**PBL-3 Rubrics (8 Marks)**

| **Criteria** | **Weightage** | **Description** |
| --- | --- | --- |
| 1. **Problem Statement & Objectives,**   **Technology Stack & Tools Used** | **1 mark** | Clear and well-defined problem statement. Objectives are specific, measurable, achievable, relevant, and time-bound (SMART).  Appropriate selection of DevOps tools such as GitHub, Jenkins, Docker, Kubernetes, Terraform, and Ansible. Explanation of why these tools were chosen. |
| **2. System Architecture** | **1 mark** | Well-defined system architecture diagram demonstrating the flow of the project (CI/CD pipeline, deployment process, etc.). |
| **3. DevOps Pipeline Implementation** | **1 marks** | Detailed explanation and successful implementation of the CI/CD pipeline using Jenkins, Docker, Kubernetes, GitHub, and other relevant tools. |
| **4. Infrastructure as Code (IaC)** | **1 mark** | Proper use of Terraform and Ansible to automate infrastructure provisioning and configuration. |
| **5. Code Quality and Version Control** | **1 mark** | Code is well-structured, follows best practices, and is pushed to GitHub with version control effectively implemented. |
| **6. Containerization and Deployment** | **1 mark** | Docker containerization of the application and successful deployment on Kubernetes with proper configuration and scaling. |
| **7. Testing & Validation** | **1 mark** | Thorough testing of the application and pipeline, including automated tests and validation of successful deployment (Jenkins logs, Kubernetes health checks). |
| **8. Challenges & Solutions** | **1 mark** | Identification of challenges faced during the implementation process and effective solutions provided. |

**Evaluation Guidelines:**

* **Problem Statement & Objectives**: Ensure that the problem is clearly identified and the objectives are achievable and relevant to the scope of the project.
* **Technology Stack & Tools**: Evaluate if the selected tools are suitable for solving the problem and how well they are integrated.
* **System Architecture**: Assess if the architecture covers all relevant components and demonstrates an understanding of DevOps practices.
* **DevOps Pipeline Implementation**: Evaluate the completeness and effectiveness of the CI/CD pipeline, including automation and integration with GitHub, Jenkins, Docker, and Kubernetes.
* **IaC**: Check if Terraform and Ansible scripts are used correctly to provision and manage infrastructure, with proper documentation.
* **Code Quality**: The code should be modular, clean, and version-controlled.
* **Containerization & Deployment**: Assess how well the application is containerized and deployed with Kubernetes, including scaling and monitoring.
* **Testing & Validation**: Look for successful execution of unit tests, integration tests, and deployment validation.
* **Challenges & Solutions**: Ensure that challenges are well documented, and solutions show critical thinking and problem-solving.