Controller:

```
packagecom.example.UserFeedback.controllers;

importorg.springframework.beans.factory.annotation.Autowired;
importorg.springframework.ui.ModelMap;
importorg.springframework.web.bind.annotation.GetMapping;
importorg.springframework.web.bind.annotation.PathVariable;
importorg.springframework.web.bind.annotation.RequestParam;
```

Controller:

```
importorg.springframework.web.bind.annotation.ResponseBody;
importorg.springframework.web.bind.annotation.RestController;

importcom.example.UserFeedback.entities.Feedback;
importcom.example.UserFeedback.repositories.FeedbackRepository;
importcom.example.UserFeedback.services.FeedbackService;

@RestController
public class FeedbackController {

        @Autowired
        FeedbackServicefeedbackService;
        @Autowired
        FeedbackRepositoryfeedbackRepository;


        @GetMapping(value="/")
public String showIndexPage(ModelMap model){
                return "<html>\n"
                                + "<head>\n"
                                + "        <style>\n"
                                + "                .center {\n"
                                + "                        text-align: center;\n"
                                + "                }\n"
                                + "                \n"
                                + "        </style>\n"
                                + "</head>\n"
                                + "<body style=\"background-color:lightblue;\">\n"
                                + "        <div class=\"center\">\n"
                                + "                <h1>User Feedback Page</h1>\n"
                                + "                \n"
                                + "                <h2 class=\"hello-title\">View and Send User
Feedback</h2>\n"
                                + "                \n"
                                + "                <a href=\"/feedback\">View all feedback</a>\n"
                                + "                <br><br>\n"
                                + "<form method=\"get\" action=\"update\">\n"
                                + "                        <br><h3>Enter your feedback below:</h3>\n"
```

```
                                        + "                                   <input type=\"text\" id=\"comment\"
name=\"comment\" placeholder=\"Comment Here\" required>\n"

                                        + "                                   <input type=\"number\" id=\"rating\"
name=\"rating\" placeholder=\"Rating Here\" required>\n"

                                        + "                                   <input type=\"text\" id=\"name\"
name=\"name\" placeholder=\"Name Here\" required> \n"

                                        + "                                   <input type=\"submit\" value=\"Enter\" />\n"

                                        + "                       </form>"

                                        + "         </div>\n"

                                        + "</body>\n"

                                        + "</html>";

   }


          @GetMapping("/feedback")

          public @ResponseBody String getAllFeedbacks() {

      // This returns a JSON or XML with the Feedbacks

Iterable<Feedback>allFB = feedbackRepository.findAll();

                    return "<html>\n"

                                        + "<head>\n"

                                        + "         <style>\n"

                                        + "                       .center {\n"

                                        + "                                   text-align: center;\n"

                                        + "                       }\n"

                                        + "                       \n"

                                        + "         </style>\n"

                                        + "</head>\n"

                                        + "<body style=\"background-color:lightblue;\">\n"
                                        + "         <div class=\"center\">\n"
                                        + "<h1>Feedback Table</h1>\n"
                    + allFB.toString()
                            + " </div>\n"
                                        + "</body>\n"
                                        + "</html>";
   }



}
packagecom.example.UserFeedback.controllers;


importorg.springframework.beans.factory.annotation.Autowired;
importorg.springframework.web.bind.annotation.GetMapping;
importorg.springframework.web.bind.annotation.RequestMapping;
importorg.springframework.web.bind.annotation.RequestMethod;
importorg.springframework.web.bind.annotation.RequestParam;
importorg.springframework.web.bind.annotation.RestController;


importcom.example.UserFeedback.entities.Feedback;
importcom.example.UserFeedback.repositories.FeedbackRepository;
importcom.example.UserFeedback.services.FeedbackService;


@RestController
public class RestUpdateController
```

```
{

            @Autowired
            FeedbackServicefeedbackService;

            @Autowired
            FeedbackRepositoryfeedbackRepository;

@GetMapping("/update")
   public String getEmployeeByName(@RequestParam("comment") String comment,
@RequestParam("rating") int rating, @RequestParam("name") String name) {

            Feedback f = new Feedback(count()+1, comment, rating, name);
            feedbackRepository.save(f);

return "<html>\n"
                                        + "<head>\n"
                                        + "       <style>\n"
                                        + "                 .center {\n"
                                        + "                           text-align: center;\n"
                                        + "                 }\n"
                                        + "                 \n"
                                        + "       </style>\n"
                                        + "</head>\n"
                                        + "<body style=\"background-color:lightblue;\">\n"
                                        + "        <div class=\"center\">\n"
                                        + "                <h1>User Feedback Page</h1>\n"
                                        + "                \n"
                                        + "                <h2 class=\"hello-title\">Successfully Added Your
Feedback</h2>\n"
                                        + "                \n"
                                        + "                <a href=\"/feedback\">Click here to view all
feedback</a>\n"
                                        + "        </div>\n"
                                        + "</body>\n"
                                        + "</html>";
  }

public Integer count() {
        int i = 1;
        while(feedbackRepository.existsById(i))
                i++;
        return i;
  }
}
```

Entity:

```
packagecom.example.UserFeedback.entities;

importjavax.persistence.Entity;
importjavax.persistence.GeneratedValue;
importjavax.persistence.GenerationType;
importjavax.persistence.Id;

importlombok.Data;

@Entity
@Data
public class Feedback {

        @Id
@GeneratedValue(strategy=GenerationType.AUTO)
private Integer id;
        private String comments;
        privateint rating;
        private String user;

        public Feedback()
        {

        }
        public Feedback(Integer id, String comments, int rating, String user)
        {
                this.id = id;
                this.comments = comments;
                this.rating = rating;
```

```
                    this.user = user;
          }

          @Override
          public String toString()
          {
                    return "<br><h3>" + user + " [" + id + "]" + " commented:</h3><h4>\"" + comments +
"\"  and rated: " + rating + "</h4><br>";
          }
}
```

Repositories:

```
packagecom.example.UserFeedback.repositories;

importorg.springframework.data.repository.CrudRepository;

importcom.example.UserFeedback.entities.*;

public interface FeedbackRepository extends CrudRepository<Feedback, Integer>{

}
```

Services:

```
packagecom.example.UserFeedback.services;
```

```
importorg.springframework.beans.factory.annotation.Autowired;

importorg.springframework.stereotype.Service;


importcom.example.UserFeedback.entities.Feedback;

importcom.example.UserFeedback.repositories.*;


@Service

public class FeedbackService {


	@Autowired

	privateFeedbackRepositoryfeedbackRepository;


	publicIterable<Feedback>GetAllFeedback() {

		returnfeedbackRepository.findAll();

	}




}
```

User Login:

```
packagecom.example.UserFeedback;


importorg.springframework.boot.SpringApplication;
importorg.springframework.boot.autoconfigure.SpringBootApplication;


@SpringBootApplication
public class UserFeedbackApplication {


        public static void main(String[] args) {
                SpringApplication.run(UserFeedbackApplication.class, args);
        }


}
packagecom.example.UserFeedback;


importorg.junit.jupiter.api.Test;
importorg.springframework.boot.test.context.SpringBootTest;


@SpringBootTest
classUserFeedbackApplicationTests {


        @Test
        voidcontextLoads() {
        }


}
#!/bin/sh
# ----------------------------------------------------------------------------
```

Pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

	xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">

	<modelVersion>4.0.0</modelVersion>

	<parent>

		<groupId>org.springframework.boot</groupId>

		<artifactId>spring-boot-starter-parent</artifactId>

		<version>2.4.3</version>

		<relativePath/><!-- lookup parent from repository -->

	</parent>

	<groupId>com.example</groupId>

	<artifactId>UserFeedback</artifactId>

	<version>0.0.1-SNAPSHOT</version>

	<name>UserFeedback</name>

	<description>Demo project for Spring Boot</description>

	<properties>

		<java.version>1.8</java.version>

	</properties>

	<dependencies>

		<dependency>

			<groupId>org.springframework.boot</groupId>
```

```xml
			<artifactId>spring-boot-starter-data-jpa</artifactId>
		</dependency>
		<dependency>
			<groupId>org.springframework.boot</groupId>
			<artifactId>spring-boot-starter-web</artifactId>
		</dependency>

		<dependency>
			<groupId>mysql</groupId>
			<artifactId>mysql-connector-java</artifactId>
			<scope>runtime</scope>
		</dependency>
		<dependency>
			<groupId>org.projectlombok</groupId>
			<artifactId>lombok</artifactId>
			<optional>true</optional>
		</dependency>
		<dependency>
			<groupId>javax.servlet</groupId>
			<artifactId>jstl</artifactId>
			<version>1.2</version>
		</dependency>



<dependency>
```

```
<groupId>org.apache.tomcat.embed</groupId>
<artifactId>tomcat-embed-jasper</artifactId>
<scope>provided</scope>
</dependency>
                <dependency>
                        <groupId>org.springframework.boot</groupId>
                        <artifactId>spring-boot-starter-test</artifactId>
                        <scope>test</scope>
                </dependency>
        </dependencies>

        <build>
                <plugins>
                        <plugin>
                                <groupId>org.springframework.boot</groupId>
                                <artifactId>spring-boot-maven-plugin</artifactId>
                                <configuration>
                                        <excludes>
                                                <exclude>
                                                        <groupId>org.projectlombok</groupId>
                                                        <artifactId>lombok</artifactId>
                                                </exclude>
                                        </excludes>
                                </configuration>
                        </plugin>
```

```
                    </plugins>

            </build>


</project>




import java.net.*;

import java.io.*;

importjava.nio.channels.*;

importjava.util.Properties;


public class MavenWrapperDownloader {


private static final String WRAPPER_VERSION = "0.5.6";

   /**

    * Default URL to download the maven-wrapper.jar from, if no 'downloadUrl' is provided.

    */

private static final String DEFAULT_DOWNLOAD_URL =
"https://repo.maven.apache.org/maven2/io/takari/maven-wrapper/"

        + WRAPPER_VERSION + "/maven-wrapper-" + WRAPPER_VERSION + ".jar";


   /**

    * Path to the maven-wrapper.properties file, which might contain a downloadUrl property to

    * use instead of the default one.

    */

private static final String MAVEN_WRAPPER_PROPERTIES_PATH =
           ".mvn/wrapper/maven-wrapper.properties";

   /**
    * Path where the maven-wrapper.jar will be saved to.
    */
private static final String MAVEN_WRAPPER_JAR_PATH =
           ".mvn/wrapper/maven-wrapper.jar";

   /**
    * Name of the property which should be used to override the default downloadurl for the wrapper.
    */
private static final String PROPERTY_NAME_WRAPPER_URL = "wrapperUrl";

public static void main(String args[]) {
System.out.println("- Downloader started");
       File baseDirectory = new File(args[0]);
System.out.println("- Using base directory: " + baseDirectory.getAbsolutePath());

       // If the maven-wrapper.properties exists, read it and check if it contains a custom
       // wrapperUrl parameter.
       File mavenWrapperPropertyFile = new File(baseDirectory, MAVEN_WRAPPER_PROPERTIES_PATH);
       String url = DEFAULT_DOWNLOAD_URL;
if(mavenWrapperPropertyFile.exists()) {
FileInputStreammavenWrapperPropertyFileInputStream = null;
try {
```

```
mavenWrapperPropertyFileInputStream = new FileInputStream(mavenWrapperPropertyFile);
            Properties mavenWrapperProperties = new Properties();
mavenWrapperProperties.load(mavenWrapperPropertyFileInputStream);
url = mavenWrapperProperties.getProperty(PROPERTY_NAME_WRAPPER_URL, url);
         } catch (IOException e) {
System.out.println("- ERROR loading '" + MAVEN_WRAPPER_PROPERTIES_PATH + "'");
         } finally {
try {
if(mavenWrapperPropertyFileInputStream != null) {
mavenWrapperPropertyFileInputStream.close();
               }
            } catch (IOException e) {
               // Ignore ...
            }
         }
      }
System.out.println("- Downloading from: " + url);

      File outputFile = new File(baseDirectory.getAbsolutePath(), MAVEN_WRAPPER_JAR_PATH);
if(!outputFile.getParentFile().exists()) {
if(!outputFile.getParentFile().mkdirs()) {
System.out.println(
                "- ERROR creating output directory '" + outputFile.getParentFile().getAbsolutePath() + "'");
         }
      }
```

```
System.out.println("- Downloading to: " + outputFile.getAbsolutePath());
try {
downloadFileFromURL(url, outputFile);
System.out.println("Done");
System.exit(0);
    } catch (Throwable e) {
System.out.println("- Error downloading");
e.printStackTrace();
System.exit(1);
    }
  }

private static void downloadFileFromURL(String urlString, File destination) throws Exception {
if (System.getenv("MVNW_USERNAME") != null &&System.getenv("MVNW_PASSWORD") != null) {
      String username = System.getenv("MVNW_USERNAME");
char[] password = System.getenv("MVNW_PASSWORD").toCharArray();
Authenticator.setDefault(new Authenticator() {
        @Override
protectedPasswordAuthenticationgetPasswordAuthentication() {
return new PasswordAuthentication(username, password);
        }
      });
    }
    URL website = new URL(urlString);
ReadableByteChannelrbc;
```

```
rbc = Channels.newChannel(website.openStream());

FileOutputStreamfos = new FileOutputStream(destination);

fos.getChannel().transferFrom(rbc, 0, Long.MAX_VALUE);

fos.close();

rbc.close();

    }


}
```