

MEDIA STREAMING WITH IBM CLOUD STREAMING

PHASE – 4

DEVELOPMENT PHASE – II

THESE CODES ARE CONTINUATION OF DEVELOPMENT PHASE -1

STEPS TO BE IMPLEMENTED

1. CODE TO RENDER HTML IN FLASK:

```
from flask import Flask,render_template, request, redirect, url_for,  
send_from_directory
```

```
import os
```

```
app = Flask(__name__)
```

```
app.config['UPLOAD_FOLDER'] = 'upload'
```

```
app.static_folder = 'static'
```

```
@app.route('/')
```

```
def hello_world(): # put application's code here
```

```
    return render_template('login.html')
```

```
@app.route('/register')
```

```
def about():
```

```
    return render_template('register.html')
```

```
@app.route('/video')
def upload_form():
    return render_template('upload.html')

@app.route('/upload', methods=['POST'])
def upload_video():
    if 'video' not in request.files:
        return redirect(request.url)

    video = request.files['video']

    if video.filename == '':
        return redirect(request.url)

    if video:
        video.save(os.path.join(app.config['UPLOAD_FOLDER'],
video.filename))

        return redirect(url_for('view_video', filename=video.filename))

@app.route('/upload/<filename>')
def view_video(filename):
    return render_template('view.html', filename=filename)

#@app.route('/upload/<filename>')
#def send_video(filename):
```

```
# return send_from_directory(app.config['UPLOAD_FOLDER'], filename)*/
```

```
@app.route('/upload/<filename>')
```

```
def send_video(filename):
```

```
    return send_from_directory(app.config['UPLOAD_FOLDER'], filename,  
                               mimetype='video/mp4')
```

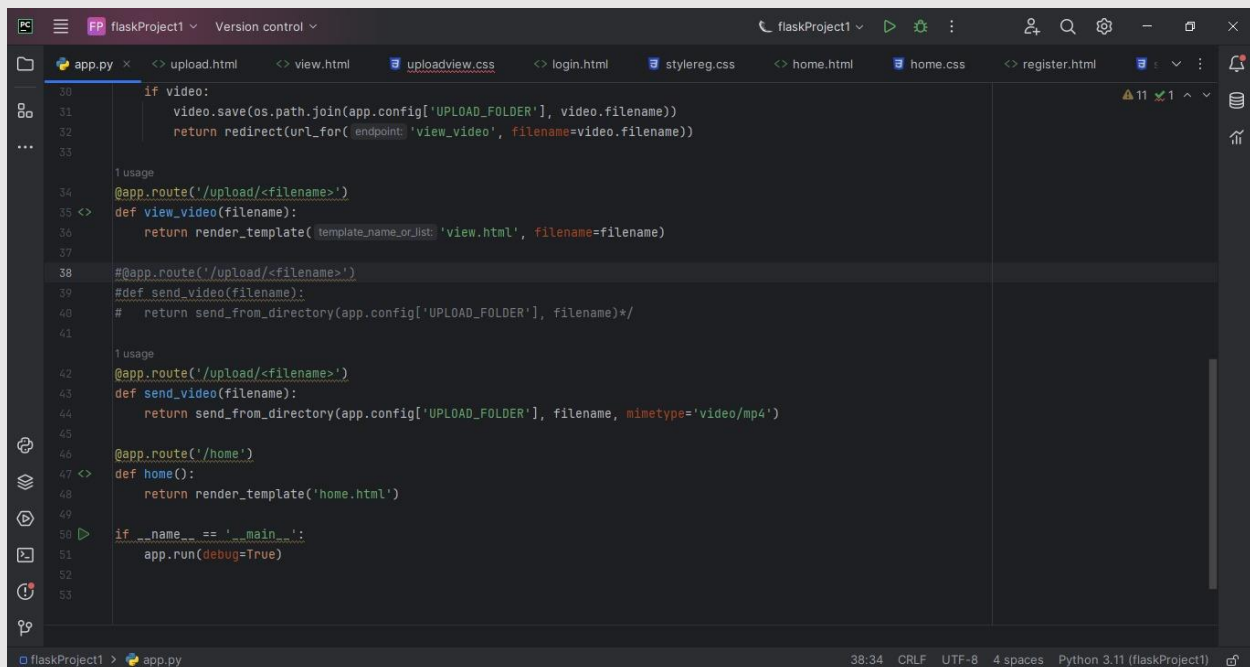
```
@app.route('/home')
```

```
def home():
```

```
    return render_template('home.html')
```

```
if __name__ == '__main__':
```

```
    app.run(debug=True)
```

A screenshot of a code editor window titled 'flaskProject1'. The editor shows a Python file 'app.py' with the following code:

```
30 if video:
31     video.save(os.path.join(app.config['UPLOAD_FOLDER'], video.filename))
32     return redirect(url_for('view_video', filename=video.filename))
33
34 1 usage
35 @app.route('/upload/<filename>')
36 def view_video(filename):
37     return render_template(template_name_or_list='view.html', filename=filename)
38
39 #@app.route('/upload/<filename>')
40 #def send_video(filename):
41 #    return send_from_directory(app.config['UPLOAD_FOLDER'], filename)*/
42
43 1 usage
44 @app.route('/upload/<filename>')
45 def send_video(filename):
46     return send_from_directory(app.config['UPLOAD_FOLDER'], filename, mimetype='video/mp4')
47
48 @app.route('/home')
49 def home():
50     return render_template('home.html')
51
52 if __name__ == '__main__':
53     app.run(debug=True)
```

 The editor interface includes a sidebar with file explorer, a top toolbar with various icons, and a bottom status bar showing '38:34 CRLF UTF-8 4 spaces Python 3.11 (flaskProject1)'.

2. HTML CODE TO CREATE A HOMEPAGE:

```
<!DOCTYPE html>

<html lang="en" dir="ltr">

  <head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-
scale=1.0">

    <title> Website Layout | CodingLab</title>

    <!link rel="stylesheet" href="css/home.css">

    <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='css/home.css') }}">

  </head>

<body>

  <nav>

    <div class="menu">

      <div class="logo">

        <a href="#">IBM Media Streaming</a>

      </div>

      <ul>

        <li><a href="#">Home</a></li>

        <li><a href="#">About</a></li>

        <li><a href="/video">upload</a></li>

        <li><a href="#">profile</a></li>
```

```
<li><a href="#">Feedback</a></li>

</ul>

</div>

</nav>

<div class="img"></div>

<div class="center">

  <div class="title">WELCOME TO IBM MEDIA STREAMING</div>

  <div class="sub_title">Watch ,Hear ,Enjoy...</div>

  <div class="btns">

    <button>watch videos</button>

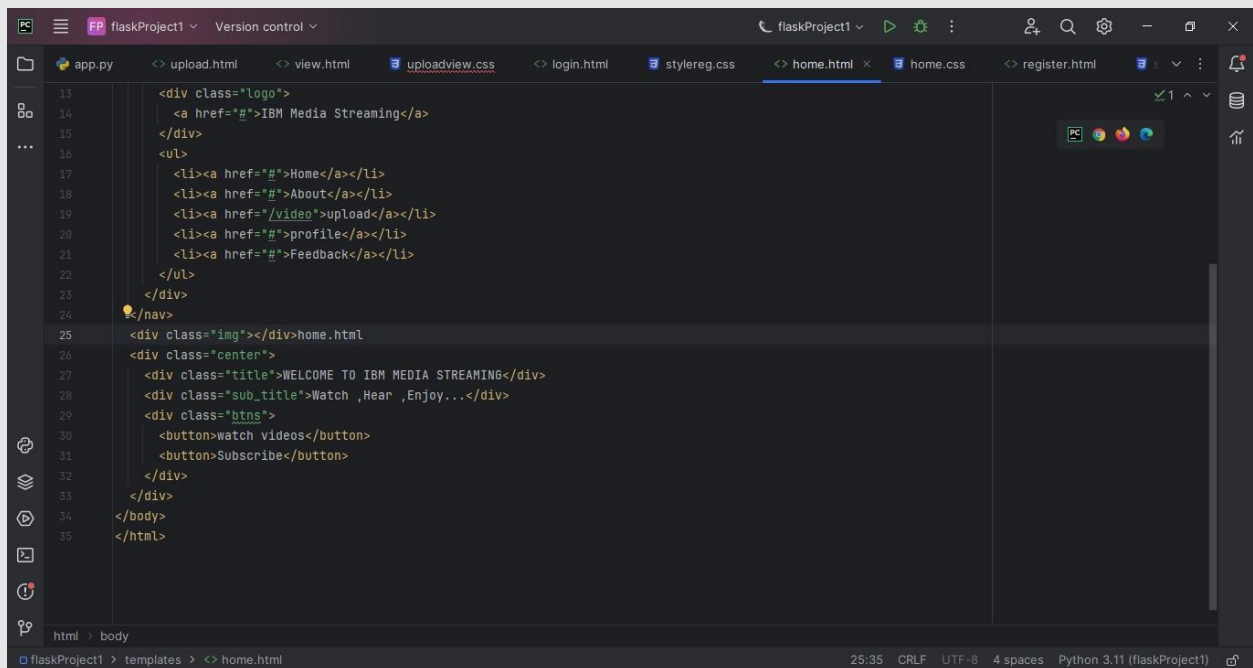
    <button>Subscribe</button>

  </div>

</div>

</body>

</html>
```



The screenshot shows a code editor with a dark theme. The file explorer on the left shows a project named 'flaskProject1' with a 'templates' folder. The editor is open to 'home.html'. The code in the editor matches the HTML code provided in the previous block. The status bar at the bottom indicates the file is 'home.html' in the 'templates' folder, with a timestamp of 25:35, encoding of CRLF, and a Python 3.11 interpreter.

```
13 <div class="logo">
14   <a href="#">IBM Media Streaming</a>
15 </div>
16 <ul>
17   <li><a href="#">Home</a></li>
18   <li><a href="#">About</a></li>
19   <li><a href="/video">upload</a></li>
20   <li><a href="#">profile</a></li>
21   <li><a href="#">Feedback</a></li>
22 </ul>
23 </div>
24 </nav>
25 <div class="img"></div>home.html
26 <div class="center">
27   <div class="title">WELCOME TO IBM MEDIA STREAMING</div>
28   <div class="sub_title">Watch ,Hear ,Enjoy...</div>
29   <div class="btns">
30     <button>watch videos</button>
31     <button>Subscribe</button>
32   </div>
33 </div>
34 </body>
35 </html>
```

3. CSS CODE TO CREATE A HOMEPAGE:

```
@import
url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;
500;600;700&display=swap');
*{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: 'Poppins',sans-serif;
}
::selection{
  color: #000;
  background: #fff;
}
nav{
  position: fixed;
  background: #1b1b1b;
  width: 100%;
  padding: 10px 0;
  z-index: 12;
}
nav .menu{
  max-width: 1250px;
  margin: auto;
  display: flex;
  align-items: center;
  justify-content: space-between;
  padding: 0 20px;
}
.menu .logo a{
  text-decoration: none;
  color: #fff;
  font-size: 35px;
  font-weight: 600;
```

```
}
.menu ul{
  display: inline-flex;
}
.menu ul li{
  list-style: none;
  margin-left: 7px;
}
.menu ul li:first-child{
  margin-left: 0px;
}
.menu ul li a{
  text-decoration: none;
  color: #fff;
  font-size: 18px;
  font-weight: 500;
  padding: 8px 15px;
  border-radius: 5px;
  transition: all 0.3s ease;
}
.menu ul li a:hover{
  background: #fff;
  color: black;
}
.img{
  /*background: url('static/images.jpg') no-repeat; bv
  width: 100%;
  height: 100vh;
  background-size: cover;
  background-position: center;
  position: relative;*/
  background-color: rgba(0, 255, 102, 0.82); /* Green background color */
  width: 100%;
  height: 100vh;
  position: relative;
```

```
}  
.img::before{  
  content: "";  
  position: absolute;  
  height: 100%;  
  width: 100%;  
  background: rgba(0, 0, 0, 0.4);  
}  
.center{  
  position: absolute;  
  top: 52%;  
  left: 50%;  
  transform: translate(-50%, -50%);  
  width: 100%;  
  padding: 0 20px;  
  text-align: center;  
}  
.center .title{  
  color: #fff;  
  font-size: 55px;  
  font-weight: 600;  
}  
.center .sub_title{  
  color: #fff;  
  font-size: 52px;  
  font-weight: 600;  
}  
.center .btns{  
  margin-top: 20px;  
}  
.center .btns button{  
  height: 55px;  
  width: 170px;  
  border-radius: 5px;  
  border: none;
```



```

margin: 0 10px;
border: 2px solid white;
font-size: 20px;
font-weight: 500;
padding: 0 10px;
cursor: pointer;
outline: none;
transition: all 0.3s ease;
}

.center .btns button:first-child{
  color: #fff;
  background: none;
}

.btns button:first-child:hover{
  background: white;
  color: black;
}

.center .btns button:last-child{
  background: white;
  color: black;
}

```

```

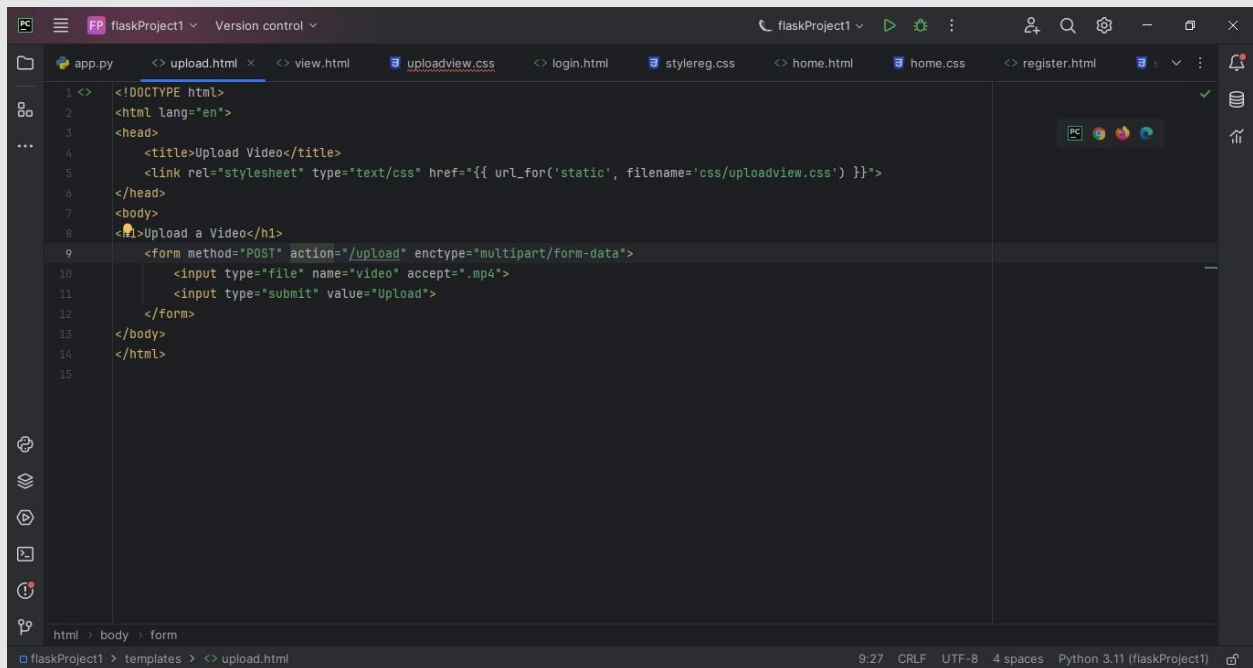
48 padding: 8px 15px;
49 border-radius: 5px;
50 transition: all 0.3s ease;
51 }
52 .menu ul li a:hover{
53   background: #fff;
54   color: black;
55 }
56 .img{
57   /*background: url('static/images.jpg') no-repeat; b
58   width: 100%;
59   height: 100vh;
60   background-size: cover;
61   background-position: center;
62   position: relative;*/
63   background-color: rgba(0, 255, 102, 0.82); /* Green background color */
64   width: 100%;
65   height: 100vh;
66   position: relative;
67 }
68 .img::before{
69   content: '';
70   position: absolute;
71   height: 100%;
72   width: 100%;
73   background: rgba(0, 0, 0, 0.4);
74 }

```

flaskProject1 > static > css > home.css 57:55 CRLF UTF-8 2 spaces* Python 3.11 (flaskProject1)

4. HTML CODE TO UPLOAD A DATA IN STREAMING SITE:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Upload Video</title>
  <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='css/uploadview.css') }}">
</head>
<body>
<h1>Upload a Video</h1>
  <form method="POST" action="/upload" enctype="multipart/form-
data">
    <input type="file" name="video" accept=".mp4">
    <input type="submit" value="Upload">
  </form>
</body>
</html>
```



5. HTML TO STREAM A DATA IN STREAMING SITE:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    .highlight-video {
      border: 2px solid rgba(0, 255, 183, 0.61); /* Add a red border around
the video */
      animation-name: none
    }
  </style>

  <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='css/styles.css') }}">
</head>
<body>
  <!-- Background Container -->
  <div class="background-container">
    <!-- Stars -->
    <div class="stars"></div>

    <!-- Twinkling Stars -->
    <div class="twinkling"></div>

    <!-- Clouds -->
    <div class="clouds"></div>

    <!-- Your Video Element -->

  </div>

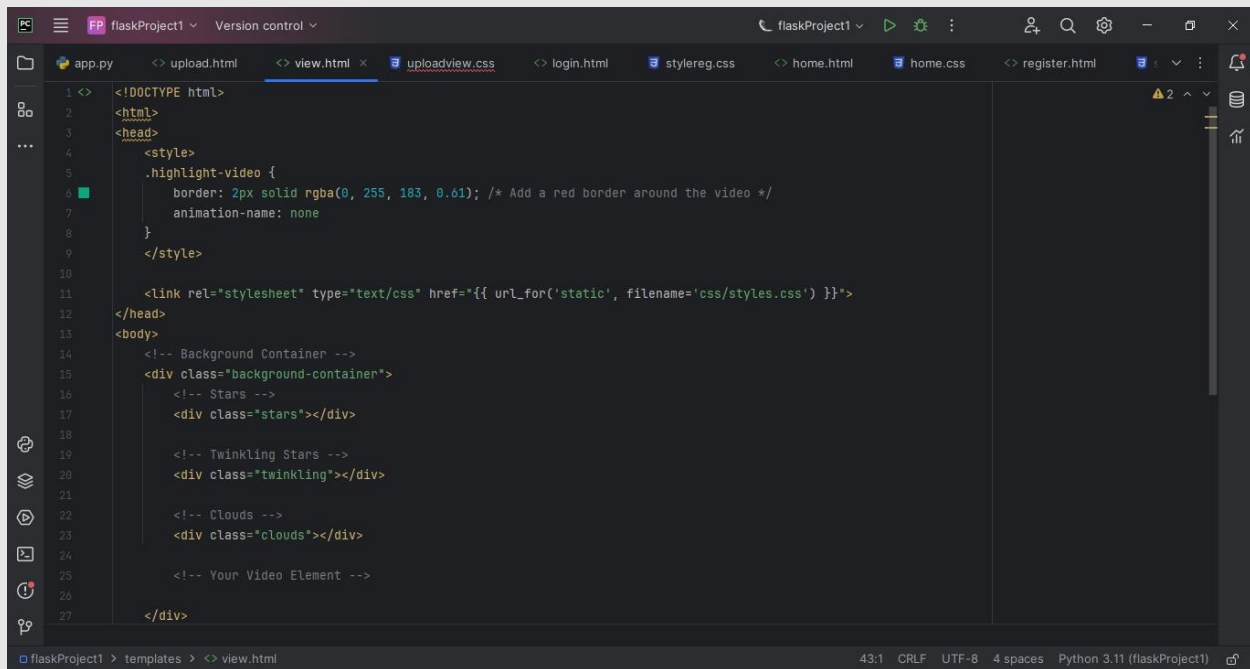
  <div class="video-container">
    <video class="highlight-video" width="640" height="480" controls >
```

```

        <source src="{{ url_for('send_video', filename=filename) }}"
type="video/mp4">
    </video>
</div>
<!-- Your regular page content goes here -->
<div class="page-content">
    <!-- More content, other sections, etc. -->
</div>

</body>
</html>

```



6. CSS CODE FOR BOTH UPLOAD AND STREAMING A VIDEO:

```

@keyframes move-background {
  from {
    -webkit-transform: translate3d(0px, 0px, 0px);
  }
  to {
    -webkit-transform: translate3d(1000px, 0px, 0px);
  }
}

```

```
}
```

```
@-webkit-keyframes move-background {  
  from {  
    -webkit-transform: translate3d(0px, 0px, 0px);  
  }  
  to {  
    -webkit-transform: translate3d(1000px, 0px, 0px);  
  }  
}
```

```
@-moz-keyframes move-background {  
  from {  
    -webkit-transform: translate3d(0px, 0px, 0px);  
  }  
  to {  
    -webkit-transform: translate3d(1000px, 0px, 0px);  
  }  
}
```

```
.background-container {  
  position: fixed;  
  top: 0;  
  left: 0;  
  bottom: 0;  
  right: 0;  
}
```

```
.stars {  
  background: black url(https://s3-us-west-2.amazonaws.com/s.cdpn.io/1231630/stars.png) repeat;  
  position: absolute;  
  top: 0;  
  bottom: 0;  
  left: 0;
```

```
right: 0;
display: block;
z-index: 0;
}
```

```
.twinkling {
width: 10000px;
height: 100%;
background: transparent url("https://s3-us-west-
2.amazonaws.com/s.cdpn.io/1231630/twinkling.png") repeat;
background-size: 1000px 1000px;
position: absolute;
right: 0;
top: 0;
bottom: 0;
z-index: 2;
}
```

```
-moz-animation: move-background 70s linear infinite;
-ms-animation: move-background 70s linear infinite;
-o-animation: move-background 70s linear infinite;
-webkit-animation: move-background 70s linear infinite;
animation: move-background 70s linear infinite;
}
```

```
.clouds {
width: 10000px;
height: 100%;
background: transparent url("https://s3-us-west-
2.amazonaws.com/s.cdpn.io/1231630/clouds_repeat.png") repeat;
background-size: 1000px 1000px;
position: absolute;
right: 0;
top: 0;
bottom: 0;
z-index: 3;
}
```

```

-moz-animation: move-background 150s linear infinite;
-ms-animation: move-background 150s linear infinite;
-o-animation: move-background 150s linear infinite;
-webkit-animation: move-background 150s linear infinite;
animation: move-background 150s linear infinite;
}

.video-container {
    position: relative;
}

/* CSS rule for highlighting the video */
.highlight-video {
    border: 2px solid red;
}

video {
    height: 70vh;
    width: 70vh;
    position: absolute;
    z-index: 3;
    right: 20px;
}

```

The screenshot shows a code editor with a dark theme. The active file is 'uploadview.css'. The code defines a keyframe animation 'move-background' and a video container. The keyframes are defined for WebKit, WebKit, and Mozilla browsers, all moving a background from 0px to 1000px over 150 seconds. The video container is a relative position that contains a video element which is absolutely positioned on the right side of the container.

```

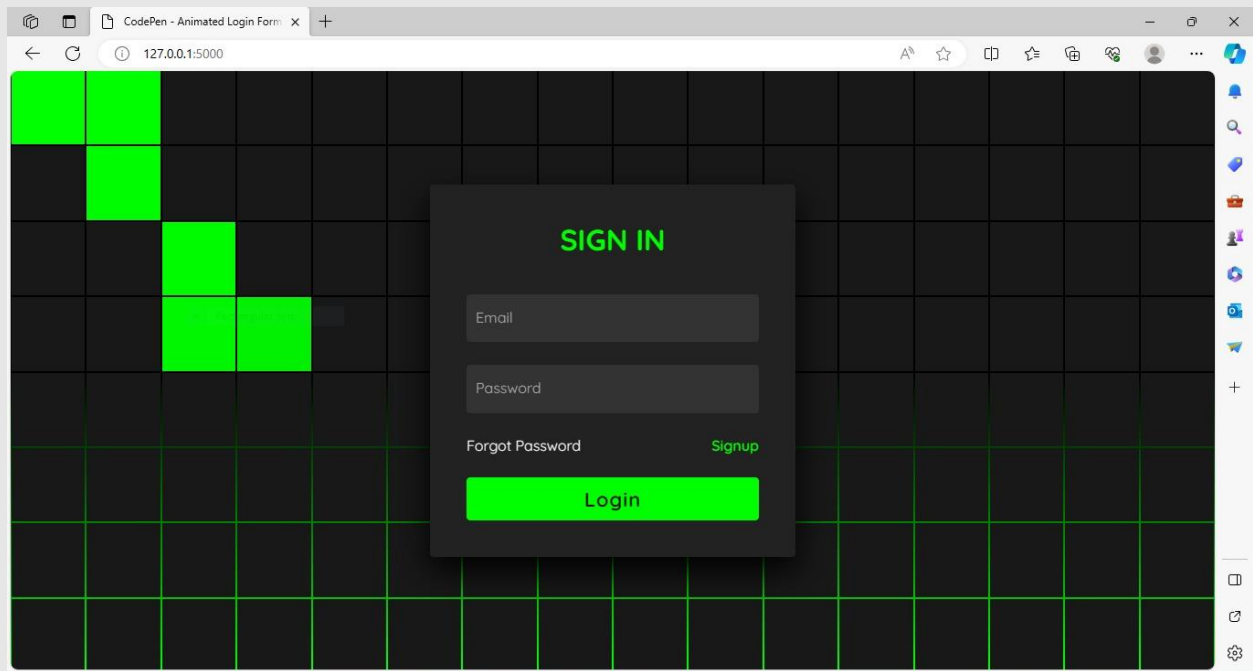
1  /* styles.css */
2  @keyframes move-background {
3      from {
4          -webkit-transform: translate3d(0px, 0px, 0px);
5      }
6      to {
7          -webkit-transform: translate3d(1000px, 0px, 0px);
8      }
9  }
10
11  @-webkit-keyframes move-background {
12      from {
13          -webkit-transform: translate3d(0px, 0px, 0px);
14      }
15      to {
16          -webkit-transform: translate3d(1000px, 0px, 0px);
17      }
18  }
19
20  @-moz-keyframes move-background {
21      from {
22          -webkit-transform: translate3d(0px, 0px, 0px);
23      }
24      to {
25          -webkit-transform: translate3d(1000px, 0px, 0px);
26      }
27  }
28
29  .video-container

```

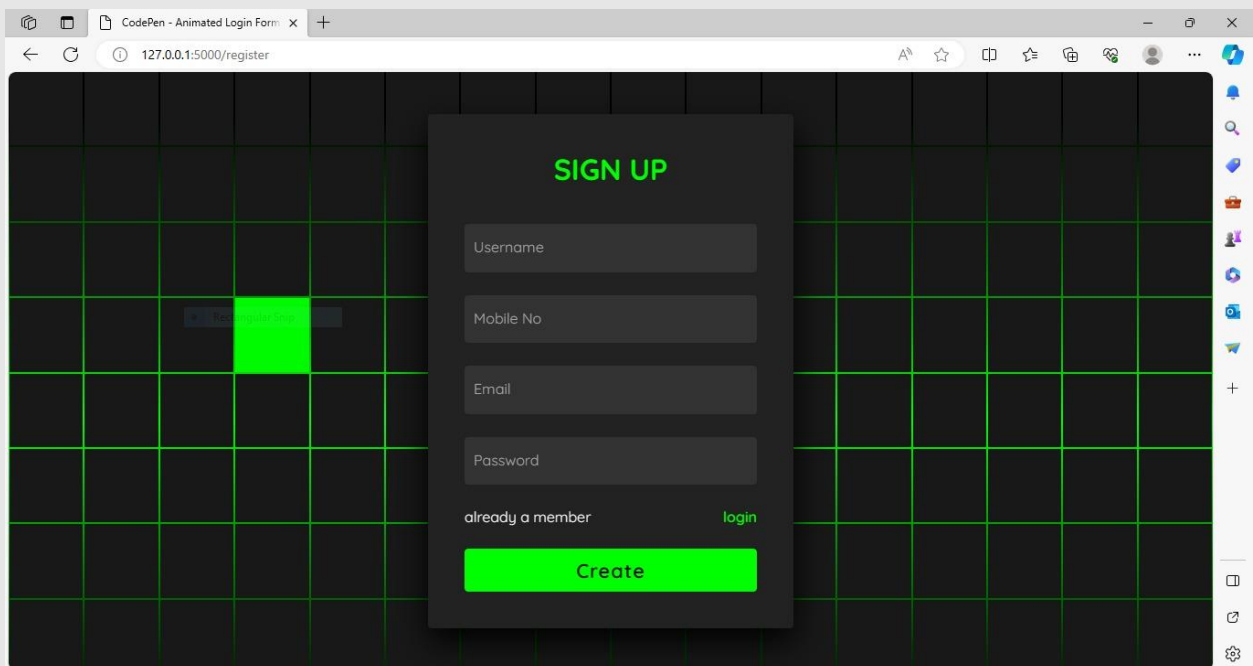
The editor's status bar at the bottom shows the file path 'flaskProject1 > static > css > uploadview.css', the encoding '84:19 (1 char)', the line ending 'CRLF', the character set 'UTF-8', the indentation '2 spaces*', the Python version 'Python 3.11 (flaskProject1)', and a small icon.

FINAL OUTPUT FOR ALL PROGRAMS:

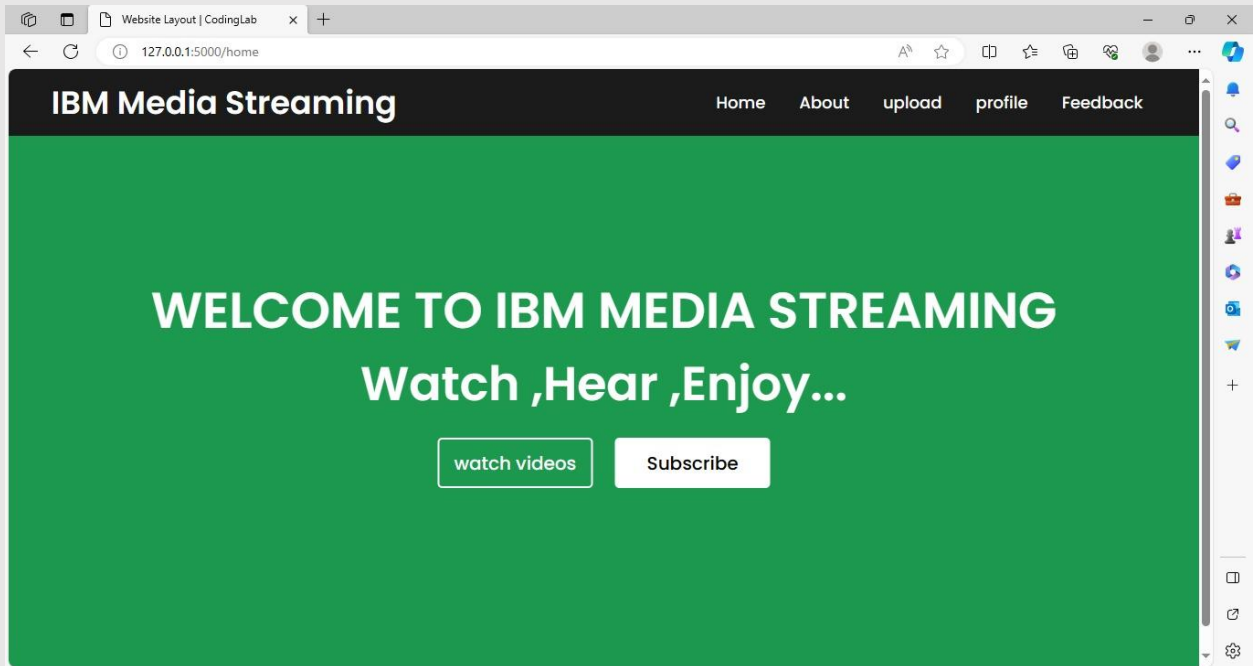
1. LOGIN PAGE TO ENTER INTO THE HOME PAGE:



2. SIGN UP PAGE TO CREATE AN ACCOUNT TO BE LOGGED IN:

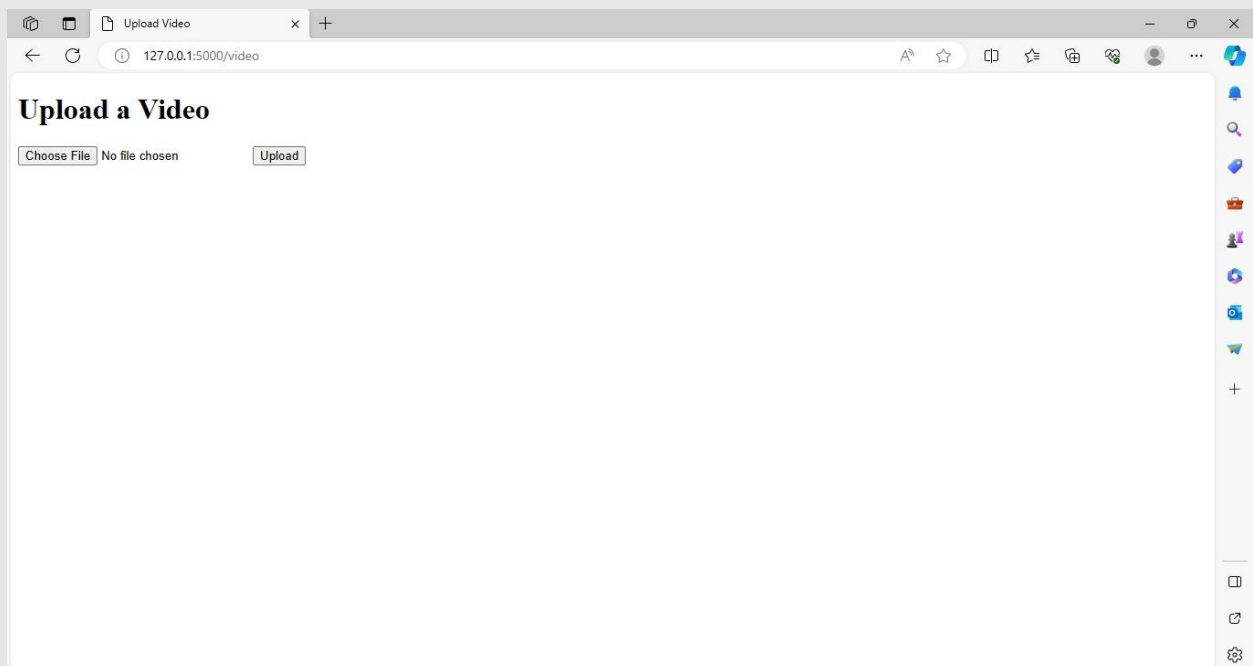


3. HOME PAGE INTERFACE FOR MEDIA STREAMING:

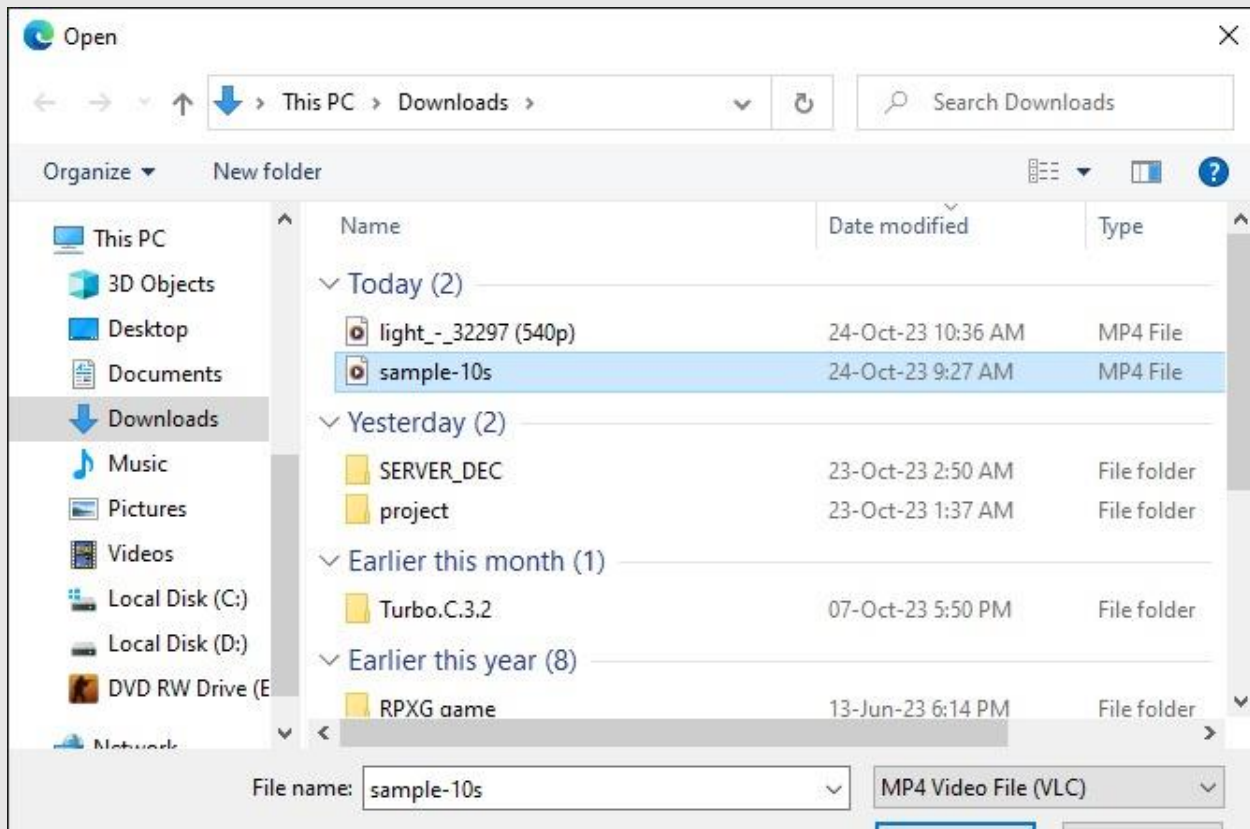


4. STEPS TO UPLOAD A FILE:

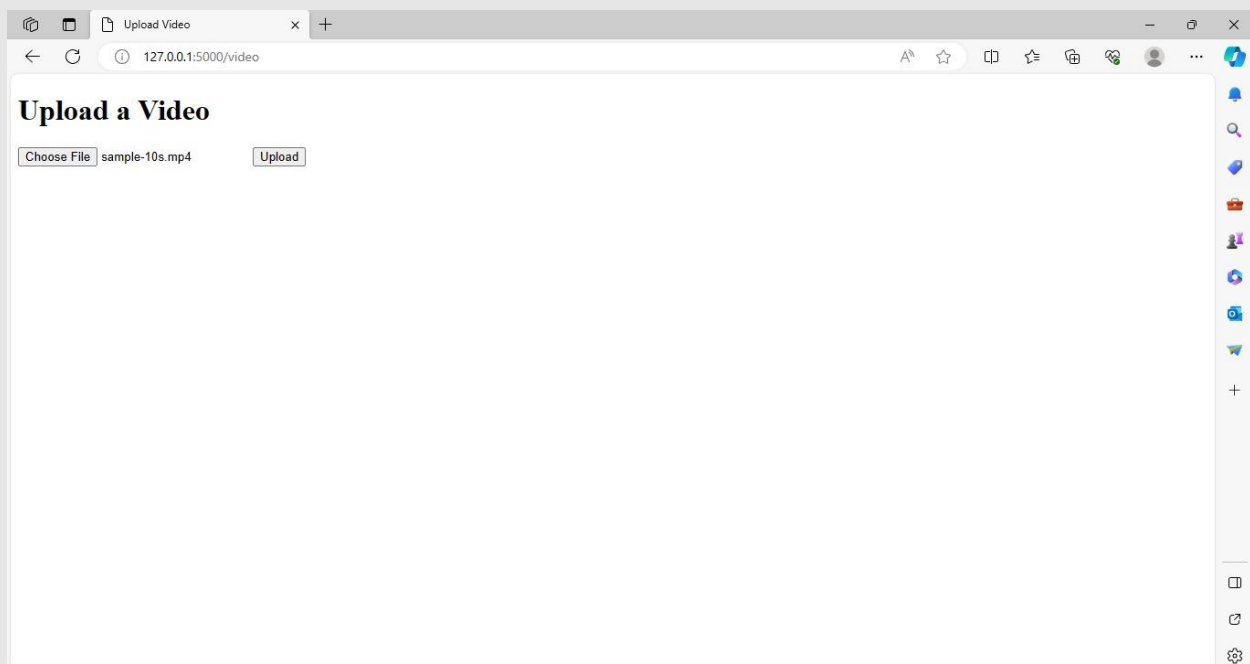
4.1) FIRST CLICK CHOOSE FILE CHECK BOX:



4.2) CHOOSE THE FILE AND CLICK OPEN TO CONTINUE:



4.3) CLICK THE UPLOAD CHECK BOX TO UPLOAD A VIDEO TO BE STREAMED:



5. STREAMING MEDIA INTERFACE:

