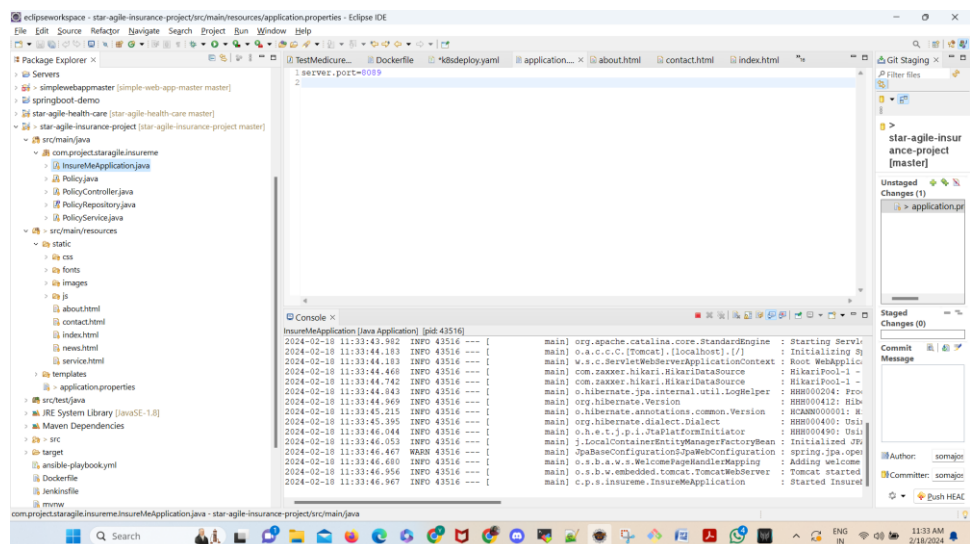
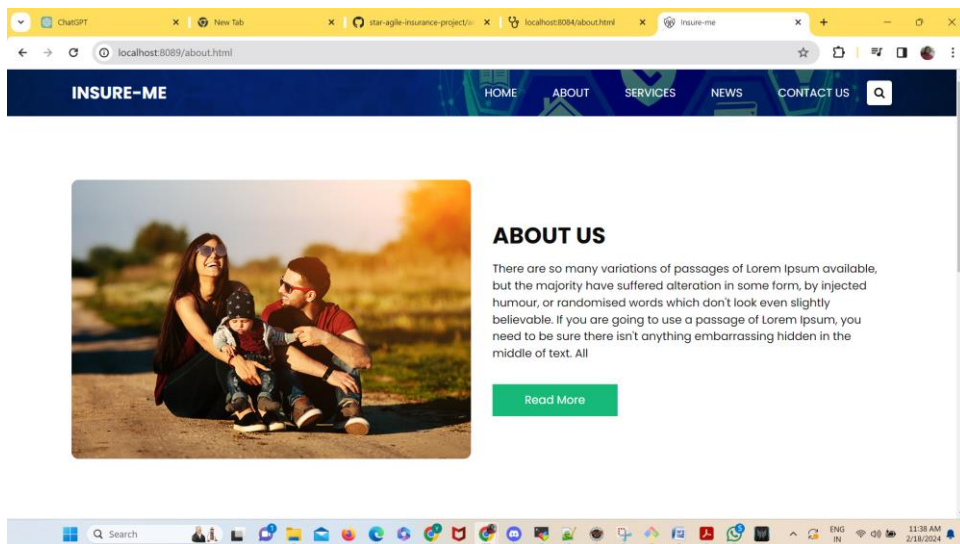
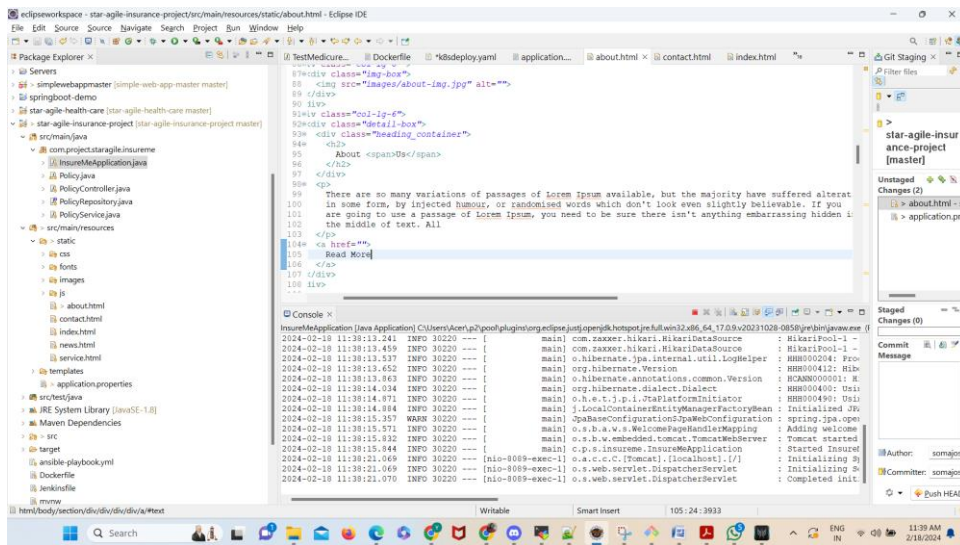


The screenshot shows an IDE with a project named 'star-agile-insurance-project' open. The left sidebar displays the 'Package Explorer' with a tree view of the project structure, including 'src/main/resources' and 'src/main/java'. The main editor area shows a file named 'index.html' containing a carousel component. The carousel is a horizontal list of images with a title 'Protect Your Family and Your Future' and a description 'Life insurance is the perfect safety net, helping you to protect your family if something unexpected happens.' The carousel is styled with a light blue background and white text. The right sidebar shows the 'Git Staging' area with a list of files to be committed. The bottom panel displays the 'Console' window with the output of a Maven build, showing that the project was successfully built and packaged into a JAR file.





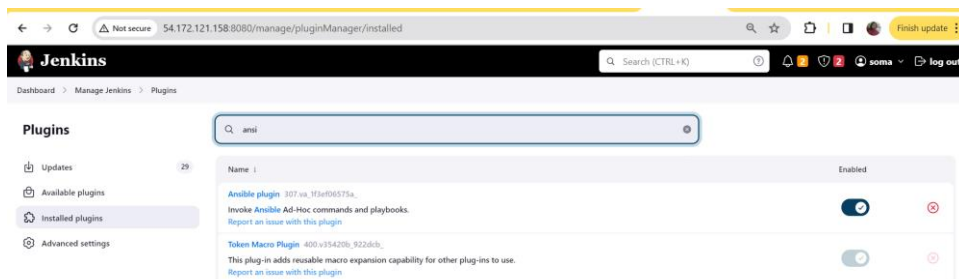
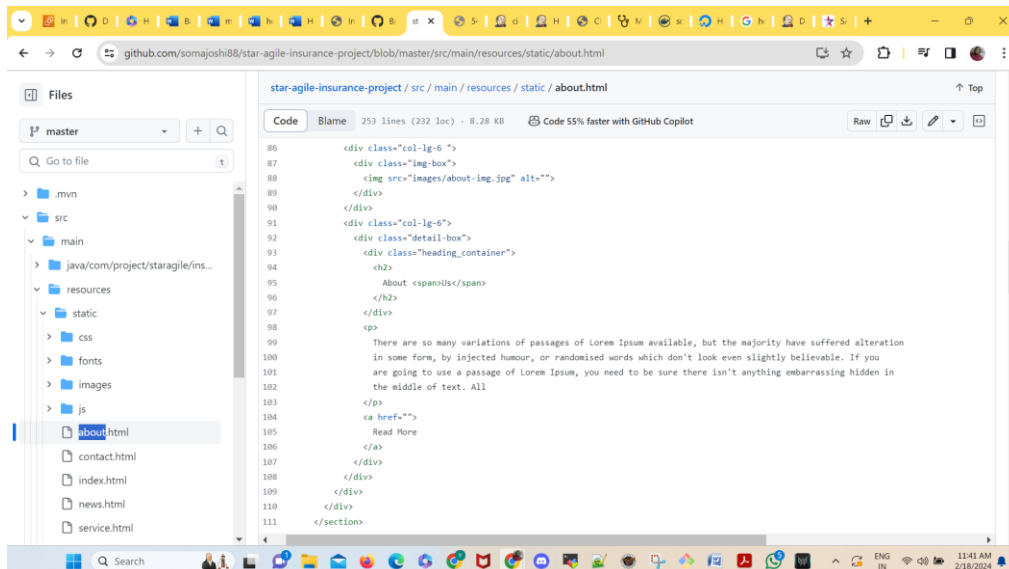
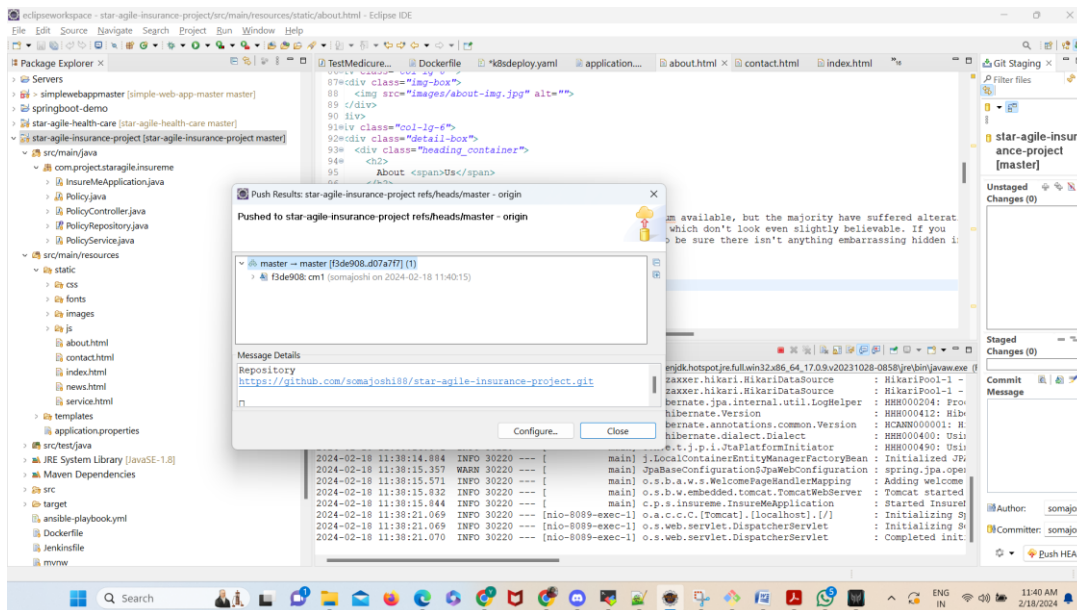


image1	devopsadmin (image1)	SSH Username with private key	image1
dockerloginid	somajoshi/***** (dockerloginid)	Username with password	dockerloginid
docker-hub	docker-hub	Secret text	docker-hub
dev-server	devopsadmin	SSH Username with private key	
dev-server-ansible	ansibleadmin (dev-server-ansible)	SSH Username with private key	dev-server-ansible

Enter an item name

Health-Insurance

* Required field

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Configuration project
Build a configuration project that need a large number of different configurations, such as testing on multiple environments, platform-specific

Configure Pipeline

Definition

Pipeline script

```

10  script {
11      git credentialsId: 'github',
12      url: 'https://github.com/somajoshi88/star-agile-insurance-project.git'
13  }
14  }
15  stage('Nexus Build') {
16      steps {
17          sh "mvn clean package"
18      }
19  }
20  }
21  stage('Docker Build') {
22      steps {
23          sh "docker build -t somajoshi/healthinsurance:${DOCKER_TAG} ."
24      }
25  }
26  }
27  }
```

☒ Use Groovy Sandbox

Pipeline Syntax

Save Apply

```
pipeline{
```

```
    agent any
```

```
    environment {
```

```
        DOCKER_TAG = getVersion()
```

```
    }
```

```
    stages{
```

```
stage('SCM'){
    steps{
        git credentialsId: 'github',
            url: 'https://github.com/somajoshi88/star-agile-insurance-project.git'
    }
}
```

```
stage('Maven Build'){
    steps{
        sh "mvn clean package"
    }
}
```

```
stage('Docker Build'){
    steps{
        sh "docker build . -t somajoshi/healthinsurance:${DOCKER_TAG} "
    }
}

stage('Login2DockerHub') {
```

```
    steps {
        withCredentials([string(credentialsId: 'docker-hub', variable: 'dockerHubPwd')]) {
            sh "docker login -u somajoshi -p ${dockerHubPwd}"
        }
    }
}
```

```
stage('DockerHub Push'){
    steps{

        sh "docker push somajoshi/healthinsurance:${DOCKER_TAG} "
```

```

    }
}

stage('Docker Deploy'){
    steps{
        ansiblePlaybook credentialsId: 'dev-server-ansible', disableHostKeyChecking: true, extras: "-e
DOCKER_TAG=${DOCKER_TAG}", installation: 'ansible', inventory: 'samplenode3.inv', playbook:
'ansible-playbook.yml'
    }
}
}
}

def getVersion(){
    def currentDir = pwd()
    echo "Present Working Directory: ${currentDir}"
    def commitHash = sh label: '', returnStdout: true, script: 'git rev-parse --short HEAD'
    return commitHash
}

```

