Project: Handling User Authentication

Project objective:

Set up a standalone project to do unit testing of the user authentication class which is used in the main web application. The objective is to create a JUnit class that will test all aspects of the authentication class.

Background of the problem statement:

As a part of developing an ecommerce web application, a test-suite is being created to do unit testing of all backend components in the web application. This project will test the user authentication class. This project will be a standalone Java application, since Junit does not directly test servlets or web pages. We are only testing the classes that have the business logic.

Step1: Creating a Maven Project which is web enabled

- Open Eclipse
- Go to the **File** menu. Choose **New->Maven Project**
- Check **Create a Simple Project** and check **Use Default Workspace Location** and click on **Next**
- From the **archetype** list, choose the row that has **Artifact Id** as **maven-archetype-webapp** and click on **Next**
- Enter **Group Id** as **com.project** and **Artifact Id** as **Authentication** and click on **Finish**
- This will create the project files in the Project Explorer
- Before building the project, we need to confirm that the **servlet.jar** has been added to the project

- Servlet is already installed in your practice labs.

Step 2: Running the project

- To run the project, open a web browser and type:
  http://localhost:8080/SpringSetup

Step 1.3.14: Pushing the code to your GitHub repositories

- Open your command prompt and navigate to the folder where you have created your files.

  cd <folder path>

- Initialize your repository using the following command:

  git init

- Add all the files to your git repository using the following command:

  git add .

- Commit the changes using the following command:

  git commit . -m "Changes have been committed."

- Push the files to the folder you initially created using the following command:

  git push -u origin master