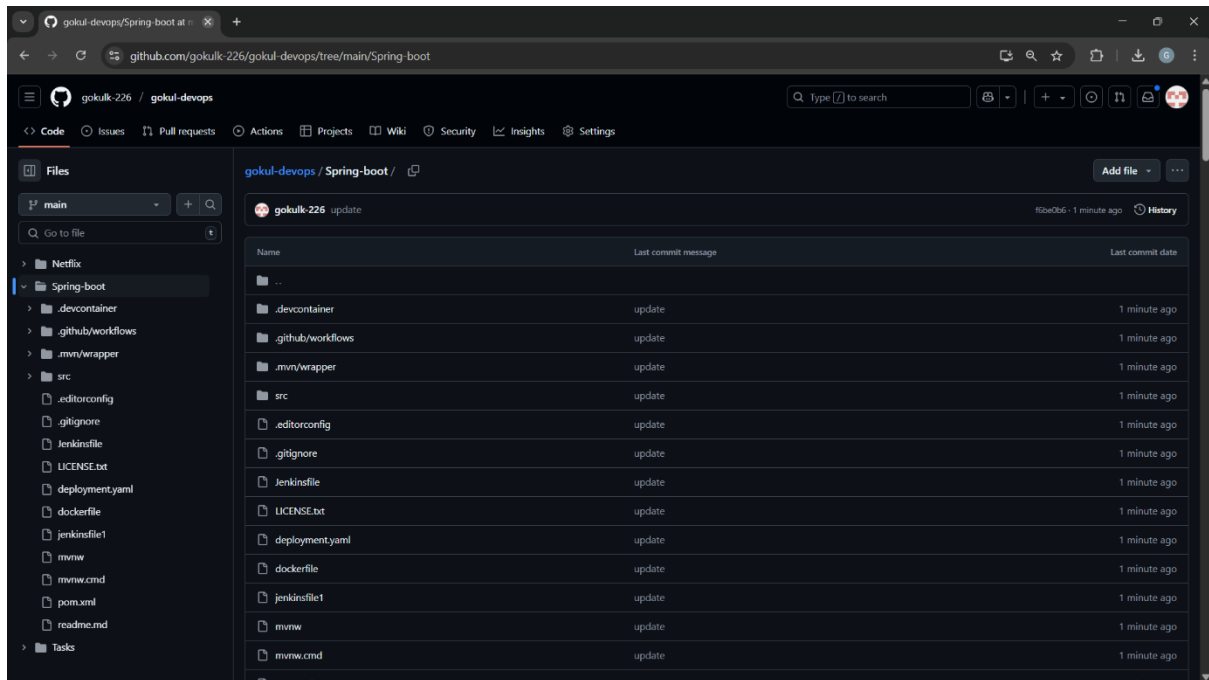# TASK-5

PUSH THE GIVEN REPO GIVEN INTO YOUR GITHUB:-

THEN USE THE BELOW PIPELINESCRIPT ACCORDING TO YOUR OWN
CREDENTIALS:-

```
pipeline {

    agent any


    environment {

        DOCKER_IMAGE = "justice/social-app"

        DOCKER_TAG = "latest"

        DOCKER_CREDENTIALS_ID = "docker"

        GITHUB_CREDENTIALS_ID = "github_seccred"

        KUBECONFIG = "/var/lib/jenkins/.kube/config"

        MAVEN_HOME = "/opt/maven"   // Set the correct Maven path

    }
```

```
stages {
    stage('Checkout Code') {
        steps {
            git credentialsId: GITHUB_CREDENTIALS_ID, url: 'https://github.com/gokulk-
226/spring-boot.git', branch: 'main'
        }
    }

    stage('Build Application') {
        steps {
            script {
                sh "${MAVEN_HOME}/bin/mvn clean package -DskipTests"
            }
        }
    }

    stage('Run Maven Tests') {
        steps {
            script {
                catchError(buildResult: 'SUCCESS', stageResult: 'FAILURE') {
                    sh "${MAVEN_HOME}/bin/mvn test"
                }
            }
        }
    }

    stage('Build Docker Image') {
        steps {
            script {
                sh "docker build -t ${DOCKER_IMAGE}:${DOCKER_TAG} ."
```

```
        }

      }

    }

    stage('Push Docker Image') {

      steps {

        withDockerRegistry([credentialsId: DOCKER_CREDENTIALS_ID, url: '']) {

          sh "docker push ${DOCKER_IMAGE}:${DOCKER_TAG}"

        }

      }

    }

    // Uncomment this stage when Kubernetes deployment is ready

    // stage('Deploy to Kubernetes') {

    //    steps {

    //        script {

    //          sh '''

    //            chmod +x scripts/deploy.sh

    //            ./scripts/deploy.sh

    //          '''

    //        }

    //    }

    // }

  }

  post {

    success {

      echo "✅ Deployment Successful!"

    }

    failure {
```

```
            echo "✖ Deployment Failed!"

        }

    }
```