1. Remove duplicate skills from two given lists: resume skills and job description skills.

2. Convert all skills in a list to lowercase and remove extra spaces.

3. Store resume skills and job description skills in a single structured dictionary.

4. Load a pretrained Sentence-BERT model using Python.

5. Generate an embedding for a single skill string and print its vector dimension.

6. Generate embeddings for a list of resume skills.

7. Generate embeddings for a list of job description skills.

8. Compute cosine similarity between two skill embeddings.

9. Compare one resume skill against all job description skills and print similarity scores.

10. Create a similarity matrix for all resume skills versus all job description skills.

11. Store the similarity matrix in a Pandas DataFrame with proper row and column labels.

12. For each job description skill, find the resume skill with the highest similarity score.

13. Define similarity thresholds and classify skills as matched, partially matched, or missing.

14. Generate a structured skill gap report containing matched, partial, and missing skills.

15. Save the skill gap report in JSON format.

16. Visualize the similarity matrix using a heatmap.

17. Add axis labels and a color legend to the heatmap.

18. Highlight the highest similarity score in each column of the similarity matrix.

19. Handle cases where resume skills or job description skills are empty.

20. Normalize abbreviations such as ML, DL, and AI before generating embeddings.

21. Compare similarity results using two different Sentence-BERT models.

22. Cache embeddings so repeated skills are not embedded multiple times.

23. Build a pipeline that takes raw resume text and job description text and outputs a skill gap report.

24. Return the top three closest resume skills for each job description skill.

25. Apply different similarity thresholds for technical skills and soft skills.

26. Compute an overall resume and job description alignment score.

27. Export the skill gap report and similarity heatmap into a single file.

28. Design a modular architecture separating embedding generation, similarity computation, and reporting.