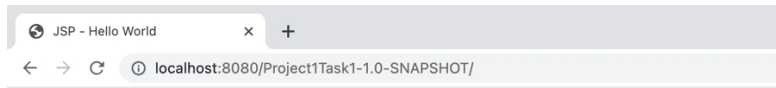
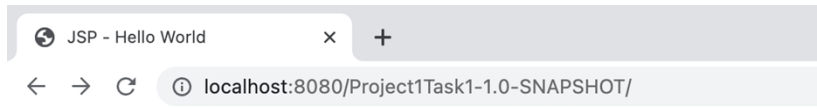


Muyu Tong (muyut)

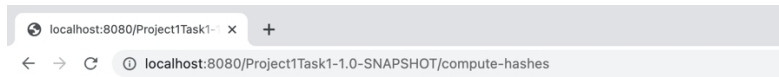
Task 1



The string to be hashed: ☐ MD5 ☒ SHA-256



The string to be hashed: ☒ MD5 ☐ SHA-256



hashed string: Hello World
hex output: B10A8DB164E0754105B7A99BE72E3FE5
base64 output: 1B2M2Y8AsgTpgAmY7PhCfg==

```
@WebServlet(name = "computeHashes", value = "/compute-hashes")
public class ComputeHashes extends HttpServlet {
    String message;

    public void init() {}

    //https://stackoverflow.com/questions/32374375/
    //how-to-read-value-of-a-input-field-in-jsp-form-through-a-servlet-using-id-of-tha
    public void doPost(HttpServletRequest request, HttpServletResponse response) throws IOException {
        //
        response.setContentType("text/html");

        MessageDigest md;

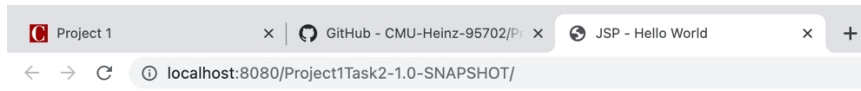
        String s = request.getParameter(s: "s"); //input string
        String hash = request.getParameter(s: "hash"); //hash method

        String hexText;
        String base64Text;
        try { //hash the string using the relevant method
            md = MessageDigest.getInstance(hash);
            md.update(s.getBytes(StandardCharsets.UTF_8));

            hexText = DatatypeConverter.printHexBinary(md.digest());
            base64Text = DatatypeConverter.printBase64Binary(md.digest());
            message = "hashed string: " + s + "<br> hex output: " + hexText + "<br> base64 output: " + base64Text;
        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
        }

        // print the hash result
        PrintWriter out = response.getWriter();
        out.println("<html><body>");
        out.println("<h1>" + message + "</h1>");
        out.println("</body></html>");
    }
}
```

Task 2



Dog Finder

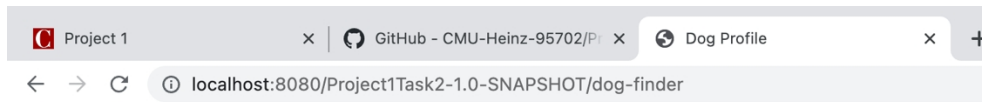
Created By Muyu Tong

Dog Breed

Choose a dog breed

submit

- Borzoi
- Boxer
- Chihuahua
- ✓ Collie
- Dachshund
- Dalmatian
- Maltese
- Otterhound
- Poodle
- Rottweiler
- Saluki
- Whippet



Dog: Collie

Friendly: 5 Stars

Height: Height: 22 to 26 inches all at the shoulder

Weight: Weight: 50 to 70 pounds

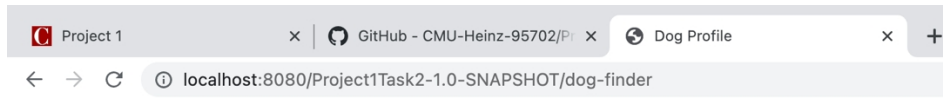
Lifespan: Life Span: 10 to 14 years

Credit: <https://dogtime.com/dog-breeds/profiles>



Credit: <https://dog.ceo/dog-api/>

Continue



Dog: Whippet

Friendly: 5 Stars

Height: Height: 18 to 22 inches tall at the shoulder

Weight: Weight: 18 to 48 pounds

Lifespan: Life Span: 12 to 15 years

Credit: <https://dogtime.com/dog-breeds/profiles>



Credit: <https://dog.ceo/dog-api/>

[Continue](#)

Code for dog's information

Ref: <https://jsoup.org/>

```
public static Dog scrapInfo(String breed) throws IOException {
    breed = breed.toLowerCase();
    int friendly = 0;
    int intelligence = 0;
    String url = Dog.infoCredit+"/"+breed+"/#/side/1"; //get url of the breed
    Document doc = Jsoup.connect(url).get();
    Elements starsEle = doc.getElementsByClass(className: "characteristic-title");
    String className;
    for (Element element: starsEle) {
        if (element.text().equals("All Around Friendliness")) { //scrape friendliness
            className = element.siblingElements().get(0).child(0).className();
            friendly = className.charAt(className.length()-1) - 48;
        } else if (element.text().equals("Intelligence")) { //scrape intelligence
            className = element.siblingElements().get(0).child(0).className();
            intelligence = className.charAt(className.length()-1) - 48;
        }
    }
    //scrape weight
    Element weightEle = doc.getElementsByClass(className: "vital-stat-title vital-stat-weight").get(0).parent();
    String weight = weightEle.text();

    //scrape height
    Element heightEle = doc.getElementsByClass(className: "vital-stat-title vital-stat-height").get(0).parent();
    String height = heightEle.text();

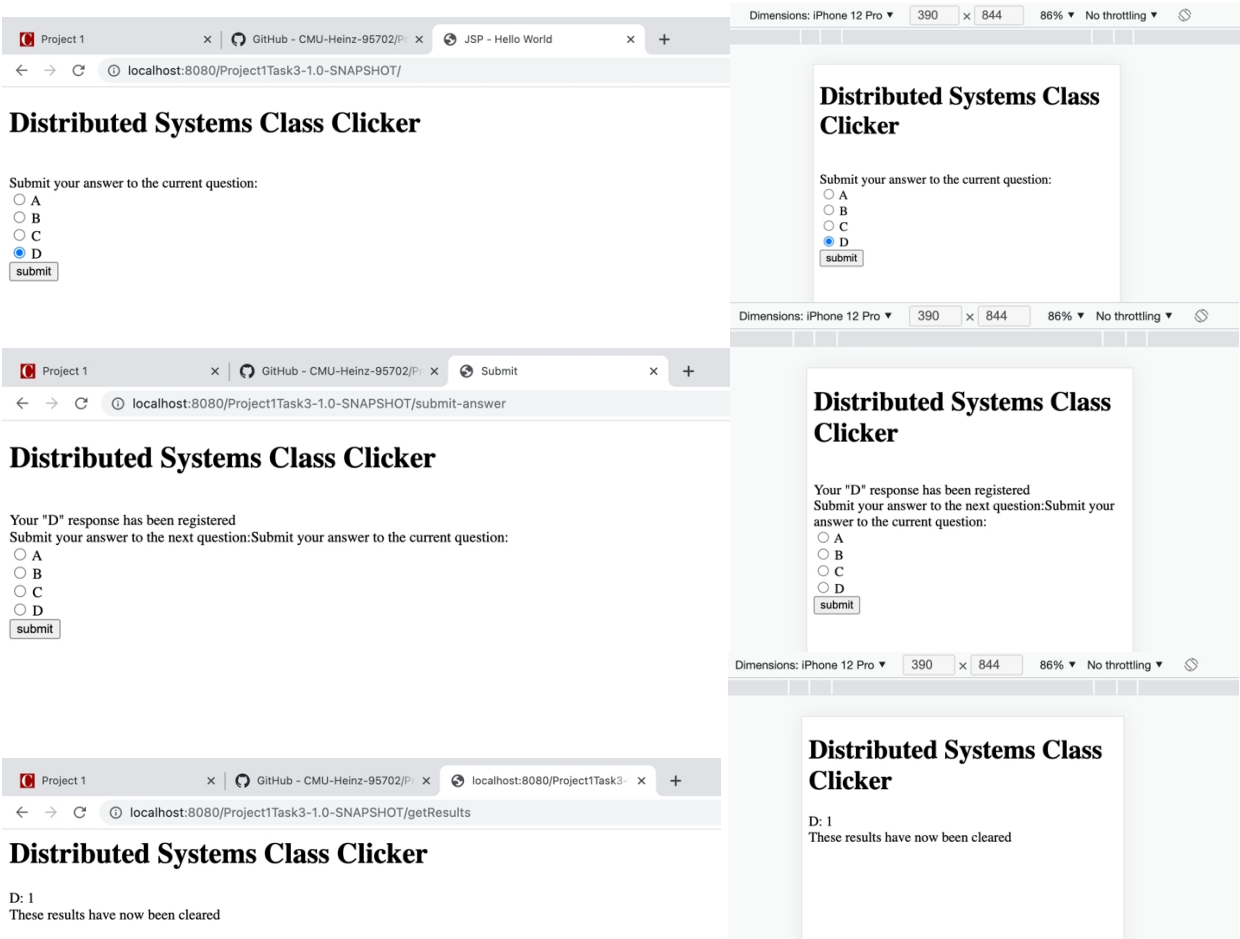
    //scrape lifespan
    Element lifespanEle = doc.getElementsByClass(className: "vital-stat-title vital-stat-lifespan").get(0).parent();
    String lifespan = lifespanEle.text();
    Dog dog = new Dog(breed, friendly, intelligence, height, weight, lifespan);
    return dog;
}
```

Code for the image

```
//https://www.tutorialspoint.com/gson/gson_quick_guide.htm
//get the url of the random image
public static String getImage(String breed) throws IOException {
    String url = "https://dog.ceo/api/breed/"+breed+"/images/random";
    URLConnection connection = new URL(url).openConnection();
    InputStream inputStream = connection.getInputStream();
    Scanner scanner = new Scanner(inputStream);
    String responseBody = scanner.useDelimiter("\\A").next();
    GsonBuilder gsonBuilder = new GsonBuilder();
    gsonBuilder.setPrettyPrinting();
    Gson gson = gsonBuilder.create();
    ImageResponse re = gson.fromJson(responseBody, ImageResponse.class);

    return re.getMessage();
}
```

Task 3



Produce the output page

```
//submit the answer
//https://www.geeksforgeeks.org/getattribute-passing-data-from-server-to-jsp/
public void doPost(HttpServletRequest request, HttpServletResponse response) throws IOException {
    response.setContentType("text/html");
    String ans = request.getParameter( s: "ans");
    if (ans.equals("A")) {
        Result.numA++;
    } else if (ans.equals("B")) {
        Result.numB++;
    } else if (ans.equals("C")) {
        Result.numC++;
    } else {
        Result.numD++;
    }
    String message = "Your \""+ans+"\" response has been registered";
    String question = "Submit your answer to the next question:";
    request.setAttribute( s: "message", message);
    request.setAttribute( s: "question", question);
    try {
        //forward to the output page
        request.getRequestDispatcher( s: "submit.jsp").forward(request, response);
    } catch (ServletException e) {
        e.printStackTrace();
    }
}
```

Output page

```
<body>
<h1>Distributed Systems Class Clicker
</h1>
<br/>
<div><%=request.getAttribute("message")%></div>
<form action="submit-answer" method="post">
    <div><%=request.getAttribute("question")%>Submit your answer to the current question: </div>
    <input type="radio" id="A" name="ans" value="A">
    <label for="A">A</label><br>
    <input type="radio" id="B" name="ans" value="B">
    <label for="B">B</label><br>
    <input type="radio" id="C" name="ans" value="C">
    <label for="C">C</label><br>
    <input type="radio" id="D" name="ans" value="D">
    <label for="D">D</label><br>
    <input type="submit" value="submit">
</form>
</body>
```

Produce the result page

```
//produce the result page
public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    out.println("<html><body>");
    out.println("<h1>Distributed Systems Class Clicker</h1>");
    if (Result.numA + Result.numB + Result.numC + Result.numD == 0) {
        out.println("There are currently no answers");
    } else {
        if (Result.numA != 0) {
            out.println("<div>A: "+Result.numA+"</div>");
        }
        if (Result.numB != 0) {
            out.println("<div>B: "+Result.numB+"</div>");
        }
        if (Result.numC != 0) {
            out.println("<div>C: "+Result.numC+"</div>");
        }
        if (Result.numD != 0) {
            out.println("<div>D: "+Result.numD+"</div>");
        }
        out.println("<div>These results have now been cleared</div>");
        Result.numA = 0;
        Result.numB = 0;
        Result.numC = 0;
        Result.numD = 0;
    }

    out.println("</body></html>");
}
```