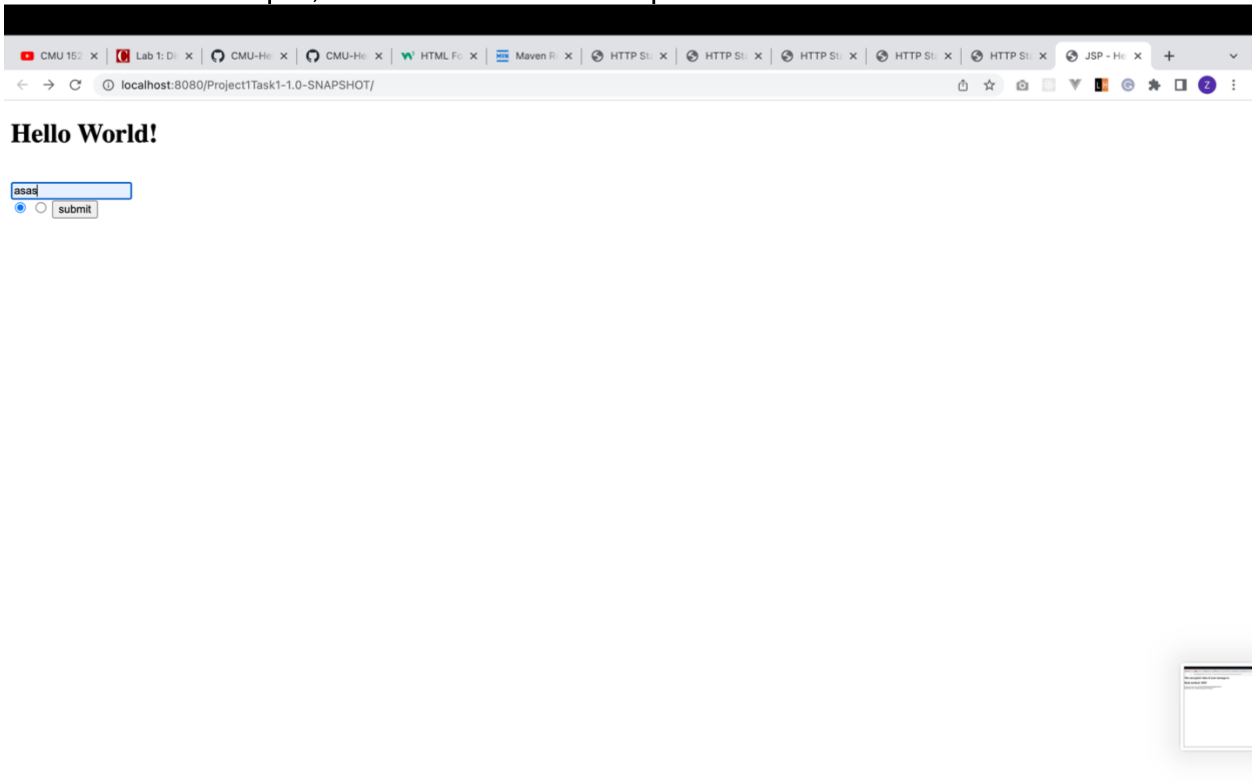
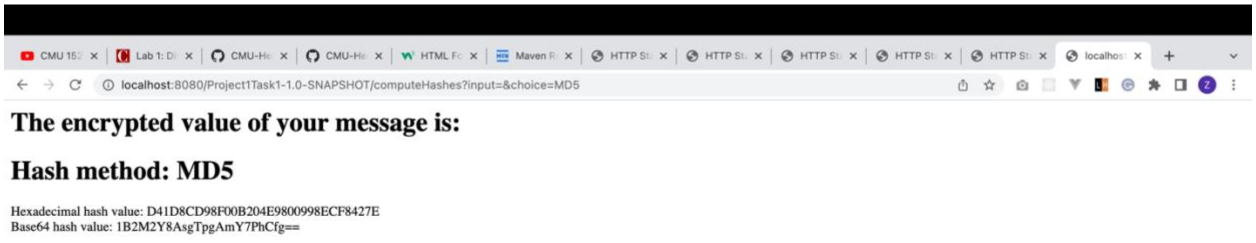
Task1:
Screen Shots of input, MD5 and SHA-256 outputs


localhost:8080/Project1Task1-1.0-SNAPSHOT/

Hello World!

asas

submit

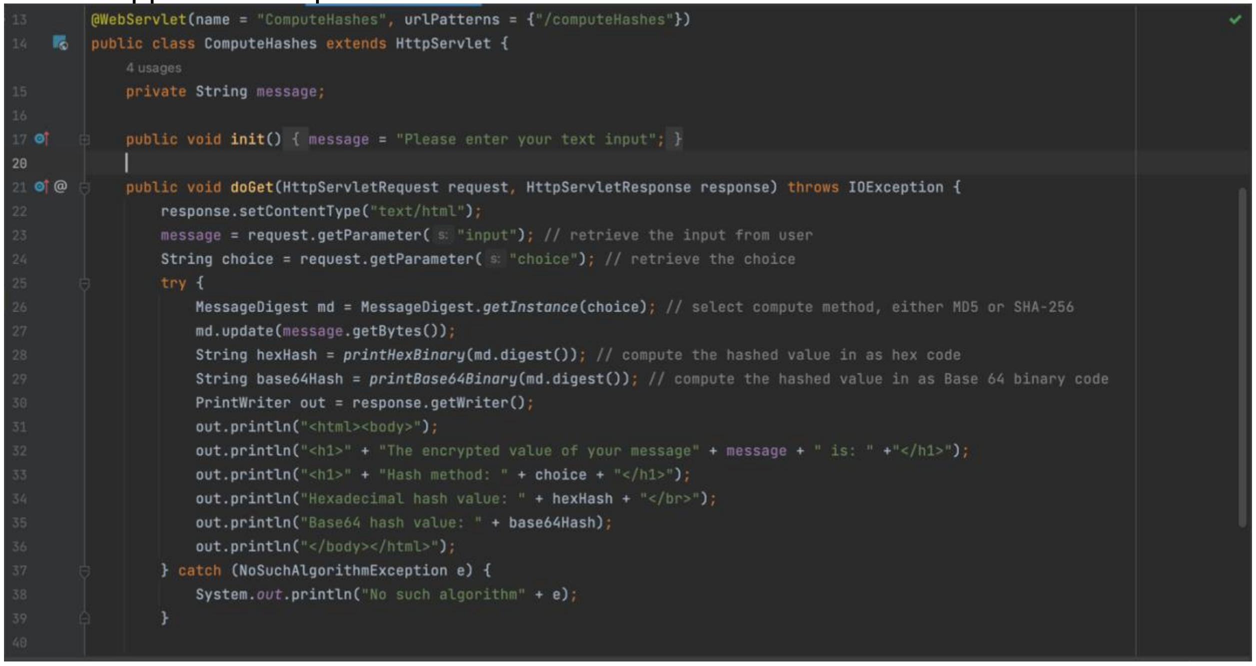

localhost:8080/Project1Task1-1.0-SNAPSHOT/computeHashes?input=&choice=MD5

The encrypted value of your message is:

Hash method: MD5

Hexadecimal hash value: D41D8CD98F00B204E9800998ECF8427E
Base64 hash value: 1B2M2Y8AsgTpgAmY7PhCfg==

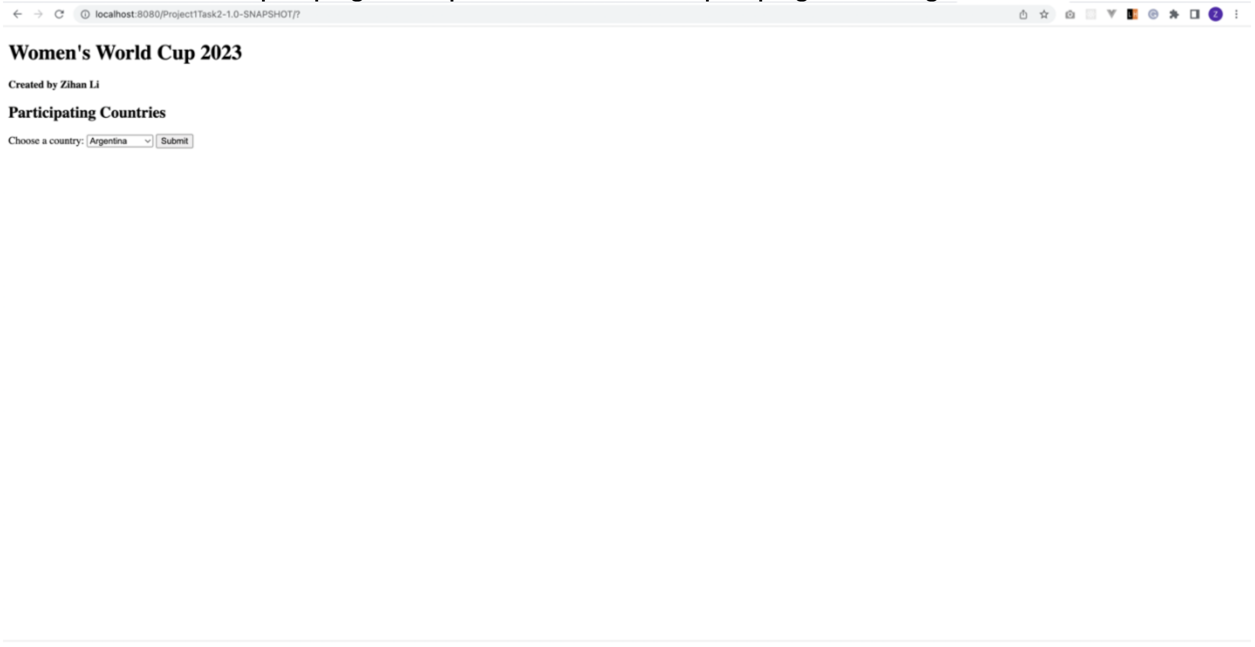
Code snippets of computation methods



```java
@WebServlet(name = "ComputeHashes", urlPatterns = {"/computeHashes"})
public class ComputeHashes extends HttpServlet {
    4 usages
    private String message;

    public void init() { message = "Please enter your text input"; }

    public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {
        response.setContentType("text/html");
        message = request.getParameter("input"); // retrieve the input from user
        String choice = request.getParameter("choice"); // retrieve the choice
        try {
            MessageDigest md = MessageDigest.getInstance(choice); // select compute method, either MD5 or SHA-256
            md.update(message.getBytes());
            String hexHash = printHexBinary(md.digest()); // compute the hashed value in as hex code
            String base64Hash = printBase64Binary(md.digest()); // compute the hashed value in as Base 64 binary code
            PrintWriter out = response.getWriter();
            out.println("<html><body>");
            out.println("<h1>" + "The encrypted value of your message" + message + " is: " +"</h1>");
            out.println("<h1>" + "Hash method: " + choice + "</h1>");
            out.println("Hexadecimal hash value: " + hexHash + "</br>");
            out.println("Base64 hash value: " + base64Hash);
            out.println("</body></html>");
        } catch (NoSuchAlgorithmException e) {
            System.out.println("No such algorithm" + e);
        }

```

Task2:

Screen shots of input page, drop-down menu, output page for England and South Africa



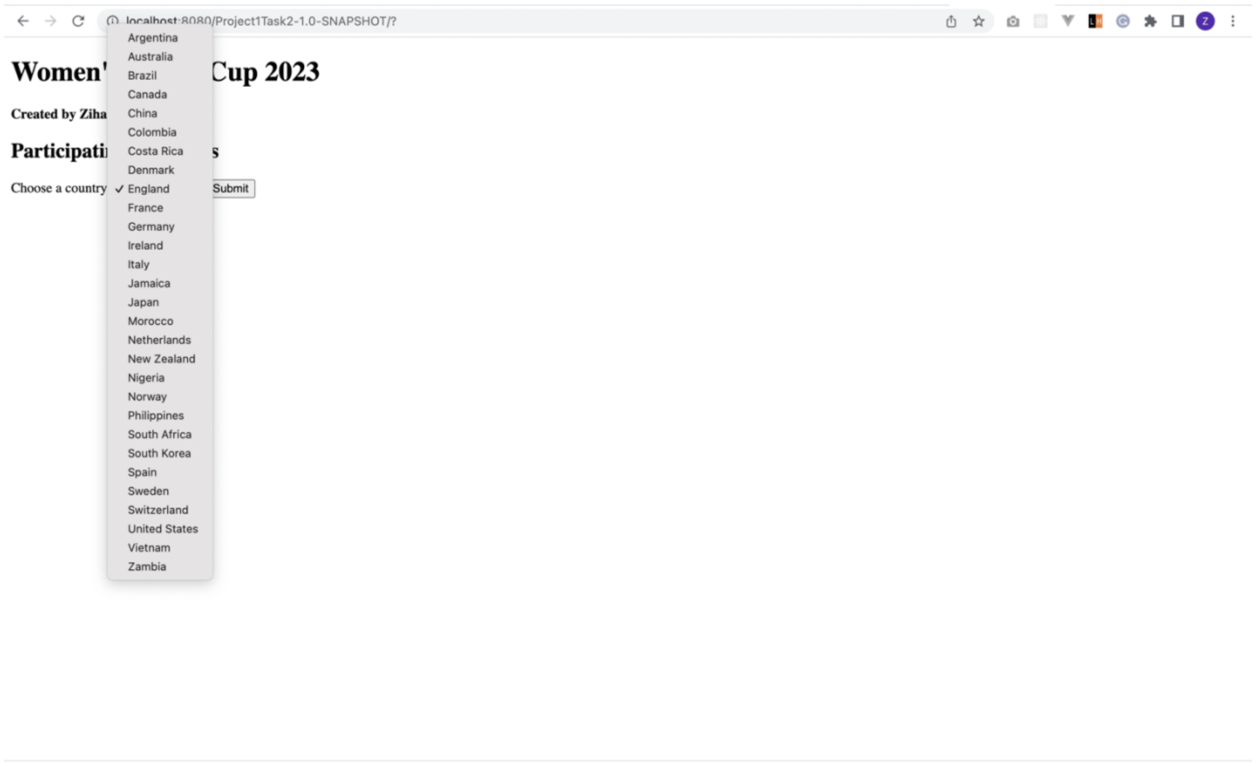
localhost:8080/Project1Task2-1.0-SNAPSHOT/?

Women's World Cup 2023

Created by Zihan Li

Participating Countries

Choose a country: Argentina Submit

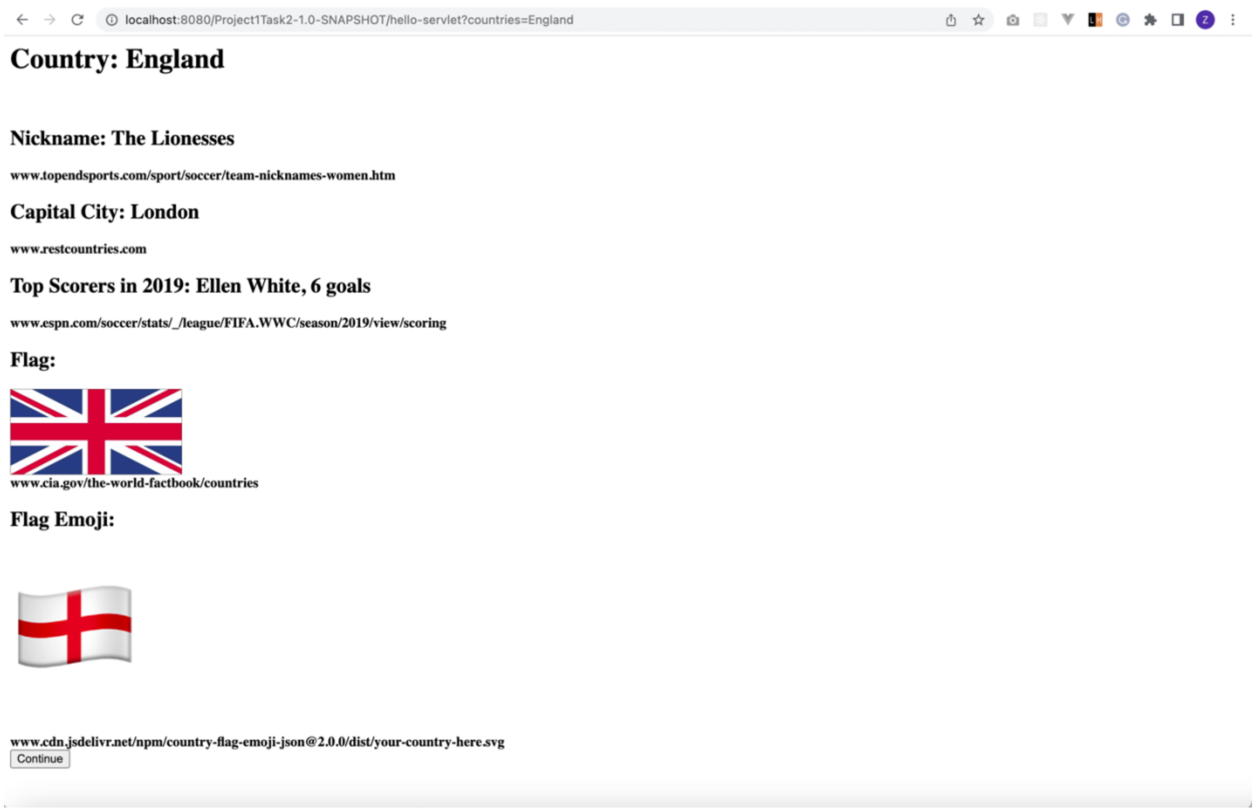

localhost:8080/Project1Task2-1.0-SNAPSHOT/?

Women' Cup 2023

Created by Ziha

Participati s

Choose a country



localhost:8080/Project1Task2-1.0-SNAPSHOT/hello-servlet?countries=England

Country: England

Nickname: The Lionesses

www.topendsports.com/sport/soccer/team-nicknames-women.htm

Capital City: London

www.restcountries.com

Top Scorers in 2019: Ellen White, 6 goals

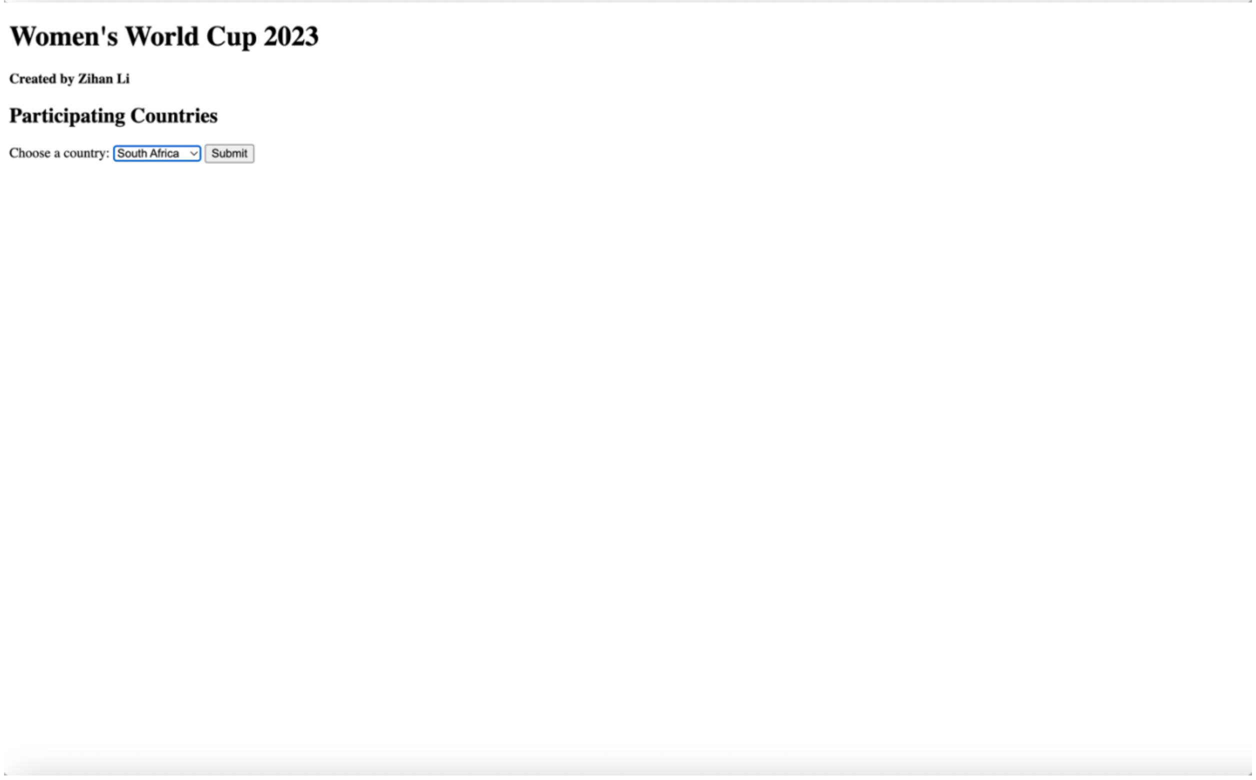
www.espn.com/soccer/stats/_/league/FIFA.WWC/season/2019/view/scoring

Flag:

www.cia.gov/the-world-factbook/countries

Flag Emoji:

www.cdn.jsdelivr.net/npm/country-flag-emoji-json@2.0.0/dist/your-country-here.svg

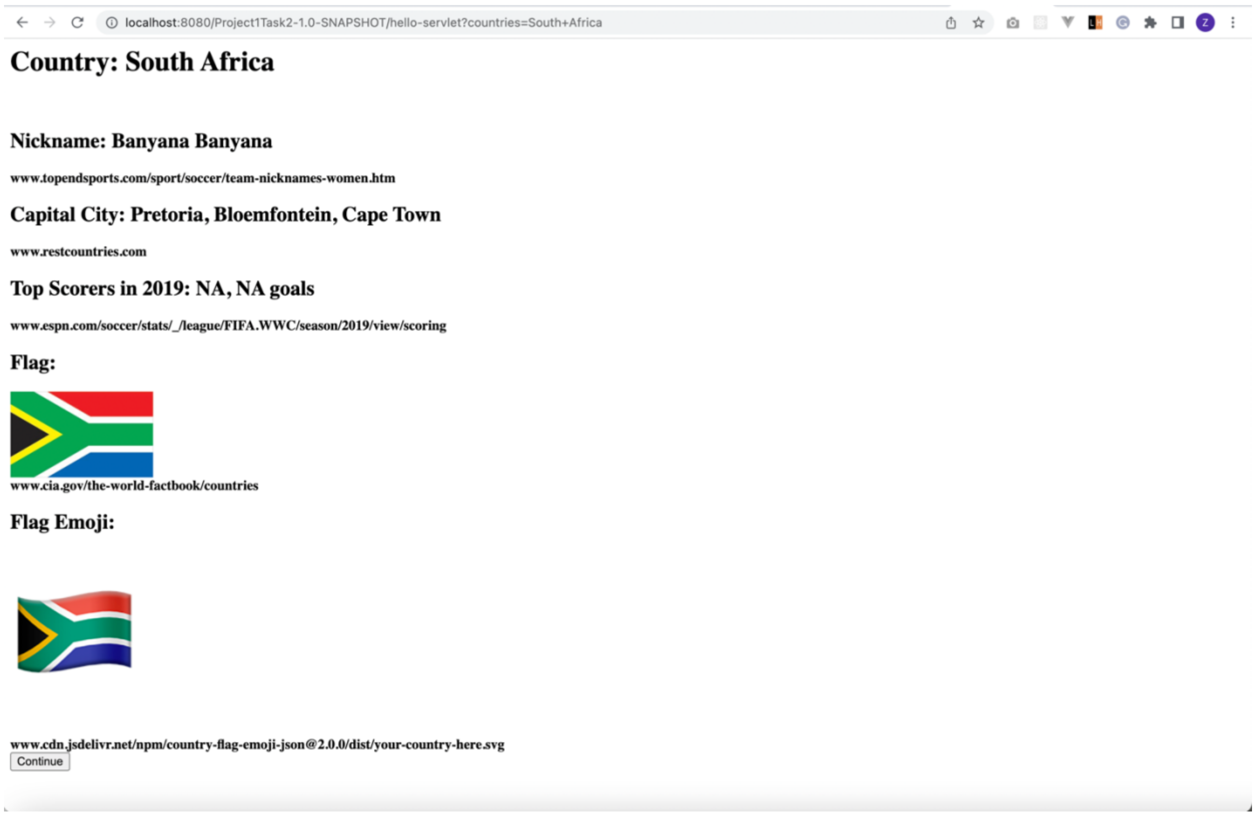


Women's World Cup 2023

Created by Zihan Li

Participating Countries

Choose a country: South Africa


localhost:8080/Project1Task2-1.0-SNAPSHOT/hello-servlet?countries=South+Africa

Country: South Africa

Nickname: Banyana Banyana

www.topendsports.com/sport/soccer/team-nicknames-women.htm

Capital City: Pretoria, Bloemfontein, Cape Town

www.restcountries.com

Top Scorers in 2019: NA, NA goals

www.espn.com/soccer/stats/_/league/FIFA.WWC/season/2019/view/scoring

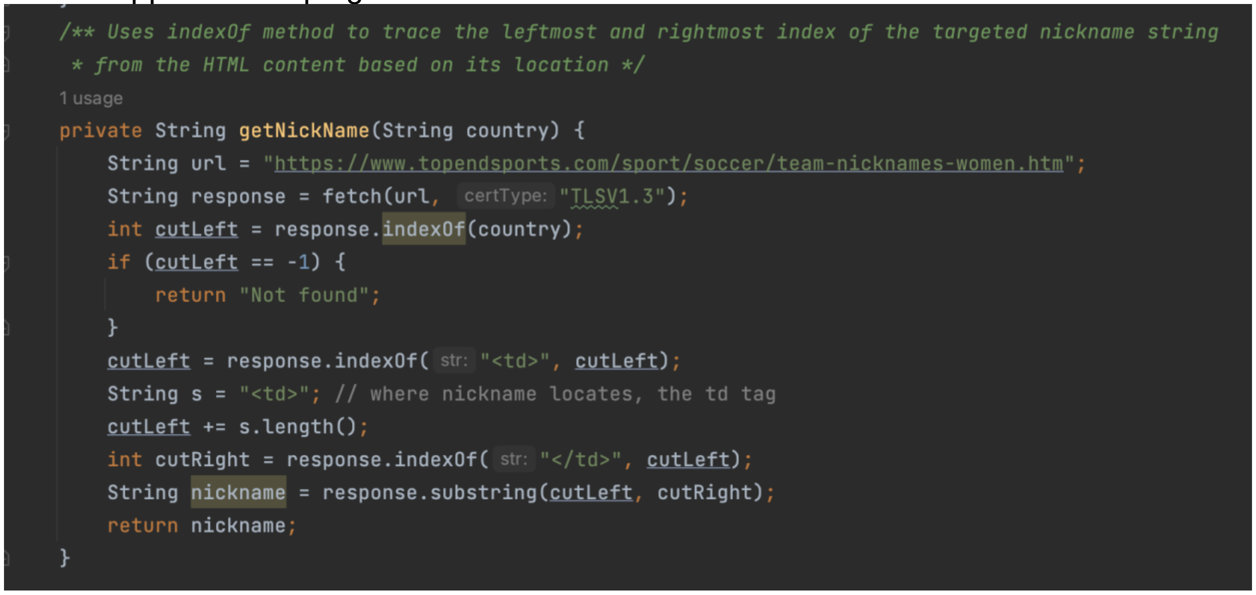
Flag:

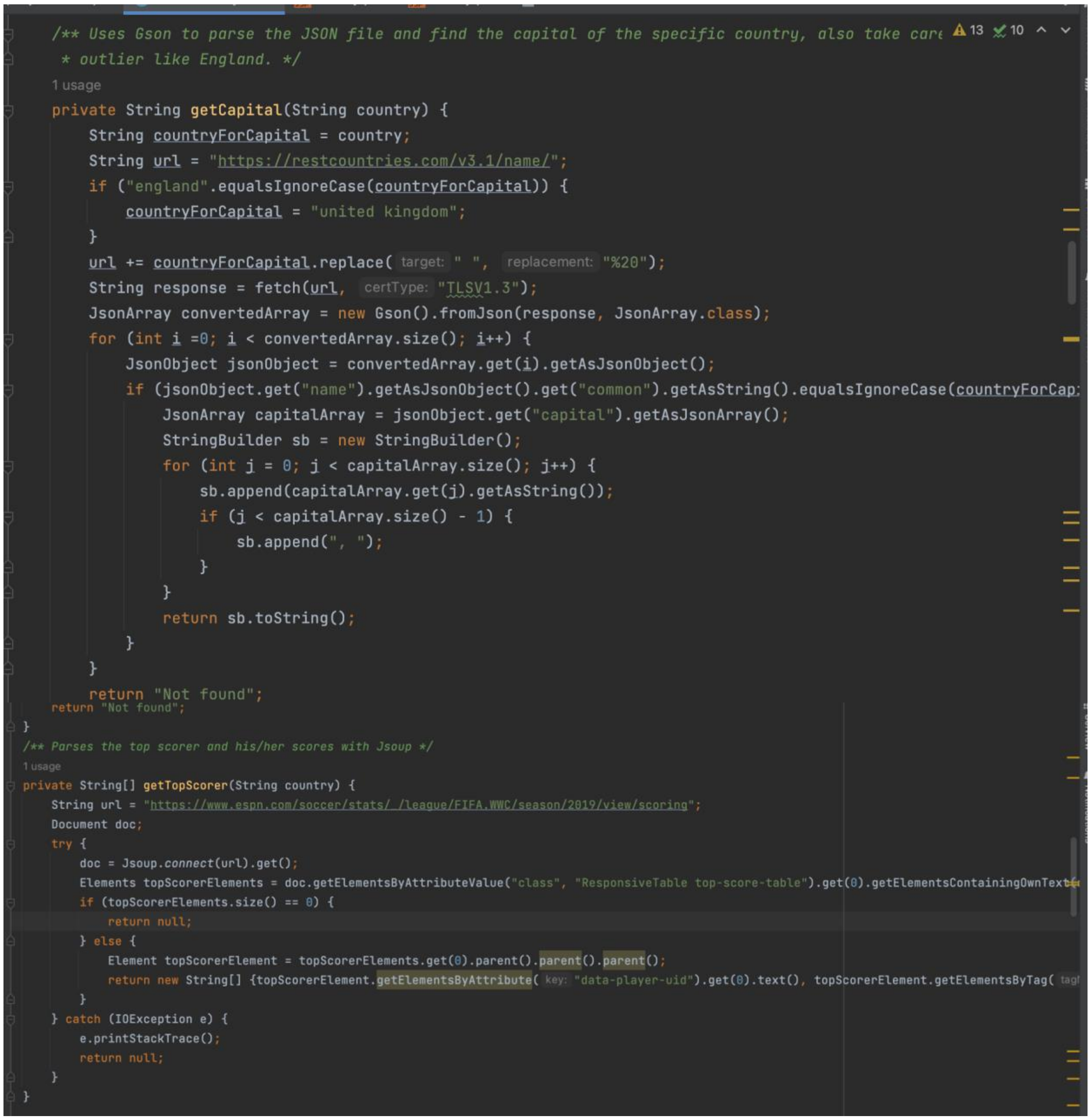
www.cia.gov/the-world-factbook/countries

Flag Emoji:

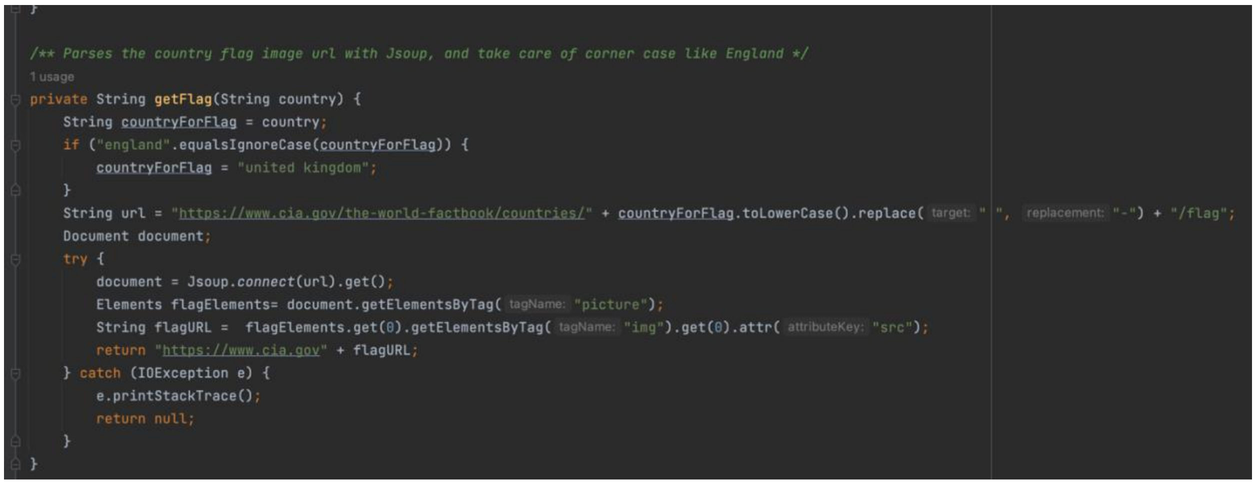
www.cdn.jsdelivr.net/npm/country-flag-emoji-json@2.0.0/dist/your-country-here.svg

Code snippets of scaping methods

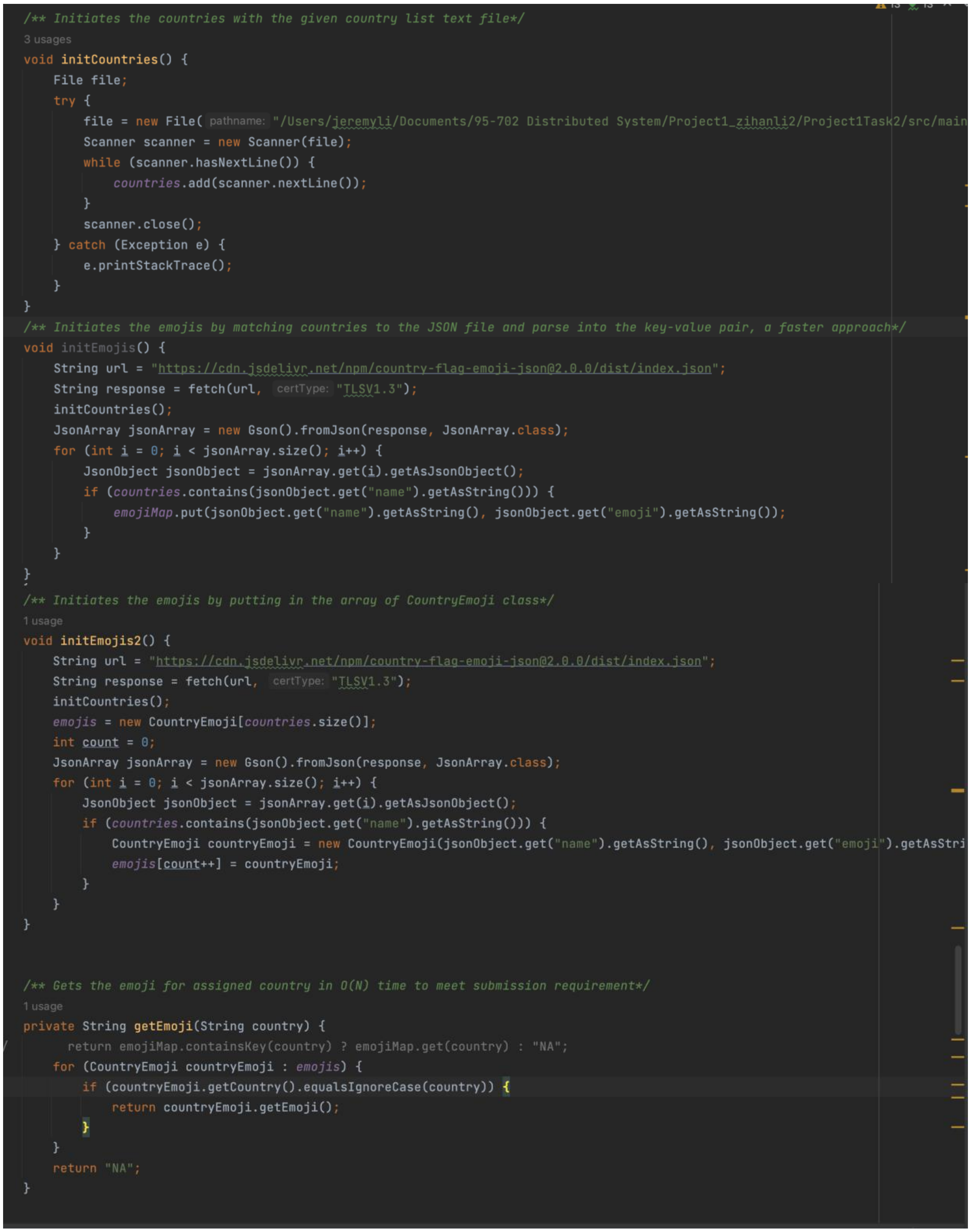

```
/** Uses indexOf method to trace the leftmost and rightmost index of the targeted nickname string
 * from the HTML content based on its location */
1 usage
private String getNickName(String country) {
    String url = "https://www.topendsports.com/sport/soccer/team-nicknames-women.htm";
    String response = fetch(url, certType: "TLSV1.3");
    int cutLeft = response.indexOf(country);
    if (cutLeft == -1) {
        return "Not found";
    }
    cutLeft = response.indexOf( str: "<td>", cutLeft);
    String s = "<td>"; // where nickname locates, the td tag
    cutLeft += s.length();
    int cutRight = response.indexOf( str: "</td>", cutLeft);
    String nickname = response.substring(cutLeft, cutRight);
    return nickname;
}
```



```
/** Uses Gson to parse the JSON file and find the capital of the specific country, also take care
 * outlier like England. */
1 usage
private String getCapital(String country) {
    String countryForCapital = country;
    String url = "https://restcountries.com/v3.1/name/";
    if ("england".equalsIgnoreCase(countryForCapital)) {
        countryForCapital = "united kingdom";
    }
    url += countryForCapital.replace( target: " ", replacement: "%20");
    String response = fetch(url, certType: "TLSV1.3");
    JsonArray convertedArray = new Gson().fromJson(response, JsonArray.class);
    for (int i =0; i < convertedArray.size(); i++) {
        JsonObject jsonObject = convertedArray.get(i).getAsJsonObject();
        if (jsonObject.get("name").getAsJsonObject().get("common").getAsString().equalsIgnoreCase(countryForCap
            JsonArray capitalArray = jsonObject.get("capital").getAsJsonArray();
            StringBuilder sb = new StringBuilder();
            for (int j = 0; j < capitalArray.size(); j++) {
                sb.append(capitalArray.get(j).getAsString());
                if (j < capitalArray.size() - 1) {
                    sb.append(", ");
                }
            }
            return sb.toString();
        }
    }
    return "Not found";
    return "Not found";
}
/** Parses the top scorer and his/her scores with Jsoup */
1 usage
private String[] getTopScorer(String country) {
    String url = "https://www.espn.com/soccer/stats/_/league/FIFA.WWC/season/2019/view/scoring";
    Document doc;
    try {
        doc = Jsoup.connect(url).get();
        Elements topScorerElements = doc.getElementsByAttributeValue("class", "ResponsiveTable top-score-table").get(0).getElementsContainingOwnText
        if (topScorerElements.size() == 0) {
            return null;
        } else {
            Element topScorerElement = topScorerElements.get(0).parent().parent().parent();
            return new String[] {topScorerElement.getElementsByAttribute( key: "data-player-uid").get(0).text(), topScorerElement.getElementsByTag( tag
        }
    } catch (IOException e) {
        e.printStackTrace();
        return null;
    }
}
```
]



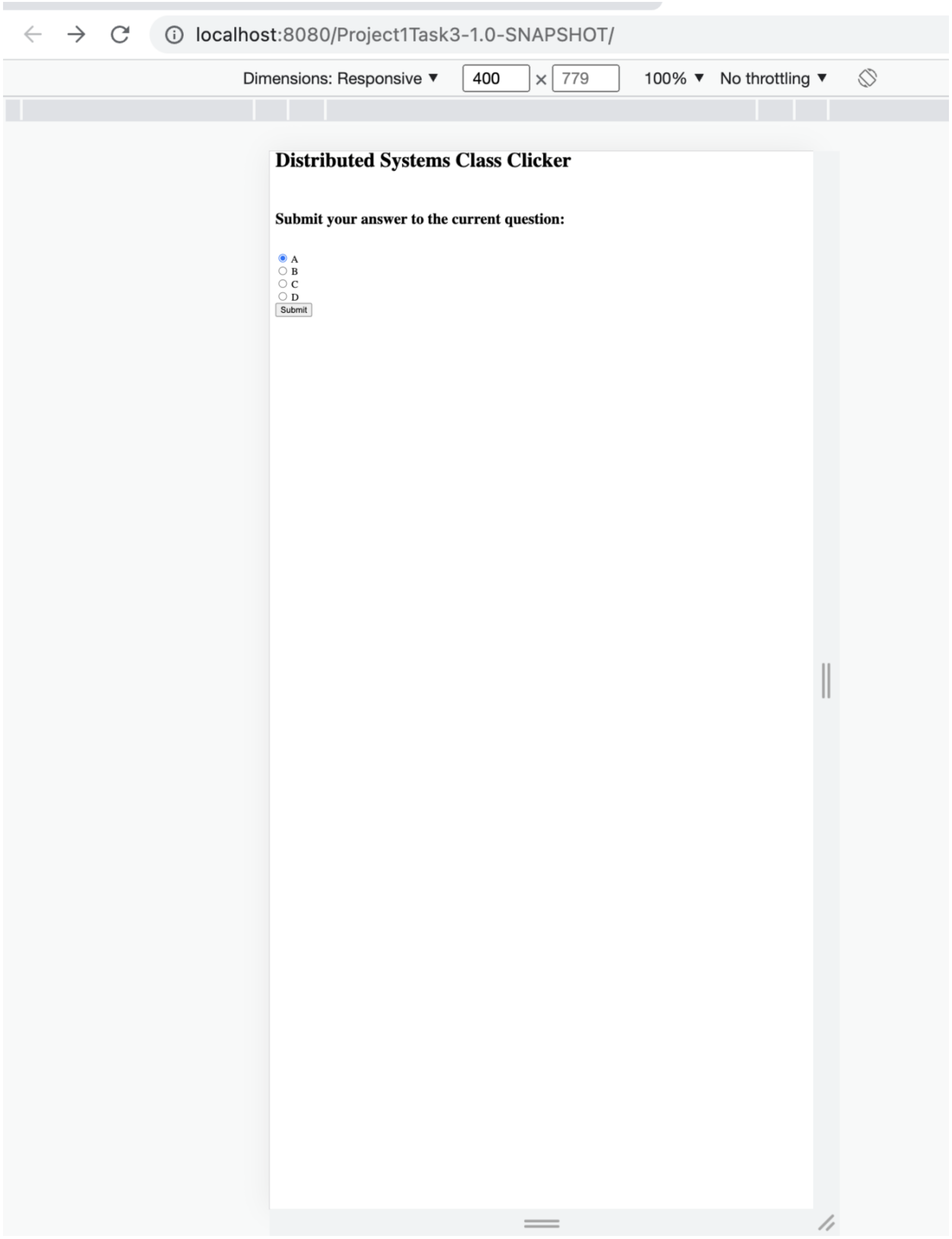
```java
}

/** Parses the country flag image url with Jsoup, and take care of corner case like England */
1 usage
private String getFlag(String country) {
    String countryForFlag = country;
    if ("england".equalsIgnoreCase(countryForFlag)) {
        countryForFlag = "united kingdom";
    }
    String url = "https://www.cia.gov/the-world-factbook/countries/" + countryForFlag.toLowerCase().replace( target: " ",  replacement: "-") + "/flag";
    Document document;
    try {
        document = Jsoup.connect(url).get();
        Elements flagElements= document.getElementsByTag( tagName: "picture");
        String flagURL =  flagElements.get(0).getElementsByTag( tagName: "img").get(0).attr( attributeKey: "src");
        return "https://www.cia.gov" + flagURL;
    } catch (IOException e) {
        e.printStackTrace();
        return null;
    }
}
```



```java
/** Initiates the countries with the given country list text file*/
3 usages
void initCountries() {
    File file;
    try {
        file = new File( pathname: "/Users/jeremyli/Documents/95-702 Distributed System/Project1_zihanli2/Project1Task2/src/main
        Scanner scanner = new Scanner(file);
        while (scanner.hasNextLine()) {
            countries.add(scanner.nextLine());
        }
        scanner.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
/** Initiates the emojis by matching countries to the JSON file and parse into the key-value pair, a faster approach*/
void initEmojis() {
    String url = "https://cdn.jsdelivr.net/npm/country-flag-emoji-json@2.0.0/dist/index.json";
    String response = fetch(url,  certType: "TLSV1.3");
    initCountries();
    JsonArray jsonArray = new Gson().fromJson(response, JsonArray.class);
    for (int i = 0; i < jsonArray.size(); i++) {
        JsonObject jsonObject = jsonArray.get(i).getAsJsonObject();
        if (countries.contains(jsonObject.get("name").getAsString())) {
            emojiMap.put(jsonObject.get("name").getAsString(), jsonObject.get("emoji").getAsString());
        }
    }
}
/** Initiates the emojis by putting in the array of CountryEmoji class*/
1 usage
void initEmojis2() {
    String url = "https://cdn.jsdelivr.net/npm/country-flag-emoji-json@2.0.0/dist/index.json";
    String response = fetch(url,  certType: "TLSV1.3");
    initCountries();
    emojis = new CountryEmoji[countries.size()];
    int count = 0;
    JsonArray jsonArray = new Gson().fromJson(response, JsonArray.class);
    for (int i = 0; i < jsonArray.size(); i++) {
        JsonObject jsonObject = jsonArray.get(i).getAsJsonObject();
        if (countries.contains(jsonObject.get("name").getAsString())) {
            CountryEmoji countryEmoji = new CountryEmoji(jsonObject.get("name").getAsString(), jsonObject.get("emoji").getAsStri
            emojis[count++] = countryEmoji;
        }
    }
}


/** Gets the emoji for assigned country in O(N) time to meet submission requirement*/
1 usage
private String getEmoji(String country) {
       return emojiMap.containsKey(country) ? emojiMap.get(country) : "NA";
    for (CountryEmoji countryEmoji : emojis) {
        if (countryEmoji.getCountry().equalsIgnoreCase(country)) {
            return countryEmoji.getEmoji();
        }
    }
    return "NA";
}
```

Task3

Screen scrapings of input page, submit page, and getResults page for both laptops and mobiles


localhost:8080/Project1Task3-1.0-SNAPSHOT/

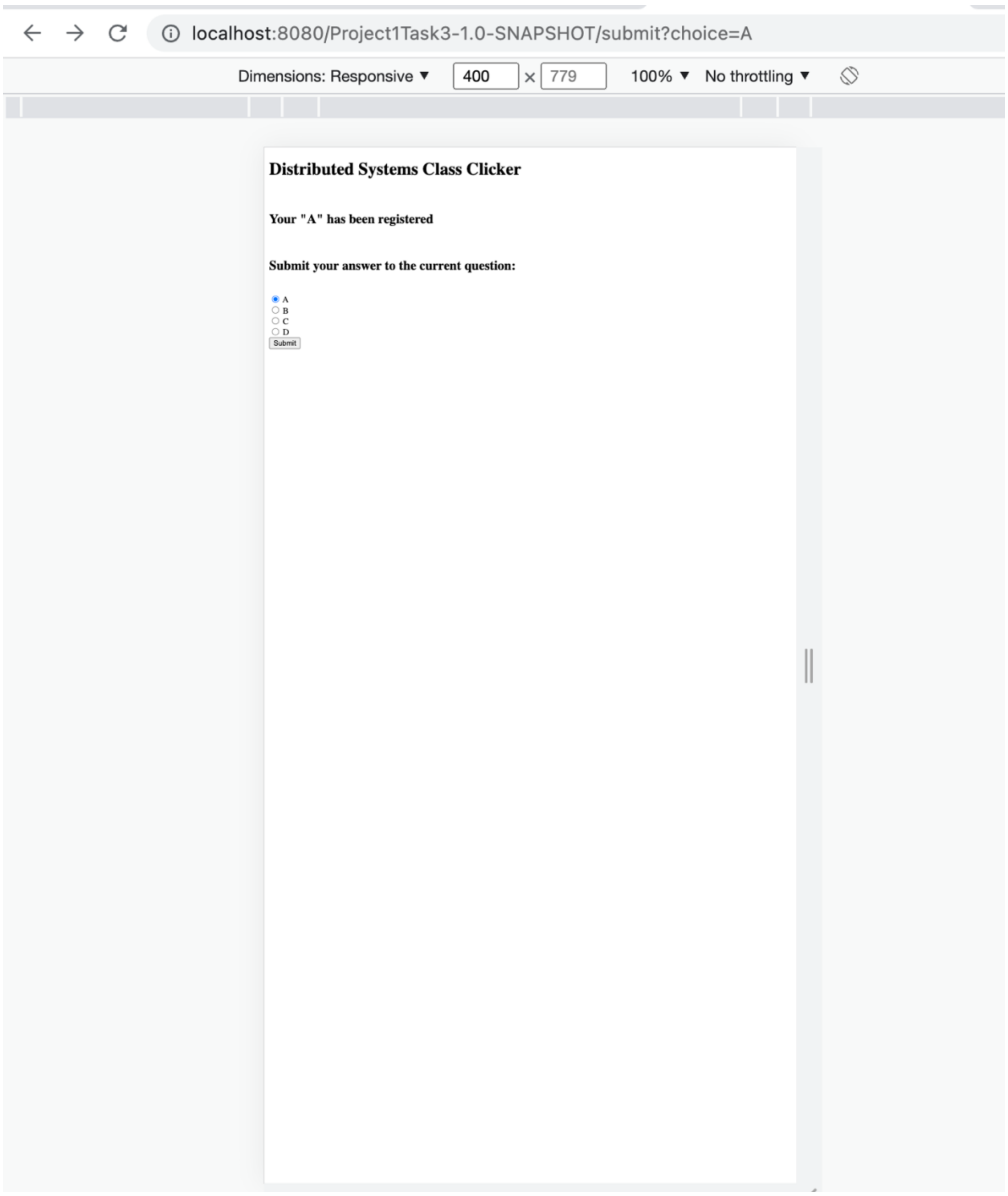
Dimensions: Responsive ▼ 400 × 779 100% ▼ No throttling ▼

Distributed Systems Class Clicker

Submit your answer to the current question:

[✓ A] B C D

Submit



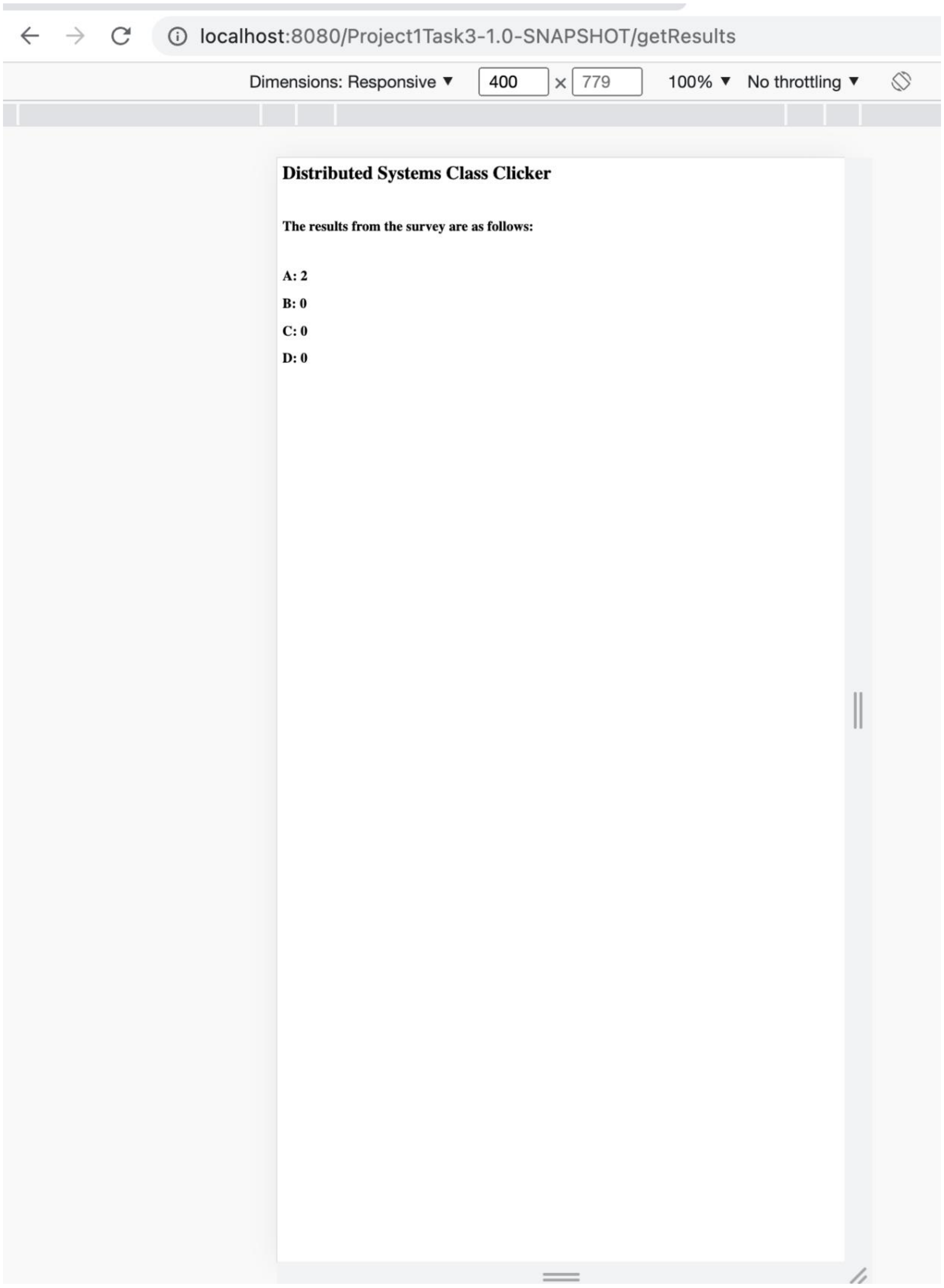
Distributed Systems Class Clicker

Your "A" has been registered

Submit your answer to the current question:

[✓ A]
B
C
D

Submit


localhost:8080/Project1Task3-1.0-SNAPSHOT/getResults

Dimensions: Responsive ▼ 400 × 779 100% ▼ No throttling ▼

Distributed Systems Class Clicker

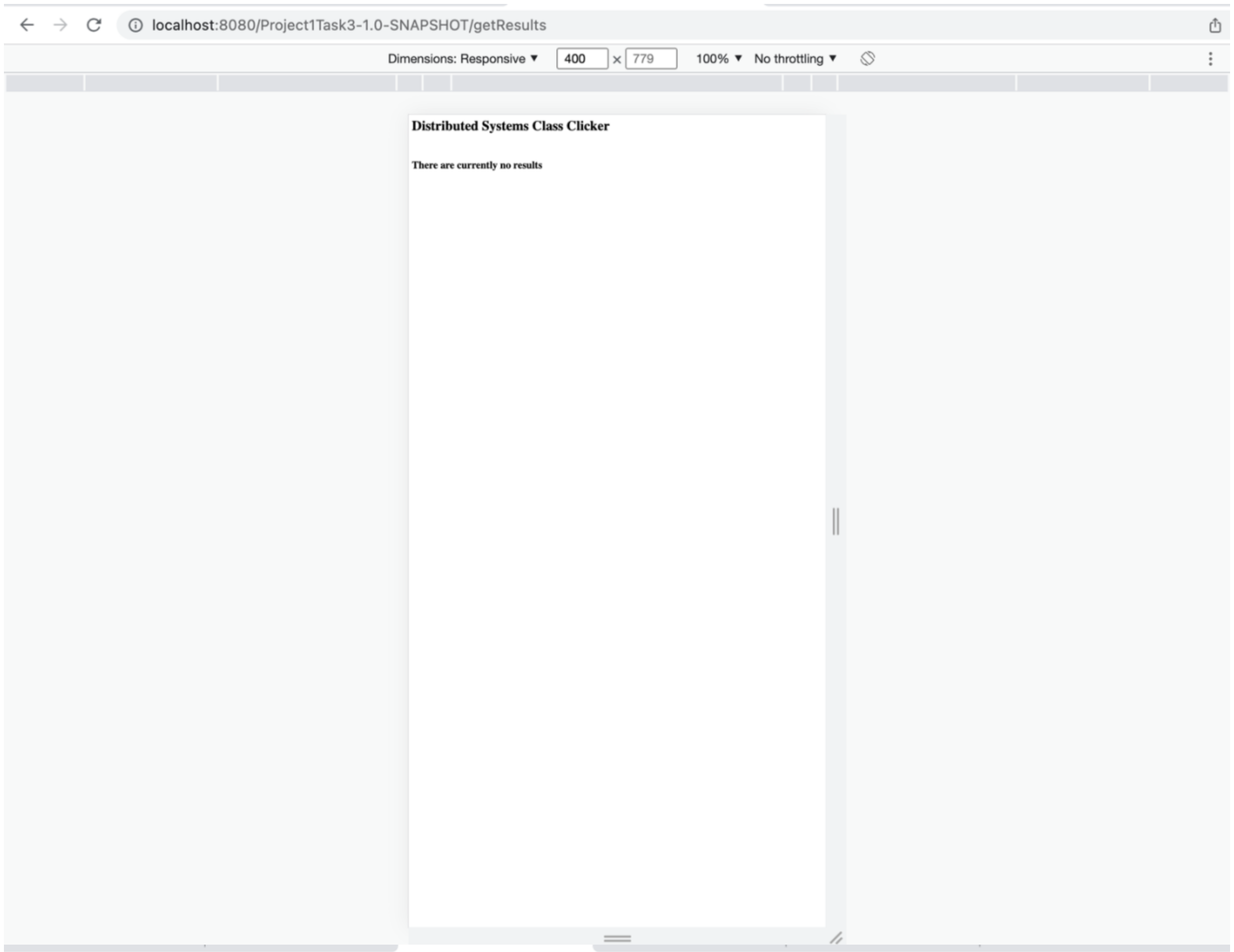
The results from the survey are as follows:

A: 2

B: 0

C: 0

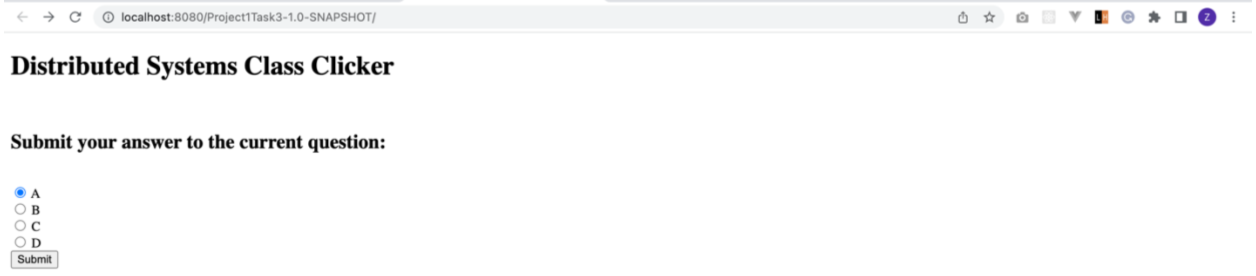
D: 0


localhost:8080/Project1Task3-1.0-SNAPSHOT/getResults

Dimensions: Responsive 400 × 779 100% No throttling

Distributed Systems Class Clicker

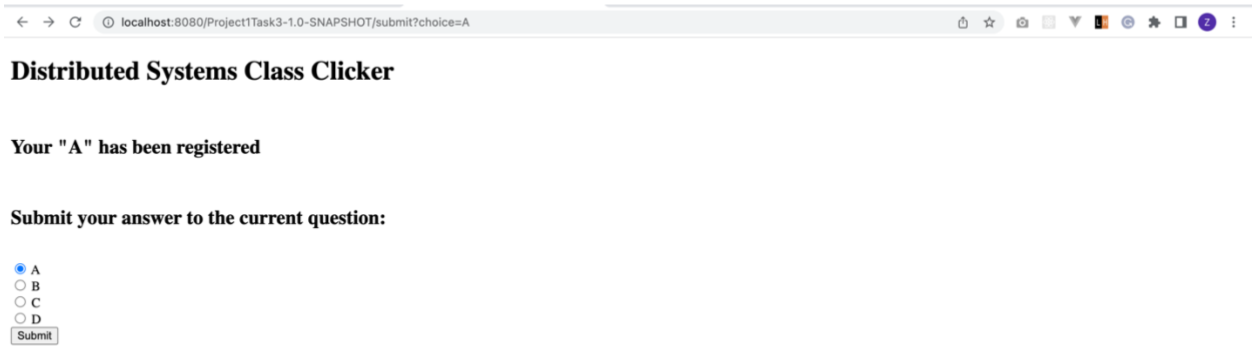
There are currently no results


localhost:8080/Project1Task3-1.0-SNAPSHOT/

Distributed Systems Class Clicker

Submit your answer to the current question:

[✓ A]
B
C
D
Submit



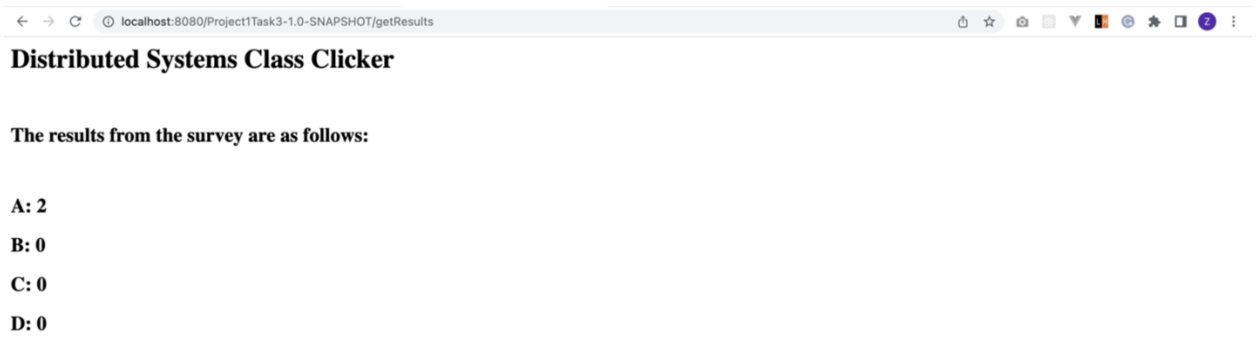
localhost:8080/Project1Task3-1.0-SNAPSHOT/submit?choice=A

Distributed Systems Class Clicker

Your "A" has been registered

Submit your answer to the current question:

[✓ A]
B
C
D



localhost:8080/Project1Task3-1.0-SNAPSHOT/getResults

Distributed Systems Class Clicker

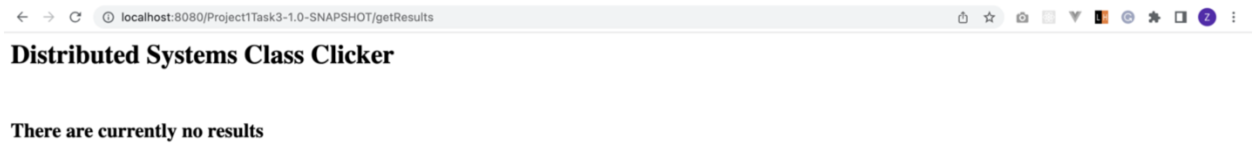
The results from the survey are as follows:

A: 2

B: 0

C: 0

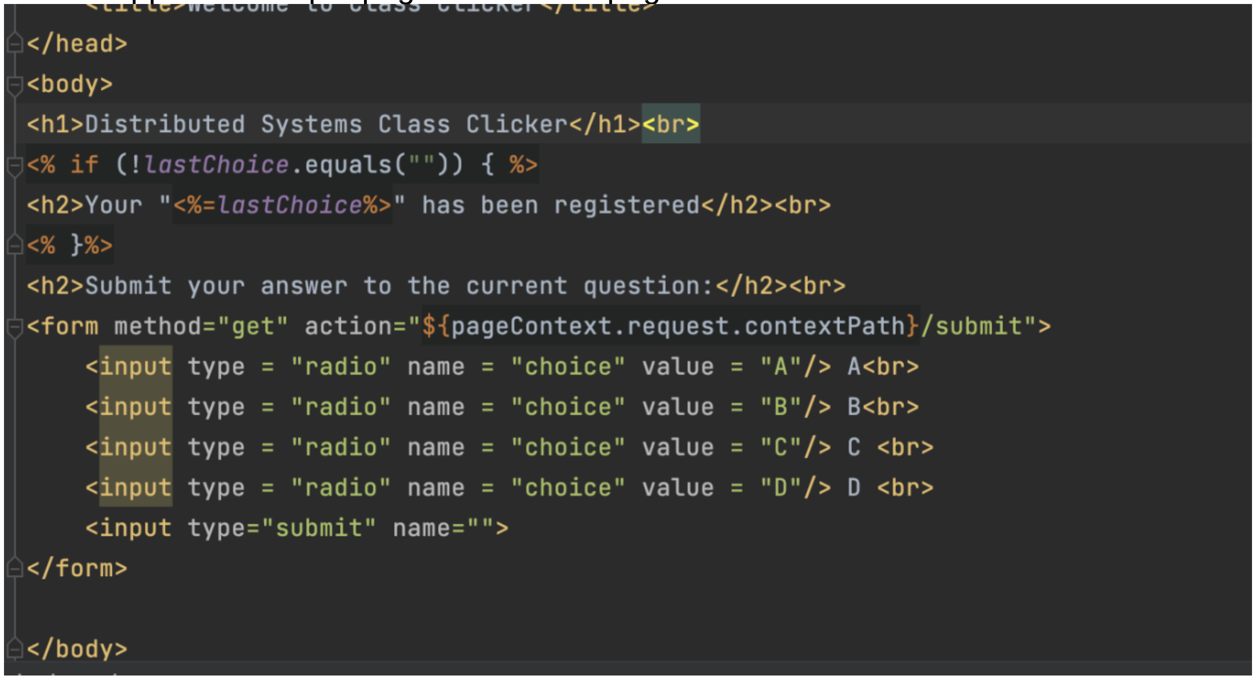
D: 0


localhost:8080/Project1Task3-1.0-SNAPSHOT/getResults

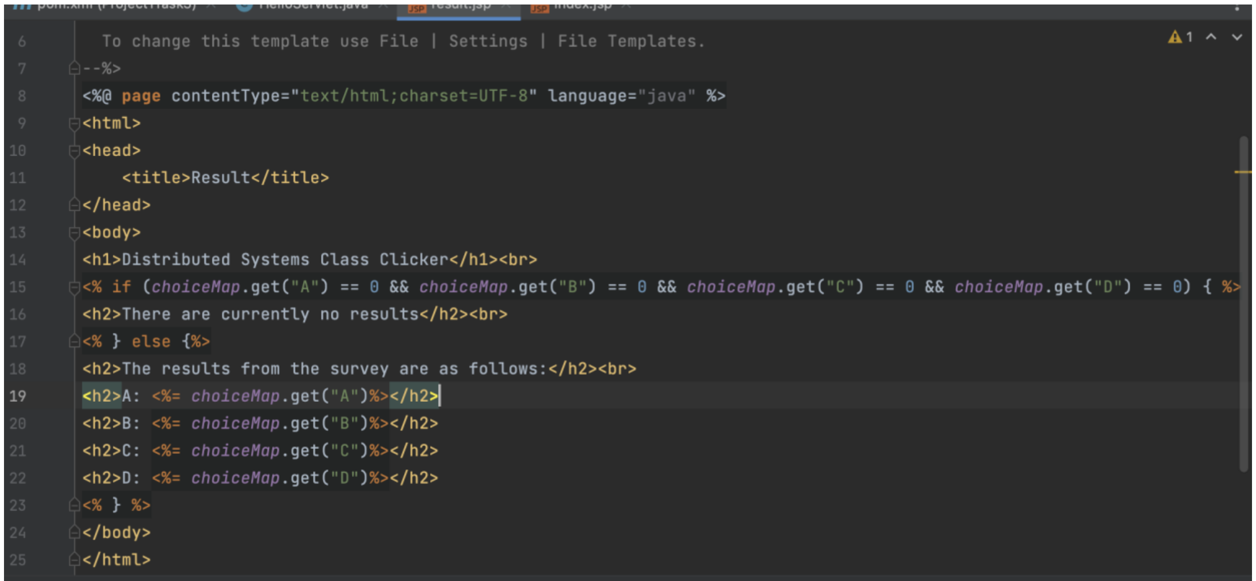
Distributed Systems Class Clicker

There are currently no results

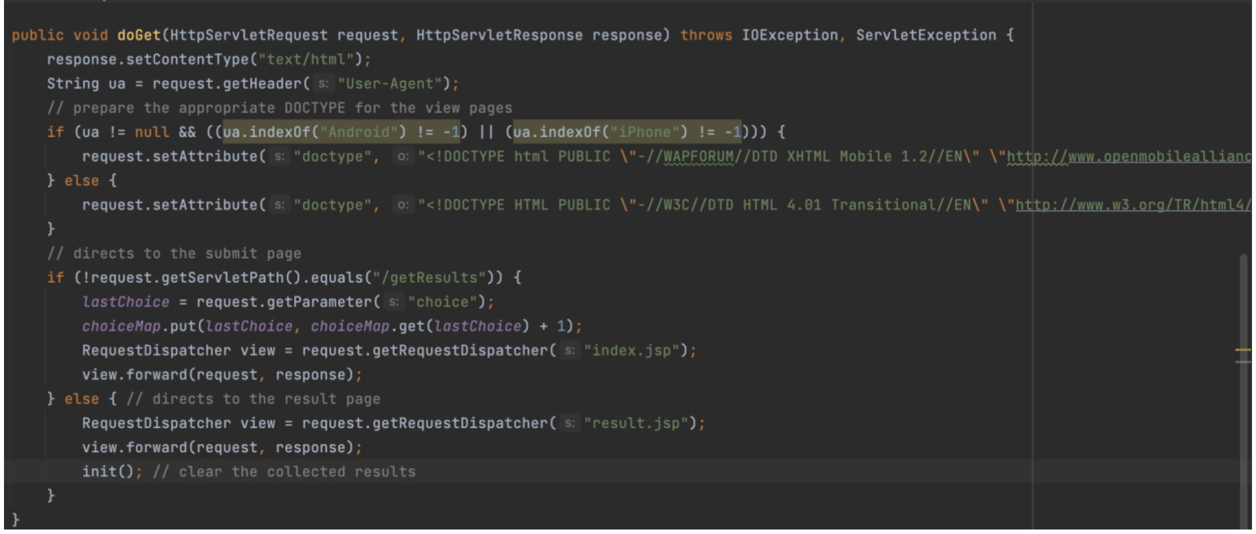
Code snippets for output page and results page



```
</head>
<body>
<h1>Distributed Systems Class Clicker</h1><br>
<% if (!lastChoice.equals("")) { %>
<h2>Your "<%=lastChoice%>" has been registered</h2><br>
<% }%>
<h2>Submit your answer to the current question:</h2><br>
<form method="get" action="${pageContext.request.contextPath}/submit">
    <input type = "radio" name = "choice" value = "A"/> A<br>
    <input type = "radio" name = "choice" value = "B"/> B<br>
    <input type = "radio" name = "choice" value = "C"/> C <br>
    <input type = "radio" name = "choice" value = "D"/> D <br>
    <input type="submit" name="">
</form>

</body>
```



```
  To change this template use File | Settings | File Templates.
--%>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head>
    <title>Result</title>
</head>
<body>
<h1>Distributed Systems Class Clicker</h1><br>
<% if (choiceMap.get("A") == 0 && choiceMap.get("B") == 0 && choiceMap.get("C") == 0 && choiceMap.get("D") == 0) { %>
<h2>There are currently no results</h2><br>
<% } else {%>
<h2>The results from the survey are as follows:</h2><br>
<h2>A: <%= choiceMap.get("A")%></h2>
<h2>B: <%= choiceMap.get("B")%></h2>
<h2>C: <%= choiceMap.get("C")%></h2>
<h2>D: <%= choiceMap.get("D")%></h2>
<% } %>
</body>
</html>
```



```
public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException {
    response.setContentType("text/html");
    String ua = request.getHeader("User-Agent");
    // prepare the appropriate DOCTYPE for the view pages
    if (ua != null && ((ua.indexOf("Android") != -1) || (ua.indexOf("iPhone") != -1))) {
        request.setAttribute("doctype", "<!DOCTYPE html PUBLIC \"-//WAPFORUM//DTD XHTML Mobile 1.2//EN\" \"http://www.openmobilealliance
    } else {
        request.setAttribute("doctype", "<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.01 Transitional//EN\" \"http://www.w3.org/TR/html4/
    }
    // directs to the submit page
    if (!request.getServletPath().equals("/getResults")) {
        lastChoice = request.getParameter("choice");
        choiceMap.put(lastChoice, choiceMap.get(lastChoice) + 1);
        RequestDispatcher view = request.getRequestDispatcher("index.jsp");
        view.forward(request, response);
    } else { // directs to the result page
        RequestDispatcher view = request.getRequestDispatcher("result.jsp");
        view.forward(request, response);
        init(); // clear the collected results
    }
}
```