app.py

```
from flask import Flask, render_template, request, redirect
import os
import datetime
import csv
import webbrowser
fname = "error"
lname = "error"
varss=True
timeleft = datetime.datetime.now()
def openTab(x):
  if x ==True: #open a new tab
    webbrowser.open('http://127.0.0.1:5000/')
    print("tab open")

openTab(varss)
varss=False
app = Flask(__name__)
@app.route('/credits')
def default():
 return "hello World" #test page
@app.route('/back')
def back():
 global fname #imports the vars into the function
 global lname

 now = datetime.datetime.now().strftime("%H:%M:%S") #gets the time to allow the p
age to display the time left
 return render_template('back.html', variable=now, fnamew=fname, lnamew=lname) #s
ends the data to page
@app.route('/')
def home():
 global fname //it is in this section that I will change. Because I am setting
the names to a variable, the system will start overwriting variables when I new
person signs in. If I set the names and times to an array, I can condense it down
to one, which will make it easier to have multiple users.
 global lname
 global timeleft

 date_today = datetime.datetime.now().strftime("%Y-%m-%d")
 date = datetime.datetime.now().strftime("%H:%M:%S")
 now = datetime.datetime.now()
 timereturn = now-timeleft
 path = os.getcwd() #gets path to folder
```

```
 f = path+'//signoutsheets//'+date_today+'signoutlog'+'.csv'
 open(f, mode='a+')
 if fname != "error" or lname != "error":
  with open(f, mode='a+') as signout: #writes the lines to the spreadsheet
   signout_writer = csv.writer(signout, delimiter=',', quotechar='"', quoting=csv
.QUOTE_MINIMAL)

   signout_writer.writerow([fname.upper(), lname.upper(), date,str("returned"), s
tr("total time out: ") + str(timereturn)]) //In this section if I had the system
check the previous row, and write to it if the names match, I could condense the
spreadsheet by half. This would make it easier to read, and take  less time to
find the data you need.

 return render_template('home.html' )

@app.route('/', methods=['POST'])#gets the data from the page
def my_form_post():
 global fname
 global lname
 global timeleft
 fname = request.form['fname']
 lname = request.form['lname']
 lname_up = lname.upper()
 fname_up = fname.upper()
 date_today = datetime.datetime.now().strftime("%Y-%m-%d")
 date = datetime.datetime.now().strftime("%H:%M:%S")
 path = os.getcwd()
 f = path+'//signoutsheets//'+date_today+'signoutlog'+'.csv'
 timeleft = datetime.datetime.now()
 if fname != 'error' or lname != 'error':
  with open(f, mode='a+') as signout:
   signout_writer = csv.writer(signout, delimiter=',', quotechar='"', quoting=csv
.QUOTE_MINIMAL)
   signout_writer.writerow([fname_up,lname_up,date,str("left")])

 return redirect('http://127.0.0.1:5000/back') #sends to next screen


if __name__ == '__main__':
 ip = '127.0.0.1'
 port = int(os.environ.get('PORT', 5000))
 app.run(host=ip, port=port, debug=False)
```

back.html

```
<!DOCTYPE html>
<html>

<head>
  <script>
    var doneStuff;
    if (!doneStuff);
    doneStuff = true;
    var timeStart = new Date().getTime();
    var teachervar = false;
    var color = red;
  </script>
{% comment %} sets the variables {% endcomment %}


</head>
<script>
  function setvariables() {
    document.getElementById("alert").innerHTML =
      "STUDENT HAS BEEN OUT FOR MORE THAN 7 MINUTES. PLEASE ALERT YOUR TEACHER!";
    document.getElementById("disable").disabled = false;
  }
</script>

<body style="background-color:color;">
  </h1>
  <h2>{{ fnamew }} is currently out</h2>
  <p>Press the button below if you are returning. </p>
  <p2>The student left at: {{ variable }}</p2>
 {% comment %} this is what is displayed on the screen. {% endcomment %}
  <p id="demo"></p>
  <p id="alert"></p>
  <!-- https://www.w3schools.com/howto/howto_js_countdown.asp -->
  <script>
    // Set the date we're counting down to
    var countDownDate = new Date().getTime();
    // Update the count down every 1 second
    var x = setInterval(function () {
      // Get today's date and time
      var now = new Date().getTime();
      // Find the distance between now and the count down date
      var distance = now - timeStart;
```

```
      var colour = "white";
      var out = 0;
      document.body.style.backgroundColor = colour;
      // Time calculations for days, hours, minutes and seconds
      var days = Math.floor(distance / (1000 * 60 * 60 * 24));
      var hours = Math.floor((distance % (1000 * 60 * 60 * 24)) / (1000 * 60 * 60
));
      var minutes = Math.floor((distance % (1000 * 60 * 60)) / (1000 * 60));
      var seconds = Math.floor((distance % (1000 * 60)) / 1000);
      // Display the result in the element with id="demo"
      document.getElementById("demo").innerHTML = hours + "h " +
        minutes + "m " + seconds + "s ";
      // 600000=10mins
      if (distance > (600000/10)*7 && !teachervar) { <!-in this section the time
the student can be out is 7. However this value cannot be changed. An extension I
could make could be setting this value in a file, as this would allow my client
to change the time that can be spent out, without having to do a deep dive into
the code.
        setvariables()
        if (seconds % 2 == 0 && !teachervar)
        {
          var colour = "red"
          document.body.style.backgroundColor = colour;
        }
      }
    }, 1000);
  </script>
  <!-
- most of the code is written by an example. I mearly addaped it for my needs. Fr
om the start of this script
  till the end of the script -->


  <button onclick="goto()">I'm back</button>

  <script>
    function goto() {
      location.replace("http://127.0.0.1:5000/")
    }
  </script>
  <button id='disable' onclick="teacher()" disabled>Turn Off Alert</button>
  <script> //this section disables the alarm when pressed
    function teacher() {
      teachervar = true;
    }
```

```
  </script>
</body>
```

home.html

```
<!DOCTYPE html>
<!-- https://www.w3docs.com/learn-html/html-form-templates.html -->
<html>
  <head>
    <title>Bathroom Sign-out Sheet</title>
    <link href="https://fonts.googleapis.com/css?family=Roboto:300,400,500,700" r
el="stylesheet">
    <style>
      html, body {
      display: flex;
      justify-content: center;
      font-family: Roboto, Arial, sans-serif;
      font-size: 15px;
      }
      form {
      border: 5px solid #f1f1f1;
      }
      input[type=text], input[type=password] {
      width: 100%;
      padding: 16px 8px;
      margin: 8px 0;
      display: inline-block;
      border: 1px solid #ccc;
      box-sizing: border-box;
      }
      button {
      background-color: #8ebf42;
      color: white;
      padding: 14px 0;
      margin: 10px 0;
      border: none;
      cursor: grabbing;
      width: 100%;
      }
      h1 {
      text-align:center;
      font-size:18;
      }
```

```
      button:hover {
      opacity: 0.8;
      }
      .formcontainer {
      text-align: left;
      margin: 24px 50px 12px;
      }
      .container {
      padding: 16px 0;
      text-align:left;
      }
      span.psw {
      float: right;
      padding-top: 0;
      padding-right: 15px;
      }
      /* Change styles for span on extra small screens */
      @media screen and (max-width: 300px) {
      span.psw {
      display: block;
      float: none;
      }
    </style>
  </head>
  <body>
    <form method="POST">
      <h1>Bathroom Sign-Out Sheet</h1>
      <div class="formcontainer">
      <hr/>
      <div class="container">
        <label for="fname"><strong>First Name</strong></label>
        <input name = "fname" type="text" placeholder="Enter First Name"  require
d>
        <label for="lname"><strong>Last Name</strong></label>
        <input name = "lname" type="text" placeholder="Enter Last Name"  required
>
      </div>
      <button type="submit">Login</button>
      <div class="container" style="background-color: #eee">
     {% comment %} this is where the data is received {% endcomment %}

      </div>
    </form>
  </body>
</html>
```