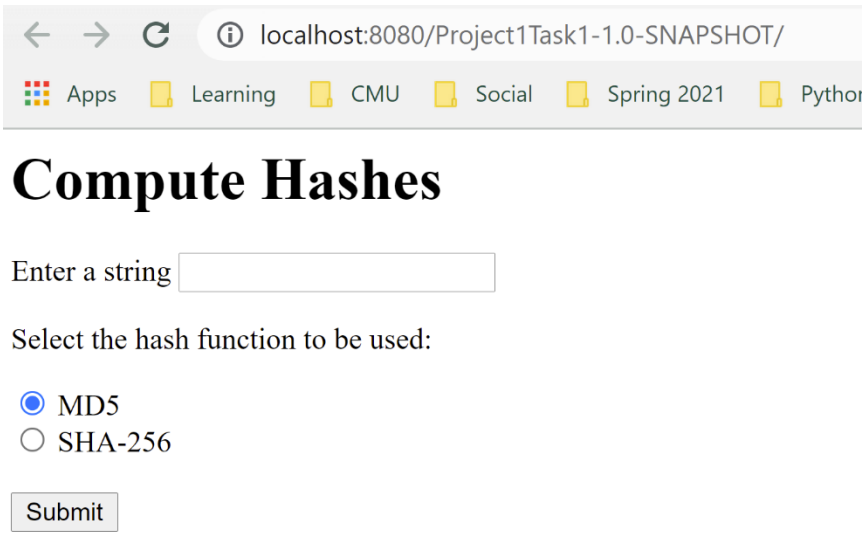
PROJECT

1. Task 1:

a) Result screenshots:



On running the server, the browser loads the following screen for the user:

Figure 1: Landing page

MD5 is the default hash function. The user can input a string of his choice in the input box, select the hash function of his choice and click on the submit button.

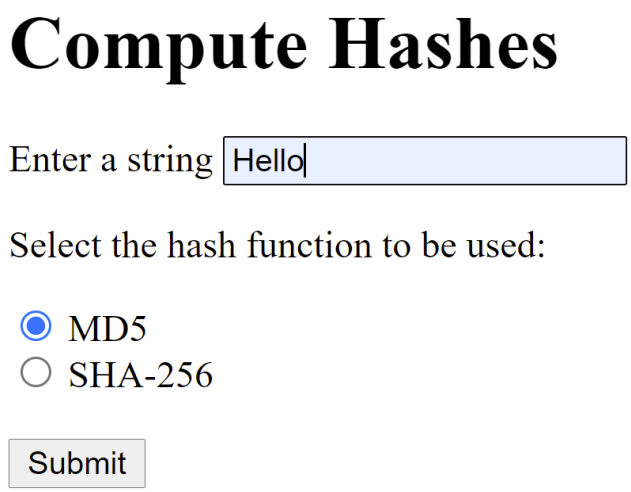


Figure 2: User inputs string "Hello" and selects default MD5 hashing

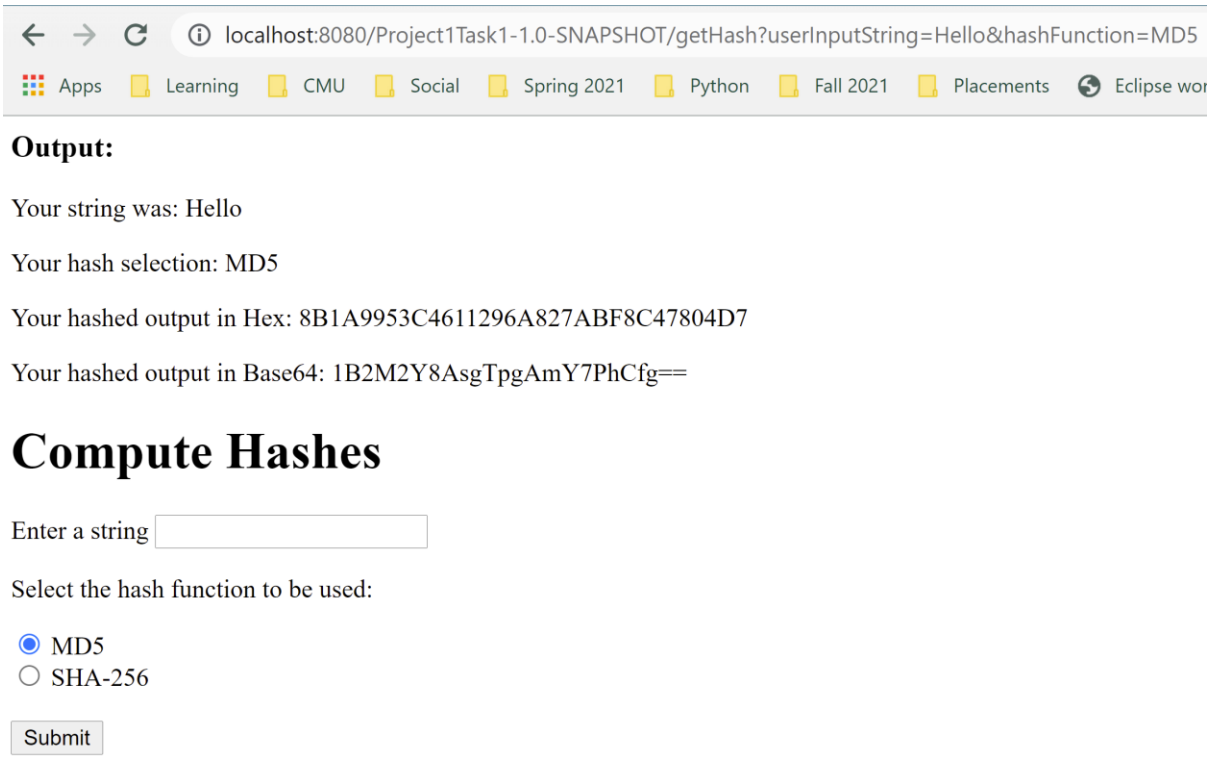


Figure 3: Hex and Base64 encoded output for string "Hello" hashed using MD5

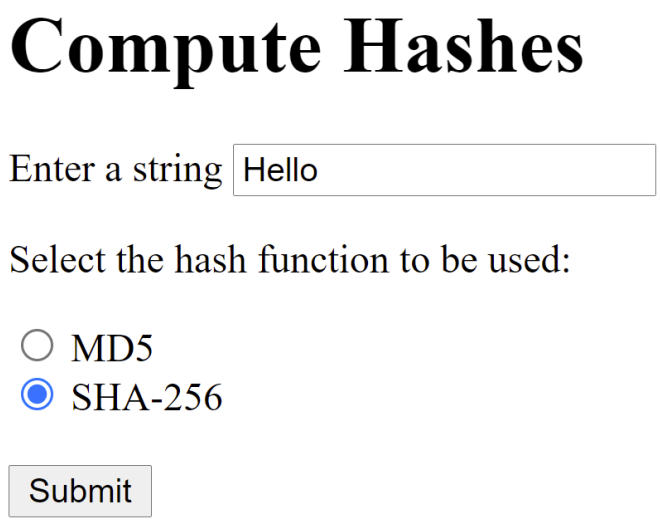


Figure 4: User inputs "Hello" and selects SHA-256 hashing

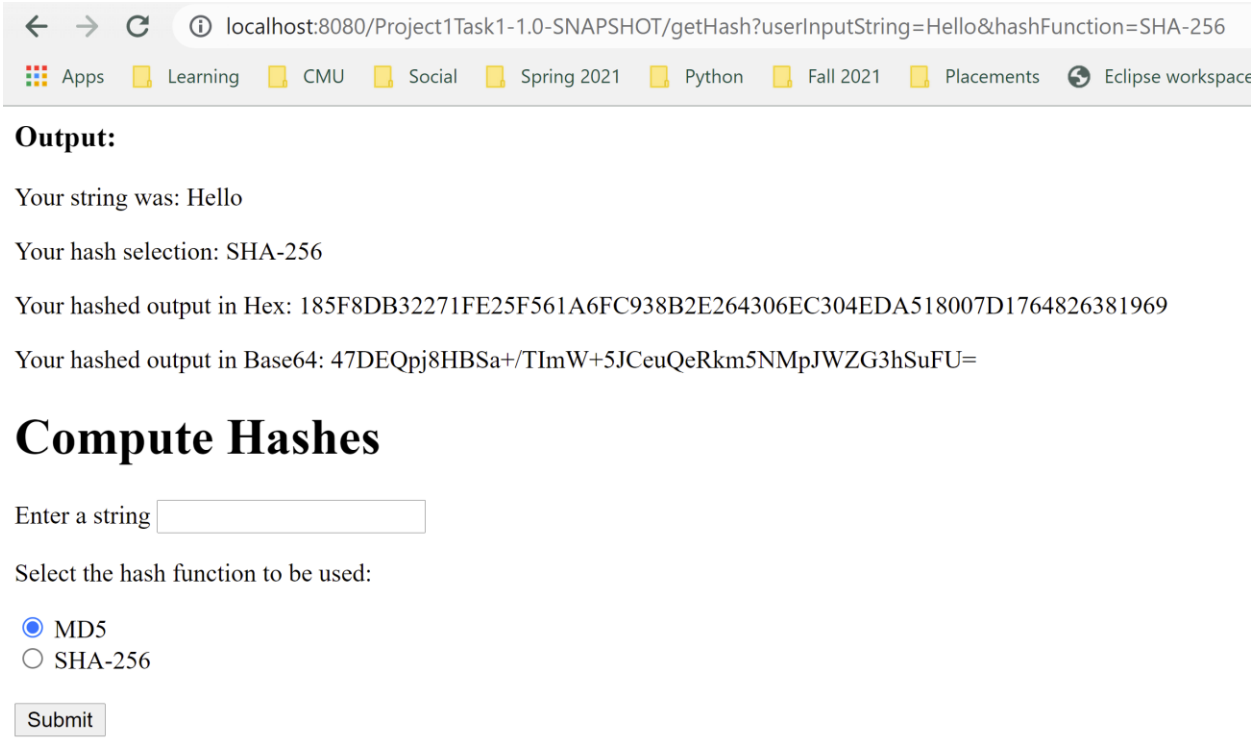


Figure 5: Hex and Base64 encoded output for string "Hello" hashed using SHA-256

b) Code snippets:

```java
public class ComputeHashesModel {

    // method returns array containing encoded strings. If exception is encountered returns null
    public String [] computeHashAndEncoding(String userInput, String hashType) {
        try {
            // code from Lab1 and Project1 instructions
            MessageDigest md = MessageDigest.getInstance(hashType);
            md.update(userInput.getBytes());
            String updatedHexString = jakarta.xml.bind.DatatypeConverter.printHexBinary(md.digest());
            String updatedBase_64_String = jakarta.xml.bind.DatatypeConverter.printBase64Binary(md.digest());
            // returns String array containing hex and base64 encoded strings.
            return new String[] {updatedHexString,updatedBase_64_String};
        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
        return null;
    }
}
```

Figure 6: Computation of hashing and encoding

2. Task 2

a) Result screenshots:

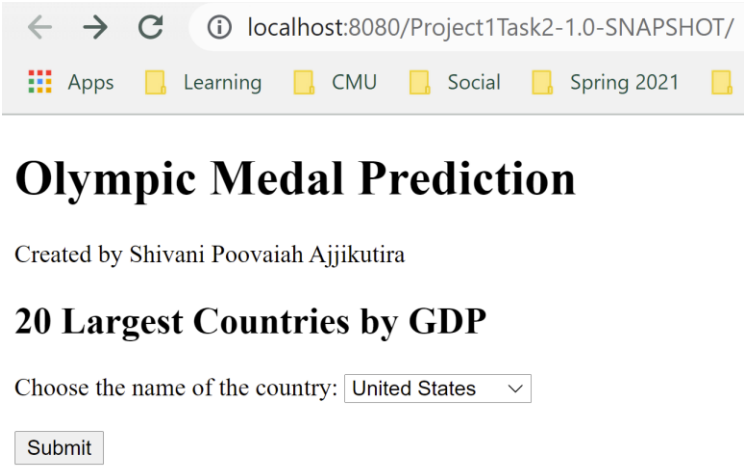


Figure 6: Input page of Task 2

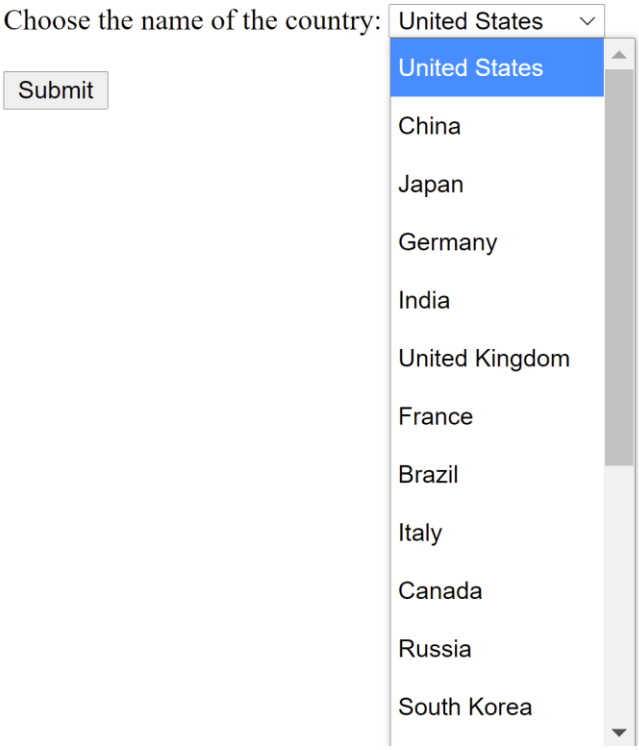


Figure 7: Dropdown menu of countries

localhost:8080/Project1Task2-1.0-SNAPSHOT/getCountryData?country=United+States

Apps Learning CMU Social Spring 2021 Python Fall 2021 Placements

Country: United States

GDP: $19,485,394,000,000

Population: 325,084,756

Credit: https://www.worldometers.info/gdp/gdp-by-country/

Gold: 39

Silver: 41

Bronze: 33

Weighted Medal Count: 113

Credit: https://olympics.com/tokyo-2020/olympic-games/en/results/all-sports/medal-standings.htm

Expected Medal Count: 497.90

Credit: Towing Icebergs, Falling Dominoes, and Other Adventures in Applied Mathematics by Robert B. Banks

Flag:

Continue

Figure 8: Output for United States

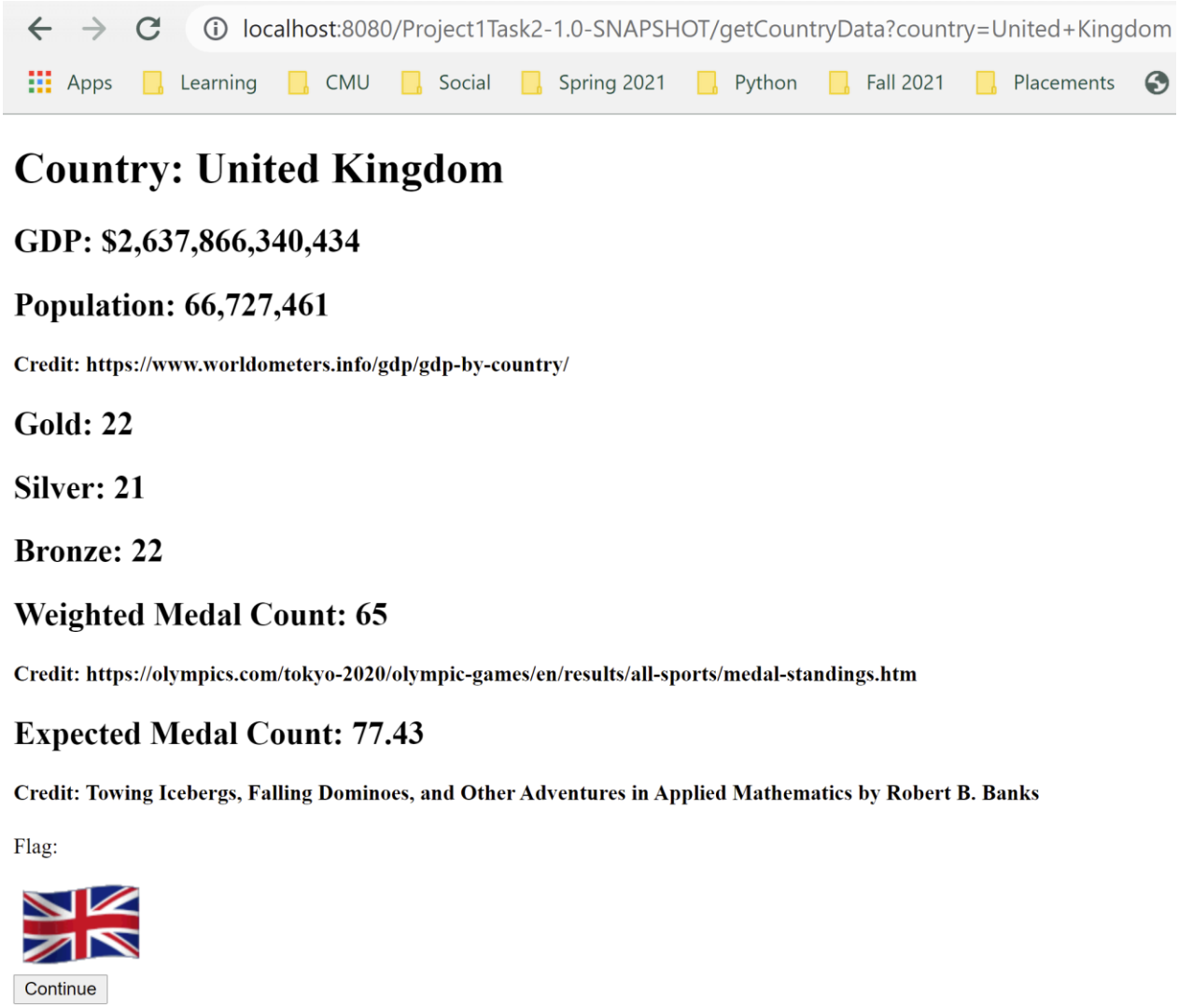


Figure 9: Output for Great Britain/ United Kingdom

b) Code snippets:

```java
public String[] getGDPAndPopulation(String searchCountry) throws IOException {
    // gets html from the url
    Document doc = Jsoup.connect( url: "https://www.worldometers.info/gdp/" +
            "gdp-by-country/").get();
    // finds table with id "example2" in element tree
    Element table = doc.getElementById("example2");
```

```java
    if (table != null) {
        // gets all table rows
        Elements rows = table.getElementsByTag( tagName: "tr");
        // loops through rows and find data cell containing country name
        for (Element row : rows) {
            // gets all data cells
            Elements dataCells = row.getElementsByTag( tagName: "td");
            if (dataCells.size() != 0) {
                Element country = dataCells.get(1).child(0);
                // if text in element equals searchCountry returns array
                // with data
                if (country.text().equalsIgnoreCase(searchCountry)) {
                    String [] data = new String[2];
                    // stores gdp
                    data[0] = dataCells.get(2).text();
                    // stores population
                    data[1] = dataCells.get(5).text();
                    return data;
                }
            }
        }
    }
    return null;
}
```

Figure 10: Scrapping of population and GDP

```java
public String[] getMedals(String searchCountry) throws IOException{
    // replaces country name for countries having different naming convention
    if(searchCountry.equalsIgnoreCase( anotherString: "United Kingdom")) {
        searchCountry = "Great Britain";
    }  else if(searchCountry.equalsIgnoreCase( anotherString: "Russia")) {
        searchCountry = "ROC";
    } else  if(searchCountry.equalsIgnoreCase( anotherString: "South Korea")) {
        searchCountry = "Republic of Korea";
    }

    // gets html from the url
    Document doc = Jsoup.connect( url: "https://olympics.com/tokyo-2020/olympic-games/en/results/" +
            "all-sports/medal-standings.htm").get();
    // finds table with id "medal-standing-table" in element tree
    Element table = doc.getElementById("medal-standing-table");
    if(table != null) {
       // gets all table rows
        Elements rows = table.getElementsByTag( tagName: "tr");
        for (Element row : rows) {
            // gets all table data cells
            Elements dataCells = row.getElementsByTag( tagName: "td");
```

```java
if (dataCells.size() != 0) {
    // gets Country name
    String country = dataCells.get(1).child(0).child(0).text();
    // if country contains search country returns array with
    // medal data
    if (country.contains(searchCountry)) {
        String [] medals = new String[4];
        /*
        * The array stores number of gold, silver, bronze and total medals.
        * For loop is used to assign data to each index of medals array.
        * The html template is designed such that countries with
        * a positive medal count have an anchor tag inside the data cell
        * which navigates users to another page. The countries having
        * 0 medal count do not have this anchor tag. The ternary operator
        * checks if the element has an anchor tag or not and assigns text
        * to the medals array accordingly and returns it
        * */
        for(int i=0; i<medals.length; i++) {
            medals[i] = dataCells.get(2+i).childrenSize() >= 1 ?
                    dataCells.get(2+i).child(0).text() : dataCells.get(2+i).text();
        }
        return medals;
    }
}
```

Figure 11: Scrapping of medals

```java
public String calculateEstimatedMedals(String gdp, String population) {
    // removes $ and , in the gdp string
    String gdpNum = gdp.replaceAll( regex: "[$,]+", replacement: "");
    // removes , in the population string
    String popNum = population.replaceAll( regex: ",", replacement: "");
    double g = Double.parseDouble(gdpNum)/1000000000;
    double p = Double.parseDouble(popNum)/1000000;
    double estimate =  0.1*(Math.pow(p*(Math.pow(g,2)),(1.0/3.0)));
    return String.format("%.2f",estimate);
}
```

Figure 12: Getting estimated medals

```java
public String getCountryFlag(String searchCountry) throws IOException {
    // gets html from the url
    Document doc = Jsoup.connect( url: "https://commons.wikimedia.org/wiki/Animated_GIF_flags").get();
    // gets all elements with class gallerybox
    Elements allFlags = doc.getElementsByAttributeValue("class","gallerybox");
    // loop through each gallery box element of every flag
    for(Element flag: allFlags) {
        // if children of flag node contains the country name
        if(flag.child(0).child(1).child(0).text().contains(searchCountry)) {
            // get the image tag from the child node
            Element image = flag.child(0).child(0).child(0).child(0).child(0);
            // return the value of attribute src in image tag
            return image.attr( attributeKey: "src");

        }
    }
    return null;
}
```

Figure 13: Scrapping of flag

3. Task 3

a) Result screenshots:

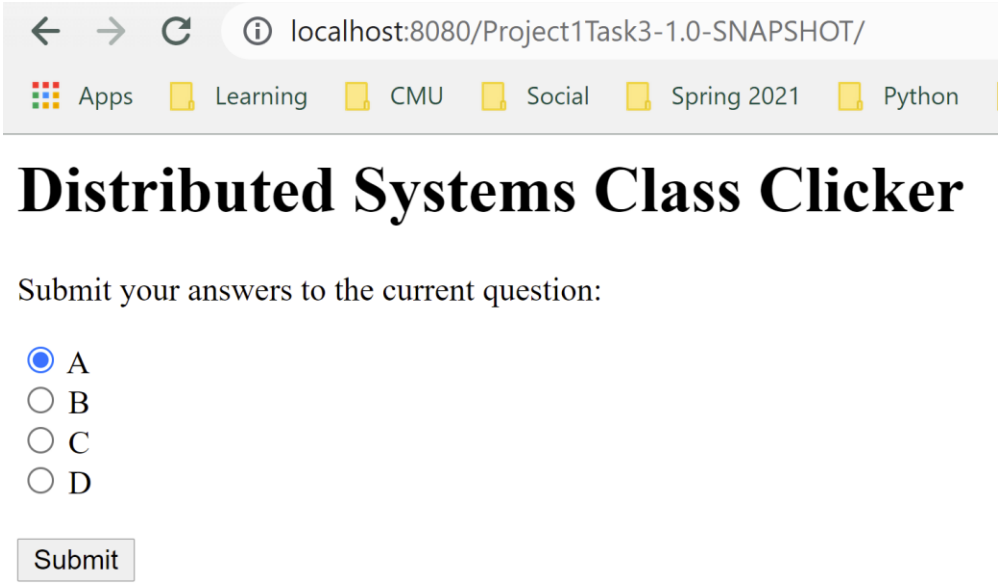


Figure 14: Input page

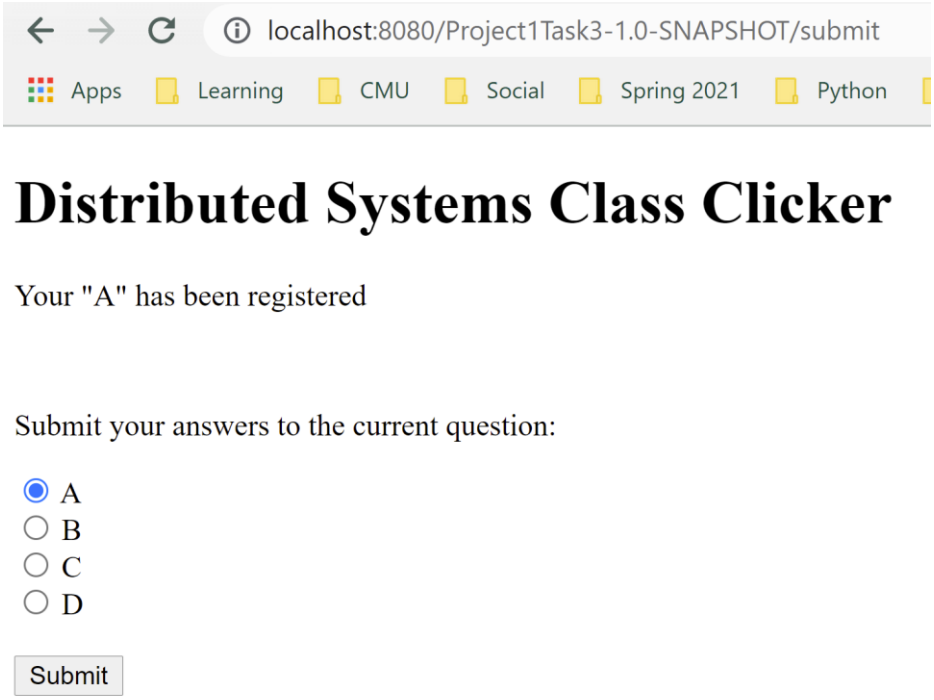
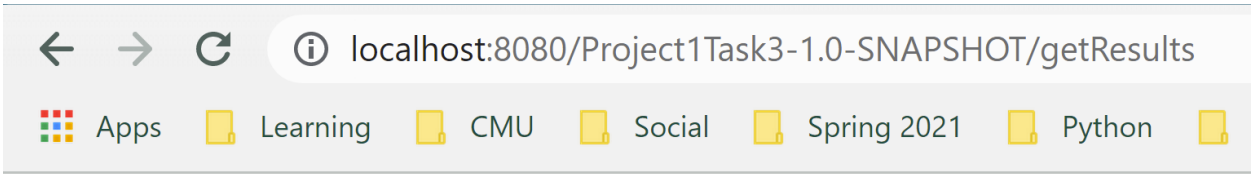


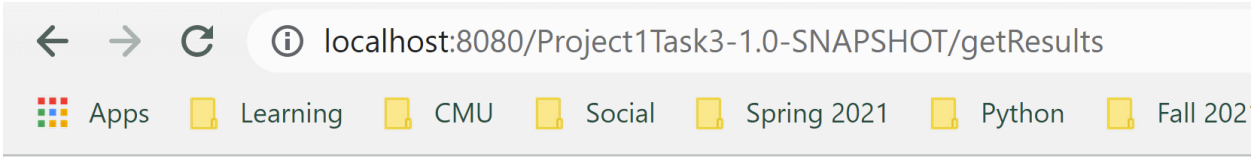
Figure 15: Output page after selection



Distributed Systems Class Clicker

There are currently no requests

Figure 16: No selection output page



Distributed Systems Class Clicker

The results from the survey are as follows:

A: 1

B: 0

C: 0

D: 0

Figure 17: Results page after selection

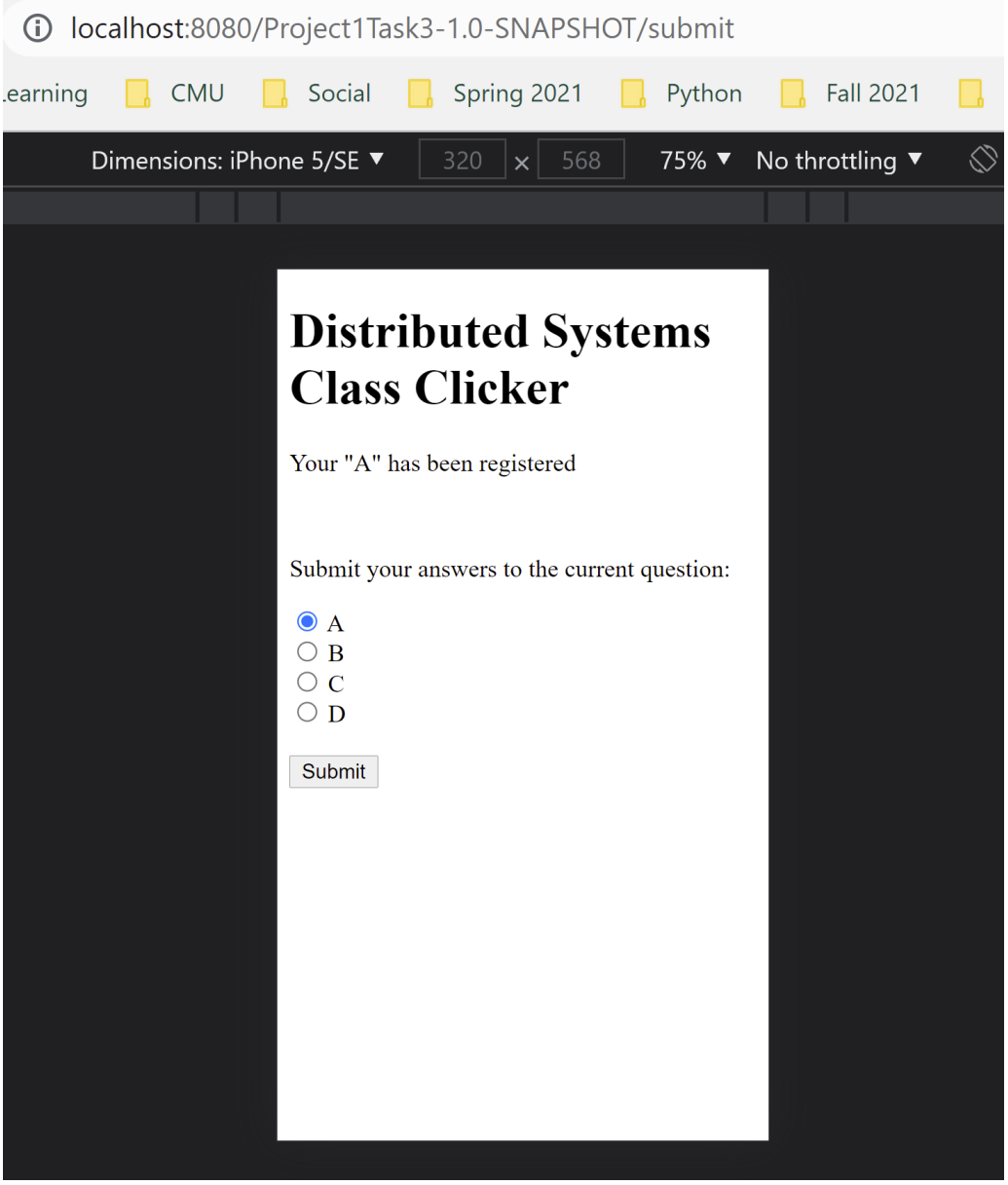


Figure 18: Mobile output after one vote

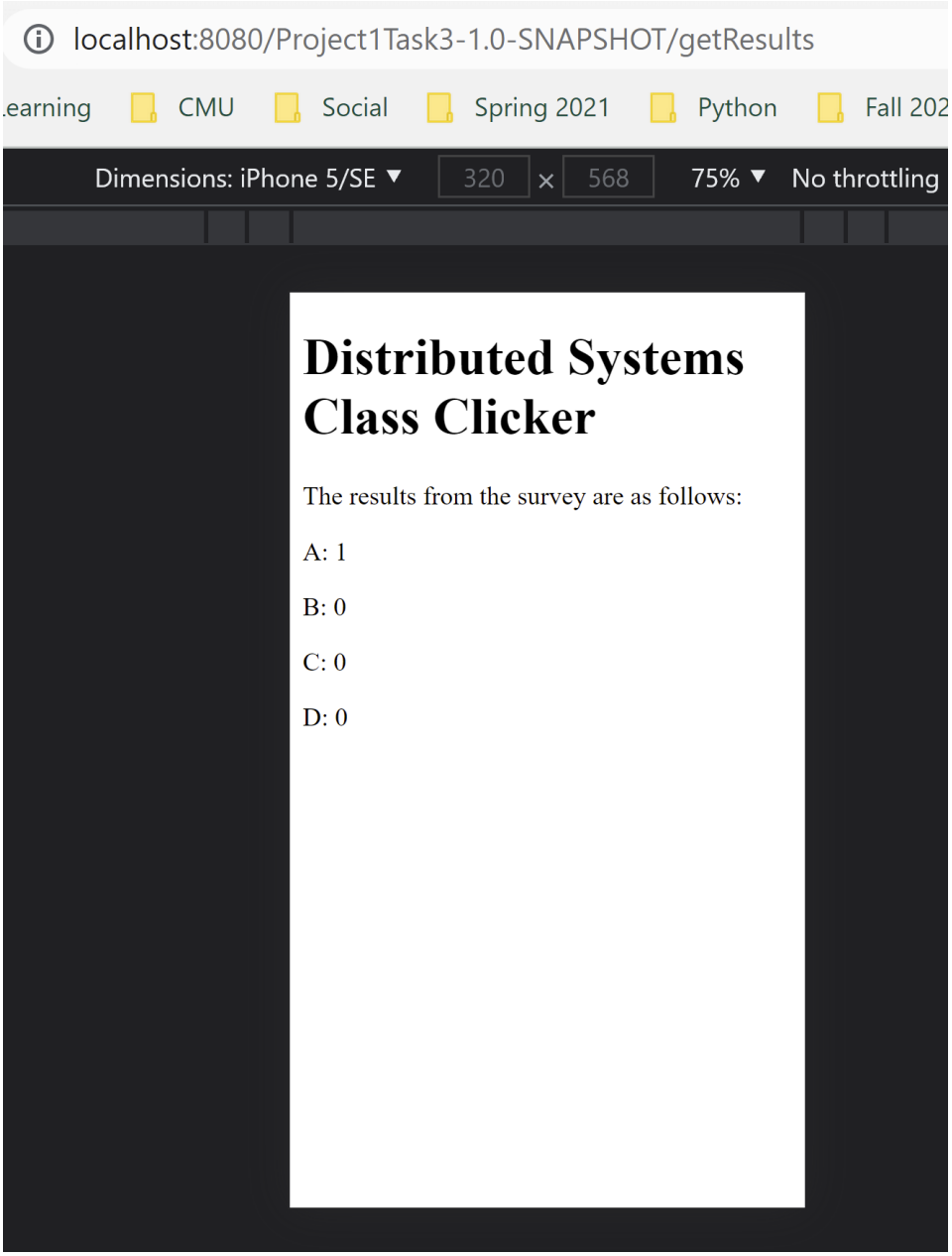


Figure 19: Mobile output for results page

b) Code snippets:

```java
static int [] tally;

// constructor to initialize tally array
public ClickerModel() {
    tally = new int[4];
}

// update count after switch case check
public void setSectionCount(String selected) {
    switch (selected) {
        case "A":
            tally[0] += 1;
            break;
        case "B":
            tally[1] += 1;
            break;
        case "C":
            tally[2] += 1;
            break;
        default:
            tally[3] += 1;
            break;
    }
}
```

```java
    // The servlet replies to the HTTP POST requests using this doPost method
    public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletExc
        // sets docType based on device
        setDocType(request);
        String input = request.getParameter( s: "section");
        sm.setSectionCount(input);
        request.setAttribute( s: "selected",input);
        if(request.getServletPath().equals("/submit")) {
            request.getRequestDispatcher( s: "index.jsp").forward(request,response);
        }
    }
}
```

```java
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException, Servlet
        // sets docType based on device
        setDocType(request);
        // gets option selection count in the form of an array from Model
        int [] tally = ClickerModel.tally;
        /*
        * If tally is not null, loop through tally array and check if
        * items on the array are equal to 0. If yes increment. If all
        * items are 0 then noSelection is equal to 4. If so, set attribute
        * tally "to noSelection" else set it to "selected". Also, if all
        * four are not 0, set attributes aCount, bCount, cCount, dCount
        * respectively from tally array.
        * */
        if(tally!= null) {
            int noSelection=0;
            for(int i=0; i<tally.length; i ++) {
                if(tally[i] == 0) {
                    noSelection++;
                }
            }
```

```
        if(noSelection != 4) {
            request.setAttribute( s: "tally", o: "selected");
            request.setAttribute( s: "aCount", tally[0]);
            request.setAttribute( s: "bCount", tally[1]);
            request.setAttribute( s: "cCount", tally[2]);
            request.setAttribute( s: "dCount", tally[3]);
        } else {
            request.setAttribute( s: "tally", o: "noSelection");
        }
    } else {
        request.setAttribute( s: "tally",  o: "noSelection");
    }
    // if url pattern is /submit pass the control to index.jsp View
    if(request.getServletPath().equals("/submit")) {
        request.getRequestDispatcher( s: "index.jsp").forward(request,response);
    } else {  // pass the control to result.jsp View
        request.getRequestDispatcher( s: "result.jsp").forward(request,response);
        sm = new ClickerModel();
    }

}
```