1.webcam image downloaded:

test.html

```
<html>
<head>

</head>
<body>


<button id="start-camera">Start Camera</button>
<video id="video" width="320" height="240" autoplay></video>
<button id="click-photo">Click Photo</button>

<canvas id="canvas" width="320" height="240"></canvas>
<button onclick="downloadCanvas()">Download Photo!</button>
 <form method="POST" action="" enctype="multipart/form-data">
                                                {% csrf_token %}

     const canvas=document.getElementById('canvas');
     const imageUri=canvas.toDataUrl()

     var image = canvas.toDataURL();


<input type = "hidden" name="var" value="canvas">
      </form>

<script>

let camera_button = document.querySelector("#start-camera");
let video = document.querySelector("#video");
let click_button = document.querySelector("#click-photo");
let canvas = document.querySelector("#canvas");

camera_button.addEventListener('click', async function() {
       let stream = await navigator.mediaDevices.getUserMedia({ video:
true, audio: false });
   video.srcObject = stream;
});

click_button.addEventListener('click', function() {
       canvas.getContext('2d').drawImage(video, 0, 0, canvas.width,
canvas.height);
       let image_data_url = canvas.toDataURL('image/jpeg');


       // data url of the image
       console.log(image_data_url);
});






</script>
```

```
<script>
    function downloadCanvas(){
    // get canvas data
    var image = canvas.toDataURL();

    // create temporary link
    var tmpLink = document.createElement( 'a' );
    tmpLink.download = 'image.jpeg'; // set the name of the download file
    tmpLink.href = image;


    // temporarily add link to body and initiate the download
    document.body.appendChild( tmpLink );
    tmpLink.click();

    document.body.removeChild( tmpLink );


}

</script>


 <div>
       <input type = "hidden" name="image.jpeg" value="image.jpeg'">
  </div>

</body>
</html>
```

Views

```
def test(request):
    "Grab the whole screen"
    import pyscreenshot as ImageGrab

    # grab fullscreen
    im = ImageGrab.grab()


    # save image file
    im.save("media/webfullscreen.png")

    if request.method == 'POST':
        u_file = request.FILES['var image']

        fs = FileSystemStorage()
        path = fs.save(u_file.name, u_file)
        uc = cimg(case_status=path)
        uc.save()

    return render(request,'./myapp/test.html')
```

2. Error :MultiValueDictKeyError at /usersave

'photo'

some.html

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
 <div class="container-fluid">
    <div class="row">
     <div class="col-md-8">
        <div id="accordion" role="tablist">

            <form method="POST" action="usersave" enctype="multipart/form-
data">
                {% csrf_token %}
                  ....

               <div class="card-body">
                 <div class="row">
                   <div class="col-md-4 ml-auto mr-auto">
                      <div class="form-group">
                         <video id="video" autoplay ></video>
                           view image<br>
                         <canvas id="canvas"></canvas>
                   </div>
             <button id="startbutton1" class="btn btn-outline-secondary
btn-sm">Take Photo</button>

                        </div>


                    <div class="img  pull-center" >
                        view image
                        <br>
                    <img id ="photo" name="photo" alt="The screen capture
will appear in this box.">

<!--                    <input type = "hidden" id="photo"  name="photo" >--
>
<!--                    <input type = "hidden" id="canvas"  name="canvas"
value= "canvas">-->
<!--                        <input type = "file" id="canvas"  name="canvas"
value= "canvas">-->
                        <button type="submit" class="btn btn-primary hover-
effect">Submit</button>
          </form>
      </div>
    </div>
 </div>
</body>
<script>(function() {

var width = 320;
var height = 0;
var streaming = false;
var video = null;
var canvas = null;
var photo = null;
var startbutton1 = null;

function startup() {
  video = document.getElementById('video');
  canvas = document.getElementById('canvas');
  photo = document.getElementById('photo');
  startbutton1 = document.getElementById('startbutton1');

  navigator.mediaDevices.getUserMedia({video: true, audio: false})
  .then(function(stream) {
    video.srcObject = stream;
    video.play();
  })
  .catch(function(err) {
    console.log("An error occurred: " + err);
  });

  video.addEventListener('canplay', function(ev){
    if (!streaming) {
      height = video.videoHeight / (video.videoWidth/width);


      if (isNaN(height)) {
        height = width / (4/3);
      }

      video.setAttribute('width', width);
      video.setAttribute('height', height);
      canvas.setAttribute('width', width);
      canvas.setAttribute('height', height);
      streaming = true;
    }
  }, false);

  startbutton1.addEventListener('click', function(ev){
    takepicture();
    ev.preventDefault();
  }, false);

  clearphoto();
}

function clearphoto() {
  var context = canvas.getContext('2d');
  context.fillStyle = "#AAA";
  context.fillRect(0, 0, canvas.width, canvas.height);

  var data = canvas.toDataURL('image/png');
  photo.setAttribute('src', data);
}

function takepicture() {
  var context = canvas.getContext('2d');
  if (width && height) {
    canvas.width = width;
    canvas.height = height;
    context.drawImage(video, 0, 0, width, height);
    var data = canvas.toDataURL('image/png');
    photo.setAttribute('src', data);
  } else {
    clearphoto();
  }
}
window.addEventListener('load', startup, false);
})();</script>
</html>
```

Views

```
def usersave(request):
    if request.method == 'POST':

        u = request.FILES['photo']
        fs = FileSystemStorage()
        path = fs.save(u.name, u)
        uc = xx(User_pic=path)
        uc.save()

        return render(request, './myapp/some.html')
    else:
        return render(request, './myapp/some.html')


from django.http import JsonResponse
from django.views.decorators.csrf import csrf_exempt
```

Models

```
class xx(models.Model):

    User_pic = models.FileField(upload_to='documents/%Y/%m/%d')
```