CodeMate.ai

CodeMate Hackathon Problem Statements and Submission Guidelines

Problem Statements

Participants must select **one** of the following problem statements to build their project. All projects must be developed using **Python** wherever specified. The use of **CodeMate Build** and **CodeMate Extension** is mandatory throughout the development process to gauge the understanding of AI tools and efficiency.

Problem Statement 1: Python-Based Command Terminal

Description

Develop a fully functioning command terminal that mimics the behavior of a real system terminal. The backend of this terminal must be built in Python. The terminal should be able to execute standard commands (such as file operations, directory navigation, process checks, etc.) and return accurate outputs.

Challenge

The key challenge is replicating the low-level behavior of a traditional terminal in a Python environment while ensuring efficiency, correctness, and extensibility.

Mandatory Requirements

- A Python backend to process and execute commands
- Support for full-fledged file and directory operations (e.g., ls, cd, pwd, mkdir, rm)
- Error handling for invalid commands
- Clean and responsive interface (can be CLI or web-based)
- Integration with system monitoring tools such as CPU, memory, and processes

Optional Enhancements

- AI-driven terminal where users can type natural language queries (e.g., *"create a new folder called test and move file1.txt into it"*) and the system interprets them into actual commands
- Command history and auto-completion

Problem Statement 2: PR (Pull Request) Review Agent

Description

Build an AI-powered agent capable of reviewing pull requests across any git server (e.g., GitHub, GitLab, Bitbucket). The agent should analyze code changes and provide constructive feedback for improvements.

Challenge

Creating a general-purpose agent that works with any git server and can understand diverse codebases. The system must evaluate changes for quality, standards, and possible issues.

Mandatory Requirements

- Compatibility with multiple git servers (not restricted to GitHub only)
- Feedback generation on code structure, standards, and possible bugs
- Written in Python with a modular structure

Possible Enhancements

- AI-driven feedback with automated suggestions for better readability, performance, or security
- Inline review comments similar to GitHub or GitLab review systems
- Integration with CI/CD pipelines for automated pre-merge reviews
- Scoring system to grade PRs on code quality

Problem Statement 3: Deep Researcher Agent

Description

Create a deep researcher agent that can search, analyze, and synthesize information from large-scale data sources. The solution must not rely on external web search APIs; instead, it should handle local embedding generation and reasoning.

Challenge

Building a high-scale system that can effectively gather, process, and retrieve relevant information without depending on external APIs. The embeddings and retrieval logic must be handled locally.

Mandatory Requirements

- Python-based system for query handling and response generation
- Local embedding generation for document indexing and retrieval
- Support for multi-step reasoning to break down queries into smaller tasks
- Efficient storage and retrieval pipeline

Possible Enhancements

- Summarization of multiple sources into a coherent research report
- Interactive query refinement where the user can ask follow-up questions to dig deeper
- AI-powered assistant that explains reasoning steps
- Export of research results in structured formats such as PDF or Markdown

CodeMate®.ai

Submission Guidelines

All participants must complete two mandatory modes of submission. Both are compulsory and will have separate evaluation criteria.

Submission on CodeMate for Education

1. Create an account on **CodeMate for Education** via the platform link: http://edu.codemate.ai/signup?org_code=e6ccbc16-5e38-4042-b3cf-4a558b408ffc&class_code=B4hiap
2. In case the above link doesn't work:
    - Head to https://edu.codemate.ai/
    - Create your account using your college official email id's.
    - Under the heading of **Public Courses – New**
    - Head to the Join a private classroom with a code
    - Enter the following code in the text box and you should be logged in: B4hiap
3. Join the assigned classroom and access your chosen project assignment
4. Upload the following deliverables:
    - Source code on CodeMate IDE (CodeMate for education, assignment submission)
    - While uploading your files ensure to **remove node modules, any venv folder, or other system level files**, they may negatively impact your evaluation.
    - A live working video demonstrating the project
    - A live hosted URL of the project
    - A GitHub repository link containing the complete source code
    - Watch the video for clear steps on uploading your project.

Submission on LinkedIn

1. Prepare a LinkedIn post with the following:
    - The working video of your project, with a short verbal or written explanation
    - An explanation of your project's purpose and features
2. Follow CodeMate's LinkedIn account
3. Tag the official CodeMate LinkedIn account and SRM University in your post
4. Add the tag of **#SRMHacksWithCodemate**

Important Notes

- Use of CodeMate Build and CodeMate Extension is mandatory
- Both submission modes, CodeMate for Education and LinkedIn, must be completed
- Failure to complete either submission may result in non-evaluation.
- Enhancements are not compulsory but may improve evaluation scores, and act as deciding factors in case of similar scores.