Name: Mustafa Baran Ercan
ID: 2881055520

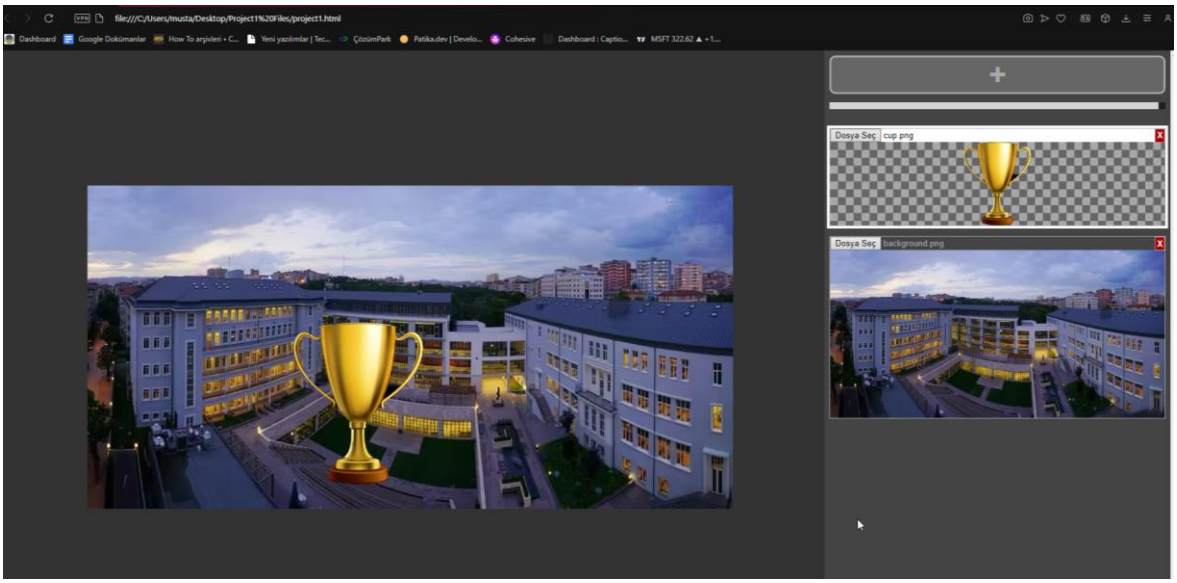
Section: CMPE_360 –SEC01
Assignment: Project 1

PROJECT 1 REPORT

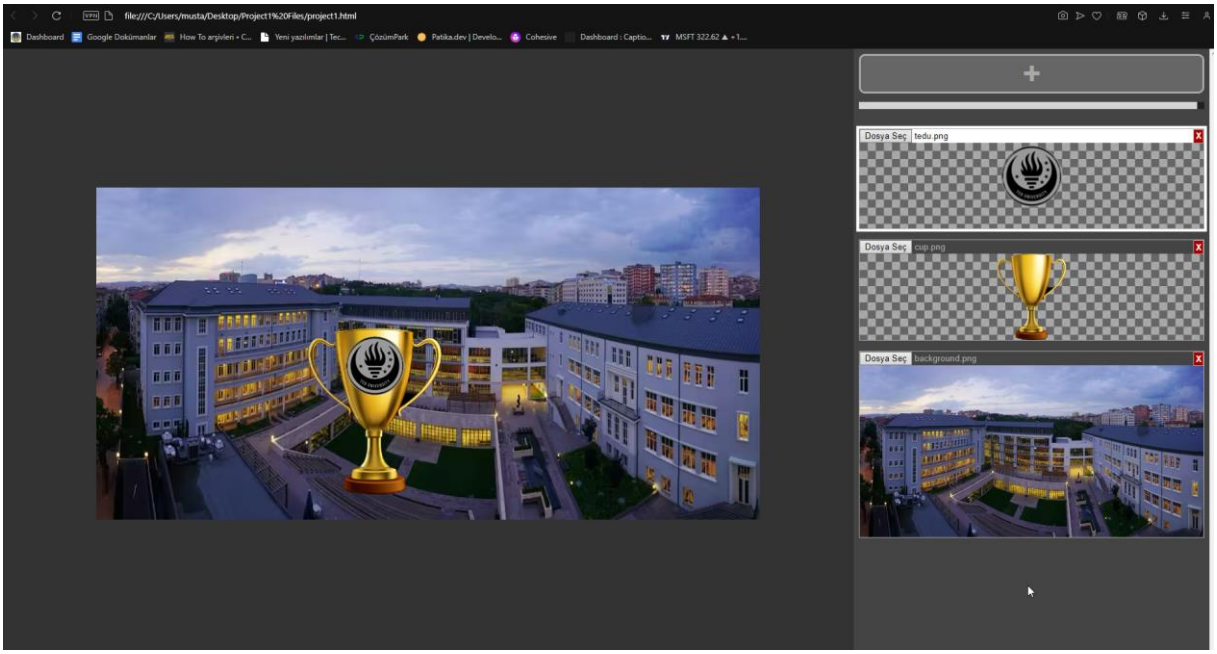
PART 1

For this part, I was able to create and save the necessary images for the assignment.

- Save and add the image with cup.png and background.png. (5 points)

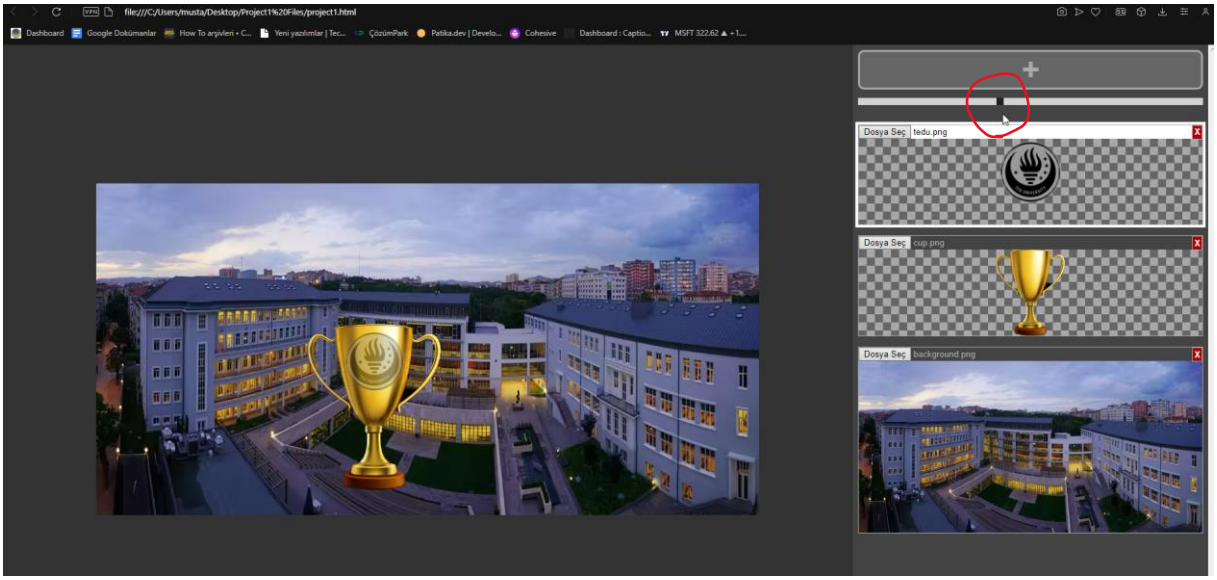


- Save and add the image with cup.png, tedu.png and background image. (5 points)



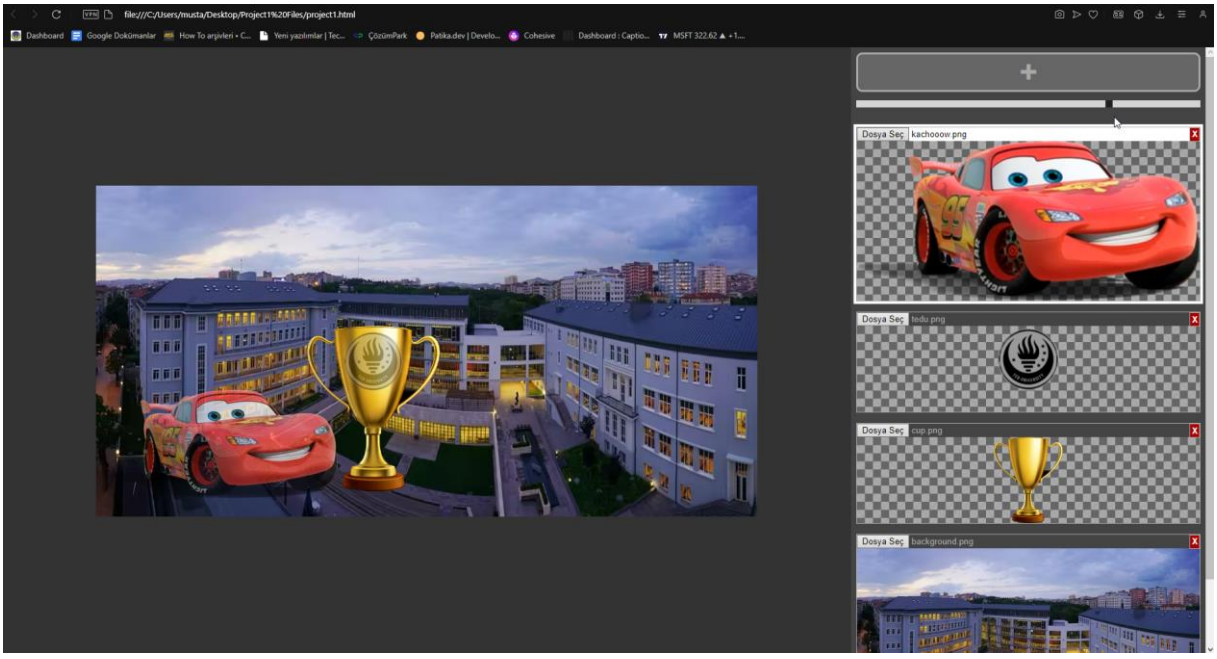
- Save and add the image with cup.png, tedu.png and background image with changing transparency and explain your process. (5 points)

  In order to give transparency to the logo (tedu.png), I adjusted the level of opacity by scrolling it to the left from the bar under the “+” sign.



- Create a new combined image by adding cup.png, tedu.png and background.png and any additional .png image you want (you can add it by finding a .png image on the internet) and save this final image and add it to your report. Also, your new added image must be transparency. (5 points)

I have chosen the fastest car alive to stand near with his new tedu-cup trophy. I also gave him a little transparency.



PART 2 Explain your Composite function in detail. Please explain all process you wrote in detailed. (30 points)

For this part, I managed to complete the composite function which blends each pixel of the foreground image onto the corresponding pixel of the background image.

First, the function loops over each pixel in the foreground image. This is done using two nested loops that iterate over the width and height of the foreground image. For each pixel in the foreground image, the function calculates index values in an array (indexFg and indexBg). Since each pixel possesses red, green, blue (RGB), and alpha (opacity) values, we allocate 4 elements for pixels. The function then gets the RGB and alpha values for the current pixel in both foreground and background images. After that, it performs alpha-blending on the images. It does that by calculating new RGB and alpha values for each foreground pixel corresponding to the background pixel. Finally, the function updates the corresponding pixels in the background image with newly calculated RGB and alpha values. However, the alpha value is multiplied by 255 because in our code we work with an alpha value between 0 and 1 (0:fully transparent, 1:opaque), and we need to convert it back to the actual image alpha value data, which is between 0 and 255.

PART3

I uploaded the project1.js file into the VPL.