Proposed Name
NodeScape
Byte the Graph
GraphVerse
Hack the Graph

Overview

This hackathon is structured into three phases, each building upon the last. Participants will demonstrate algorithmic understanding, DevOps efficiency, and machine learning skills in a full-stack environment.

🔹 **Phases:**

🔸 **Phase 1: Graph Traversal Visualization**

- **Goal**: Visualize the BFS and DFS algorithms.
- **Tech Stack**: Any frontend technology (React, Vue, Angular, etc.)
- **Output**: Interactive UI showing traversal steps over a user-defined graph.

🔸 **Phase 2: Dockerize & Deploy**

- **Goal**: Dockerize the full application and implement CI/CD via GitHub Actions.
- **Tasks**:

    - Containerize the app using Docker.
    - Push Docker image to Docker Hub.
    - Automate builds and deployment using GitHub Actions.
- **Evaluation Metrics**:
    - Number of CI/CD script retries (more retries → higher penalty).
    - Nicely documented and Screen Recording.
    - Successful build and deployment pipeline.

🔸 Phase 3: Machine Learning Graph Classifier

- **Goal**: Add a Machine Learning model that classifies the graph into:
    - Tree
    - Cyclic
    - DAG (Directed Acyclic Graph)
- **Deliverables**:
    - ML model integrated into the existing application.
    - Prediction endpoint or UI feature.
- **Penalties**:

- Time taken to integrate the model.
- Poor model performance (e.g., low accuracy or misclassification).

📘 Hackathon Rule Book

🔹 General Rules

1. **This is a team-based hackathon; individuals may form groups of up to 3 members.**
2. **All phases must be completed in sequence.**
3. **You must use GitHub to maintain your project repository.**
4. **All code should be original or properly credited.**

🔹 Phase 1 Rules – BFS/DFS Visualizer

- **Participants can choose any frontend library or framework.**
- **Graph input must be dynamic (user-provided).**
- **Visualization should clearly indicate the order of visited nodes.**

🔹 Phase 2 Rules – Docker & CI/CD

- **Use Docker to containerize the complete application.**
- **Implement a CI/CD pipeline using GitHub Actions.**
- **CI should build, test, and push Docker images to Docker Hub.**
- **Multiple failed builds will result in time penalties.**

🔹 Phase 3 Rules – ML Graph Type Classifier

- **The ML model must be trained to classify graphs into:**
  - **Tree**
  - **Cyclic**
  - **DAG**
- **You may use any framework (e.g., Scikit-Learn, TensorFlow, PyTorch).**
- **Performance will be tested with hidden graph datasets.**

🏁 Submission Guidelines

- **All code must be pushed to your team's public GitHub repository.**

- **Docker images must be available on Docker Hub.**

- **Submit a final README with:**
  - **Setup Instructions**
  - **Link to Docker image**
  - **Link to deployed demo**
  - **Description of ML model used and its performance**

⛔ Disqualification Criteria

- **Misuse of CI/CD or manual Docker pushes during Phase 2.**

- **Submission after the final deadline.**