Project Title

Professional Event Web Application Deployment with DevOps Practices

Project Objective

Deliver and maintain a visually engaging, multi-section event web app built with Flask and deployed using DevOps methodologies. Interns will gain exposure to cloud infrastructure setup, automation workflows, containerization, and performance monitoring tailored for dynamic yet lightweight applications.

Technologies Implemented

- **Backend Framework**: Flask (Python-based)
- **Languages**: HTML5 and CSS3
- **Styling & Layout**: Custom CSS
- **Development Tools**: Python 3.x, Git, Visual Studio Code
- **Hosting**: AWS S3, EC2, or EKS based on project scope

Phase 1: Cloud Infrastructure & Initial Deployment (First Submission)

Week 1: Introduction & Development Environment Setup

Tasks:

- Review project structure and application flow: Home, About, Schedule, Contact.
- Set up essential tools on the local machine:
    - Python, Flask, Git, Docker, AWS CLI, Terraform.
- Obtain cloud access credentials.
- Design initial infrastructure using **Infrastructure as Code (IaC)**:
    - Use Terraform to create resources such as:
        - S3 bucket or EC2 instance for app hosting.
        - Security Groups, IAM roles, or VPC as needed.
- Store scripts in a version-controlled GitHub repository.

Week 2: Dockerization of the Application

Tasks:

- Create a Dockerfile to containerize the Flask web application.
- Include required dependencies in requirements.txt.
- Test the Docker container locally to ensure the app launches correctly.
- Push the image to **AWS ECR** or a public registry like **Docker Hub**.

Week 3: Kubernetes Deployment

Tasks:

- Write Kubernetes YAML files:
    - Deployment, Service, Ingress, and optional ConfigMaps.
- Deploy to **AWS EKS** or simulate with **Minikube** locally.
- Test endpoints and ensure assets load from the container.
- Implement auto-scaling and health checks for best practices.

📌 **Expected Deliverables:**

- Terraform scripts, Dockerfile, K8s manifests.
- Hosted version of the app on chosen infrastructure.
- GitHub repo with full source and infra code.
- **Deadline: 10/05/2025**

Phase 2: CI/CD, Observability & Final Handoff (Second Submission)

Week 4: Setting Up CI/CD Workflows

Tasks:

- Automate deployment using GitHub Actions or Jenkins.
- Create workflows to:
    - Trigger deployment on main branch commits.
    - Build, test, and push Docker image.
    - Deploy image to S3/EC2/EKS via script or IaC.

- Ensure secure storage of AWS credentials via GitHub secrets.

Week 5: Application Monitoring & Log Management

Tasks:

- Set up **AWS CloudWatch** to monitor logs, errors, and performance.
- If using Kubernetes:
    - Integrate **Prometheus** and **Grafana** for visual metrics.
    - Configure basic alerts for downtime or CPU/memory thresholds.

Week 6: Final Review & Student Presentation

Tasks:

- Prepare final documentation in the GitHub repository.
- Conduct a live demo covering:
    - Deployment flow.
    - Monitoring tools.
    - App walkthrough with feature explanations.
- Reflect on challenges and resolutions in the presentation.

📌 **Final Submission Includes:**

- GitHub repo with:
    - Infrastructure scripts, workflows, Docker/K8s files.
    - Full deployment and monitoring instructions.
- Monitoring dashboard screenshots and sample logs.
- Final presentation slides.
- **Deadline: 10/06/2025**

Submission & Collaboration Guidelines

Documentation Requirements

- Maintain a professional README.md with:

- Overview of the project.
- System architecture and cloud service descriptions.
- Step-by-step guide for local setup and cloud deployment.
- CI/CD and monitoring documentation.

Version Control Best Practices

- Use structured branches per component:
  - infra/terraform, ci-cd/github-actions, monitoring/setup, etc.
- Submit pull requests for all contributions with detailed summaries.
- Complete PR reviews and updates within 48 hours.

Team Communication & Review

- Weekly sync-up meetings for task tracking and queries.
- Final evaluation includes code quality, infrastructure readiness, and clarity of documentation.