

GameOn



**Nouvelle page d'accueil et
formulaire d'inscription aux concours**

Créez une landing page avec Javascript

- Programmer en JavaScript

Soutenance du mercredi 3 août 2022

Table des matières

Objectif du projet

Présentation du travail effectué : vue du prototype

La Programmation en JavaScript

La Correction des erreurs d'affichage

Notices complémentaires : tests

Conclusion

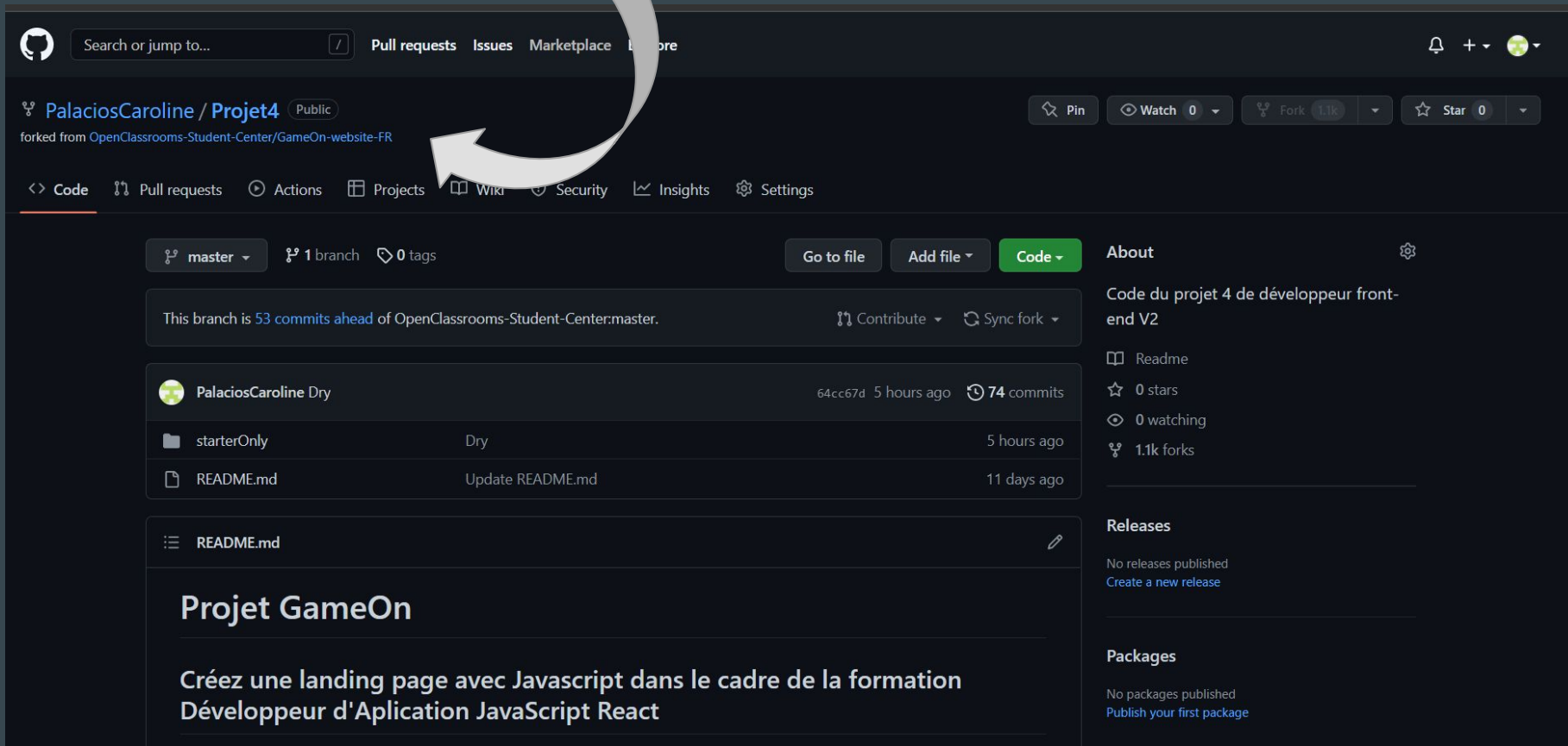
Objectif du projet

GameOn, PME spécialisée dans les conférences et les concours de jeux a simplifié la nouvelle page d'accueil et l'inscription aux concours du marathon national du jeux vidéo.

Il fallait prendre le relais de Jason et compléter la partie Front-End.

Mon travail a donc consisté à ajouter le code JavaScript manquant pour que le formulaire soit pleinement fonctionnel et corriger les erreurs d'affichage existantes.

Projet effectué à partir d'un repository GitHub forké



GitHub interface showing a repository named **PalaciosCaroline / Projet4** (Public), forked from **OpenClassrooms-Student-Center/GameOn-website-FR**.

The repository is on the **master** branch, which is **53 commits ahead** of the upstream repository. It has **1 branch** and **0 tags**.

Files listed:

- starterOnly** (Dry) - 5 hours ago
- README.md** (Update README.md) - 11 days ago

The README content includes the title **Projet GameOn** and the description: **Créez une landing page avec Javascript dans le cadre de la formation Développeur d'Aplication JavaScript React**.

Repository statistics:

- Code du projet 4 de développeur front-end V2
- 0 stars
- 0 watching
- 1.1k forks

Buttons: **Go to file**, **Add file**, **Code**, **About**, **Releases**, **Packages**.


Présentation du site

Résolutions des issues

Issue 1 : fermer la modale

```
const modalBtnClose = document.querySelector(".close");
```

Ajout également de la fermeture de la modale sur le bouton rouge présent sur le formulaire lors du message de réussite d'envoi en plus du bouton en croix.



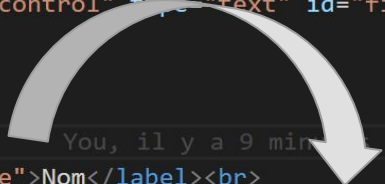
```
// close modal
const closeModal = () => {
  modalbg.style.display = "none";
  pageMain.style.display = 'block';
  pageFooter.style.display = "block";
}

// Close modal event
modalBtnClose.onclick = closeModal;
document.getElementById('close2').onclick = closeModal;
```


Issue 2 : (1) Lier les labels aux entrées dans le HTML en utilisant les attributs "for" et "id" dans le code existant.

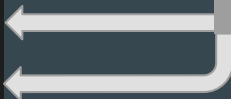
```
<label for="firstName">Prénom</label><br>
<input class="text-control" type="text" id="firstName" name="firstName"
minlength="2"/><br>
</div>
<div
class="formData">
  <label for="lastName">Nom</label><br>
  <input class="text-control" type="text" id="lastName" name="lastName"
minlength="2"/><br>
</div>
<div class="formData">
  <label for="email">E-mail</label><br>
  <input class="text-control" type="email" id="email" name="email"/><br>
</div>
<div class="formData">
  <label for="birthdate">Date de naissance</label><br>
  <input class="text-control" type="date" id="birthdate" name="birthdate"/>

```



```
<input class="checkbox-input" type="checkbox" id="checkbox1" checked
required/>
<label class="checkbox2-label label-cgu" for="checkbox1">
  <span class="checkbox-icon" id="span-cgu"></span>
  J'ai lu et accepté les conditions d'utilisation.
</label>
<br>
<input class="checkbox-input" type="checkbox" id="checkbox2" />
<label class="checkbox2-label" for="checkbox2">
  <span class="checkbox-icon"></span>
  Je souhaite être prévenu des prochains évènements.
</label>
```

Liaison des labels aux entrées dans le HTML en utilisant les attributs "for" et "id" dans le code



Issue 2 : (1) Corriger le code HTML quand nécessaire.

Changement des balises
afin d'utiliser des balises
sémantiques :

<header> et <nav>
ainsi que la création des
éléments pour le
hamburger

```
<header class="flex page_header">
  <div class="header-logo">
    
  </div>
  <div class="hamburger" id="hamburger">
    <div class="bar1"></div>
    <div class="bar2"></div>
    <div class="bar3"></div>
  </div>
  <nav class="header_nav">
    <ul class="ul-content navbar nav-ul flex" id="nav_ul">
      <li><a href="#">Détails de l'évènement</a></li>
      <li><a href="#">À propos</a></li>
      <li><a href="#">Contact</a></li>
      <li><a href="#">Évènements passés</a></li>
    </ul>
  </nav>
</header>
```

Ajout des éléments du Dom manquant afin de permettre leur manipulation pour les issues 2, 3 et 4

```
// DOM Elements
const modalbg = document.querySelector(".bground");
const modalBtn = document.querySelectorAll(".modal-btn");
const formData = document.querySelectorAll(".formData");
const modalBtnClose = document.querySelector(".close");
const form = document.getElementById('entry')
const firstName = document.getElementById('firstName');
const lastName = document.getElementById('lastName');
const email = document.getElementById('email');
const birthdate = document.getElementById('birthdate');
const quantity = document.getElementById('quantity');
const radios = document.querySelectorAll('input[name="location"]');
const checkBorder = document.querySelectorAll('.locations > .checkbox-label > span[class="checkbox-icon"]');
const location1 = document.getElementById("location1");
const checkbox1 = document.getElementById('checkbox1');
const cguBorder = document.getElementById('span-cgu')
const pageMain = document.getElementById('pageMain');
const pageFooter = document.getElementById('pageFooter');
const textControl = document.querySelectorAll('.text-control');
```

Issue 2 : Conservation des données du formulaire (ne pas effacer le formulaire) lorsqu'il ne passe pas la validation.

Utilisation de `preventDefault()` sur le formulaire, puis fonction `validate()` pour tester les inputs, si form validé alors log des résultats des différents inputs et vidage de ceux-ci.

```
const form = document.getElementById('entry')
```

```
//retain data when the form is incorrectly completed or send, log and delete input
form.addEventListener("submit", (event) => {
  event.preventDefault();
  validate();
  if (validate()) {
    logResults();
    emptyInput()}
  })
```


Issue 3 : Ajouter la validation ou le message d'erreur : gestion des messages d'erreur

```
.formData[data-error]::after {
  content: attr(data-error);
  font-size: 0.8em;
  display: block;
  margin-top: 4px;
  margin-bottom: 8px;
  text-align: left;
  line-height: 0.8;
  opacity: 0;
  transition: 0.3s;
}

.formData[data-error].white::after {
  color: white;
  opacity: 1;
}

.formData[data-error].red::after {
  color: red;
  opacity: 1;
}

.colorred{
  border: 2px solid red;
}
```

Modification du css dans la gestion des message :
Suppression de l'attribut en true.
Ajout de classe.

Ajout des messages d'erreurs dans le DOM pour l'instant invisible puisqu'en opacity 0 .

```
//Preparation of Error messsage (opacity = 0)
const placeMessage = (input,message) => input.parentNode.setAttribute('data-error',message);
placeMessage(firstName,'Entrer 2 lettres minimum pour le champ du prénom.')
placeMessage(lastName,'Entrer 2 lettres minimum pour le champ du nom.')
placeMessage(email,'Une adresse email valide est requise')
placeMessage(birthdate,'Entrer votre date de naissance, 12ans minimum sont requis')
placeMessage(quantity,'Une valeur numérique doit être saisie')
placeMessage(location1,'Veuillez choisir une option.')
placeMessage(checkbox1,'Veuillez vérifier que vous acceptez bien les termes et conditions.')
```

Ajout de message d'aide au remplissage des inputs (message d'erreur en blancs lors du focus de chaque input).

```
//writing white help message (color=white and opacity=1)
const errorWhite = (element) =>{element.parentNode.classList = ("formData white");}
```

Préparation des fonctions d'introduction de l'apparition des messages d'erreurs et des bordures en rouge.
Les messages d'erreur s'afficheront sous le champ de saisie associé.

```
//Writing red error message (color=red and opacity=1)
const errorRed = (element) => {element.parentNode.classList = ("formData red")};
function redBorder(input) {
  if(input == 'location1') {
    return checkBorder.forEach((i) => i.classList.add('colorred'))
  } else if (input == 'checkbox1') {
    return cguBorder.classList.add('colorred')
  }
}
```

Disparition du message d'erreur et des bordures rouges lorsque les données sont saisies correctement.

```
//Deletion of messages
const noError = (element) => element.parentNode.classList = ("formData");
function noBorder(input) {
  if (input == 'location1') {return checkBorder.forEach((i) => i.classList.remove('colorred'))}
  else if (input == 'checkbox1') {return cguBorder.classList.remove('colorred')}
}
```

Préparation des validations : regex et tests (prénom, nom, email, date de naissance, nombre de participations antérieures, choix du tournoi, et acceptation des cgu.)

```
//regex and test
const nameRegex = /^[a-zA-ZáàâãäåçèéêëíîïñóôõöúûüýÿæÀÀÃÄÅÇÈÉÊËÍÎÏÑÓÔÕÖÙÚÛÜÝŸÆ\']
{2,}[a-zA-ZáàâãäåçèéêëíîïñóôõöúûüýÿæÀÀÃÄÅÇÈÉÊËÍÎÏÑÓÔÕÖÙÚÛÜÝŸÆ\ -\' \s]{0,}$/;
function isTestOk(id) {
  switch (id) {
    case 'firstName': results['firstName']=firstName.value;
    |   return nameRegex.test(firstName.value.trim());
    |   break;
    case 'lastName': results['lastName']=lastName.value;
    |   return nameRegex.test(lastName.value.trim());
    |   break;
    case 'email' : results['email']=email.value;
    |   return (email.value.trim()).match(/^\\w+([\\.-]?\\w+)*@\\w+([\\.-]?\\w+)*(\\.\\w{2,3})+$/);
    |   break;
    case 'quantity' : results['participations précédentes à des tournois']=quantity.value;
    |   return quantity.value.match(/^\\[0-9]+$/);
    |   break;
```

Suite de la fonction isTestOk(id)

```
case 'birthdate':
  const date = new Date(birthdate.value);
  const now = Date.now();
  const oneYear = 365.25 * 24 * 60 * 60 * 1000; //one year of secondes
  const age = (now - date) / oneYear;
  if (!(date instanceof Date) || isNaN(date) || age < 12 || age > 130){
    return false;
  } else {
    results['birthdate']=birthdate.value;
    results['age']=`${age.toFixed(0)} ans`;
    return true;}
  break;
case 'location1':
  const Locationchecked = document.querySelector("input[name='location']:checked");
  if(Locationchecked !== null) {results['choice of tournament']=Locationchecked.value;
  } else {results['choice of tournament']= 'aucun tournoi n\'a été choisi';}
  return Locationchecked !== null;
  break;
case 'checkbox1' : results[checkbox1] = checkbox1.checked ;
  return checkbox1.checked;
  break;
default :
  break;
}}
```


Vérification en cours de saisie des données ou de son changement

Aide et vérification :

Un message d'aide en blanc est mis lors de l'écriture. Lorsque l'email est valide, le message disparaît.

Pour le choix du tournoi, le message d'erreur est supprimé si on en choisit un.

Pour le Cgu, un message d'erreur est mis lors du décochage de la checkbox.

```
//help and control on write
textControl.forEach((input) => input.addEventListener("keyup", function() {
  errorWhite(input);
  (isTestOk(this.id))? noError(input) : '';
}))
radios.forEach((input) => input.addEventListener("change", () => verif(location1)))//if change
checkbox1.addEventListener("change", () => verif(checkbox1));//warning if uncheck cgu
```

Vérification lors de la fin de remplissage : focusout

Aide et vérification :

Lors de la sortie du focus, la fonction de vérification est lancée pour vérifier si une erreur a été faite afin de ne pas attendre la soumission du formulaire.

```
//control focusout
textControl.forEach((input) => input.addEventListener("focusout", function() {
  |   (isTestOk(this.id))? noError(input) : errorRed(input);
}))
```

Vérification prénom, nom, email, date de naissance, quantité de tournoi, choix du tournoi et acceptation des cgu.

- Issue 2 (1) et (2) Le champ a un minimum de 2 caractères / n'est pas vide.
 (3) L'adresse électronique est valide.
 (4) Pour le nombre de concours, une valeur numérique est saisie.
 (5) Un bouton radio est sélectionné.
 (6) La case des conditions générales est cochée
- Issue 3 mise en place des messages d'erreurs

```
//Control firstName, lastName, email, quantity, birthdate, location of tournament and cgu
function verif(input) {
  if (!isTestOk(input.id)) {
    errorRed(input);
    if (input == location1 || input == checkbox1) {
      redBorder(input.id);
    }
    return false;
  } else {
    noError(input);
    if (input == location1 || input == checkbox1) {
      noBorder(input.id);
    }
    return true;
  }
}
```

Issue 2 : Validation en JavaScript du formulaire : le formulaire doit être valide quand l'utilisateur clique sur Submit . Les données doivent être saisies correctement

Vérification finale au moment du submit avec le lancement de toutes les vérification de tous les inputs.

et envoi seulement lorsque tous les inputs passent leur test et sont ok (true)

```
//form validation function
function validate() {
  //control of every inputs
  let i=0
  for(let input of textControl) {
    verif(input);
    if(verif(input) == false){i++}
  }
  verif(location1); //validate if one in locations is checked
  if (verif(location1) == false) {i++}
  verif(checkbox1);
  if(verif(checkbox1) == false){i++}
  if (i > 0) {
    return false;
  } else {
    messageEnvoi.style.display = 'flex';
    form.classList.add('invisibleForm');
    bruitAchievement();
    return true;
  }
}
```

Issue 4 : Ajouter confirmation quand envoi réussi - Préparation en HTML et CSS

Après une validation réussie, inclure un message de confirmation de la soumission réussie pour l'utilisateur.

Mise en place dans le HTML du message à faire apparaître à l'envoi du formulaire.

Et dans le CSS, mise en place de son style ainsi que d'une class permettant de faire disparaître le formulaire mais de conserver sa hauteur et largeur exacte (respect de la maquette)

```
<div class="bground">
  <div class="content" id="navBar">
    <span class="close"></span>
    <div class="flex content" id="messageEnvoi">
      <span>Merci pour
      <br>votre inscription</span>
      <button class="btn-close flex" id="close2">FERMER</button>
    </div>
  </div>
```

```
#messageEnvoi {
  justify-content: space-between;
  align-items: center;
  text-align: center;
  flex-direction: column;
  height: 60%;
  max-width: 550px;
  width: 100%;
  font-size: 2.5rem;
  position: absolute;
  z-index : 99;
  top:40%;
}
```

```
.btn-close{
  margin-bottom: 30px;
  background: #fe142f;
  justify-content: center;
  align-items: center;
  font-size: 16px;
  width: 40%;
  border-radius: 7px;
  padding: 20px 40px;
  color: #fff;
  cursor: pointer;
  border: 0;
}
```

```
.invisibleForm{
  opacity : 0;
}
```

```

//inputs values record for log in console
const results = {
  'firstName' : firstName.value,
  'lastName' : lastName.value,
  'email' : email.value,
  'birthdate' : birthdate.value,
  'age': '',
  'participations précédentes à des tournois' : quantity.value,
  'choice of tournement' : '',
  'cgu' : checkbox1.checked
}

//function log all results in console after validate
function logResults() {
  for (let property in results) {
    console.log(`${property}: ${results[property]}`)}
}

```

Création d'un Objet pour enregistrer les résultats des différentes valeurs des inputs afin de pouvoir les afficher dans la console lors de la validation du formulaire.

Fonction pour loguer les inputs et leur valeur dans la console.

Fonction de vidage des valeurs des inputs utilisée une fois le formulaire validé.


```

function emptyInput() {
  textControl.forEach((input) => input.value = '');
  radios.forEach((input) => input.checked = false);
}

```



issue 4 suite => Apparition du message de réussite d'envoi ou de sa disparition en JavaScript

Si réussite d'envoi déjà faite : Mise en place au lancement de la modal de la disparition du message afin de pouvoir ré-utiliser le formulaire pour une nouvelle inscription.



```
// launch modal open form
function launchModal() {
  bruitcongrats();
  modalbg.style.display = "block";
  pageMain.style.display = "none";
  pageFooter.style.display = "none";
  //rajout de réutilisation du formulaire après confirmation et fermeture
  messageVisible.style.display = 'none';
  form.classList.remove('invisibleForm');
```

Apparition du message de réussite en lui appliquant un display flex avec disparition du formulaire par un opacity à 0 ce qui permet de conserver la taille exacte du formulaire qui reste dessous grâce à un empilement avec un z-index inférieur.



```
//form validation function
function validate() {
  //control of every inputs
  let i=0
  for(let input of textControl) {
    verif(input);
    if(verif(input) == false){i++}
  }
  verif(location1); //validate if one in locations is checked
  if (verif(location1) == false) {i++}
  verif(checkbox1);
  if(verif(checkbox1) == false){i++}
  if (i > 0) {
    return false;
  } else {
    messageEnvoi.style.display = 'flex';
    form.classList.add('invisibleForm');
    bruitAchievement();
    return true;
  }
}
```

Check des tâches qu'il fallait effectuer :



Issue 1 :

fermer la modale

Ajouter la fonctionnalité au bouton (x)



Issue 2 :

Implémenter entrées du formulaire

- ✗ (1) Lier les labels aux entrées dans le HTML en utilisant les attributs "for" et "id" dans le code existant.
- ✗ Corriger le code HTML quand nécessaire.
- ✗ (2) Utiliser du JavaScript pur (pas de jQuery) pour terminer le formulaire :
- ✗ Le formulaire doit être valide quand l'utilisateur clique sur "Submit"
- ✗ Les données doivent être saisies correctement :
- ✗ (1) Le champ Prénom a un minimum de 2 caractères / n'est pas vide.
- ✗ (2) Le champ du nom de famille a un minimum de 2 caractères / n'est pas vide.
- ✗ (3) L'adresse électronique est valide.
- ✗ (4) Pour le nombre de concours, une valeur numérique est saisie.
- ✗ (5) Un bouton radio est sélectionné.
- ✗ (6) La case des conditions générales est cochée, l'autre case est facultative / peut être laissée décochée.
- ✗ Conserver les données du formulaire (ne pas effacer le formulaire) lorsqu'il ne passe pas la validation.

✖ Issue 3 :

Ajouter validation ou messages d'erreur

Des messages d'erreur spécifiques doivent apparaître sous l'entrée qui n'est pas correcte. Les messages d'erreur doivent s'afficher sous le champ de saisie associé.

✖ Issue 4 :

Ajouter confirmation quand envoi réussi

Après une validation réussie, inclure un message de confirmation de la soumission réussie pour l'utilisateur (ex. "Merci ! Votre réservation a été reçue.")

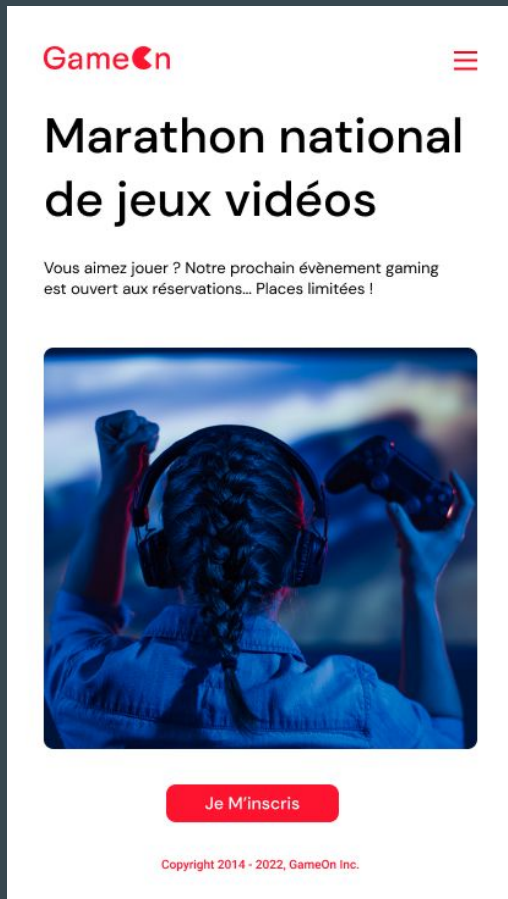
✖ Issue 5 :

Tests manuels

Visualiser et tester l'interface utilisateur dans les dernières versions de Chrome et de Firefox, ainsi que dans les versions mobile et desktop. Corriger les erreurs d'affichage existantes.

Tester toutes les fonctionnalités des boutons et des entrées de formulaire (tester les valeurs correctes et incorrectes)

Correction des erreurs
d'affichages
capture des tests de l' issue 5



Maquette

Ouverture du formulaire

Site

L'accès au menu hamburger est maintenu sur la version mobile alors que le formulaire est ouvert à la différence de la version desktop.

GameOn



Prénom

Nom

E-mail

Date de naissance

À combien de tournois avez-vous déjà participé ?

19

A quel tournoi souhaitez-vous participer cette année ?

☒ New York ☐ San Francisco ☐ Seattle ☐ Chicago
☐ Boston ☐ Portland

☒ J'ai lu et accepté les conditions d'utilisation.

☐ Je souhaite être prévenu des prochains événements.

C'est parti

GameOn



Merci pour
votre inscription

Fermer

GameOn



Merci pour
votre inscription

FERMER

GameOn



Prénom

Nom

Entrer 2 lettres minimum pour le champ du nom.

E-mail

Date de naissance

À combien de tournois GameOn avez-vous déjà participé ?

A quel tournoi souhaitez-vous participer cette année ?

☐ New York ☐ San Francisco ☐ Seattle ☐ Chicago ☐ Boston
☐ Portland

☒ J'ai lu et accepté les conditions d'utilisation.

☐ Je souhaite être prévenu des prochains événements.

C'est parti

Version Desktop

Ouverture du formulaire

Maquette

Prénom

Nom

E-mail

Un email valide doit être renseigné pour recevoir les notifications.

Date de naissance

À combien de tournois GameOn avez-vous déjà participé ?

A quel tournoi souhaitez-vous participer cette année ?

☒ New York ☐ San Francisco ☐ Seattle ☐ Chicago
☐ Boston ☐ Portland

☒ J'ai lu et accepté les conditions d'utilisation.

☐ Je souhaite être prévenu des prochains événements.

C'est parti

En Keyup un message d'aide s'inscrit en blancs. Si Ok le message disparaît. En focusout, en cas d'erreur, un message d'erreur rouge apparaît et la bordure de l'input se met également en rouge.

Site

Prénom

Entrer 2 lettres minimum pour le champ du prénom.

Nom

E-mail

Date de naissance

À combien de tournois GameOn avez-vous déjà participé ?

A quel tournoi souhaitez-vous participer cette année ?

☒ New York ☐ San Francisco ☐ Seattle ☐ Chicago ☐ Boston
☐ Portland

☒ J'ai lu et accepté les conditions d'utilisation.

☐ Je souhaite être prévenu des prochains événements.

C'est parti

En version mobile :
Accès de la barre
de navigation
conservée

GameOn



Marathon national de jeux vidéos

Vous aimez jouer ? Notre prochain évènement gaming est ouvert aux réservations... Places limitées !



Je M'inscris

Copyright 2014 - 2022, GameOn Inc.

GameOn



Prénom

Nom

E-mail

Date de naissance

jj/mm/aaaa 

À combien de tournois GameOn avez-vous déjà participé ?

A quel tournoi souhaitez-vous participer cette année ?

- ☐ New York ☐ San Francisco ☐ Seattle ☐ Chicago ☐ Boston
☐ Portland

☒ J'ai lu et accepté les conditions d'utilisation.

☐ Je souhaite être prévenu des prochains événements.

C'est parti

Les liens dans la barre de navigation deviennent rouge au survol et à l'active et sont surlignés en blanc



GameOn

Détails de l'évènement

À Propos

Contact

Évènements passés

Marathon national de jeux vidéos

Vous aimez jouer ? Notre prochain
événement gaming est ouvert aux
réservations... Places limitées !

Je M'inscris



GameOn

Détails de l'évènement

À propos

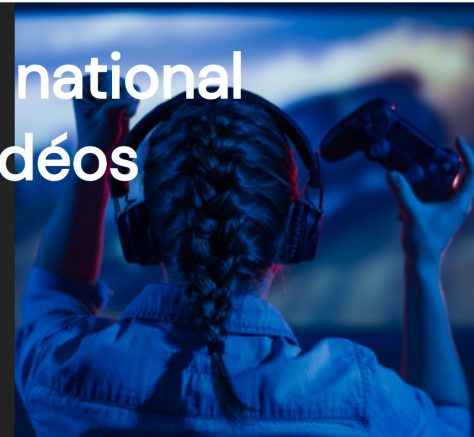
Contact

Évènements passés

Marathon national de jeux vidéos

Vous aimez jouer ? Notre prochain
événement gaming est ouvert aux
réservations... Places limitées !

Je M'inscris



Mozilla Firefox

Prototype mobile

Google Chrome

GameOn

×

Prénom

Nom

E-mail

Une adresse email valide est requise

Date de naissance

jj / mm / aaaa

À combien de tournois GameOn avez-vous déjà participé ?

A quel tournoi souhaitez-vous participer cette année ?

☐ New York

☐ San Francisco

☐ Seattle

☐ Chicago

☐ Boston

☐ Portland

☒ J'ai lu et accepté les conditions d'utilisation.

☐ Je souhaite être prévenu des prochains événements.

C'est parti

GameOn

×

Prénom

Nom

Entrer 2 lettres minimum pour le champ du nom.

E-mail

Date de naissance

jj / mm / aaaa

À combien de tournois GameOn avez-vous déjà participé ?

A quel tournoi souhaitez-vous participer cette année ?

☐ New York

☐ San Francisco

☐ Seattle

☐ Chicago

☐ Boston

☐ Portland

☒ J'ai lu et accepté les conditions d'utilisation.

☐ Je souhaite être prévenu des prochains événements.

C'est parti

Mozilla Firefox

Prototype Desktop

Google Chrome

GameOn

Détails de l'évènement

À propos

Contact

Évènements passés

Marathon national de jeux vidéos

Vous aimez jouer ? Notre prochain
événement gaming est ouvert aux
réservations... Places limitées !

Je M'inscris



GameOn

Détails de l'évènement

À propos

Contact

Évènements passés

Marathon national de jeux vidéos

Vous aimez jouer ? Notre prochain
événement gaming est ouvert aux
réservations... Places limitées !

Je M'inscris



Tentative de
soumission du
formulaire :
tous les tests se
déclenchent.
Les messages
d'erreurs
apparaissent.

GameOn

Prénom

Entrez 2 lettres minimum pour le champ du prénom.

Nom

Entrez 2 lettres minimum pour le champ du nom.

E-mail

Une adresse email valide est requise

Date de naissance

jj/mm/aaaa

Entrez votre date de naissance, 12ans minimum sont requis

À combien de tournois GameOn avez-vous déjà participé ?

Une valeur numérique doit être saisie

A quel tournoi souhaitez-vous participer cette année ?

☐ New York ☐ San Francisco ☐ Seattle ☐ Chicago ☐ Boston

☐ Portland

☐ Vous devez choisir une option.

☐ J'ai lu et accepté les conditions d'utilisation.

☐ Je souhaite être prévenu des prochains événements.

Veillez vérifier que vous acceptez bien les termes et conditions.

C'est parti

GameOn

Prénom

Entrez 2 lettres minimum pour le champ du prénom.

Nom

Re

E-mail

Une adresse email valide est requise

Date de naissance

jj/mm/aaaa

Entrez votre date de naissance, 12ans minimum sont requis

À combien de tournois GameOn avez-vous déjà participé ?

Une valeur numérique doit être saisie

A quel tournoi souhaitez-vous participer cette année ?

☐ New York ☒ San Francisco ☐ Seattle ☐ Chicago ☐ Boston

☐ Portland

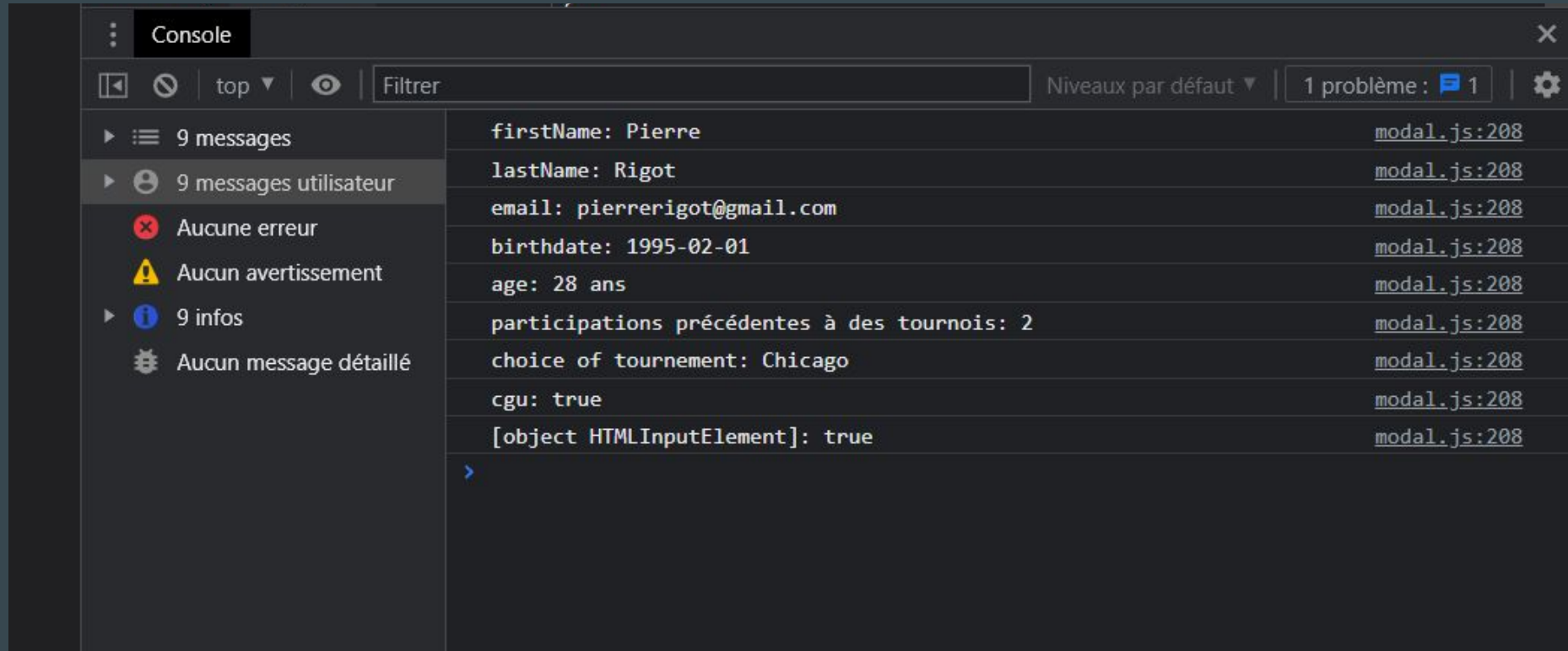
☒ J'ai lu et accepté les conditions d'utilisation.

☐ Je souhaite être prévenu des prochains événements.

C'est parti

Lors de la
correction, le
message d'erreur
passe en blancs,
l'input passe en
même temps le test,
si ok alors
disparition du
message d'erreur.

Valeurs des différents input afficher dans la console



The screenshot shows a web browser's developer console with the 'Console' tab selected. The console displays a list of 9 messages, with the first message expanded to show a user profile object. The profile data is as follows:

Property	Value	Source
firstName	Pierre	modal.js:208
lastName	Rigot	modal.js:208
email	pierrerrigot@gmail.com	modal.js:208
birthdate	1995-02-01	modal.js:208
age	28 ans	modal.js:208
participations précédentes à des tournois	2	modal.js:208
choice of tournament	Chicago	modal.js:208
cgu	true	modal.js:208
[object HTMLInputElement]	true	modal.js:208

The console interface includes a search bar, a filter dropdown, and a status bar indicating '1 problème : 1'.

Bilan du projet

Utilisation de JavaScript qui bloque l'accès du formulaire aux personnes qui n'auraient pas JavaScript activé. Possibilité de le mettre plus tôt sur une page d'inscription mise en lien avec une vérification en HTML5 des entrées des inputs avec une stylisation seulement des messages d'erreurs.

Également faire attention aux regex qui peuvent rendre inaccessible le formulaire à certaines personnes lorsqu'elles sont trop strictes.

Notices :

W3C validator résultat identique pour les deux versions
filtres (button ou checkbox):

- HTML5 : aucune erreur
- CSS3 : aucune erreur

Test HTML

Nu Html Checker

Cet outil est une expérience en cours pour une meilleure vérification HTML, et son comportement reste sujet à changement

Affichage des résultats pour le fichier téléchargé index.html

Entrée de vérificateur

Spectacle ☐ la source ☐ présenter ☐ Options de rapport d'image ...

Vérifier par téléchargement de fichiers ▼ Choisir un fichier Aucun fichier choisi

Les fichiers téléchargés avec des extensions .xhtml ou .xht sont analysés à l'aide de l'analyseur XML.

Vérifier

Vérification des documents terminée. Aucune erreur ou avertissement à afficher.

J'ai utilisé l'analyseur HTML.

Temps d'exécution total 17 millisecondes.

[À propos de ce vérificateur](#) • [Signaler un problème](#) • Version : 22.7.25

Test CSS



Service de validation CSS du W3C

Résultats de la validation W3C CSS de modal.css (CSS niveau 3 + SVG)

Aller à: [Avertissements \(3\)](#)

[CSS valide](#)

Résultats de la validation W3C CSS de modal.css (CSS niveau 3 + SVG)

Félicitations ! Aucune erreur trouvée.

Ce document est valide conformément à la recommandation [CSS niveau 3 + SVG](#) !

Pour montrer à vos lecteurs votre souci d'interopérabilité lors de la création de cette page Web, vous pouvez afficher cette icône sur toutes les pages valides. Voici le fragment de XHTML que vous pouvez utiliser pour ajouter cette icône à votre page Web:



```
<p>
  <a href="http://jigsaw.w3.org/css-validator/check/referer">
    
    </a>
  </p>
```



```
<p>
  <a href="http://jigsaw.w3.org/css-validator/check/referer">
    
    </a>
  </p>
```

