| **Track(s): Devops** |
| --- |
| **Project ID:** |
| **Project Title: GameBoardPipeline** |
| **Group ID: 4** |
| **Project Short Description:**  **Board Game Full-Stack Web Application: This application displays board games and reviews. Users can log in to add or edit board games and reviews, while managers have additional rights to edit or delete reviews. The project includes setting up a CI/CD pipeline in Jenkins, which integrates SonarQube for code analysis, enforces a quality gate, and Dockerizes the application for easy deployment. Monitoring tools are used to ensure performance and reliability.** |
| **Project Context (third party interfaces, APIs, or other third party tools that will interact with your proposed design):**  **1- Jenkins**  **2- Docker**  **3- Github**  **4- SonarQube**  **5- Trivy**  **7- Nexus repository**  **8- Prometheus**  **9- Grafana** |
| **Example Application of the Proposed Project:**  **Example Use Case: Imagine a gaming community website where users can explore and review different board games. A user logs in to view a list of popular board games, leave reviews, and share their favorite picks. Managers, who oversee the content, can edit or remove inappropriate reviews. The application ensures smooth user interaction by deploying regular updates through a Jenkins CI/CD pipeline, which includes a SonarQube quality check to maintain high code standards.** |
| **Tools (Hardware and Software needed for the project / needed to build a prototype):** **Hardware:**  * **VMware Virtual Machines (VMs): You have set up 4 VMs, each with 4 GB of RAM and 40 GB of memory. These VMs will act as your development and testing environment.**  **Software:**  1. **Operating System:**    * **Linux-based OS for the VMs ( Ubuntu 22 )as it is efficient for web app hosting, Docker, and monitoring setups.** 2. **CI/CD Pipeline:**    * **Jenkins: For automating the build, testing, and deployment processes.**    * **SonarQube: Integrated with Jenkins to ensure code quality and enforce quality gates.** 3. **Containerization:**    * **Docker: For Dockerizing the application, ensuring consistency across different environments.** 4. **Monitoring Tools:**    * **Prometheus or Nagios: For monitoring performance, availability, and resource usage.**    * **Grafana: For visualizing monitoring data.** 5. **Version Control:**    * **Git/GitHub: For source control and collaboration.** |
| **Deliverables:**  **Github Project Link : there are under ( /docs/ ) all documentation and graph desin for our project →<https://github.com/moelgzar/Devops_Project>** |
| **Project Notes (Background, Data, references…..etc):** **References :**  1. **SonarQube Documentation: For setting up code analysis and quality gates.**    * **Reference: SonarQube Documentation** 2. **Jenkins Documentation: For CI/CD pipeline setup and integration with SonarQube.**    * **Reference: Jenkins Documentation** 3. **Docker Documentation: For containerizing the application and ensuring environment consistency.**    * **Reference: Docker Documentation** 4. **Monitoring Tool Documentation:**    * **Prometheus : For monitoring system performance and uptime.**    * **Grafana: For visualizing monitoring data and setting up dashboards.**    * **Reference: Prometheus, Grafana** |