



GEETHANJALI COLLEGE OF ENGINEERING AND TECHNOLOGY
(Autonomous)

Cheeryal (V), Keesara (M), Medchal Dist., Telangana - 501 301

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MINI PROJECT ABSTRACT
IV B.Tech. I SEM CSE - C Section

BATCH NUMBER: C10	Mini Project	Academic Year: 2024-2025
--------------------------	---------------------	-------------------------------------

PROJECT TITLE: Implementing CI/CD Pipeline for Website Monitoring

TEAM MEMBERS:

S.No.	Roll Number	Student Name	MailId	Contact Number
1.	21R11A05C5	Kakumanu GnanaSwaroop	21r11a05c5@gcet. edu.in	8332061573
2.	21R11A05D3	Mandha Ruchith	21r11a05d3@gcet. edu.in	8143171621
3.	21511A05E8	Srija Reddy Mitta	21r11a05e8@gcet. edu.in	9515591927

GUIDE DETAILS:

Name of the Guide	Dr. R V Sudhakar
Designation	Associate Professor
Department	CSE
MailID	rayapati1113@gmail.com
Contact Number	9705006254

*Signature of the
Project In-charge*

*Signature of the
Guide with Date*

*Signature of the
Project Coordinator*

ABSTRACT

The project is implementing a robust CI/CD pipeline for website monitoring, integrating tools like SonarQube, Nexus, Jenkins, Docker, Kubernetes, Prometheus, and Grafana. The process begins with code commits to a version control system, triggering Jenkins to start the pipeline. Jenkins uses SonarQube for static code analysis to ensure quality and security, then interacts with Nexus to manage dependencies. The application is built and packaged into Docker containers, which are pushed to a Docker registry managed by Nexus. Kubernetes handles the deployment, scaling, and management of these containers. For monitoring, Prometheus collects performance metrics, which are visualized in Grafana for real-time insights. This integrated workflow automates quality checks, dependency management, containerization, deployment, and monitoring, leading to a more efficient and reliable software delivery process.

This CI/CD pipeline implementation integrates a suite of modern tools to automate and streamline the software development lifecycle. Each tool contributes to different stages of the pipeline, from code quality assurance to deployment and monitoring, resulting in a robust, scalable, and reliable system. This integrated approach not only enhances operational efficiency but also supports a continuous delivery model, facilitating faster and more reliable software updates.

Objective: The objectives of CI/CD pipelining are to streamline software development, enhance code quality, and accelerate deployment processes. By automating build, test, and deployment tasks, CI/CD pipelines aim to achieve faster iteration cycles, reduced time-to-market, and improved collaboration among development teams. Continuous integration ensures that code changes are quickly integrated and tested, while continuous deployment automates the release process, minimizing manual intervention and reducing the risk of errors. Ultimately, CI/CD pipelines enable organizations to deliver high-quality software more efficiently, enabling them to meet customer demands and stay competitive in today's fast-paced technology landscape.

Commercializable: Yes/No: No

REFERENCES:

<https://www.redhat.com/en/topics/devops/what-cicd-pipeline>

<https://about.gitlab.com/topics/ci-cd/>

<https://www.browserstack.com/guide/building-ci-cd-pipeline>

www.metricfire.com

www.comparitech.com

Date of Submission: 29-04-2024

**Signature of the
Guide with Date**

**Signature of the
Project In-charge**