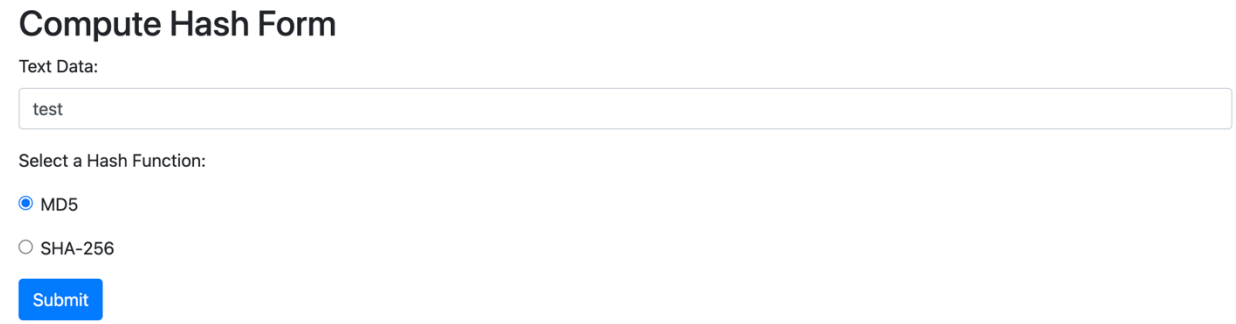
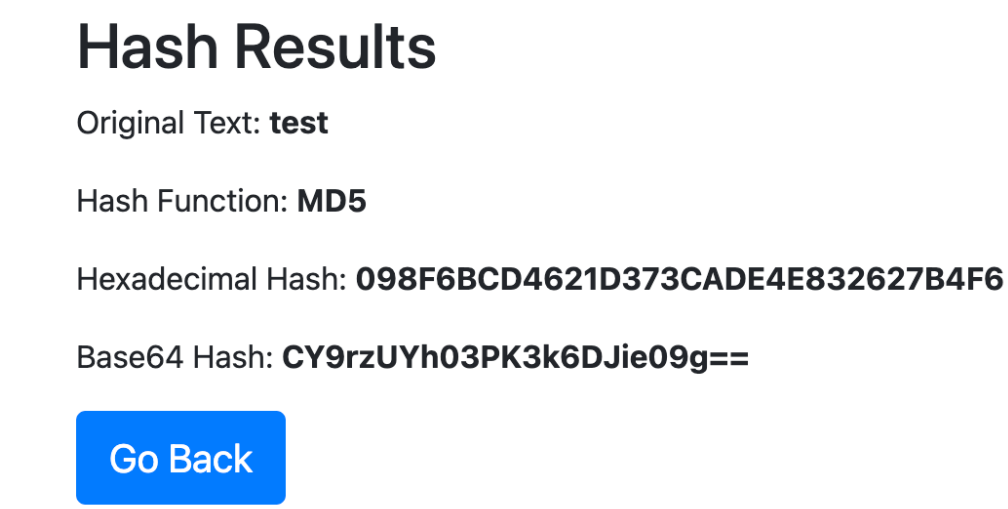
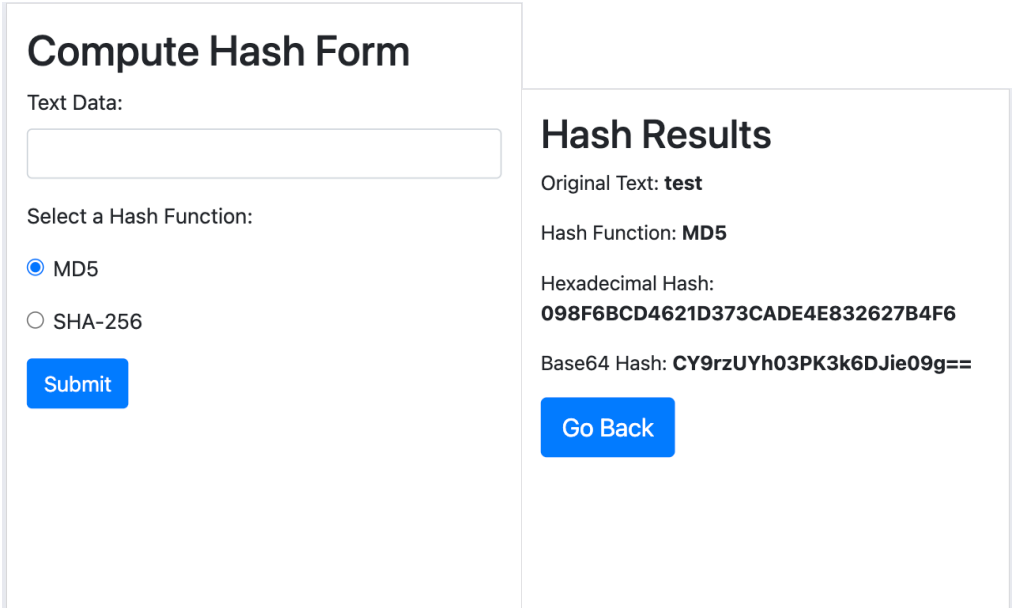
Project 1

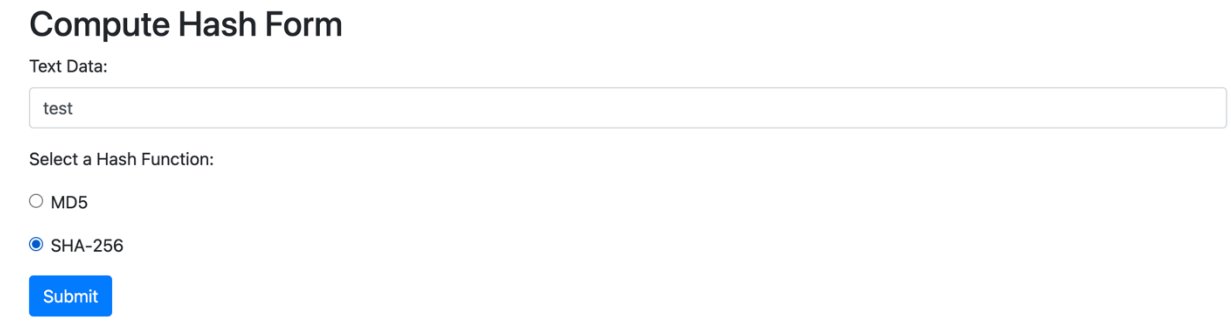
Louis Chang (hungyic)

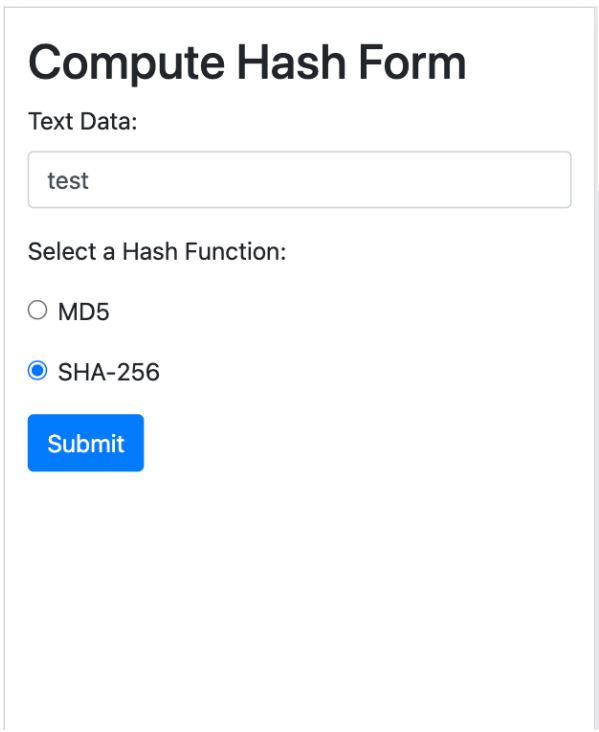
Task 1

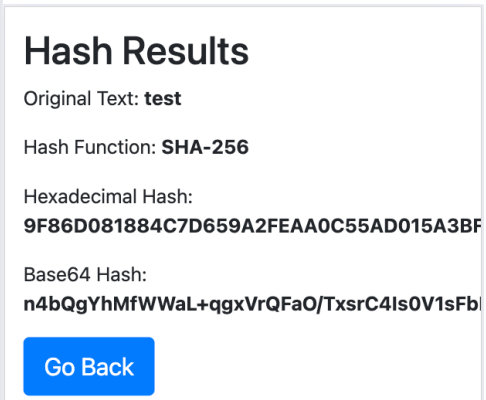










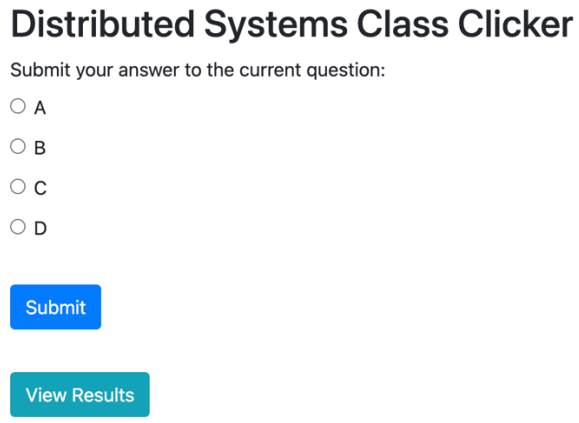


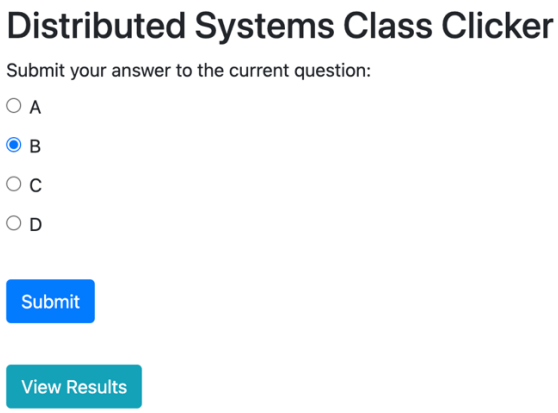

```
/**
 * Author: Louis Chang (hungyic)
 * Last Modified: 02/05/2024
 */
```

```java
@WebServlet(name = "ComputeHashes", urlPatterns = {"/ComputeHashes"})
public class ComputeHashes extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
        String textData = req.getParameter("textData");
        String hashFunction = req.getParameter("hashFunction");

        String hexHash = "";
        String base64Hash = "";

        try {
            // Compute the hash value
            MessageDigest digest =
MessageDigest.getInstance(hashFunction);
            byte[] hashedBytes = digest.digest(textData.getBytes("UTF-
8"));

            // Convert to Hex and Base64
            hexHash = DatatypeConverter.printHexBinary(hashedBytes);
            base64Hash = DatatypeConverter.printBase64Binary(hashedBytes);

        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
        }

        // Set the content type to HTML as the response will be in HTML
        resp.setContentType("text/html;charset=UTF-8");
        PrintWriter out = resp.getWriter();

        // Write the HTML response
        out.println("<!DOCTYPE html>");
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Hash Results</title>");
        // Import Bootstrap CSS
        out.println("<link rel='stylesheet'
href='https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min
.css'>");
        out.println("<meta name='viewport' content='width=device-width,
initial-scale=1, shrink-to-fit=no'>"); // 添加视口元标签
        out.println("</head>");
        out.println("<body>");
        out.println("<div class='container mt-3'>");
        out.println("<h2>Hash Results</h2>");
        out.println("<p>Original Text: <strong>" + textData +
"</strong></p>");
        out.println("<p>Hash Function: <strong>" + hashFunction +
"</strong></p>");
        out.println("<p>Hexadecimal Hash: <strong>" + hexHash +
"</strong></p>");
        out.println("<p>Base64 Hash: <strong>" + base64Hash +
"</strong></p>");
```

```
        out.println("<button type='button' class='btn btn-primary btn-lg'
onclick='goBack()'>Go Back</button>"); // 使用 btn-lg 类来增大按钮
        out.println("</div>");
        out.println("<script src='https://code.jquery.com/jquery-
3.5.1.slim.min.js'></script>");
        out.println("<script
src='https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.2/dist/umd/popper.min
.js'></script>");
        out.println("<script
src='https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.j
s'></script>");
        out.println("<script>function goBack()
{ window.history.back(); }</script>");
        out.println("</body>");
        out.println("</html>");
    }
```

Task 2





Distributed Systems Class Clicker

Your option "B" has been registered.

Submit your answer to the current question:

- A
- B
- C
- D

Submit

View Results

Distributed Systems Class Clicker

Your option "C" has been registered.

Submit your answer to the current question:

- A
- B
- C
- D

Submit

View Results

Distributed Systems Class Clicker

Your option "B" has been registered.

Submit your answer to the current question:

- A
- B
- C
- D

Submit

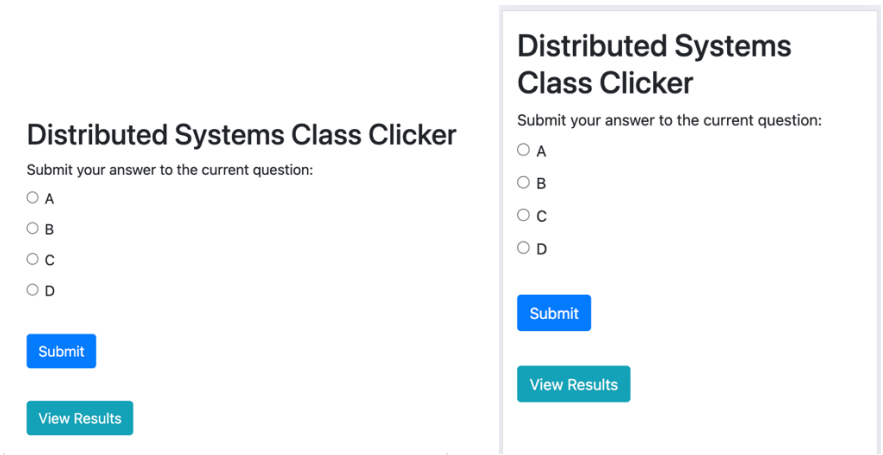
View Results

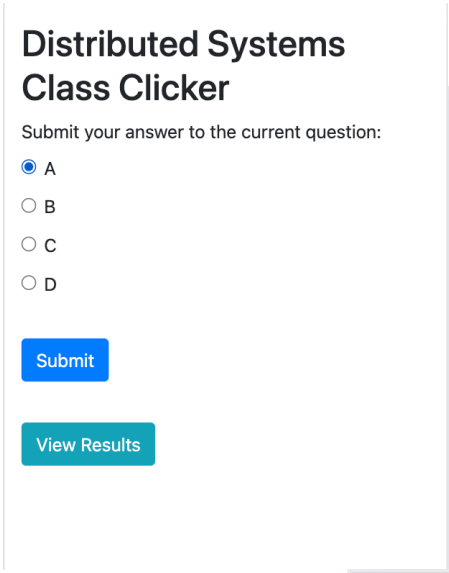
Distributed Systems Class Clicker - Results

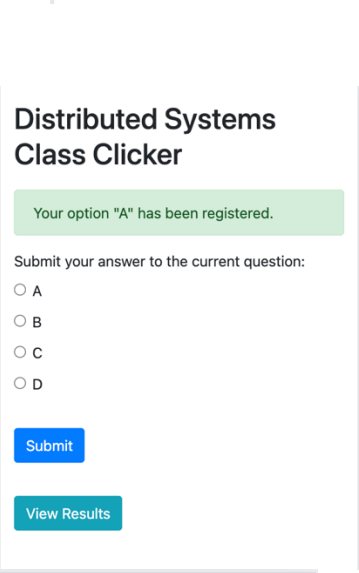
The results from the survey are as follows:

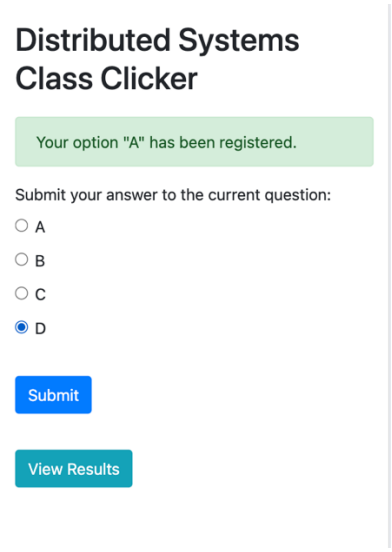
B: 1
C: 1

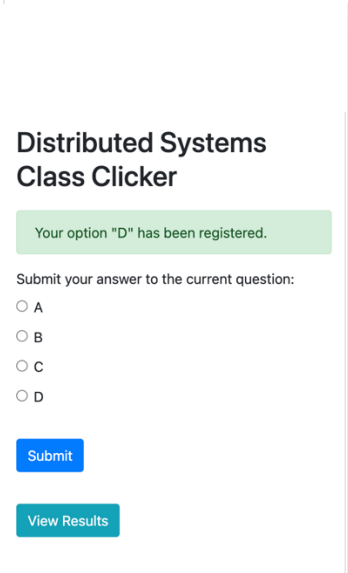
Back to Questions

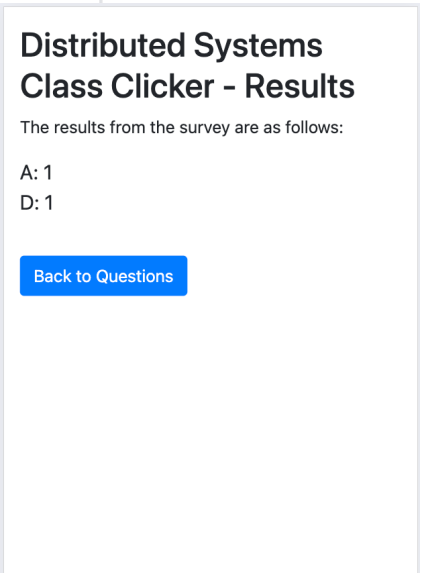












Results.jsp

```
<div class="container mt-3">
    <h2>Distributed Systems Class Clicker - Results</h2>
```

```
    <!-- Check if the results object is not empty -->
    <% if (request.getAttribute("results") != null) { %>
    <p>The results from the survey are as follows:</p>
    <ul class="list-unstyled">
        <!-- Iterate over the results map entries -->
        <c:forEach items="${requestScope.results}" var="entry">
            <li class="result-item">${entry.key}: ${entry.value}</li>
        </c:forEach>
    </ul>
    <% } else { %>
    <div class="alert alert-secondary" role="alert">
        There are currently no results.
    </div>
    <% } %>

    <a href="index.jsp" class="btn btn-primary back-button">Back to
Questions</a>
</div>
```

SubmitServlet

```
/**
 * Author: Louis Chang (hungyic)
 * Last Modified: 02/05/2024
 */
@WebServlet(name = "Submit", urlPatterns = {"/Submit"})
public class SubmitServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        // Forward to submission form JSP
        RequestDispatcher dispatcher =
request.getRequestDispatcher("/index.jsp");
        dispatcher.forward(request, response);
    }


    /**
     * get the option user chose from index.jsp, save it to the map, and
then go back to index.jsp
     * @param request
     * @param response
     * @throws ServletException
     * @throws IOException
     */
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        String option = request.getParameter("option");
        ServletContext context = getServletContext();
        if (option != null) {
            Map<String, Integer> results = (Map<String, Integer>)
context.getAttribute("results");
            if (results == null) {
```

```
                results = new HashMap<>();
                context.setAttribute("results", results);
            }
            // if the option has existed, add the count to value
            results.merge(option, 1, Integer::sum);
            request.setAttribute("message", "Your option \"" + option +
"\" has been registered. ");
            request.getRequestDispatcher("/index.jsp").forward(request,
response);
        }
    }
```

ResultsServlet

```
/**
 * Author: Louis Chang (hungyic)
 * Last Modified: 02/05/2024
 */
@WebServlet(name = "ResultsServlet", urlPatterns = {"/getResults"})
public class ResultsServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        // Get the current session
        ServletContext context = getServletContext();
        // Retrieve the answers map from the session
        Map<String, Integer> results = (Map<String, Integer>)
context.getAttribute("results");
        request.setAttribute("results", results);
        // Clear the stored results so that a new question can be posed
        context.removeAttribute("results");

        // Forward to the results JSP
        RequestDispatcher dispatcher =
request.getRequestDispatcher("/results.jsp");
        dispatcher.forward(request, response);
    }

}
```

Task 3

Know Your State

*Your Name:

Enter your name

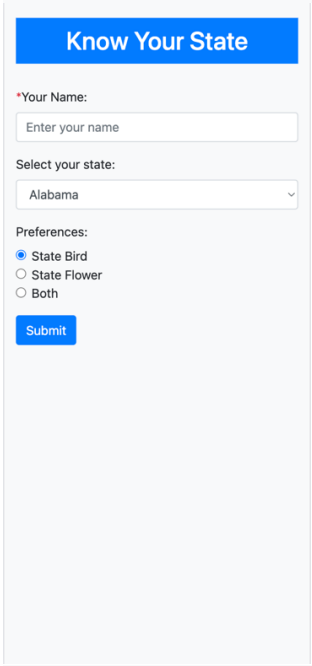
Select your state:

Alabama

Preferences:

- State Bird
- State Flower
- Both

Submit



Know Your State

*Your Name:

Louis

Select your state:

Alabama

Preferences:

- State Bird
- State Flower
- Both

Submit

Know Your State

Hi, Louis

01 Alabama

Population: 4893186

State Bird: Wild turkey (state game bird)

Scientific Name: Meleagris gallopavo

Year: 1980

Credit:Mad Tinman at https://en.wikipedia.org/

Go Back

Know Your State

Hi, Louis

01 Alabama

Population: 4893186

State Bird: Wild turkey (state game bird)

Scientific Name: Meleagris gallopavo

Year: 1980

Credit:Mad Tinman at https://en.wikipedia.org/

Go Back

Know Your State

*Your Name:

Ken

Select your state:

Alabama

Preferences:

- State Bird
- State Flower
- Both

Submit

Know Your State

Hi, Ken

01 Alabama

Population: 4893186

State Flower: Camellia (state flower)

Scientific Name: Camellia japonica

Year: 1959

Credit:Mad Tinman at https://en.wikipedia.org/

Go Back

Know Your State

*Your Name:

Ken

Select your state:

Alabama

Preferences:

- State Bird
- State Flower
- Both

Submit

Know Your State

Hi, Ken

01 Alabama

Population: 4893186

State Flower: Camellia (state flower)

Scientific Name: Camellia japonica

Year: 1959

Credit:Mad Tinman at https://en.wikipedia.org/

Go Back

Know Your State

*Your Name:

Barbie

Select your state:

Idaho

Preferences:

- State Bird
- State Flower
- Both

Submit

Know Your State

Hi, Barbie

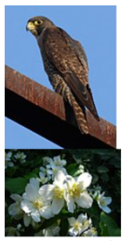
16 Idaho

Population: 1754367

State Bird: Peregrine falcon (state raptor)

Scientific Name: Falco peregrinus

Year: 2004



State Flower: Syringa, mock orange

Scientific Name: Philadelphus lewisii

Year: 1931

Credit:Mad Tinman at https://en.wikipedia.org/

Go Back

Know Your State

Hi, Barbie

16 Idaho

Population: 1754367

State Bird: Peregrine falcon (state raptor)

Scientific Name: Falco peregrinus

Year: 2004

State Flower: Syringa, mock orange

Scientific Name: Philadelphus lewisii

Year: 1931

Credit:Mad Tinman at https://en.wikipedia.org/

Know Your State

*Your Name:

Barbie

Select your state:

Idaho

Preferences:

- State Bird
- State Flower
- Both

Submit

InitServlet

```
/**
 * Read the data of each state and parse it from JSON to list, and then
send out via request
 * @param request
 * @param response
 * @throws ServletException
 * @throws IOException
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    JSONParser parser = new JSONParser();
    List<String> stateList = new ArrayList<>();
    try {
        // get the JSON file and parse it
        String path =
getServletContext().getRealPath("resource/states.json");
        Object obj = parser.parse(new FileReader(path));
        JSONObject jsonObject = (JSONObject) obj;
        JSONArray states = (JSONArray) jsonObject.get("states");

        // Add the parsed data to list
        for (Object state : states) {
            stateList.add((String) state);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }

    // put the state list into request and send to index.jsp
    request.setAttribute("states", stateList);
    RequestDispatcher dispatcher =
request.getRequestDispatcher("/index.jsp");
    dispatcher.forward(request, response);
}
```

StatesServlet

```
/**
 * Get the information from Census Bureau using API and state bird info
from Wikipedia by screen scraping, and then send to stateInfoBird.jsp
 * @param request
 * @param response
 * @throws ServletException
 * @throws IOException
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    // Retrieve the userName from the dropdown
    String userName = request.getParameter("userName");
```

```
    // Retrieve the selected state from the dropdown
    String state = request.getParameter("state");

    // Retrieve the value of the selected radio button
    String displayOption = request.getParameter("displayOption");

    HttpURLConnection httpURLConnection = (HttpURLConnection) new
URL(apiUrl).openConnection();
    httpURLConnection.setRequestMethod("GET");
    Gson gson = new Gson();
    InputStreamReader reader = new
InputStreamReader(httpURLConnection.getInputStream());
    Type listType = new TypeToken<List<List<String>>>(){}.getType();
    List<List<String>> data = gson.fromJson(reader, listType);
    reader.close();

    Map<String, String> infoMap = new HashMap<>();
    for (int i = 1; i < data.size(); i++) {
        List<String> stateData = data.get(i);
        if (state.equals(stateData.get(0))) {
            infoMap.put("Name", stateData.get(0));
            infoMap.put("Population", stateData.get(1));
            infoMap.put("StateID", stateData.get(2));
            break;
        }
    }

    String nextJsp = "";
    if ("bird".equals(displayOption)) {
        // get data from birdPicModel, which scrape the data from
Wikipedia, and save it into the map
        Map<String, String> birdInfoMap =
birdPicModel.doSearch(infoMap.get("Name"));
        if (birdInfoMap != null) {
            infoMap.put("birdName", birdInfoMap.get("Name"));
            infoMap.put("birdSciName", birdInfoMap.get("sciName"));
            infoMap.put("birdImgUrl", birdInfoMap.get("imgUrl"));
            infoMap.put("birdYear", birdInfoMap.get("year"));
        }
        nextJsp = "/stateInfoBird.jsp";
    } else if ("flower".equals(displayOption)) {
        // get data from birdPicModel, which scrape the data from
Wikipedia, and save it into the map
        Map<String, String> flowerInfoMap =
flowerPicModel.doSearch(infoMap.get("Name"));
        if (flowerInfoMap != null) {
            infoMap.put("flowerName", flowerInfoMap.get("Name"));
            infoMap.put("flowerSciName", flowerInfoMap.get("sciName"));
            infoMap.put("flowerImgUrl", flowerInfoMap.get("imgUrl"));
            infoMap.put("flowerYear", flowerInfoMap.get("year"));
        }
        nextJsp = "/stateInfoFlower.jsp";
    } else {
        // get data from birdPicModel, which scrape the data from
```

```
Wikipedia, and save it into the map
        Map<String, String> birdInfoMap =
birdPicModel.doSearch(infoMap.get("Name"));
        if (birdInfoMap != null) {
            infoMap.put("birdName", birdInfoMap.get("Name"));
            infoMap.put("birdSciName", birdInfoMap.get("sciName"));
            infoMap.put("birdImgUrl", birdInfoMap.get("imgUrl"));
            infoMap.put("birdYear", birdInfoMap.get("year"));
        }

        // get data from birdPicModel, which scrape the data from
Wikipedia, and save it into the map
        Map<String, String> flowerInfoMap =
flowerPicModel.doSearch(infoMap.get("Name"));
        if (flowerInfoMap != null) {
            infoMap.put("flowerName", flowerInfoMap.get("Name"));
            infoMap.put("flowerSciName", flowerInfoMap.get("sciName"));
            infoMap.put("flowerImgUrl", flowerInfoMap.get("imgUrl"));
            infoMap.put("flowerYear", flowerInfoMap.get("year"));
        }
        nextJsp = "/stateInfoBoth.jsp";
    }

    // put the data to request and send it to stateInfo
    request.setAttribute("infoMap", infoMap);
    request.setAttribute("userName", userName);
    RequestDispatcher dispatcher = request.getRequestDispatcher(nextJsp);
    dispatcher.forward(request, response);
}
```

Abstract PicModel

```
/*
 * Make an HTTP request to a given URL
 *
 * @param urlString The URL of the request
 * @return A string of the response from the HTTP GET.  This is identical
 * to what would be returned from using curl on the command line.
 */
protected String fetch(String urlString) {
    String response = "";
    try {
        URL url = new URL(urlString);
        /*
         * Create an HttpURLConnection.  This is useful for setting
headers
         * and for getting the path of the resource that is returned
(which
         * may be different than the URL above if redirected).
         * HttpsURLConnection (with an "s") can be used if required by the
site.
         */
        HttpURLConnection connection = (HttpURLConnection)
url.openConnection();
```

```
        // Read all the text returned by the server
        BufferedReader in = new BufferedReader(new
InputStreamReader(connection.getInputStream(), "UTF-8"));
        String str;
        // Read each line of "in" until done, adding each to "response"
        while ((str = in.readLine()) != null) {
            // str is one line of text readLine() strips newline
characters
            response += str;
        }
        in.close();
    } catch (IOException e) {
        System.out.println("Eeek, an exception");
        // Do something reasonable.  This is left for students to do.
    }
    return response;
}
```

BirdPicModel

```
/**
 * scrape data on Wikipedia using the searchTag, which is state, that user
chose
 * @param searchTag The tag of the photo to be searched for.
 * photo requested.
 */
@Override
public Map<String, String> doSearch(String searchTag) throws IOException {
    /*
     * URL encode the searchTag, e.g. to encode spaces as %20
     *
     * There is no reason that UTF-8 would be unsupported.  It is the
     * standard encoding today.  So if it is not supported, we have
     * big problems, so don't catch the exception.
     */
    Map<String, String> result = new HashMap<>();

    // Read the html content using Jsoup
    Document doc = Jsoup.connect(webURL).get();
    Elements rows = doc.select("tr:has(th:contains(" + searchTag + "))");
    for (Element row : rows) {
        Elements tds = row.select("td");
        for (int i=0; i<tds.size(); i++) {
            Element td = tds.get(i);
            // Process each <td> as needed
            switch (i) {
                // the first column is the bird's name, the second column
is its scientific name, and the last column is the year
                case 0:
                    result.put("Name", td.text());
                    break;
                case 1:
                    result.put("sciName", td.text());
                    break;
```

```
                case 3:
                    if (td.text().length() >= 4) {
                        result.put("year", td.text().substring(0, 4));
                    } else {
                        result.put("year", td.text());
                    }
                    break;
            }
            // If the <td> contains an <img>, print its 'src' attribute
            Elements images = td.select("img");
            for (Element img : images) {
                // the third column, also the only column contains <img>,
is the image of the state bird
                String src = img.attr("src");
                result.put("imgUrl", "https:"+src);
            }
        }
    }

    return result;
}
```

FlowerPicModel

```
/**
 * scrape data on Wikipedia using the searchTag, which is state, that user
chose
 *
 * @param searchTag The tag of the photo to be searched for.
 *                  photo requested.
 */
@Override
public Map<String, String> doSearch(String searchTag) throws IOException {
    /*
     * URL encode the searchTag, e.g. to encode spaces as %20
     *
     * There is no reason that UTF-8 would be unsupported.  It is the
     * standard encoding today.  So if it is not supported, we have
     * big problems, so don't catch the exception.
     */
    Map<String, String> result = new HashMap<>();

    // Read the html content using Jsoup
    // The searchTag is the state name
    Document doc = Jsoup.connect(webURL).get();
    Elements rows = doc.select("tr:has(td:contains(" + searchTag + "))");
    // Process each <tr> as needed
    for (Element row : rows) {
        Elements tds = row.select("td");
        for (int i=0; i<tds.size(); i++) {
            Element td = tds.get(i);
            // Process each <td> as needed
            switch (i) {
                // the first column is the bird's name, the second column
```

```
is its scientific name, and the last column is the year
                case 1:
                    result.put("Name", td.text());
                    break;
                case 2:
                    result.put("sciName", td.text());
                    break;
                case 4:
                    if (td.text().length() >= 4) {
                        result.put("year", td.text().substring(0, 4));
                    } else {
                        result.put("year", td.text());
                    }
                    break;
            }
            // If the <td> contains an <img>, print its 'src' attribute
            Elements images = td.select("img");
            for (Element img : images) {
                // the third column, also the only column contains <img>,
is the image of the state bird
                String src = img.attr("src");
                result.put("imgUrl", "https:"+src);
            }
        }
    }

    return result;
}
```