/\*

resources used for this are

http://stackoverflow.com/

http://api.jquery.com/

https://www.w3schools.com/

All resources were used as general references for this project

\*/

var bookCoverImage;

$(document).ready(function () {

$("#add-book").on("click", function (e) {

e.preventDefault();

addBook();

displayMessage("Book added to storage");

clearfields();

});

$(".clear-list").on("click", function (e) {

e.preventDefault();

localStorage.clear();

window.location.reload();

});

//drop zone for image If the element doesn't exist (as in the second page, don't add a listner)

var dz = document.getElementById('book-cover');

if (dz != null) {

dz.addEventListener('dragover', function (e) {

e.preventDefault();

e.dataTransfer.dropEffect = 'copy';

}, false);

dz.addEventListener('drop', dropFile, false);

}

$("#delete-by-name").on("click", removeBookByName);

});

//book object

function Book(name, author, isbn, publisher, price, image) {

this.name = name;

this.author = author;

this.isbn = isbn;

this.publisher = publisher;

this.price = price;

this.image = image;

}

function addBook() {

var name = $("#book-name").val();

var author = $("#author").val();

var isbn = $("#isbn").val();

var publisher = $("#publisher").val();

var price = $("#price").val();

var image = "";

//null check in case the book cover is not added

if (bookCoverImage != null) {

image = bookCoverImage.src; //src

}

var book = new Book(name, author, isbn, publisher, price, image);

localStorage.setItem(isbn, JSON.stringify(book));

}

/\*

Remove the field contents, clear the image and replace the text for dropping image

\*/

function clearfields() {

$("#book-name, #author, #isbn, #publisher, #price").val("");

$("#book-cover").children("img:first").remove();

$("#book-cover").append("<p class='text-center'>Drop book cover here</p>");

}

//simple display message that flashes in and out

function displayMessage(message) {

$("#info-message").html(message).fadeIn("slow").delay(1000).fadeOut("slow");

}

function dropFile(e) {

e.preventDefault();

var file = e.dataTransfer.files[0];

getSrcFromImage(file);

}

function getSrcFromImage(file) {

var reader = new FileReader();

reader.onload = function () {

bookCoverImage = new Image();

bookCoverImage.src = reader.result;

var bookImage = document.getElementById('book-cover');

bookImage.removeChild($("#book-cover > p").get(0));

bookImage.appendChild(bookCoverImage);

};

reader.onerror = function () {

alert("There was an error reading the file");

};

reader.readAsDataURL(file);

}

/\*

Clone the div for display. remove the clone class, add contents and insert into the dom

\*/

function displayBooks() {

//loop through each book and add them to the element

for (var i = 0; i < localStorage.length; i++) {

//clone elements

var idCounter = "book-" + i;

var element = $(".book-record").clone().removeAttr("style").attr("id", idCounter).appendTo(".book-display");

// remove the class from the element to prevent double records getting created

element.removeClass("book-record");

var key = localStorage.key(i);

var book = JSON.parse(localStorage.getItem(key));

var infoBlock = $("#" + idCounter).find('#book-info');

//create list of details for book information

var bookKeys = ["name", "author", "isbn", "publisher", "price"];

$.each(bookKeys, function (index, value) {

var content = value + " : " + book[value];

$(infoBlock).append("<li>" + content + "</li>");

});

if (book.image != "") {

// only add image if the source is not null

$("#" + idCounter).find('#book-image').attr("src", book.image);

}

//add key to the button so it can identify the book and get the isbn

$("#" + idCounter).find('#delete').attr("rel", book.isbn).attr("pos", i);

}

}

/\*

remove the item from local storage and hide the div in the dom

\*/

function removeBook(e) {

var isbn = $(e).attr("rel");

var counter = $(e).attr("pos");

localStorage.removeItem(isbn);

$("#book-" + counter).fadeOut("slow");

}

/\*

remove book from local storage by name

\*/

function removeBookByName() {

var bookName = $("#book-delete-title").val();

for (var i = 0; i < localStorage.length; i++) {

//search through local storage an see if there's a match for name

//if there is, remove it from local storage and reload the page

var key = localStorage.key(i);

var book = JSON.parse(localStorage.getItem(key));

if (bookName == book.name) {

localStorage.removeItem(key);

window.location.reload();

}

}

}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>BookStore Display</title>

<link rel="stylesheet" type="text/css" href="../styles/bootstrap.min.css">

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

</head>

<body onload="displayBooks()">

<div class="row main">

<div class="col-md-8 book-display">

</div>

<div class="col-md-4">

<div class="form-group">

<label for="book-delete-title">Book Name</label>

<div class="input-group">

<input type="text" class="form-control" id="book-delete-title" placeholder="Name">

<div class="input-group-addon">

<span class="glyphicon glyphicon-trash" aria-hidden="true" id="delete-by-name"></span>

</div>

</div>

<span class="help-block">Enter the name of the book you wish to delete from local storage.</span>

</div>

<button class="btn btn-warning clear-list">Clear List</button>

</div>

</div>

</div>

<div class="row main book-record">

<div class="panel panel-info">

<div class="panel-heading">

<h3 class="panel-title">Book Information<span class="glyphicon glyphicon-trash" aria-hidden="true"

onclick="removeBook(this)" id="delete"></span>

</h3>

</div>

<div class="panel-body">

<div class="row">

<div class="col-md-6" id="book-details">

<div>

<ul id="book-info">

</ul>

</div>

</div>

<div class="col-md-6">

<img src="../images/not-found.png" width="200px" height="200px" id="book-image"/>

</div>

</div>

</div>

</div>

</div>

<div class="row">

<div class="col-md-12">

<h3 class="bg-success" id="info-message"></h3>

</div>

</div>

<script src="../js/jquery-3.1.1.min.js"></script>

<script src="q1.js"></script>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>BookStore</title>

<link rel="stylesheet" type="text/css" href="../styles/bootstrap.min.css">

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

</head>

<body>

<div class="row main-body">

<div class="panel panel-info">

<div class="panel-heading">

<h3 class="panel-title">Book Information</h3>

</div>

<div class="panel-body">

<form>

<div class="row">

<div class="col-md-8">

<div class="form-group">

<label for="book-name">Book Name</label>

<input type="text" class="form-control" id="book-name">

</div>

<div class="form-group">

<label for="author">Author</label>

<input type="text" class="form-control" id="author">

</div>

<div class="form-group">

<label for="isbn">ISBN</label>

<input type="text" class="form-control" id="isbn">

</div>

</div>

<div class="col-md-4 upload-image" id="book-cover">

<p class="text-center">Drop book cover here, preferably in a 200 x 200 size</p>

</div>

</div>

<div class="form-group">

<label for="publisher">Publisher</label>

<input type="text" class="form-control" id="publisher">

</div>

<div class="form-group">

<label for="price">Price</label>

<input type="text" class="form-control" id="price" placeholder="€">

</div>

<div class="row">

<div class="col-md-4 left">

<a href="bookRetrieve.html" class="btn btn-primary" role="button">View Books</a>

</div>

<div class="col-md-4 right col-md-offset-4">

<button class="btn btn-success" id="add-book">Add Book</button>

</div>

</div>

<div class="row">

<div class="col-md-12">

<h3 class="bg-success" id="info-message"></h3>

</div>

</div>

</form>

</div>

</div>

</div>

<script src="../js/jquery-3.1.1.min.js"></script>

<script src="q1.js"></script>

</body>

</html>

.main-body {

width: 50%;

margin: 20px auto;

}

.main{

margin: 20px;

}

.right {

text-align: right;

}

.left {

text-align: left;

float: left;

}

.center {

text-align: center;

}

.upload img {

position: absolute;

top: 80%;

left: 50%;

margin-right: -50%;

transform: translate(-50%, -50%)

}

.upload-image {

border: 1px dashed lightgray;

height: 210px;

overflow: hidden;

}

.text-center{

color: lightgray;

}

#info-message{

display:none;

}

.glyphicon{

float: right;

}

#delete{

width:350px;

height:350px;

border:1px dashed black;

}

#delete > span {

margin-top: 80px;

color: #777777;

padding-right: 60px;

}

#body{

margin: 10px;

}

/\*

The following resources were used in this project

http://stackoverflow.com/questions/21065458/trying-to-make-2-happy-sad-faces-with-canvas

https://developer.mozilla.org/en-US/

https://www.w3schools.com/

W3Schools and MDN were both used as a general reference for dealing with canvas and drawing

\*/

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

if (canvas != null) {

var ctx = canvas.getContext("2d");

}

$(document).ready(function () {

//adding event listeners

$("#next-page").on("click", function (e) {

e.preventDefault();

saveToStorage();

});

$("#draw").on("click", function (e) {

e.preventDefault();

drawShape();

});

$("delete").on("drop", function () {

});

});

// saving local storage

function saveToStorage() {

var type = $("#type").val();

var wink = $("#wink").val();

var background = $("#background").val();

var lineColor = $("#line-colour").val();

var lineWidth = $("#line-width").val();

var obj = {

"type": type,

"wink": wink,

"background": background,

"lineColor": lineColor,

"lineWidth": lineWidth

};

localStorage.setItem("shape", JSON.stringify(obj));

window.location.href = "Render.html";

}

function drawShape() {

var data = JSON.parse(localStorage.getItem("shape"));

var x = canvas.width / 2;

var y = canvas.height / 2;

//set the radius of the circle to the average of the width and height of the canvas

// - 10 so to fill the canvas element almost

var radius = ((x + y) / 2) - 10;

var startAngle = 0;

var endAngle = 2 \* Math.PI;

//set colours

ctx.strokeStyle = data.lineColor;

ctx.fillStyle = data.background;

//draw circle for face

drawFace(data, x, y, radius, startAngle, endAngle);

drawEyes(data, startAngle, endAngle, data.wink);

drawSmile(data.type, x, y, startAngle, radius);

}

function drawSmile(type, x, y, startAngle, radius) {

var angle = Math.PI;

var antiCloclwise = false;

if (type == "sad") {

// reduce the radius of the arc to fit inside the circle

// move the starting point of the circle further down

radius -= 40;

y += radius;

//invert the arc to draw upwards, instead of down

antiCloclwise = true;

}

else if (type == "cheeky") {

// reduce the angle by half to draw a quarter circle instead of a semi circle

angle = Math.PI / 2;

}

ctx.moveTo(110, 75);

ctx.beginPath();

ctx.arc(x, y, radius - 50, startAngle, angle, antiCloclwise);

ctx.stroke();

}

function drawEyes(data, startAngle, endAngle, wink) {

var centerY = canvas.height / 3;

// x position of left eye is 1 third of the canvas

var leftX = canvas.width / 3;

// x position of the right eye is 2 thirds of the canvas

var rightX = (canvas.height / 3) \* 2;

var radius = 10;

ctx.fillStyle = data.lineColor;

ctx.beginPath();

//left eye

ctx.arc(leftX, centerY, radius, startAngle, endAngle);

ctx.stroke();

ctx.fill();

//right eye

ctx.moveTo(rightX, centerY);

if (wink == "yes") {

//shrink the diameter for a wink & add the radius to the

//center to draw the semi circle instead of the circle

ctx.lineWidth = 8;

ctx.moveTo(rightX + radius, centerY);

ctx.arc(rightX, centerY, radius, startAngle, Math.PI);

ctx.stroke();

ctx.lineWidth = getLineWidth(data.lineWidth);

}

else {

ctx.arc(rightX, centerY, radius, startAngle, endAngle);

ctx.stroke();

ctx.fill();

}

}

//draws the initial face

function drawFace(data, x, y, radius, startAngle, endAngle) {

ctx.beginPath();

ctx.arc(x, y, radius, startAngle, endAngle);

ctx.lineWidth = getLineWidth(data.lineWidth);

ctx.stroke();

ctx.fill();

}

//returns an integer value to convert text size into digits.

function getLineWidth(size) {

var rVal;

switch (size) {

case "small":

rVal = 1;

break;

case "large":

rVal = 15;

break;

default:

rVal = 1;

break;

}

return rVal;

}

//preventing default action on drop

function allowDrop(e) {

e.preventDefault();

}

// clearing local storage, removing the canvas and playing the sound

function drop(e) {

var audio = new Audio('deleted.mp3');

audio.play();

$(".removable").fadeOut("slow", function(){

$(this).remove();

});

localStorage.clear();

}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<link rel="stylesheet" type="text/css" href="../styles/bootstrap.min.css">

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

</head>

<body id="body">

<form>

<button class="btn btn-primary" id="draw">Draw</button>

</form>

<div class="row">

<div class="col-md-6">

<canvas id="canvas" width=350 height=350 draggable="true" style="border: 1px solid" class="removable"></canvas>

</div>

<div class="col-md-6">

<div id="delete" class="removable" ondrop="drop(event)" ondragover="allowDrop(event)">

<span class="glyphicon glyphicon-trash" aria-hidden="true" style="font-size: 15em"></span>

</div>

</div>

</div>

<script src="../js/jquery-3.1.1.min.js"></script>

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

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

<link rel="stylesheet" type="text/css" href="../styles/bootstrap.min.css">

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

</head>

<body>

<div class="row main-body">

<div class="panel panel-info">

<div class="panel-heading">

<h3 class="panel-title">Shape 'n Stuff</h3>

</div>

<div class="panel-body">

<form>

<div class="row">

<div class="col-md-12">

<div class="form-group">

<label for="type">Face Type</label>

<select class="form-control" id="type">

<option></option>

<option value="happy">Happy</option>

<option value="sad">Sad</option>

<option value="cheeky">Cheeky</option>

</select>

</div>

<div class="form-group">

<label for="wink">Wink</label>

<select class="form-control" id="wink">

<option></option>

<option value="yes">Yes</option>

<option value="no">No</option>

</select>

</div>

<div class="form-group">

<label for="line-width">Line Width</label>

<select class="form-control" id="line-width">

<option></option>

<option value="large">Large</option>

<option value="small">Small</option>

</select>

</div>

<div class="form-group">

<label for="background">Colour</label>

<input type="color" class="form-control" id="background">

</div>

<div class="form-group">

<label for="line-colour">Line Colour</label>

<input type="color" class="form-control" id="line-colour" value="#ffffff">

</div>

</div>

</div>

<div class="row">

<div class="col-md-4 right col-md-offset-8">

<a href="Render.html" class="btn btn-primary" role="button" id="next-page">Go</a>

</div>

</div>

<div class="row">

<div class="col-md-12">

<h3 class="bg-success" id="info-message"></h3>

</div>

</div>

</form>

</div>

</div>

</div>

<script src="../js/jquery-3.1.1.min.js"></script>

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

</body>

</html>

<!--CSS absolute positioning-->

<!--http://stackoverflow.com/questions/6802956/how-to-position-a-div-in-a-specific-coordinates-->

<!--Jquey animate-->

<!--http://api.jquery.com/animate/-->

<!--jQUery Rotate:-->

<!--http://jqueryrotate.com/-->

<!--This is not necessarily a unique animation in jquery. In fact, I'm pretty sure its not. However, what I've tried to do is give-->

<!--the animation a random time, easing and rotation for the image to go to, so every time the animation is called it will do a different animation-->

<!--for a different period of time with a different rotation. I also change the image randomly as the animation loops to roate to a different image as it goes around

I also made use of jqueryUI to give more easing effects for animation.

I've cleared the screen after 50 images are added to avoid chewing up too much processing power from the browser.

-->

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Q3</title>

</head>

<script src="../js/jquery-3.1.1.min.js"></script>

<script src="../js/jquery-ui.min.js"></script>

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

<script>

var screenWidth, screenHeight, count = 0, angle = 0;

var animationTypes = ["linear", "swing", "easeOutBounce", "easeInBounce", "easeInOutBack", "easeOutExpo"];

$(document).ready(function () {

//subtracting 50 from the width and height to prevent the image going out of the bounds of the screen

screenWidth = $(window).width() - 50;

screenHeight = $(window).height() - 100;

animate(2000, 0, getAnimation());

//infintely call rotate on all images except the main one

setInterval(function () {

angle += 10;

$(".image:not(#circle)").rotate(angle);

}, 50);

});

function animate(time, easing) {

var randomX = Math.floor(Math.random() \* screenWidth);

var randomY = Math.floor(Math.random() \* screenHeight);

$("#circle").animate({

"left": randomX + "px",

"top": randomY + "px"

}, {

duration: time,

step: function () {

// $(this).css('-webkit-transform','rotate('+rotation+'deg)')

},

easing: easing,

complete: function () {

//using ceiling to round up as there is no image with a 0 name

animate(Math.floor(Math.random() \* 3000), getAnimation());

$("#icon").attr("src", "emoji/" + Math.ceil(Math.random() \* 35) + ".svg");

}

});

$('body').append($("#circle").clone().removeAttr("id"));

count++;

if (count == 50) {

$(".image:not(#circle)").fadeOut("slow", function () {

$(this).remove();

count = 0;

})

}

}

function getAnimation() {

var x = Math.floor(Math.random() \* animationTypes.length) - 1;

return animationTypes[x];

}

</script>

<style>

.image {

width: 100px;

height: 100px;

position: absolute;

}

</style>

<body>

<div class="image" id="circle"><img id="icon" src="emoji/1.svg"/></div>

</body>

</html>