**Jenkins Script:**

pipeline {

agent any

environment {

DOCKER\_IMAGE = "libsystem-app"

REGISTRY = "albeena09/libsystem-app"

MYSQL\_DATABASE = "librarydb"

MYSQL\_ROOT\_PASSWORD = "Password@1"

SPRING\_DATASOURCE\_USERNAME = "root"

SPRING\_DATASOURCE\_PASSWORD = "Password@1"

}

stages {

stage('Checkout') {

steps {

echo "Checking out source code..."

checkout scm

}

}

stage('Build with Maven') {

steps {

echo "Running Maven build..."

sh 'chmod +x ./mvnw || true'

sh './mvnw clean package -DskipTests'

}

}

stage('Unit Tests') {

steps {

echo "Running unit tests..."

sh './mvnw test'

}

}

stage('Build Docker Image') {

steps {

echo "Building Docker image..."

sh "docker build -t ${DOCKER\_IMAGE}:latest ."

}

}

stage('Push Docker Image to DockerHub') {

when {

branch 'main'

}

steps {

echo "Pushing image to DockerHub..."

withCredentials([usernamePassword(credentialsId: 'dockerhub-cred', usernameVariable: 'DOCKER\_USER', passwordVariable: 'DOCKER\_PASS')]) {

sh '''

echo "$DOCKER\_PASS" | docker login -u "$DOCKER\_USER" --password-stdin

docker tag ${DOCKER\_IMAGE}:latest ${REGISTRY}:latest

docker push ${REGISTRY}:latest

'''

}

}

}

stage('Deploy with Docker Compose') {

steps {

echo "Deploying application using Docker Compose..."

sh 'docker-compose down || true'

sh 'docker-compose up -d --build'

}

}

}

post {

always {

echo "Pipeline finished running."

}

success {

echo "Build and deployment successful!"

}

failure {

echo "Build or deployment failed."

}

}

}