Hackathon

Python – Full stack

1. **Hackathon Structure & Sprints**
2. **Technical Guidelines**
3. **General Guidelines & Expectations**
4. **Communication & Support**

1. Hackathon Structure & Sprints

- a one-day hackathon
- three fast-paced sprints

Sprint 0: Planning & Setup (First 1-1.5 hours)

- **Goal:** Ideation, setup, and strategy.
- **Activities:**
  - Understanding the problem statement.
  - Brainstorming & finalizing the solution approach.
  - Setting up repositories, environment, and cloud services.
  - Listing tasks (backend, frontend, etc.).
- **Deliverable:** A clear roadmap and approach. Everything on Readme file in Github repo

Sprint 1: Core Development (Next 3-4 hours)

- **Goal:** Implement core functionality.
- **Activities:**
  - Backend: API development / integration (CRUD operations, authentication, (optional) database connections).
  - (optional) Frontend: UI development with React, integrating APIs.
  - Database: Setting up DB and testing data transactions.
  - Midpoint check-in with Chapter Members for quick feedback.
- **Deliverable:** A working prototype with core functionalities integrated.

Sprint 2: Refinement, Testing, Submission & Demo (Final 1 hour)

- **Goal:** Complete the project, test, and deploy.
- **Activities:**
  - API refinement, debugging, and demo.
  - Good to have: Deployment on the cloud (AWS, Vercel, or another service).
  - Preparing the final submission (GitHub repo, demo link, and pitch).
- **Deliverable:** A **fully functional MVP (Minimum Viable Product)** with clear documentation.

2. Technical Guidelines

- **Tech Stack:**
    - **(Optional) Frontend:** ReactJS or Angular
    - **Backend:** Python with Django or FastApi
    - **Database:** MongoDB (Atlas setup, Mongoose ORM), DynamoDB, Postgres, MySQL etc
    - **(Optional) Cloud:** Deployment (AWS, Vercel, or another service).
    - **Version Control:** GitHub (branching).
- **Required Tools:**
    - Code editor (VS Code or PyCharm or Sublime Text or Atom)
    - Python (3.7 or higher)
    - (Optional) ReactJS or Angular with NodeJS framework
    - Virtual Environment Tool - venv, virtualenv, conda
    - FastApi or Django framework
    - Postman/Thunder Client / Swagger - For testing APIs.
    - Git: For version control
    - Public GitHub Account - All participants are expected create separate repos for Hackathon
    - (optional) Docker: For containerizing your FastAPI / Django application.
    - Database: Either on local laptop or online database with prior setup - Postgres/MySQL/Oracle/MongoDB/DynamoDB
    - (Optional) MongoDB Atlas for database management.
    -

3. General Guidelines & Expectations

- **Team Collaboration:** Teams should work in parallel (backend and frontend).
- **Code Quality:** Keep it clean, modular, and well-documented.
- **Submission Requirements:**
  - GitHub repository with a README.
  - Live deployment link (optional).
  - A short pitch explaining the solution.
- **Judging Criteria:**
  - **Technical Execution:** Proper implementation of the stack.
  - **Innovation & Impact:** How unique and practical is the solution?
  - **User Experience:** UI/UX quality and usability.
  - **Functionality:** How well does the project work?

4. Communication & Support

- **Chapter Support:** Available throughout for quick problem-solving.

5. Key Reminders

- Time is limited—prioritize essential features first.
- Collaborate efficiently and leverage team strengths.
- Test your work early to avoid last-minute failures.
- Stay agile—adapt quickly to roadblocks.
- Most importantly, **enjoy the process!**

Q&A