "id": 9,
        "degree": "Classes towards a Master of Science in Geography (GIS Track)",
        "application_id": 5,
        "gpa": null,
        "year": null
      }
    ]
  }
]
```

```

{
  "institution": "University of Connecticut",
  "id": 10,
  "degree": "Bachelor of Arts in Economics (Certificate in Quantitative
Economics & Minor in Mathematics)",
  "application_id": 5,
  "gpa": 3.4,
  "year": 2024
}
],
"experiencedSkills": [
  {
    "yearsOfExperience": 5,
    "skill": "API Development",
    "application_id": 6,
    "id": 61
  },
  {
    "yearsOfExperience": 1,
    "skill": "SDK Development",
    "application_id": 6,
    "id": 63
  },
  {
    "yearsOfExperience": 4,
    "skill": "Front-End Development",
    "application_id": 6,
    "id": 64
  },
  {
    "yearsOfExperience": 2,
    "skill": "AWS",
    "application_id": 5,
    "id": 39
  },
  {
    "yearsOfExperience": null,
    "skill": "AWS",
    "application_id": 5,
    "id": 48
  }
],
"skillsAndExperience": "Skills: \n\n\nWork Experiences: \n\n  Senior
Software Engineer\n  May 2023 - Present\n  People Data Labs\n  Remote, USA\n
Focusing on building scalable APIs and cloud-based solutions that enhance customer
interactions with data. Collaborating\n  with design to bring joy to internal
users of our Admin Website.\n  Technologies: Python, Rust, TypeScript, React,
AWS, Pulumi, Flask, PostgreSQL\n  • API Development: Led development of scalable
RESTful APIs using Python and Flask, reducing data access issues\n  by 25%.\n
• Project Leadership: Coordinated monthly data update launches between API and
Data teams, ensuring seamless and\n  timely releases.\n  • SDK Development:
Authored the Rust SDK and contributed to multi-language SDKs (Python, Go,
JavaScript),\n  improving developer experience.\n  • Front-End Development:
Built internal admin interfaces with React and TypeScript to empower sales and

```

```
support\n    teams in serving customers efficiently.\n    \n\n    Software Engineer\n    May 2022 - Mar 2023\n    IndigoAg\n    Remote, USA\n    Contributed to Carbon by Indigo, enabling farmers to generate revenue via sustainable practices and verified carbon credits.\n    Collaborated with external partners integrating data into carbon sequestration models.\n    Technologies: Flask, Python, PostgreSQL, Docker, AWS\n    • API Development: Created a next-generation Carbon Sequestration API with Flask, delivering a scalable, high-\n    performance platform that met stakeholder goals.\n    "
  },
]
```

Error Responses:

- **404**: No applications found matching the search criteria
- **500**: Internal server error

4. Get CV by ID

GET /cv/{id}

Retrieve details of a specific CV by ID.

Path Parameters:

- **id** (string, required): Application ID

Request Example:

```
curl -X GET "http://localhost:8000/cv/8"
```

Response (Example):

```
{
  "application": {
    "id": 8,
    "name": "Hamza B.",
    "gitRepo": "Hamza B.",
    "lastUpdated": "2025-06-18T17:32:17",
    "vectorDbUuid": "e825928e-ceba-4b62-bdb5-133bf144cf64",
    "email": "Hamzamrbek@gmail.com",
    "phone": "+20*****2806",
    "linkedIn": "Hamza B",
    "yearsOfExperience": null
  },
  "education": [
    {
      "institution": "Open University",
      "id": 18,
```

```

    "degree": "Bachelor of Business Management",
    "application_id": 8,
    "gpa": "3.5/ 4.0",
    "year": "2026"
  }
],
"experiencedSkills": [],
"skillsAndExperience": {
  "id": null,
  "metadata": {},
  "page_content": "Address: Egypt\nSkills: \nC#, .NET, Blazor, SignalR, Entity Framework, SQL, Tailwind CSS, HTML, CSS, Git, RESTful APIs\n\nWork Experiences: \n\n\nProjects: \n\n    South Mart - Full E-Commerce Management System\n    - User Authentication and Role Management: Implemented secure user registration, login, and role-based access control for customers and administrators.\n    - Product Catalog Management: Built an intuitive interface for adding, updating, and categorizing products with advanced search and filtering capabilities\n    - Order and Inventory Management: Developed functionality for tracking orders, managing stock levels, and sending low-stock notifications\n    - Admin Dashboard: Designed a comprehensive dashboard for monitoring sales, revenue, and customer data analytics\n\n    \n\n    AOU Community Platform - Real-Time Chat, Social Networking, Marketplace, and Event Management\n    - Real-Time Chat Feature: Integrated a real-time chat feature using SignalR, enabling instant communication between users and fostering collaboration\n    - Post and Announcement System: Developed a system for users to create, share, and interact with posts, announcements, and community updates\n    - Marketplace Integration: Built a marketplace feature allowing users to buy and sell items with secure transactions and easy listing management\n\n    \n\n    TailsBlazor - Component Library for Blazor Applications\n    - Custom Blazor Components: Developed a collection of reusable, customizable components tailored for Blazor applications, enhancing UI consistency and user experience\n    - Tailwind CSS Integration: Integrated Tailwind CSS to create responsive, modern, and visually appealing UI components that adapt seamlessly to different screen sizes\n    ",
  "type": "Document"
}
}

```

Error Responses:

- 404: Application not found
- 500: Internal server error

5. Delete CV

DELETE /cv/{id}

Delete a specific CV by ID.

Path Parameters:

- id (string, required): Application ID

Request Example:

```
curl -X DELETE "http://localhost:8000/cv/1"
```

Response:

```
{
  "message": "Application deleted successfully.",
  "application_id": 1
}
```

Responses:

- 200: Delete successfully
 - 404: Application not found
 - 500: Internal server error
-

6. List CVs with Pagination**GET /cv/**

Retrieve a paginated list of all CVs.

Query Parameters:

- **page** (integer, optional, default: 1): Page number
- **size** (integer, optional, default: 10): Number of items per page
- **orderBy** (string, optional, default: null): Order to sort the result. orderBy can be 'name', 'nameDesc', 'id', 'lastUpdated', default the option null is sorting by 'lastUpdated' descending. If using 'lastUpdated', the order will be sorting by 'lastUpdated' but ascending.

Request Example:

```
curl -X GET "http://localhost:8000/cv/?page=1&size=2&orderBy=name"
```

Response:

```
[
  {
    "application": {
      "id": 8,
      "name": "Shubham",
      "gitRepo": null,
      "lastUpdated": "2025-06-18T17:32:18",
    }
  }
]
```



```

"vectorDbUuid": "33ba69e6-cb96-4de0-b251-890e92b7ce59",
"email": null,
"phone": null,
"linkedIn": null,
"yearsOfExperience": null
},
"education": [
  {
    "application_id": 10,
    "id": 20,
    "institution": "University of Maryland",
    "degree": "Master's in Computer Engineering",
    "gpa": null,
    "year": "May 2019"
  },
  {
    "application_id": 10,
    "id": 21,
    "institution": "University of Pune",
    "degree": "Bachelor's in Electronics and Telecomms Engineering",
    "gpa": null,
    "year": "May 2017"
  }
],
"experiencedSkills": [],
"skillsAndExperience": "Address: None\nSkills: \nJava 8, Java 10, JavaScript (ES6), Python, C++, ReactJS, HTML5, CSS3, Sass, Bootstrap4, Figma, Github, Bitbucket, Jira, Git Workflows, Bitbucket Pipelines, Agile, Scrum, Kanban, Trello, Slack, Spring Boot, RESTful Web Services, Microservices, Hibernate, JDBC, JPA, PL/SQL, JUnit4, Postman, Designing APIs, Consuming APIs, App integrations, Design Patterns, SOLID principles\n\nWork Experiences: \n\n    Senior Software Engineer at Infosys Technologies Aug 2019 -Present\n    Indianapolis, IN\n    - Successfully developed methodologies to streamline the process of data conversion leveraging latest tools\n    - Created optimized PL/SQL scripts to automate data gathering and transformation on various database technologies\n    - Refactored existing processes and built component to reduce manual efforts and performed unit tests\n    - Conducted meetings and workshops with client to document requirements and provide roadmap for the project\n    \n\n    Java Developer at School of Education Jan -May 2019\n    College Park, MD\n    - Successfully architected and implemented features in Spring Boot to enhance the capabilities of web application\n    - Designed and developed various UI pages in ReactJS to support application and contributed in creating process workflows\n    - Achieved a maintainable and optimised code by refactoring and following best coding and design practices\n    - Implemented unit tests in JUnit4 and Mockito providing full code coverage\n    \n\n    Software Engineer Intern at Institute of Physical Sciences & Tech Jan2018-Feb2019\n    College Park, MD\n    - Designed and developed an application to automate the production of binary executables using Python and Pandas dataframes\n    - Implemented UI components using ReactJS with custom CSS and Bootstrap4 for layout and positioning\n    - Developed applications in Test Driven Development and performed extensive code reviews and thorough testing\n    - Interacted with stakeholders to gather and document requirements and provided weekly updates on progress and consultation\n    \n\nProjects: \n\nPodcastinator\n    Search for your favorite podcast from around 2 million podcasts and 91 million episodes all in one place.\n    "

```

```

},
{
  "application": {
    "id": 10,
    "name": "Hamza Bek",
    "gitRepo": "Hamza Bek",
    "lastUpdated": "2025-06-18T17:32:17",
    "vectorDbUuid": "e825928e-ceba-4b62-bdb5-133bf144cf64",
    "email": "Hamzamrbek@gmail.com",
    "phone": "+201123452806",
    "linkedIn": "Hamza Bek",
    "yearsOfExperience": null
  },
  "education": [
    {
      "application_id": 8,
      "id": 18,
      "institution": "Open University",
      "degree": "Bachelor of Business Management",
      "gpa": "3.5/ 4.0",
      "year": "2026"
    }
  ],
  "experiencedSkills": [],
  "skillsAndExperience": "Address: Egypt, Cairo\nSkills: \nC#, .NET, Blazor, SignalR, Entity Framework, SQL, Tailwind CSS, HTML, CSS, Git, RESTful APIs\n\nWork Experiences: \n\n\nProjects: \n\n    South Mart - Full E-Commerce Management System\n    - User Authentication and Role Management: Implemented secure user registration, login, and role-based access control for customers and administrators.\n    - Product Catalog Management: Built an intuitive interface for adding, updating, and categorizing products with advanced search and filtering capabilities\n    - Order and Inventory Management: Developed functionality for tracking orders, managing stock levels, and sending low-stock notifications\n    - Admin Dashboard: Designed a comprehensive dashboard for monitoring sales, revenue, and customer data analytics\n    \n\n    AOU Community Platform - Real-Time Chat, Social Networking, Marketplace, and Event Management\n    - Real-Time Chat Feature: Integrated a real-time chat feature using SignalR, enabling instant communication between users and fostering collaboration\n    - Post and Announcement System: Developed a system for users to create, share, and interact with posts, announcements, and community updates\n    - Marketplace Integration: Built a marketplace feature allowing users to buy and sell items with secure transactions and easy listing management\n    \n\n    TailsBlazor - Component Library for Blazor Applications\n    - Custom Blazor Components: Developed a collection of reusable, customizable components tailored for Blazor applications, enhancing UI consistency and user experience\n    - Tailwind CSS Integration: Integrated Tailwind CSS to create responsive, modern, and visually appealing UI components that adapt seamlessly to different screen sizes\n    "
  }
]

```

Error Responses:

- 404: No applications found

- **500**: Internal server error
-

Data Models

Application

- **id**: Integer (Primary key)
- **vectorDbUuid**: String (UUID linking to vector database)
- **name**: String (Applicant's name)
- **email**: String (Email address, optional)
- **phone**: String (Phone number, optional)
- **linkedIn**: String (LinkedIn profile URL, optional)
- **gitRepo**: String (Git repository URL, optional)
- **yearsOfExperience**: Float (Total years of experience, optional)
- **lastUpdated**: DateTime (Last updated timestamp)

Education

- **id**: Integer (Primary key)
- **application_id**: Integer (Foreign key to Application)
- **degree**: String (Degree obtained)
- **institution**: String (Educational institution)
- **year**: String (Graduation year or period, optional)
- **gpa**: String (Grade Point Average, optional)

ExperiencedSkill

- **id**: Integer (Primary key)
 - **application_id**: Integer (Foreign key to Application)
 - **skill**: String (Skill name)
 - **yearsOfExperience**: Float (Years of experience in skill, optional)
-

Supported File Types

The API supports the following CV file formats:

- **PDF** (.pdf)
- **Microsoft Word** (.docx) (Bonus)
- **OpenDocument Text** (.odt) (Bonus)

Google Drive Integration

The API supports Google Drive integration with the following URL formats (as the Google Drive download functionality is from gdown):

- Individual files: https://drive.google.com/file/d/FILE_ID/view
- Folders: https://drive.google.com/drive/folders/FOLDER_ID

Note: Google Drive folders are not supported for update operations.

Error Handling

All endpoints return standard HTTP status codes:

- **200:** Success
- **400:** Bad Request (invalid parameters or data)
- **422:** Unprocessable Entity (Invalid format, format of the error below)
- **404:** Not Found (resource doesn't exist)
- **500:** Internal Server Error

Example of a 422 error which always has `"message": "Invalid data in request."` and an array of errors, each error contains `errorAt` (indicates where the error is in, for example body or header), `attribute` (attribute with errors), and `errorMessage` (error message for that attribute):

```
{
  "message": "Invalid data in request.",
  "errors": [
    {
      "errorAt": "body",
      "attribute": "name",
      "errorMessage": "Input should be a valid string"
    },
    {
      "errorAt": "body",
      "attribute": "Rust",
      "errorMessage": "Input should be a valid number, unable to parse string as a
number"
    }
  ]
}
```

Error responses (except 422 error which has the format like the above) include a `detail` field with a descriptive error message:

```
{
  "detail": "No CV files or Google Drive link provided."
}
```

Search Functionality

The search endpoint combines both:

1. **Structured Search:** SQL-based filtering on specific fields (name, email, phone, etc.)
2. **Semantic Search:** Vector-based similarity search on skills, experience, and project descriptions

This dual approach enables both precise filtering and intelligent matching based on job requirements and skills.