**Validator.js**

class Validator {

constructor() {

$('#error00'). hide ();

$('#error0'). hide ();

$('#error01'). hide ();

$('#error02'). hide ();

$('#error03'). hide ();

$('#error04'). hide ();

$('#error05'). hide ();

$('#error06'). hide ();

$('#error07'). hide ();

$('#error08'). hide ();

$('#error09'). hide ();

$('#error10'). hide ();

$('#error11'). hide ();

$('#error12'). hide ();

$('#error13'). hide ();

}

check(member) {

for (let i in member) {

if (member[i] === "") {

$('#error03'). show ()

return false

}

else {

$('#error03'). hide ()

return true

}

}

}

**//THIS IS FOR TO LET THE FIRST NAME IS NOT EMPTY**

firstname(member) {

let First\_name = $('#input-name'). val ();

if (First\_name == "") {

$('#error04'). show ();

return false;

}

else {

$('#error04'). hide ();

return true;

}

}

**//THIS IS FOR TO LET THE MEMBERSHIP NUMBER IS NOT EMPTY**

membershipnumber(member) {

let membershipnumber = $('#input-membership'). val ();

if (membershipnumber == "") {

$('#error0'). show ();

return false;

}

else {

$('#error0'). hide ();

return true;

}

}

**//THIS IS FOR TO LET THE JOIN DATE IS NOT OLDER THAN THE DATE OF BIRTH**

compare(member) {

var datebirth = $('#dob'). val (). substr (0,4)

var jDate = $('#jd'). val (). substr (0,4)

if (Number(datebirth)>Number(jDate)) {

$('#error13'). show ();

return false;

} else {

$('#error13'). hide ();

return true;

}

}

**//THIS IS FOR TO LET THE LAST NAME IS NOT EMPTY**

lastname(member) {

var Last\_name = $('#input-last'). val ();

if (Last\_name == "") {

$('#error05'). show ();

return false;

}

else {

$('#error05'). hide ();

return true;

}

}

**//THIS IS FOR TO LET THE BIRTH IS NOT EMPTY**

birth(member) {

if ($('#dob'). val () == "" || $('#dob'). val (). length!== 10) {

$('#error07'). show ();

return false;

}

else {

$('#error07'). hide ();

return true;

}

}

**//THIS IS FOR TO LET THE JOIN DATE IS NOT EMPTY**

joinDate(member) {

if ($('#jd'). val () === "" || $('#jd'). val (). length !== 10 ) {

$('#error06'). show ();

return false;

}

else {

$('#error06'). hide ();

return true;

}

}

**//THIS IS FOR TO LET THE ADDRESS IS NOT EMPTY**

address(member) {

var Address = [$('#address1'). val (), $('#address2'). val (), $('#address3'). val ()];

if (Address [0] == "") {

$('#error09'). show ();

return false;

}

else {

$('#error09'). hide ();

return true;

}

}

**//THIS IS FOR TO LET THE POSTCODE IS NOT LESS THAN 5**

postcode(member) {

if ($('#Postcode'). val () == "" || $('#Postcode'). val (). length !== 5) {

$('#error08'). show ();

return false;

} else {

$('#error08'). hide ();

return true

}

}

**//THIS IS FOR TO LET THE MEMBERSHIP NUMBER IS NOT EMPTY**

membership(member) {

var membership = $('#input-membership-type'). val ();

if (membership == "") {

$('#error10'). show ();

return false;

}

else {

$('#error10'). hide ();

return true;

}

}

**//THIS IS FOR TO LET THE SUBSCRIPTION DUE MONTH IS NOT EMPTY**

day(member) {

var day = $('#day'). val ();

if (day == "") {

$('#error11'). show ();

return false;

}

else {

$('#error11'). hide ();

return true;

}

}

**//THIS IS FOR TO LET THE SEX IS NOT EMPTY**

sex(member) {

var sex = $('#sex'). val ();

if (sex == "") {

$('#error12'). show ();

return false;

}

else {

$('#error12'). hide ();

return true;

}

}

**//THIS IS FOR TO LET THE MEMBERSHIP NUMBER IS MODULES 11**

checkValidate(member)

{

// modulus 11

let x = $('#input-membership'). val ();

let y = x.toString();

let z = 0;

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

let l = 6 - i;

z += Number(y[i]) \* l;

}

if ((z % 11)!== 0) {

$('#error01'). show ()

return false

}

else if ((z % 11) === 0 || y. length === 6) {

$('#input-membership'). parent (). remove Class('error')

$('#error01'). hide ()

return true

}

else {

$('#error03'). hide ();

}

}

}

module.exports = Validator