Date 12/4/2023

CSM3023

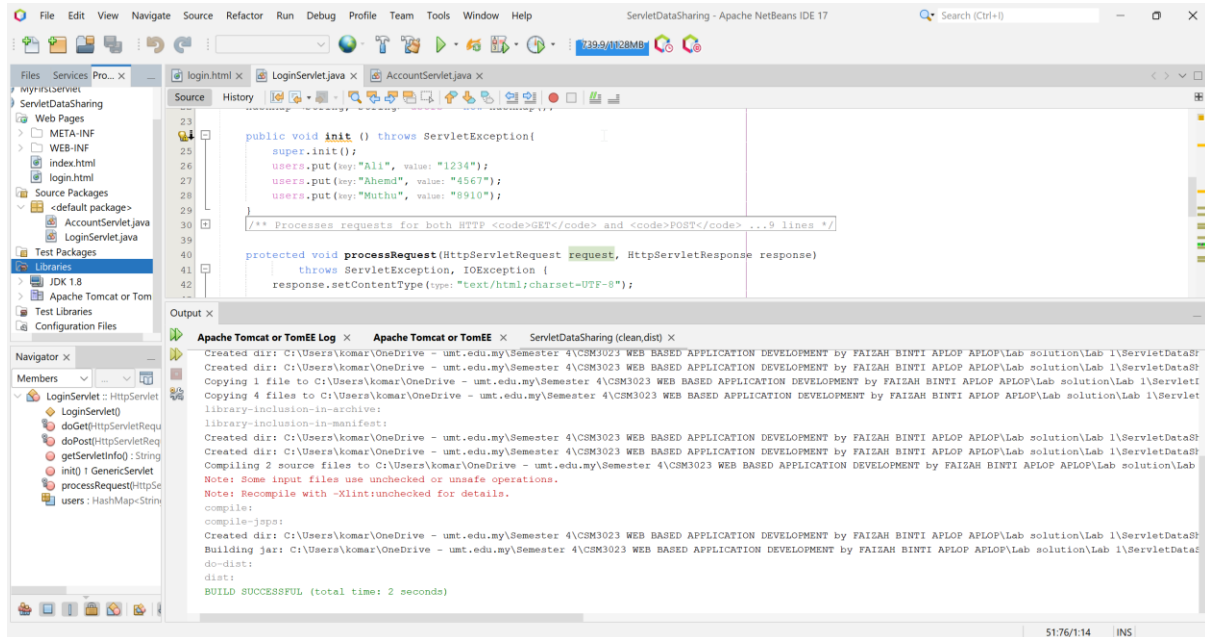*WEB BASED APPLICATION DEVELOPMENT*

DR FAIZAH BINTI APLOP APLOP
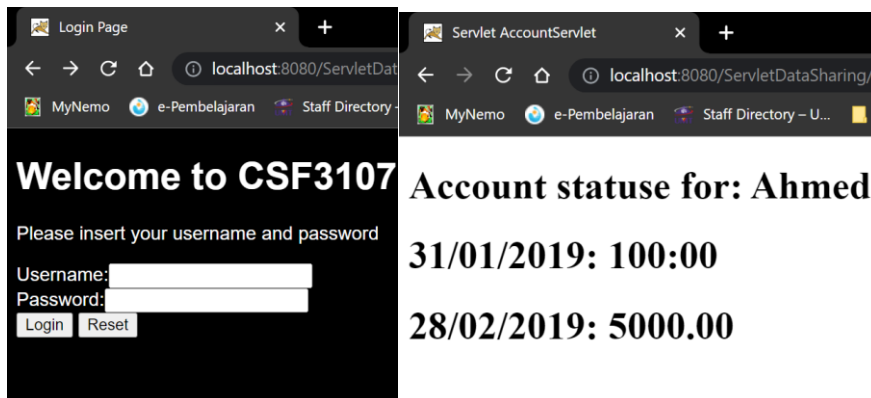
OMAR ISMAIL ABDJALEEL ALOMORY

LAB 2 (MP2)

Task 1

Complied successfully.

Output

1. What have you you learnt from this exercise?

We leant how to server side is working, checking if the information enter is valid then process and deliver the data from the database or anywhere.

2. What are the common methods used in Java Servlet?

- destroy()
- getServletConfig()
- getServletInfo()
- init()
- service(ServletRequest, ServletResponse)
- doGet()
- doPost()

Task 2

		id	username	password	roles
☐ Edit Copy Delete		1	Ali	1234	admin
☐ Edit Copy Delete		2	Ahmed	4567	user

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	
username	varchar(100)	NO		NULL	
password	varchar(255)	NO		NULL	
roles	varchar(10)	NO		NULL	

Task 3

Adding mySQL Driver
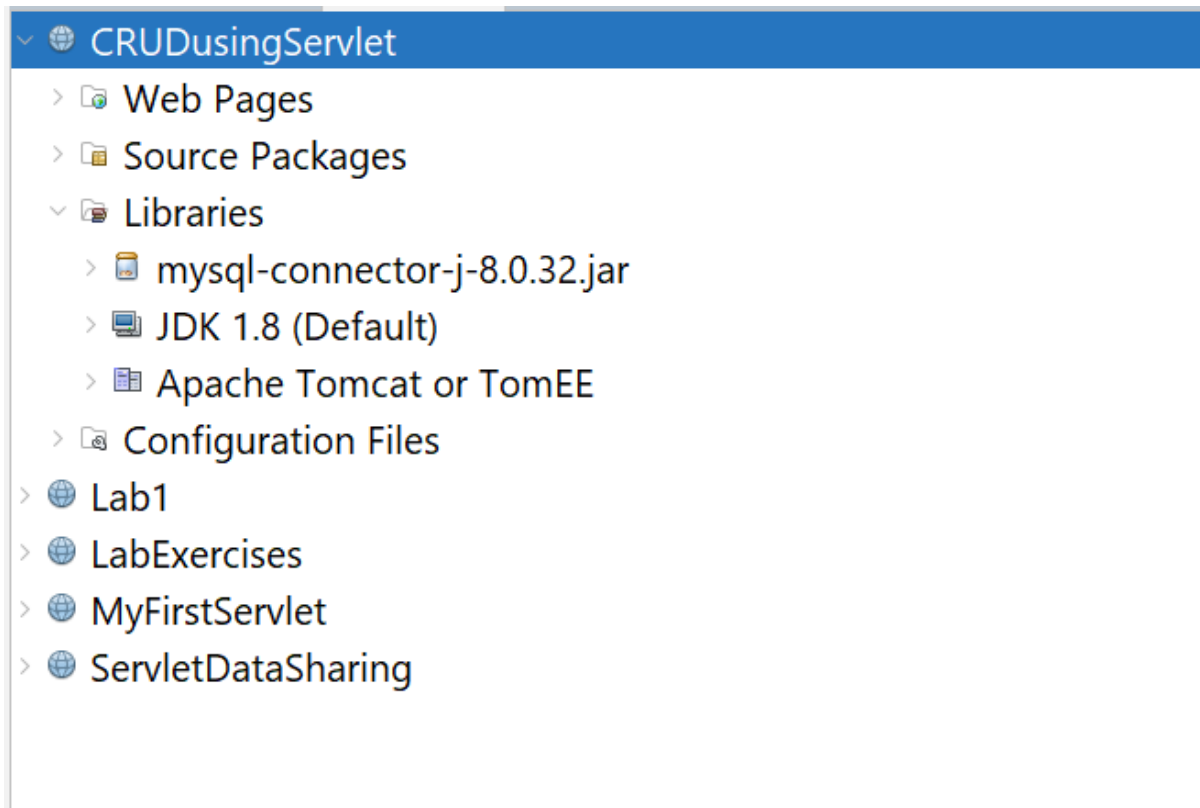
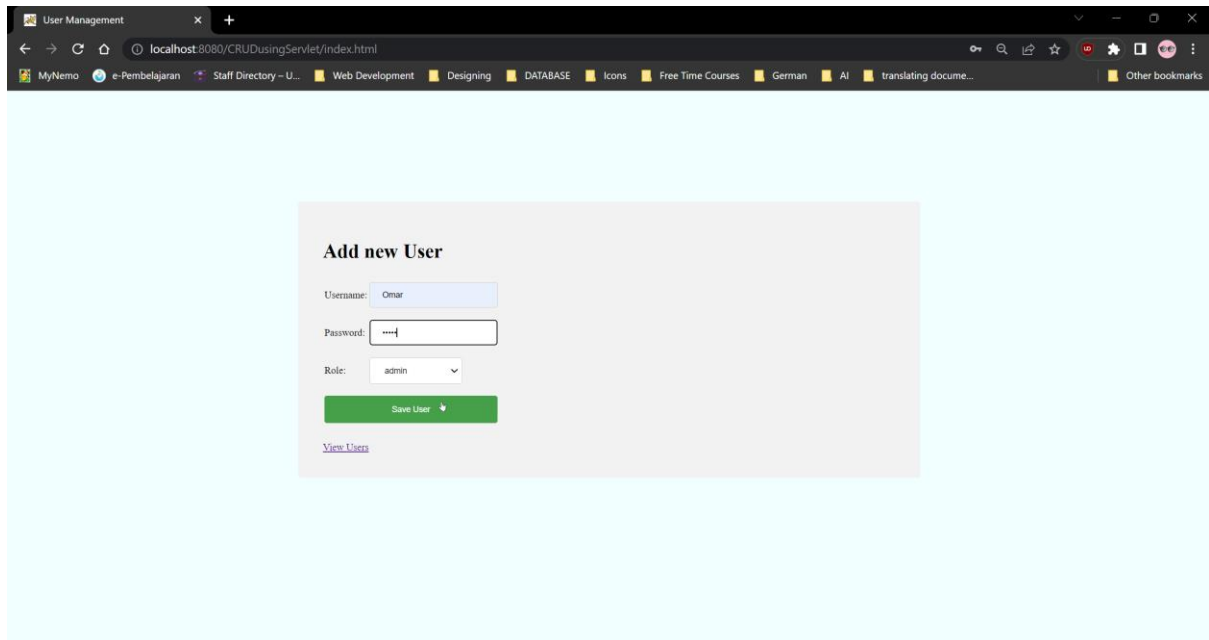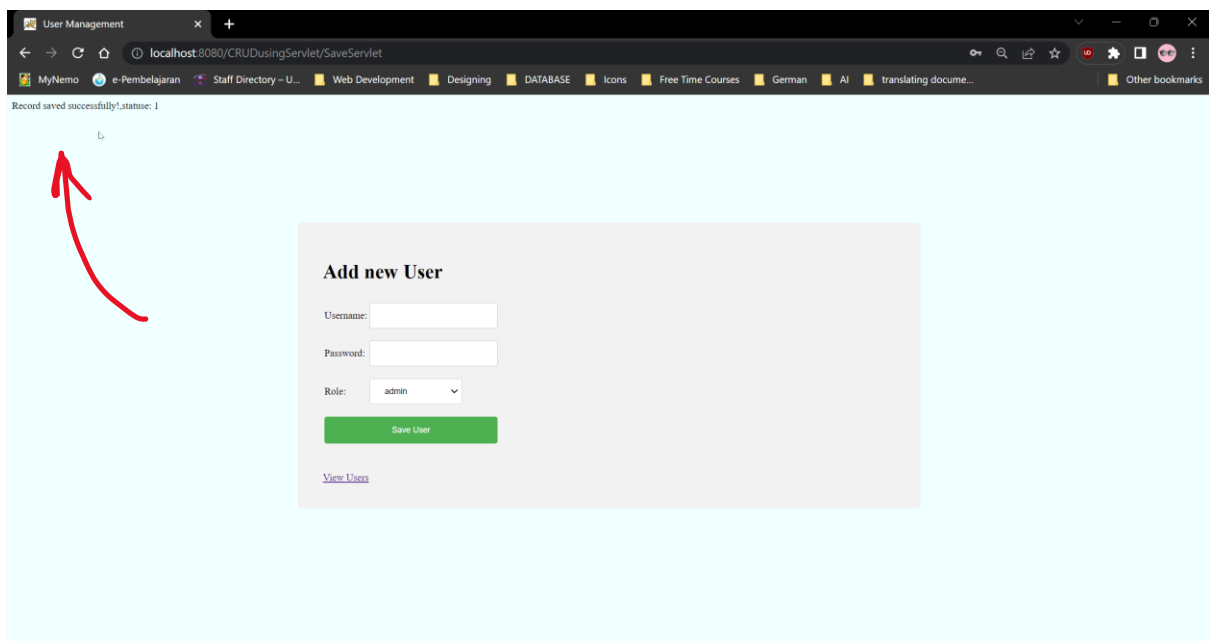# Task 4

Create: adding user/admin

Respond success

Retrieve: Added successfully (id = 11 because I tried multiple times 😊)

Update: Edit user/admin

Respond success and redirect to index.html

Add new User

Username: [   ]

Password: [   ]

Role: [admin ▾]

[Save User]

View Users

Delete: remove user/admin

Add New User

User List

Id	Name	Password	Role	Edit	Delete
1	Ali	1234	admin	edit	delete
2	Ahmad	4567	user	edit	delete
11	OMAR ALOMORY	55848	admin	edit	delete

Add New User

User List

Id	Name	Password	Role	Edit	Delete
1	Ali	1234	admin	edit	delete
2	Ahmad	4567	user	edit	delete

Reflections:

1. What is the name of the Java Library that you need to import before coding the web application with database operations?

mysql-connector-j-8.0.32.jar (MySQL JDBS Driver)

2. Which folder keeps the web.xml file? Copy the contents of the file and explain in brief the tags included such as <servlet-name><servlet-class> <servlet-mapping>. etc.

The folder that keeps the web.xml file is WEB-INF.

Code of web.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd">
    <servlet>
        <servlet-name>SaveServlet</servlet-name>
        <servlet-class>SaveServlet</servlet-class>
    </servlet>
    <servlet>
        <servlet-name>ViewServlet</servlet-name>
        <servlet-class>ViewServlet</servlet-class>
    </servlet>
    <servlet>
        <servlet-name>EditServlet</servlet-name>
        <servlet-class>EditServlet</servlet-class>
    </servlet>
    <servlet>
        <servlet-name>EditServlet2</servlet-name>
        <servlet-class>EditServlet2</servlet-class>
    </servlet>
    <servlet>
        <servlet-name>DeleteServlet</servlet-name>
        <servlet-class>DeleteServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>SaveServlet</servlet-name>
        <url-pattern>/SaveServlet</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
        <servlet-name>ViewServlet</servlet-name>
        <url-pattern>/ViewServlet</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
        <servlet-name>EditServlet</servlet-name>
        <url-pattern>/EditServlet</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
        <servlet-name>EditServlet2</servlet-name>
        <url-pattern>/EditServlet2</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
        <servlet-name>DeleteServlet</servlet-name>
        <url-pattern>/DeleteServlet</url-pattern>
    </servlet-mapping>
    <session-config>
        <session-timeout>
            30
        </session-timeout>
    </session-config>
</web-app>
```

- The XML declaration, **<?xml version="1.0" encoding="UTF-8"?>**, specifies that the file is an XML document and uses UTF-8 encoding.

- The root element **<web-app>** is used to define the entire web application and contains other elements that define the configuration for the application.

- The version attribute of the **<web-app>** element specifies the version of the Servlet API that the web application conforms to. In this case, the version is 3.1.

- The **xmlns** attribute defines the namespace of the elements and attributes used in the file.

- The value **http://xmlns.jcp.org/xml/ns/javaee** indicates that the file is using the Java EE namespace.

- The **xmlns:xsi** attribute is used to declare the namespace of the XML schema instance, which is used for validating the XML document.

- The **xsi:schemaLocation** attribute specifies the location of the XML schema that defines the elements and attributes used in the file. In this case, it points to the XML schema for version 3.1 of the Servlet API.

- **<servlet-name>** is used to provide a unique name for the servlet within the web application. It is used to identify the servlet in the web.xml deployment descriptor file and also used to map the servlet to a URL.

- **<servlet-class>** is used to specify the fully qualified class name of the servlet. This class must implement the javax.servlet.Servlet interface and provide implementation for its methods.

- **<servlet-mapping>** is used to map the servlet to a URL pattern. This tag consists of two sub-tags - **<servlet-name>** and **<url-pattern>**. The **<servlet-name>** tag specifies the name of the servlet as defined in the **<servlet>** tag, and the **<url-pattern>** tag specifies the URL pattern that the servlet should handle.

- The **<session-config>** tag is used to configure session-related parameters for a web application. The **<session-timeout>** tag is a sub-tag of **<session-config>** that specifies the amount of time (in minutes) that a session should remain active before it is invalidated by the server. (In our example is set to 30 minutes This means that if a user is inactive for 30 minutes or more, their session will be invalidated by the server and any session data associated with that session will be lost.)

3. Define the usage of Data Access Object (DAO) servlet. How it eases the business process in your servlet-based web application?

- It handles requests related to data persistence. Acting as an intermediary between the business logic of the application and the data source, allowing the business logic to interact with the data source in a more abstracted and simplified manner.
- Encapsulate the details of creating and managing database connections, performing queries and updates, and handling transactions.
- It makes the application easy to maintain, changes to the underlying data source, such as switching to a different database or changing the schema, without having to modify the business logic of the application.