AYDIN ADNAN MENDERES UNIVERSITY

ENGINEERING FACULTY

COMPUTER SCIENCE ENGINEERING DEPARTMENT

Web Interface For Wordcount On Cloud Platform

CSE423 Cloud Computing, Spring 2023/2024

TEAM 16

191805060 - Semih Utku Polat

191805061 - Salih Can Aydoğdu (Team Leader)

201805051 - Utku Enes Baki

191805063 - Emre Karataş

Lecturer:

Asst. Prof. Dr.Hüseyin ABACI

Introduction

In this Project, we built a web application that is coded with Flask Framework to perform a simple word count job. MVC structure is used in the web app. Two functions “index” and “upload” are used for the “get” and “post” methods. “index” is used for displaying web pages with the “get” method and “upload” is used with “post”. The user selects a Txt file to upload to the web app and clicks on to “Count” button to perform the word count operation. Most occurring ten words listed on the web page.

To Do

1. Write a word-counting algorithm with the Python programming language
2. Design a modern user interface with HTML5 and CSS3
3. Add upload form to user interface
4. Add a results section that prints number of counted words to the user interface
5. Make responsive the user interface
6. Create key pair on AWS to connect PuTTY to the EC2 instance
7. Create an EC2 instance that uses the Ubuntu operating system on AWS
8. Connect to the remote instance that runs on Cloud platform(entering OS@public-IPv4-DNS as hostname and uploading key to Ubuntu configurations)
9. Install the Apache2 service to see the web page on the browser
10. Build backend and router codes with the Python programming language
11. Import word count algorithm
12. Import HTML and CSS files
13. Write backend code with MVC (Model View Controller) architecture. (skip the model part because the database does not exist)
14. Test and debug the project

1. Write a word-counting algorithm with the Python programming language

```
with open('usertxtfile.txt', 'r') as file:
    data = file.read()
    words = data.split()
    word_counts = {}

    for word in words:
        # make lowercase
        word = word.lower()

        # remove punctuations
        word = word.translate(str.maketrans('', '', string.punctuation))

        # remove digits
        word = word.translate(str.maketrans('', '', string.digits))

        if (len(word) == 0):
            pass
        else:
            if word in word_counts:
                word_counts[word] += 1
            else:
                word_counts[word] = 1

    sorted_word_counts = sorted(word_counts.items(), key=lambda x: x[1], reverse=True)
```

The code opens and reads the Txt file. After reading all the words in the file, it splits them into an array. Then it reads the array word by word and normalizes it. Normalize operation: includes converting to lowercase, removing punctuation, and removing digits.

Encountered Error:

Just at this stage, we had a problem which is some of the word groups like “the 3 car” normalized to “the car”. This normalized form is ambiguous because there are two spaces. And this caused the counter to count empty words like: "“.

Solution: We fixed this problem with this part:

```
if (len(word) == 0):
    pass
else:
    if word in word_counts:
        word_counts[word] += 1
    else:
        word_counts[word] = 1
```

The code filters the word if it has no length and then forwards it.

2. Design a modern user interface with HTML5 and CSS3 (index.html)

3. Add upload form to user interface

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>2023 Cloud Final</title>
    <link rel="stylesheet" href="/static/style.css">
</head>

<body>
    <main>
        <div class="container">
            <h1>Word Counter</h1>
            <section class="table-wrapper">
                <div class="title-2">UPLOAD A TXT FILE</div>
            </section>
            <section class="upload-file">
                <form action="/upload" method="POST" enctype="multipart/form-data">
                    <input type="file" name="file">
                    <button type="submit">COUNT</button>
                </form>
            </section>
            <section class="team">
                <div class="col">
                    <div>191805060</div>
                    <div>Semih Utku Polat</div>
                </div>
                <div class="col">
                    <div>201805051</div>
                    <div>Utku Enes Baki</div>
                </div>
                <div class="col">
                    <div>191805063</div>
                    <div>Emre Karataş</div>
                </div>
                <div class="col">
                    <div>191805061</div>
                    <div>Salih Can Aydoğdu</div>
                </div>
            </section>
        </div>
    </main>

</body>

</html>
```

The user interface skeleton is written under SEO rules with HTML5. For charset, compatibility charset meta tag set to UTF-8. To make the design responsive for mobile devices there is added “viewport” meta tag. And then linked to CSS file for coloring and shaping.

The code contains an upload form that takes a Txt file from the user and posts the file to “/upload” with the help of the ‘enctype=” multipart/form-data”’ parameter. This post request will catch from the backend router and process.

Encountered Error:

Since we were constantly changing the server, the sub-domain name had changed. The "action" parameter in the form tag contained the full URL. This caused the post process to go to other sub domains.

Solution: We solved this complexity by assigning "/upload" to the "action" tag of the form element. So the form set where to go beforehand thanks to its HTTP header informations.

```
<form action="/upload" method="POST" enctype="multipart/form-data">
```

4.Add a results section that prints number of counted words to the user interface (final.html)

Part 1:

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>2023 Cloud Final</title>
    <link rel="stylesheet" href="/static/style.css">
</head>

<body>
    <main>
        <div class="container">
            <h1>Word Counter</h1>
            <section class="table-wrapper">
                <span>Top ten unique words sorted by value in descending :</span>
                <table>
                    <thead>
                        <tr>
                            <th>WORD</th>
                            <th>NUMBER</th>
                        </tr>
                    </thead>
                    <tbody>
                        <tr>
                            <td>{{data[0][0]}}</td>
                            <td>{{data[0][1]}}</td>
                        </tr>
                        <tr>
                            <td>{{data[1][0]}}</td>
                            <td>{{data[1][1]}}</td>
                        </tr>
                        <tr>
                            <td>{{data[2][0]}}</td>
                            <td>{{data[2][1]}}</td>
                        </tr>
                        <tr>
                            <td>{{data[3][0]}}</td>
                            <td>{{data[3][1]}}</td>
                        </tr>
                        <tr>
                            <td>{{data[4][0]}}</td>
                            <td>{{data[4][1]}}</td>
                        </tr>
                        <tr>
                            <td>{{data[5][0]}}</td>
                            <td>{{data[5][1]}}</td>
                        </tr>
                        <tr>
                            <td>{{data[6][0]}}</td>
                            <td>{{data[6][1]}}</td>
                        </tr>
```

Part 2:

```
                        <tr>
                            <td>{{data[5][0]}}</td>
                            <td>{{data[5][1]}}</td>
                        </tr>
                        <tr>
                            <td>{{data[6][0]}}</td>
                            <td>{{data[6][1]}}</td>
                        </tr>
                        <tr>
                            <td>{{data[7][0]}}</td>
                            <td>{{data[7][1]}}</td>
                        </tr>
                        <tr>
                            <td>{{data[8][0]}}</td>
                            <td>{{data[8][1]}}</td>
                        </tr>
                        <tr>
                            <td>{{data[9][0]}}</td>
                            <td>{{data[9][1]}}</td>
                        </tr>
                    </tbody>
                </table>
            </section>
            <section class="upload-file">
                <form action="/upload" method="POST" enctype="multipart/form-data">
                    <input type="file" name="file">
                    <button type="submit">COUNT</button>
                </form>
            </section>
            <section class="team">
                <div class="col">
                    <div>191805060</div>
                    <div>Semih Utku Polat</div>
                </div>
                <div class="col">
                    <div>201805051</div>
                    <div>Utku Enes Baki</div>
                </div>
                <div class="col">
                    <div>191805063</div>
                    <div>Emre Karataş</div>
                </div>
                <div class="col">
                    <div>191805061</div>
                    <div>Salih Can Aydoğdu</div>
                </div>
            </section>
        </div>
    </main>

</body>

</html>
```

The code has the same meta tags in "index.html". It shows the word count results after the user posts his/her text file to the backend. Backend processes and counts the words then after storing the results in an array, responds to the results to "final.html".

The code reaches responded array with double curly brackets and prints them in a table. The table shows the top 10 counted words and their counts (number of words).

2. Design a modern user interface with HTML5 and CSS3 (style.css)

Part 1:

```css
* {
    margin: 0px;
    padding: 0px;
    -webkit-box-sizing: border-box;
    -moz-box-sizing: border-box;
    box-sizing: border-box;
}

::-webkit-scrollbar {
    background-color: #333333;
    width: 12px;
}

::-webkit-scrollbar-thumb {
    background-color: #000000;
    border: 3px solid #333333;
}

html {
    font-size: 8px;
    scroll-behavior: smooth;
}

body {
    width: 100%;
    height: 100%;
    overflow-x: hidden;
    font-family: sans-serif;
    font-weight: 400;
    background-color: #0d1220;
}

main {
    width: 100%;
    height: 100%;
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
}

.container {
    height: 100%;
    width: 100%;
    padding-top: 5vh;
    padding-bottom: 5vh;
    padding-right: 1rem;
    padding-left: 1rem;
}

h1 {
    min-height: 10vh;
    font-size: 3rem;
    color: #b22400;
    margin-bottom: 10vh;
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
    box-shadow: 0 10px 15px -3px rgba(0, 0, 0, .75), 0 4px 6px -4px rgba(0, 0, 0, .75);
    padding-right: 2rem;
    padding-left: 2rem;
    border-radius: 1rem;
    border: 1px solid #f8f9fa;
```

Part 2:

```css
h1 {
    min-height: 10vh;
    font-size: 3rem;
    color: #b22400;
    margin-bottom: 10vh;
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
    box-shadow: 0 10px 15px -3px rgba(0, 0, 0, .75), 0 4px 6px -4px rgba(0, 0, 0, .75);
    padding-right: 2rem;
    padding-left: 2rem;
    border-radius: 1rem;
    border: 1px solid #f8f9fa;
}


.title-2 {
    font-size: 3rem;
    font-weight: 700;
    letter-spacing: 0.2rem;
    color: #f8f9fa;
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
}

.table-wrapper {
    min-height: 40vh;
    max-height: 40vh;
    background-color: #2C3845;
    border-top-right-radius: 1rem;
    border-top-left-radius: 1rem;
    position: relative;
    overflow-y: scroll;
}

.table-wrapper span {
    position: absolute;
    bottom: calc(100% + 2vh);
    left: 0;
    color: #f8f9fa;
    font-size: 1.2rem;
}

table {
    width: 100%;
    border-top-right-radius: 1rem;
    border-top-left-radius: 1rem;
}

th {
    border-radius: 1rem;
    width: 50%;
    padding: 1rem;
    font-size: 1.6rem;
    color: #b22400;
    background-color: #202932;
    border: 1px solid #202932;
    font-weight: 700;
    text-align: left;
}
```

Part 3:

```css
th {
    border-radius: 1rem;
    width: 50%;
    padding: 1rem;
    font-size: 1.6rem;
    color: #b22400;
    background-color: #202932;
    border: 1px solid #202932;
    font-weight: 700;
    text-align: left;
}

td {
    border-radius: 1rem;
    font-size: 1.4rem;
    color: #f8f9fa;
    padding: 1rem;
    text-align: left;
}

tr:nth-child(2n+2) td {
    background-color: #242e39;
}

.upload-file {
    min-height: 10vh;
    background-color: #f8f9fa;
    border-bottom-right-radius: 1rem;
    border-bottom-left-radius: 1rem;
}

.upload-file form {
    width: 100%;
    height: 10vh;
    display: flex;
    justify-content: center;
    align-items: center;
    padding-left: 1rem;
    padding-right: 1rem;
}

.upload-file form button {
    padding-top: 1rem;
    padding-bottom: 1rem;
    padding-left: 1.5rem;
    padding-right: 1.5rem;
    background-color: #b22400;
    color: #FCFCFD;
    font-size: 1.4rem;
    font-weight: 700;
    border: none;
    border-radius: 1rem;
    box-shadow: #2d234266 0 0.2rem 0.4rem, #2d23424d 0 0.7rem 1.3rem -0.3rem, #D6D6E7 0 -0.3rem 0 inset;
    cursor: pointer;
    -webkit-transition-property: all;
    -moz-transition-property: all;
    -o-transition-property: all;
    transition-property: all;
    -webkit-transition-duration: 0.2s;
    -moz-transition-duration: 0.2s;
    -o-transition-duration: 0.2s;
    transition-duration: 0.2s;
    -webkit-transition-timing-function: ease;
    -moz-transition-timing-function: ease;
```

5. Make responsive the user interface

Part 4:

```css
.upload-file form button:focus {
    box-shadow: #D6D6E7 0 0 0 0.1rem inset, #2d234266 0 0.2rem 0.4rem, #2d23424d 0 0.7rem 1.3rem -0.3rem, #D6D6E7 0 -0.3rem 0 inset;
}

.upload-file form button:hover {
    box-shadow: #2d234266 0 0.4rem 0.8rem, #2d23424d 0 0.7rem 1.3rem -0.3rem, #D6D6E7 0 -0.3rem 0 inset;
    transform: translateY(-0.2rem);
}

.upload-file form button:active {
    box-shadow: #D6D6E7 0 0.3rem 0.7rem inset;
    transform: translateY(-0.2rem);
}

.team {
    min-height: 10vh;
    margin-top: 10vh;
}

.team .col {
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
    font-size: 1.4rem;
    line-height: 1.4;
    color: #f8f9fa;
    box-shadow: 0 10px 15px -3px rgba(0, 0, 0, .75), 0 4px 6px -4px rgba(0, 0, 0, .75);
    padding: 1rem;
    border-radius: 1rem;
    border: 1px solid #f8f9fa;
    margin-bottom: 2rem;
}

@media only screen and (min-width: 768px) {
    body {
        height: 100vh;
    }

    .team {
        display: flex;
        justify-content: space-between;
        align-items: center;
    }

    .team .col {
        width: 22%;
        margin-bottom: 0;
    }
}

@media only screen and (min-width: 992px) {
    .container {
        padding-right: 0;
        padding-left: 0;
        width: 60%;
    }

    .upload-file form button {
        padding-top: 1.5rem;
        padding-bottom: 1.5rem;
        padding-left: 2rem;
        padding-right: 2rem;
    }
```

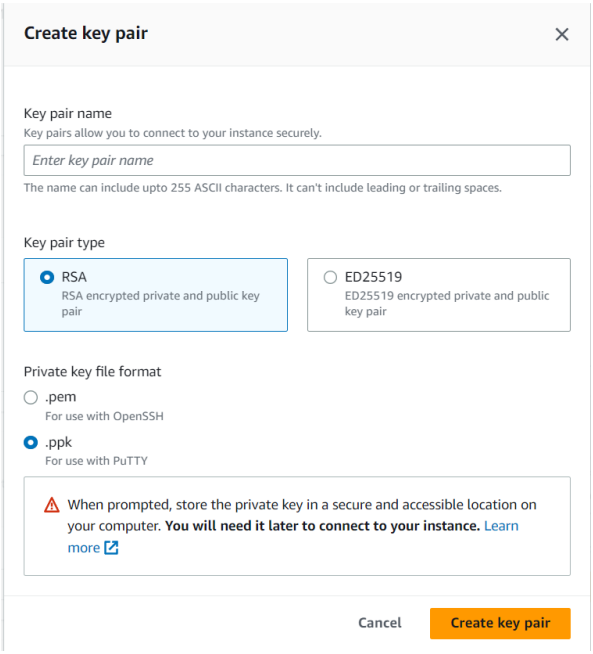
Part 5:

```css
@media only screen and (min-width: 992px) {
    .container {
        padding-right: 0;
        padding-left: 0;
        width: 60%;
    }

    .upload-file form button {
        padding-top: 1.5rem;
        padding-bottom: 1.5rem;
        padding-left: 2rem;
        padding-right: 2rem;
    }
}

@media only screen and (min-width: 1200px) {
    html {
        font-size: 12px;
    }
}

@media only screen and (min-width: 1680px) {
    html {
        font-size: 13px;
    }
}

@media only screen and (min-width: 2050px) {
    html {
        font-size: 18px;
    }
}

@media only screen and (min-width: 2400px) {
    html {
        font-size: 20px;
    }
}
```

The CSS codes target a modern and good-looking user interface with a responsive design for those using mobile or tablet devices.

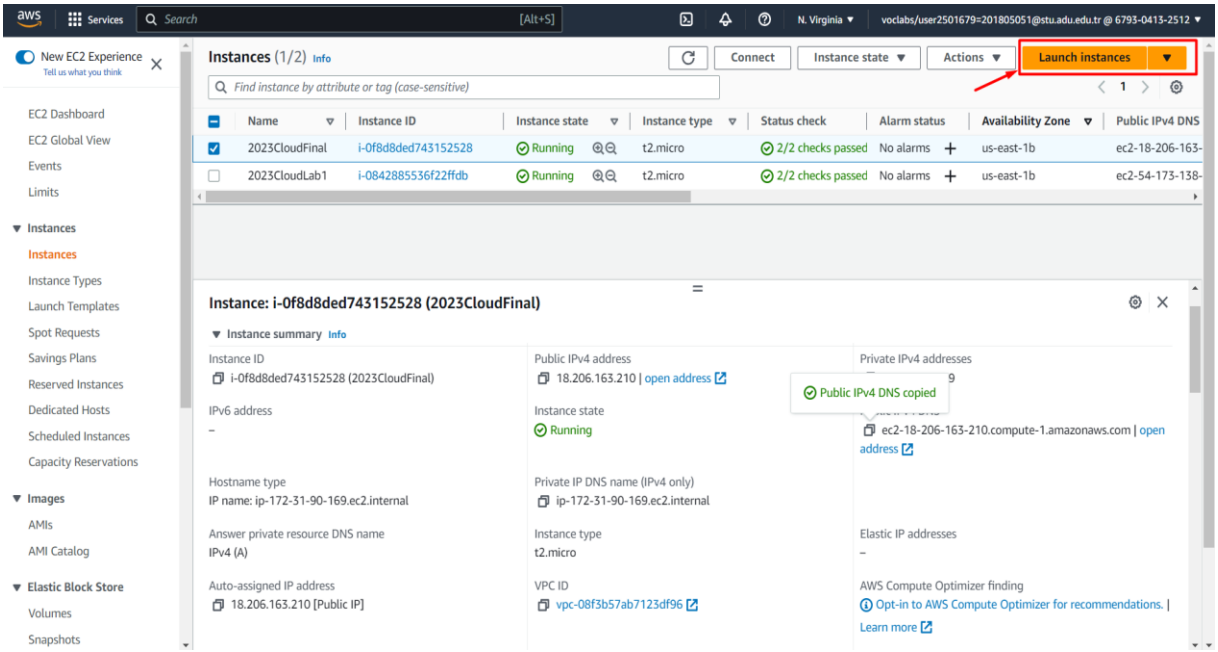
6. Create key pair on AWS to connect PuTTY to the EC2 instance

In this section, we created file which has .ppk extension for PuTTY.



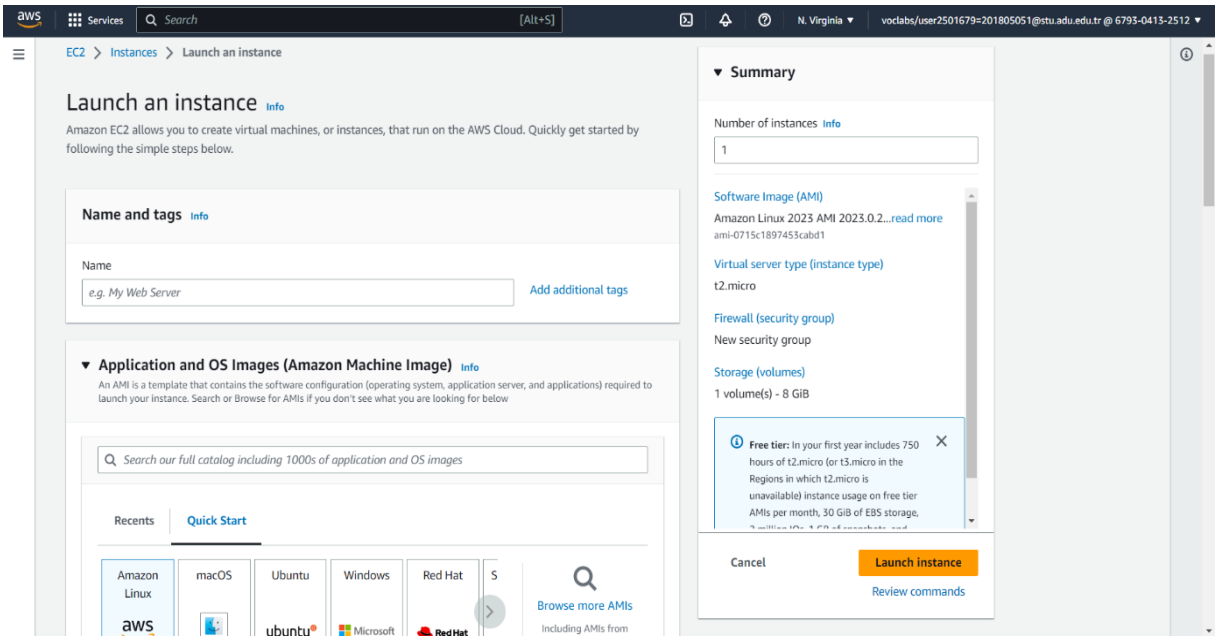
7. Create an EC2 instance that uses the Ubuntu operating system on AWS

We clicked "Launch Instances" button then...



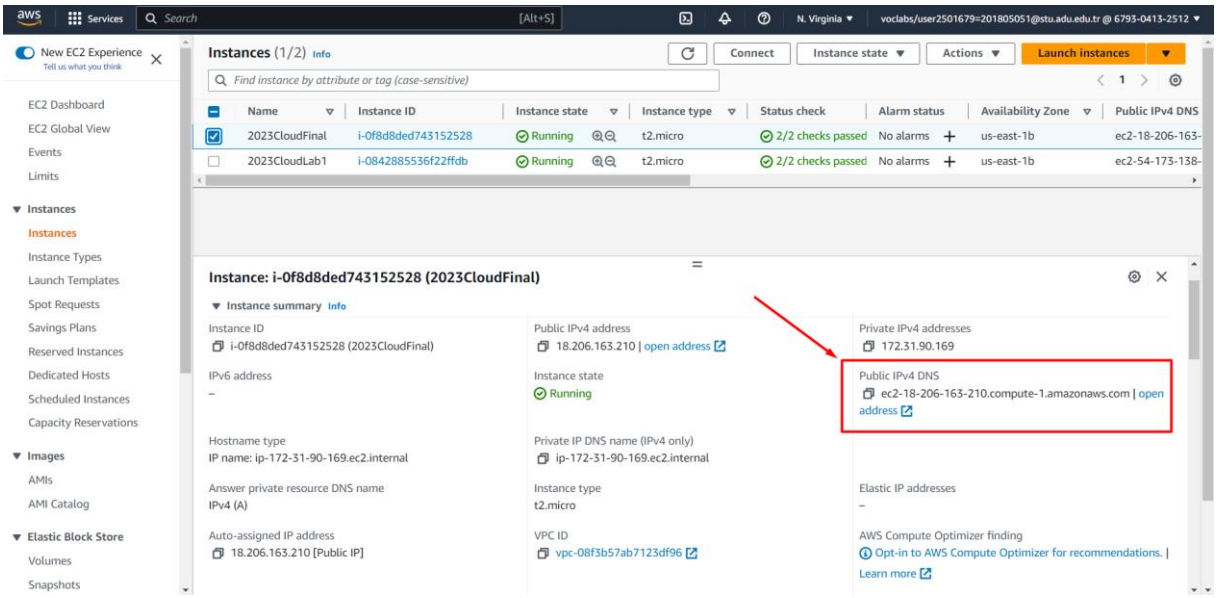
...we launched instance from the orange button after making:

- Selecting instance type,
- Selecting UBUNTU server 22.04 LTS
- Selecting t2.micro instance type (free)
- Selecting key-pair
- Doing network settings
    - Default 22 port already setted. We set extra 80 and 5000 ports. (80 port for web server. 5000 port for Flask)

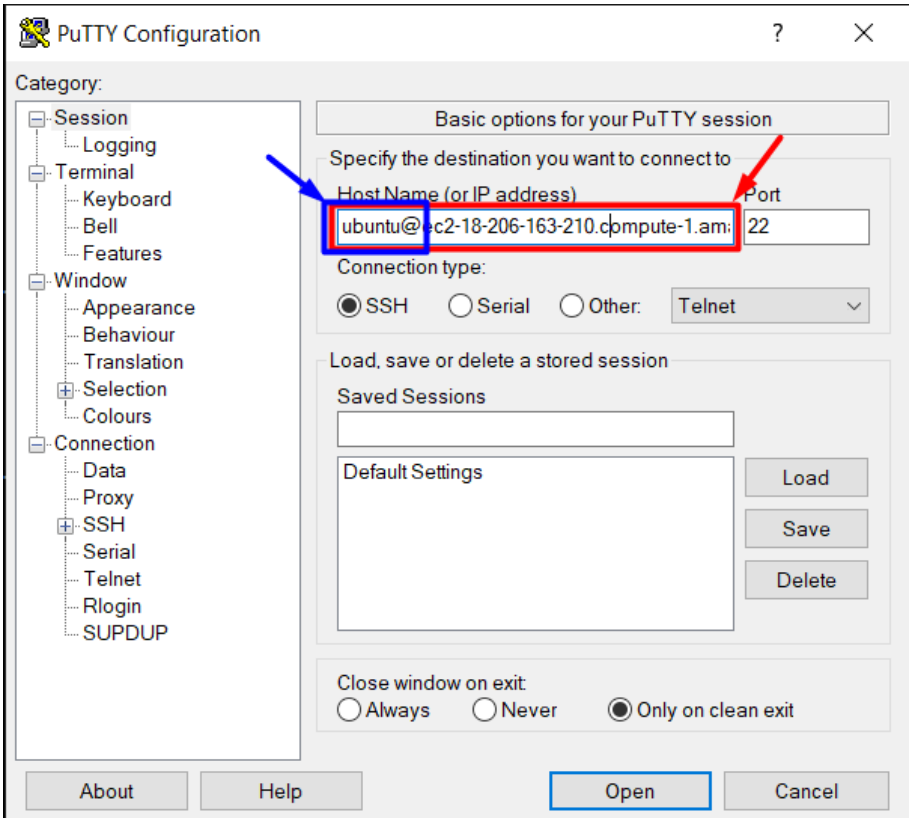


8. Connect to the remote instance that runs on Cloud platform(entering OS@public-IPv4-DNS as hostname and uploading key to Ubuntu configurations)

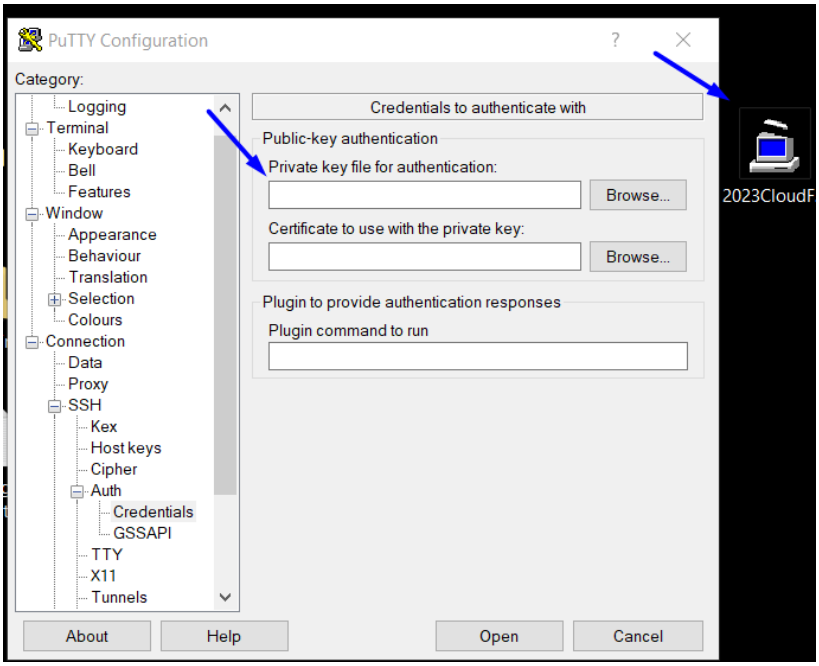
Our public IPv4DNS is shown on below.



We filled the "Host Name" section



Then we added the Auth credential from a file.



9. Install the Apache2 service to see the web page on the browser

Then we wrote following commands for this step:

```
sudo apt-get update
```

```
sudo apt-get install apache2
```

10. Build backend and router codes with the Python programming language

11. Import word count algorithm

12. Import HTML and CSS files

13. Write backend code with MVC (Model View Controller) architecture. (skip the model part because the database does not exist)

```python
import flask
import string


app = flask.Flask(__name__,template_folder='template',static_folder='static')

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/upload', methods = ['POST','GET'])
def upload():
    if request.method == 'POST':
        f = request.files['file']
        f.save('usertxtfile.txt')

        newdata = {}
        i = 0

        with open('usertxtfile.txt', 'r') as file:
            data = file.read()
            words = data.split()
            word_counts = {}

            for word in words:
                # make lowercase
                word = word.lower()

                # remove punctuations
                word = word.translate(str.maketrans('', '', string.punctuation))

                # remove digits
                word = word.translate(str.maketrans('', '', string.digits))

                if (len(word) == 0):
                    pass
                else:
                    if word in word_counts:
                        word_counts[word] += 1
                    else:
                        word_counts[word] = 1

            sorted_word_counts = sorted(word_counts.items(), key=lambda x: x[1], reverse=True)

            for word, count in sorted_word_counts[:10]:
                newdata[i] = [word, count]
                i += 1

        return render_template('final.html',data=newdata)
    else:
        return redirect(url_for('index'))


if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000, debug=True)
```

The code imports flask and string for the flask framework and normalizes operations.

```
Flask(__name__,template_folder='template',static_folder='static')
```

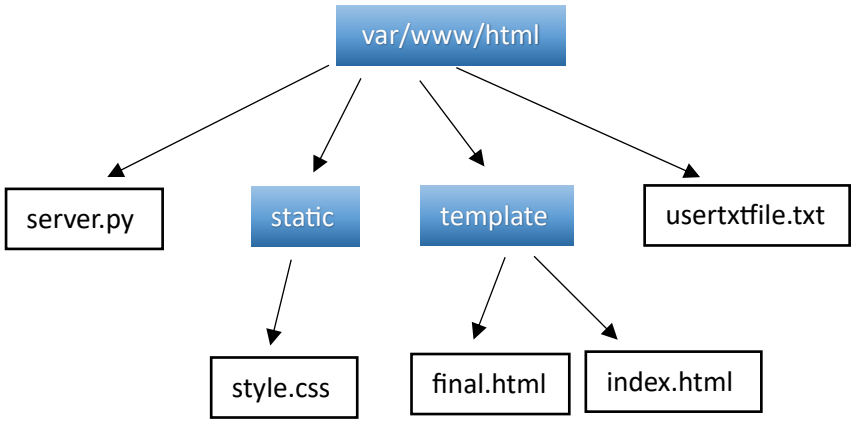
Function defines HTML and CSS files path. In our architecture, the" template" folder defines HTML files which are "index.html" (home page) and "final.html" (results page). The "static" folder defines for CSS files which are "style.css". Other than these two view folders; there are server.py which is our backend file and usertxtfile.txt which is an auxiliary file for recording and reading operations for word count.

Encountered Error:

At first, we tries to import our HTML and CSS files to a new anathor folder named "assets". But we figure out that this would not work.

Solution: We imported our files to the already defined "static" and "template" folders, and after introducing the path of these folders to Flask, we were able to call them.

Tree of working directory




```
@app.route('/')
def index():
    return render_template('index.html')
```

The index controller triggers when pure domain ("/") calls, it returns the "index.html" from the templates folder. So If a user requests the domain, the backend responses to the form page.

```
@app.route('/upload', methods = ['POST','GET'])
def upload():
```

The upload controller triggers when the continuation of the domain is "/upload". The controller accepts only two methods which are POST and GET.

```
if request.method == 'POST':
        // ...
    else:
        return redirect(url_for('index'))
```

In the upload controller, there is a validation for the request method. This validation is to prevent the program from exceptions. Because the user must not enter this controller with the GET method. This controller process posted the user txt file and counts it.

In Post Method:

```
f = request.files['file']
f.save('usertxtfile.txt')

newdata = {}
i = 0
```

Reads a posted file and assigns it to the "f" variable. After that saves the file to a local new file for the counting process.

Encountered Error:

The txt file can't be used from the HTTP Post method's body.

Solution:

So we saved the posted file to a new local txt file. And after saving we could read the text line by line.

```
with open('usertxtfile.txt', 'r') as file:
    data = file.read()
    words = data.split()
    word_counts = {}

    for word in words:
```

The program reads text from the local txt file line by line, split the words, and assign them to an words array. After that loops words and processes word for word.

```
# make lowercase
word = word.lower()

# remove punctuations
word = word.translate(str.maketrans('', '',
string.punctuation))

# remove digits
word = word.translate(str.maketrans('', '', string.digits))
```

Normalized the words. Normalize operation uses string library for punctuations and digits and converting to lowercase, removing punctuation, and removing digits.

Encountered Error:

Just at this stage, we had a problem which is some of the word groups like "the 3 car" normalized to "the car". This normalized form is ambiguous because there are two spaces. And this caused the counter to count empty words like: "".

Solution: We fixed this problem with this part:

```
if (len(word) == 0):
    pass
else:
    if word in word_counts:
        word_counts[word] += 1
    else:
        word_counts[word] = 1
```

The code filters the word if it has no length and then forwards it.

```
sorted_word_counts = sorted(word_counts.items(), key=lambda
x: x[1], reverse=True)
```

After normalization algorithm sorts the words with their counts in descending order.

```
for word, count in sorted_word_counts[:10]:
     newdata[i] = [word, count]
     i += 1
```

After the sorting process, filters the top 10 counted words and create a 2-dimensional array named newdata. This created array ("newdata") will respond to "final.html" file and prints the results.

Encountered Error:

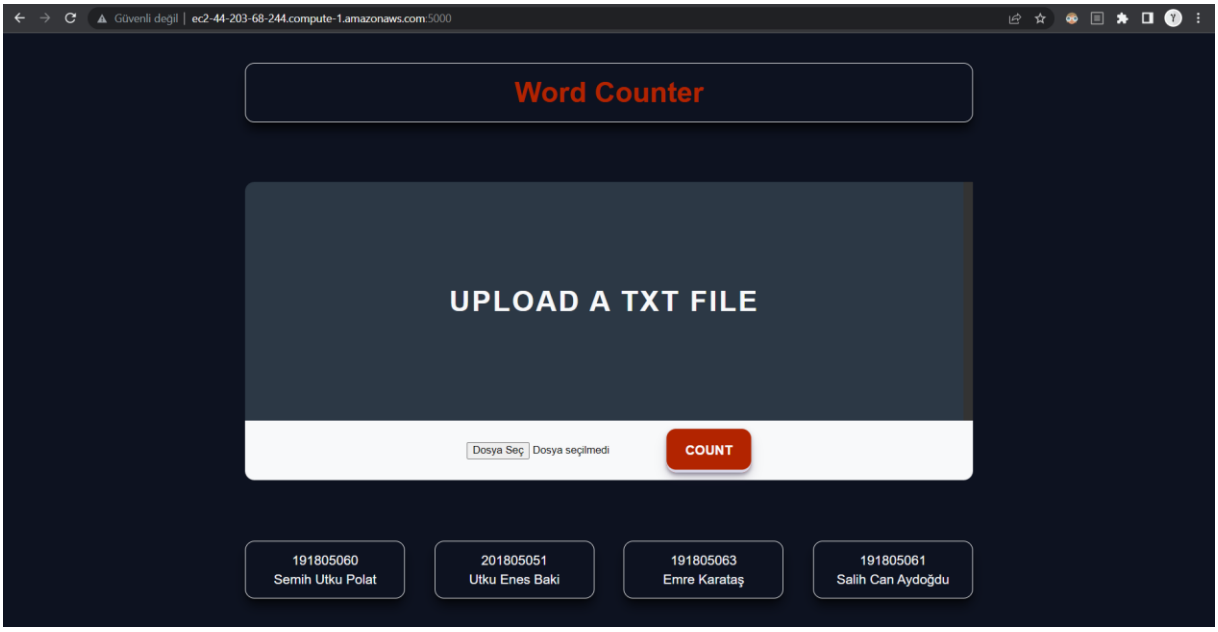
The first name of the "newdata" array was "data". But this causes confliction for the porgram beacuse of render_template() function has a "data" parameter too.

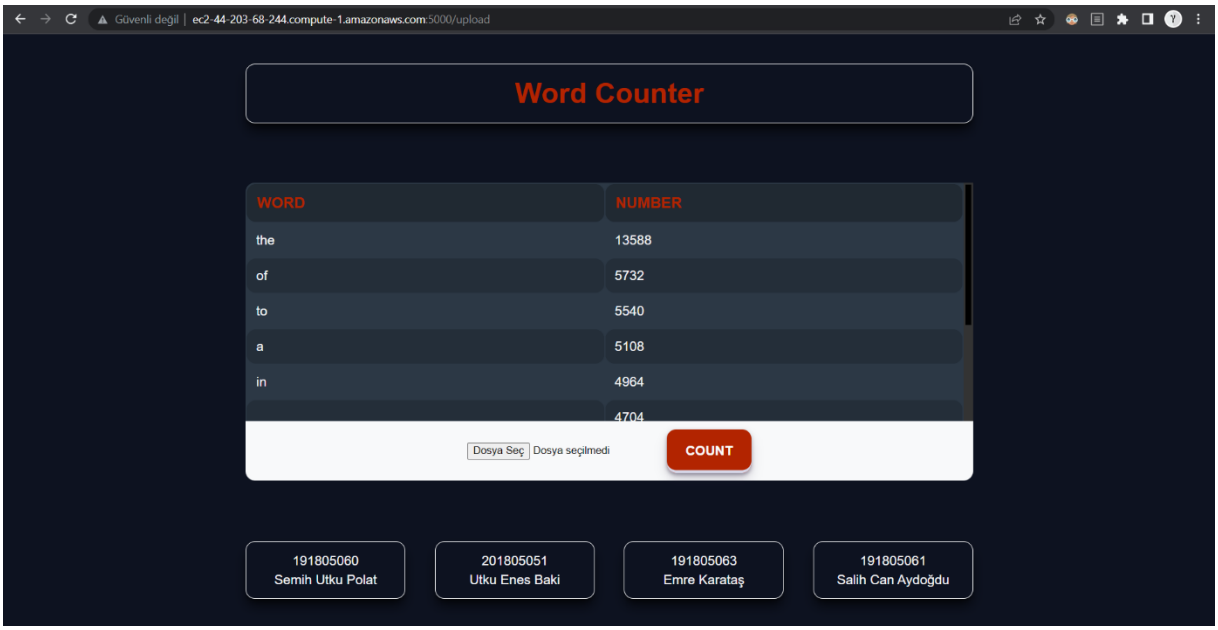
Solution: Even though it was difficult, we realized this and changed the name of the array to "newdata", and it solved the problem. As a result, it was necessary to keep new variable names away from common names.

```
return render_template('final.html',data=newdata)
```

Returns "final.html" view file with passing newdata array as data.

14. Test and debug the project





We've started the app with sudo python3 server.py. We've tested mobile and desktop views to test responsiveness. When we visit the /upload URL with the get method the backend redirects to the main URL (etc. index.html), and we've routed to the main page. In the backend upload function, the method validation is established. After a TXT file was posted, it was attempted to post the TXT file again from the result screen, and success was achieved.

Encountered Error:

When the backend code try to save posted file to a new local txt file, we encountered a "permission denied Error:13" exception.

Solution: The error was due to a lack of permission. So we started our server with "sudo python3 server.py" instead of "python3 server.py". 'sudo' grants privileges.

Working Policy

Name	Duty	Percent
Salih_Can_Aydoğdu_191805061	Front-End (index.html)	50%
Utku_Enes_Baki_201805051	Front-End (index.html)	50%
Salih_Can_Aydoğdu_191805061	Front-End (final.html)	50%
Utku_Enes_Baki_201805051	Front-End (final.html)	50%
Emre_Karataş_191805063	Front-End (style.css)	50%
Semih_Utku_Polat_191805060	Front-End (style.css)	50%
Utku_Enes_Baki_201805051	Connection set-up to the remote service	25%
Salih_Can_Aydoğdu_191805061	Connection set-up to the remote service	25%
Semih_Utku_Polat_191805060	Connection set-up to the remote service	25%
Emre_Karataş_191805063	Connection set-up to the remote service	25%
Semih_Utku_Polat_191805060	Back-End (word count)	50%
Utku_Enes_Baki_201805051	Back-End (word count)	50%
Salih_Can_Aydoğdu_191805061	Testing and Optimization	50%
Emre_Karataş_191805063	Testing and Optimization	50%
Semih_Utku_Polat_191805060	Reporting	50%
Emre_Karataş_191805063	Reporting	50%

References:

- ***AWS documantation for ec2:*** ***https://docs.aws.amazon.com/ec2/?icmpid=docs_homepage_featuredsvcs***
- ***Flask:*** ***https://flask.palletsprojects.com/en/2.3.x/***
- ***Render HTML:*** ***https://codeforgeek.com/render-html-file-in-flask/***
- ***VIM Commands:*** ***https://opensource.com/article/19/3/getting-started-vim***
- ***CSS:*** ***https://www.w3schools.com/css/default.asp***
- ***CSS:*** ***https://getcssscan.com/css-box-shadow-examples***
- ***CSS:*** ***https://developer.mozilla.org/en-US/docs/Web/CSS***

Our Project Codes (without snipping-all at once):

final.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>2023 Cloud Final</title>
    <link rel="stylesheet" href="./style.css">
</head>

<body>
    <main>
        <div class="container">
            <h1>Word Counter</h1>
            <section class="table-wrapper">
                <span>Top ten unique words sorted by value in descending :</span>
                <table>
                    <thead>
                        <tr>
                            <th>WORD</th>
                            <th>NUMBER</th>
                        </tr>
                    </thead>
                    <tbody>
                        <tr>
                            <td>{{data[0][0]}}</td>
                            <td>{{data[0][1]}}</td>
                        </tr>
                        <tr>
                            <td>{{data[1][0]}}</td>
                            <td>{{data[1][1]}}</td>
                        </tr>
                        <tr>
                            <td>{{data[2][0]}}</td>
                            <td>{{data[2][1]}}</td>
                        </tr>
                        <tr>
                            <td>{{data[3][0]}}</td>
                            <td>{{data[3][1]}}</td>
                        </tr>
                        <tr>
                            <td>{{data[4][0]}}</td>
                            <td>{{data[4][1]}}</td>
                        </tr>
                        <tr>
                            <td>{{data[5][0]}}</td>
```

```
                        <td>{{data[5][1]}}</td>
                    </tr>
                    <tr>
                        <td>{{data[6][0]}}</td>
                        <td>{{data[6][1]}}</td>
                    </tr>
                    <tr>
                        <td>{{data[7][0]}}</td>
                        <td>{{data[7][1]}}</td>
                    </tr>
                    <tr>
                        <td>{{data[8][0]}}</td>
                        <td>{{data[8][1]}}</td>
                    </tr>
                    <tr>
                        <td>{{data[9][0]}}</td>
                        <td>{{data[9][1]}}</td>
                    </tr>
                </tbody>
            </table>
        </section>
        <section class="upload-file">
            <form action="/upload" method="POST" enctype="multipart/form-data">
                <input type="file" name="file">
                <button type="submit">COUNT</button>
            </form>
        </section>
        <section class="team">
            <div class="col">
                <div>191805060</div>
                <div>Semih Utku Polat</div>
            </div>
            <div class="col">
                <div>201805051</div>
                <div>Utku Enes Baki</div>
            </div>
            <div class="col">
                <div>191805063</div>
                <div>Emre Karataş</div>
            </div>
            <div class="col">
                <div>191805061</div>
                <div>Salih Can Aydoğdu</div>
            </div>
        </section>
    </div>
</main>

</body>

</html>
```

index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>2023 Cloud Final</title>
    <link rel="stylesheet" href="./style.css">
</head>

<body>
    <main>
        <div class="container">
            <h1>Word Counter</h1>
            <section class="table-wrapper">
                <div class="title-2">UPLOAD A TXT FILE</div>
            </section>
            <section class="upload-file">
                <form action="/upload" method="POST" enctype="multipart/form-data">
                    <input type="file" name="file">
                    <button type="submit">COUNT</button>
                </form>
            </section>
            <section class="team">
                <div class="col">
                    <div>191805060</div>
                    <div>Semih Utku Polat</div>
                </div>
                <div class="col">
                    <div>201805051</div>
                    <div>Utku Enes Baki</div>
                </div>
                <div class="col">
                    <div>191805063</div>
                    <div>Emre Karataş</div>
                </div>
                <div class="col">
                    <div>191805061</div>
                    <div>Salih Can Aydoğdu</div>
                </div>
            </section>
        </div>
    </main>

</body>

</html>
```

server.py

```
import flask
import string


app = flask.Flask(__name__,template_folder='template',static_folder='static')

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/upload', methods = ['POST','GET'])
def upload():
    if request.method == 'POST':
        f = request.files['file']
        f.save('usertxtfile.txt')

        newdata = {}
        i = 0

        with open('usertxtfile.txt', 'r') as file:
            data = file.read()
            words = data.split()
            word_counts = {}

            for word in words:
                # make lowercase
                word = word.lower()

                # remove punctuations
                word = word.translate(str.maketrans('', '', string.punctuation))

                # remove digits
                word = word.translate(str.maketrans('', '', string.digits))

                if (len(word) == 0):
                    pass
                else:
                    if word in word_counts:
                        word_counts[word] += 1
                    else:
                        word_counts[word] = 1

            sorted_word_counts = sorted(word_counts.items(), key=lambda x: x[1], reverse=True)

            for word, count in sorted_word_counts[:10]:
                newdata[i] = [word, count]
                i += 1

        return render_template('final.html',data=newdata)
```

```
    else:
        return redirect(url_for('index'))


if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000, debug=True)
```

style.css

```
* {
    margin: 0px;
    padding: 0px;
    -webkit-box-sizing: border-box;
    -moz-box-sizing: border-box;
    box-sizing: border-box;
}

::-webkit-scrollbar {
    background-color: #333333;
    width: 12px;
}

::-webkit-scrollbar-thumb {
    background-color: #000000;
    border: 3px solid #333333;
}

html {
    font-size: 8px;
    scroll-behavior: smooth;
}

body {
    width: 100%;
    height: 100%;
    overflow-x: hidden;
    font-family: sans-serif;
    font-weight: 400;
    background-color: #0d1220;
}

main {
    width: 100%;
    height: 100%;
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
}
```

```css
.container {
    height: 100%;
    width: 100%;
    padding-top: 5vh;
    padding-bottom: 5vh;
    padding-right: 1rem;
    padding-left: 1rem;
}

h1 {
    min-height: 10vh;
    font-size: 3rem;
    color: #b22400;
    margin-bottom: 10vh;
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
    box-shadow: 0 10px 15px -3px rgba(0, 0, 0, .75), 0 4px 6px -4px rgba(0, 0, 0, .75);
    padding-right: 2rem;
    padding-left: 2rem;
    border-radius: 1rem;
    border: 1px solid #f8f9fa;
}


.title-2 {
    font-size: 3rem;
    font-weight: 700;
    letter-spacing: 0.2rem;
    color: #f8f9fa;
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
}

.table-wrapper {
    min-height: 40vh;
    max-height: 40vh;
    background-color: #2C3845;
    border-top-right-radius: 1rem;
    border-top-left-radius: 1rem;
    position: relative;
    overflow-y: scroll;
}

.table-wrapper span {
    position: absolute;
    bottom: calc(100% + 2vh);
    left: 0;
```

```
    color: #f8f9fa;
    font-size: 1.2rem;
}

table {
    width: 100%;
    border-top-right-radius: 1rem;
    border-top-left-radius: 1rem;
}

th {
    border-radius: 1rem;
    width: 50%;
    padding: 1rem;
    font-size: 1.6rem;
    color: #b22400;
    background-color: #202932;
    border: 1px solid #202932;
    font-weight: 700;
    text-align: left;
}

td {
    border-radius: 1rem;
    font-size: 1.4rem;
    color: #f8f9fa;
    padding: 1rem;
    text-align: left;
}

tr:nth-child(2n+2) td {
    background-color: #242e39;
}

.upload-file {
    min-height: 10vh;
    background-color: #f8f9fa;
    border-bottom-right-radius: 1rem;
    border-bottom-left-radius: 1rem;
}

.upload-file form {
    width: 100%;
    height: 10vh;
    display: flex;
    justify-content: center;
    align-items: center;
    padding-left: 1rem;
    padding-right: 1rem;
}
```

```
.upload-file form button {
    padding-top: 1rem;
    padding-bottom: 1rem;
    padding-left: 1.5rem;
    padding-right: 1.5rem;
    background-color: #b22400;
    color: #FCFCFD;
    font-size: 1.4rem;
    font-weight: 700;
    border: none;
    border-radius: 1rem;
    box-shadow: #2d234266 0 0.2rem 0.4rem, #2d23424d 0 0.7rem 1.3rem -0.3rem, #D6D6E7 0 -0.3rem 0
inset;
    cursor: pointer;
    -webkit-transition-property: all;
    -moz-transition-property: all;
    -o-transition-property: all;
    transition-property: all;
    -webkit-transition-duration: 0.2s;
    -moz-transition-duration: 0.2s;
    -o-transition-duration: 0.2s;
    transition-duration: 0.2s;
    -webkit-transition-timing-function: ease;
    -moz-transition-timing-function: ease;
    -o-transition-timing-function: ease;
    transition-timing-function: ease;
}

.upload-file form button:focus {
    box-shadow: #D6D6E7 0 0 0 0.1rem inset, #2d234266 0 0.2rem 0.4rem, #2d23424d 0 0.7rem 1.3rem -
0.3rem, #D6D6E7 0 -0.3rem 0 inset;
}

.upload-file form button:hover {
    box-shadow: #2d234266 0 0.4rem 0.8rem, #2d23424d 0 0.7rem 1.3rem -0.3rem, #D6D6E7 0 -0.3rem 0
inset;
    transform: translateY(-0.2rem);
}

.upload-file form button:active {
    box-shadow: #D6D6E7 0 0.3rem 0.7rem inset;
    transform: translateY(-0.2rem);
}

.team {
    min-height: 10vh;
    margin-top: 10vh;
}

.team .col {
    display: flex;
```

```
    flex-direction: column;
    justify-content: center;
    align-items: center;
    font-size: 1.4rem;
    line-height: 1.4;
    color: #f8f9fa;
    box-shadow: 0 10px 15px -3px rgba(0, 0, 0, .75), 0 4px 6px -4px rgba(0, 0, 0, .75);
    padding: 1rem;
    border-radius: 1rem;
    border: 1px solid #f8f9fa;
    margin-bottom: 2rem;
}

@media only screen and (min-width: 768px) {
    body {
        height: 100vh;
    }

    .team {
        display: flex;
        justify-content: space-between;
        align-items: center;
    }

    .team .col {
        width: 22%;
        margin-bottom: 0;
    }
}

@media only screen and (min-width: 992px) {
    .container {
        padding-right: 0;
        padding-left: 0;
        width: 60%;
    }

    .upload-file form button {
        padding-top: 1.5rem;
        padding-bottom: 1.5rem;
        padding-left: 2rem;
        padding-right: 2rem;
    }
}

@media only screen and (min-width: 1200px) {
    html {
        font-size: 12px;
    }
}
```

```css
@media only screen and (min-width: 1680px) {
    html {
        font-size: 13px;
    }
}

@media only screen and (min-width: 2050px) {
    html {
        font-size: 18px;
    }
}

@media only screen and (min-width: 2400px) {
    html {
        font-size: 20px;
    }
}
```