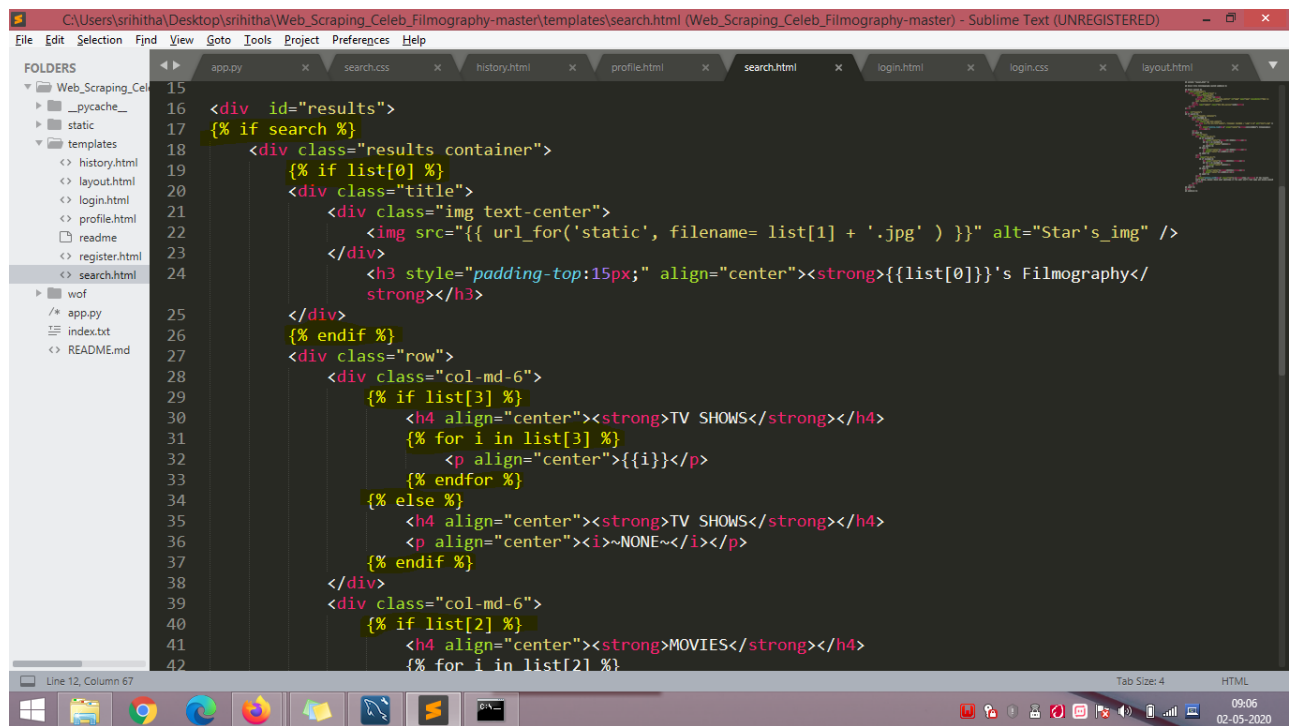


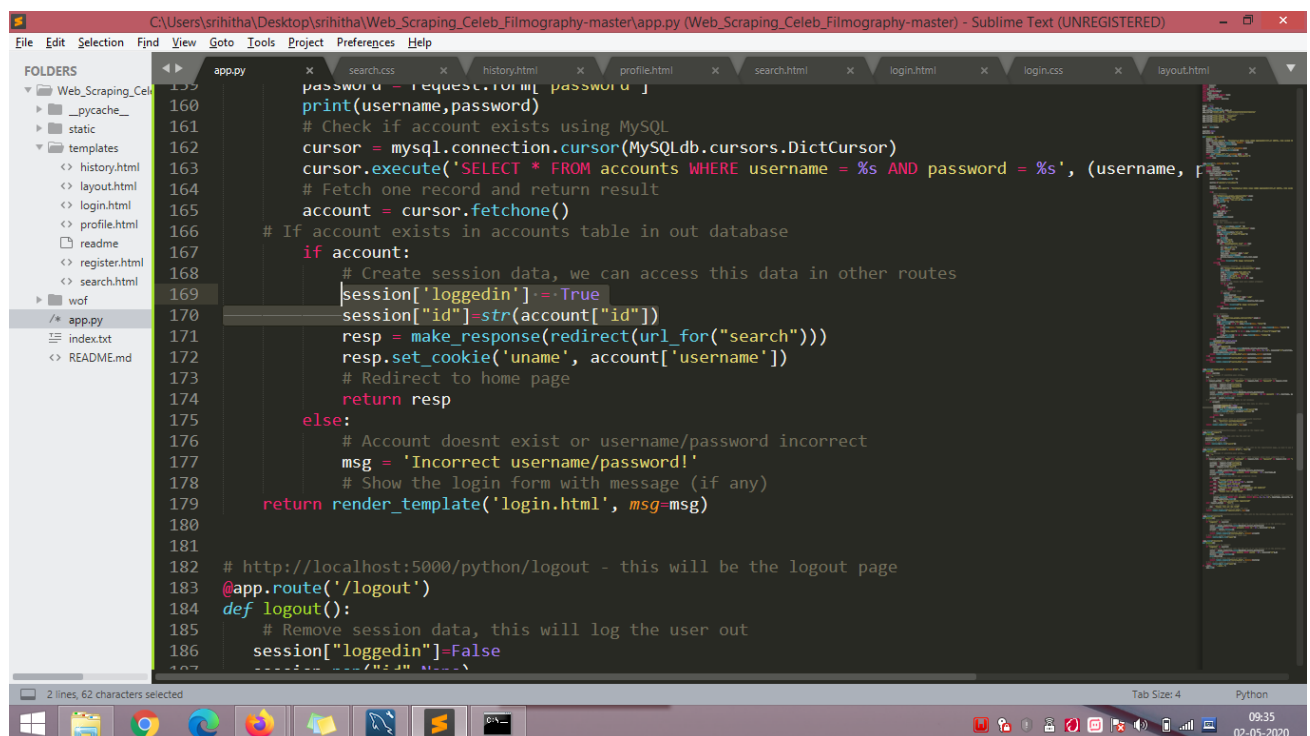
# Concepts Used

## 1. Python Basics



```
15 <div id="results">
16 {% if search %}
17 <div class="results container">
18 {% if list[0] %}
19 <div class="title">
20 <div class="img text-center">
21 
22 </div>
23 <h3 style="padding-top:15px;" align="center"><strong>{{list[0]}}'s Filmography</strong></h3>
24 </div>
25 {% endif %}
26 <div class="row">
27 <div class="col-md-6">
28 {% if list[3] %}
29 <h4 align="center"><strong>TV SHOWS</strong></h4>
30 {% for i in list[3] %}
31 <p align="center">{{i}}</p>
32 {% endfor %}
33 {% else %}
34 <h4 align="center"><strong>TV SHOWS</strong></h4>
35 <p align="center"><i>~NONE~</i></p>
36 {% endif %}
37 </div>
38 <div class="col-md-6">
39 {% if list[2] %}
40 <h4 align="center"><strong>MOVIES</strong></h4>
41 {% for i in list[2] %}
42
```

## 2. Session



```
155 password = request.form[ 'password' ]
160 print(username,password)
161 # Check if account exists using MySQL
162 cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
163 cursor.execute('SELECT * FROM accounts WHERE username = %s AND password = %s', (username, password))
164 # Fetch one record and return result
165 account = cursor.fetchone()
166 # If account exists in accounts table in our database
167 if account:
168     # Create session data, we can access this data in other routes
169     session['loggedin']=True
170     session['id']=str(account["id"])
171     resp = make_response(redirect(url_for("search")))
172     resp.set_cookie('uname', account['username'])
173     # Redirect to home page
174     return resp
175 else:
176     # Account doesnt exist or username/password incorrect
177     msg = 'Incorrect username/password!'
178     # Show the login form with message (if any)
179     return render_template('login.html', msg=msg)
180
181
182 # http://localhost:5000/python/logout - this will be the logout page
183 @app.route('/logout')
184 def logout():
185     # Remove session data, this will log the user out
186     session["loggedin"]=False
187
```

```
C:\Users\srihitha\Desktop\srihitha\Web_Scraping_Celeb_Filmography-master\app.py (Web_Scraping_Celeb_Filmography-master) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
Web_Scraping_Celeb_Filmography-master
  _pycache_
  static
  templates
    history.html
    layout.html
    login.html
    profile.html
    register.html
    search.html
  wof
  app.py
  index.txt
  README.md
app.py
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
if not l:
    l=c.find(class_="title").get_text() for c in soup.findAll(class_="junCMe")]
if not l:
    l=[c['aria-label'] for c in soup.findAll("a", {"class":"llgymd"})]
if not l:
    l=[x.get_text() for x in soup.findAll(class_="FLP8od")]
querylist.append(l)
except:
    querylist=[None,None,None]
print("QUERYLIST",querylist)
searched=True
if session["loggedin"]:
    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
    cursor.execute('INSERT INTO history VALUES (%s, NULL, %s, %s)', (session["id"],datetime.now(), querylist))
    mysql.connection.commit()
    return render_template('search.html',List=querylist,search=searched)
try:
    return render_template('search.html',List=querylist,search=searched)
except:
    return render_template('search.html',List=querylist,search=searched)
p.route("/login.html", methods=['GET', 'POST'])
login():
global searched
# Output message if something goes wrong...
msg = ""
```

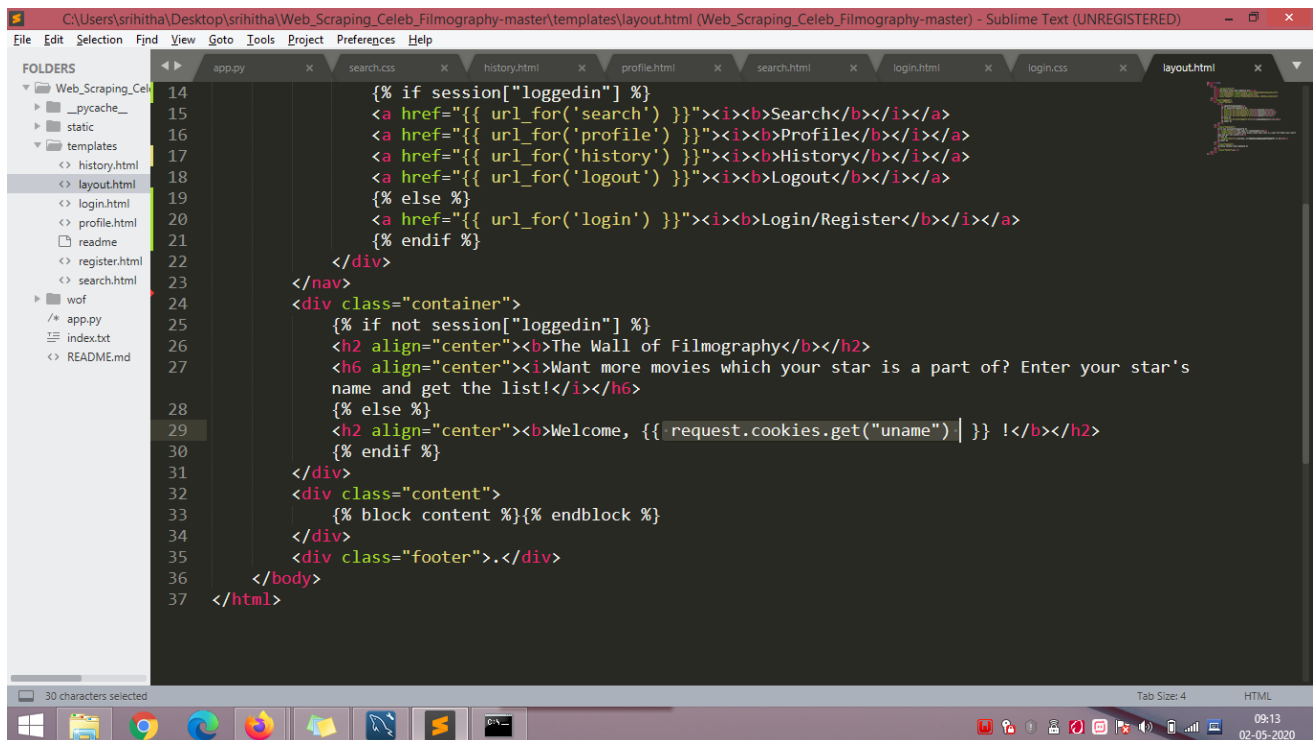
```
C:\Users\srihitha\Desktop\srihitha\Web_Scraping_Celeb_Filmography-master\app.py (Web_Scraping_Celeb_Filmography-master) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
Web_Scraping_Celeb_Filmography-master
  _pycache_
  static
  templates
    history.html
    layout.html
    login.html
    profile.html
    register.html
    search.html
  wof
  app.py
  index.txt
  README.md
app.py
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
# Redirect to home page
return resp
else:
    # Account doesnt exist or username/password incorrect
    msg = 'Incorrect username/password!'
    # Show the login form with message (if any)
    return render_template('login.html', msg=msg)
# http://localhost:5000/python/logout - this will be the logout page
@app.route('/logout')
def logout():
    # Remove session data, this will log the user out
    session["loggedin"]=False
    session.pop("id",None)
    # Redirect to login page
    return redirect(url_for('search'))
# http://localhost:5000/python/login/register - this will be the registration page, we need to use
@app.route('/register.html', methods=['GET', 'POST'])
def register():
    # Output message if something goes wrong...
    msg = ''
    # Check if "username", "password" and "email" POST requests exist (user submitted form)
    if request.method == 'POST' and 'username' in request.form and 'password' in request.form and 'email' in request.form:
        # Create variables for easy access
        username = request.form['username']
        password = request.form['password']
```

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>{% block title %}{% endblock %}</title>
    <link rel="stylesheet" media="screen" href="static/bootstrap.min.css">
    <link rel="stylesheet" href="static/search.css">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body class="loggedin">
    <nav class="navtop">
      <div>
        <h1>WallOfFilmography</h1>
        {% if session["loggedin"] %}
        <a href="{{ url_for('search') }}"><i><b>Search</b></i></a>
        <a href="{{ url_for('profile') }}"><i><b>Profile</b></i></a>
        <a href="{{ url_for('history') }}"><i><b>History</b></i></a>
        <a href="{{ url_for('logout') }}"><i><b>Logout</b></i></a>
        {% else %}
        <a href="{{ url_for('login') }}"><i><b>Login/Register</b></i></a>
        {% endif %}
      </div>
    </nav>
    <div class="container">
      {% if not session["loggedin"] %}
      <h2 align="center"><b>The Wall of Filmography</b></h2>
      <h6 align="center"><i>Want more movies which your star is a part of? Enter your star's
      name and get the list!</i></h6>
      {% else %}
```

### 3. Cookies

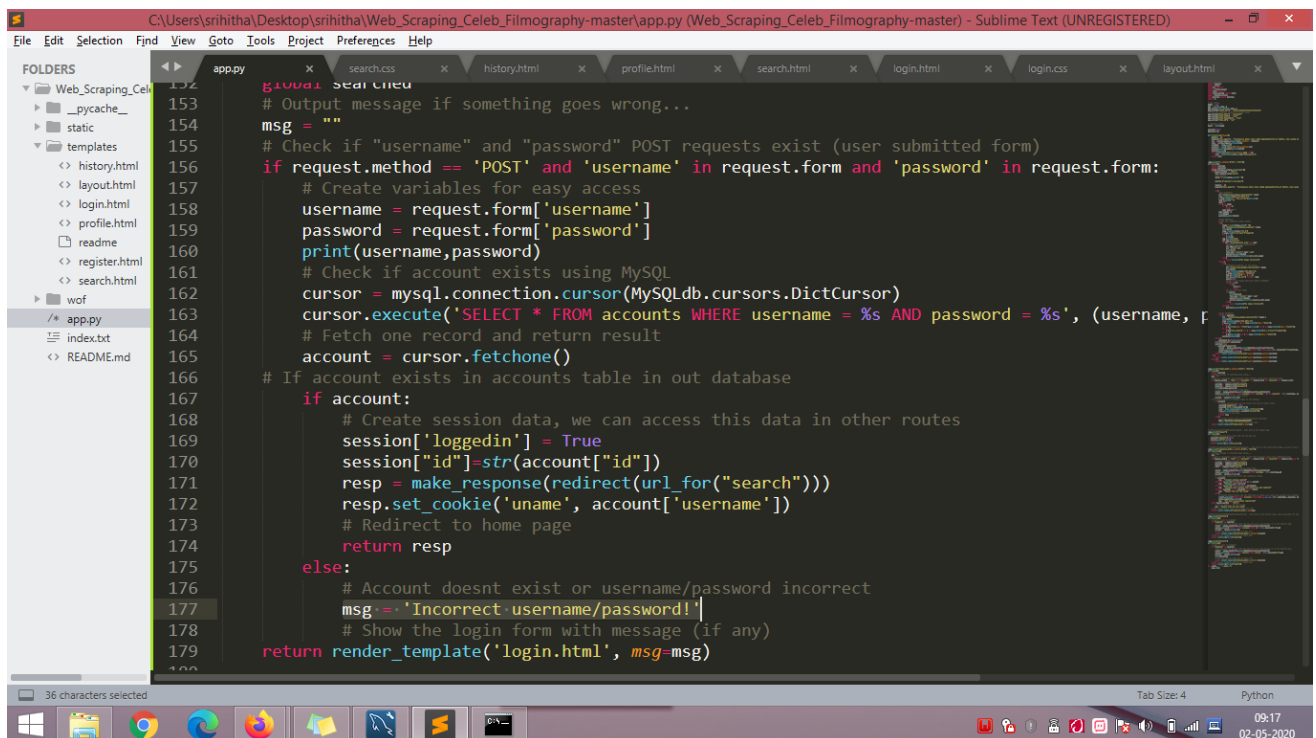
```
password = request.form['password']
print(username,password)
# Check if account exists using MySQL
cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
cursor.execute('SELECT * FROM accounts WHERE username = %s AND password = %s', (username, password))
# Fetch one record and return result
account = cursor.fetchone()
# If account exists in accounts table in our database
if account:
    # Create session data, we can access this data in other routes
    session['loggedin'] = True
    session['id'] = str(account['id'])
    resp = make_response(redirect(url_for("search")))
    resp.set_cookie('uname', account['username'])
    # Redirect to home page
    return resp
else:
    # Account doesn't exist or username/password incorrect
    msg = 'Incorrect username/password!'
    # Show the login form with message (if any)
    return render_template('login.html', msg=msg)

# http://localhost:5000/python/logout - this will be the logout page
@app.route('/logout')
def logout():
    # Remove session data, this will log the user out
    session["loggedin"] = False
```



```
14 {% if session["loggedin"] %}
15 <a href="{{ url_for('search') }}"><i><b>Search</b></i></a>
16 <a href="{{ url_for('profile') }}"><i><b>Profile</b></i></a>
17 <a href="{{ url_for('history') }}"><i><b>History</b></i></a>
18 <a href="{{ url_for('logout') }}"><i><b>Logout</b></i></a>
19 {% else %}
20 <a href="{{ url_for('login') }}"><i><b>Login/Register</b></i></a>
21 {% endif %}
22 </div>
23 </nav>
24 <div class="container">
25     {% if not session["loggedin"] %}
26     <h2 align="center"><b>The Wall of Filmography</b></h2>
27     <h6 align="center"><i>Want more movies which your star is a part of? Enter your star's
28     name and get the list!</i></h6>
29     {% else %}
30     <h2 align="center"><b>Welcome, {{ request.cookies.get("uname") }} !</b></h2>
31     {% endif %}
32 </div>
33 <div class="content">
34     {% block content %}{% endblock %}
35 </div>
36 <div class="footer"></div>
37 </body>
38 </html>
```

#### 4. Error



```
153 # Output message if something goes wrong...
154 msg = ""
155 # Check if "username" and "password" POST requests exist (user submitted form)
156 if request.method == 'POST' and 'username' in request.form and 'password' in request.form:
157     # Create variables for easy access
158     username = request.form['username']
159     password = request.form['password']
160     print(username,password)
161     # Check if account exists using MySQL
162     cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
163     cursor.execute('SELECT * FROM accounts WHERE username = %s AND password = %s', (username, password))
164     # Fetch one record and return result
165     account = cursor.fetchone()
166     # If account exists in accounts table in our database
167     if account:
168         # Create session data, we can access this data in other routes
169         session['loggedin'] = True
170         session['id']=str(account["id"])
171         resp = make_response(redirect(url_for("search")))
172         resp.set_cookie('uname', account['username'])
173         # Redirect to home page
174         return resp
175     else:
176         # Account doesn't exist or username/password incorrect
177         msg = 'Incorrect username/password!'
178         # Show the login form with message (if any)
179     return render_template("login.html", msg=msg)
```

```
File Edit Selection Find View Goto Tools Project Preferences Help
C:\Users\srihitha\Desktop\srihitha\Web_Scraping_Celeb_Filmography-master\templates\login.html (Web_Scraping_Celeb_Filmography-master) - Sublime Text (UNREGISTERED)

FOLDERS
Web_Scraping_Celeb_Filmography-master
├── _pycache_
├── static
├── templates
│   ├── history.html
│   ├── layout.html
│   ├── login.html
│   ├── profile.html
│   ├── readme
│   ├── register.html
│   └── search.html
└── wof
    ├── app.py
    ├── index.txt
    └── README.md

7 <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.1/css/all.css">
8 </head>
9 <body>
10 <div class="login">
11 <h1>Login</h1>
12 <div class="links">
13 <a href="{{ url_for('login') }}" class="active">Login</a>
14 <a href="{{ url_for('register') }}">Register</a>
15 </div>
16 <form action="{{ url_for('login') }}" method="post">
17 <label for="username">
18 <i class="fas fa-user"></i>
19 </label>
20 <input type="text" name="username" placeholder="Username" id="username" required>
21 <label for="password">
22 <i class="fas fa-lock"></i>
23 </label>
24 <input type="password" name="password" placeholder="Password" id="password"
25 required>
26 <div class="msg">{{ msg }}</div>
27 <input type="submit" value="Login">
28 </form>
29 </div>
30 </body>
</html>
```

