CSE 816: Software Production Engineering Course

Final Project Guidelines and Evaluation Criteria

Total Marks: 30

- ❖ **Project Expectations:**
  The final project requires students to design and implement a complete **DevOps framework** to automate the Software Development Life Cycle (SDLC) using appropriate DevOps tools. The implementation is expected to include:

  - **Version Control**: Git and GitHub
  - **CI/CD Automation**: Jenkins, GitHub Hook Trigger for GITScm Polling, and Jenkins pipelines
  - **Containerization**: Docker and Docker Compose
  - **Configuration Management**: Ansible Playbooks
  - **Orchestration and Scaling**: Kubernetes (K8s)
  - **Monitoring and Logging**: ELK Stack (Elasticsearch, Logstash, and Kibana)

Alternatively, students may choose equivalent tools to achieve the same objectives if justified appropriately.

- ❖ **Evaluation Expectations:**
  Your project will be evaluated based on its ability to simulate real-world DevOps workflows, showcasing automation, modular design, and scalability. The following functionalities are mandatory:

  1. Incremental updates to the Git repository should trigger automated processes, including:
     - Jenkins fetching and building the updated code.
     - Running automated tests.
     - Pushing the generated Docker images to Docker Hub.
     - Deploying the Docker images to a target deployment system.
  2. Upon refreshing the application, the new changes should be visible seamlessly.

3. Application logs must feed into the ELK Stack, and the **Kibana dashboard** should visualize these logs, providing insights into application activities.

❖ **Security and Advanced Features Encouraged:**
We strongly encourage incorporating security practices and advanced features such as:

- **Secure Storage**: Use tools like Vault to securely store sensitive credentials (e.g., usernames and passwords).
- **Modular Design**: Implement modular code, such as roles in Ansible Playbooks.
- **High Availability and Scalability**: Use Horizontal Pod Autoscaling (HPA) in Kubernetes for dynamic scalability.
- **Live Patching**: Implement live patching to update the application without downtime.

❖ **Marks Distribution:**

1. **Working Project** (20 Marks):
   - Fully functional and deployable project: **20 Marks**
   - Partially functional due to last-minute issues (project demonstrates substantial completion): **15 Marks**
2. **Advanced Features** (3 Marks):
   - Usage of Vault, Roles in Ansible, and Kubernetes HPA: **3 Marks**
3. **Innovation** (2 Marks):
   - Creative or innovative solutions implemented in the project: **2 Marks**
4. **Domain-Specific Projects** (5 Marks):
   - Instead of a generic full-stack application or some web applications, projects targeting specific domains such as **MLOps**, **AIOps**, **DevSecOps**, **Networking (e.g., NFV implementation)**, **Big Data**, **Healthcare**, or **Finance** will earn additional marks: **5 Marks**

By following these guidelines, students are expected to demonstrate their understanding of DevOps methodologies and their practical application in automating complex SDLC workflows.