**Project**

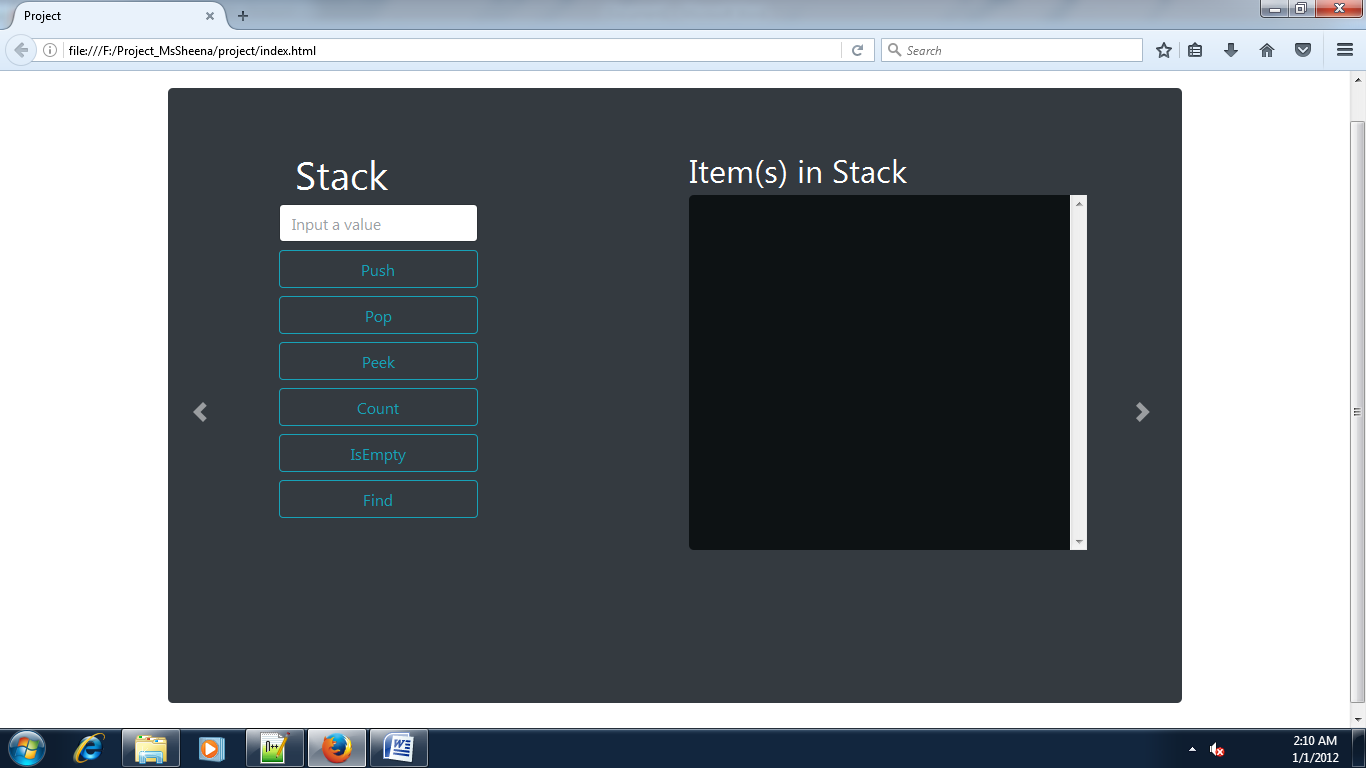
**In**

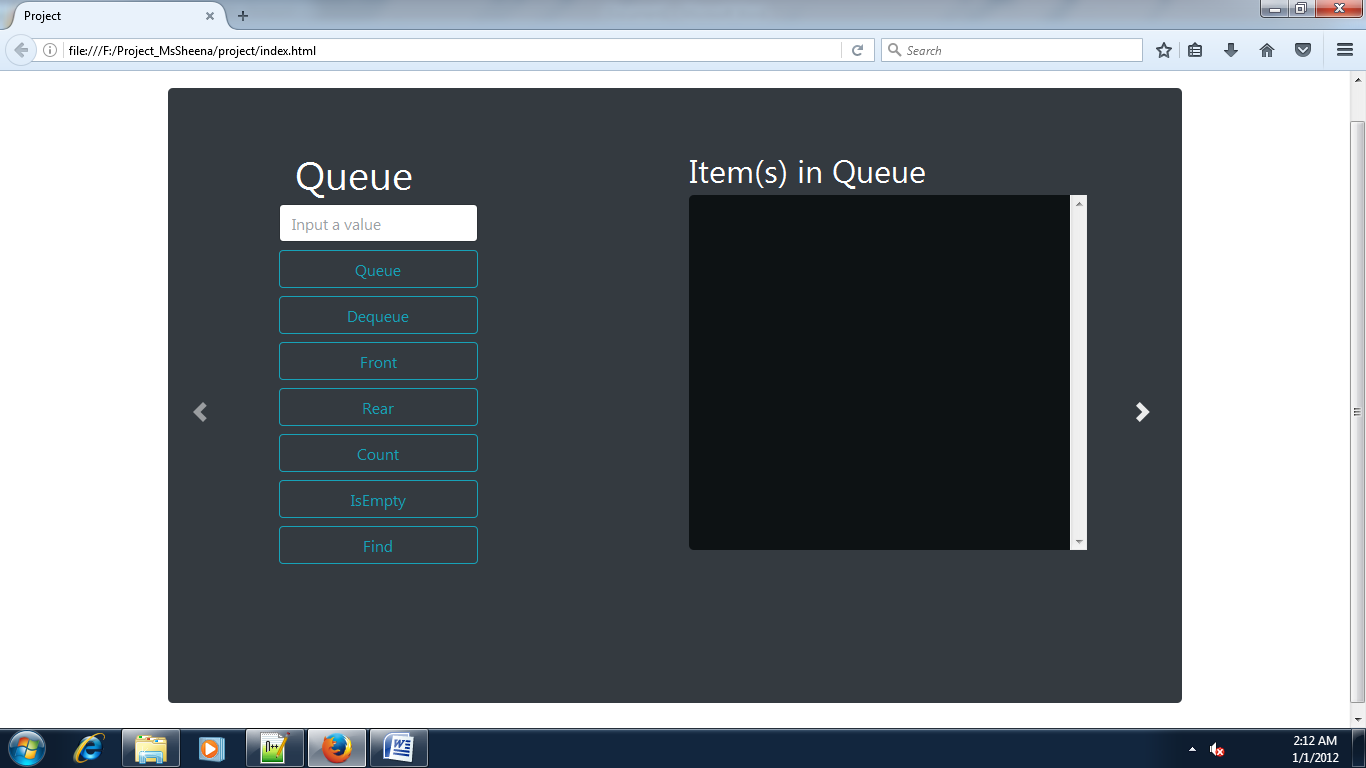
**IT410**

**Submitted by: Villaluna, Vincent A.**

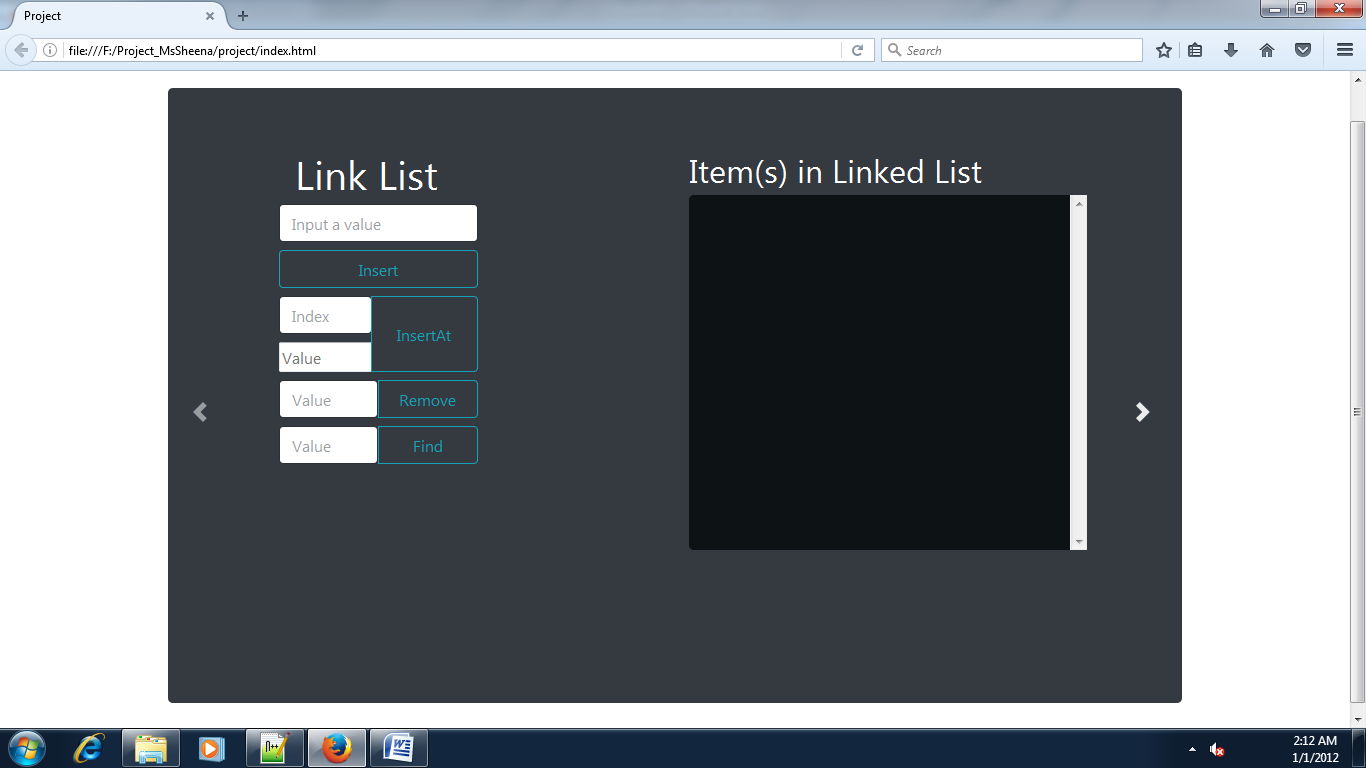
**Teacher: Ms. Maria Sheena Venanzuela**

**Stack**

**Queue**

****

**Link List**

****

**Source Code**

**HTML**

<!DOCTYPE html>

<html lang="en">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<!-- Bootstrap CSS -->

<link rel="stylesheet" href="bootstrap/css/bootstrap.min.css" crossorigin="anonymous">

<link rel="stylesheet" href="style.css" crossorigin="anonymous">

<Title>Project</Title>

</head>

<body>

<div class="container">

<!-- Carousel -->

<div id="carouselExampleControls" data-interval="false" class="carousel slide" data-ride="carousel">

<div class="carousel-inner">

<div class="carousel-item active">

<!--here is Stack-->

<div class="jumbotron fix-height mr-5 ml-5 bg-dark">

<div class="row">

<div class="col ml-3">

<div class="container ml-5">

<h1 class="header"><span class="badge-dark rad bg-dark pl-3 pr-3">Stack</span></h1>

<input type="text" class="form-control w-50 mb-2" autocomplete="off" placeholder="Input a value" id="stack-value">

<input class="btn btn-outline-info form-control w-50 mb-2" type="submit" value="Push" id="stack-push" onclick="stkPush()">

<input class="btn btn-outline-info form-control w-50 mb-2" type="submit" value="Pop" id="stack-pop" onclick="stkPop()">

<input class="btn btn-outline-info form-control w-50 mb-2" type="submit" value="Peek" id="stack-front" onclick="stkPeek()">

<!-- <input class="btn btn-outline-info form-control w-50 mb-2" type="submit" value="Back" id="stack-back" onclick="stkBack()"> -->

<input class="btn btn-outline-info form-control w-50 mb-2" type="submit" value="Count" id="stack-count" onclick="stkCount()">

<input class="btn btn-outline-info form-control w-50 mb-2" type="submit" value="IsEmpty" id="stack-isEmpty" onclick="stkIsEmpty()">

<input class="btn btn-outline-info form-control w-50" type="submit" value="Find" id="stack-Find" onclick="stkFind()">

</div>

</div>

<div class="col mr-5">

<div class="container">

<h2>Item(s) in Stack</h2>

<div class="result-container">

<div class="jumbotron item-list" id="stacked">

</div>

</div>

<div class="alert" id="stack-alert" role="alert">

</div>

</div>

</div>

</div>

</div>

<!--here is Stack-->

</div>

<div class="carousel-item">

<!--Here is Queue-->

<div class="jumbotron fix-height mr-5 ml-5 bg-dark ">

<div class="row">

<div class="col ml-3">

<div class="container ml-5">

<h1 class="header">

<span class="badge-dark rad bg-dark pl-3 pr-3">Queue</span>

</h1>

<div class="form-group" action="#">

<input type="text" class="form-control w-50 mb-2" placeholder="Input a value" autocomplete="off" id="queue-value">

<input class="btn btn-outline-info form-control w-50 mb-2" type="submit" value="Queue" onclick="QuePush()">

<input class="btn btn-outline-info form-control w-50 mb-2" type="submit" value="Dequeue" onclick="QuePop()">

<input class="btn btn-outline-info form-control w-50 mb-2" type="submit" value="Front" id="stack-front" onclick="queFront()">

<input class="btn btn-outline-info form-control w-50 mb-2" type="submit" value="Rear" id="stack-back" onclick="queBack()">

<input class="btn btn-outline-info form-control w-50 mb-2" type="submit" value="Count" id="stack-count" onclick="queCount()">

<input class="btn btn-outline-info form-control w-50 mb-2" type="submit" value="IsEmpty" id="stack-isEmpty" onclick="queIsEmpty()">

<input class="btn btn-outline-info form-control w-50" type="submit" value="Find" id="que-Find" onclick="queFind()">

</div>

</div>

</div>

<div class="col mr-5">

<div class="container">

<h2>Item(s) in Queue</h2>

<div class="result-container">

<div class="jumbotron item-list" id="queued">

</div>

</div>

<div class="alert" id="queue-alert" role="alert">

</div>

</div>

</div>

</div>

</div>

<!--Here is Queue-->

</div>

<div class="carousel-item">

<!--Here is Link List-->

<div class="jumbotron fix-height mr-5 ml-5 bg-dark ">

<div class="row">

<div class="col ml-3">

<div class="container ml-5">

<h1 class="header"><span class="badge-dark rad bg-dark pl-3 pr-3">Link List</span></h1>

<div class="form-group">

<!--Insert-->

<input type="text" class="form-control w-50 mb-2" autocomplete="off" placeholder="Input a value" id="linklist-value">

<input class="btn btn-outline-info form-control w-50 mb-2" type="submit" value="Insert" onclick="listpush()">

<!--Insert Index At-->

<div class="btn-group mb-2 w-50" role="group">

<div class="btn-group btn-group-vertical" role="group">

<input type="text" class="form-control mb-2 w-100" autocomplete="off" placeholder="Index" id="linklist-index-value">

<input type="text" class="form-contro w-100" autocomplete="off" placeholder="Value" id="linklist-value-at">

</div>

<input class="btn btn-outline-info form-control" type="submit" value="InsertAt" onclick="insertIndexAt()">

</div>

<!--Remove-->

<div class="btn-group mb-2 w-50" role="group">

<div class="btn-group btn-group-inline" role="group">

<input type="text" class="form-control w-50" autocomplete="off" placeholder="Value" id="linklist-remove">

<input class="btn btn-outline-info form-control" type="submit" value="Remove" onclick="remove()">

</div>

</div>

<!--Find-->

<div class="btn-group mb-2 w-50">

<div class="btn-group form-inline" role="group">

<input type="text" class="form-control w-50" autocomplete="off" placeholder="Value" id="linklist-find">

<input class="btn btn-outline-info form-control w-50" type="submit" value="Find" onclick="find()">

</div>

</div>

</div>

</div>

</div>

<div class="col mr-5">

<div class="container">

<h2>Item(s) in Linked List</h2>

<div class="result-container">

<div class="jumbotron item-list" id="linkedList">

</div>

</div>

<div class="alert" id="link-alert" role="alert">

</div>

</div>

</div>

</div>

</div>

<!--Here is Link List-->

</div>

</div>

<a class="carousel-control-prev" href="#carouselExampleControls" role="button" data-slide="prev">

<span class="carousel-control-prev-icon" aria-hidden="true"></span>

<span class="sr-only">Previous</span>

</a>

<a class="carousel-control-next" href="#carouselExampleControls" role="button" data-slide="next">

<span class="carousel-control-next-icon" aria-hidden="true"></span>

<span class="sr-only">Next</span>

</a>

</div>

<!-- Carousel -->

</div>

</body>

<script src="bootstrap/js/jquery-3.2.1.slim.min.js"></script>

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

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

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

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

</html>

**Stack.js**

var stacks = [];

var reverse = 0;

function stkPush() {

var c = 0;

if (document.getElementById('stack-value').value != "") {

$('#stack-alert').hide();

c = 0;

var values = "";

stacks.push(document.getElementById('stack-value').value);

reverse = stacks.length - 1;

while (reverse != -1) {

values = values + 'index[' + c + '] : ' + stacks[reverse] + '<br>';

c++;

reverse--;

}

document.getElementById('stacked').innerHTML = values;

} else {

document.getElementById("stack-alert").classList.add('alert-danger');

document.getElementById("stack-alert").innerHTML = "Can't Push with empty boxes!";

$('#stack-alert').show();

// document.getElementById("stack-alert").classList.remove('alert-danger'); used to removed class

}

document.getElementById('stack-value').value = "";

}

function stkPop() {

if (stacks.length != 0) {

$('#stack-alert').hide();

var c = 0;

var values = "";

stacks.pop();

reverse = stacks.length - 1;

c = 0;

while (reverse != -1) {

values = values + 'index[' + c + '] : ' + stacks[reverse] + '<br>';

c++;

reverse--;

}

document.getElementById('stacked').innerHTML = values;

document.getElementById("stack-alert").classList.add('alert-info');

document.getElementById("stack-alert").classList.remove('alert-danger');

document.getElementById("stack-alert").innerHTML = "Successfully Pop!";

$('#stack-alert').show();

} else {

document.getElementById("stack-alert").classList.add('alert-danger');

document.getElementById("stack-alert").innerHTML = "Can't Pop with empty stack!";

$('#stack-alert').show();

}

}

function stkPeek() {

if (stacks.length != 0) {

document.getElementById("stack-alert").classList.remove('alert-danger');

document.getElementById("stack-alert").classList.add('alert-info');

document.getElementById("stack-alert").innerHTML = 'Front is : ' + stacks[stacks.length - 1];

$('#stack-alert').show();

} else {

document.getElementById("stack-alert").classList.remove('alert-info');

document.getElementById("stack-alert").classList.add('alert-danger');

document.getElementById("stack-alert").innerHTML = "Error no items on stack";

$('#stack-alert').show();

}

}

// Function here is no longer used because we comment the html element intput back

function stkBack() {

if (stacks.length != 0) {

document.getElementById("stack-alert").classList.remove('alert-danger');

document.getElementById("stack-alert").classList.add('alert-info');

document.getElementById("stack-alert").innerHTML = 'Back is : ' + stacks[0];

$('#stack-alert').show();

} else {

document.getElementById("stack-alert").classList.remove('alert-info');

document.getElementById("stack-alert").classList.add('alert-danger');

document.getElementById("stack-alert").innerHTML = "Error no items on stack";

$('#stack-alert').show();

}

}

function stkCount() {

if (stacks.length != 0) {

document.getElementById("stack-alert").classList.remove('alert-danger');

document.getElementById("stack-alert").classList.add('alert-info');

document.getElementById("stack-alert").innerHTML = stacks.length;

$('#stack-alert').show();

} else {

document.getElementById("stack-alert").classList.remove('alert-info');

document.getElementById("stack-alert").classList.add('alert-danger');

document.getElementById("stack-alert").innerHTML = 'Nothing to count no stacks';

$('#stack-alert').show();

}

}

function stkIsEmpty() {

if (stacks.length == 0) {

document.getElementById("stack-alert").classList.add('alert-info');

document.getElementById("stack-alert").innerHTML = 'Yes';

$('#stack-alert').show();

} else {

document.getElementById("stack-alert").classList.add('alert-info');

document.getElementById("stack-alert").innerHTML = 'No';

$('#stack-alert').show();

}

}

function stkFind() {

var found = false;

var indx = [];

var count = -1;

if (document.getElementById('stack-value').value != "") {

if (stacks.length != 0) {

var mask = stacks.length;

for (var c = 0; c < stacks.length; c++) {

mask--;

if (stacks[c] == document.getElementById('stack-value').value) {

count++;

indx[count] = 'index[' + mask + '] = ' + stacks[c];

found = true;

}

}

if (found == false) {

document.getElementById("stack-alert").classList.remove('alert-danger');

document.getElementById("stack-alert").classList.add('alert-info');

document.getElementById("stack-alert").innerHTML = 'sorry cannot find!';

$('#stack-alert').show();

} else {

document.getElementById("stack-alert").classList.remove('alert-danger');

document.getElementById("stack-alert").classList.add('alert-info');

document.getElementById("stack-alert").innerHTML = indx;

$('#stack-alert').show();

}

} else {

document.getElementById("stack-alert").classList.remove('alert-info');

document.getElementById("stack-alert").classList.add('alert-danger');

document.getElementById("stack-alert").innerHTML = 'Error no items on stack';

$('#stack-alert').show();

}

} else {

document.getElementById("stack-alert").classList.remove('alert-info');

document.getElementById("stack-alert").classList.add('alert-danger');

document.getElementById("stack-alert").innerHTML = "Can't Find with empty boxes!";

$('#stack-alert').show();

}

document.getElementById('stack-value').value = "";

}

**Queue.js**

var Queues = [];

function QuePush() {

if (document.getElementById('queue-value').value != "") {

var values = "";

Queues.push(document.getElementById('queue-value').value);

for (var c = 0; c < Queues.length; c++) {

values = values + 'index[' + c + '] : ' + Queues[c] + '<br>';

}

document.getElementById('queued').innerHTML = values;

} else {

document.getElementById("queue-alert").classList.add('alert-danger');

document.getElementById("queue-alert").innerHTML = "Can't Push with empty boxes!";

$('#queue-alert').show();

}

document.getElementById('queue-value').value = "";

}

function QuePop() {

if (Queues.length != 0) {

var values = "";

Queues.shift();

for (var c = 0; c < Queues.length; c++) {

values = values + 'index[' + c + '] : ' + Queues[c] + '<br>';

}

document.getElementById('queued').innerHTML = values;

document.getElementById("queue-alert").classList.add('alert-info');

document.getElementById("queue-alert").classList.remove('alert-danger');

document.getElementById("queue-alert").innerHTML = "Successfully Dequeue!";

$('#queue-alert').show();

} else {

document.getElementById("queue-alert").classList.add('alert-danger');

document.getElementById("queue-alert").innerHTML = "Can't Dequeue with empty Queue!";

$('#queue-alert').show();

}

document.getElementById('queue-value').value = "";

}

function queFront() {

if (Queues != 0) {

document.getElementById("queue-alert").classList.remove('alert-danger');

document.getElementById("queue-alert").classList.add('alert-info');

document.getElementById("queue-alert").innerHTML = 'Front is : ' + Queues[0];

$('#queue-alert').show();

} else {

document.getElementById("queue-alert").classList.remove('alert-info');

document.getElementById("queue-alert").classList.add('alert-danger');

document.getElementById("queue-alert").innerHTML = "Error no items on Queue";

$('#queue-alert').show();

}

}

function queBack() {

if (Queues != 0) {

document.getElementById("queue-alert").classList.remove('alert-danger');

document.getElementById("queue-alert").classList.add('alert-info');

document.getElementById("queue-alert").innerHTML = 'Back is : ' + Queues[Queues.length - 1];

$('#queue-alert').show();

} else {

document.getElementById("queue-alert").classList.remove('alert-info');

document.getElementById("queue-alert").classList.add('alert-danger');

document.getElementById("queue-alert").innerHTML = "Error no items on Queue";

$('#queue-alert').show();

}

}

function queCount() {

if (Queues != 0) {

document.getElementById("queue-alert").classList.remove('alert-danger');

document.getElementById("queue-alert").classList.add('alert-info');

document.getElementById("queue-alert").innerHTML = Queues.length;

$('#queue-alert').show();

} else {

document.getElementById("queue-alert").classList.remove('alert-info');

document.getElementById("queue-alert").classList.add('alert-danger');

document.getElementById("queue-alert").innerHTML = 'Nothing to count no queues';

$('#queue-alert').show();

}

}

function queIsEmpty() {

if (Queues == 0) {

document.getElementById("queue-alert").classList.add('alert-info');

document.getElementById("queue-alert").innerHTML = 'Yes';

$('#queue-alert').show();

} else {

document.getElementById("queue-alert").classList.add('alert-info');

document.getElementById("queue-alert").innerHTML = 'No';

$('#queue-alert').show();

}

}

function queFind() {

var found = false;

var indx = [];

var count = -1;

if (document.getElementById('queue-value').value != "") {

if (Queues.length != 0) {

for (var c = 0; c < Queues.length; c++) {

if (Queues[c] == document.getElementById('queue-value').value) {

count++;

indx[count] = 'index[' + c + '] = ' + Queues[c];

found = true;

}

}

if (found == false) {

document.getElementById("queue-alert").classList.remove('alert-danger');

document.getElementById("queue-alert").classList.add('alert-info');

document.getElementById("queue-alert").innerHTML = 'sorry cannot find!';

$('#queue-alert').show();

} else {

document.getElementById("queue-alert").classList.remove('alert-danger');

document.getElementById("queue-alert").classList.add('alert-info');

document.getElementById("queue-alert").innerHTML = indx;

$('#queue-alert').show();

}

} else {

document.getElementById("queue-alert").classList.remove('alert-info');

document.getElementById("queue-alert").classList.add('alert-danger');

document.getElementById("queue-alert").innerHTML = 'Error no items on queue';

$('#queue-alert').show();

}

} else {

document.getElementById("queue-alert").classList.remove('alert-info');

document.getElementById("queue-alert").classList.add('alert-danger');

document.getElementById("queue-alert").innerHTML = "Can't Find with empty boxes!";

$('#queue-alert').show();

}

document.getElementById('queue-value').value = "";

}

**linkedlist.js**

var list = [];

var values = "";

function listpush() {

if (document.getElementById('linklist-value').value != "") {

$('#link-alert').hide();

list.push(document.getElementById('linklist-value').value);

values = "";

for (var c = 0; c < list.length; c++) {

values = values + 'index[' + c + '] : ' + list[c] + "<br>";

}

} else {

document.getElementById("link-alert").classList.remove('alert-info');

document.getElementById("link-alert").classList.add('alert-danger');

document.getElementById("link-alert").innerHTML = "Can't Push with empty boxes!";

$('#link-alert').show();

}

document.getElementById('linkedList').innerHTML = values;

document.getElementById('linklist-value').value = "";

}

function insertIndexAt() {

try {

if (document.getElementById('linklist-index-value').value != "" && document.getElementById('linklist-value-at').value != "") {

if (list.length != 0) {

values = "";

$('#link-alert').hide();

list.splice(document.getElementById('linklist-index-value').value, 0, document.getElementById('linklist-value-at').value);

for (var c = 0; c < list.length; c++) {

values = values + 'index[' + c + '] : ' + list[c] + '<br>';

}

document.getElementById('linkedList').innerHTML = values;

} else {

document.getElementById("link-alert").classList.remove('alert-info');

document.getElementById("link-alert").classList.add('alert-danger');

document.getElementById("link-alert").innerHTML = "Error no items on List";

$('#link-alert').show();

}

} else {

document.getElementById("link-alert").classList.remove('alert-info');

document.getElementById("link-alert").classList.add('alert-danger');

document.getElementById("link-alert").innerHTML = "Please fill in the index and value boxes!";

$('#link-alert').show();

}

document.getElementById('linklist-index-value').value = "";

document.getElementById('linklist-value-at').value = "";

} catch (err) {

document.getElementById("link-alert").classList.remove('alert-info');

document.getElementById("link-alert").classList.add('alert-danger');

document.getElementById("link-alert").innerHTML = err.message;

$('#link-alert').show();

}

}

function remove() {

try {

values = "";

if (document.getElementById('linklist-remove').value != "") {

if (list.length != 0) {

var x=list.indexOf(document.getElementById('linklist-remove').value);

list.splice(x, 1);

$('#link-alert').hide();

for (var c = 0; c < list.length; c++) {

values = values + 'index[' + c + '] : ' + list[c] + '<br>';

}

document.getElementById('linkedList').innerHTML = values;

document.getElementById("link-alert").classList.remove('alert-danger');

document.getElementById("link-alert").classList.add('alert-info');

document.getElementById("link-alert").innerHTML = "Successfully Removed!";

$('#link-alert').show();

} else {

document.getElementById("link-alert").classList.remove('alert-info');

document.getElementById("link-alert").classList.add('alert-danger');

document.getElementById("link-alert").innerHTML = "Error no items on List";

$('#link-alert').show();

}

} else {

document.getElementById("link-alert").classList.remove('alert-info');

document.getElementById("link-alert").classList.add('alert-danger');

document.getElementById("link-alert").innerHTML = "Can't remove with empty boxes!";

$('#link-alert').show();

}

} catch (err) {

document.getElementById("link-alert").classList.remove('alert-info');

document.getElementById("link-alert").classList.add('alert-danger');

document.getElementById("link-alert").innerHTML = err.message;

$('#link-alert').show();

}

document.getElementById('linklist-remove').value = "";

}

function find() {

var found = false;

var indx = [];

var count = -1;

if (document.getElementById('linklist-find').value != "") {

if (list.length != 0) {

for (var c = 0; c < list.length; c++) {

if (list[c] == document.getElementById('linklist-find').value) {

count++;

indx[count] = 'index[' + c + '] = ' + list[c];

found = true;

}

}

if (found == false) {

document.getElementById("link-alert").classList.remove('alert-danger');

document.getElementById("link-alert").classList.add('alert-info');

document.getElementById("link-alert").innerHTML = 'sorry cannot find!';

$('#queue-alert').show();

} else {

document.getElementById("link-alert").classList.remove('alert-danger');

document.getElementById("link-alert").classList.add('alert-info');

document.getElementById("link-alert").innerHTML = indx;

$('#link-alert').show();

}

} else {

document.getElementById("link-alert").classList.remove('alert-info');

document.getElementById("link-alert").classList.add('alert-danger');

document.getElementById("link-alert").innerHTML = 'Error no items on List';

$('#link-alert').show();

}

} else {

document.getElementById("link-alert").classList.remove('alert-info');

document.getElementById("link-alert").classList.add('alert-danger');

document.getElementById("link-alert").innerHTML = "Can't Find with empty boxes!";

$('#link-alert').show();

}

document.getElementById('linklist-find').value = "";

}