Phase2 JavaScript:

Doctor I added 5 extra pages that aid me with the need of JavaScript

Most of these pages had a separate file because to be honest I forgot the name of the css styles tags in the style sheet so I had to redo the styling for these pages instead of searching for the name then the characteristic and If it works or not

The added pages include:

- A modern admin page: contains tables that can access the database when I start with the php and when someone enters as an admin, this page will appear in the navbar
- login page and sign in pages: Although it is a blog, the user can login and sign up to view the new exciting events! and donate
- Cart page: you can check this page by entering the donation page →cards and click on donate, you'll have a variety of donation features. I choose this to be a cart to use it later for local storage and DOM manipulation plus this organization can't help only one person so the amount of money you want to spend will be divided to many children. However, I choose many $$ range to embrace the idea of helping even if you have 5$ (Lebanese bank rate: p)
- Visa page: when you donate you have the right to check your card and remove what you feel inappropriate. However, there is a submit button that will direct you to another page which is the checkout where you pay by providing the visa information and so on.

I added input validation for the login and sig up page for the sake of fulfilling phase2 requirements. However, the others will be validated through PHP.

And doctor while grading note that I'm doing the project by my own and I'm writing this report at 3:00 AM in the morning because of the overload. So please be lenient while grading, I tried to do my best <3 and I'm glad to know that the checklist idea helped you while grading.

On the technical side:

I have more than a folder up to 14-15 html pages, 3-4 css pages and 3 JavaScript folders and as I mentioned before this is due to the fact that I wanted a complete different design and keeping them in on folder while linking the css page while result in many problems.

JavaScript requirements:

Note that: There are many examples to show in every requirement but I'm choosing only one. Most of my examples will address cart → index.js

- ✓ use conditional statements and loop statements:
  if you check cart →index.js

```
for(let i=0; i<attToCartBtn.length; i++){
    attToCartBtn[i].addEventListener("click",function(e){
        if(typeof(Storage) !== 'undefined'){
            let item = {
                id:i+1,
```

I used a for loop to add to local storage.

- ✓ use functions:
  if you check login →login.js

```
form.addEventListener('submit', e => {
    e.preventDefault();

    checkInputs();
});

function checkInputs() {
```

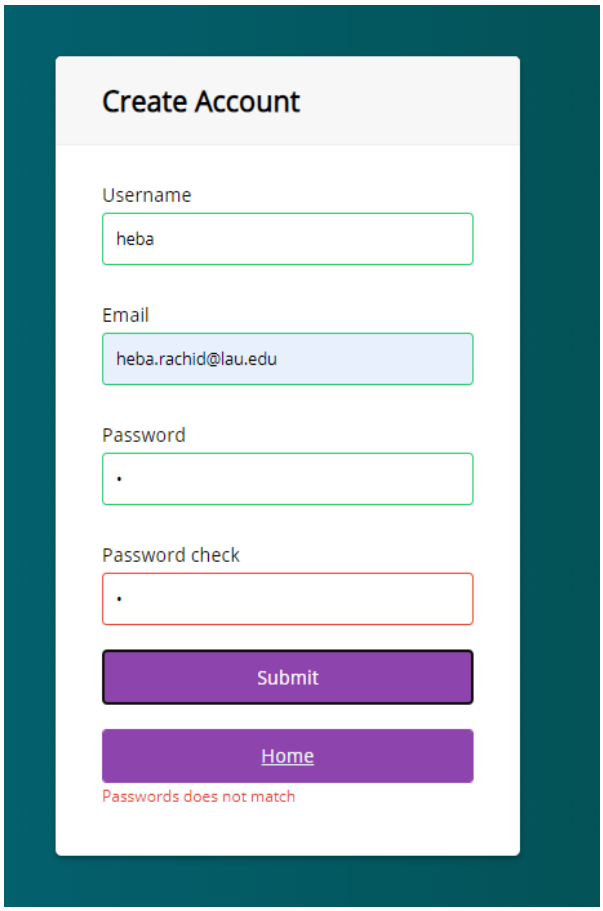
I used => to define a function
plus, the usual way which is checkInputs ().

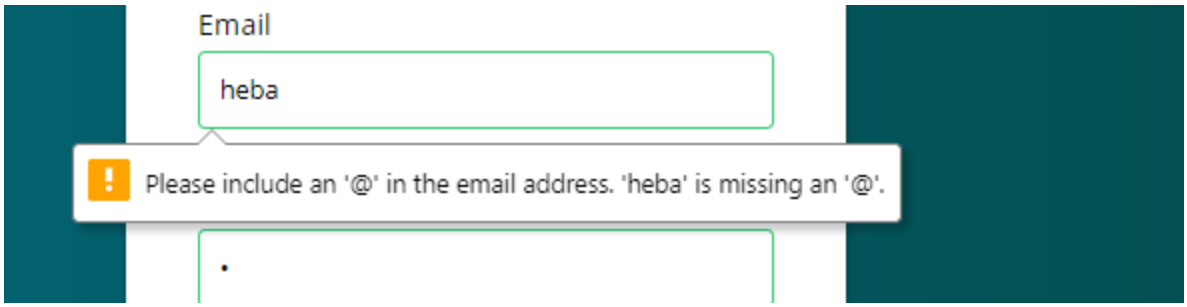
- ✓ use JavaScript arrays:
  if you check cart → index.js

```
let items = [];
for(let i=0; i<attToCartBtn.length; i++){
    attToCartBtn[i].addEventListener("click",function(e){
        if(typeof(Storage) !== 'undefined'){
            let item = {
                id:i+1,
                name:e.target.parentElement.children[0].textContent,
                price:e.target.parentElement.children[1].children[0].textContent,
                no:1
            };
```

- ✓ show the use of JavaScript for front-end form input validation
  if you check login →login.js
  the sign up and the log in contain validations for the email and the rechecked password

```
if(usernameValue === '') {
    setErrorFor(username, 'Username cannot be blank');
} else {
    setSuccessFor(username);
}

if(emailValue === '') {
    setErrorFor(email, 'Email cannot be blank');
} else if (!isEmail(emailValue)) {
    setErrorFor(email, 'Not a valid email');
} else {
    setSuccessFor(email);
}

if(passwordValue === '') {
    setErrorFor(password, 'Password cannot be blank');
} else {
    setSuccessFor(password);
}

if(password2Value === '') {
    setErrorFor(password2, 'Password2 cannot be blank');
} else if(passwordValue !== password2Value) {
    setErrorFor(password2, 'Passwords does not match');
} else{
    setSuccessFor(password2);
}


ction setErrorFor(input, message) {
 const formControl = input.parentElement;
 const small = formControl.querySelector('small');
 formControl.className = 'form-control error';
 small.innerText = message;
```





please note that you need to submit before the error pops up plus these validations and added to the contact us section of every page.

✓ show the use of query selectors:

```
const floating_btn = document.querySelector('.floating-btn');
const close_btn = document.querySelector('.close-btn');
const social_panel_container = document.querySelector('.social-panel-container');
```

login →login.js

I tried to use everything I learned in this project so I used const in one JavaScript file and var in another one. even though we used var more frequently.

- ✓ show the use of event listeners (e.g. on button click).
  I used many event listeners on submit form, button click …
  login →login.js

```
floating_btn.addEventListener('click', () => {
    social_panel_container.classList.toggle('visible')
});

close_btn.addEventListener('click', () => {
    social_panel_container.classList.remove('visible')
});
```

- ✓ demonstrate the use of DOM manipulation (e.g. dynamically create elements)
  cart →index.js

```
    // adding data to shopping cart
    const iconShoppingP = document.querySelector('.iconShopping p');
    let no = 0;
    JSON.parse(localStorage.getItem('items')).map(data=>{
        no = no+data.no
;   });
    iconShoppingP.innerHTML = no;


    //adding cartbox data in table
    const cardBoxTable = cartBox.querySelector('table');
    let tableData = '';
    tableData += '<tr><th>S no.</th><th>Item Name</th><th>Item No</th><th>item Price</th><th></th></tr>';
    if(JSON.parse(localStorage.getItem('items'))[0] === null){
        tableData += '<tr><td colspan="5">No items found</td></tr>'
    }else{
        JSON.parse(localStorage.getItem('items')).map(data=>{
            tableData += '<tr><th>'+data.id+'</th><th>'+data.name+'</th><th>'+data.no+'</th><th>'+data.price+'</th><th><a hre
        });
    }
    cardBoxTable.innerHTML = tableData;
}
```

The number will be incremented when you add and decremented when you remove

Donations

S no.	Item Name	Item No	item Price	
5	Provide housing and care	1	50	Delete
2	Provide medication for the ill	1	10	Delete
3	Put a smile with new clothes and gifts	2	7	Delete

Submit

- ✓ demonstrate dynamically setting element attributes (e.g. dynamically setting the src attribute of the image element)

I don't have to dynamically change the background of any image however I did it for only one image for the sake of this requirement.

```
<img  id="myImage" src="images/4686.jpg" />
<script> document.getElementById("myImage").src = 'images/Haitian-boy-2-with-food.png'; </script>
```

you can check this in the cart.php

if you want I can do as the dice game and when I click on something the image changes, but I guess there is no need and plus the only extra thing is

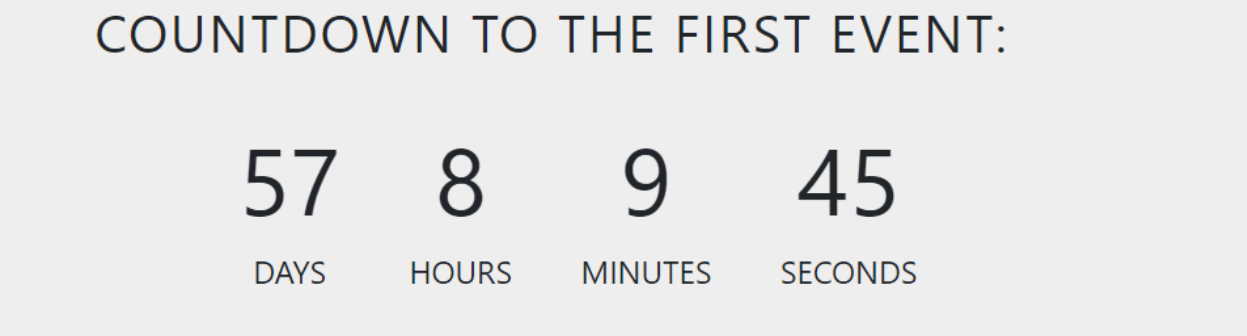
addEventListener while clicking the button and the function would set the image.

- ✓ demonstrate the use of setInterval and clearInterval methods (e.g. to create animated effects)

Check the admin page (to access it you need to be an admin sign up go to the database edit boolean admin be an admin then you can enter). There is a countdown to the first event planned and what you only have to do is set the date!

```
<script>(function () {
 const second = 1000,
 minute = second * 60,
 hour = minute * 60,
 day = hour * 24;

let birthday = "Feb 2, 2021 00:00:00",
 countDown = new Date(birthday).getTime(),
 x = setInterval(function() {

  let now = new Date().getTime(),
   distance = countDown - now;

  document.getElementById("days").innerText = Math.floor(distance / (day)),
   document.getElementById("hours").innerText = Math.floor((distance % (day)) / (hour)),
   document.getElementById("minutes").innerText = Math.floor((distance % (hour)) / (minute)),
   document.getElementById("seconds").innerText = Math.floor((distance % (minute)) / second);

  //do something later when date is reached
  if (distance < 0) {
   let headline = document.getElementById("headline"),
    countdown = document.getElementById("countdown"),
    content = document.getElementById("content");

   headline.innerText = "It's the first event";
   countdown.style.display = "none";
   content.style.display = "block";

   clearInterval(x);
  }
  //seconds
 }, 0)
}());
</script>
```

The script is within the page, admin.php.



✓ show the use of local storage (e.g. to locally store your shopping basket)

```
// adding data to localstorage
const attToCartBtn = document.getElementsByClassName('attToCart');
let items = [];
for(let i=0; i<attToCartBtn.length; i++){
    attToCartBtn[i].addEventListener("click",function(e){
        if(typeof(Storage) !== 'undefined'){
            let item = {
                    id:i+1,
                    name:e.target.parentElement.children[0].textContent,
                    price:e.target.parentElement.children[1].children[0].textContent,
                    no:1
                };
            if(JSON.parse(localStorage.getItem('items')) === null){
                items.push(item);
                localStorage.setItem("items",JSON.stringify(items));
                window.location.reload();
            }else{
                const localItems = JSON.parse(localStorage.getItem("items"));
                localItems.map(data=>{
                    if(item.id == data.id){
                        item.no = data.no + 1;
                    }else{
                        items.push(data);
                    }
                });
                items.push(item);
                localStorage.setItem('items',JSON.stringify(items));
                window.location.reload();
            }
        }else{
            alert('local storage is not working on your browser');
        }
    });
```

```
▼ [{id: 5, name: "Provide housing and care", price: "50", no: 1},…]
  ▶ 0: {id: 5, name: "Provide housing and care", price: "50", no: 1}
  ▶ 1: {id: 2, name: "Provide medication for the ill", price: "10", no: 1}
  ▶ 2: {id: 3, name: "Put a smile with new clothes and gifts", price: "7", no: 2}
```

^ If you check the local storage everything is stored there even when refreshing.

Bonus:

I used JQuery just as a bonus in the timeline in the index.html page. It is quite similar to php and JavaScript combined

phase 1 →script.js.

```
(document).ready(function() {
  $('.nav-button').click(function() {
    $('.nav-button').toggleClass('change');
  });

  $(window).scroll(function() {
    let position = $(this).scrollTop();
    if(position >= 200) {
      $('.nav-menu').addClass('custom-navbar');
    } else {
      $('.nav-menu').removeClass('custom-navbar');
    }
  });

  $(window).scroll(function() {
    let position = $(this).scrollTop();
    if(position >= 650) {
      $('.camera-img').addClass('fromLeft');
      $('.mission-text').addClass('fromRight');
    } else {
      $('.camera-img').removeClass('fromLeft');
      $('.mission-text').removeClass('fromRight');
    }
  });
  $('.gallery-list-item').click(function() {
      let value = $(this).attr('data-filter');
      if(value === 'all') {
        $('.filter').show(300);
      } else {
        $('.filter').not('.' + value).hide(300);
        $('.filter').filter('.' + value).show(300);
      }
    });

    $('.gallery-list-item').click(function() {
      $(this).addClass('active-item').siblings().removeClass('activ
    });
    $(window).scroll(function() {
      let position = $(this).scrollTop();
      if(position >= 4300) {
        $('.card-1').addClass('moveFromLeft');
        $('.card-2').addClass('moveFromBottom');
        $('.card-3').addClass('moveFromRight');
```