

Integrating client and server side (First Application)

1. Servlets, JSP and mapping servlets (web.xml or annotation)

Servlets and JSP Basics: Servlets are Java classes that handle HTTP requests and responses. JSP (JavaServer Pages) is a technology for creating dynamic web content using HTML, Java code, and JSP tags.

Mapping Servlets: Servlets can be mapped in `web.xml` or using annotations.

Example using `web.xml`:

```
<!-- web.xml configuration -->
<servlet>
    <servlet-name>HelloServlet</servlet-name>
    <servlet-class>com.example.HelloServlet</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>HelloServlet</servlet-name>
    <url-pattern>/hello</url-pattern>
</servlet-mapping>
```

Example using Annotations:

```
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/hello")
public class HelloServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request,
        HttpServletResponse response) {
        response.setContentType("text/html");
        try (PrintWriter out = response.getWriter()) {
            out.println("<h1>Hello, World!</h1>");
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

2. HTTP Methods and Status Codes

HTTP Methods:

- **GET:** Retrieves data from the server.
- **POST:** Submits data to the server.
- **PUT:** Updates data on the server.
- **DELETE:** Deletes data from the server.

HTTP Status Codes:

- **200 OK:** Request succeeded.
- **404 Not Found:** Resource not found.
- **500 Internal Server Error:** Server encountered an error.

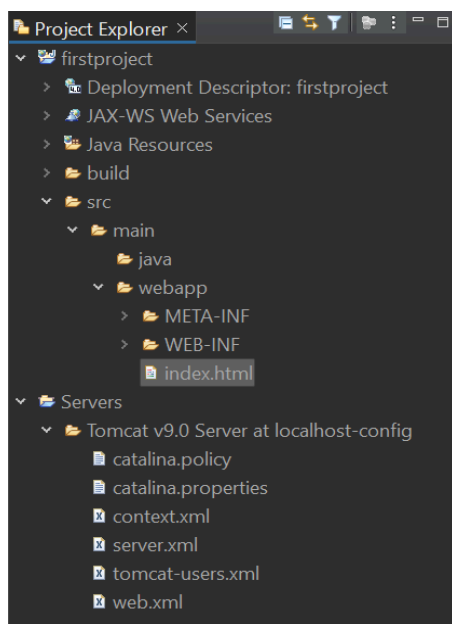
3. Create a Simple Web Application (Hello World)

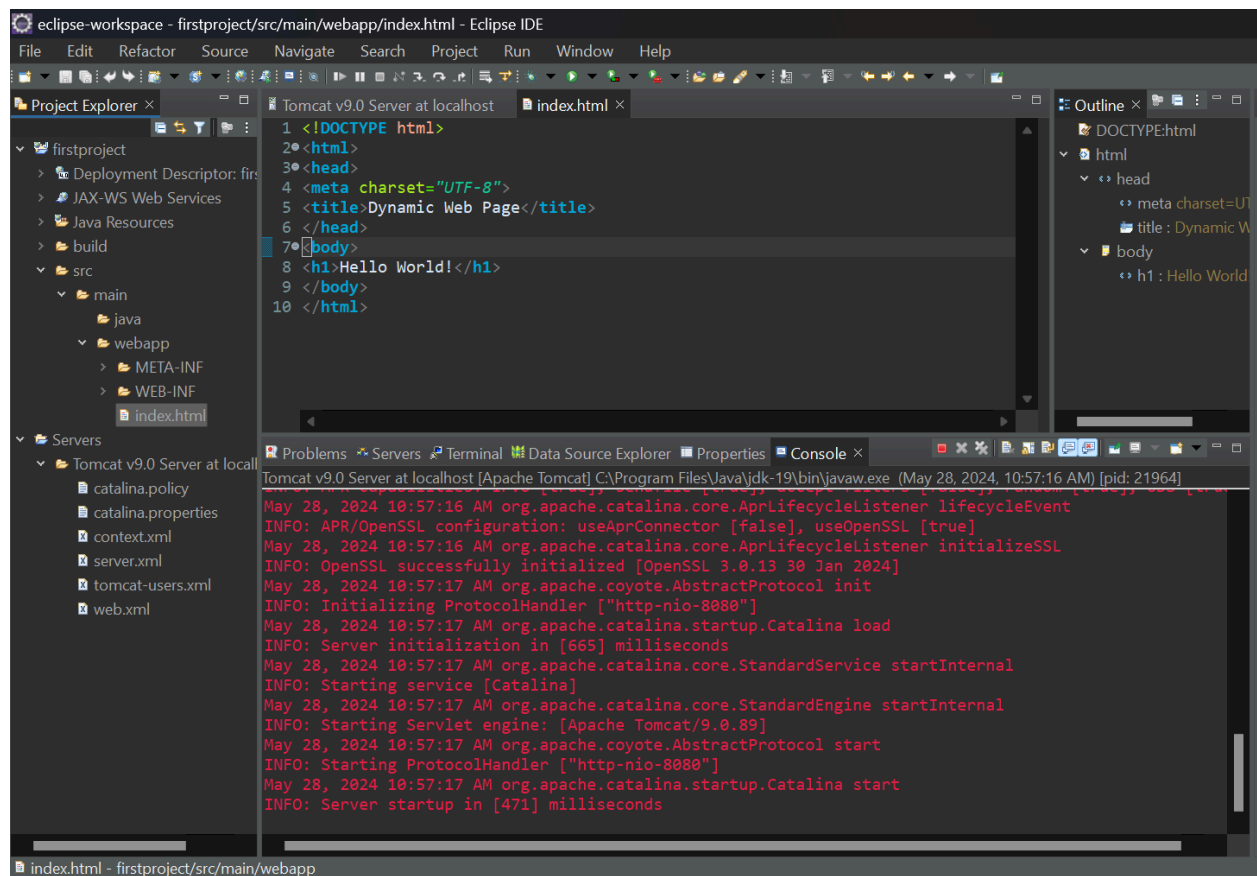
Create a simple web application (dynamic web project in eclipse) which will display hello world. Add tomcat server to host the application. Familiarize with the structure of dynamic projects.

Setting Up a Dynamic Web Project in Eclipse:

1. Open Eclipse and create a new Dynamic Web Project.
2. Configure the project to use Apache Tomcat.
3. Create a new Servlet and map it to a URL pattern.

Project Structure:





Hello World!



4. Create a simple web application with login and registration form using JSP and Servlet. On successful login it must welcome the user. If the user is new he/she needs to register. Add a logout button on welcome page. On logout, the user should be directed again to the login page. If user enters incorrect password a proper message should be displayed and login form must be displayed again.

5. The required validations are to be added using JavaScript

DB:

```
Command Prompt - mysql -u X
C:\Users\Khushi Mahamunkar>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 8.0.37 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

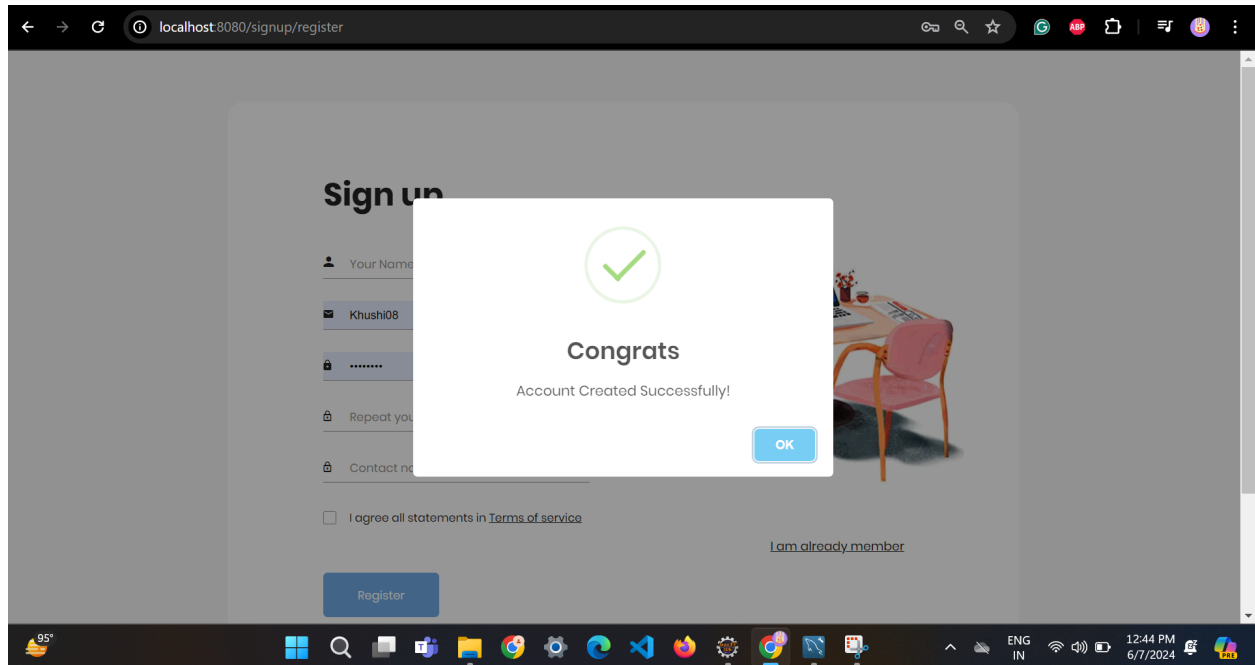
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database registration
-> ^C
mysql> create database registration;
Query OK, 1 row affected (0.01 sec)

mysql> use registration;
Database changed
mysql> desc users;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id     | int  | NO   | PRI | NULL    | auto_increment |
| uname  | varchar(50) | YES |     | NULL    |               |
| upwd   | varchar(50) | YES |     | NULL    |               |
| uemail | varchar(50) | YES |     | NULL    |               |
| umobile | varchar(20) | YES |     | NULL    |               |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

SIGNUP



DATABASE:

The screenshot displays a database management interface. The top section shows a SQL query editor with the following code:

```
1 • use registration;
2 • create table users(id int primary key auto_increment, uname varchar(50), upwd varchar(50), uemail varchar(50)
3 • desc users;
4
5 • Select * from users;
```

Below the editor is a 'Result Grid' showing the output of the queries. The first three queries (1-3) are successful. The fourth query (4) returns 2 rows, and the fifth query (5) returns 3 rows. The 'Result Grid' table is as follows:

| | id | uname | upwd | uemail | umobile |
|---|------|-------------------|------|--------------------|------------|
| 1 | 1 | Khushi Mahamunkar | 123 | Khushi08@gmail.com | 9137490103 |
| 2 | 2 | k1 | 321 | Khu@gmail.com | 1234561234 |
| 3 | 3 | Khushi M | 321 | Khushi8@gmail.com | 1234543212 |
| 4 | NULL | NULL | NULL | NULL | NULL |


The bottom section shows the 'Output' tab with a table of execution results:

| # | Time | Action | Message | Duration / Fetch |
|---|----------|-----------------------------------|-------------------|-----------------------|
| 1 | 12:31:44 | use registration | 0 row(s) affected | 0.015 sec |
| 2 | 12:31:55 | desc users | 5 row(s) returned | 0.000 sec / 0.000 sec |
| 3 | 12:32:01 | desc users | 5 row(s) returned | 0.016 sec / 0.000 sec |
| 4 | 12:32:03 | Select * from users LIMIT 0, 1000 | 2 row(s) returned | 0.000 sec / 0.000 sec |
| 5 | 12:44:43 | Select * from users LIMIT 0, 1000 | 3 row(s) returned | 0.000 sec / 0.000 sec |

On the right side, there is a 'SQL Additions' panel with a message: 'Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.'

Sign IN

localhost:8080/signup/login.jsp



Sign in

☐ Remember me

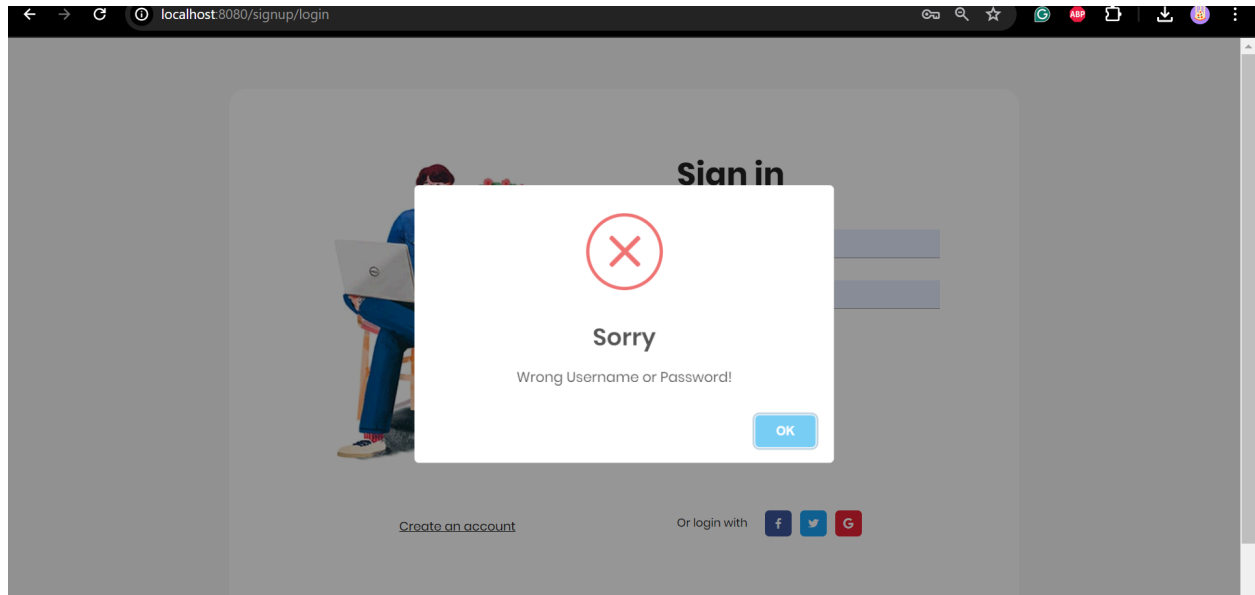
[Log in](#)

[Create an account](#)

Or login with [f](#) [t](#) [G](#)



SERVER VALIDATION



CLIENT VALIDATION:

