TECHFEST 2023-24

HackAI

[Overview of fetch.ai]

ROUND 1 PROBLEM STATEMENT :

All challenges must utilize Fetch.ai's uAgents library to perform the core functionalities.
The participants can choose to work on and submit **ANY ONE** of the following problem statements:

1. Temperature Alert Agent

Your challenge is to create the Temperature Alert Agent using uAgent library, a tool that:

a) Connects to a free weather API to fetch real-time temperatures for the specified location.
b) Lets users set their preferred temperature range (e.g., a minimum and maximum temperature) and location.
c) Sends an alert/notification to the user when the current temperature in their chosen location goes below the minimum or above the maximum threshold they've set.

2. Currency Exchange Monitor & Alert Agent

Your challenge is to create the Currency Exchange Monitor and Alert Agent using uAgent library, a tool that:

a) Allows users to select their base currency and one or more foreign currencies they wish to monitor.
b) Connects to a currency exchange API to fetch real-time exchange rates.
c) Lets users set thresholds for alerts (e.g., notify me if 1 USD becomes more than 82.60 INR or less than 82.55 EUR).
d) Sends an alert/notification to the user when the exchange rate crosses the thresholds they've set.

RESOURCES:

Github Repository
YouTube Tutorials
uAgents Documentation
uAgents Examples

Directory Structure:

Organise your project files using the following directory structure:

```
Python
├── poetry.lock
├── pyproject.toml
├── README.md
└── src
    ├── agents
    │   ├── module_x
    │   │   ├── model_x.py
    │   │   └── __init__.py
    │   └── __init__.py
    ├── __init__.py
    ├── main.py
    ├── messages
    │   ├── message_x.py
    │   ├── __init__.py
    └── utils
        └── utility_x.py
```

Example:

```
Python
├── poetry.lock
├── pyproject.toml
├── README.md
└── src
    ├── agents
    │   ├── flights
    │   │   ├── flights.py
    │   │   └── __init__.py
    │   └── __init__.py
    ├── __init__.py
    ├── main.py
    ├── messages
    │   ├── flight.py
    │   ├── __init__.py
    └── utils
        └── llm.py
```

For reference, you can look here.

Semantic Commits

Use semantic commit messages to make it easier to understand the changes you've made.

Code Style:

Follow the coding style guidelines provided by the programming language you're using. Make sure to include comments that clarify complex parts of your code. Coding style is an important factor in making your code accessible and readable.

- Use a consistent indentation style. For example, if you're using 4 spaces for Python code, stick with it throughout.
- Include meaningful variable and function names.
  Example: calculate_area() is more understandable than func1()

README file
Include a README.md file with the following:
- Project name
- Description of the project
- Instructions to run the project
- Special considerations, if any

Plagiarism Policy
Any code copied from external sources must be properly attributed in comments. Plagiarized projects will be disqualified.

Judging Criteria
Your project will be judged on the following:
- Functionality: Is your project functional end-to-end?
- Creativity: Does your project introduce a new idea or an innovative solution?
- Code Quality: Is your code clean, organized, and well-commented?
- Relevance: Does your project align with the hackathon theme?

API Keys and Environment Variables
If your project involves the use of API keys or other sensitive data:
- Do not push sensitive keys or passwords to the public repository.
- Instead, include a sample .env file with placeholders for these values and detailed instructions on where to get the actual values and how to set them up.

SUBMISSION GUIDELINES
To ensure a smooth and standardized submission process for all hackers, please adhere to the following guidelines.

Public Repository:
- Ensure that your project is stored in a public GitHub repository. This will make it accessible for the judging panel to review your code.
- The master or main branch should be the most up-to-date version of your project.
- Please refrain from pushing any code after the deadline as it may lead to disqualification.

SUBMISSION DETAILS
- All the files must be stored in a single folder & this folder must be sent via email to hackai@techfest.org
- Subject of the mail: “HackAI_TeamID”
- Filename of the GitHub repository: “HackAI_TeamID”
- Mention your team details clearly in the mail.

ROUND 2: FINAL ROUND

The Problem Statement of the final round will be shared with the winners of Round 1.
The selected teams will qualify for a hackathon at IIT Bombay during Techfest 2023-24.

COMPETITION TIMELINE

Last Date of Registration	9th October
Round 1 Submission Deadline	10th October
First Round Results Announcement	16th October
Final Round at IIT Bombay	27th December

TEAM SPECIFICATIONS

- A team may consist of a maximum of 4 members.
- Individual Participation is allowed. However, the participant must mandatorily get a Team ID. This Team ID will be used for all further communication
- Students from different educational institutes can form a team.

ELIGIBILITY

All students with a valid identity card from their respective educational institutes are eligible to participate.

PRIZE

The prize money will be awarded to the top 3 winners via NEFT and will be processed within 30 working days after receiving the prize money from sponsors. The top 10 participants will get a certificate of excellence, and the top 60% of participants will get a certificate of participation. Winners have to mail the following information (immediately after the announcement of results) to akshat@techfest.org

Format of Mail

Subject: Competition, Team Id, Position (example- “HackAI, HAI-211003, 1st Position”)

Body of mail

The body of the mail should contain the relevant bank account details of the team leader. The exact details required will be conveyed to the winners as soon as the res