PSG COLLEGE OF TECHNOLOGY

DEPARTMENT OF APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCES

HACKATHON PROBLEM STATEMENTS

1. Design and implement a chatbot system capable of ingesting and interpreting uploaded documents (e.g., PDFs) to provide accurate, fact-based responses quickly and reliably. The chatbot should utilize LLM APIs and other retrieval techniques.

Deliverables:

1. Responsive REST APIs connected with Simple UI

Good to have:

a. Support bulk upload and processing
b. Security by design
c. Improved User experience using UI and streaming APIs
d. Employ effective techniques (e.g., prompt engineering, context-verification, or grounding) to prevent “hallucinations” by verifying that all responses directly reference the source material.

2. Build a Hackathon Management platform for end-to-end event handling

Deliverables:

1. Allow participants to register with details

2. Provide a list of hackathon themes or categories.

3. Teams can upload their projects (e.g., code repositories(github links), documents, and presentations), Store submissions securely with timestamp.

4.Build AI/ML techniques to help judges with insights from the submissions like:

a. Feedback Suggestions:Auto-generate constructive feedback for submissions to save judges time.
b. Scoring Assistance: LLMs can pre-score submissions by:
   i. Summarizing key project features.
   ii. Analyzing problem-statement adherence.
   iii. Suggesting scores for innovation, feasibility, and impact based on predefined prompts.

Additional Features:

1. Skill Matching: Use LLMs to analyze participant profiles and suggest ideal teams based on complementary skills or experience.
2. Code Explanation: Analyze uploaded code to generate concise summaries for judges (e.g., purpose, key functions, and dependencies).

3. Build a secure platform(Personal Data Vault) to store and manage sensitive personal data, ensuring privacy and accessibility.

Deliverables

1. Implement secure APIs with Flask/Node.js for data storage and encryption.
2. Develop a simple UI for managing and viewing stored data.
3. Use AES or RSA for encryption and implement authentication.
4. Allow users to upload and categorize sensitive files (e.g., financial documents, passwords).
5. Provide access control for sharing files.

Additional Features:

1. Add a data breach detection module to alert users of compromised accounts.
2. Integrate password management features.

Key Points to Note for the Hackathon

1. **Code Submission**:
   - Implement any one of the above problem statements and upload your code to a **GitHub** repo with public access.
   - Include a **README file** with clear instructions and details about your project.
2. **Demo and Abstract**:
   - Submit a **video** demo along with a concise **abstract** of your solution.
   - Deadline for submission: **Before 9:00 AM on Thursday, 26th December.**
3. **Presentation**:
   - Prepare a **presentation** of the solution.
4. **Further Details**:
   - Additional information and updates will be shared as needed.

All the best to all participants!