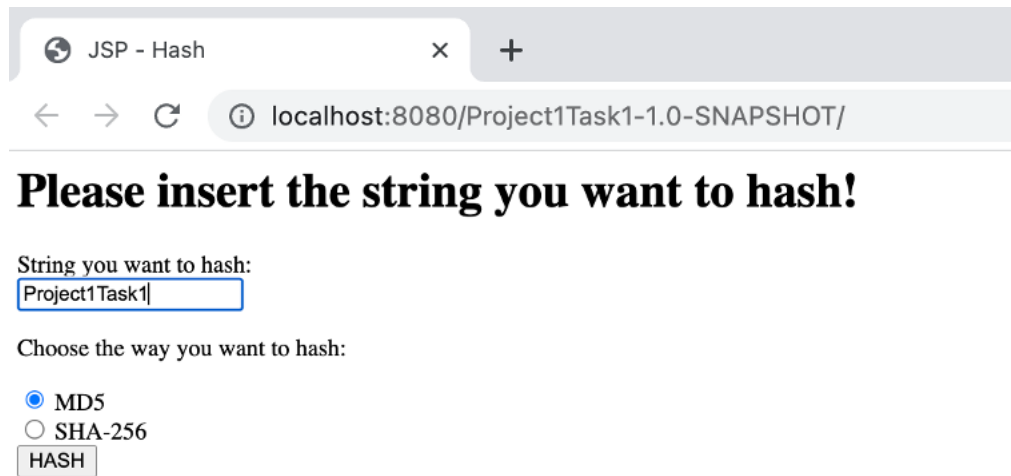
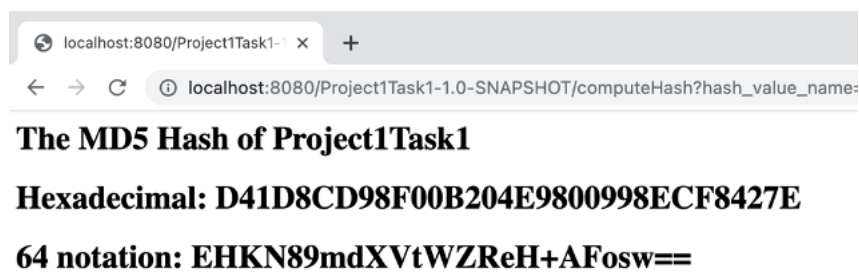


# Task 1

## Screen Shots



A screenshot of a web browser window. The address bar shows 'localhost:8080/Project1Task1-1.0-SNAPSHOT/'. The page title is 'JSP - Hash'. The main content area has a heading 'Please insert the string you want to hash!'. Below it, a text input field contains 'Project1Task1'. Underneath, there's a section 'Choose the way you want to hash:' with two radio buttons: 'MD5' (selected) and 'SHA-256'. At the bottom of this section is a button labeled 'HASH'.



A screenshot of a web browser window. The address bar shows 'localhost:8080/Project1Task1-1.0-SNAPSHOT/computeHash?hash\_value\_name=Project1Task1&hash\_way=MD5'. The page title is 'The MD5 Hash of Project1Task1'. The main content area displays the results: 'Hexadecimal: D41D8CD98F00B204E9800998ECF8427E' and '64 notation: EHKN89mdXVtWZReH+AFosw=='. There is a search icon and a share icon in the top right corner of the browser window.



A screenshot of a web browser window. The address bar shows 'localhost:8080/Project1Task1-1.0-SNAPSHOT/computeHash?hash\_value\_name=Project1Task1&hash\_way=SHA-256'. The page title is 'The SHA-256 Hash of Project1Task1'. The main content area displays the results: 'Hexadecimal: E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855' and '64 notation: igS0tTwF39OKb52kpH+jUUw05GvLpQJOXpRF/kXY/xA='. There is a search icon and a share icon in the top right corner of the browser window.

## Code Snippet:

### Hash way

```
String hash_way = request.getParameter("hash_way");
```

```
try {  
    String hash_hexadecimal;  
    String hash_64notation;  
    MessageDigest md;  
    if(hash_way.equals("SHA-256")){  
        // Code for SHA-265  
        md = MessageDigest.getInstance("SHA-256");
```

```
}

else{

    // Code for MD5

    md = MessageDigest.getInstance("MD5");

}
```

## Task 2

### Screen Shots

← → ↻ ⓘ localhost:8080/Project1Task2-1.0-SNAPSHOT/

## State Information

Create by Leo Lin

## U.S. States

Choose a state:

Alabama ▼

Submit

### State Information

Create by Leo Lin

### U.S. States

Choose a state:

✓ Alabama  
Alaska  
Arizona  
Arkansas  
California  
Colorado  
Connecticut  
Delaware  
Florida  
Georgia  
Hawaii  
Idaho  
Illinois  
Indiana  
Iowa  
Kansas  
Kentucky  
Louisiana  
Maine  
Maryland  
Massachusetts  
Michigan  
Minnesota  
Mississippi  
Missouri  
Montana  
Nebraska  
Nevada  
New Hampshire  
New Jersey  
New Mexico  
New York  
North Carolina  
North Dakota  
Ohio  
Oklahoma  
Oregon  
▼

State: Pennsylvania

**Population: 13002700**

**Nickname: Keystone State**

**Capital: Harrisburg**

**Song: Pennsylvania**

**Flower:**



**Credit: <https://statesymbolsusa.org/categories/flower>**

**Flag:**



**Credit: <https://states101.com/flags>**

Select another state.

Continue

State: New York

**Population: 20201249**

**Nickname: Empire State**

**Capital: Albany**

**Song: I love New York**

**Flower:**



**Credit: <https://statesymbolsusa.org/categories/flower>**

**Flag:**



**Credit: <https://states101.com/flags>**

Select another state.

Continue

## Code Snippet

### Get population api

```
public String getPopulation(String searchTag){
    String stateCode = fips_code.get(searchTag);
    String uri =
        "https://api.census.gov/data/2020/dec/pl?
get=NAME,P1_001N&for=state:"
        + stateCode
        + "&key=a5406f80f9aefd5891b625e07ca533e36106e6f5";
    String response = fetch(uri);
    String[] info = response.split(",");
    String population = info[4].replace("\\"", "");
    return population;
}
```

### Scrape nickname

```
public String getNickName(String searchTag){
    int cutLeft = nickNameResponse.indexOf(searchTag + "</a></td><td>") +
searchTag.length() + 13;
    int cutRight = nickNameResponse.indexOf("</td>", cutLeft);
    String nickname = nickNameResponse.substring(cutLeft, cutRight);
    return nickname;
}
```

### Scrape Capital

```
public String getCapital(String searchTag){
    int cutLeft = capitalResponse.indexOf(searchTag + " (") +
searchTag.length() + 2;
    int cutRight = capitalResponse.indexOf(")", cutLeft);
    String capital = capitalResponse.substring(cutLeft, cutRight);
    return capital;
}
```

```
}
```

## Scrape Songname

```
public String getSong(String searchTag){  
    int cut1 = songResponse.indexOf(searchTag + "</dt><dd><a")  
+searchTag.length() + 11;  
    int cutLeft = songResponse.indexOf(">", cut1) + 1;  
    int cutRight = songResponse.indexOf("<", cutLeft);  
    String song = songResponse.substring(cutLeft, cutRight);  
    return song;  
}
```

## Scrape Flower picture url

```
public String doFlowerSearch(String searchTag)  
    throws UnsupportedOperationException {  
    /*  
        * Getting the url from html page, the reason to use another string  
variable  
        * is if there is space in the string, we need to replace it with  
hyphen to match the html  
    */  
    String hyphen = searchTag.replace(' ', '-');  
    int cut1 = flowerResponse.indexOf(hyphen.toLowerCase() + "/state-  
flower");  
    int cutLeft = flowerResponse.indexOf("src=", cut1)+5;  
    int cutRight = flowerResponse.indexOf("width=", cutLeft) -2;  
  
    String flowerURL = flowerResponse.substring(cutLeft, cutRight);  
    return flowerURL;  
}
```

## Scrape Flag picture url

```
public String doFlagSearch(String searchTag)
    throws UnsupportedOperationException {
    /*
     * Getting the url from html page, the reason to use another string
    variable
     * is if there is space in the string, we need to replace it with
    hyphen to match the html
     */
    String hyphen = searchTag.replace(' ', '-');
    int cut1 = flagResponse.indexOf("flags/" + hyphen.toLowerCase());
    int cutLeft = flagResponse.indexOf("src=", cut1) + 5;
    int cutRight = flagResponse.indexOf("alt=", cutLeft) - 2;
    // https://www.states101.com

    String flagURL = "https://www.states101.com" +
    flagResponse.substring(cutLeft, cutRight);
    return flagURL;
}
```

## Task 3

### Screen Shots

← → ↻ ⓘ localhost:8080/Project1Task3-1.0-SNAPSHOT/

## 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

## Distributed Systems Class Clicker

**A: 1**

**B: 0**

**C: 0**

**D: 0**

Dimensions: Responsive ▾

400

×

779

100% ▾

⋮

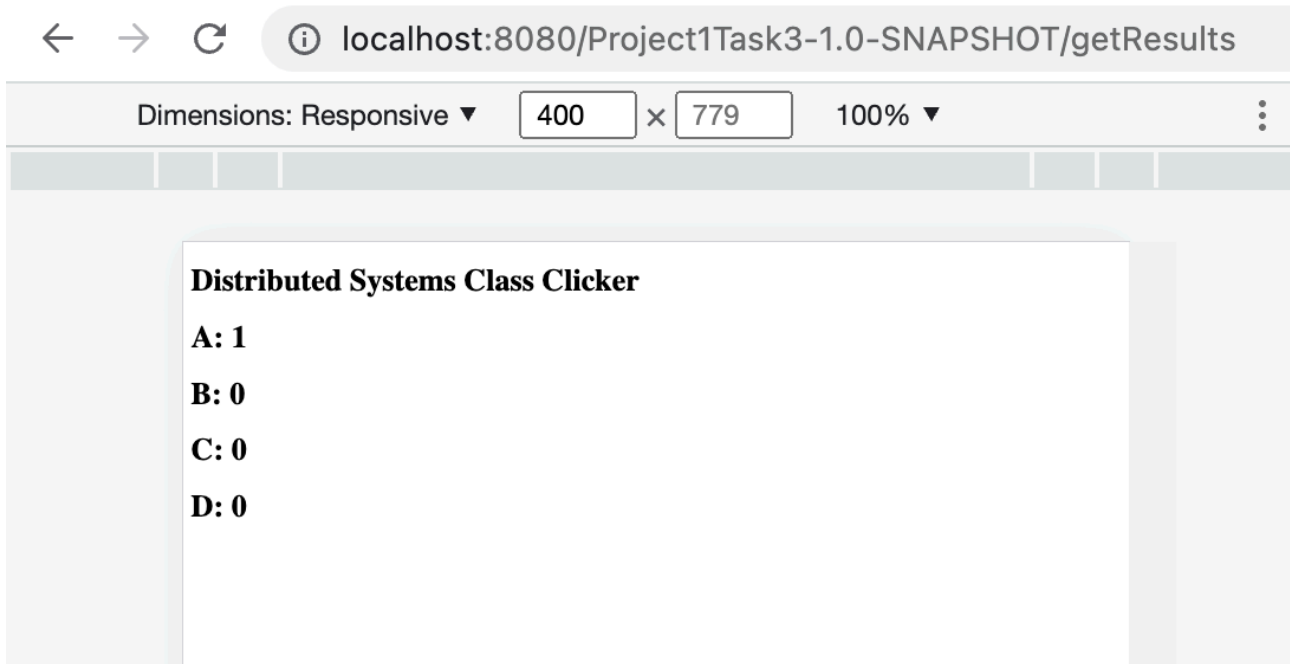
## Distributed Systems Class Clicker

Your "A" has been registered

Submit your answer to the current question:

- ☐ A
- ☐ B
- ☐ C
- ☐ D

Submit



## Code Snippet

```
String path = request.getServletPath();
if(path.equals("/getResults")){
    int numA = acm.answer.get("A");
    int numB = acm.answer.get("B");
    int numC = acm.answer.get("C");
    int numD = acm.answer.get("D");
    int sum = numA + numB + numC + numD;
    request.setAttribute("numA", numA);
    request.setAttribute("numB", numB);
    request.setAttribute("numC", numC);
    request.setAttribute("numD", numD);
    request.setAttribute("sum", sum);
    nextView = "check.jsp";
}
else{
```



```

/*
    * Check if the search parameter is present.
    * If not, then give the user instructions and prompt for a search
string.
    * If there is a search parameter, then do the search and return the
result.
*/
if (ans != null) {
    String picSize = (mobile) ? "mobile" : "desktop";
    /*
        * Attributes on the request object can be used to pass data to
        * the view. These attributes are name/value pairs, where the
name
        * is a String object. Here the pictureURL is passed to the view
        * after it is returned from the model interestingPictureSize
method.
    */
    acm.addNAnswer(ans);

    request.setAttribute("answer", ans);
    // Pass the user search string (pictureTag) also to the view.
    nextView = "result.jsp";
} else {
    // no search parameter so choose the prompt view
    nextView = "prompt.jsp";
}
}

```