Source code

Jenkins pipeline script

```
pipeline{
 agent any
 tools{
    maven 'maven'
 }
 stages{
    stage('build maven war file'){
       steps{
        checkout scmGit(branches: [[name: '*/master']], extensions: [], userRemoteConfigs:
[[credentialsId: 'add3a6cc-aec7-4e37-a947-62384fd54448', url:
'https://github.com/navyamanju/Docker-Swarn.git']])
         bat 'mvn clean install'
       }
    }
    stage('build docker image'){
       steps{
          script{
            bat 'docker build -t navyamanju/awsdemo .'
          }
       }
    }
    stage('push the docker image to docker hub'){
       steps{
          script{

              withCredentials([string(credentialsId: 'dockerpassword1', variable: 'dockerpasskey')]) {

              bat 'docker login -u navyamanju -p %dockerpasskey%'
           }
           bat 'docker push navyamanju/awsdemo'
          }
       }
    }

 }
}
```

Docker file

```
FROM openjdk:8
EXPOSE 8090
ADD  target/Springproduct-0.0.1-SNAPSHOT.war Springproduct-0.0.1-SNAPSHOT.war
ENTRYPOINT ["java","-jar","/Springproduct-0.0.1-SNAPSHOT.war"]
```

Spring boot application code

Product Controller

```
package com.example.demo;


import java.util.List;
import java.util.logging.Logger;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.servlet.ModelAndView;



@Controller
public class ProductController {
@Autowired
        ProductDAO dao;
Logger log=Logger.getAnonymousLogger();

@RequestMapping("/")
public ModelAndView displayHome(HttpServletRequest request,HttpServletResponse response) {
        ModelAndView mv=new ModelAndView();
        log.info("object for model and view created ");
        mv.setViewName("index.jsp");
        log.info("going to the page index.jsp");
        return mv;
}

@RequestMapping("/insert")
public ModelAndView insert(HttpServletRequest request,HttpServletResponse response) {
        ModelAndView mv=new ModelAndView();
        log.info("object for model and view created ");
```

```
    Product product=new Product();
    product.setPname(request.getParameter("pname"));
    product.setCost(Integer.parseInt(request.getParameter("pcost")));
    log.info("setting of values done" +product.getPname()+"  "+product.getCost());
    Product order=dao.insert(product);
    if(order!=null) {
        mv.setViewName("success.jsp");
        log.info("going to success.jsp page");
    }
    else {
        mv.setViewName("fail.jsp");
        log.info("going to fail.jsp page");
    }
    return mv;
}

@RequestMapping("/display")
public ModelAndView display(HttpServletRequest request,HttpServletResponse response) {
    ModelAndView mv=new ModelAndView();
    log.info("object for model and view created ");
    Product product=new Product();
    List<Product> order=dao.getall();
    mv.addObject("list",order);
    mv.setViewName("display.jsp");
    return mv;
}


}
```

Product repository

```
package com.example.demo;


import org.springframework.data.jpa.repository.JpaRepository;
import com.example.demo.Product;
                            //PojoClass,pK type
public interface ProductRepository extends JpaRepository<Product,Integer> {
//this interface has all the crud operations with predefined methods in it
//in this interface if we need our own query we can frame it by using JPQL @Query
}
```

Index.jsp

```
<!DOCTYPE html>
<html>
<head>
   <meta charset="ISO-8859-1">
   <title>Insert title here</title>
</head>
<body>
   <h1 style="text-align: center;">Welcome to Our Product Page</h1>
   <form action="addproduct.jsp" style="text-align: center; margin-top: 20px;">
       <input type="submit" value="Add product" style="background-color: #007bff; color: #fff; border: none; padding: 10px 20px; border-radius: 5px; cursor: pointer;">
   </form>
</body>
</html>
```