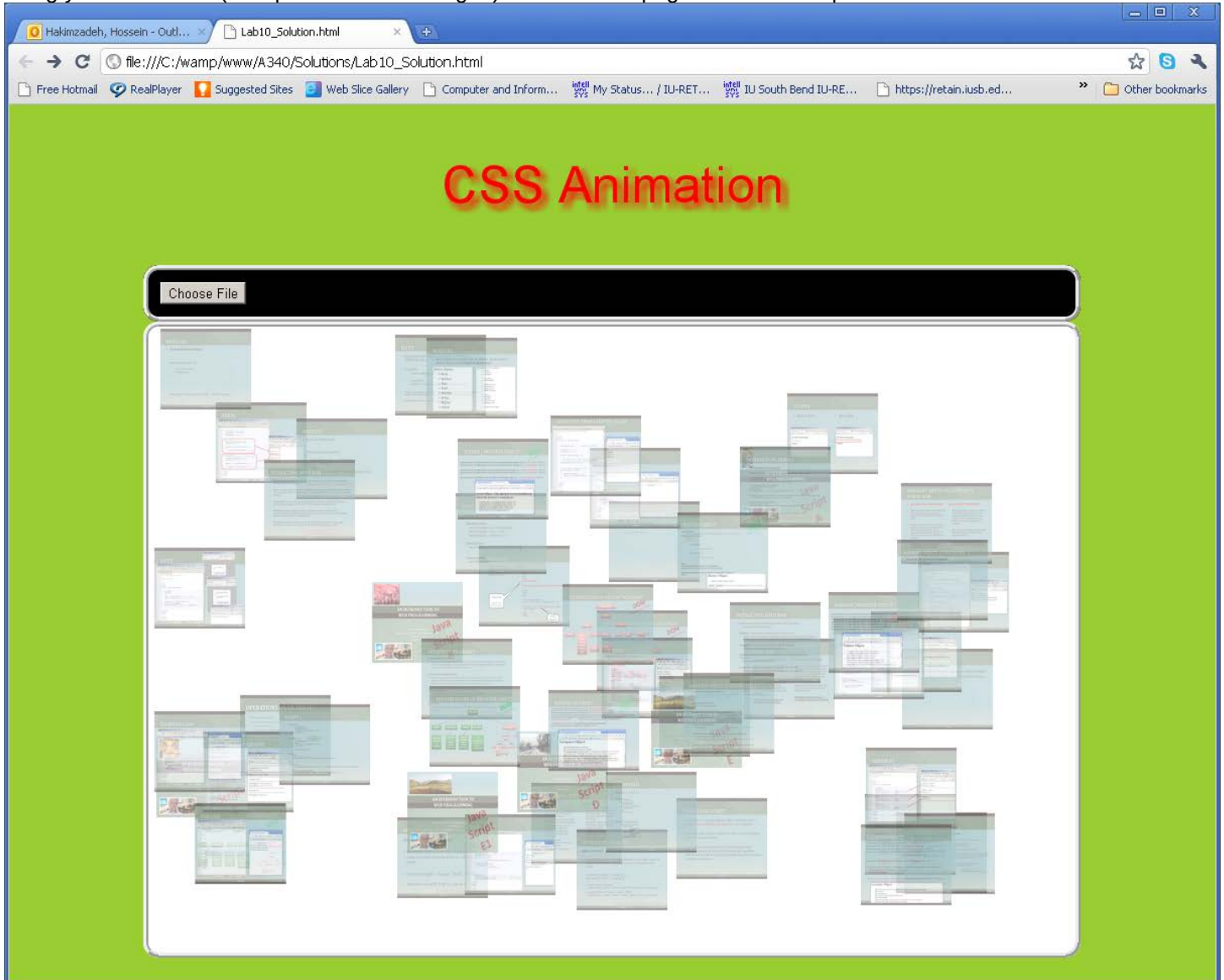


Laboratory Session #10

LAB GOALS

- CSS
- Use of inline, embedded or external CSS to manipulate the following properties: (positioning and layering, z-index, transformation, borders, simple event handling)

Using your text editor (Notepad++ / TextWrangler) create a web page similar to the pictures shown below.



Step 1: Start with a simple HTML web site similar to figure below. The body includes only 3 components. A paragraph and two DIVs. The first DIV has an <input> tag which you should be familiar with from previous labs and assignments.

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
  <p id="title" >
    CSS Animation
  </p>

  <div id="reverse" >
    <input id="files"
      type="file"
      name="files[]"
      accept="image/*"
      onchange="loadImages();"
      multiple
    />
  </div>

  <div id="images" style="height:600px; position:relative;" >
    Images will go here...
  </div>

</body>
</html>
```

Step 2: The rest of the work is going to be done using one JavaScript function called “loadImage()”, and a few CSS rules. The JavaScript function is not much different than what you did in assignment number 2. It simply extracts the file names from the <input> tag array and copies them into an array called myImages[]. Then it creates a long string which holds an image tag for each image file in the myImages[] array. Finally, it stores the long string of image tags into the innerHTML of the second DIV tag (see above). You should also notice that the image tag is slightly different than before. This image tag has its own “inline” style attribute which allows us to position each image at a specific “absolute” location on the screen.

```
function loadImages()
{
  var myImages = new Array();

  var myFiles = document.getElementById("files");

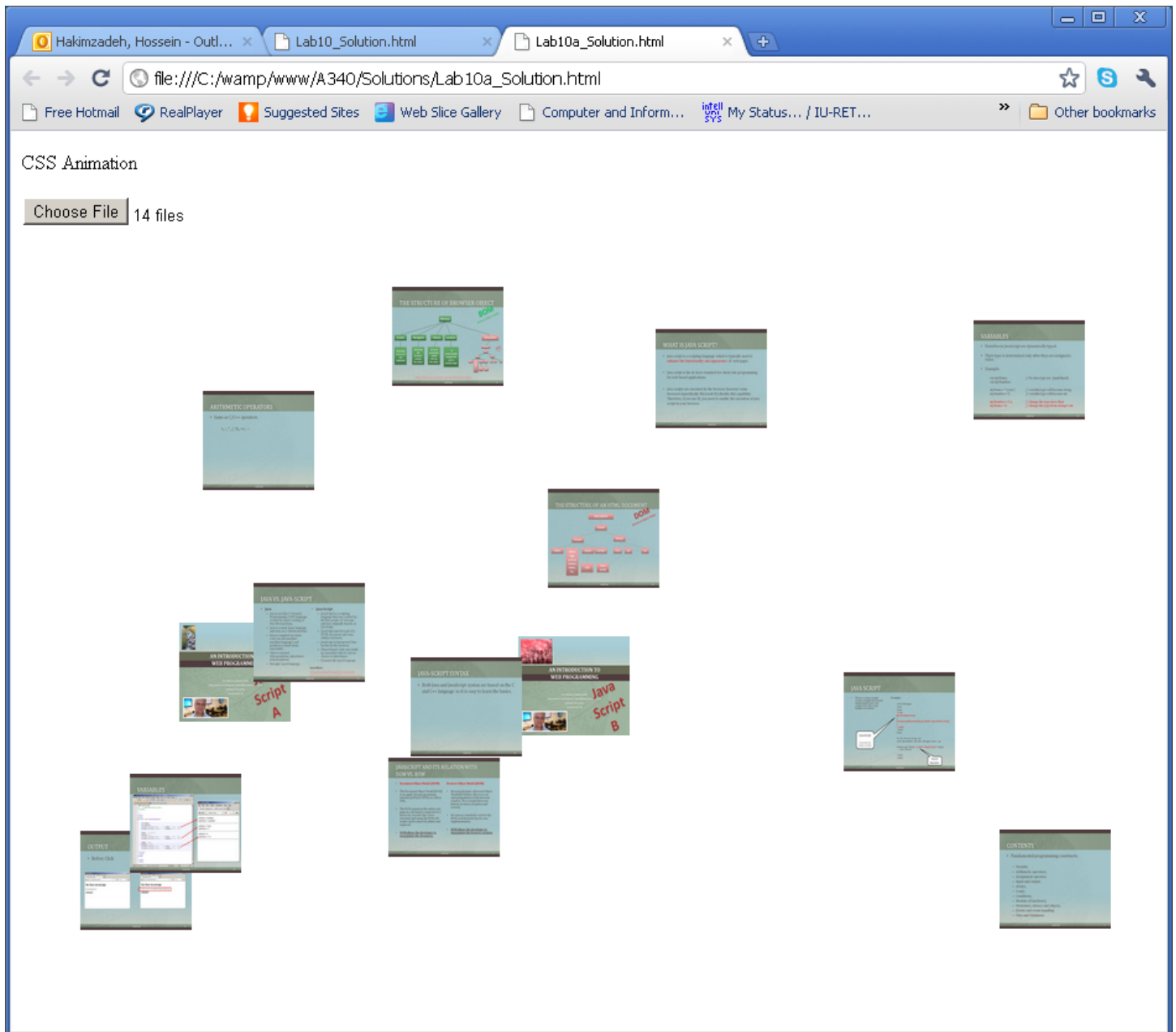
  for (var i=0; i < myFiles.files.length; i++) // you can avoid this for loop if you wish.
  {
    myImages[i] = myFiles.files[i].name;
  }

  var theImages = document.getElementById("images");
  theImages.innerHTML = ""; // Clearout the image area.
  for (var i=0; i < myFiles.files.length; i++)
  {
    randomTop = Math.floor(Math.random() * 500);
    randomLeft = Math.floor(Math.random() * 800);
    theImages.innerHTML = theImages.innerHTML +

    "<img id='pic" + i + "' style='padding:1px; " +
    "    left : " + randomLeft + "px; " +
    "    top  : " + randomTop + "px; " +
    "    position:absolute; z-index: " + i + "," +
    "    src='" + myImages[i] + "' +
    "    width = '90' " +
    "    height= '80' " +
    "  /> ";

  }
}
```

Step 3: After step two test your program to see how it works. Your output should look similar to the one shown below:



Step 3: Create a series of CSS rules that treats the following elements. (Create these rules as embedded CSS)

Element	Parameter	Specification
Body of the document	Font	Arial,Helvetica,sans-serif;
	Font color	White
	Background	yellowgreen
ID Selector with the ID of "title"	Justification	Center
	Font size	50 pixel
	Font color	Red
	Text shadow	5px, North-East, with 10px blur and color of your choice (perhaps red)
DIV	Font color	Black
	Background	White
	Width	900 pixels
	Margins	Left and right should both be auto (this will center your div)
	Padding	10 pixels in every direction
	Border	Thick border, double border style, and rounded corners (15 pixel radius)
DIV with ID of "reverse"	Font Color	White
	Background	Black
DIV with ID of "reverse" and a "hover" event	Background	Sienna
Img	Original Opacity	50%
	Transition	transition:all 2s; /* 2 second transition */ -moz-transition:all 2s; /* Firefox 4 */ -webkit-transition:all 2s; /* Safari and Chrome */ -o-transition:all 2s; /* Opera */
Img with "hover" event	Opacity	100%
	Transformation	(translate 40 pixels down, scale x 6, and rotate counter clockwise by 6 degrees. -moz-transform: translateY(40px) scale(6) rotate(-5deg); -webkit-transform: translateY(40px) scale(6) rotate(-5deg); -o-transform: translateY(40px) scale(6) rotate(-5deg); transform: translateY(40px) scale(6) rotate(-5deg);

Step 4: Once the CSS rules are included, your program should look like the figure in page 1. If you hover your mouse over any image the image will become larger (gradually within 2 seconds) and will tilt counter clockwise. Moving your mouse away will cause the image to go back to its original size and orientation. (See figure below)

