



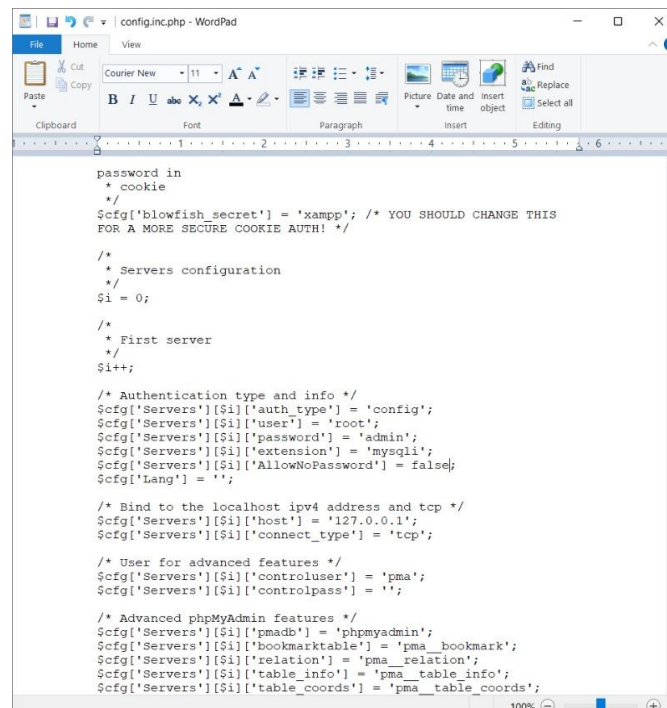
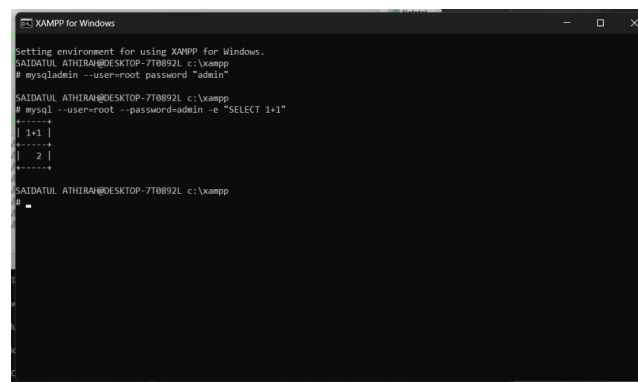
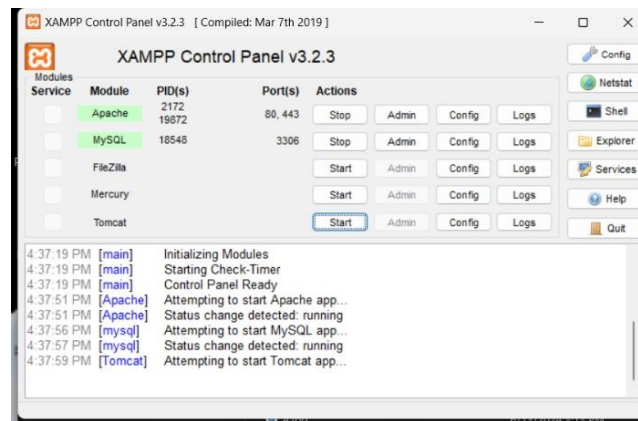
CSM3023 WEB APPLICATION DEVELOPMENT (K1)

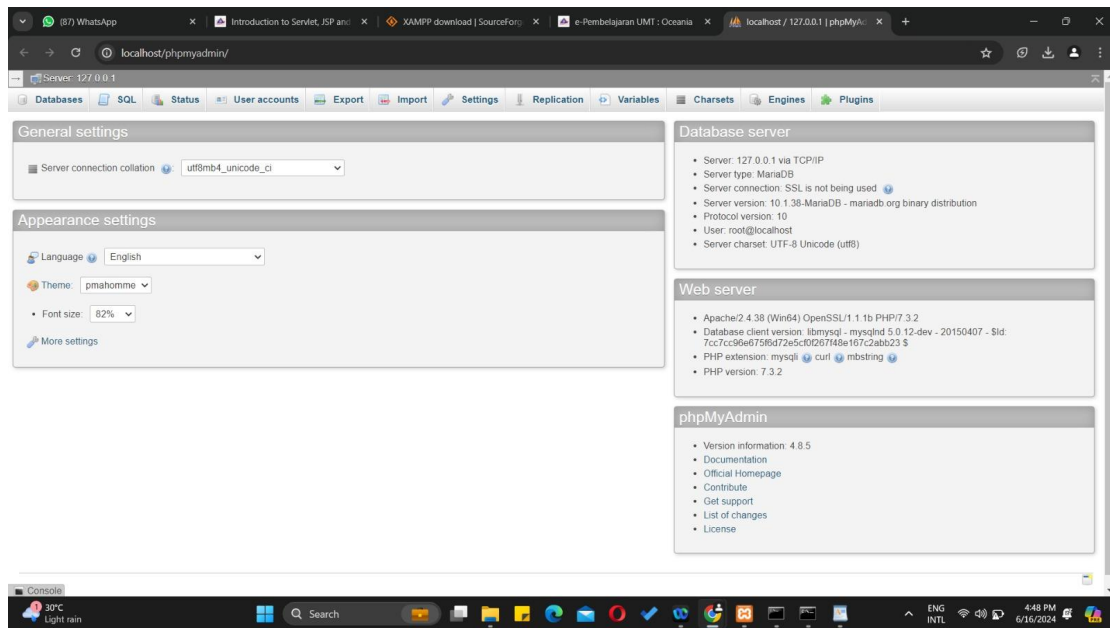
SEMESTER IV, SESSION 2023/2024

**LAB 1 : INTRODUCTION TO SERVLET, JSP AND
MYSQL DATABASE**

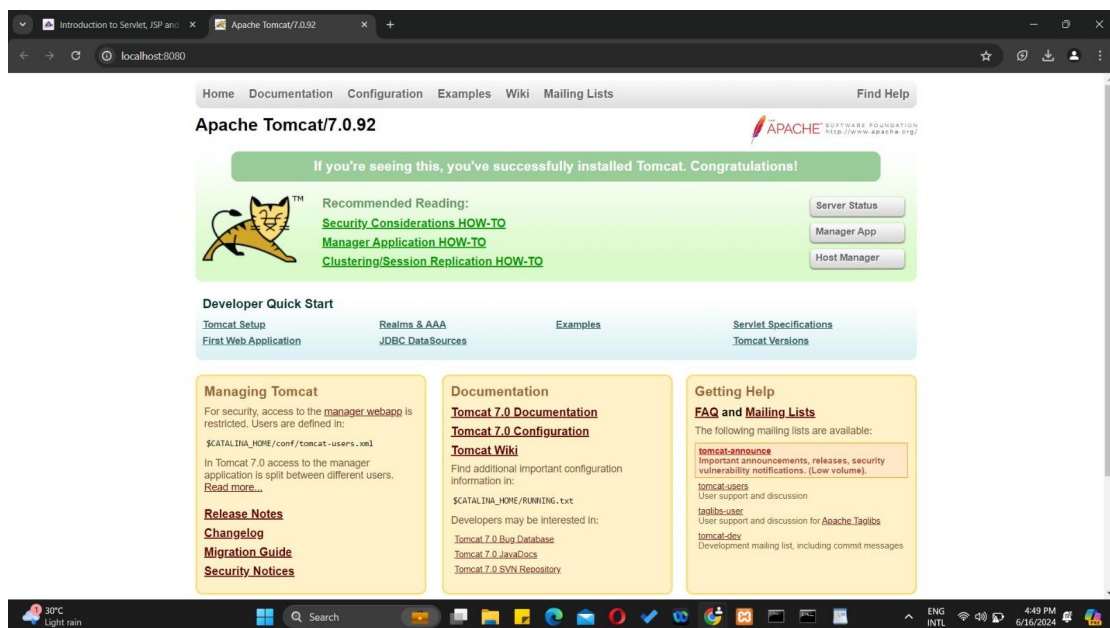
NAME :	SAIDATUL ATHIRAH BINTI MOHD SAIDI
MATRIC NO. :	S65302
COURSE :	BACHELOR OF COMPUTER SCIENCE (MOBILE COMPUTING) WITH HONORS
FACULTY :	FACULTY OF COMPUTER SCIENCE AND MATHEMATICS
LECTURER :	DR.MOHAMAD NOR BIN HASSAN

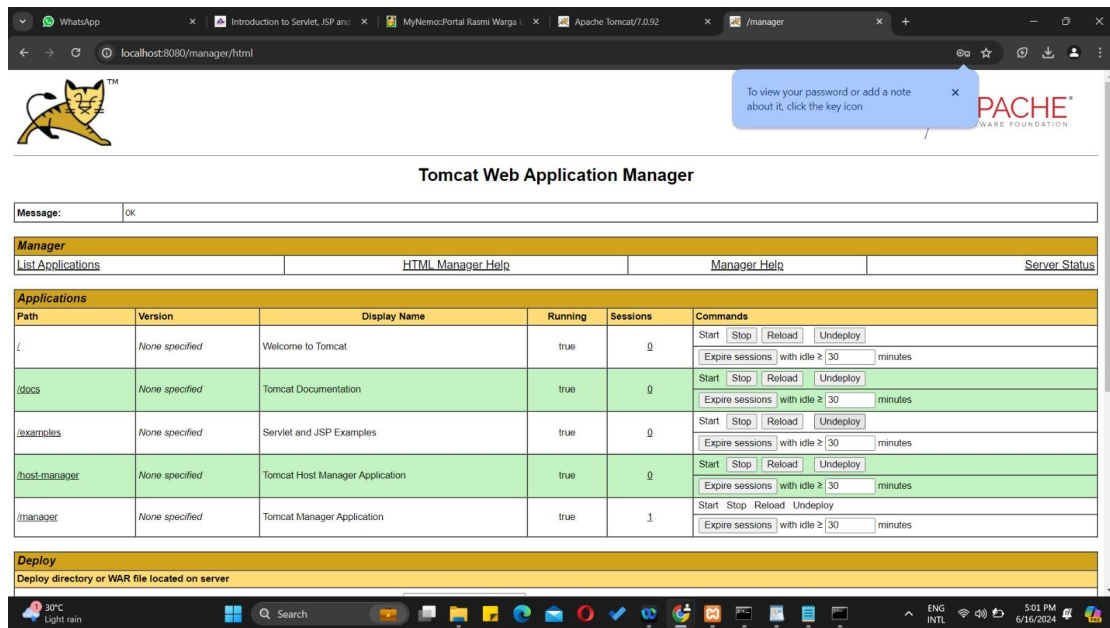
Task 2: Change the Default Root Password of MySQL Database





Task 3: Managing Apache Tomcat





Task 5: Linking Netbeans to Apache Tomcat and Writing a Simple Java Servlet

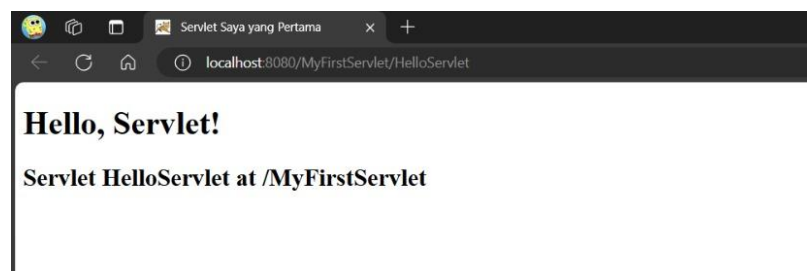
Code:

```

19  /**
20   * Processes requests for both HTTP GET and POST
21   * methods.
22   *
23   * @param request HttpServletRequest request
24   * @param response HttpServletResponse response
25   * @throws ServletException if a servlet-specific error occurs
26   * @throws IOException if an I/O error occurs
27   */
28
29  protected void processRequest(HttpServletRequest request, HttpServletResponse response)
30      throws ServletException, IOException {
31      response.setContentType("text/html;charset=UTF-8");
32      try (PrintWriter out = response.getWriter()) {
33          /* TODO output your page here. You may use following sample code. */
34          out.println("<doctype html>");
35          out.println("<html>");
36          out.println("<head>");
37          out.println("<title>Servlet Saya yang Pertama</title>");
38          out.println("</head>");
39          out.println("<body>");
40          out.println("<h1>Hello, Servlet!</h1>");
41          out.println("<h2>Servlet HelloServlet at " + request.getContextPath() + "</h2>");
42          out.println("</body>");
43          out.println("</html>");
44      }
45  }
46

```

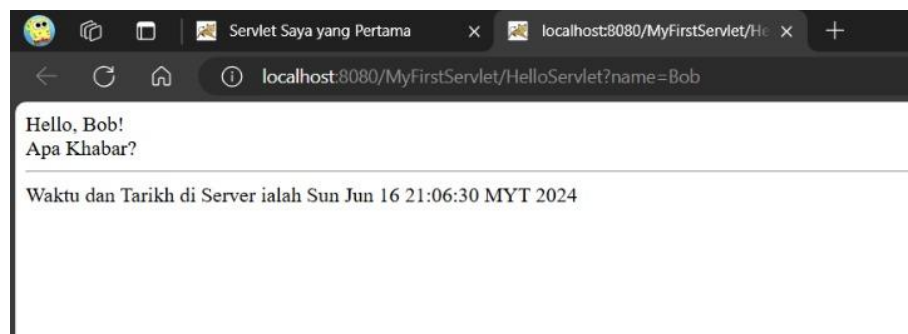
Output:



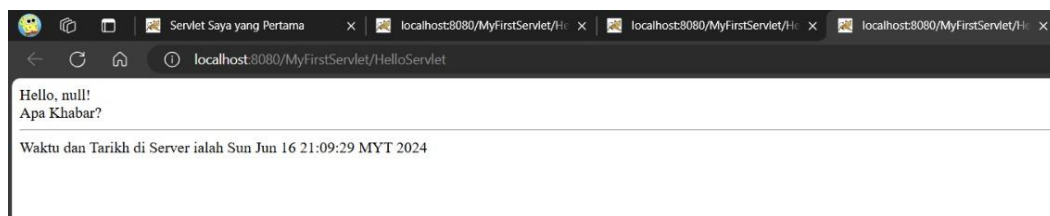
Code:

```
--jsp index.html x Student.java x studentservlet.java x StudentDao.java x registerTraining.jsp x Message1.jsp x Message.java x index.html x HelloServlet.java x
Source History
49 * Handles the HTTP <code>GET</code> method.
50 *
51 * @param request <code>HttpServletRequest</code> request
52 * @param response <code>HttpServletResponse</code> response
53 * @throws ServletException if a <code>Servlet</code>-specific error occurs
54 * @throws IOException if an I/O error occurs
55 */
56 @Override
57 protected void doGet(HttpServletRequest request, HttpServletResponse response)
58     throws ServletException, IOException {
59
60     response.setContentType("text/html");
61
62     PrintWriter out = response.getWriter();
63
64     String name = request.getParameter("name");
65
66     if (name == null)
67         processRequest(request, response);
68
69     out.println("<html><body>");
70
71     out.println("Hello, " + name + "! " + "<br>");
72     out.println("Apa Kabar? " + "<br>");
73     out.println("Waktu dan Tarikh di Server ialah " + new java.util.Date());
74     out.println("</html></body>");
75
76 }
77 }
```

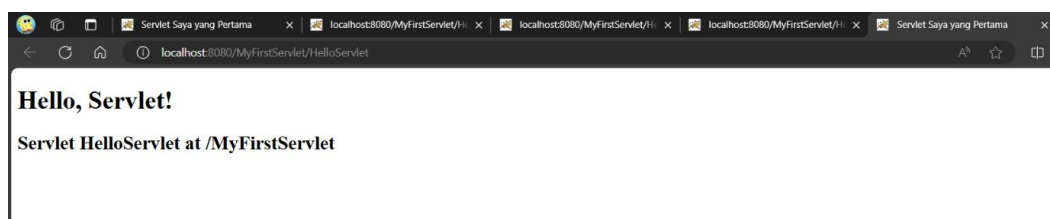
Output:



Output(without parameter):



Output(Avoid nulls in browser):



Task 6: Writing a Simple JSP Program



Reflection

1. What have you learned from this exercise?

How to run JSP file

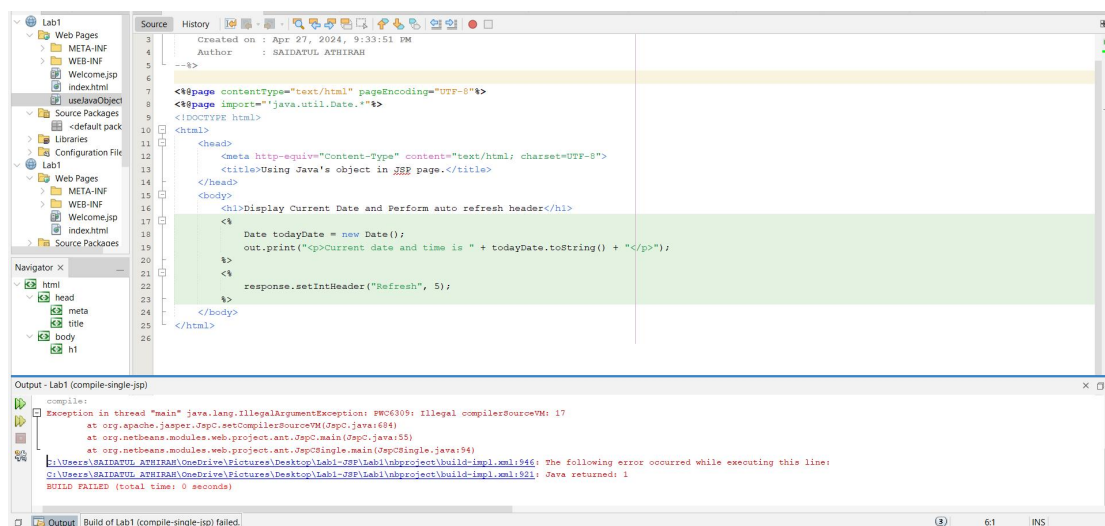
2. Explain the general concept of how the JSP's file work?

JSP files blend HTML with embedded Java code for creating dynamic web pages, processed on the server to generate interactive content for client browsers.

3. Based on your observation of the previous tasks (Task 3 and Task 4), what are the differences you can find between servlet and JSP?

Servlets handle requests programmatically, while JSP allows embedding Java in HTML for dynamic web content.

Task 7: Use Java Reference Datatype/Class Wrapper in JSP



Reflection

1. What have you learnt from this exercise?

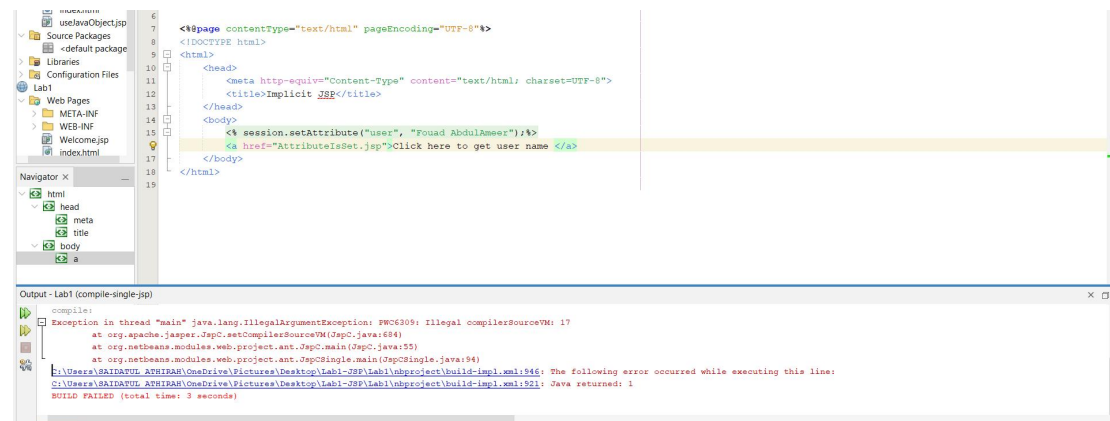
How to use java reference data type

A Java scriptlet is a block of Java code embedded within a JSP file, enclosed by <% and %> tags, used to execute dynamic server-side logic to generate content.

3. How to use Java code in your JSP's page?

<% ... %> for scriptlets, <%= ... %> for expressions, and <%! ... %> for declarations.

Task 8: Using JSP Implicit object in JSP page



Reflection

1. How do you want to submit specific information from one form to next form?

Use form submission with HTTP POST or GET methods to transfer specific information from one form to the next form.

2. What happened if the field name you specify in `request.getParameter("field_name")` in the second page is different from the field name you defined in the first page?

It will return null