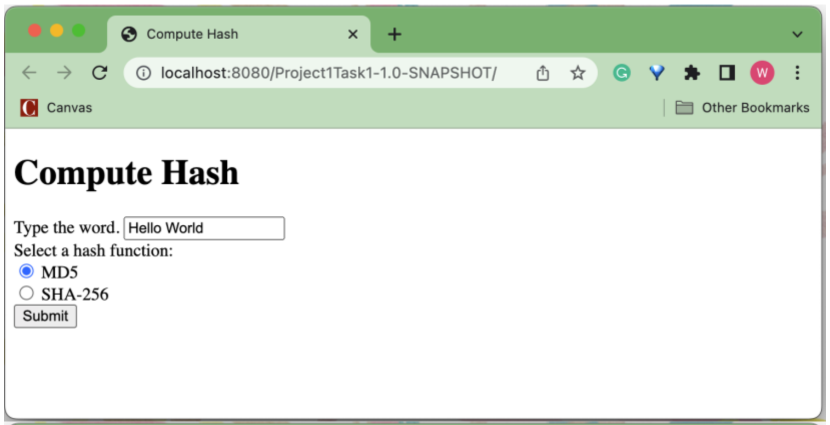
S23 95-702 Distributed Systems Project 1

Candice Chiang (wanteinc)

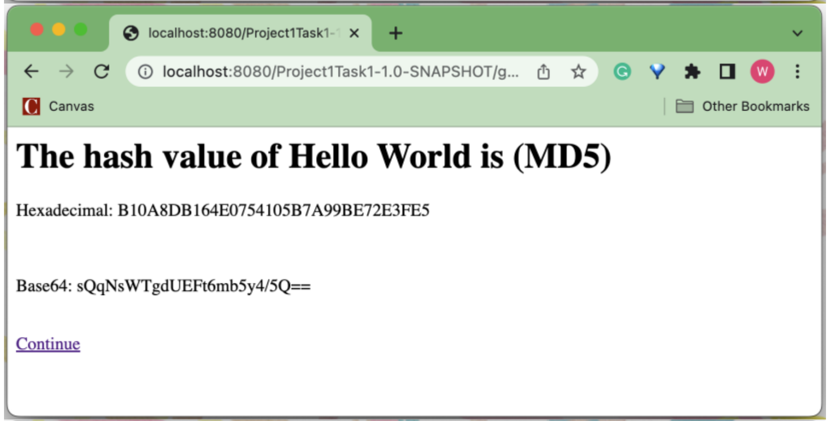
Task 1

1. Screenshots



Compute Hash

Type the word. Hello World
Select a hash function:
[✓ MD5]
SHA-256
Submit

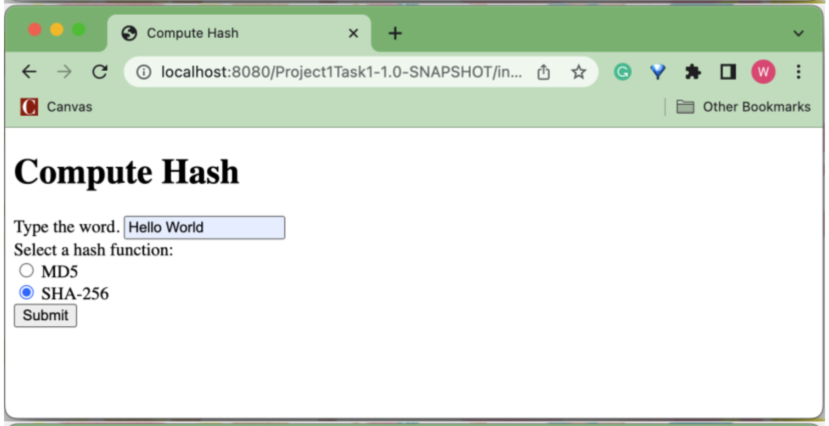

localhost:8080/Project1Task1-
localhost:8080/Project1Task1-1.0-SNAPSHOT/g...
Canvas
Other Bookmarks

The hash value of Hello World is (MD5)

Hexadecimal: B10A8DB164E0754105B7A99BE72E3FE5

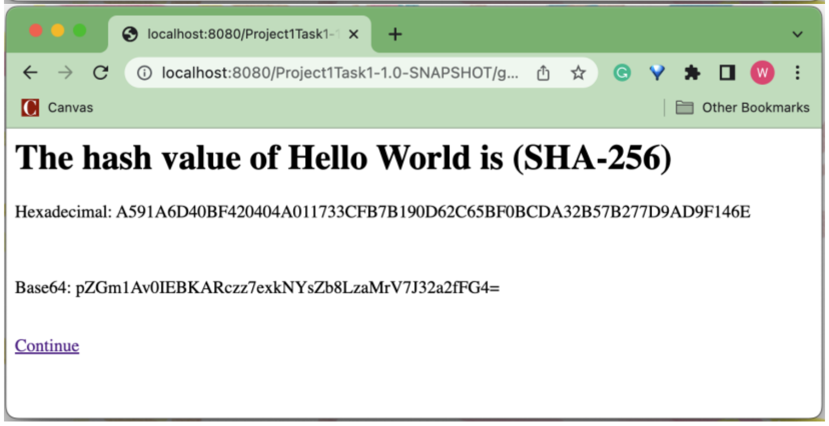
Base64: sQqNsWTgdUEFt6mb5y4/5Q==

Continue



Compute Hash

Type the word. Hello World
Select a hash function:
MD5
[✓ SHA-256]
Submit


localhost:8080/Project1Task1-
localhost:8080/Project1Task1-1.0-SNAPSHOT/g...
Canvas
Other Bookmarks

The hash value of Hello World is (SHA-256)

Hexadecimal: A591A6D40BF420404A011733CFB7B190D62C65BF0BCDA32B57B277D9AD9F146E

Base64: pZGm1Av0IEBKARczz7exkNYsZb8LzaMrV7J32a2fFG4=

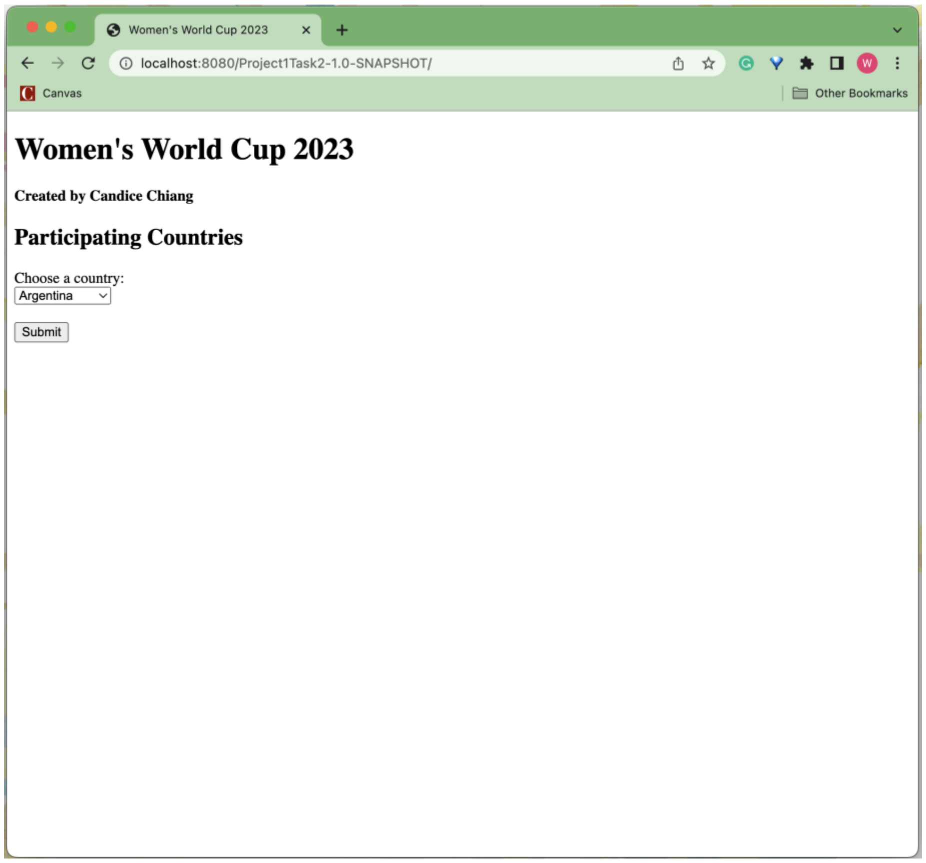
Continue

2. Code Snippet

```
MessageDigest md = null;
// Get the search and hash function parameters.
String search = request.getParameter("searchWord");
String hashFunc = request.getParameter("hash_func");
try {
    // Compute hash values using the requested hash function.
    md = MessageDigest.getInstance(hashFunc);
    byte[] searchHash = md.digest(search.getBytes(StandardCharsets.UTF_8));
    String searchBase64 =
jakarta.xml.bind.DatatypeConverter.printBase64Binary(searchHash);
    String searchHex =
jakarta.xml.bind.DatatypeConverter.printHexBinary(searchHash);
```

Task 2

1. Screenshots
   a. Input page



Women's World Cup 2023

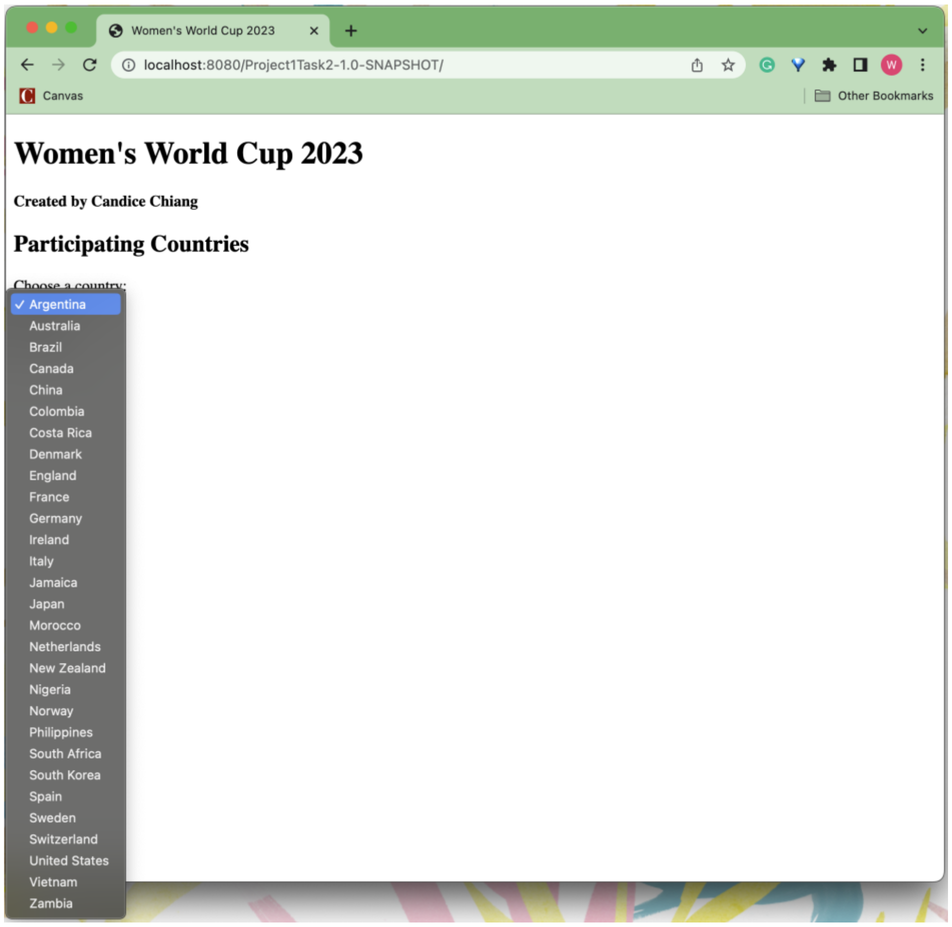
Created by Candice Chiang

Participating Countries

Choose a country:
Argentina

Submit

b. Drop-down menu



Women's World Cup 2023

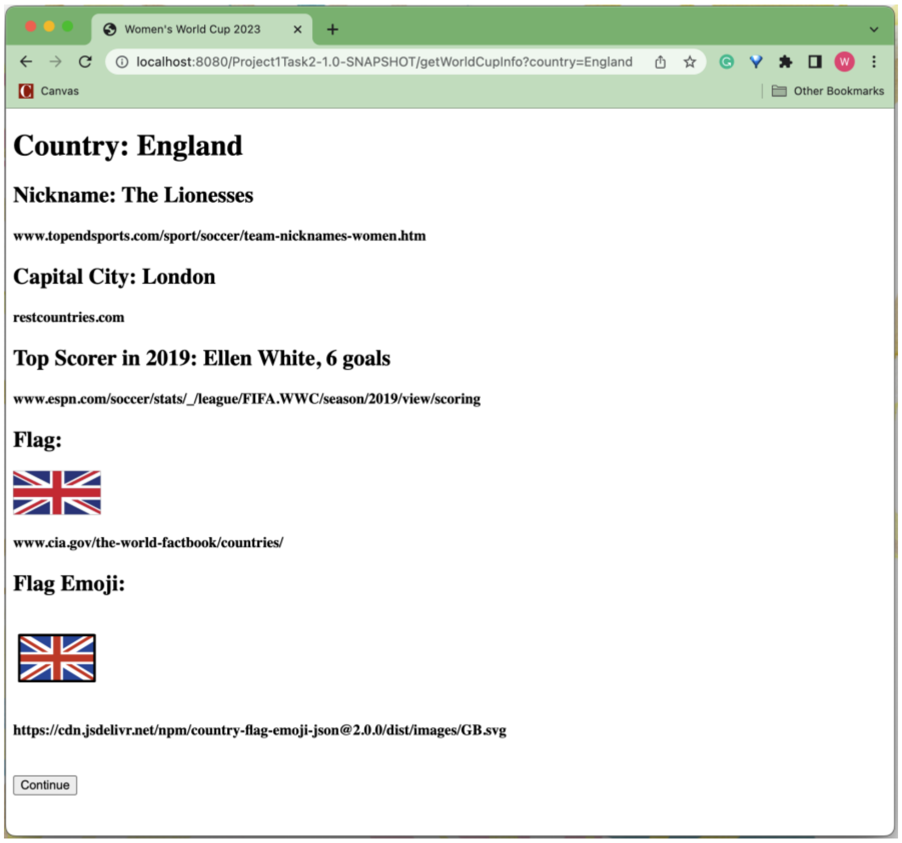
Created by Candice Chiang

Participating Countries

Choose a country:

✓ Argentina
Australia
Brazil
Canada
China
Colombia
Costa Rica
Denmark
England
France
Germany
Ireland
Italy
Jamaica
Japan
Morocco
Netherlands
New Zealand
Nigeria
Norway
Philippines
South Africa
South Korea
Spain
Sweden
Switzerland
United States
Vietnam
Zambia

c. Output page: England


Women's World Cup 2023
localhost:8080/Project1Task2-1.0-SNAPSHOT/getWorldCupInfo?country=England
Canvas
Other Bookmarks

Country: England

Nickname: The Lionesses

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

Capital City: London

restcountries.com

Top Scorer in 2019: Ellen White, 6 goals

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

Flag:

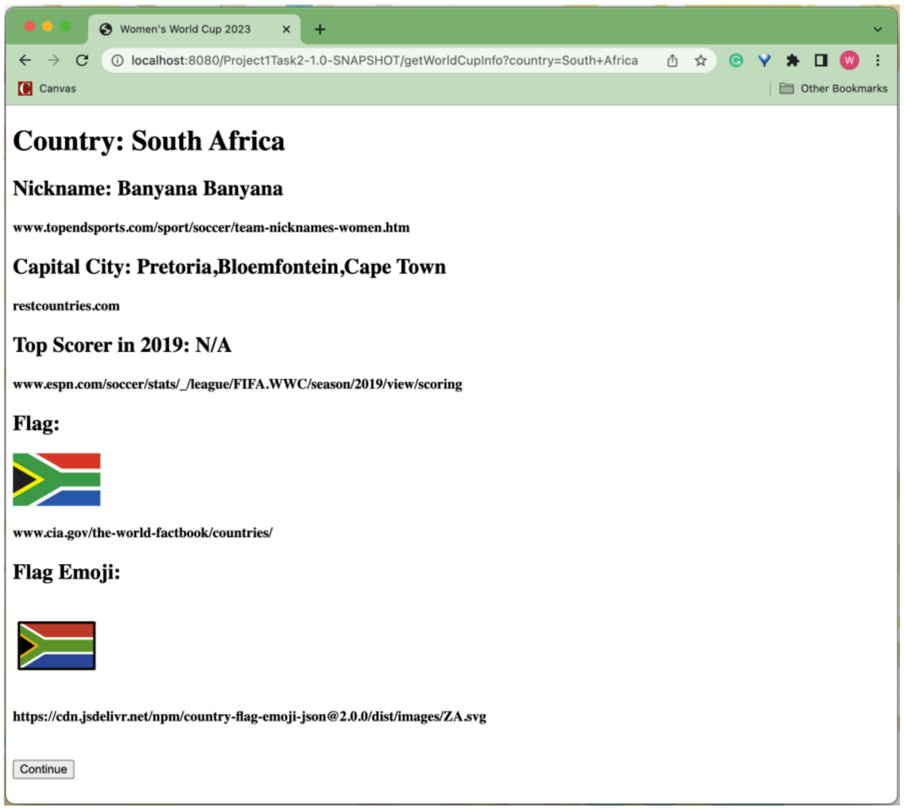
www.cia.gov/the-world-factbook/countries/

Flag Emoji:

https://cdn.jsdelivr.net/npm/country-flag-emoji-json@2.0.0/dist/images/GB.svg

Continue

d. Output page: South Africa


Women's World Cup 2023
localhost:8080/Project1Task2-1.0-SNAPSHOT/getWorldCupInfo?country=South+Africa
Canvas
Other Bookmarks

Country: South Africa

Nickname: Banyana Banyana

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

Capital City: Pretoria,Bloemfontein,Cape Town

restcountries.com

Top Scorer in 2019: N/A

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

Flag:

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

Flag Emoji:

https://cdn.jsdelivr.net/npm/country-flag-emoji-json@2.0.0/dist/images/ZA.svg

Continue

2. Code Snippets
   a. Scraping for the nickname

```
   private void setCountryNicknameMap() throws IOException {
           countryNicknameMap = new HashMap<>();
           Document nickNameList =
Jsoup.connect("https://" + nickNameSource).validateTLSCertificates(false).get();
           // Get the table of country-nickname data.
           Element table = nickNameList.select("table").get(0);
           // Get rows in the table.
           Elements rows = table.select("tr");
           // Extract country names and nicknames and put in the map.
           for (int i = 1; i < rows.size(); i++) {
               Element row = rows.get(i);
               Elements cols = row.select("td");
               if (countryList.contains(cols.get(0).text().trim())) {
                   countryNicknameMap.put(cols.get(0).text().trim(),
   cols.get(1).text().trim());
               }
           }
   }
   public String searchNickname(String searchKey) {
           return countryNicknameMap.getOrDefault(searchKey, "Not Found");
   }
```

   b. Scraping for the capital

```
        private void setCountryCapitalMap() throws IOException {
           String capitalSource = capitalBaseSource + "/v3.1/all";
           String jsonStr= Jsoup.connect("https://" +
   capitalSource).ignoreContentType(true).execute().body();
           // Regex pattern to capture groups (country name) (country code)
   (capital)
           // Extract country code for further emoji mapping.
           String patternStr =
   "\\{\"name\":\\{\"common\":\"(.*?)\".+?(?<=\"cca2\":\")(\\w+)\".+?(?<=\"capit
   al\":\\[)(.*?)]";
           Pattern pattern = Pattern.compile(patternStr);
           Matcher matcher = pattern.matcher(jsonStr);
           countryCapitalMap = new HashMap<>();
           countryCodeMap = new HashMap<>();
           while (matcher.find()) {
               String countryName = matcher.group(1);
               if (countryList.contains(countryName)) {
                   String capital = matcher.group(3).replace("\"", "");
                   countryCapitalMap.put(countryName, capital);
                   countryCodeMap.put(countryName, matcher.group(2));
               } else if (countryName.equals("United Kingdom")) { // handle
   England
                   String capital = matcher.group(3).replace("\"", "");
                   countryCapitalMap.put(countryName, capital);
                   countryCapitalMap.put("England", capital);
                   countryCodeMap.put(countryName, matcher.group(2));
                   countryCodeMap.put("England", matcher.group(2));
```

```
            }
        }
    }
    public String getCapital(String searchKey) {
        return countryCapitalMap.getOrDefault(searchKey, "Not Found");
    }
```

c. Scraping for the top scorer with number of goals

```
    public String getTopScorer(String searchKey) throws IOException {
        Document topScorerList = Jsoup.connect("https://" +
topScorerSource).validateTLSCertificates(false).get();
        String result = "N/A";
        // Get the table.
        Element table = topScorerList.select("table").get(0);
        // Get rows.
        Elements rows = table.select("tr");
        for (int i = 1; i < rows.size(); i++) {
            Element row = rows.get(i);
            Elements cols = row.select("td");
            if (cols.get(2).select("span >
a.AnchorLink").text().trim().equals(searchKey)) {
                // Get name of the scorer.
                String scorer = cols.get(1).select("span >
a.AnchorLink").text().trim();
                // Get total goals.
                String score = cols.get(4).select("span.tar").text().trim();
                result = scorer + ", " + score + " goals";
                break;
            }
        }
        return result;
    }
```

d. Scraping of the flag

```
    public String getFlag(String searchKey) throws IOException {
        // Handle exceptions.
        if (searchKey.equals("England")) {
            searchKey = "United Kingdom";
        } else if (searchKey.equals("South Korea")) {
            searchKey = "Korea South";
        }
        searchKey = searchKey.replace(" ", "-").toLowerCase();
        Document flag = Jsoup.connect("https://" + flagSource
searchKey).validateTLSCertificates(false).get();
        Element infoBox = flag.select("div.col-md-6.mb30").get(0);
        Elements imge = infoBox.select("div.wfb-card-wrapper > div.row.no-
gutters > div.col-md-3.align-self-center > div.wfb-card__image-container >
div.gatsby-image-wrapper.gatsby-image-wrapper-constrained.wfb-card__image");
        Element image = imge.select("img").get(1);
        String result = image.attr("data-src");
        result = "https://www.cia.gov" + result;
        return result;
        }
```

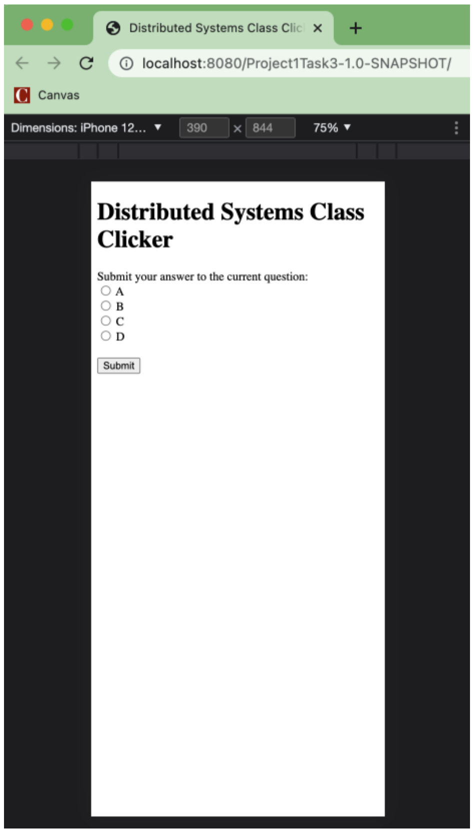
e. Api call for the flag emoji JSON record, including the conversion to a Java array of objects.

```
private static class Emoji {
     private String name;
      private String emoji;
      private String unicode;
      private String image;

      Emoji () {

      }

      Emoji (String n, String e, String u, String i) {
         super();
         this.name = n;
         this.emoji = e;
         this.unicode = u;
         this.image = i;
      }
      public String getName() {
         return name;
      }

      public String getEmoji() {
         return emoji;
      }

     public String getUnicode() {
         return unicode;
      }

      public String getImage() {
         return image;
      }
   }
    private void setCountryEmojiList() throws IOException {
         String emojiSource = "https://cdn.jsdelivr.net/npm/country-flag-
emoji-json@2.0.0/dist/by-code.json";
         String jsonStr=
Jsoup.connect(emojiSource).ignoreContentType(true).validateTLSCertificates(fa
lse).execute().body();
         JSONObject jsonObject = new JSONObject(jsonStr);
         JSONArray names = jsonObject.names();
         countryEmojiList = new ArrayList<>();
         Gson gson = new Gson();
         Emoji emoji;
         for (int i = 0; i < names.length(); i++) {
             if (countryCodeMap.containsValue(names.getString(i))) {
                 emoji =
gson.fromJson(jsonObject.get(names.getString(i)).toString(), Emoji.class);
                 if (names.getString(i).equals("GB")) {
                    Emoji eng = new Emoji("England", emoji.getEmoji(),
emoji.getUnicode(), emoji.getImage());
                    countryEmojiList.add(eng);
                }
                countryEmojiList.add(emoji);
            }
        }
    }
    public String getEmoji(String searchKey) {
        String result = "Not Found";
        for (Emoji emoji : countryEmojiList) {
            if (emoji.getName().equals(searchKey)) {
                result = emoji.getImage();
                break;
            }
        }
        return result;
    }
```

Task 3

1. Screenshots

a. Input page



Distributed Systems Class Clic
localhost:8080/Project1Task3-1.0-SNAPSHOT/
Canvas
Dimensions: iPhone 12...
390
844
75%

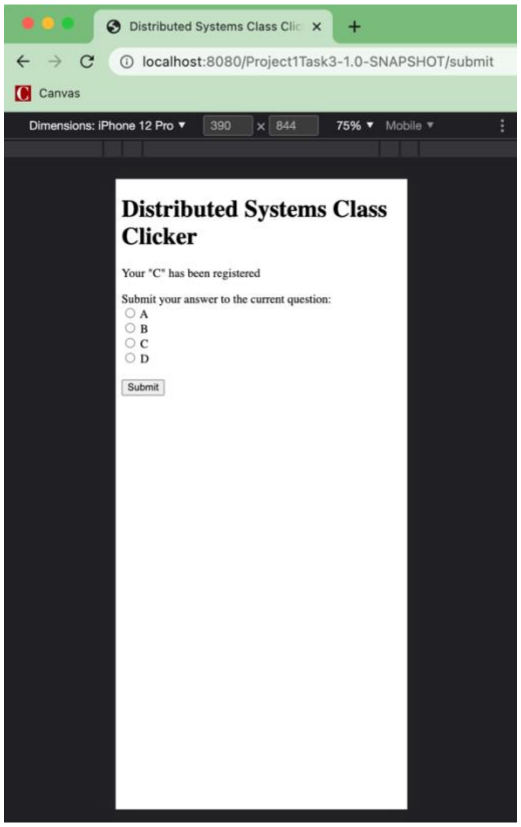
Distributed Systems Class Clicker

Submit your answer to the current question:

- ○ A
- ○ B
- ○ C
- ○ D

Submit

b. Output page (one vote)



Distributed Systems Class Clic
localhost:8080/Project1Task3-1.0-SNAPSHOT/submit
Canvas
Dimensions: iPhone 12 Pro
390
844
75%
Mobile

Distributed Systems Class Clicker

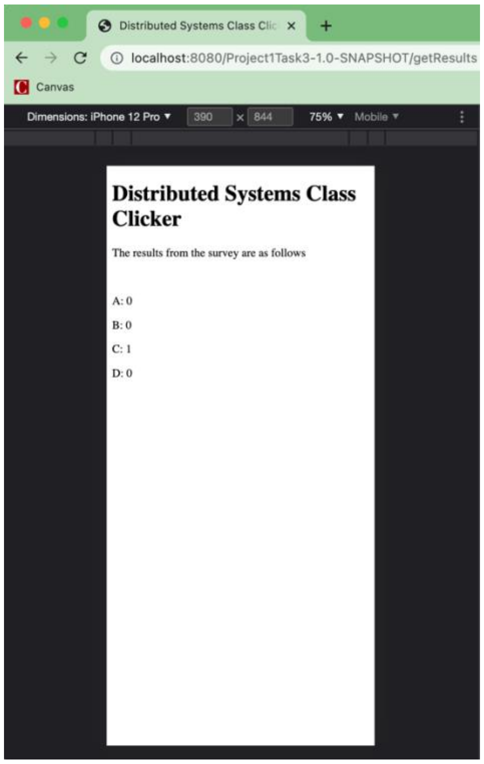
Your "C" has been registered

Submit your answer to the current question:

- ○ A
- ○ B
- ○ C
- ○ D

Submit

c. Results page



Distributed Systems Class Clic
localhost:8080/Project1Task3-1.0-SNAPSHOT/getResults
Canvas
Dimensions: iPhone 12 Pro
390
844
75%
Mobile

Distributed Systems Class Clicker

The results from the survey are as follows

A: 0

B: 0

C: 1

D: 0

2. Code Snippets
   a. Output page

      Servlet

```
} else {
            // Record the answer.
            cm.addResult(answer);
            // Go to index.jsp.
            nextView = "index.jsp";
}
```

      Model

```
public void addResult(String option) {
    if (option != null) {
        this.answerMap.put(option, answerMap.get(option) + 1);
    }
}
```

      JSP

```
<%-- Check if no answers, no previous recorded answer will be shown. --%>
<% if (request.getParameter("answer") != null) { %>
    <p>Your "<%= request.getParameter("answer") %>" has been registered</p>
<% } %>
<form action="submit" method="POST">
    <label for="letter">Submit your answer to the current question:
</label><br>
    <input type="radio" id="optionA" name="answer" value="A">
    <label for="optionA">A</label><br>
    <input type="radio" id="optionB" name="answer" value="B">
    <label for="optionB">B</label><br>
    <input type="radio" id="optionC" name="answer" value="C">
    <label for="optionC">C</label><br>
    <input type="radio" id="optionD" name="answer" value="D">
    <label for="optionD">D</label><br><br>
    <input type="submit" value="Submit" />
</form>
```

   b. Results page

      Servlet

```
if(path.equals("/getResults")) {
            // Get the recorded answers.
            int totalA = cm.getTotal("A");
            int totalB = cm.getTotal("B");
            int totalC = cm.getTotal("C");
            int totalD = cm.getTotal("D");
            int sum = totalA + totalB + totalC + totalD;
            // Pass the answer attributes to the view.
            request.setAttribute("totalA", totalA);
            request.setAttribute("totalB", totalB);
            request.setAttribute("totalC", totalC);
            request.setAttribute("totalD", totalD);
            request.setAttribute("sum", sum);
            // Go th resut.jsp.
            nextView = "result.jsp";
            // Reset the recorded answers.
            cm = new DSClickerModel();
}
```

JSP

```
    <%-- Check if no answers. --%>
    <% if (request.getAttribute("sum").equals(0)) { %>
        <p>There are currently no results</p>
    <% } else { %>
        <p>The results from the survey are as follows</p><br>
        <p>A: <%= request.getAttribute("totalA") %></p>
        <p>B: <%= request.getAttribute("totalB") %></p>
        <p>C: <%= request.getAttribute("totalC") %></p>
        <p>D: <%= request.getAttribute("totalD") %></p>
    <% } %>
```