

Group: A

TP2 Assignment

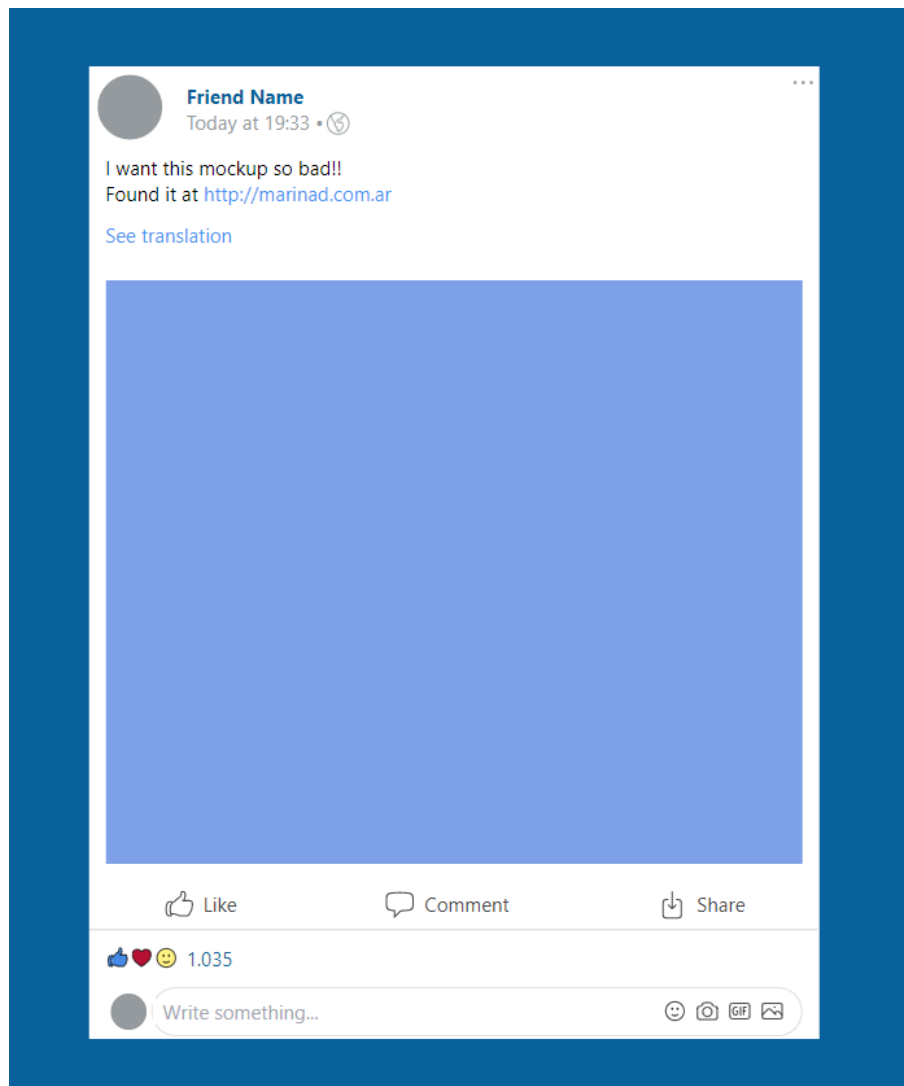
Git url: <https://github.com/nmaury/TpsCambodge/tree/main/InternetProgramming/tp2>

Ex01. Design the following UI using Tailwind

[illegible]

[illegible]

Output :



Ex02. Runaway Button

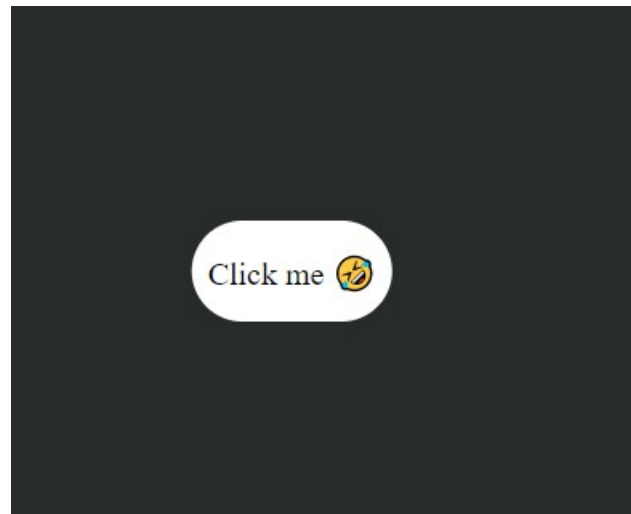
```
<!DOCTYPE html>
<html>
  <head>
    <script src="../../tween.js/dist/tween.umd.js"></script>
  </head>
  <body style="background-color: #222725">
    <div id="box" class="box" style="background-color: #FFFFFF; width: 100px; height: 50px; display: flex;
    justify-content: center; align-items: center; border-radius: 30px;">
      <div>Click me 🐼</div>
    </div>
    <script>
      //import TWEEN from '@tweenjs/tween.js'
      //setup the animation loop
      function animate(time){
        requestAnimationFrame(animate)
        TWEEN.update(time)
      }
      requestAnimationFrame(animate)

      /*const moveFromSourceToDest =(startX,startY,EndX,EndY)=>{
        const coords = {x:startX,y:startY}
        const tween = new TWEEN.Tween()
          .to({x:EndX,y:EndY},1000)
          .easing(TWEEN.Easing.Quadratic.Out)
          .onUpdate(()=>{
            console.log(`${coords.x}px,${coords.y}px`);
          })
          .start()
      }
      moveFromSourceToDest(0,0,200,150);*/

      const box = document.getElementById('box');
      function moveAway(){
        const coords = {x: box.getBoundingClientRect().x, y: box.getBoundingClientRect().y} // Start at (0, 0)
        const nextX = Math.floor(Math.random() * screen.width);
        const nextY = Math.floor(Math.random() * screen.height);

        const tween = new TWEEN.Tween(coords) // Create a new tween that modifies 'coords'.
          .to({x: nextX, y: nextY}, 1000) // Move to (300, 200) in 1 second.
          .easing(TWEEN.Easing.Quadratic.Out) // Use an easing function to make the animation smooth.
          .onUpdate(() => {
            // Called after tween.js updates 'coords'.
            // Move 'box' to the position described by 'coords' with a CSS translation.
            box.style.setProperty('transform', `translate(${coords.x}px, ${coords.y}px)`);
          })
          .start() // Start the tween immediately.
        }
        document.querySelector(".box").addEventListener('mouseover', (e)=>{moveAway()})
      }
    </script>
  </body>
</html>
```

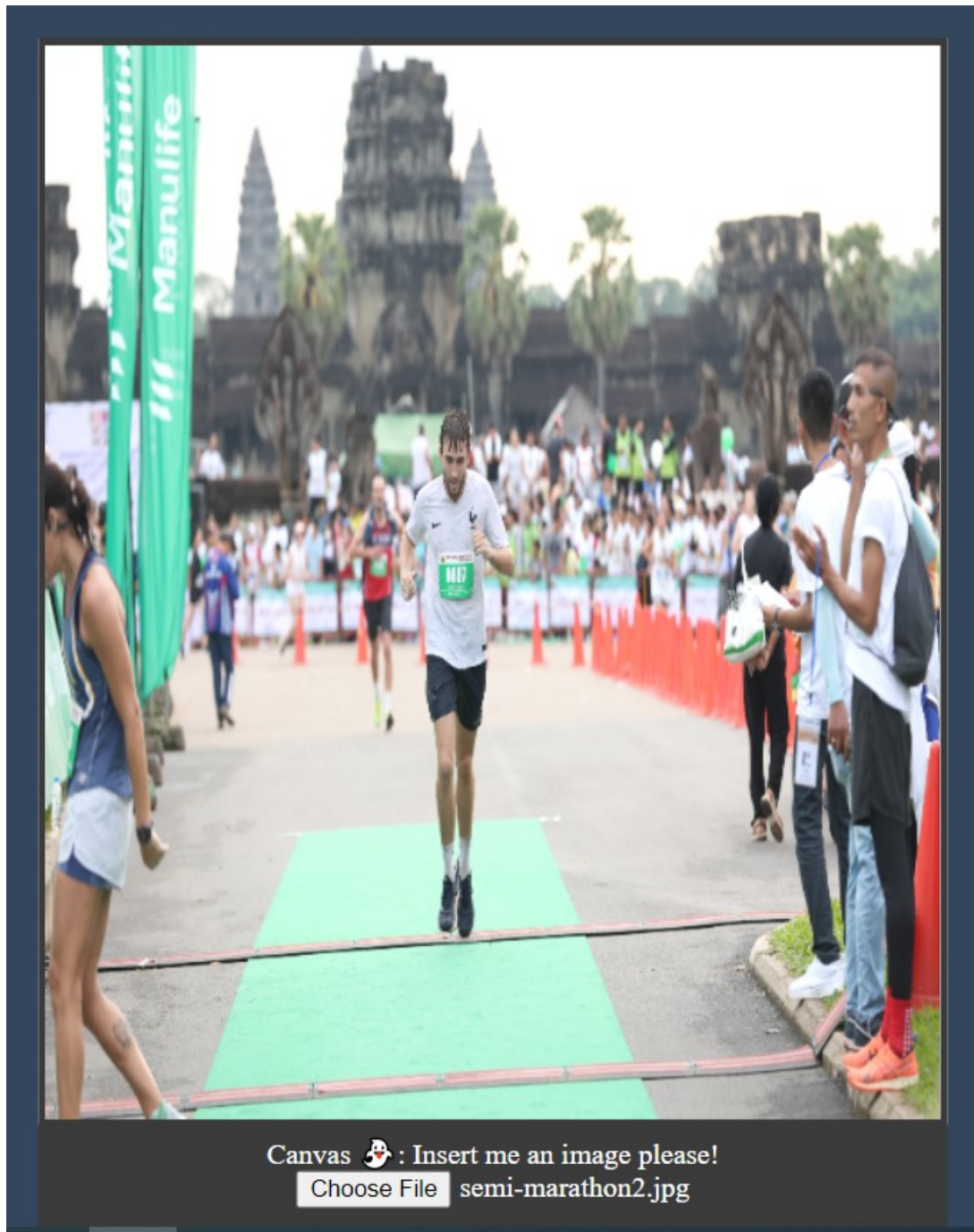
Output:



Ex03. Draw an uploaded image to a Canvas

```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body style="background-color: #2E4057">
    <canvas id="canvas" width="500px" height="600px" style="border:solid #353536 4px;background-color:#FFFFFF;position: absolute;top:50%;left:50%;transform: translate(-50%,-50%);">
    </canvas>
    <div style="text-align:center;color:#FFFFFF;width:500px;height:50px;background-color:#353536;border:solid #353536 4px;position: absolute;top:96%;left:50%;transform: translate(-50%,-50%);">
      <div>Canvas: Insert me an image please! </div>
      <div style="display:flex;justify-content: center;">
        <button class="button" onclick="document.getElementById('file-input').click();">Choose File</button>
        <input id="file-input" type="file" name="name" style="display: none;" />
        <div id="text" style="padding-left:5px">No file chosen</div>
      </div>
    </div>
    <script>
      function handleImage(uploadedImgFile) {
        var reader = new FileReader();
        reader.onload = function (event) {
          var img = new Image();
          img.onload = function () {
            const canvas = document.getElementById("canvas");
            let ctx = canvas.getContext('2d');
            ctx.clearRect(0, 0, canvas.width, canvas.height);
            ctx.drawImage(this,0,0,canvas.width,canvas.height);
          }
          img.src = event.target.result;
        }
        reader.readAsDataURL(uploadedImgFile);
      }
      const input = document.getElementById('file-input');
      input.onchange = e =>{
        console.log(input.files[0]);
        document.getElementById('text').innerHTML=input.files[0].name;
        handleImage(input.files[0]);
      }
    </script>
  </body>
</html>
```

Output:



Ex04. Draw a live camera video to a Canvas

```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body style="background-color: #2E4057">
    <video autoplay style="display:none"></video>
    <canvas id="canvas" width="500px"height="600px"style="border:solid #353536 4px;background-color:#FFFFFF;position: absolute;top:50%;left:50%;transform: translate(-50%,-50%);">
    </canvas>
    <div style="text-align:center;color: #FFFFFF;width:500px;height:50px;background-color: #353536;border:solid #353536 4px;position: absolute;top:96%;left:50%;transform: translate(-50%,-50%);">
      <div>Canvas 🐼: please show your face on me </div>
      <div style="display:flex;justify-content: center;">
        <button *ngIf="playing"class="button" onclick="playing=false">Pause</button>
      </div>
    </div>
    <script>
      var playing=true;
      var video = document.querySelector('video')
      navigator.mediaDevices.getUserMedia({
        audio: false,
        video: true
      }).then((stream)=>{
        video.srcObject=stream;
      }).catch((err)=>{
        console.log('navigator.MediaDevices.getUserMedia error: ',err.message,err.name);
      });

      (function loop(){
        if(playing){
          const canvas = document.getElementById("canvas");
          let ctx = canvas.getContext('2d');
          ctx.drawImage(video,0,0,canvas.width,canvas.height);
          setTimeout(loop,1000/30);
        }
      })();
    </script>
  </body>
</html>
```

Output:



Ex05. Draw a live camera video into multiple projection within Canvas and crop them with OpenCV library

```
<!DOCTYPE html>
<html>
  <head>
    <script src="../../opencv.js" type="text/javascript"></script>
    <style>
      .canvas-container{
        display:flex;
        flex-wrap:wrap;
        width:525px;
      }
      .item{
        width: 166.67px;
        height:200px;
      }
    </style>
  </head>
  <body style="background-color: #2E4057">
    <video autoplay style="display:none"></video>
    <div style="display:flex;position: absolute;top:50%;left:50%;transform: translate(-50%,-50%);">
      <div><canvas id="canvas" width="500px"height="600px"style="border:solid #353536 4px;background-color:#FFFFFF;">
        </canvas>
        <div style="text-align:center;color:#FFFFFF;width:500px;height:50px;background-color:#353536;border:solid #353536 4px;">
          <div>Canvas: please show your face on me </div>
          <div style="display:flex;justify-content: center;">
            <button *ngIf="playing"class="button" onclick="playing=false">Pause</button>
          </div>
        </div>
      </div>
      <div class="canvas-container">
        <canvas class="item"id="output-canvas1" style="border:solid #353536 4px;background-color:#FFFFFF;"></canvas>
        <canvas class="item"id="output-canvas2" style="border:solid #353536 4px;background-color:#FFFFFF;"></canvas>
        <canvas class="item" id="output-canvas3" style="border:solid #353536 4px;background-color:#FFFFFF;"></canvas>
        <canvas class="item"id="output-canvas4" style="border:solid #353536 4px;background-color:#FFFFFF;"></canvas>
        <canvas class="item"id="output-canvas5" style="border:solid #353536 4px;background-color:#FFFFFF;"></canvas>
        <canvas class="item"id="output-canvas6" style="border:solid #353536 4px;background-color:#FFFFFF;"></canvas>
        <canvas class="item"id="output-canvas7" style="border:solid #353536 4px;background-color:#FFFFFF;"></canvas>
      </div>
    </div>
  </body>
</html>
```

```

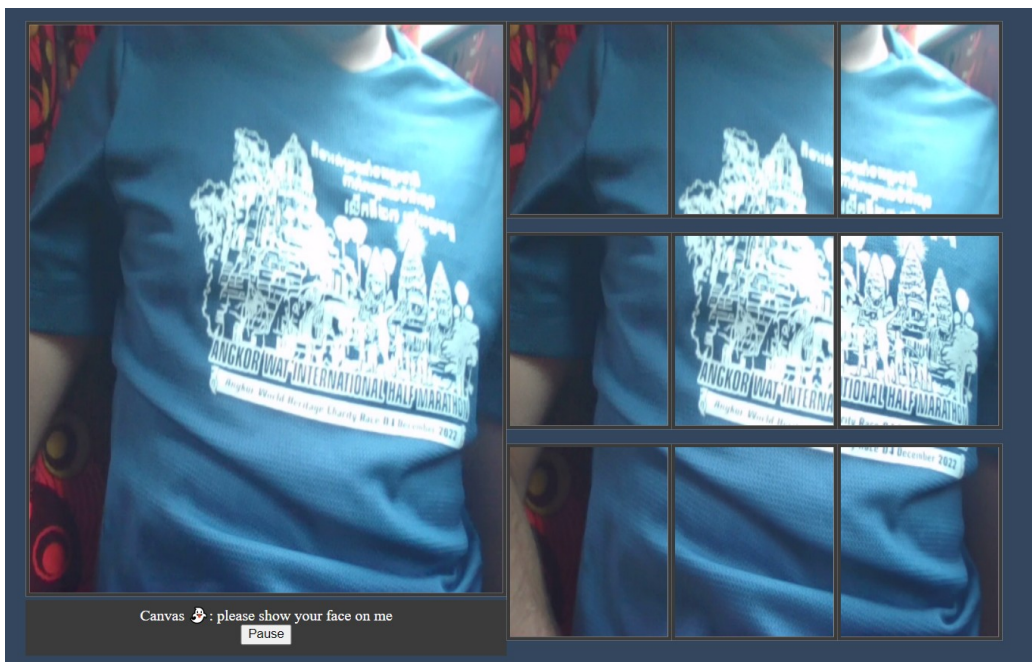
<canvas class="item" id="output-canvas8" style="border:solid #353536 4px;background-color: #FFFFFF;"></canvas>
<canvas class="item" id="output-canvas9" style="border:solid #353536 4px;background-color: #FFFFFF;"></canvas>
</div>
</div>
<script>
var playing=true;
var video = document.querySelector('video')
navigator.mediaDevices.getUserMedia({
  audio: false,
  video: true
}).then((stream)=>{
  video.srcObject=stream;
}).catch((err)=>{
  console.log('navigator.MediaDevices.getUserMedia error: ',err.message,err.name);
});
(function loop(){
  if(playing){
    const canvas = document.getElementById("canvas");
    let ctx = canvas.getContext('2d');
    ctx.drawImage(video,0,0,canvas.width,canvas.height);

    drawPart(1,0,0);
    drawPart(2,1,0);
    drawPart(3,2,0);
    drawPart(4,0,1);
    drawPart(5,1,1);
    drawPart(6,2,1);
    drawPart(7,0,2);
    drawPart(8,1,2);
    drawPart(9,2,2);

    setTimeout(loop,1000/30);
  }
})();
function drawPart(number,numberLine,numberColumn){
  let src = cv.imread('canvas');
  let dst = new cv.Mat();
  let rect = new cv.Rect(numberLine*canvas.width/3,numberColumn*canvas.height/3 , canvas.width/3, canvas.height/3);
  dst = src.roi(rect);
  cv.imshow('output-canvas'+number, dst);
}
</script>
</body>
</html>

```

Output:



Ex06. Book shop CRUD (Create, Remove, Update, and Delete)

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="tp2-06.css">
  </head>
  <body>
    <div id="div-wrapper" class="div-wrapper">
      <div class="form-wrapper">
        <div>Name:</div>
        <input id="name" placeholder="c programming">
        <div>Category:</div>
        <input id="category" placeholder="coding">
        <div>Price:</div>
        <input id="price" placeholder="1.000 riels">
        <div><button class="add">Add</button>
      </div>
      <div class="list-wrapper" id="list-wrapper">
        <!--<div class="changeInput" id="changeInput">
          <div>New name: </div>
          <input id="input" placeholder=" name">
          <button onclick="changeOK()">OK!</button>
        </div-->
      </div>
    </div>
    <script>
      document.querySelector('.add').addEventListener('click', (e) => {
        const name = document.getElementById('name').value;
        const category = document.getElementById('category').value;
        const price = document.getElementById('price').value;
        if (name === "") {
          alert("the name is missing");
        } else if (category === "") {
          alert("the category is missing");
        } else if (price === "") {
          alert("the price is missing");
        } else {
          addItem(name, category, price);
        }
      });
      var listItems = [];
      function addItem(name, category, price) {
        listItems.push({ name: name, category: category, price: price });
        displayItems();
      }
    </script>
  </body>
</html>
```

```

function displayItems(){
  var listContent="";
  listItems.forEach(elt=>{
    const name = encodeURIComponent(elt.name);
    listContent=listContent+`<div class="item">
    <div class="button-wrapper">
      <button onClick=deleteClick('${name}')>Delete</button>
      <button onClick=changeName('${name}')>Change name</button>
    </div>
    <div>
      <svg style="height:100px;padding:30px" xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24" strokeWidth={1.5} stroke="#595457" className="w-6 h-6">
        <path strokeLinecap="round" strokeLinejoin="round" d="M12 6.042A8.967 8.967 0 06 3.75c-1.052 0-2.062 1.8-3 .512v14.25A8.987 8.987 0 016 2.292m0-14.25a8.966
        </svg>
      </div>
      <div>Name: ${elt.name}</div>
      <div>Price: ${elt.price}</div>
      <div>Category:${elt.category}</div>
    </div>`;
  });
  document.getElementById('list-wrapper').innerHTML=listContent;
}

function deleteClick(name){
  console.log(name);
  listItems.forEach(elt=>{
    if(encodeURIComponent(elt.name)===name){
      listItems.pop(elt);
    }
  });
  displayItems();
}

function changeName(name){
  const newDiv = document.createElement("div");
  newDiv.classList.add("changeInput");
  const divText = document.createElement("div");
  const newContent = document.createTextNode("New name: ");
  divText.appendChild(newContent);
  const inputChange = document.createElement("input");
  const button = document.createElement("button");
  const buttonText = document.createTextNode("OK!");
  button.appendChild(buttonText);
  button.addEventListener('click', (e)=>{
    if(inputChange.value===""){
      alert("the new name is empty!")
    }else{
      listItems.forEach(elt=>{

```

```

        if(encodeURIComponent(elt.name)===name){
          elt.name = encodeURIComponent(inputChange.value);
          newDiv.remove();
          displayItems();
        }
      });
    }

    newDiv.appendChild(divText);
    newDiv.appendChild(inputChange);
    newDiv.appendChild(button);

    const currentDiv = document.getElementById('list-wrapper');
    document.getElementById("div-wrapper").insertBefore(newDiv,currentDiv);
    //document.getElementById('changeInput').display="block";
  }
  /*function changeOk(){
    console.log(document.getElementById('changeInput').value);
  }*/

```

</script>

</body>

</html>

You, 11 hours ago • internet programming OK ...

```
.div-wrapper{
  display: flex;
}
.form-wrapper{
  width: 250px;
  height: 700px;
  background-color: #CCDBDC;
}
.list-wrapper{
  width: 900px;
  height: 700px;
  background-color: #CCDBDC;
  margin-left: 10px;
  display: flex;
  flex-wrap: wrap;
}
.item{
  width: 200px;
  height: 250px;
  background-color: #FFFFFF;
  margin: 10px;
  text-align: center;
}
.button-wrapper{
  display: flex;
  justify-content: flex-end;
}
.changeInput{
  position: absolute;
  top: 60px;
  left: 450px;
  width: 220px;
  height: 70px;
  background-color: #FFFFFF;
  border-radius: 10px;
}
```

Output:

Name:

java

Category:

coding


Price:

2000

Add

Delete

Change name




Name: c

Price: 1000

Category: coding

Delete

Change name



Name: java

Price: 2000

Category: coding