MATCHING EMPLOYEES TO PROJECTS

Canon - EMEA

By
Saad Wali Muhammad Joiya

Submitted to Central European University - Private University
Department of Economics

In partial fulfilment of the requirements for the degree of Master of Science in Business Analytics

Supervisor: Eduardo Arino de la Rubia

Vienna, Austria
2025

COPYRIGHT NOTICE

For bibliographic and reference purposes this thesis/dissertation should be referred to as:
Joiya, Saad. 2025. Matching Employees to Projects for Canon - EMEA. MSC Thesis, Business Analytics, Central European University, Vienna.

AUTHOR’S DECLARATION

I, the undersigned, **Saad Wali Muhammad Joiya**, candidate for MSc degree in Business Analytics declare herewith that the present thesis is exclusively my own work, based on my research and only such external information as properly credited in notes and bibliography. I declare that no unidentified and illegitimate use was made of the work of others, and no part of the thesis infringes on any person’s or institution’s copyright.

I also declare that no part of the thesis has been submitted in this form to any other institution of higher education for an academic degree.

Vienna, 08 June 2025

Saad Wali Muhammad Joiya

EXECUTIVE SUMMARY

This project focuses on designing a smart, data-driven employee–project matching system to support staffing decisions at scale. In large organizations like Canon EMEA, where roles are highly specialized and projects vary greatly in scope and skill requirements, optimally matching resources to projects is vital for maximizing efficiency and enhancing delivery. The goal is to assist project managers and HR teams in identifying the most suitable employees for a given project by evaluating multiple dimensions of fit, thereby reducing manual effort, bias, and inefficiency in the resource allocation process.

Traditional project staffing often relies on intuition, making it difficult to objectively assess candidate suitability — especially when evaluating large teams or highly specialized roles. To address this, our system uses structured and unstructured data to score employees based on their alignment with project requirements.

We simulated realistic employee and project datasets using thematically aligned roles, summaries, and domain-specific pools for skills, products, certifications, and languages. These were based on industry research and job descriptions inspired by Canon EMEA's organizational structure. This thematic structure helped ensure that the matching logic made practical sense — for example, you would not see a legal advisor matched to a cloud infrastructure rollout. Each employee and project record includes detailed features such as job descriptions, role summary, skills with proficiency levels, language fluency, certifications, availability, and soft skills like cultural awareness and leadership experience and more. We then define corresponding features from both tables for scoring purposes.

Matching is performed using a multi-criteria scoring framework. This includes similarity scoring between text-based features, fuzzy matching for fields with human-written inputs (e.g., skills, locations, languages), keyword overlap for list-based features multi-step statistical scoring algorithms for numeric fields and some additional bonus scores for employee values and characteristics. We also incorporated logic for filtering out employees who are not available within a project's timeline.

All these scores are weighted — meaning decision-makers can prioritize what matters most (skills, availability, certifications, etc.) and rolled into a final score that ranks the top best-fit employees per project. The result is a flexible, interpretable, and fair system that offers a scalable foundation for intelligent workforce planning.

For demonstration purposes we implement the complete process in a Streamlit dashboard for interactively displaying top employees for each project based on feature weightages and a detailed comparison and explanation for each proposed employee ranking.

ACKNOWLEDGEMENTS OR DEDICATIONS

I would like to pay my gratitude to Professor Gábor Békés and Professor Eduardo Arino de la Rubia from CEU for their guidance. Special thanks to Martin Brueggemann from Canon – EMEA for giving me the opportunity to work on this project and providing guidance throughout the course of its duration.

This journey would not have been possible without the unwavering support of my father, Saeed Joiya, and my mother, Naima Saeed, whose guidance and encouragement have been my foundation.

A heartfelt thank you to my wife, Mashal, for her constant support, patience, and belief in me throughout every step of this process.

TABLE OF CONTENTS

LIST OF FIGURES, TABLES, OR ILLUSTRATIONS

CHAPTER 1: OVERVIEW

Partner

The partner for our Capstone project is Canon – Europe Middle East and Africa**.** Founded in 1937, Canon provides state-of-the-art imaging solutions to its clients which are spread over a vast array of domains. Canon develops industry-leading technology supporting future demands of photographers, videographers, office workers, professional printers, medical imaging experts and more. It has around 150,000 employees across the globe with 15,000 working for Canon EMEA.

Problem Statement

Currently, HR of Canon EMEA uses a manual process to match suitable employees for incoming projects. This process is not optimal considering time, effort, scalability and human errors. Moreover, an organized framework for projects and employee data collection is also currently absent. There is a need for creating a proof of concept that streamlines end to end data collection from feature list to feature data types, utilizes this data to create a framework of scoring employees against projects and interactively presents these results to the user.

Project Motivation

For an organization of Canon's size and stature it becomes imperative to optimally match employees and projects to maximize efficiency and enhance delivery. This project would reduce a lot of workload from HR and other managers who can easily find the best employees for different project needs. Moreover, we can enhance the quality of employee allocation by more accurate and consistent recommendations based on matching features. This system can be improved over time with innovations as there has been less prior work done in this regard.

Project Aim

Our aim for this project is to create proof of concept that provides a comprehensive solution streamlining the complete process from data collection to optimal employee allocation. Ideally, this system would be dynamic enough to incorporate future changes and mature over time with additional data.

Approach

For this project, we did not have access to real world data. Hence, we defined the feature space for both the employees and projects table. This was done in collaboration with Canon. This mock data with relevant data format and values mimicked real world data closely. For calculating the matching scores for different employees, matching features from both feature spaces were outlined and different matching techniques based on data types of the features involved were used. This allowed for optimal generation of scores for different feature subgroups. Additionally various weightages were defined for each feature subgroup to generate final scoring. Moreover, some features were used in post processing for filtering optimal employees. To demonstrate our results, Streamlit app was leveraged that presented the top employees for each selected project while giving the user control over feature weights and filters.

Pre-Requisites for Project	**Source for Pre-requisite**
Employees Data	Mock data creation based on client recommendation
Projects Data	Mock data creation based on client recommendation
Data Fields for both tables	Project Sponsor from Canon
Data Types for all features	Project Sponsor from Canon
Relevant values for features in both tables	Online research
Algorithm for matching relevant fields	Algorithms research
Feedback on match scores	Arena Approach (collaboration with project sponsor)

*Table 1**:** Pre-Requisites for the Project*

Limitations

There was no historic data provided for the project due to accessibility issues. Mock data was created based on the feature sets recommended by the client. This required research on the respective features values in context to the job offerings and projects at Canon. Moreover, due to the absence of any true labels ML models could not be trained, so the reliance was mainly on statistical approaches. Moreover, the validity of the implemented algorithm’s output scores would be based on project sponsors’ judgement. Furthermore, human errors can be present during data collection for projects table as some of the data fields are hand filled by humans. We cater to this in our algorithm during matching and scoring. Together with this perception bias can also skew results in real life during data entry. HR needs to be made aware of these consequences and trained to be mindful while filling out the project forms.

Advantages of our approach

The approach implemented is highly interpretable. It is easy to grasp even for people with non-technical backgrounds. Weights were used as needed to control which features contribute more than others to the final score and ranking. This method does not have the usual limitations of a Machine Learning model, making it highly dynamic and extendable. Moreover, it provided the leverage to tweak the model based on client expectations. For example, more features can be added over time, fuzzy thresholds can be tweaked and scoring functions can be easily updated.

Project Delivery

For easy reproducibility and collaboration, the following were generated for this project:

Artifact	Purpose
Capstone Report	Contains project details, complete overview of data, matching and scoring mechanism and additional information
Jupyter Notebook	Proof of concept of the complete process from data creation to final scoring would be implemented in a Jupyter notebook containing all dependencies and explanations
StreamLit App	A Streamlit app would be deployed on a VM and be publicly accessible for demonstration purposes. The code for the complete dashboard including all processes (including those in proof of concept) is provided in a shared repository
GitHub Repository[1]	All artifacts are uploaded in a public GitHub repository, shared with the project sponsor

Table 2: Deliverables for the Project

[1] GitHub Repo:
https://github.com/joiya-saad/Capstone-Project

CHAPTER 2: MOCK DATA CREATION - DESIGN AND STRATEGY

To build a robust, realistic project-to-employee matching system, a structured approach was implemented for generating mock data that mimics real-world conditions at a company like Canon EMEA. This data serves as the foundation for testing our feature-matching algorithms and scoring logic.

Goal

The goal was to generate two high-quality datasets:

- Projects Table: Represents incoming client projects Canon may handle
- Employees Table: Represents internal talent pool with skillsets and preferences

Each row in both datasets is rich in features relevant to HR and resource allocation.

Realism and Consistency

To avoid random or inconsistent data, several real-world design constraints were imposed while mock data creation. This was labelled as Thematic Control. This is done to ensure internal consistency between fields of the same project/employee. For example, if the project requires a person from HR, required skills would have values such as Talent Management and required certifications would have values such as PMP. Whereas, if a technical resource is required the skills might include Data Analysis, cloud services etc. and required certifications would have values such as Microsoft Azure Certification. This approach brought internal consistency and realism into the implementation.

We categorized all roles and projects under six high-level business themes:

- Technical
- Sales
- Marketing
- HR
- Legal
- Consulting

This ensured internal consistency between:

- Products involved
- Required skills and Expertise
- Customer Preferences (Certifications)
- Integration Requirements (Expertise Areas)
- Project Summary
- Scope and Deliverables

Each theme had predefined pools of realistic values.

Note: *These feature names are benchmarked from projects table. We might have matching columns in the employees' table but with different names. For information on theme specific values, refer to the* ***appendix.***

Employee Role and Project Summaries

To mimic Canon EMEA's context themes and thematically controlled features discussed in point 2, Google Gemini API[2] was used to create relevant employee roles and project summary/scope & deliverables. Prompts were provided to Gemini to analyze job postings on

[2] Google Gemini 1.5 Flash API: https://cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/1-5-flash

LinkedIn and Canon's career portal [3]and curate unique roles with concise theme-based free text descriptions for employee roles, project summary and project scope & deliverables. In the event of failure to fetch this data from API, some thematically controlled employee roles and project summaries templates were curated based on manual research online specifically from Canon's website and other job portals *(check* ***appendix*** *for details)*. These would act as a fallback option in case of any errors during data creation for these features.

Generic Pools

Unlike theme specific pools that derive feature values based on themes for consistency and realism in data, some of the features are more generic *(more details in* ***appendix****)*. The following generic vocabularies were designed:

Category	**Description**
Work Location	Main European cities like Berlin, Vienna, London etc.
Work Flexibility	Working options like remote, onsite or hybrid
Languages Required	European languages like English, French, Italian, German etc.
Language Level	Selected from CEFR levels (A1 to C2)
Industries	Canon-relevant verticals (Finance, Retail, etc.)

Table 3: Generic Pool Features

[3] https://careers.peopleclick.eu.com/careerscp/client_canoneurope/external/search.do
https://www.linkedin.com/company/canon-emea/jobs/

Human-Like Data Variability

To simulate real-world messiness:

- Projects include intentional typos (e.g., "Brlin" for "Berlin") in fields like:
  - Work location
  - Products Involved
  - Customer Industry
  - Language Required
  - Required Skills and Expertise
- Employees remain clean (mirroring HR system records)

This forces the matching logic to rely on fuzzy similarity algorithms.

Scalar Fields in Data

Below is a list of features which have numeric/scalar values in our data. These are relatively straightforward to create.

Table	**Feature**	**Format / Data Type**
Projects	Expertise Value in Required Skills & Expertise (dictionary value)	Rating: 1 to 10
Projects	Complexity	Rating: 1 to 10
Projects	CEFR of Languages Known (dictionary value)	A1 to C2
Projects	Effort	Int: Hours Required
Projects	Requested Timeline	Date: Delivery Date
Employees	Expertise of Core Competencies (dictionary value)	Rating: 1 to 10
Employees	CEFR of Languages Known (dictionary value)	A1 to C2

Table	Feature	Format / Data Type
Employees	Cultural Awareness	Rating: 1 to 10
Employees	Problem Solving	Rating: 1 to 10
Employees	Leadership	Rating: 1 to 10

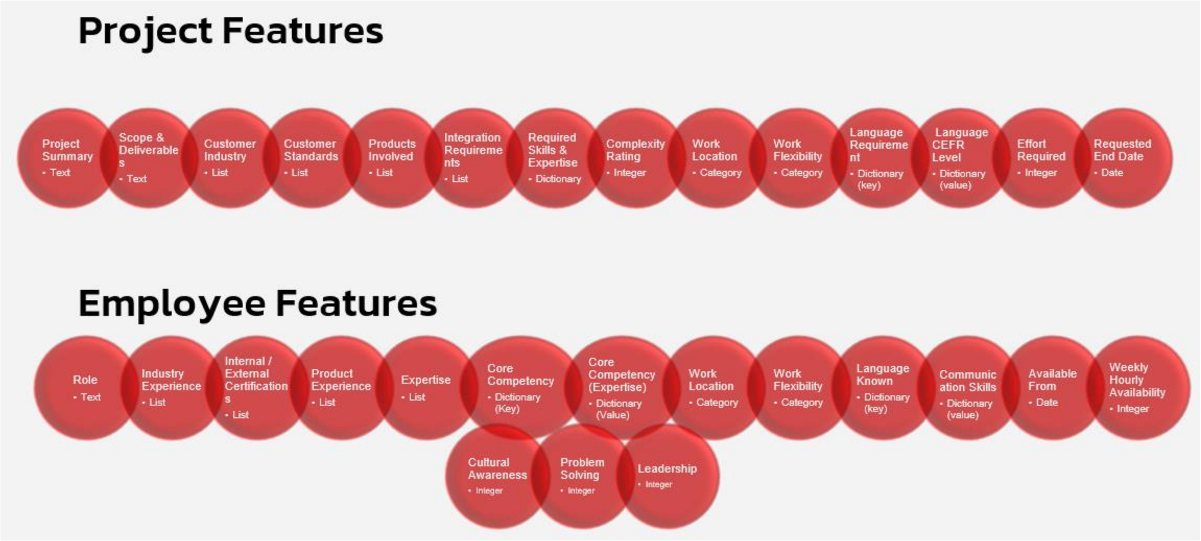
Table 4: Scalar Features

Outcome

This mock data creation strategy ensures highly realistic data tables with all relevant features are created. A combination of thematically controlled and generic features with human like data variability brings extra credibility to our mock data. This data allows full feature coverage for testing matching models and easy extensibility for demos or ML-based learning systems.

CHAPTER 3: REQUIRED DATA TABLES AND FEATURES

As discussed in the previous sections, we require two tables, namely Projects and Employees, to execute our algorithm. The features, data types and details for these tables were thoroughly discussed with Canon-EMEA. The following figures and tables give us the details on each of these tables.



Project Features
Project Summary • Text
Scope & Deliverables • Text
Customer Industry • List
Customer Standards • List
Products Involved • List
Integration Requirements • List
Required Skills & Expertise • Dictionary
Complexity Rating • Integer
Work Location • Category
Work Flexibility • Category
Language Requirement • Dictionary (key)
Language CEFR Level • Dictionary (value)
Effort Required • Integer
Requested End Date • Date
Employee Features
Role • Text
Industry Experience • List
Internal / External Certifications • List
Product Experience • List
Expertise • List
Core Competency • Dictionary (Key)
Core Competency (Expertise) • Dictionary (Value)
Work Location • Category
Work Flexibility • Category
Language Known • Dictionary (key)
Communication Skills • Dictionary (value)
Available From • Date
Weekly Hourly Availability • Integer
Cultural Awareness • Integer
Problem Solving • Integer
Leadership • Integer

Figure 1: Project and Employee Table Features

Project Table Details

Features	Description	Data Type	Details
Project Summary	Give a summary of the project	Text	Free text, ex. "This project will support the European patent attorney team by building internal systems and support processes aligned with their strategic objectives."
Scope and Deliverables	Outline the scope and list milestones	Text	Free text, ex. "Design support processes, create documentation for best practices, collaborate with business units, and track implementation progress."

Customer Industry	Specify the customer's industry (e.g., finance, healthcare)	List	Finance, Healthcare, Retail etc
Customer Preferences or Standards	Any specific standards, compliance, or customer communication preferences	List	Free text or nonfinite list, ex "Needs to be compliant with ABC100 standard" or "XYZ project methodology is a must"
Products Involved	The specific solutions the project centers on, which is crucial if certain employees have experience in those	List	List. For e.g. MVP AI SCAN, WORKFLOW2000, PRINT2.0
Integration Requirements	Any required integrations with existing systems, third-party tools, or APIs	List	Integration with ERP System, Master data exchange with rest API etc.
Required Skills and Expertise	Skill Requirements for the project e.g., JavaScript, project management, Printing	Dictionary	JavaScript, Python, Project Management, Graphic Designing, and expertise level (1 to 10 expertise level)
Complexity Rating	A subjective rating classifies the project's complexity	Integer	1 to 10 ***Note**: This value is highly subjective based on who enters the data. Junior employees can underestimate this value whereas experienced employees can consider different complications to make it more realistic. During data collection we should use something like "Planning Poker" (reference to Scrum), with group estimates based on a reference project.*
Work Location	Location of Work Office	Category	Canon Offices Cities/European City
Work Flexibility	Indicate whether the project can be remote, hybrid, or if it requires on-site work in specific locations	Category	Value from either of onsite, Remote, Hybrid
Language Requirements	Languages required to communicate in this project	Dictionary (key)	Languages like English, German, French, Portuguese etc. (nonfinite list can put everything else in others)

Language Level	CEFR Level of languages required to communicate in this project	Dictionary (value)	A1, A2, B1, B2, C1, C2
Effort	Estimated workload in hours required for the project	Integer	Estimated hours required to complete the project
Requested End	Desired End Date	Date	Desired date for project completion

Table 5: Project Dataset Details

Employee Table Details

Features	Description	Data Type	Details
Role	Job Description	Text	For e.g. "Provides high-quality patent services including drafting, filing, and prosecuting patents"
Industry Experience	Specify industries experienced in	List	Finance, Healthcare, Retail etc
Internal /External Certifications	Relevant certifications (e.g., PMP, Six Sigma).	List	For e.g. PMP, Six Sigma certifications etc.
Product Experience	Canon Product Names	List	List. For e.g. MVP AI SCAN, WORKFLOW2000, PRINT2.0
Expertise	Define expertise of employee	List	For e.g. i.e. Scripting, Integration, Color Reproduction, Cloud & Infrastructure
Core Competencies	Specific skills or technologies, e.g., programming languages, project management methodologies.	Dictionary (key)	For e.g. JavaScript, Python, Project Management, Graphic Designing
Core Competencies (Expertise)	Rating Specific skills or technologies, e.g., programming languages, project management methodologies.	Dictionary (value)	Core Competencies expertise level (1 to 10 expertise level)
Work Location	Location of Work Office	Category	Canon Offices Cities/European City

Work Flexibility	Indicate whether the project can be remote, hybrid, or if it requires on-site work in specific locations	Category	Value from either of Onsite, Remote, Hybrid
Languages Known	Languages required to communicate in this project	Dictionary (key)	Languages like English, German, French, Portuguese etc
Communication Skills	CEFR Level of languages required to communicate in this project	Dictionary (value)	A1, A2, B1, B2, C1, C2
Available From	Availability Start Date	Date	Date from which employee has working bandwidth
Weekly Availability in Hours	Hourly Bandwidth per Week	Integer	The number of hours employee is free to work per week
Cultural Awareness	Openness to diversity and experience in international settings, valuable for global teams.	Integer	Openness Rating 1 to 10
Problem Solving	Critical thinking and adaptability in projects	Integer	Past Project Diversity Rating 1 to 10
Leadership	Leadership or mentoring/coaching experience.	Integer	Past Experience Rating 1 to 10

Table 6: Employee Dataset Details

Chapter 4: Matching and Scoring Criteria

Using the above-mentioned approaches, mock data necessary for implementing our matching algorithm is created. The next step was to define how different features would interact with each other (matching employee and project features) to generate scores for different subgroups that are used for final scoring and employee ranking. The following table gives the complete summary of the matching features between projects and employees data sets and the scoring methods used. Each scoring method is described in detail in the next section. The score notes column gives extra information like which features combine to create the respective score and what names we assign to the score.

Project Feature	Employee Feature	Matching Method	Score Notes
Project Summary	Role	Text Embedding / Cosine Similarity	**Job Description Match Score**: General fit & thematic similarity based on text embeddings and cosine similarity
Scope and Deliverables	Role	Text Embedding / Cosine Similarity	
Customer Industry	Industry Experience	Category Similarity Coverage (Fuzzy match with threshold)	**Industry Match Score**
Customer Preferences or Standards	External Certifications / Internal Certifications	Category Similarity Coverage (Without Fuzzy Match)	**Certification Match Score**
Products Involved	Product Experience	Category Similarity Coverage (Fuzzy match with threshold)	**Product Match Score**
Integration Requirements	Expertise	Category Similarity Coverage (Without Fuzzy Match)	**Expertise Match Score**
Required Skills and Expertise	Core Competencies	Category Similarity Coverage (Fuzzy match with threshold)	**Skill Match Score:** Primary skill matching coverage followed by comparing Employee
Complexity Rating	Core Competencies (proficiency level)	Coverage x proficiency fit (employee capability	

		score) then compare complexity	Capability based on skills to Complexity
Work Location	Work Location	Category Similarity (Fuzzy match with threshold). Irrelevant if remote	**Location Match Score:** Location coverage followed by Remote/Hybrid/On-site scoring factoring for location coverage
Work flexibility	Work flexibility	Category Similarity (Fuzzy match with threshold)	
Language Requirements	Languages Known	Category Similarity Coverage (Fuzzy match with threshold)	**Language Match Score:** Check if required language is present for coverage. Followed by scoring using match or near-match on required fluency factoring for language coverage
Language Level	Communication skills (proficiency level)	Coverage x proficiency fit (language capability)	
Effort	Weekly Availability in Hours	Availability Score Calculation	**Availability Score:** Statistical Algorithm for calculation
Requested End	Available From	Availability Score Calculation	
-	Cultural Awareness	Optional Bonus (Normalized Score)	Used as soft bonus
-	Problem Solving	Optional Bonus (Normalized Score)	Used as soft bonus
-	Leadership	Optional Bonus (Normalized Score)	Used as soft bonus

Table 7: Matching and Scoring Criteria

Text Embedding with Cosine Similarity

Used to calculate: Job Description Match Score

To evaluate how well an employee's role matches the intent of a project, **Project Summary** and **Scope and Deliverables** fields were compared with the employee's **Role Description**.

Hugging Face AI BAAI General Embedding [4] was used to convert text into vector embeddings and then we use **cosine similarity** to compute the semantic closeness between the text pairs.

The two similarity scores (one for the summary, one for the scope) are **averaged** to produce a final Job description match score, representing how relevant an employee's responsibilities are to the project described.

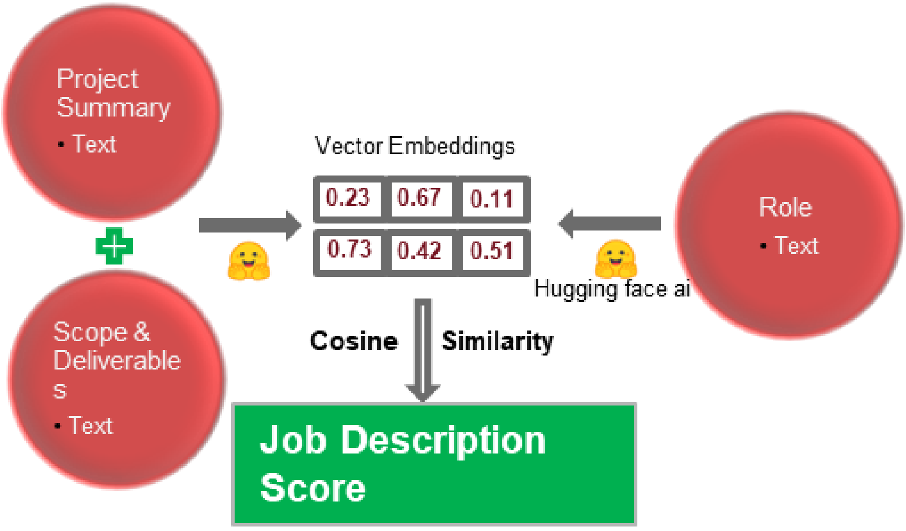


Figure 2: Job Description Match Score using Vector Embeddings and Cosine similarity

Comparison Pair	**Description**	**Output**
Project Summary vs Role Description	Measures how well the overall project aligns with the role	summary_sim (0–1)
Scope and Deliverables vs Role Description	Measures how well the specific responsibilities align	scope_sim (0–1)
Job Description Score	Average of the two scores	(summary_sim + scope_sim) / 2

[4] BAAI/BGE-large on Hugging Face is a large language model embedding model. It's used to generate embeddings, which are vector representations of text, for a variety of tasks like: semantic search, text similarity, and more. https://huggingface.co/BAAI/bge-large-en-v1.5

Table 8: Job Description Score calculation

Category Similarity Coverage (Without Fuzzy Match)

Used to calculate: Certification Match Score or Expertise Match Score

A list of certifications are present in the projects table that the project requires. Similarly, a list of certifications that the employee has completed would be present in the employees table. The coverage is calculated based on how many required certifications are possessed by the employee that the project requires. For example, if a project requires Microsoft Azure Certification and ISO 27001 and an employee has only ISO 27001, the coverage would be 50%. If an employee has completed both of the mentioned certifications (or additional) then coverage would be 100%. If no certifications are required, the employee gets a full score (100%). Note that the same process is used to calculate Expertise Match Score as well.

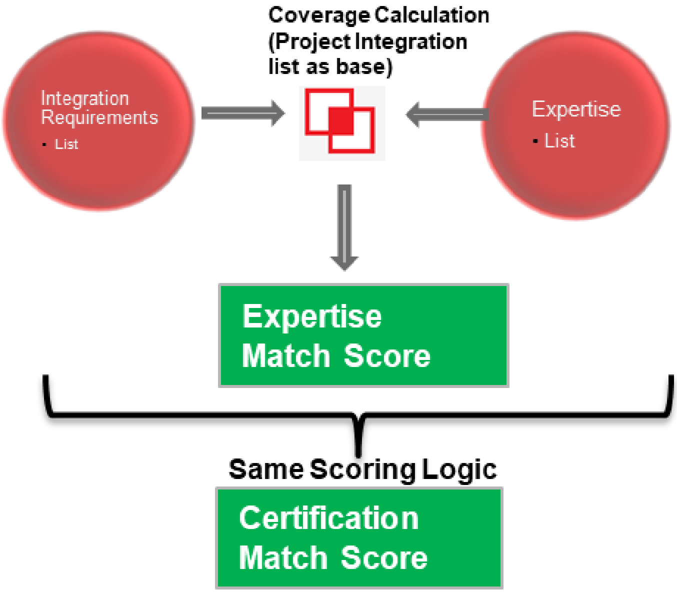


Figure 3: Certification or Expertise Match Score using category similarity coverage (without fuzzy match)

Category Similarity Coverage (With Fuzzy Match)

Used to calculate: Product Match Score or Industry Match Score

A list of products is present in the projects table and a similar list is in the employees table. As this field in projects data is human filled, we first use fuzzy matching to get rid of any mismatching (due to spelling mistakes) with product names in employee's table then we match the columns. The coverage is calculated based on how many required products are possessed by the employee. For example, if a project requires products AIScan, Print2.0 and Workflow2000 and employee knows only AIScan, the coverage would be 33%. If an employee knows any two of the above then 66% and in case of knowing all three or more, it would be 100%. Note that the same process is used to calculate Industry Match Score as well.

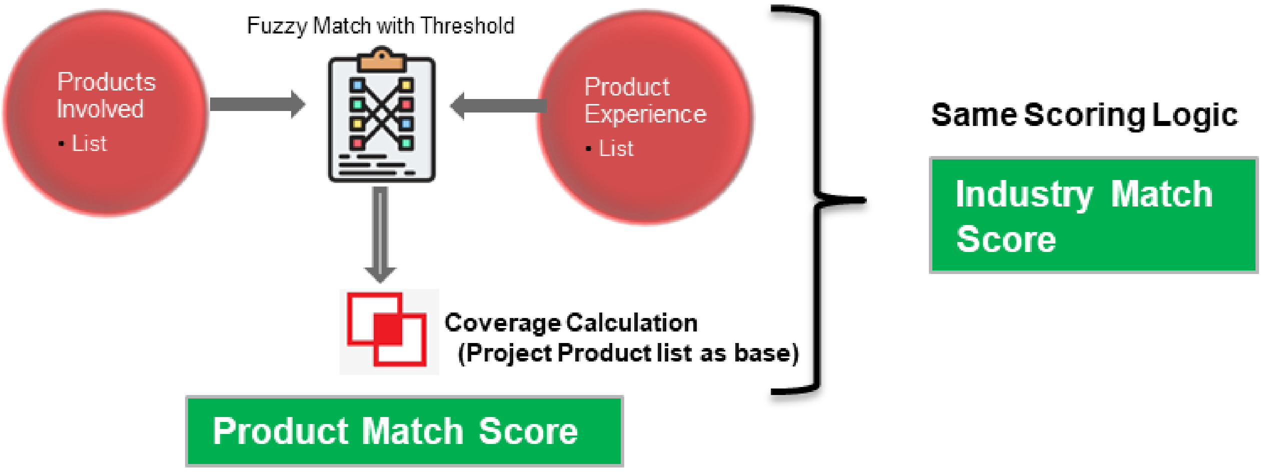


Figure 4: Product or Industry Match Score using category similarity coverage (with fuzzy match)

Category Similarity Coverage (With Fuzzy Match) with Proficiency Fit

Used to calculate: Skill Match Score or Language Match Score

Scoring is done on how well an employee's core competencies align with the skills required for a project, considering:

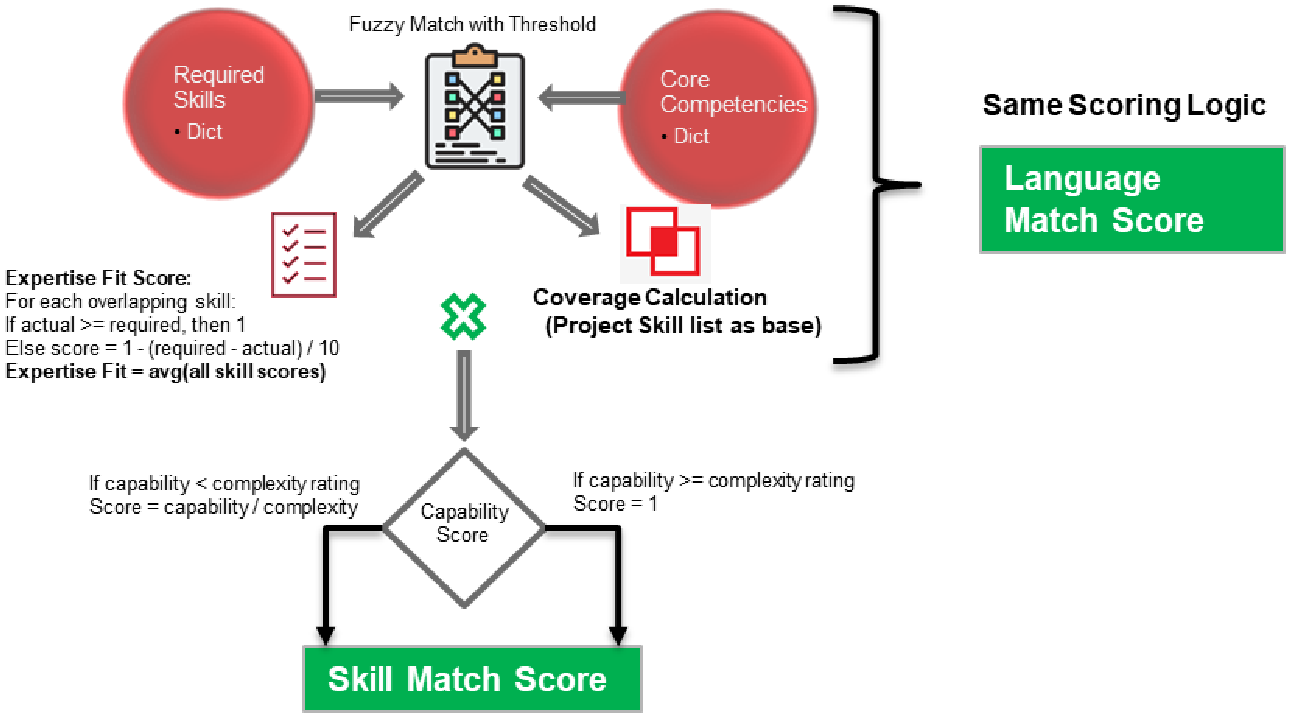
1. **Coverage** — How many required skills they know.
2. **Expertise Fit** — How experienced are they in those skills.
3. **Complexity Fit** — How well their skills meet the challenge of the project

Steps for calculating Skill Score:

1. **Fuzzy Skill Matching**
   Match each required skill (may contain typos) with the closest skill in the employee's core competency using fuzzy logic.
2. **Coverage Score**
   Calculate what fraction of required skills are present in the employee's skill set.
   Formula:
   coverage = matched skills / total required skills

3. **Proficiency Fit**
   For **each matched skill**, compare employee's level to the required level:
   - If level is equal or higher → score = 1.0
   - If lower → score = 1 - (required - actual) / 10
   - Average these for **proficiency fit**.
4. **Capability Score**
   Multiply **coverage** by **proficiency fit**
   capability = coverage × proficiency fit
5. **Final Skill Score (Complexity Fit)**
   Compare capability with project complexity rating (normalized as complexity / 10):
   - If capability ≥ complexity → score = 1.0
   - Else → capability / complexity

Note: The same process excluding the last step (complexity fit) is used to calculate Language Match Scores as well. Capability Score when calculated for Language columns would result in Language match score.



Required Skills
• Dict
Fuzzy Match with Threshold
Core Competencies
• Dict
Same Scoring Logic
Language Match Score
Expertise Fit Score:
For each overlapping skill:
If actual >= required, then 1
Else score = 1 - (required - actual) / 10
Expertise Fit = avg(all skill scores)
Coverage Calculation
(Project Skill list as base)
If capability < complexity rating
Score = capability / complexity
Capability Score
If capability >= complexity rating
Score = 1
Skill Match Score

Figure 5: Skill or Language Match Score using category similarity coverage (with fuzzy match) with Proficiency Fit

Rule-Based Scoring

Used to calculate: Location Match Score

This calculation assigns a **location match score** based on both **work flexibility compatibility** and **Location similarity (using fuzzy matching)**

Note: As Locations are filled by Humans, to match them we would use fuzzy matching.

The figure below describes the scoring mechanism.

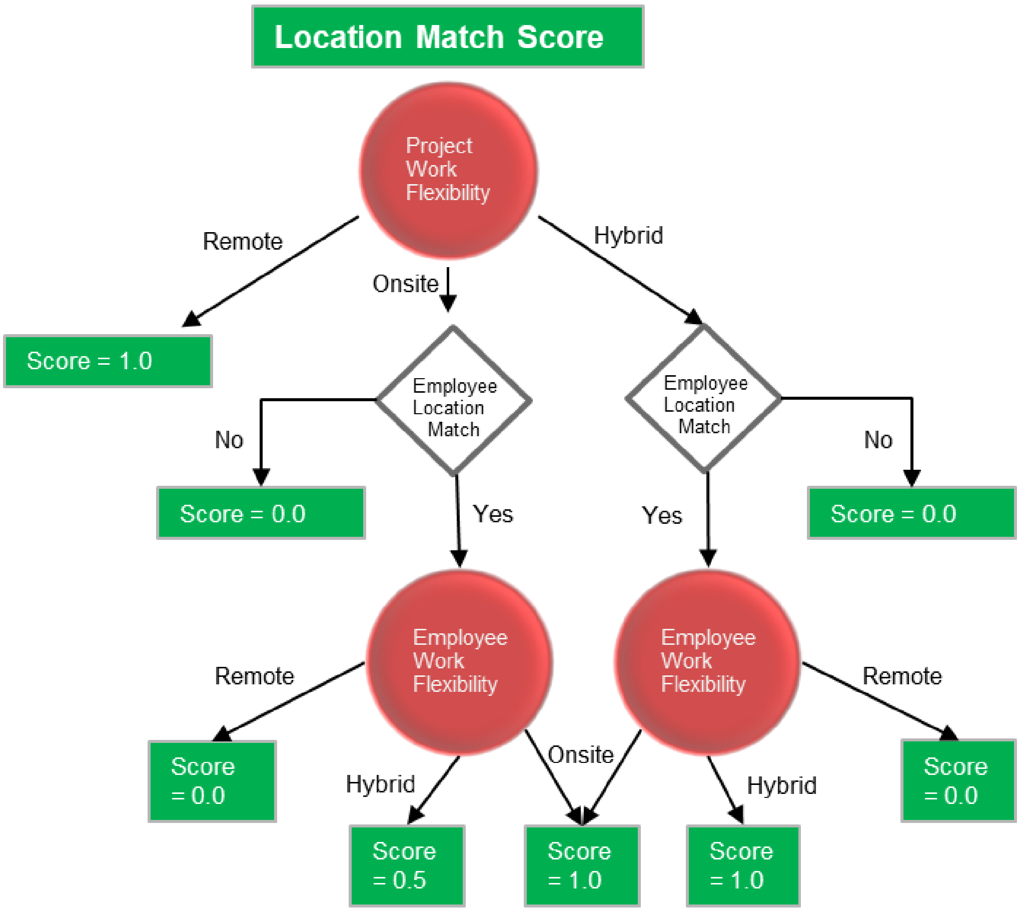


Figure 6: Location Match Score using a rule-based approach

Statistical Scoring

Used to calculate: Availability Score

The **Availability Score** measures whether an employee can complete a project within the required time frame based on their availability start date and weekly working capacity.

Steps For scoring:

1. **Compare dates**: First employee's Available From date is compared to project's Requested End date. If the employee is available only after Requested End Date then the score is 0.0 (not available).

2. **Estimate working time**:
   - Calculate total **calendar days** between employee's Available From Date and Project's Requested End Date.
   - Convert the calendar days to **working days** using a 5/7 multiplier (for a standard 5-day workweek).
   - Derive **available working weeks** by dividing working days by 5.
3. **Calculate required effort per week**:
   - Required Weekly Effort = Project Effort / Available Weeks
4. **Score logic**:
   - If Required Weekly Effort ≤ Employee Weekly Capacity, then we say the score is 1.0
   - Otherwise, the score is a decimal ratio (e.g. 0.7), based on how much of the requirement the employee can handle.

Additionally, an **Available** flag is created that is 1 if Availability Score is 1 and 0 otherwise. The reason to create "Availability Score" and "Available" flag separately are that if no instance has Available Flag as 1, we can consider high Availability Score employees for the project.

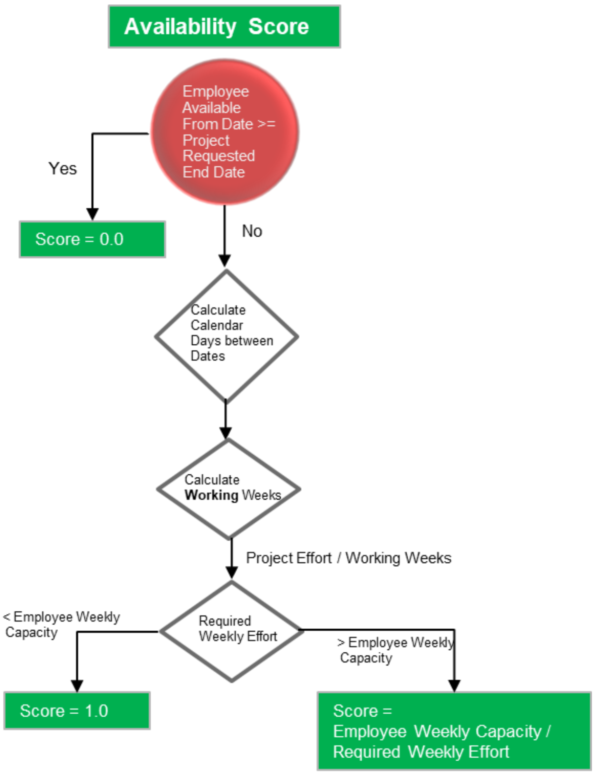

Availability Score
Employee Available From Date >= Project Requested End Date
Yes
Score = 0.0
No
Calculate Calendar Days between Dates
Calculate Working Weeks
Project Effort / Working Weeks
Required Weekly Effort
< Employee Weekly Capacity
Score = 1.0
> Employee Weekly Capacity
Score = Employee Weekly Capacity / Required Weekly Effort

Figure 7: Availability Score using a statistical approach

Bonus Scores for Employees

Bonus scores are added for employees based on their profile. Employee's cultural awareness, problem solving and leadership scores are used for calculating bonus scores. These 3 features are scored from 1 to 10 and normalized from 0 to 1 before adding to the employee score.

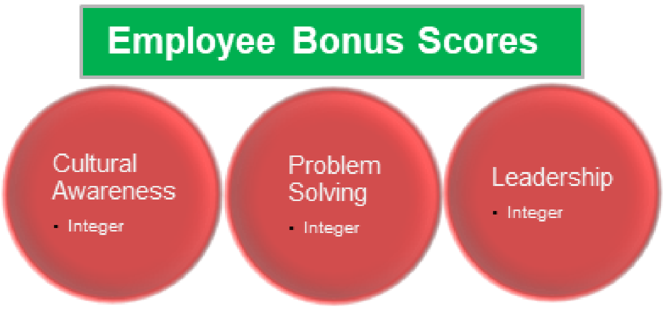


Figure 8: Bonus Scores

Final Ranking

To calculate **the final ranking of employees for each project**, a structured multi-step approach is adopted. A set of adjustable weights for each of the score dimensions discussed above (except availability score) were defined, such as job description, product, location, skill match scores etc. even including the bonus scores. The defined weights sum up to 1 to scale the final scores from 0 to 1. These weights reflect the relative importance of each criterion in determining a good project–employee fit. Using these weights, a weighted score was computed for each employee–project pair by multiplying each individual score with its corresponding weight and summing the results. This resulted in a **comprehensive final score** representing the overall suitability of each employee for a given project. After computing the final scores, an option to filter by buckets of availability scores such as 1, > 0.75, > 0.5 etc is available. This is to give flexibility to the user to check the tradeoff between their choice of employees and their respective availability. Finally, the data is sorted by ProjectID, and the employees are ranked in descending order of their weighted scores. This gives the ranking of each employee based on project criteria. This weighted scoring and ranking process ensured that employee recommendations were both quantitative and context-aware, based on a configurable combination of hard skills, logistics, experience, and soft skills.

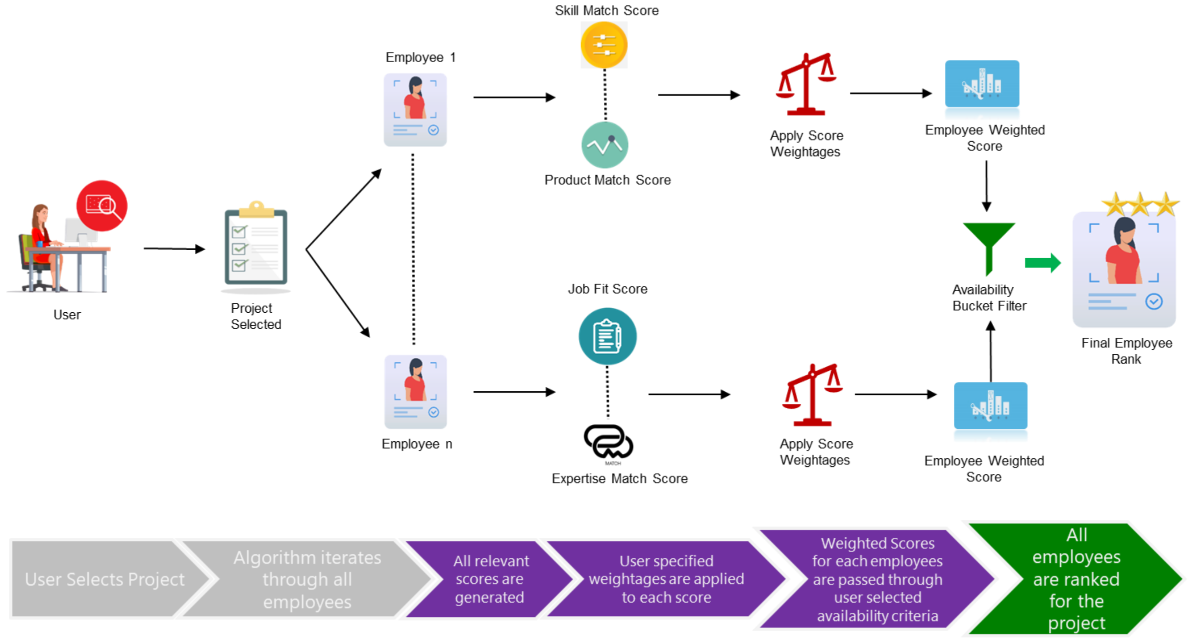

User
Project Selected
Employee 1
Employee n
Skill Match Score
Product Match Score
Job Fit Score
Expertise Match Score
Apply Score Weightages
Apply Score Weightages
Employee Weighted Score
Employee Weighted Score
Availability Bucket Filter
Final Employee Rank
User Selects Project
Algorithm iterates through all employees
All relevant scores are generated
User specified weightages are applied to each score
Weighted Scores for each employees are passed through user selected availability criteria
All employees are ranked for the project

Figure 9: Employee Rank Calculation per Project

Evaluation of Employee Ranks

To evaluate how well employees matched various projects, a custom scoring algorithm was developed to align with the client's specific requirements. In the absence of historical data or predefined benchmarks, an iterative and collaborative approach was adopted to refine results. Using an “arena-style” method, employee scores and rankings were tested and continuously refined based on direct client feedback. This process enabled improvements to both the input data quality and the scoring logic. Through several rounds of iteration, the algorithm was finetuned to closely reflect the client’s expectations. The results demonstrated high sophistication in matching employees to projects, leading to client satisfaction.

CHAPTER 5: HR MATE AI DASHBOARD

Scope of the HR Mate AI Proof-of-Concept

This Capstone Project delivered a functional proof-of-concept encompassing the core functionalities required for an intelligent employee-project matching system encapsulating mock data creation (chapter 2 and 3) and employee scoring mechanism (chapter 4). For demonstration purposes a dashboard was created called HR Mate AI to implement our matching system.

In-Scope Functionalities:

- **Comprehensive Data Definition:** Detailed feature sets and data types for employee and project entities were defined, as outlined in Chapter 2 & 3.

- **AI-Augmented Synthetic Data Generation:** Python scripts (generate_employees.py, generate_projects.py, common_data.py) were developed to produce mock employee and project data, with textual fields (like "Role Description" and "Project Summary") enriched using the Google Gemini API. Data output was in JSON format.

- **Semantic Search and Retrieval Engine:** A backend system (retriever.py) was implemented using the BAAI/bge-large-en-v1.5 embedding model and Chroma DB [5] for persistent storage and efficient querying of employee profile embeddings.

- **Multi-Criteria Scoring Engine:** A sophisticated scoring module (scorer.py) was developed, featuring configurable weights (config.py) to assess candidates across numerous dimensions including skill match, availability, product experience,

[5] ChromaDB for embedding storage and querying: https://docs.trychroma.com/docs/overview/introduction

location/flexibility, language proficiency, industry fit, certifications, expertise, and select employee attributes (e.g., cultural awareness).

- **Interactive Web User Interface:** A Streamlit application [6] (streamlit_app.py) was created, providing features for project selection, display of ranked candidate lists, detailed employee profiles with visual score breakdowns (radar charts), and interactive controls for filtering results and adjusting scoring weights.
- **Basic UI Customization:** The UI incorporated Canon's logo and was designed with a professional aesthetic.

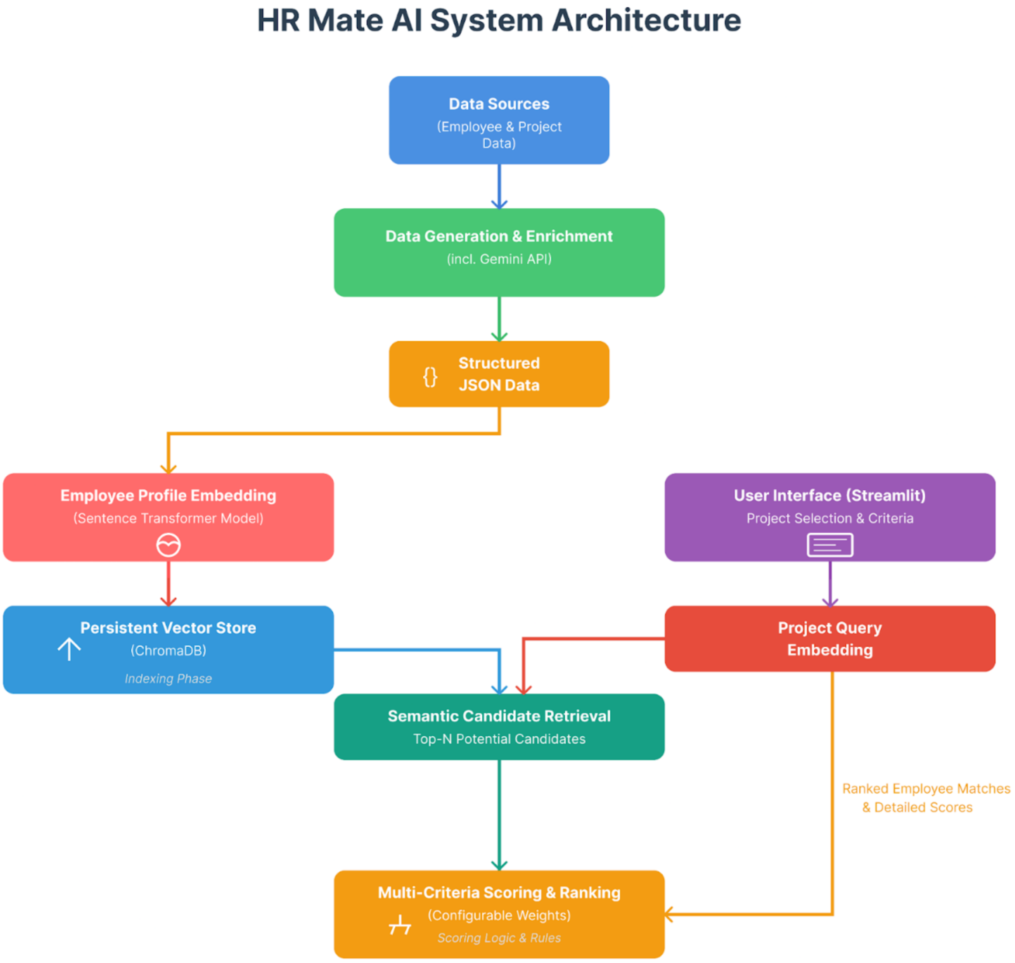

HR Mate AI System Architecture
Data Sources
(Employee & Project Data)
Data Generation & Enrichment
(incl. Gemini API)
Structured JSON Data
Employee Profile Embedding
(Sentence Transformer Model)
User Interface (Streamlit)
Project Selection & Criteria
Persistent Vector Store
(ChromaDB)
Indexing Phase
Project Query Embedding
Semantic Candidate Retrieval
Top-N Potential Candidates
Ranked Employee Matches & Detailed Scores
Multi-Criteria Scoring & Ranking
(Configurable Weights)
Scoring Logic & Rules

Figure 10: Dashboard Architecture – HR Mate AI

[6] Streamlit Documentation: https://docs.streamlit.io/

Deployment Preparation: The project was structured for deployment on Streamlit Community Cloud[7], including the creation of a requirements.txt file and considerations for secure API key management.

Architectural Overview: A Modular Approach

The architecture of HR Mate AI is predicated on a modular design, where distinct functional units handle specific stages of the matching process. This approach allows for independent development and testing of each module and provides flexibility for future enhancements or replacements of individual components without necessitating a complete system overhaul.

The HR Mate AI system is comprised of several distinct but interconnected components, each fulfilling a specific role in the overall matching process:

Data Generation Module

- **Purpose:** To create the foundational datasets of employee profiles and project specifications required for the system's operation and testing.

- **Functionality:** This module programmatically constructs employees and projects records based on predefined schemas (as discussed in section 2 & 3). It utilizes lists of common attributes (such as job titles, skills, industry types, and locations) to ensure a degree of realism and consistency. A key feature is its integration with Google's Gemini API, which is employed to generate rich, contextually relevant textual content for fields

[7] Streamlit Community Cloud Documentation
https://docs.streamlit.io/deploy/streamlit-community-cloud

like "Role Description" in employee profiles and "Project Summary" or "Scope and Deliverables" in project descriptions. This AI-driven text generation enhances the natural language quality and diversity of the synthetic data. The module also incorporates logic to introduce minor, human-like variations (e.g., typos in project data fields) to simulate real-world data imperfections, ensuring the subsequent matching and scoring algorithms are robust. 100 projects and 1000 employees were generated for our demo. Each entry is theme-aware and realistically structured. The system is extensible, it can generate 10000+ unique rows for projects or employees with identical consistency by using different feature combinations.

- **Interaction:** It produces structured JSON files that serve as the input for the Semantic Search and Retrieval Module (for indexing employee data) and the User Interface (for selecting projects).

Semantic Search and Retrieval Module (Chroma DB & Embedding Model)

- **Purpose:** To perform an initial, intelligent filtering of the employee pool by identifying candidates whose profiles semantically align with the core requirements of a given project.
- **Functionality:** This module is built around two key technologies:
  - **Sentence Embedding Model (**This powerful NLP model (Hugging Face) converts textual descriptions from employee profiles (e.g., role descriptions, summaries of experience) and project queries into dense vector representations (embeddings). These embeddings capture the underlying meaning and context of the text, going beyond simple keyword matching.

  - **Vector Database (Chroma DB):** Chroma DB is used to store and efficiently index the employee embeddings generated along with their associated metadata (the full employee profile). It is configured for persistent storage, meaning the index is saved to disk and reloaded, avoiding the need to re-embed all employee profiles every time the application starts. When a project query is received, it's embedded, and Chroma DB performs a similarity search to find the employee embeddings that are closest (most similar) in the vector space.

- **Interaction:** It receives project query details from the User Interface. It accesses the pre-indexed employee embeddings from Chroma DB. It outputs a list of top-N semantically relevant candidate employee profiles (including their raw data and initial similarity scores) to the Multi-Criteria Scoring Module.

Multi-Criteria Scoring Module

- **Purpose:** To conduct a detailed, granular assessment of the candidates shortlisted by the semantic retrieval module, evaluating them against a comprehensive set of specific project requirements and preferences.

- **Functionality:** This module implements a sophisticated scoring algorithm. For each candidate, it calculates individual scores across various dimensions mentioned in chapter 4.

- **Interaction:** It receives the list of candidate employees from the Semantic Search and Retrieval Module and the detailed requirements of the selected project from the User Interface. It uses scoring weights defined in the system's configuration. It outputs a ranked list of candidates, each with a detailed breakdown of their individual and overall scores, back to the User Interface.

Interactive User Interface Module (Streamlit)

- **Purpose:** To serve as the primary point of interaction for the end-user, enabling them to define matching needs, trigger the process, and analyze the results.
- **Functionality:** Developed using the Streamlit framework, this web application provides:
    - Intuitive controls for selecting a project for which to find employees.
    - Display of detailed information for the selected project.
    - Sidebar controls allowing users to specify the number of top candidates to retrieve, filter results based on availability scores and dynamically adjust the weights of different scoring criteria to see their impact on rankings.
    - A clear, ranked presentation of matched employees.
    - Expandable sections for each candidate, offering:
        - A summary of key employee information.
        - A visual score profile (radar chart) illustrating their performance across various matching dimensions.
        - Detailed tabs for attribute-by-attribute comparison against project requirements, a breakdown of all individual scores contributing to the overall match, and access to the raw employee data.

The UI is designed to be clear, informative, and easy to navigate, incorporating visual elements like the Canon logo for branding.

- **Interaction:** It receives user input for project selection and matching preferences. It orchestrates calls the Semantic Search and Retrieval Module and the Multi-Criteria Scoring Module. It then visualizes the processed results and detailed analytics for the user.

These core components are designed to operate in a coordinated manner, transforming raw data and user queries into actionable intelligence for employee-project matching.

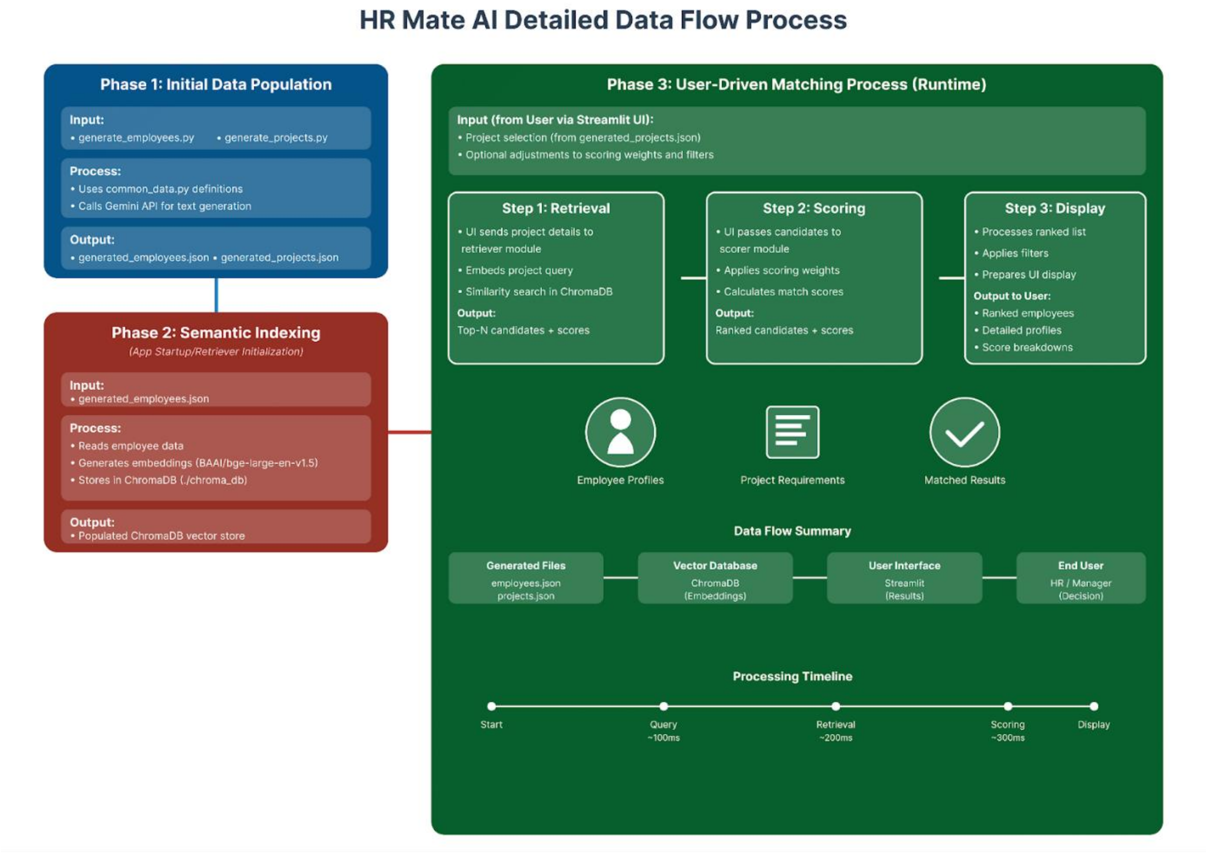

HR Mate AI Detailed Data Flow Process
Phase 1: Initial Data Population
Input:
• generate_employees.py
• generate_projects.py
Process:
• Uses common_data.py definitions
• Calls Gemini API for text generation
Output:
• generated_employees.json • generated_projects.json
Phase 2: Semantic Indexing
(App Startup/Retriever Initialization)
Input:
• generated_employees.json
Process:
• Reads employee data
• Generates embeddings (BAAI/bge-large-en-v1.5)
• Stores in ChromaDB (./chroma_db)
Output:
• Populated ChromaDB vector store
Phase 3: User-Driven Matching Process (Runtime)
Input (from User via Streamlit UI):
• Project selection (from generated_projects.json)
• Optional adjustments to scoring weights and filters
Step 1: Retrieval
• UI sends project details to retriever module
• Embeds project query
• Similarity search in ChromaDB
Output:
Top-N candidates + scores
Step 2: Scoring
• UI passes candidates to scorer module
• Applies scoring weights
• Calculates match scores
Output:
Ranked candidates + scores
Step 3: Display
• Processes ranked list
• Applies filters
• Prepares UI display
Output to User:
• Ranked employees
• Detailed profiles
• Score breakdowns
Employee Profiles
Project Requirements
Matched Results
Data Flow Summary
Generated Files
employees.json
projects.json
Vector Database
ChromaDB
(Embeddings)
User Interface
Streamlit
(Results)
End User
HR / Manager
(Decision)
Processing Timeline
Start
Query
~100ms
Retrieval
~200ms
Scoring
~300ms
Display

Figure 11: Detailed Data Flow Process– HR Mate AI

Chapter 6: User Interface and Interaction: The HR Mate AI Dashboard

The user interface (UI) is the primary gateway through which users interact with the HR Mate AI system[8]. A well-designed UI is crucial for translating complex semantic search and multi-criteria scoring into an accessible, understandable, and actionable experience for HR personnel and project managers. This chapter details the objectives for the UI and provides a comprehensive walkthrough of the dashboard's key functionalities, illustrated with references to the provided screenshots.

Objectives

The development of the HR Mate AI dashboard was guided by several key design objectives aimed at maximizing usability and effectiveness:

- **Clarity:** Information, from project details to candidate rankings and score breakdowns, must be presented in a clear, organized, and unambiguous manner. Users should easily understand what they are seeing and why certain candidates are prioritized.
- **Interactivity:** The interface must allow users to actively engage with the system, such as selecting projects, adjusting matching parameters (like scoring weights or filters), and drilling down into detailed information for specific candidates.
- **Transparency:** The system's recommendations should be transparent. Users need to see not just the final match score but also the contributing factors, fostering trust in the AI-driven suggestions.

[8] HR Mate AI Dashboard: http://142.93.238.167:8501/

- **Seamless Python Integration:** As the backend logic (retrieval, scoring) is also in Python, Streamlit allows for direct and easy integration, making the entire application cohesive.

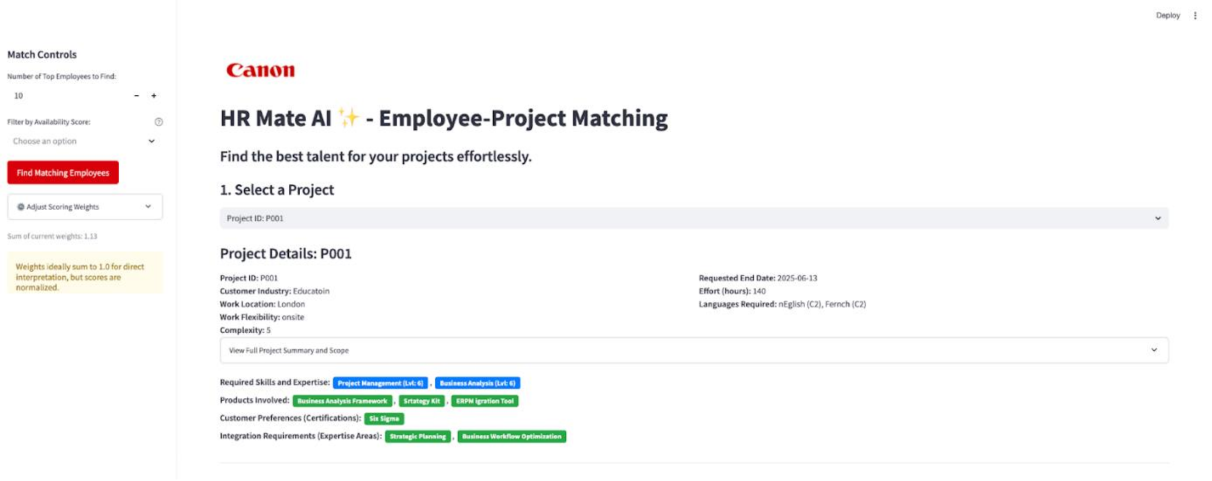
- **Ease of Deployment:** Streamlit applications can be readily deployed using platforms like Streamlit Community Cloud.

Dashboard Walkthrough and Key Functionalitie1s

The HR Mate AI dashboard is structured to guide the user logically through the process of selecting a project and finding suitable employee matches.

Initial View and Project Selection

Upon launching the application, the user is presented with the main dashboard view.



Number of Top Employees to Find: 10

Filter by Availability Score: Choose an option

Sum of current weights: 1.13

Weights ideally sum to 1.0 for direct interpretation, but scores are normalized.

HR Mate AI ✨ - Employee-Project Matching

Find the best talent for your projects effortlessly.

1. Select a Project

Project ID: P001

Project Details: P001

Project ID: P001
Customer Industry: Educatoin
Work Location: London
Work Flexibility: onsite
Complexity: 5

Requested End Date: 2025-06-13
Effort (hours): 140
Languages Required: nEglish (C2), Fernch (C2)

Required Skills and Expertise: Project Management (Lvl: 6) , Business Analysis (Lvl: 6)
Products Involved: Business Analysis Framework , Srtategy Kit , ERPN igration Tool
Customer Preferences (Certifications): Six Sigma
Integration Requirements (Expertise Areas): Strategic Planning , Business Workflow Optimization

Figure 12: Dashboard Homepage View – HR Mate AI

Display of Selected Project’s Detailed Information

Once a project is selected from the dropdown, its key details are immediately displayed below the selection area, providing the user with context before initiating the matching process.

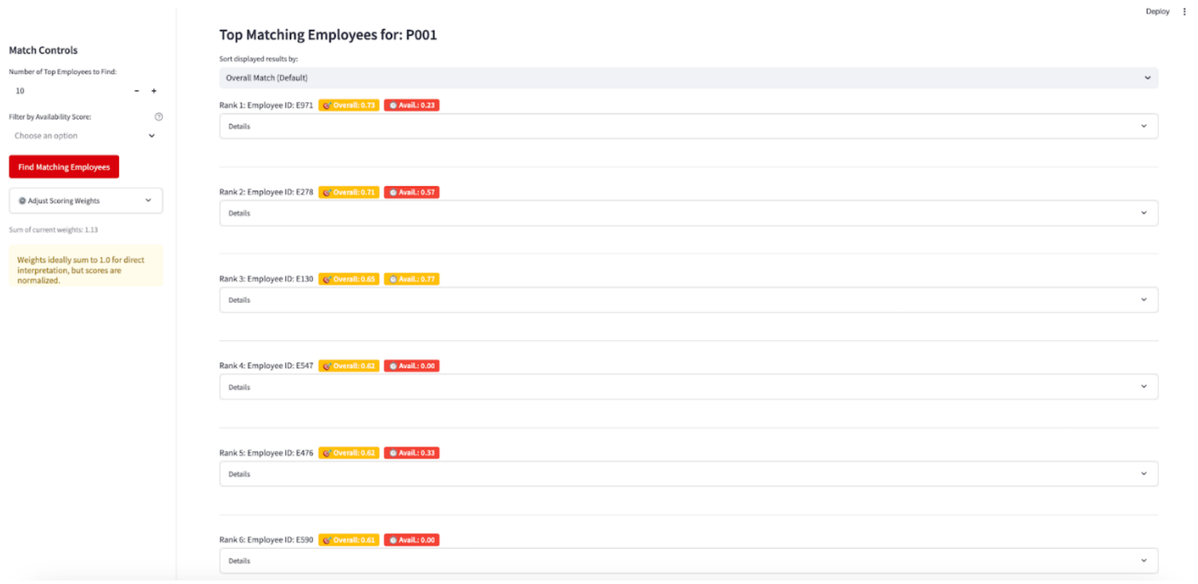
- **Structured Details:** Information such as "Project ID," "Customer Industry," "Work Location," "Work Flexibility," "Complexity," "Requested End Date," and "Effort (hours)" are clearly presented.

- **Expandable Summary and Scope:** An expander allows users to view the full "Project Summary" and "Scope and Deliverables" if they need more detailed context, keeping the initial view uncluttered.

- **Key Requirements as Chips:** Critical project requirements like "Required Skills and Expertise" (with proficiency levels), "Products Involved," "Customer Preferences (Certifications)," and "Integration Requirements (Expertise Areas)" are displayed using a "chip-like" visual style. This makes it easy to quickly scan the primary needs of the project.

Candidate Matching Results: Ranked List

After the user clicks the "Find Matching Employees" button in the sidebar (detailed in 6.4.4), the system processes the request and displays a ranked list of the most suitable employees in the main area.



Match Controls

Number of Top Employees to Find: 10

Filter by Availability Score: Choose an option

Sum of current weights: 1.13

Weights ideally sum to 1.0 for direct interpretation, but scores are normalized.

Top Matching Employees for: P001

Sort displayed results by: Overall Match (Default)

Rank 1: Employee ID: E971 Overall: 0.73 Avail.: 0.23
Details

Rank 2: Employee ID: E278 Overall: 0.71 Avail.: 0.57
Details

Rank 3: Employee ID: E130 Overall: 0.65 Avail.: 0.77
Details

Rank 4: Employee ID: E547 Overall: 0.62 Avail.: 0.00
Details

Rank 5: Employee ID: E476 Overall: 0.62 Avail.: 0.33
Details

Rank 6: Employee ID: E590 Overall: 0.61 Avail.: 0.00
Details

Figure 13: Ranked Employee List View– HR Mate AI

- **Header:** A sub header indicates which project the results are for (e.g., "Top Matching Employees for: P001").

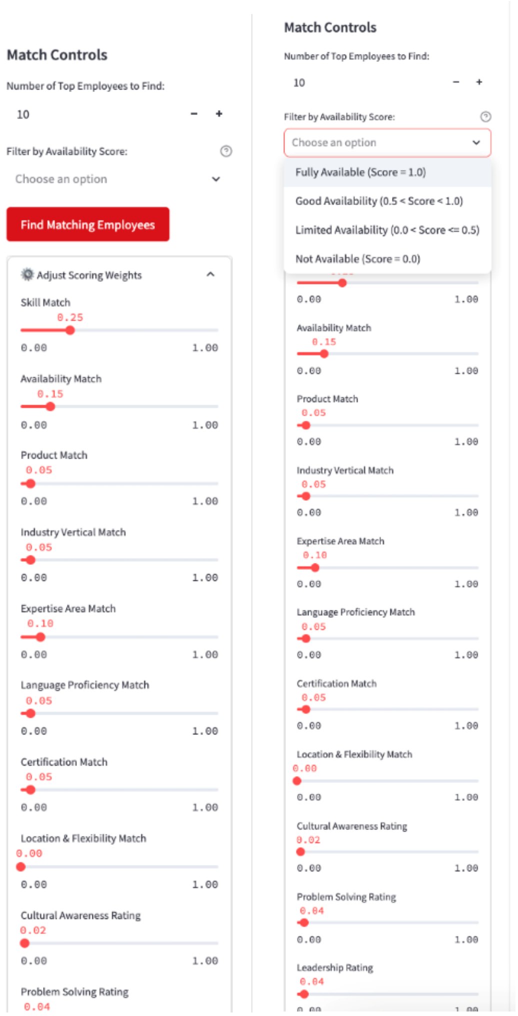
- **Individual Candidate Entries:** Each matched employee is presented in a distinct expandable section.

  - **Rank and Name:** The rank (e.g., "Rank 1," "Rank 2") and the employee's name (or ID) are prominently displayed.

  - **Score Badges:** Key scores, such as "Overall Match" and "Availability," are shown as colored badges directly in the title of the expander for quick assessment. The color of the badge often reflects the score's favorability (e.g., green for high, red for low).

  - **Expander for Details:** Each entry is an expander labeled "Details," which, when clicked, reveals a more in-depth profile of the candidate's match.

Sidebar: Match Controls



Match Controls

Number of Top Employees to Find: 10

Filter by Availability Score: Choose an option

Adjust Scoring Weights

Skill Match 0.25

Availability Match 0.15

Product Match 0.05

Industry Vertical Match 0.05

Expertise Area Match 0.10

Language Proficiency Match 0.05

Certification Match 0.05

Location & Flexibility Match 0.00

Cultural Awareness Rating 0.02

Problem Solving Rating 0.04

Match Controls

Number of Top Employees to Find: 10

Filter by Availability Score: Choose an option

Availability Match 0.15

Product Match 0.05

Industry Vertical Match 0.05

Expertise Area Match 0.10

Language Proficiency Match 0.05

Certification Match 0.05

Location & Flexibility Match 0.00

Cultural Awareness Rating 0.02

Problem Solving Rating 0.04

Leadership Rating 0.04

- +
Find Matching Employees
Fully Available (Score = 1.0)
Good Availability (0.5 < Score < 1.0)
Limited Availability (0.0 < Score <= 0.5)
Not Available (Score = 0.0)
0.00
1.00

Figure 14: Weightages and Filters on Sidebar – HR Mate AI

The sidebar provides users with interactive controls to customize the matching process and filter results:

- **Number of Top Employees to Find:** A number input field allows the user to specify how many top-ranked candidates they wish to see (e.g., 10, 20).
- **Filter by Availability Score:** A multi-select dropdown enables users to filter the displayed candidates based on their "Availability Score" buckets (e.g., "Fully Available

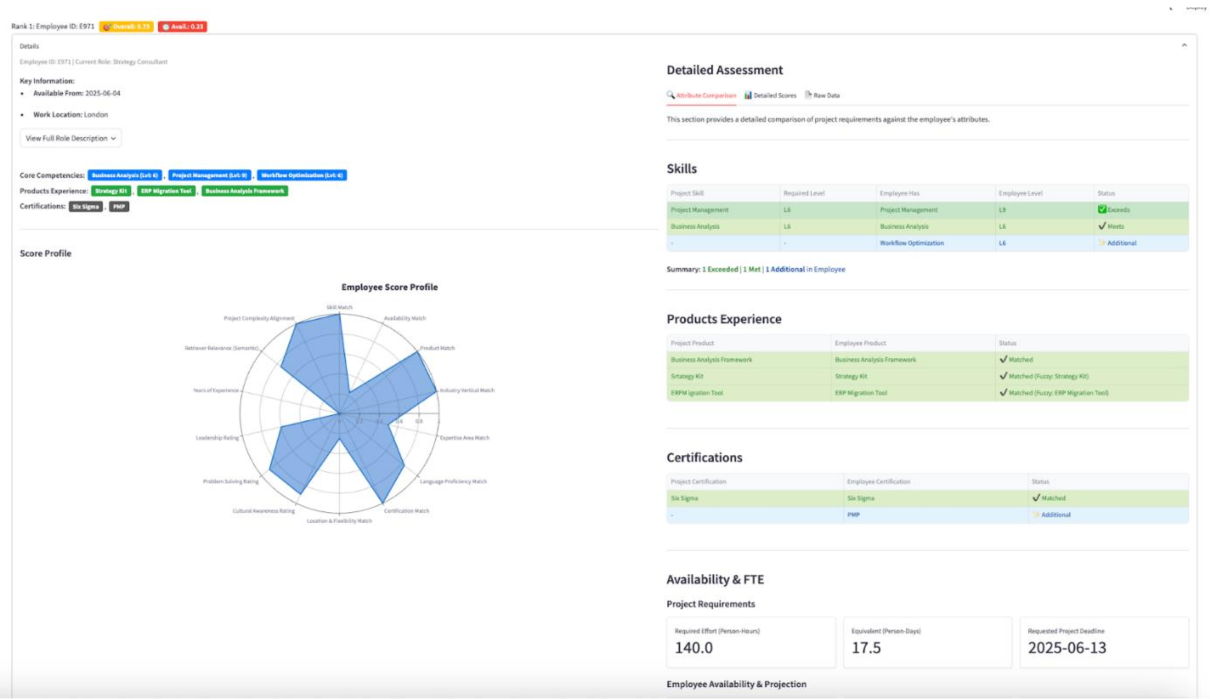
(Score = 1.0)," "Good Availability (0.5 < Score < 1.0)," etc.). This is useful for quickly identifying candidates who meet specific timeline urgencies.

- **Adjust Scoring Weights** This powerful feature allows users to dynamically change the importance (weight) of each scoring criterion (e.g., Skill Match, Product Match, Language Proficiency). Sliders are provided for each criterion, typically ranging from 0.0 to 1.0. Changes made here are immediately reflected in the st.session_state.custom_weights and are used if the "Find Matching Employees" button is clicked again, or if the results are re-sorted/re-filtered dynamically (depending on implementation). A caption shows the sum of current weights, with a warning if they don't ideally sum to 1.0 (though scores are normalized).
- **"Find Matching Employees" Button:** A primary button that triggers the backend retrieval and scoring process based on the currently selected project and control settings.

Expanded Candidate View: Detailed Assessment

When a user expands a candidate's entry from the ranked list, a detailed assessment view is presented, typically organized into columns and tabs for clarity.

- **Left Column: Key Information and Score Profile**



Rank 1: Employee ID: E971 Overall: 0.73 Avail.: 0.23

Details

Employee ID: E971 | Current Role: Strategy Consultant

Key Information:

- Available From: 2025-06-04
- Work Location: London

View Full Role Description

Core Competencies: Business Analysis (Lvl: 6) , Project Management (Lvl: 9) , Workflow Optimization (Lvl: 6)

Products Experience: Strategy Kit , ERP Migration Tool , Business Analysis Framework

Certifications: Six Sigma , PMP

Score Profile

Detailed Assessment

This section provides a detailed comparison of project requirements against the employee's attributes.

Skills

Project Skill	Required Level	Employee Has	Employee Level	Status
Project Management	L6	Project Management	L9	✅ Exceeds
Business Analysis	L6	Business Analysis	L6	✓ Meets
-	-	Workflow Optimization	L6	✨ Additional

Summary: 1 Exceeded | 1 Met | 1 Additional in Employee

Products Experience

Project Product	Employee Product	Status
Business Analysis Framework	Business Analysis Framework	✓ Matched
Srtategy Kit	Strategy Kit	✓ Matched (Fuzzy: Strategy Kit)
ERPM igration Tool	ERP Migration Tool	✓ Matched (Fuzzy: ERP Migration Tool)

Certifications

Project Certification	Employee Certification	Status
Six Sigma	Six Sigma	✓ Matched
-	PMP	✨ Additional

Availability & FTE

Project Requirements

Required Effort (Person-Hours)	Equivalent (Person-Days)	Requested Project Deadline
140.0	17.5	2025-06-13

Employee Availability & Projection

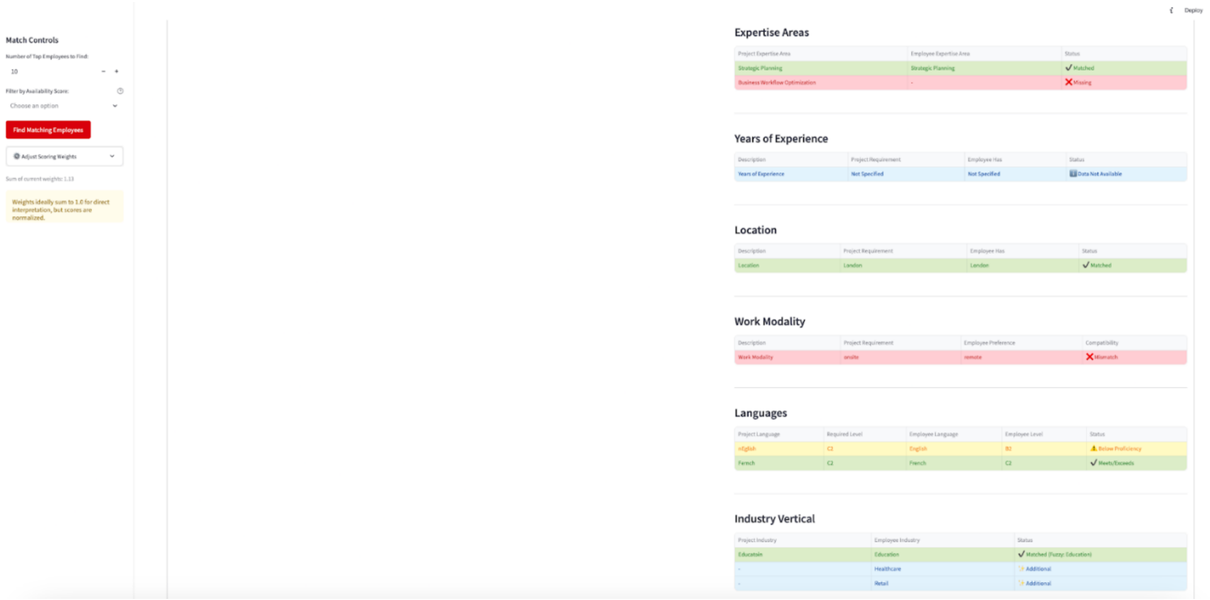
Figure 15: Employee Scoring Detail – HR Mate AI

- ○ **Employee Identifiers:** Employee ID and Current Role.
- ○ **Key Attributes:** A quick summary of "Available From," "Work Location," and a popup to view the full "Role Description."
- ○ **Competencies as Chips:** "Core Competencies" (with levels), "Products Experience," and "Certifications" are displayed using the chip visual style for easy scanning.
- ○ **Visual Score Profile: Radar Chart:** A prominent radar chart (generated by Plotly[9]) provides a visual representation of the employee's scores across various weighted dimensions (e.g., Skill Match, Availability, Industry, Language). This allows for a quick, holistic understanding of the candidate's strengths and weaknesses relative to the

[9] Plotly for Python: https://plotly.com/python/

project. The radar chart axes correspond to the SCORE_DISPLAY_NAMES and scores are normalized to a 0-1 range.

- **Right Column: Detailed Assessment Tabs**



Match Controls

Number of Top Employees to Find: 10

Filter by Availability Score: Choose an option

Sum of current weights: 1.13

Weights ideally sum to 1.0 for direct interpretation, but scores are normalized.

Expertise Areas

Project Expertise Area	Employee Expertise Area	Status
Strategic Planning	Strategic Planning	✓ Matched
Business Workflow Optimization	-	✗ Missing

Years of Experience

Description	Project Requirement	Employee Has	Status
Years of Experience	Not Specified	Not Specified	ℹ Data Not Available

Location

Description	Project Requirement	Employee Has	Status
Location	London	London	✓ Matched

Work Modality

Description	Project Requirement	Employee Preference	Compatibility
Work Modality	onsite	remote	✗ Mismatch

Languages

Project Language	Required Level	Employee Language	Employee Level	Status
nEglish	C2	English	B2	⚠ Below Proficiency
Fernch	C2	French	C2	✓ Meets/Exceeds

Industry Vertical

Project Industry	Employee Industry	Status
Educatoin	Education	✓ Matched (Fuzzy: Education)
-	Healthcare	✨ Additional
-	Retail	✨ Additional

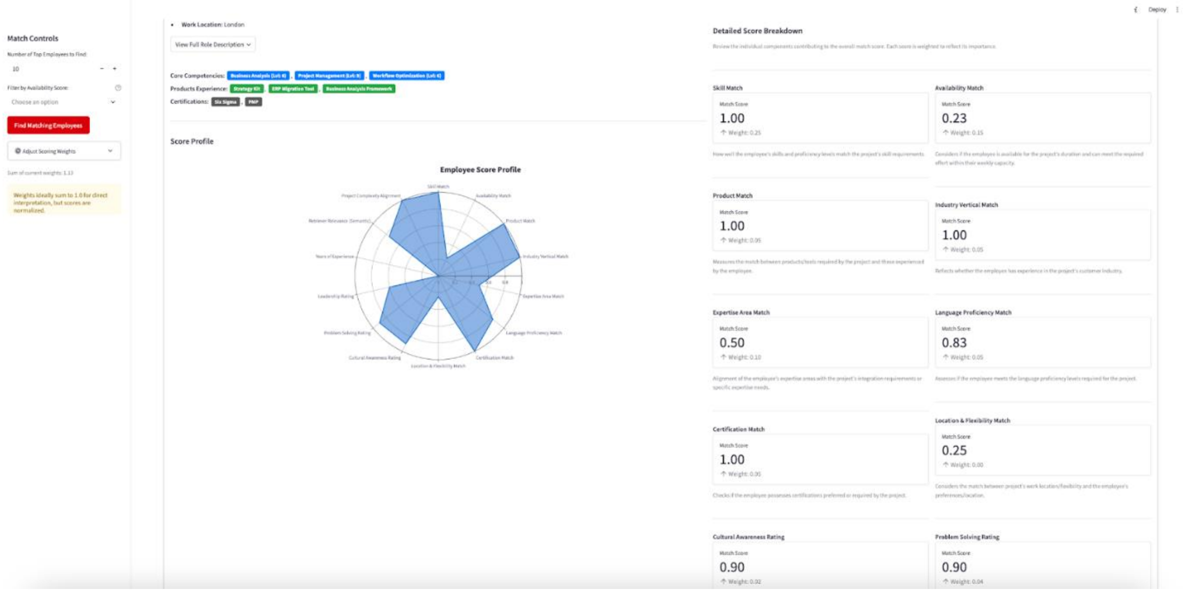
Figure 16: Employee Scoring Detail Attribute Comparison – HR Mate AI

This section uses tabs to organize different types of detailed information:

- **Tab 1: Attribute Comparison**
  - **Purpose:** Provides a direct, side-by-side comparison of specific project requirements against the employee's corresponding attributes.
  - **Content:** For each configured attribute (Skills, Products Experience, Certifications, Availability, Expertise Areas, Location, Work Modality, Languages, Industry Vertical), this tab shows:

        - The project's requirement.
        - What the employee possesses/prefers.
        - A "Status" indicating the level of match (e.g., "✅ Exceeds," "✔️ Meets," "⚠️ Below," "❌ Missing," "✨ Additional," "✔️ Matched (Fuzzy)").
    - The presentation often uses styled dataframes or custom markdown to highlight matches and mismatches with colors and icons, making it easy to pinpoint areas of strength or concern.

- **Tab 2: Detailed Scores**



Match Controls

Number of Top Employees to Find: 10

Filter by Availability Score: Choose an option

Find Matching Employees

Adjust Scoring Weights

Sum of current weights: 1.13

Weights ideally sum to 1.0 for direct interpretation, but scores are normalized.

- **Work Location:** London

View Full Role Description

Core Competencies: Business Analysis (Lvl: 4), Project Management (Lvl: 3), Workflow Optimization (Lvl: 4)

Products Experience: Strategy Kit, ERP Migration Tool, Business Analysis Framework

Certifications: Six Sigma, PMP

Score Profile

Detailed Score Breakdown

Review the individual components contributing to the overall match score. Each score is weighted to reflect its importance.

Skill Match — Match Score 1.00 — Weight: 0.25

How well the employee's skills and proficiency levels match the project's skill requirements.

Availability Match — Match Score 0.23 — Weight: 0.15

Considers if the employee is available for the project's duration and can meet the required effort within their weekly capacity.

Product Match — Match Score 1.00 — Weight: 0.05

Measures the match between products/tools required by the project and those experienced by the employee.

Industry Vertical Match — Match Score 1.00 — Weight: 0.05

Reflects whether the employee has experience in the project's customer industry.

Expertise Area Match — Match Score 0.50 — Weight: 0.10

Alignment of the employee's expertise areas with the project's integration requirements or specific expertise needs.

Language Proficiency Match — Match Score 0.83 — Weight: 0.05

Assesses if the employee meets the language proficiency levels required for the project.

Certification Match — Match Score 1.00 — Weight: 0.05

Checks if the employee possesses certifications preferred or required by the project.

Location & Flexibility Match — Match Score 0.25 — Weight: 0.00

Considers the match between project's work location/flexibility and the employee's preferences/location.

Cultural Awareness Rating — Match Score 0.90 — Weight: 0.02

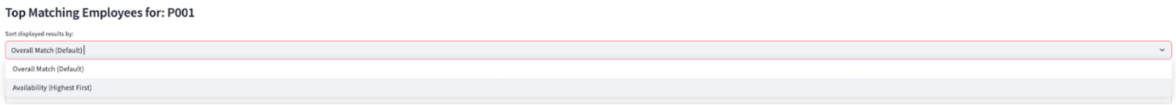
Problem Solving Rating — Match Score 0.90 — Weight: 0.04

Figure 17: Employee Scoring Detailed Scores Breakdown – HR Mate AI

- **Purpose:** Shows the numerical breakdown of all individual scores that contribute to the overall weighted match score.

- **Content:** For each scoring component (e.g., "Skill Match," "Availability Match," "Product Match"), it displays:
    - The display name of the score.
    - The calculated "Match Score" value (e.g., 1.00, 0.23).
    - The "Weight" assigned to that score component in the current configuration.
    - A brief "Explanation" of what the score represents (from SCORE_EXPLANATIONS).
    - This view uses st.metric for a clean presentation of scores and weights, often in a two-column layout for better readability. (Ref: **Figure 16 - list of scores like "Skill Match 1.00 Weight: 0.20", "Availability Match 0.23 Weight: 0.10" etc.**)
- **Tab 3: Raw Data**
    - **Purpose:** Provides access to the raw JSON data of the employee's calculated scores and details, primarily for debugging or advanced users who want to see the unprocessed scoring output.
    - **Content:** Displays the JSON object containing all Scores and Details for that specific employee-project match as generated by the scorer module.

6.2.6. Sorting Options for Results



Top Matching Employees for: P001

Sort displayed results by:

Overall Match (Default)

Figure 18: Employee Rank Sorting Option – HR Mate AI

Above the list of matched employees, a dropdown allows users to change the sort order of the displayed results.

- **Options:** "Overall Match (Default)," "Availability (Highest First)."
- **Functionality:** Selecting an option re-sorts the currently displayed list of candidates based on the chosen criterion, providing flexibility in how users prioritize and review matches.

This comprehensive UI design, with its clear information hierarchy, interactive controls, and detailed drill-downs, empowers users to effectively leverage the intelligence of the HR Mate AI system, making informed and efficient decisions about employee-project assignments.

CHAPTER 7: DEPLOYMENT AND ACCESSIBILITY: THE HR MATE AI DASHBOARD

Ensuring the HR Mate AI application is accessible and reliably operable for its intended users is a critical final step. This project employed a robust deployment strategy using Docker containerization [10] on a Virtual Machine (VM), providing a consistent and isolated environment.

Deployment Strategy: Docker Containerization on a Virtual Machine

The primary method chosen for deploying HR Mate AI involves packaging the entire application, including its Python environment and dependencies, into a Docker image. This image is then run as a container on a Virtual Machine (VM), offering several advantages:

- **Environment Consistency:** Docker ensures that the application runs in the exact same environment (OS, Python version, library versions) on the VM as it did during development, eliminating "it works on my machine" issues.
- **Dependency Management:** All dependencies are bundled within the Docker image, simplifying the setup process on the VM.
- **Isolation:** The application runs in an isolated container, preventing conflicts with other software or configurations on the VM.

[10] Dockers Containerization:
https://docs.docker.com/reference/

- **Portability:** The Docker image, once built, can be easily transferred and run on any system that supports Docker, including different cloud provider VMs or on-premise servers. For our purpose we utilized Digital Ocean [11]for running the VM.

Dockerization Process

The process of containerizing the HR Mate AI application involves the following key steps:

1. A Dockerfile is created in the root directory of the project. This text file contains a set of instructions that Docker uses to build the application image. A typical Dockerfile for this Streamlit application would include:
   - **Base Image:** Specifying a base Python image (e.g., FROM python:3.9-slim).
   - **Setting Working Directory:** Defining a working directory within the image (e.g., WORKDIR /app).
   - **Copying Requirements File:** Copying the requirements.txt file into the image (e.g., COPY requirements.txt .).
   - **Installing Dependencies:** Running pip install -r requirements.txt --no-cache-dir to install all necessary Python libraries.
   - **Copying Application Code:** Copying the entire project directory (all Python scripts, data files like generated_employees.json, generated_projects.json, media assets, assets for CSS, .streamlit configuration) into the image's working directory (e.g., COPY . .).

[11] Digital Ocean VM: https://docs.digitalocean.com/products/droplets/

- **Exposing Port:** Specifying the port on which the Streamlit application will run (Streamlit defaults to 8501, e.g., EXPOSE 8501).
- **Defining Entrypoint/Command:** Specifying the command to run when the container starts (e.g., CMD ["streamlit", "run", "streamlit_app.py", "--server.port=8501", "--server.address=0.0.0.0"]). The --server.address=0.0.0.0 makes the Streamlit app accessible from outside the container.

2. Once the Dockerfile is prepared, the Docker image is built using the command: docker build -t hr-mate-ai:latest

3. This command, run in the project's root directory, creates an image tagged as hr-mate-ai with the version latest.

Image Transfer and Deployment on the Virtual Machine

1. **Saving the Docker Image:** After a successful build, the Docker image is saved as a .tar archive for easy transfer:

   docker save hr-mate-ai:latest > hr-mate-ai-latest.tar

2. **Transferring to VM:** The hr-mate-ai-latest.tar file is then transferred to the target Virtual Machine using a secure method like scp (Secure Copy Protocol) or any file transfer tool suitable for the VM environment.

3. **Loading the Image on VM:** On the VM (which must have Docker installed), the image is loaded from the .tar file:

   docker load < hr-mate-ai-latest.tar

Running the Docker Container - The application is then run as a Docker container using a command similar to:

```
docker        run        -d        -p        80:8501        \
     -v  $(pwd)/chroma_db_vm:/app/chroma_db  \
   -e GEMINI_API_KEY="your_actual_api_key" \
   --name hr-mate-container \ hr-mate-ai:latest
```

- -d: Runs the container in detached mode (in the background).

- -p 80:8501: Maps port 80 on the VM's host to port 8501 inside the container (where Streamlit is running). This allows users to access the application via the VM's IP address or domain name on the standard HTTP port.

- -v $(pwd)/chroma_db_vm:/app/chroma_db: This is a crucial volume mount. It maps a directory named chroma_db_vm (created in the current directory on the VM, e.g., where the docker run command is executed) to the /app/chroma_db directory inside the container. The CHROMA_DB_PATH within the application's config.py should be set to /app/chroma_db. This ensures that the ChromaDB data is persisted on the VM's filesystem, outside the container, and survives container restarts or updates.

- -e GEMINI_API_KEY="your_actual_api_key": Passes the Google Gemini API key as an environment variable to the container. This is a secure way to provide secrets to the application running inside the container. Alternatively, Docker secrets or other environment management tools available on the VM could be used.

- --name hr-mate-container: Assigns a recognizable name to the running container for easier management.
- hr-mate-ai:latest: Specifies the image to run.

Accessing and Using the Deployed Application

Dockerized VM Deployment: Once the Docker container is running on the VM and port mapping is correctly configured (e.g., port 80 on VM to port 8501 in container), users can access the HR Mate AI application by navigating to the VM's public IP address or its assigned domain name in their web browser (e.g. http://<vm_ip_address>) for e.g. http://142.93.238.167:8501/ in our case.

CHAPTER 8: CONCLUSION

The HR Mate AI Capstone Project successfully conceptualized, developed, and demonstrated a robust proof-of-concept for an intelligent employee-project matching system. The project has laid a strong foundation for transforming a traditionally manual and often subjective HR process into a more efficient, data-driven, and strategic function. This concluding chapter summarizes the project's achievements against its initial objectives, highlights its contributions and value, and outlines a comprehensive vision for future enhancements.

Recapitulation of Project Achievements Against Objectives

The HR Mate AI project met its core objectives by:

1. **Streamlining Data Collection:** Detailed schemas for employees and projects data were established in consultation with the client. This framework can serve as a baseline for data collection and implementation of our matching algorithm and other future data related projects

2. **Comprehensive Data Modeling and Generation:** A sophisticated synthetic data generation pipeline was implemented in collaboration with Canon EMEA. The data generation pipeline, leveraging thematic controls and Google's Gemini API for text enrichment, produced realistic and diverse datasets crucial for development and testing.

3. **Advanced Semantic Candidate Retrieval:** An effective retrieval engine was built using the BAAI/bge-large-en-v1.5 embedding model and Chroma DB. This allows the system to identify potentially suitable candidates based on a deep understanding of textual similarity, moving beyond simple keyword matching.

4. **Configurable and Transparent Multi-Criteria Scoring:** A flexible scoring engine was developed, enabling the evaluation of candidates across numerous weighted attributes. The ability for users to adjust these weights via the UI provides adaptability to varying project priorities and fosters transparency in the matching process.

5. **Intuitive User Interface:** A Streamlit-based web application was created, offering users an easy-to-navigate dashboard for project selection, candidate review, and in-depth analysis of match scores and attribute comparisons, supported by visual aids like radar charts.

6. **Successful Proof-of-Concept Demonstration:** The end-to-end project execution show casing all inputs, processes and outputs for employee ranking effectively demonstrates the viability and potential of our algorithm for employee-project matching.

Future Innovations

The current HR AI Mate serves as a solid foundation as proof of concept. There are some future innovations that can be introduced to further enhance the system.

1. **Integrate Into Current System:** Develop APIs or connectors to integrate HR Mate AI with existing Human Resource Management Systems (HRMS)

2. **User Feedback Mechanisms and Model Retraining:** The model performance can be logged and analyzed with additional feedback on match quality from users. This can be used to adjust weights to improve model performance. Moreover, with the collection of labels for match quality, this system can be transitioned into a ML model which improves over time as data is collected.

3. **More Sophisticated Pools:** The generic pools and the thematic pools that are being utilized for data generation can be expanded. These can be extended to include other themes, departments and values for added sophistication.

4. **Enhancing Matching Criteria:** Relationships between different feature values can be defined that are used in score calculations. Standardizing ontologies would result in the algorithm being more flexible to human inputs. For example, establishing relationship between “Data Science” and “Python”.

5. **Extending to Other Use Cases:** The framework developed can be extended to not only match employees with current projects but also to suggest projects or training that align with their stated career goals. We can also enhance it further to assist in forming optimal project teams, considering not just individual skills but also team dynamics, complementary skill sets, and diversity.

Final Concluding Remarks

The HR Mate AI project demonstrates the significant potential of leveraging artificial intelligence to create a more efficient, accurate, and strategic employee-project matching system. While the current implementation serves as a robust proof-of-concept, the outlined future directions highlight a clear path towards evolving it into an indispensable enterprise-grade tool. HR Mate AI can empower organizations like Canon EMEA to unlock the full potential of their human capital, fostering both project success and employee growth in an increasingly dynamic business environment. This Capstone Project not only addresses a tangible business problem but also contributes to the growing understanding of how AI can positively transform core HR functions.

APPENDIX

```
employee_roles_templates = [
    {"Role Name": "Solution Architect", "Role Description": "Designs high-
level technical solutions for enterprise customers.", "Theme": "Technical"},
    {"Role Name": "Sales Account Manager", "Role Description": "Manages
customer accounts and drives sales processes.", "Theme": "Sales"},
    {"Role Name": "Digital Marketing Specialist", "Role Description":
"Executes online campaigns, SEO, and branding strategies.", "Theme":
"Marketing"},
    {"Role Name": "Senior HR Manager", "Role Description": "Manages HR
operations and employee relations.", "Theme": "HR"},
    {"Role Name": "Legal Counsel", "Role Description": "Provides legal support
for contracts and compliance.", "Theme": "Legal"},
    {"Role Name": "IT Systems Engineer", "Role Description": "Maintains and
optimizes internal IT infrastructure.", "Theme": "Technical"},
    {"Role Name": "Workflow Consultant", "Role Description": "Analyzes
business processes and recommends workflow improvements.", "Theme":
"Consulting"},
    {"Role Name": "Project Manager", "Role Description": "Oversees project
delivery and coordinates cross-functional teams.", "Theme": "Consulting"},
    {"Role Name": "Data Analyst", "Role Description": "Analyzes data and
delivers business insights.", "Theme": "Technical"},
    {"Role Name": "Customer Success Manager", "Role Description": "Supports
post-sales success and client satisfaction.", "Theme": "Sales"},
    {"Role Name": "Field Support Engineer", "Role Description": "Provides
onsite technical support for Canon products.", "Theme": "Technical"},
    {"Role Name": "Pre-Sales Engineer", "Role Description": "Prepares
technical demos and solution proposals for prospects.", "Theme": "Sales"},
    {"Role Name": "Compliance Manager", "Role Description": "Ensures adherence
to regulations and company standards.", "Theme": "Legal"},
    {"Role Name": "HR Business Partner", "Role Description": "Collaborates
with leadership to align HR strategy.", "Theme": "HR"},
    {"Role Name": "Corporate Trainer", "Role Description": "Designs and
delivers employee training programs.", "Theme": "HR"},
    {"Role Name": "Technical Support Specialist", "Role Description":
"Resolves technical issues reported by customers.", "Theme": "Technical"},
    {"Role Name": "Content Creator", "Role Description": "Develops written,
video, and visual content for marketing.", "Theme": "Marketing"},
    {"Role Name": "Solutions Support Consultant", "Role Description":
"Provides technical guidance and second-line support for Canon solutions
during and after customer deployment.", "Theme": "Technical"},
    {"Role Name": "Integration Developer", "Role Description": "Develops
integrations between Canon products and third-party systems.", "Theme":
"Technical"},
    {"Role Name": "Strategy Consultant", "Role Description": "Advises leadership on business growth and optimization strategies.", "Theme": "Consulting"},
]


project_summary_templates = [
    {"Project Summary": "Implement scalable workflow automation system", "Scope and Deliverables": "Deploy Workflow2000, integrate with client systems", "Theme": "Technical"},
    {"Project Summary": "CRM integration for loyalty program", "Scope and Deliverables": "Customize CRM modules and train sales team", "Theme": "Sales"},
    {"Project Summary": "Launch digital marketing portal", "Scope and Deliverables": "Create website, SEO, lead funnels", "Theme": "Marketing"},
    {"Project Summary": "HR digital onboarding system", "Scope and Deliverables": "Implement HRIS system, self-service portals", "Theme": "HR"},
    {"Project Summary": "Contract management system deployment", "Scope and Deliverables": "Deploy document archiving and e-signature workflows", "Theme": "Legal"},
    {"Project Summary": "Upgrade internal IT infrastructure", "Scope and Deliverables": "Replace old servers, migrate systems to cloud", "Theme": "Technical"},
    {"Project Summary": "Business workflow audit", "Scope and Deliverables": "Map processes and suggest automation improvements", "Theme": "Consulting"},
    {"Project Summary": "Manage ERP migration project", "Scope and Deliverables": "Deliver milestones for new ERP roll-out", "Theme": "Consulting"},
    {"Project Summary": "Data warehouse design", "Scope and Deliverables": "Create new analytics-ready database", "Theme": "Technical"},
    {"Project Summary": "Post-sale onboarding program", "Scope and Deliverables": "Develop client onboarding workflow", "Theme": "Sales"},
    {"Project Summary": "Onsite print solutions setup", "Scope and Deliverables": "Install Print2.0 platform for retail client", "Theme": "Technical"},
    {"Project Summary": "Pre-sales technical proof-of-concept setup", "Scope and Deliverables": "Build demo environments for prospects", "Theme": "Sales"},
    {"Project Summary": "Regulatory compliance documentation project", "Scope and Deliverables": "Standardize processes, deliver compliance documentation", "Theme": "Legal"},
    {"Project Summary": "Organizational culture development initiative", "Scope and Deliverables": "Conduct workshops, employee surveys", "Theme": "HR"},
    {"Project Summary": "Employee training", "Scope and Deliverables": "Give overview on Learning Management System (LMS)", "Theme": "HR"},
    {"Project Summary": "Customer remote support setup", "Scope and Deliverables": "Setup online ticketing and remote assistance systems", "Theme": "Technical"},
    {"Project Summary": "Content library migration", "Scope and Deliverables":
"Migrate marketing content to new CMS", "Theme": "Marketing"},
    {"Project Summary": "Quality assurance framework rollout", "Scope and
Deliverables": "Implement QA policies across departments", "Theme":
"Technical"},
    {"Project Summary": "API and system integration project", "Scope and
Deliverables": "Develop middleware for integration of ERP/CRM", "Theme":
"Technical"},
    {"Project Summary": "Business strategy development program", "Scope and
Deliverables": "Assist C-suite with market expansion strategy", "Theme":
"Consulting"},
]

# Controlled Product Pools per Theme
product_pools = {
    "Technical": ["Workflow2000", "Print2.0", "AIScan", "CloudSuite",
"IntegrationHub"],
    "Sales": ["CRM Pro", "Sales Enablement Suite", "Loyalty CRM", "SalesForce
Light"],
    "Marketing": ["Digital Campaign Manager", "SEO Toolkit", "Content CMS",
"Social Media Manager"],
    "HR": ["HRIS Plus", "Onboarding Suite", "Employee Experience Platform"],
    "Legal": ["Compliance Suite", "Contract Manager Pro", "Regulatory
Tracker"],
    "Consulting": ["ERP Migration Tool", "Business Analysis Framework",
"Strategy Kit"]
}

# Controlled Skill Pools per Theme
skill_pools = {
    "Technical": ["Data Analysis", "Workflow Automation", "Cloud Services",
"IT Infrastructure", "API Development"],
    "Sales": ["CRM Integration", "Negotiation", "Client Management", "Customer
Relationship Management"],
    "Marketing": ["SEO Optimization", "Content Strategy", "Campaign
Management", "Copywriting", "Branding"],
    "HR": ["Digital HR", "Organizational Development", "Talent Management",
"Communication Skills"],
    "Legal": ["Contract Management", "Regulatory Knowledge", "Document
Review", "Compliance Documentation"],
    "Consulting": ["Business Analysis", "Strategic Planning", "Workflow
Optimization", "Project Management", "Change Management"]
}

# Controlled Certifications Pool per Theme
certification_pools = {
    "Technical": ["ITIL", "ISO 27001", "Microsoft Azure Certification"],
    "Sales": ["Certified Sales Professional (CSP)", "CRM Specialist
Certification"],
    "Marketing": ["Digital Marketing Certification", "Google Ads 
Certification", "HubSpot Marketing Certification"],
    "HR": ["PMP", "SHRM-CP", "HR Analytics Certification"],
    "Legal": ["Certified Compliance Officer", "GDPR Certification", "Contract 
Law Certification"],
    "Consulting": ["PMP", "Six Sigma", "Agile Practitioner", "Business 
Analysis Certification"]
}

# Controlled Expertise Areas Pool per Theme
expertise_pools = {
    "Technical": ["Scripting", "API Integration", "Cloud Infrastructure", 
"Networking", "Cybersecurity"],
    "Sales": ["CRM Integration", "Sales Pipeline Automation", "Client 
Relationship Systems"],
    "Marketing": ["SEO Optimization", "Content Management Systems", "Social 
Media Integration"],
    "HR": ["HRIS Systems", "Employee Experience Platforms", "Organizational 
Development Systems"],
    "Legal": ["Document Archiving", "Contract Management Systems", "Regulatory 
Compliance Tools"],
    "Consulting": ["Strategic Planning", "Business Workflow Optimization", 
"ERP Systems Integration"]
}

# Generic Pools (Predefined vocabularies)
locations_master = ["Berlin", "Vienna", "London"]
work_flexibility_options = ["onsite", "remote", "hybrid"]
languages_master = ["English", "French", "German", "Italian"]
fluency_levels = ["A1", "A2", "B1", "B2", "C1", "C2"]
industries_master = ["Healthcare", "Education", "Finance", "Manufacturing", 
"Retail"]
```

BIBLIOGRAPHY

Sources Referenced

Raghunandanan, Arjun. 2025. “How to Generate Synthetic Business Dummy Data with Gemini: using Gemini Library” https://arjunraghunandanan.medium.com/generate-sythetic-data-with-google-gemini-07e2179b448e

Dhungana, Kamal. 2023. “An Overview of ChromaDB: The Vector Database” https://medium.com/@kbdhunga/an-overview-of-chromadb-the-vector-database-206437541bdd

Nidhiworah. 2024. “ChromaDB – Introduction”. https://medium.com/@nidhiworah02/chroma-db-introduction-25718915bae6

Belagatti, Pavan. 2025. “Semantic Search: What is it + How Does it Work?” https://www.singlestore.com/blog/a-complete-guide-to-semantic-search-for-beginners/

Espejel, Omar. 2022. “Getting Started with Embeddings” https://huggingface.co/blog/getting-started-with-embeddings

Karabiber, Faith. “Cosine Similarity” Accessed June XX, 2025 https://www.learndatasci.com/glossary/cosine-similarity/

Sundriyal, Harshir. 2024. “Rapid Development of Streamlit App in Cloud (Virtual Machine) https://medium.com/@harshitsundriyal/rapid-deployment-of-streamlit-app-in-cloud-virtual-machine-7193f629be58

Name of Company/website. “Deploy Streamlit using Docker” Accessed June XX, 2025

https://docs.streamlit.io/deploy/tutorials/docker

AI DISCLAIMER

This project may incorporate content—such as ideas, analyses, text, code, or supporting material—generated or assisted by large language models (LLMs) and other artificial intelligence (AI) tools. These technologies were used to support research, drafting, coding, or source gathering processes. All AI-assisted content was critically reviewed and validated by the author(s). Responsibility for the methodology, analysis, and conclusions presented herein lies solely with the author(s).