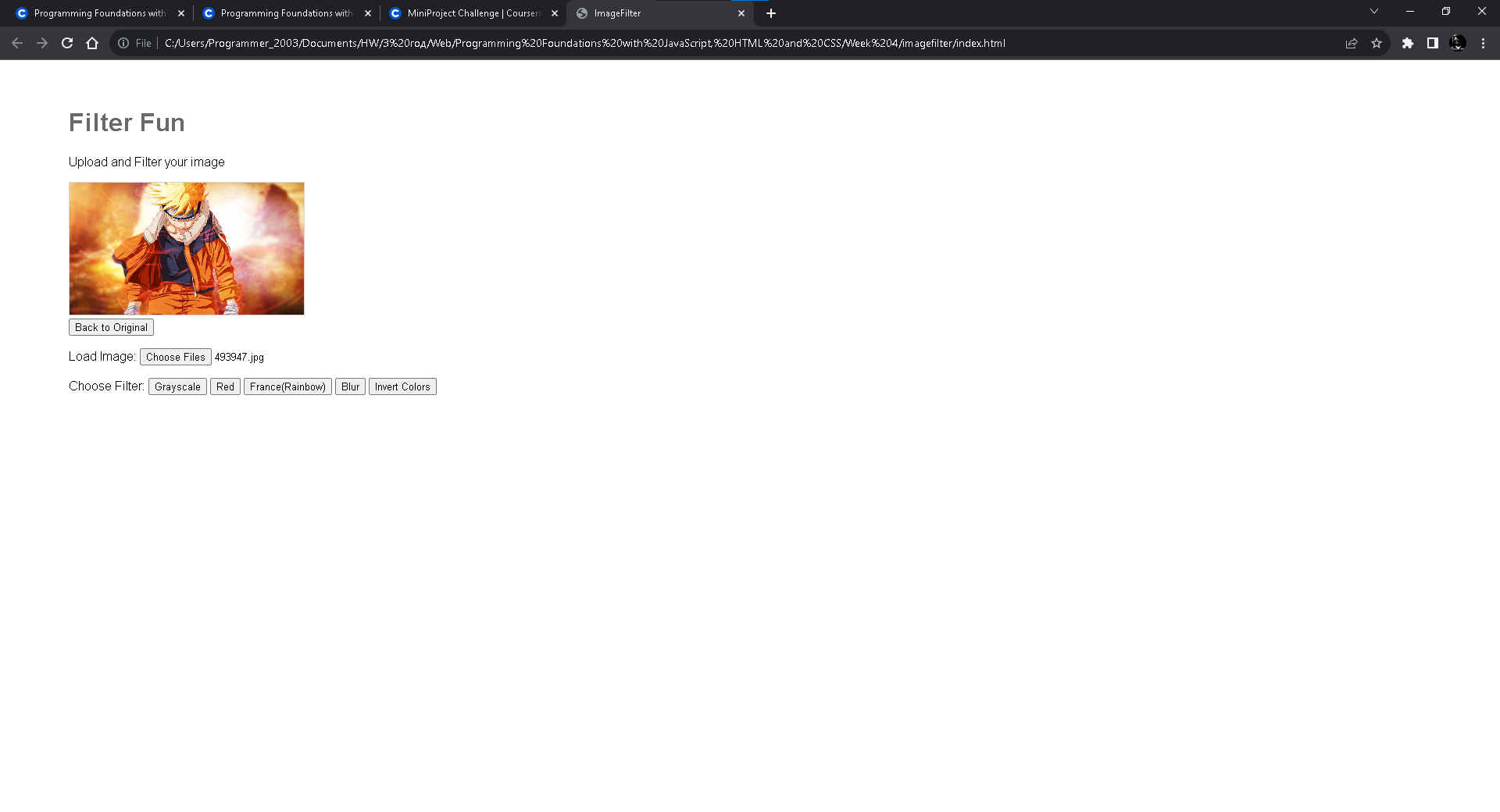
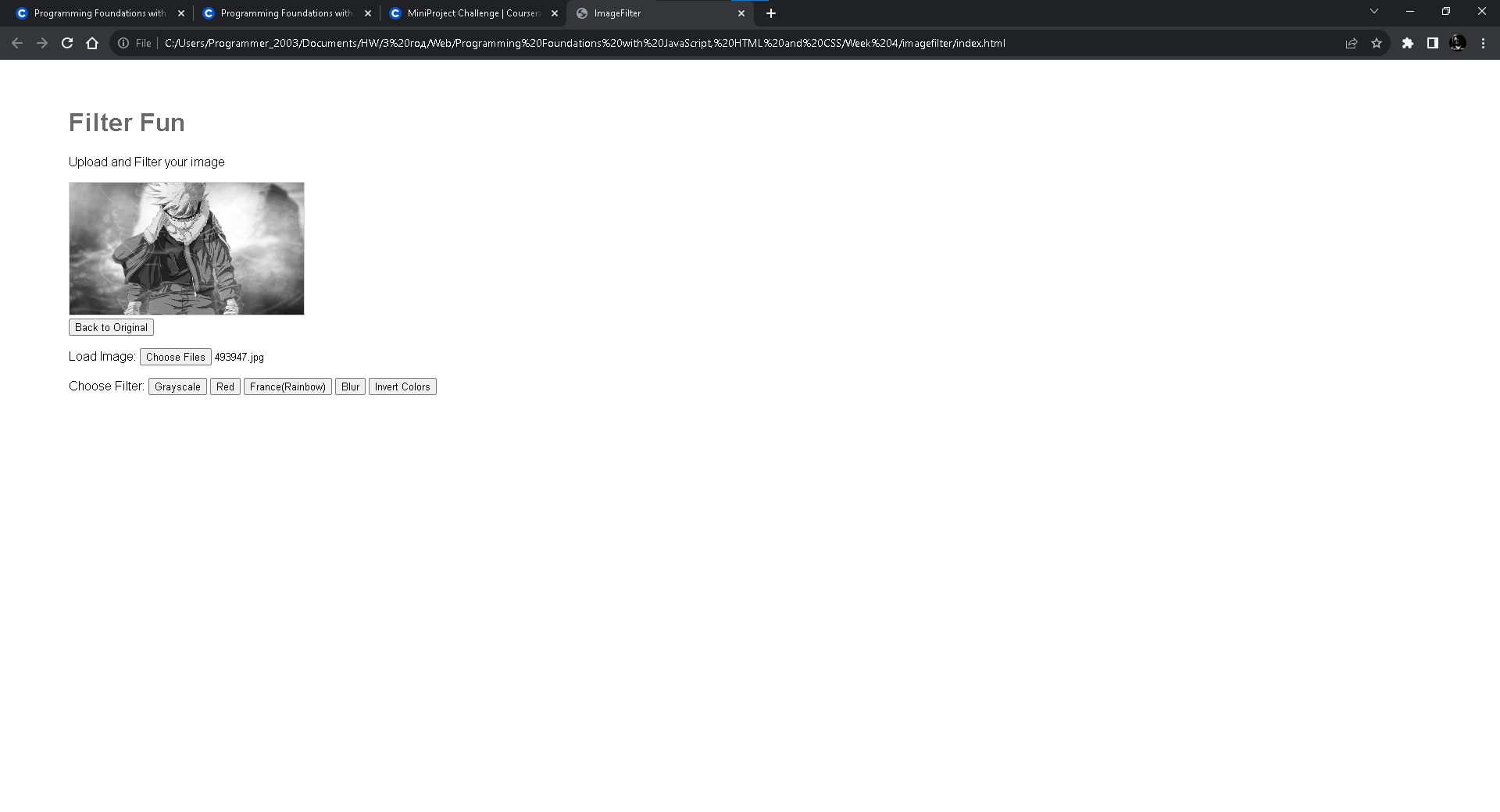
MiniProject: Part 1 & Part 2 & MiniProject Challenge

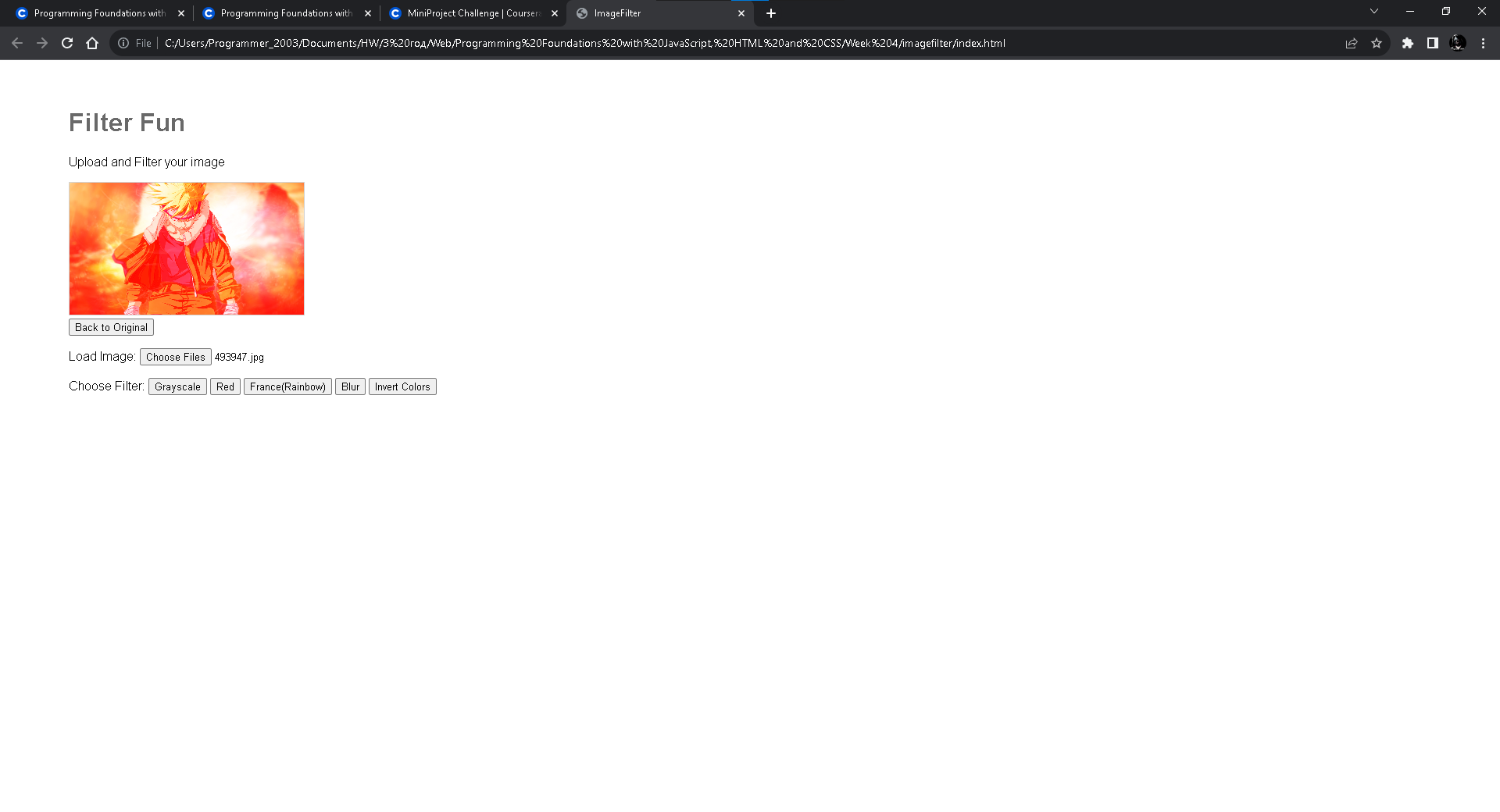
Choose File:



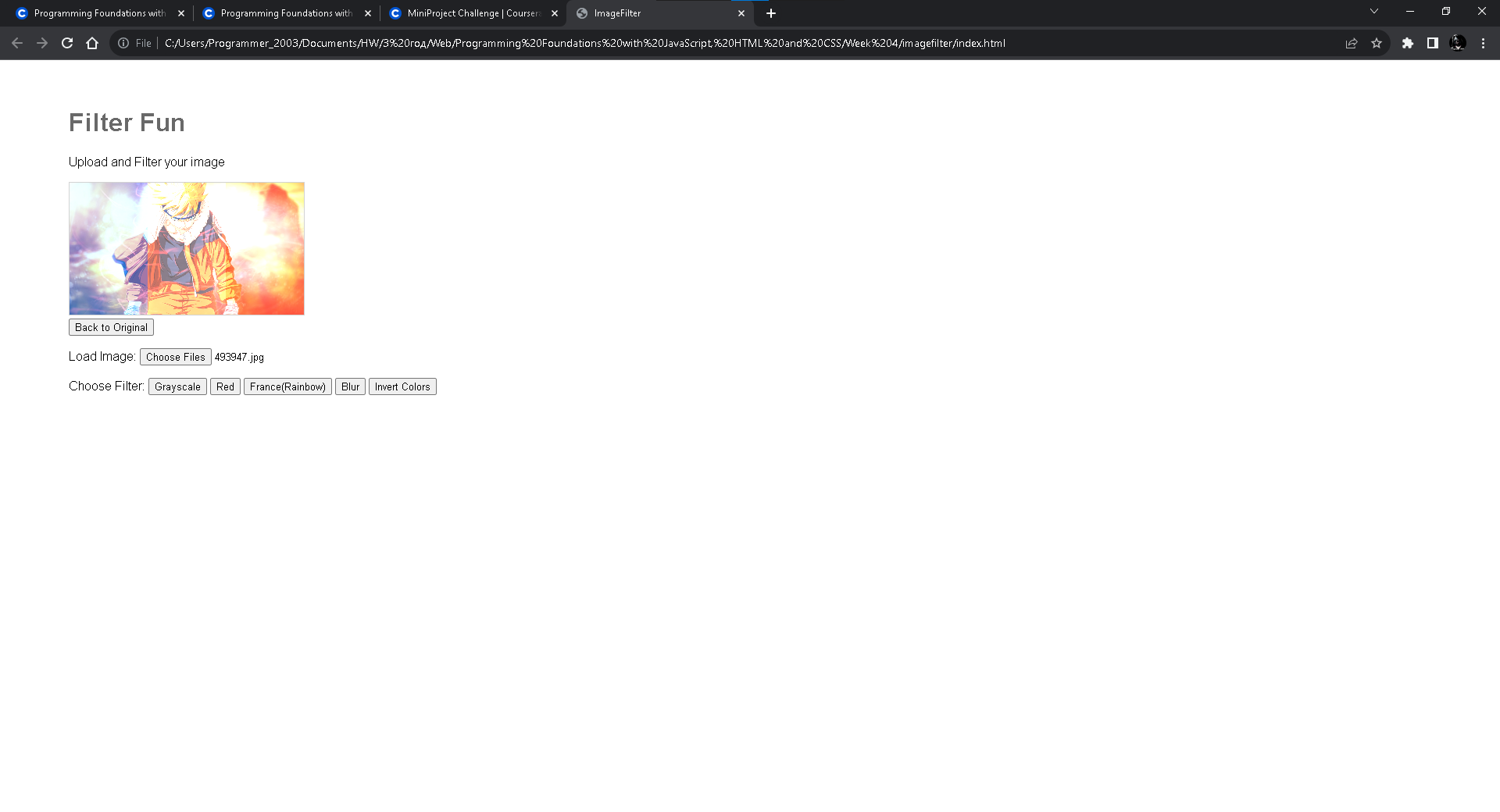
Grayscale:



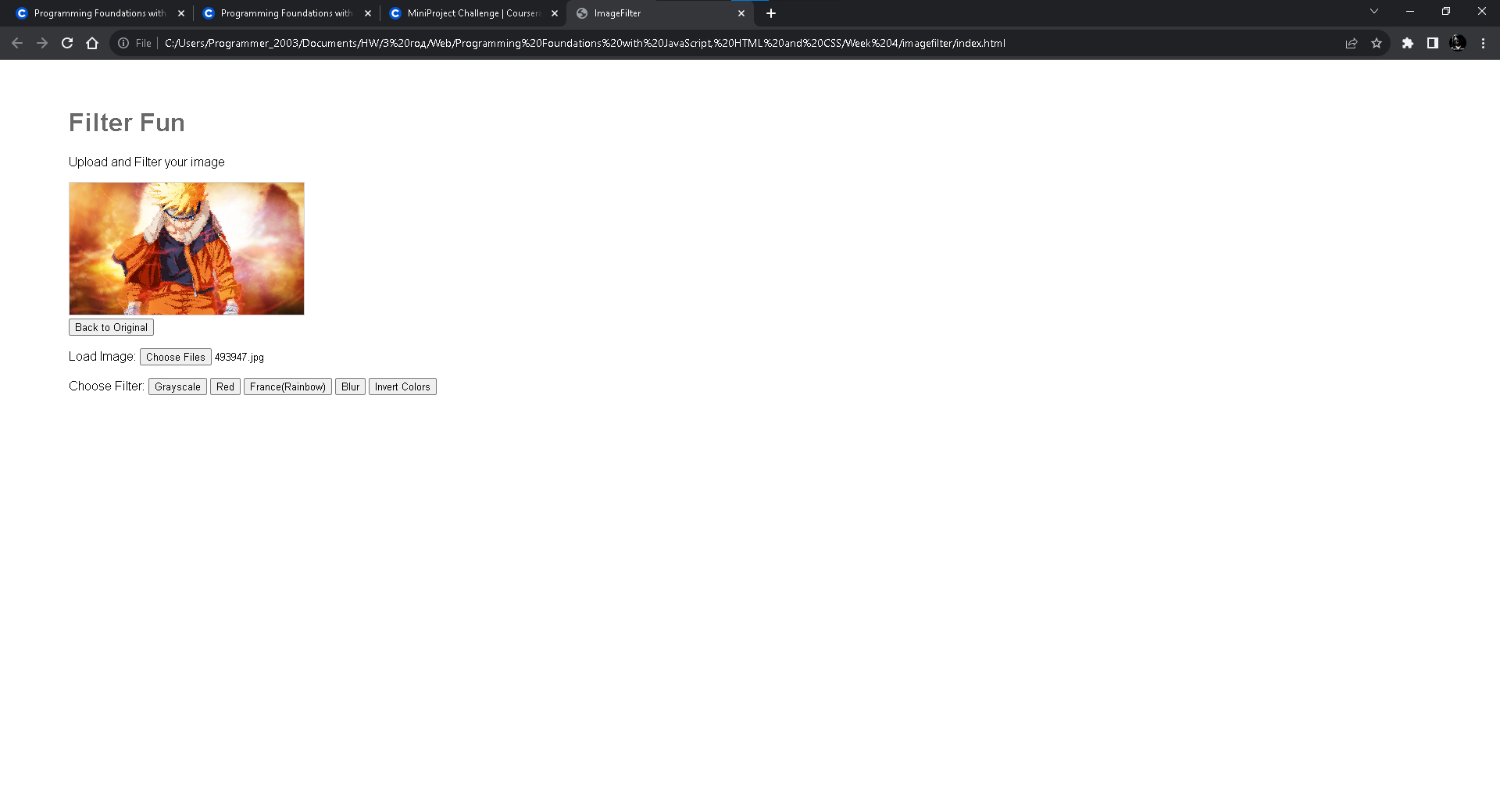
Red:



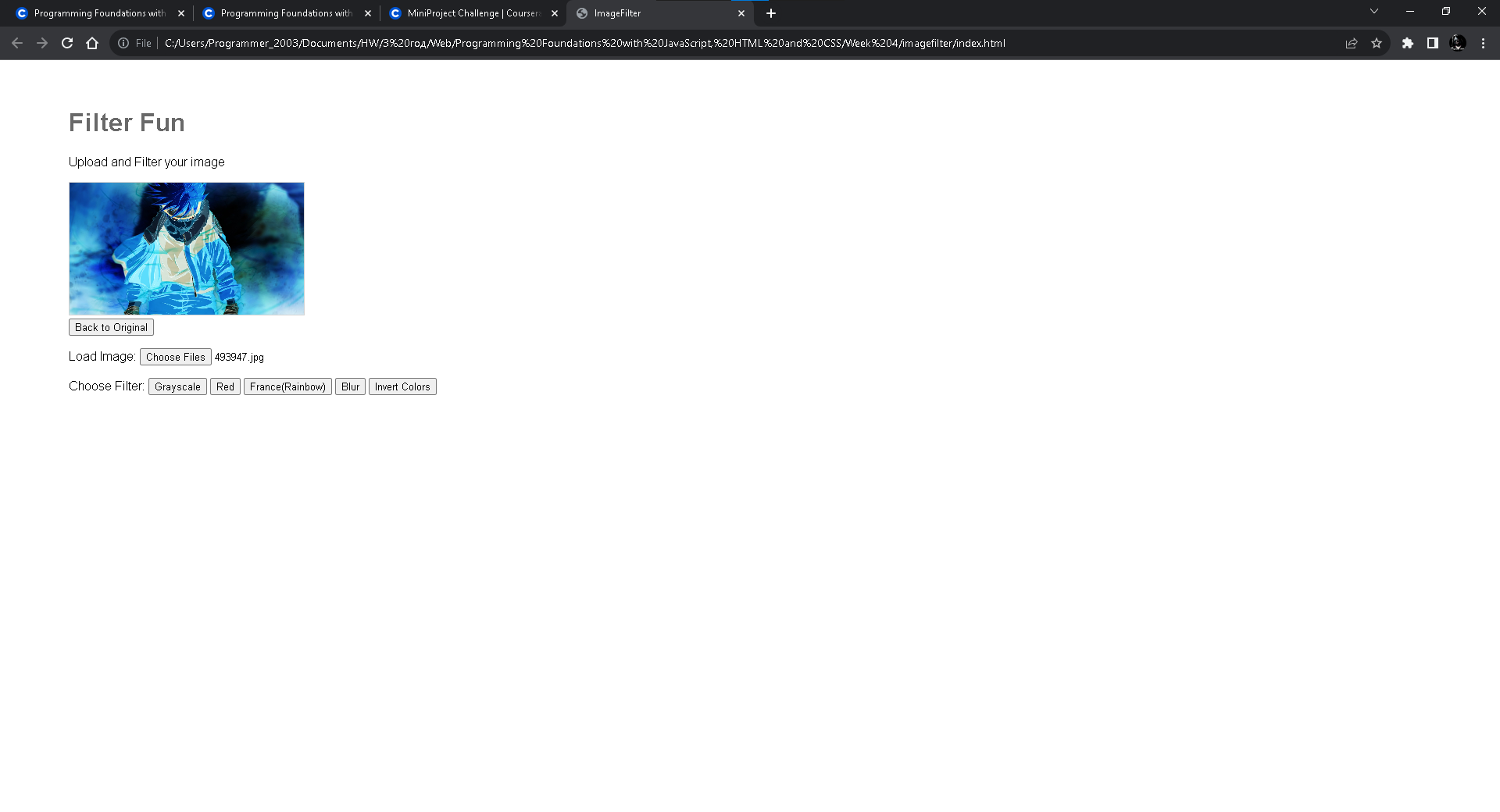
France(Rainbow):



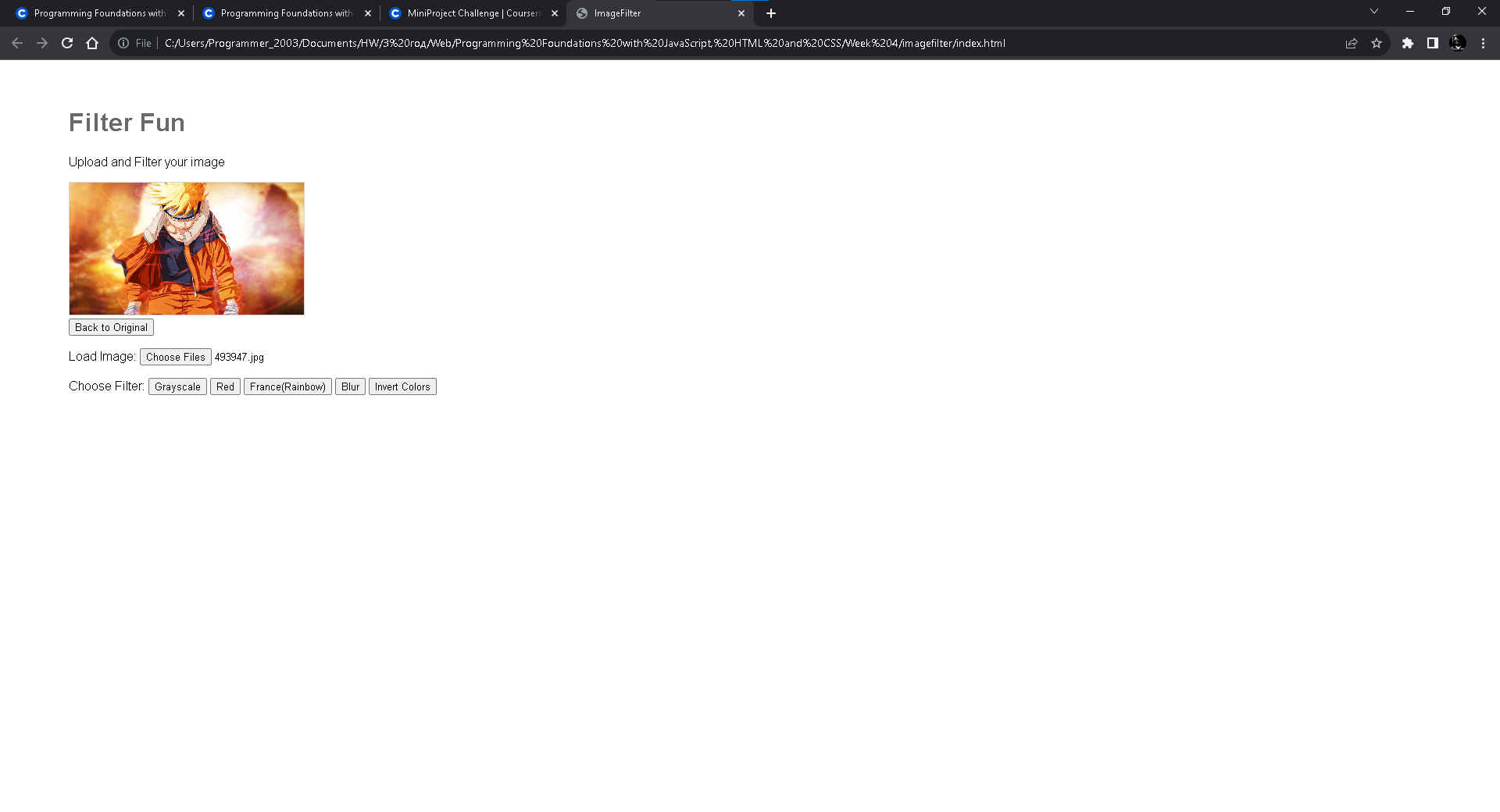
Blur:



Invent Colors:



Back to Original:



**index.html:**

<!DOCTYPE html>

<html lang="en" >

<head>

<meta charset="UTF-8">

<title>ImageFilter</title>

<link rel="stylesheet" href="./style.css">

</head>

<body>

<!-- partial:index.partial.html -->

<script src="https://www.dukelearntoprogram.com/course1/common/js/image/SimpleImage.js"></script>

<div class="body">

<h1>Filter Fun</h1>

<p>Upload and Filter your image</p>

<div class="canvas">

<canvas id="canvas"></canvas>

</div>

<div class="bar">

<input type="button" value="Back to Original" id="backtoOriginal" onclick="backtoOriginal()">

</div>

<div class="image">

<p>Load Image:

<input type="file" multiple="false" accept="image/\*" id="image" onchange="loadImage()">

</p>

</div>

<div class="filter">

<p>Choose Filter:

<input type="button" value="Grayscale" id="grayscale" onclick="doGrayscale()">

<input type="button" value="Red" id="red" onclick="doRed()">

<input type="button" value="France(Rainbow)" id="france" onclick="doFrance()">

<input type="button" value="Blur" id="blur" onclick="doBlur()">

<input type="button" value="Invert Colors" id="invertColors" onclick="doInvertColors()">

</p>

</div>

</div>

<!-- partial -->

<script src="./script.js"></script>

</body>

</html>

**style.css:**

body {

font-family: arial;

}

.body {

padding: 2em 5em;

}

h1 {

color: dimgray;

}

canvas {

border: 1px solid lightgray;

width: 300px;

}

**script.js:**

var canvasElement = document.getElementById("canvas");

var imageElement = document.getElementById("image");

var image;

var filteredImage;

function loadImage () {

image = new SimpleImage(imageElement);

image.drawTo(canvasElement);

}

function backtoOriginal () {

image.drawTo(canvasElement);

}

function doGrayscale() {

if (image == null || !image.complete()) {

alert("No image loaded yet");

}

else {

grayscale();

filteredImage.drawTo(canvasElement);

}

}

function doRed() {

if (image == null || !image.complete()) {

alert("No image loaded yet");

}

else {

red();

filteredImage.drawTo(canvasElement);

}

}

function doFrance() {

if (image == null || !image.complete()) {

alert("No image loaded yet");

}

else {

france();

filteredImage.drawTo(canvasElement);

}

}

function doBlur () {

if (image == null || !image.complete()) {

alert("No image loaded yet");

}

else {

blur();

filteredImage.drawTo(canvasElement);

}

}

function grayscale () {

filteredImage = new SimpleImage(image.getWidth(), image.getHeight());

for (var pixel of image.values()) {

var x = pixel.getX();

var y = pixel.getY();

filteredPixel = filteredImage.getPixel(x,y)

var average = (pixel.getRed() + pixel.getGreen() + pixel.getBlue()) / 3;

filteredPixel.setRed(average);

filteredPixel.setGreen(average);

filteredPixel.setBlue(average);

}

}

function red () {

filteredImage = new SimpleImage(image.getWidth(), image.getHeight());

for (var pixel of image.values()) {

var x = pixel.getX();

var y = pixel.getY();

filteredPixel = filteredImage.getPixel(x,y)

filteredPixel.setRed(255);

filteredPixel.setGreen(pixel.getGreen());

filteredPixel.setBlue(pixel.getBlue());

}

filteredImage.drawTo(canvasElement);

}

function france () {

filteredImage = new SimpleImage(image.getWidth(), image.getHeight());

var w = image.getWidth();

for (var pixel of image.values()) {

var x = pixel.getX();

var y = pixel.getY();

filteredPixel = filteredImage.getPixel(x,y);

var red = pixel.getRed();

var green = pixel.getGreen();

var blue = pixel.getBlue();

var maxNum = 255;

var minNum = 0;

if (x < w/3) {

filteredPixel.setRed(red);

Math.max(minNum, Math.min(maxNum, filteredPixel.setGreen(green + 85)));

Math.max(minNum, Math.min(maxNum, filteredPixel.setBlue(blue + 164)));

}

else if (x >= w/3 && x < 2\*w/3) {

Math.max(minNum, Math.min(maxNum, filteredPixel.setRed(red + 100)));

Math.max(minNum, Math.min(maxNum, filteredPixel.setGreen(green + 100)));

Math.max(minNum, Math.min(maxNum, filteredPixel.setBlue(blue + 100)));

}

else {

Math.max(minNum, Math.min(maxNum, filteredPixel.setRed(red + 239)));

Math.max(minNum, Math.min(maxNum, filteredPixel.setGreen(green + 65)));

Math.max(minNum, Math.min(maxNum, filteredPixel.setBlue(blue + 53)));

}

}

}

function blur () {

filteredImage = new SimpleImage(image.getWidth(), image.getHeight());

for (var pixel of image.values()) {

var x = pixel.getX();

var y = pixel.getY();

if (0.5 < Math.random()) {

filteredImage.setPixel(x,y,pixel);

}

else {

do {

randomX = x+getRandomInt(10);

randomY = y+getRandomInt(10);

} while (!isValid(randomX, image.getWidth()) || !isValid(randomY, image.getHeight()));

console.log(!isValid(randomX, image.getWidth()) + ", " + randomX + ", " + randomY);

randomPixel = image.getPixel(randomX, randomY);

filteredImage.setPixel(x,y,randomPixel);

}

}

}

function getRandomInt(num) {

if (0.5 < Math.random()) {

return Math.floor(Math.random() \* num);

}

else {

return -Math.floor(Math.random() \* num);

}

}

function isValid (num, max) {

if (0 <= num && num < max) {

return true;

}

else {

return false;

}

}

function doInvertColors() {

if (image == null || !image.complete()) {

alert("No image loaded yet");

} else {

invertColors();

filteredImage.drawTo(canvasElement);

}

}

function invertColors() {

filteredImage = new SimpleImage(image.getWidth(), image.getHeight());

for (var pixel of image.values()) {

var x = pixel.getX();

var y = pixel.getY();

filteredPixel = filteredImage.getPixel(x, y);

// Invert each color component

filteredPixel.setRed(255 - pixel.getRed());

filteredPixel.setGreen(255 - pixel.getGreen());

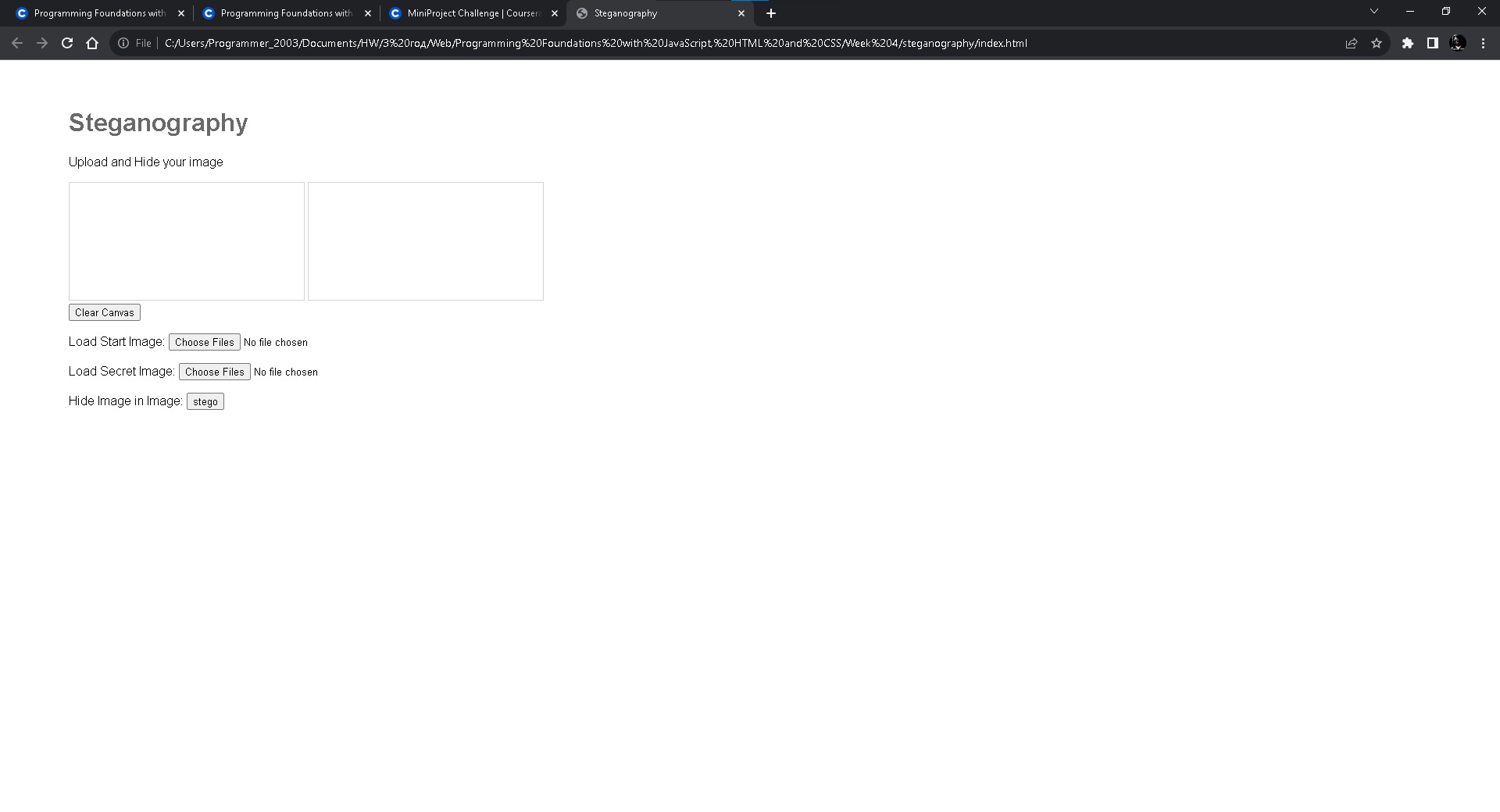
filteredPixel.setBlue(255 - pixel.getBlue());

}

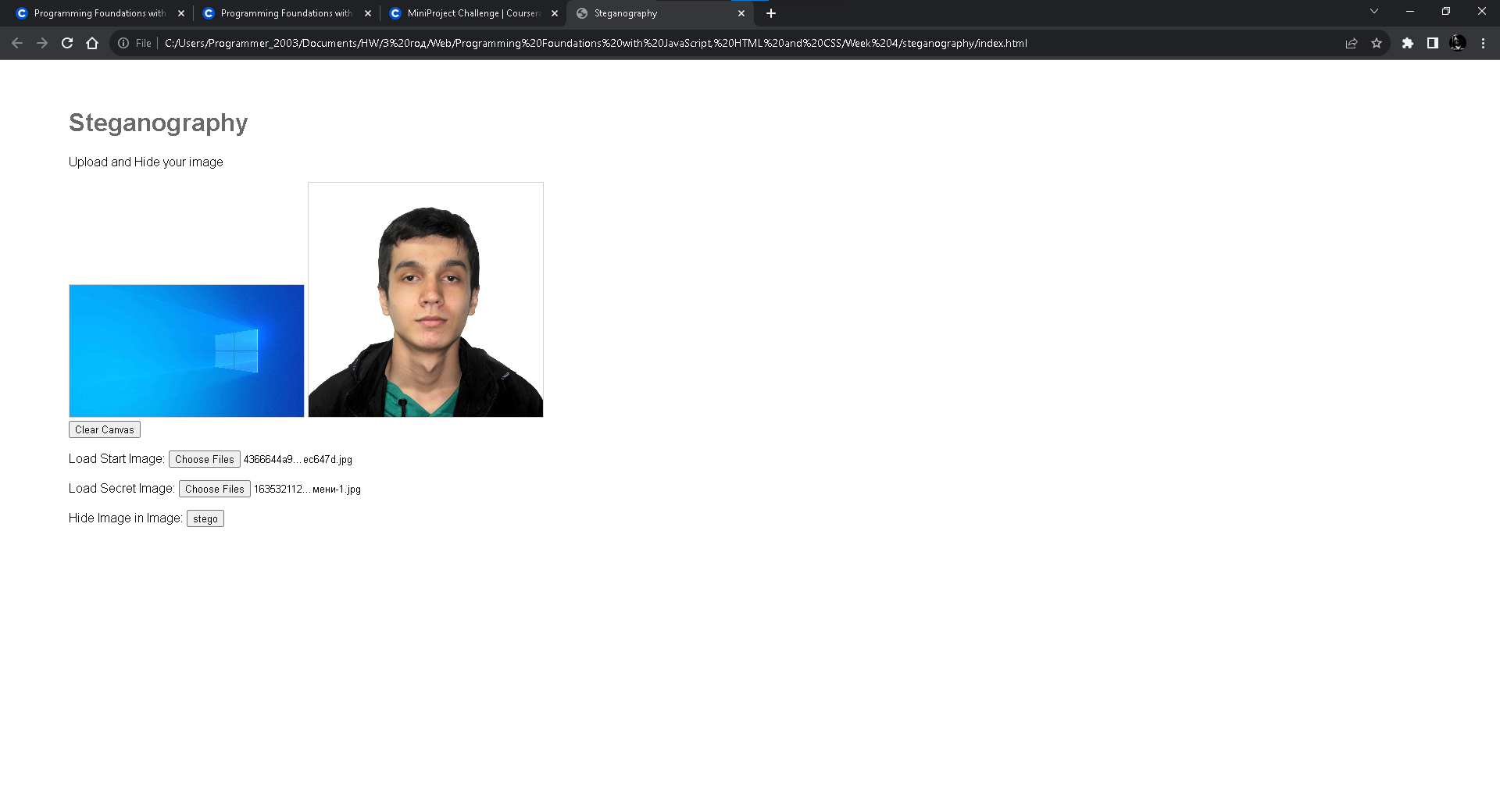
}

Steganography: Part 1, 2, 3

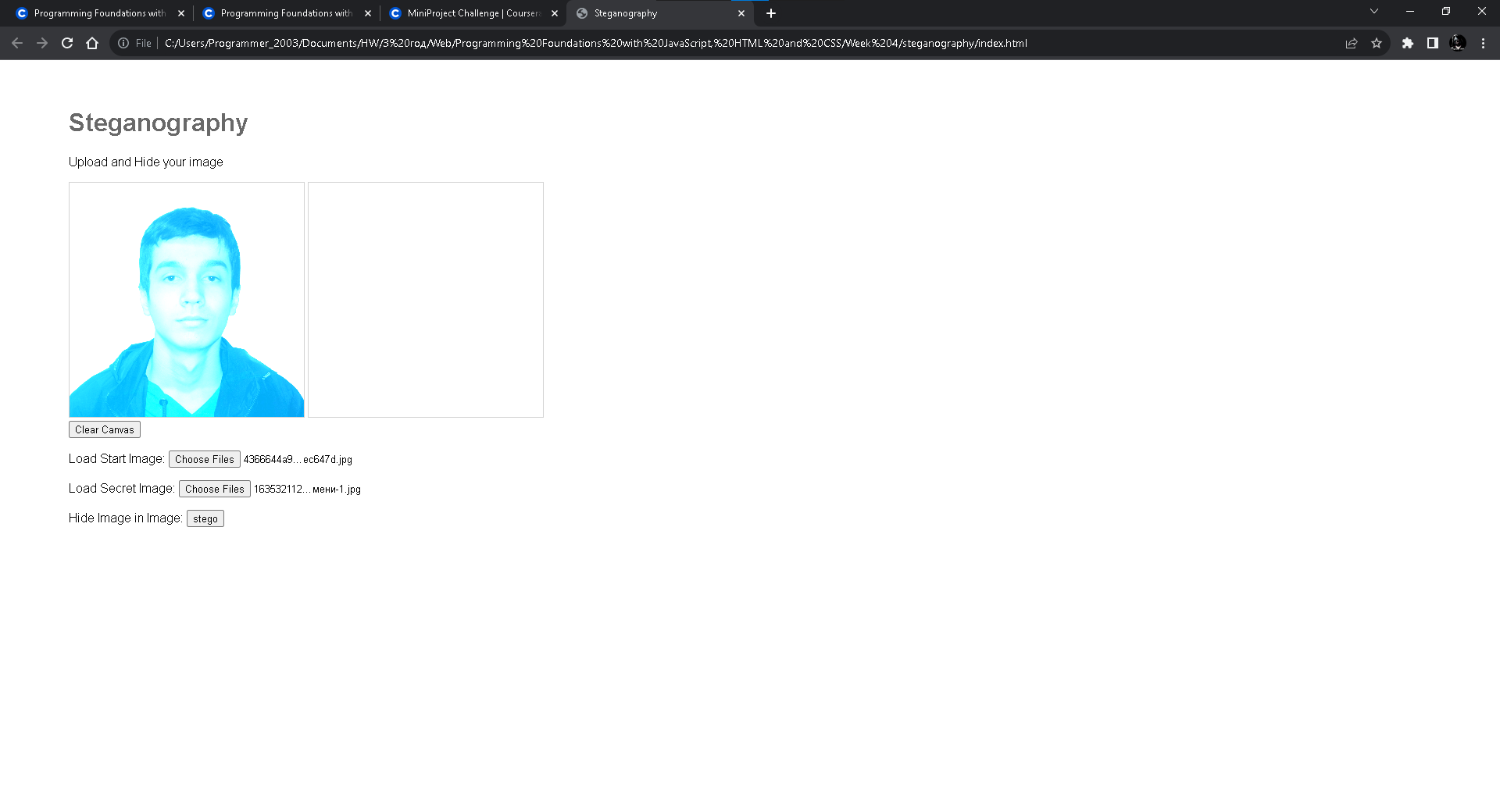
Clear Canvas:



Local Start Image & Local Secret Image:



Hide Local Image(stego):



**index.html:**

<!DOCTYPE html>

<html lang="en" >

<head>

<meta charset="UTF-8">

<title>Steganography</title>

<link rel="stylesheet" href="./style.css">

</head>

<body>

<!-- partial:index.partial.html -->

<script src="https://www.dukelearntoprogram.com/course1/common/js/image/SimpleImage.js"></script>

<div class="body">

<h1>Steganography</h1>

<p>Upload and Hide your image</p>

<div class="canvas">

<canvas id="canvas-a"></canvas>

<canvas id="canvas-b"></canvas>

</div>

<div class="bar">

<input type="button" value="Clear Canvas" id="backtoOriginal" onclick="clearCanvas()">

</div>

<div class="image">

<p>Load Start Image:

<input type="file" multiple="false" accept="image/\*" id="start-image" onchange="loadStartImage()">

</p>

<p>Load Secret Image:

<input type="file" multiple="false" accept="image/\*" id="secret-image" onchange="loadSecretImage()">

</p>

</div>

<div class="stego">

<p>Hide Image in Image:

<input type="button" value="stego" id="stego" onclick="stego()">

</p>

</div>

</div>

<!-- partial -->

<script src="./script.js"></script>

</body>

</html>

**style.css:**

body {

font-family: arial;

}

.body {

padding: 2em 5em;

}

h1 {

color: dimgray;

}

canvas {

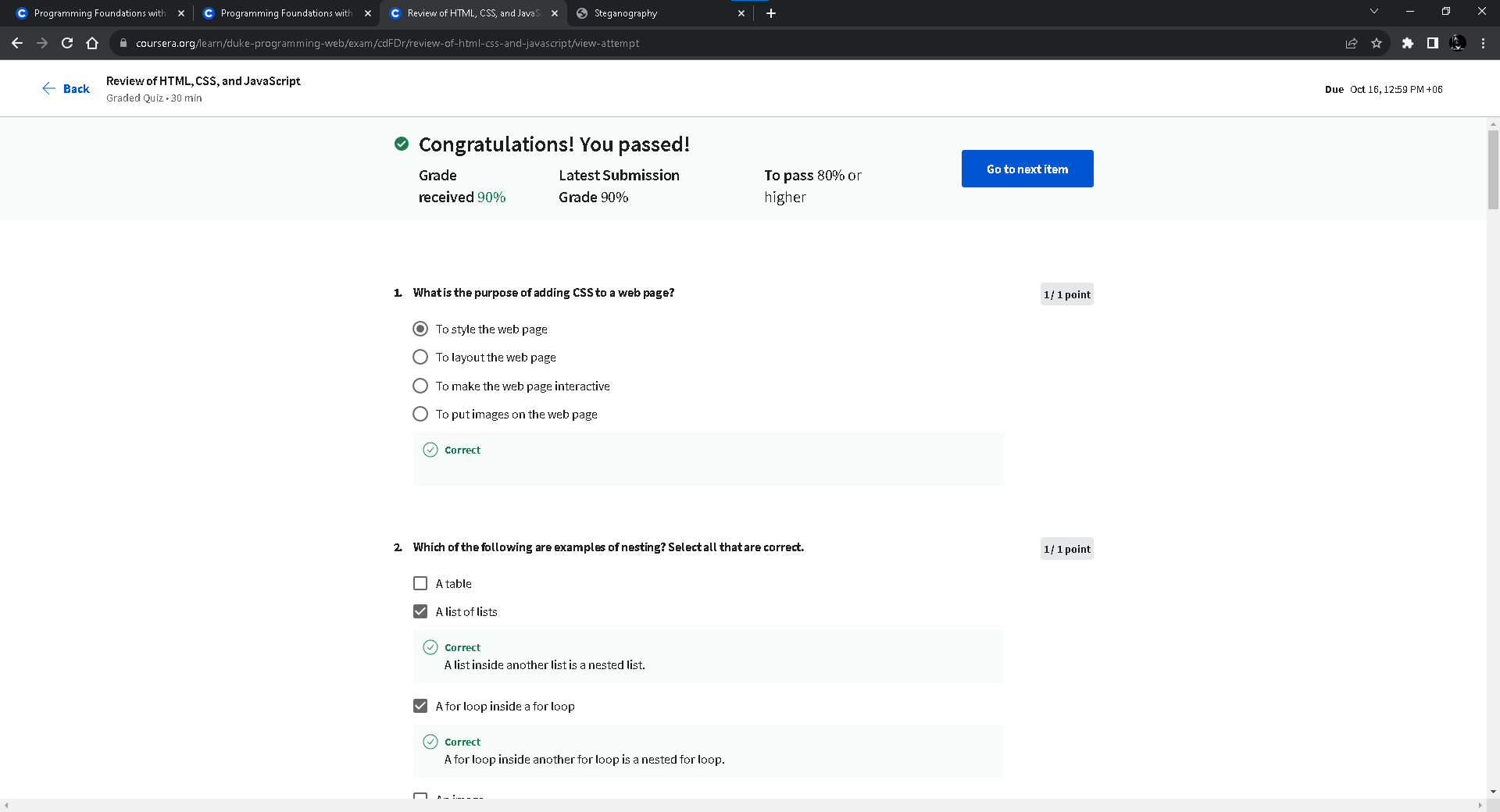
border: 1px solid lightgray;

width: 300px;

}

Quizzes:

**Review of HTML, CSS, and JavaScript:**

****

# Steganography:

# 

# Certificate:

# 