**Project Title: Building a CI/CD Pipeline with Jenkins**

**Project objective:**

Use Jenkins to set up a CI/CD pipeline that will compile and test a Maven project and deploy it to a Tomcat Server.

**Background of the problem statement:**

You’re a DevOps engineer at Pied Piper, a software company that provides web and mobile app development services. Your team is tasked with building a web app for an online book shop named Bookzy. The developers have decided to use Springboot to generate the project, Maven for compilation, and Git as the Source Code Management system for the project. You’re required to set up a Jenkins pipeline that involves three different freestyle jobs for code compilation, testing, and deployment respectively. Your Pipeline has to poll the SCM nightly for commits, build the project, and trigger the test job if the build is stable. The test job has to run unit tests, publish the JUnit test report, and trigger the freestyle job for deployment. The deployment job is required to package the Maven project, deploy the war file to a Tomcat server, and notify you via email if deployment failed.

**You must use the following:**

* **Java:** To create the website
* **Git:** As a version control system for the program
* **Jenkins:** To create the build pipeline
* **Spring boot:** To create the Maven app
* **Maven**: To compile the program
* **Tomcat**: To host the website
* **AWS EC2**: To run Tomcat
* **JUnit**: To run tests and publish results

**The following requirements should be met:**

* The app should be built with Maven.
* The Tomcat server should allow remote deployment.
* The pipeline should consist of three freestyle jobs for compilation, test, and deployment respectively.

**Prerequisite:**

* To create the Maven project, go to ​[start.​spring.​io/​](https://start.spring.io/) and select Maven as the project type. Enter Group and Artifact values and click on Generate.