HAAI++ Capstone Project Instructions

1. Presentation is not required.
2. Projects are individual projects.
3. Marks Distribution: Report (30) + Programming (70)
4. Partial marks will be given if a participant is unable to complete the code provided the report is in sync with the code.
5. Although we suggest proper modularity in the code as per general coding standard but marks will be given based on the approach and algorithms.
6. All the projects must use at least two LLMs. Choice of the models must be properly justified.
7. You must use the shared directory structure to submit the project.
    a. Keep report in the Report directory in '.pdf' format.
    b. Keep the code/data/output in Codebase directory.
    c. Replace the ID in the folder name 'Capstone_Project-HPPCS[**ID**]' with the project ID. Please find your corresponding project ID below (Point 9).
8. Specific instructions related to Report:
    a. Report template in MS word (*docx*) format is shared.
    b. You can change the Titles and Sub-Titles as per your requirement.
    c. Report should not exceed 3 Pages considering minimum text font size 12.
    d. Use only text in the report and avoid using images.
    e. Save the final report as PDF in the same directory.
9. Specific instructions related to Programming:
    a. Should have *main.py* that starts the code (code without *main.py* will not be evaluated).
    b. Upload execution syntax in a text file (*execution.txt* example is shared).
    c. You must provide proper comments in the code. We advise you on providing comments on each and every function and wherever possible in the code to allow for proper evaluation and avoid confusions.
    d. Do not upload additional files other than the required python/input-output/dataset files.
    e. No sub-directories are allowed. Keep everything in the same directory so that all the files and data can be accessed using ("./").
    f. You can use single or multiple python files as per your requirement but all the files should be in the same directory.
    g. You can use API/local models that are paid/free/open-source as per your choice.
    h. If using any API key, then it should be provided as a parameter to *main.py*. You are not required to share the API key.
    i. The project objective and guidelines are for helping you but you are not required to be strictly followed to select the models/packages. You can use any alternate model/package as per availability.
    j. You are free to use any approach as long as it is logically correct and solves the problem of your objective.
    k. You can use UI/command-line interface for your project as per your choice.
    l. You can execute your code in Terminal or Notebooks - Kaggle/Collab etc. but the final code should be in '*.py*' only. FYI, you can easily run '*.py*' file in Collab or Kaggle, so local system limitation should not be an issue.
    m. You have to keep the data and program files in the directory provided. Then finally zip and upload the entire directory. Uploading in GitHub/Google Drive is not accepted.

10. Specific instructions related to project objectives:
    a. CV Creation using LLMs **(HPPCS[01])**
        i. Input: Keep 10 sample unstructured user profile data
        ii. Output: Keep 10 final CVs from the input files

b. CV Sorting using LLMs **(HPPCS[02])**

   i. Input: Keep 10 CVs and 1 Job-description.

   ii. Output: Ranked list of CVs in an output file.

c. Advertisement Creator - Using Image Generation **(HPPCS[03])**

   i. Input: Keep 3 Advertisement Text and 1 Image Advertisement Template.

   ii. Output: 3 Final Advertisements.

d. Multimodal Medical Assistant **(HPPCS[04])**

   i. Input: 5 Medical Images corresponding to 5 Text Prescriptions/Patient Details.

   ii. Output: Conversation in a JSON output file for each of the inputs (total 5 output files).

e. Sports Commentator from Video **(HPPCS[05])**

   i. Input: Training Videos with commentary and 1 Testing Video without commentary.

   ii. Output: Commentary in a JSON format that will have commentary text with time-line information from the test video.

f. Text Puzzle Game Development using LLMs **(HPPCS[06])**

   i. Input: As per requirement.

   ii. Output: Output the entire gameplay and instructions in a text format (for terminal) and screenshot (for UI based).