DevOps Internship Module (6 Weeks)

Week 1: Introduction to DevOps & Environment Setup (Beginner)

Objective: Understand DevOps fundamentals and set up essential tools.

Tasks:

1. Research and write a document on **What is DevOps?** — its principles, tools, and lifecycle.
2. Install and configure **Git, Docker, and VS Code** on your system.
3. Create your **first GitHub repository** and push a sample README.md file.
4. Set up a **Docker environment** — pull and run a basic Docker image (e.g., nginx or hello-world).
5. Write a **simple shell script** to automate file creation and directory listing.

Week 2: Version Control & Continuous Integration (Beginner to Intermediate)

Objective: Learn to manage code versions and implement basic CI.

Tasks:

Work with **Git branching, merging, and resolving conflicts**.
Set up **GitHub Actions** for automatic code formatting or testing on each push.
Create a **Dockerfile** for a simple Python/Node.js app and build a custom image.
Push the Docker image to **Docker Hub** (create a free account if needed).
Research and document the **CI/CD pipeline flow** with a diagram.

Week 3: Containerization & Orchestration Basics (Intermediate)

Objective: Understand container orchestration and networking.

Tasks:

1. Run **multiple containers** using Docker Compose (e.g., web app + database).
2. Learn about **Kubernetes basics** — create and document a deployment YAML file for a sample app.
3. Set up a **local Kubernetes cluster** using Minikube or Docker Desktop.
4. Deploy a containerized application to your local Kubernetes cluster.
5. Implement **Docker volume and networking concepts** in your Docker Compose file.

Week 4: Infrastructure as Code (Intermediate to Advanced)

Objective: Automate infrastructure provisioning.

Tasks:

1. Install and configure **Terraform**.
2. Write a **Terraform script to launch an EC2 instance on AWS** (if cloud access possible — otherwise simulate).
3. Learn about **Ansible basics** — write a playbook to install Apache on a server.
4. Automate a **Docker container deployment** using an Ansible playbook.
5. Document the difference between **Ansible, Puppet, Chef, and Terraform**.

Week 5: Continuous Deployment & Monitoring (Advanced)

Objective: Automate deployments and monitor applications.

Tasks:

1. Set up a **CI/CD pipeline using Jenkins or GitHub Actions** for your Dockerized app.
2. Automate **Docker image building, testing, and deployment** on every push.
3. Install and configure **Prometheus and Grafana** for container monitoring.
4. Create and visualize **basic system metrics (CPU, Memory, Disk)** in Grafana.
5. Research and document **DevSecOps** practices and tools.

Week 6: Capstone Project — Automated Deployment (Advanced)

Objective: Deploy a complete CI/CD integrated application with monitoring.

Tasks:

1. Design a **multi-tier app (frontend + backend + database)** using Docker Compose or Kubernetes.
2. Integrate **GitHub Actions/Jenkins for CI/CD** to automate testing and deployment.
3. Implement **Prometheus + Grafana monitoring** for the app.
4. Write infrastructure scripts using **Terraform** for cloud resource provisioning (or simulate locally).
5. Record a **demo video of your working deployment pipeline** and submit your final project report.