

Community Event

Partie utilisateur	3
En-tête (header)	3
Pied de page (footer)	3
Page d'accueil (/index.php)	3
Catalogue d'événements (/searchevent.php)	3
Page d'événement	4
Ajout d'événement (/add_event_form.php)	5
Création de compte (/new_user.php)	5
Partie développeur	7
Présentation de la base de donnée	7
Principales fonctionnalités du site : leur principe	7
Créer un nouvel utilisateur (table Users)	7
Créer un événement (table Event et la relation isofcategory)	7
Ajouter une catégorie (table Category)	7
Ajouter une photo à un événement (table Photo, la relation isphotoof et postedby)	7
Afin d'évaluer un événement (relation Rated)	8
Pour participer à un événement (relation Participated)	8
Ajouter un lien parent/enfant (relation isparentof)	8
Suggestion request:	9
Liste des événements évaluable :	10
Le compte admin	10
Répartition du travail	11

Johnson VILAYVANH

Alexandre ROBIC

Partie utilisateur

L'accès au site web se fait à partir de l'adresse <http://communityevent.tech/>.

➤ En-tête (header)

- L'en tête contient les informations suivantes:
 - Titre du site (Community Event)
 - Logo du site
 - Nom, prenom et age de l'utilisateur (si il est connecté, par défaut on mettra "Not logged in")
 - Date et heure (au format US)
 - Le bouton de déconnection (dans le cas ou on est connecté)

➤ Pied de page (footer)

- Le pied de page contient les informations suivantes:
 - Un lien vers la page [About](#) qui contient les informations sur les développeurs (Alexandre et Johnson) ainsi que le nombre d'événements, d'utilisateurs et ratings au total. Une fois ce rapport terminé, il sera mis sur cette page
 - Un lien vers la page [Legal](#) qui définit la licence du logiciel (open source) et permet de télécharger un fichier zip contenant tous les fichier nécessaires à la mise en place de votre propre community event. Ce fichier zip est généré à chaque téléchargement (l'ajout/suppression de données dans la base entraînera une version différente du zip au prochain téléchargement)

➤ Page d'accueil (/index.php)

Sur la page d'accueil du site, l'utilisateur a accès à plusieurs liens et informations:

- La devise de communityevent.tech (A convenient tool to manage small scale local events) ainsi que le logo (disponible sur toutes les pages en en-tête)
- 2 liens concernant les comptes:
 - Un premier permettant d'accéder à la page /new_user.php qui permet de créer un compte (et donc de rajouter une ligne à la table users)
 - Un deuxième permettant d'accéder au contenu à partir d'un compte déjà établi (</login.php>). On est ramené à cette page dès qu'on choisit de se déconnecter (cliquer sur le bouton Log off situé dans l'en-tête)
- L'événement le plus populaire du site (celui avec le plus d'utilisateurs inscrits)
- L'événement le mieux noté (on prend tous les événements et leur note sur 5 et tri descendant et on sélectionne le premier)

➤ Catalogue d'événements (/searchevent.php)

- Cette page permet d'accéder à tous les événements dont l'âge de participation minimale est inférieure ou égale à l'âge de l'utilisateur (calcule lors du login grâce à

la différence entre la date d'anniversaire et la date du jour). Par défaut quand on arrive sur la page, les événements sont triés par ordre d'ajout dans la base. On peut trier les événements selon plusieurs critères:

- Nom
- Date (de début, par ordre décroissant)
- Lieu
- Rating
- Catégorie (Parmi celles proposées, seuls les événements étant de cette catégorie seront affichés)
- Les événements sont également triés de manière implicite par la couleur de fond du résultat de recherche:
 - Événement FINI (Done)
 - Événement EN COURS (Ongoing)
 - Événement À VENIR (Upcoming)
- Chaque événement dans les résultats de recherche est un lien vers la page de cet événement.
- Pour chaque événement dans le résultat de recherche, on a accès aux informations suivantes:
 - Titre
 - Photo de couverture
 - Lieu
- De plus, sur cette page on trouvera des liens permettant de [créer un événement](#) et avoir des [suggestions d'événements](#)
- Les recherches permettent de chercher les événements avec le critère dans la barre de recherche par titre de l'événement ou par ces mêmes mots dans la description

➤ **Page d'événement**

Il s'agit de la page générée lorsque l'on clique sur un événement. Sur cette page on retrouvera les informations suivantes:

- Un lien vers le catalogue d'événements (un bouton Back)
- La présentation de l'événement:
 - Titre (ex: Meet and Greet With Elmo)
 - Photo de couverture (vue dans le catalogue d'événements)
 - Dates de début et de fin (intervalle de l'événement)
 - Lieu
 - Statut de l'événement ([voir catalogue d'événements](#))
 - Description
 - Âge minimum de participation (si il y en a pas, "Event for everyone" sera affiché)
 - Une barre contenant plus de photos de l'événement si elles ont été ajoutées
 - Hôte de l'événement (ainsi qu'un lien mailto pour contacter l'hôte pour plus de questions)

- Un formulaire pour uploader une photo de l'évènement
- Un formulaire pour ajouter une évaluation de l'évènement (qui sera remplacé par un message d'erreur si on a déjà mis assez d'évaluations pour le moment ou si l'évènement n'est pas passé ou si on a pas participé)
- Le rating moyen (entre 1 et 5)
- Les ratings des différents utilisateurs ayant participé. Les informations disponibles sont:
 - Nom, prénom
 - Score (entre 1 et 5)
 - Date de participation
 - Commentaire (justificatif de l'évaluation)
- Un lien pour accéder au formulaire d'inscription à l'évènement (dans le cas où l'utilisateur est déjà inscrit, il lui sera précisé qu'il ne peut pas s'inscrire. Même chose si l'évènement est déjà passé)
- Le nombre de personnes participant ou ayant participé (incluant l'utilisateur)

➤ **Ajout d'évènement (/add_event_form.php)**

- Cette page permet de créer un événement en fournissant les informations suivantes dont certaines **obligatoires**:
 - **Titre de l'évènement**
 - **Lieu**
 - **Date de Début**
 - **Date de Fin**
 - Limite d'âge (âge minimum, par défaut 0 donc pas de limite)
 - Description
 - Photo de couverture
- A tout cela s'ajoute des catégories permettant de mieux classer (et donc mieux proposer) les événements aux utilisateurs. L'utilisateur peut en choisir plusieurs et en créer une s'il n'en trouve pas à son goût.

➤ **Création de compte (/new_user.php)**

- Un nouvel utilisateur débarque! Pour commencer à utiliser communityevent.tech, on doit préciser obligatoirement:
 - Prenom
 - Nom
 - Email
 - Date de naissance
 - Mot de passe (à confirmer avant de pouvoir s'inscrire)
- Une fois ces informations ajoutées, l'utilisateur est amené à se connecter grâce à un lien vers la page de [login](#)

Partie développeur

➤ Présentation de la base de donnée

La base de donnée n'ayant pas changé depuis le 2ème rapport, voici un [lien](#) vers la documentation de la base que nous avons utilisée.

➤ Principales fonctionnalités du site : leur principe

Avant toute chose, il est important de souligner que pour accéder à toutes les fonctionnalités principales du site (créer un événement, évaluer un événement, etc...) il est nécessaire de se connecter : afin de contrôler qu'un utilisateur est bien connecté à une page, on utilisera le principe des **SESSIONS**.

Pour chacune des tables et relations de la base de donnée, nous utilisons des **formulaires** afin de les remplir avec les informations correspondantes :

- Créer un nouvel utilisateur (table **Users**)
 - Un simple formulaire dans lequel chaque champ correspond à une information concernant l'utilisateur. Chaque champ présente l'option "**require**" qui impose que celui-ci doit être rempli, sinon on ne peut pas valider le formulaire
- Créer un événement (table **Event** et la relation **isofcategory**)
 - De la même façon que le formulaire pour créer un utilisateur, on doit d'abord informer dans chacun des champs présents les informations correspondant à l'événement qu'on veut créer
 - Sur le formulaire, on va également utiliser un tableau qui recense toutes les catégories qui ont été créées par les utilisateurs, tout en laissant la possibilité d'en créer une nouvelle
 - On modifie également la relation **isofcategory** étant donné qu'on a la possibilité d'ajouter les catégories auxquelles un événement correspond
- Ajouter une catégorie (table **Category**)
 - Lorsque l'utilisateur choisit de créer une catégorie lors de sa création d'événement, il va être redirigé vers un formulaire à champ unique dans lequel l'utilisateur doit indiquer le nom de la catégorie qu'il veut créer
- Ajouter une photo à un événement (table **Photo**, la relation **isphotoof** et **postedby**)
 - En dehors de la photo de couverture qui est ajoutée lors de la création d'un événement, les utilisateurs ont également la possibilité d'ajouter des photos à n'importe quel événement en passant par un formulaire également à champ unique : il s'agit d'un *file* qui accepte les images en format png/jpg.

- Afin d'évaluer un événement (relation ***Rated***)
 - On utilise également un formulaire pour évaluer un événement, cependant contrairement aux précédents formulaires, de nombreux prérequis doivent être validés par l'utilisateur afin qu'il puisse accéder à la page pour évaluer un événement.
 - Est-ce que l'événement a déjà commencé ?
 - Est-ce que l'utilisateur participe celui-ci ?
 - Est-ce que l'utilisateur a déjà participé à cet événement, donc est-ce qu'on est bien à une date postérieure à sa date de participation
 - Est-ce que l'utilisateur a déjà évalué l'événement autant de fois qu'il a participé à celui-ci ?
 - On va utiliser de nombreuses requêtes afin de récupérer chacune des informations qu'il nous faut, donc une assez simple, mais qui utilise une option qui est assez rarement utilisé : **EXCEPT** (on reparle de la façon dont on va l'utiliser dans une partie suivante concernant quelques suggestions de requêtes)
- Pour participer à un événement (relation ***Participated***)
 - On liste toutes les dates d'un événement (sauf celles qui sont déjà passées) dans un formulaire où l'utilisateur coche les dates auxquelles il veut participer à celui-ci
- Ajouter un lien parent/enfant (relation ***isparentof***)
 - Cette fonctionnalité présente une particularité car seul le compte **admin** (décrit dans une partie suivante) peut avoir accès à cette fonctionnalité. Cependant, son fonctionnement reste identique aux formulaires précédents : on choisit seulement 2 utilisateurs, un parent, un enfant. On note qu'on a interdit de lier par un lien parental 2 utilisateurs ayant moins de 14 ans d'écart (sauf cas vraiment exceptionnel, cela reste tout de même rare d'avoir un enfant à moins de 14 ans...)

On peut également souligner quelques autres fonctionnalités, qui ne vont pourtant pas modifier la base de donnée, mais extraire des informations qui nous sont utiles :

➤ **Suggestion request:**

- La suggestion d'événements pour un utilisateur se fait en 2 étapes
 - La première consiste à récupérer les catégories des événements auxquels l'utilisateur s'est inscrit:

```
SELECT DISTINCT category_id AS cat_id FROM category
NATURAL JOIN participated --Links event and user
NATURAL JOIN users --Links user and participation
NATURAL JOIN isofcategory --Links event and category
WHERE user_email = 'EMAIL') AS favcategories
```

Dans cette requête, on lie les tables participated (pour voir les événements auxquels les utilisateurs ont participé), users (pour recouper les informations obtenues avec un utilisateur en particulier), isofcategory (pour obtenir les catégories des événements). Cette requête deviendra ensuite une sous requête lors de la suggestion finale

- La deuxième consiste à récupérer tous les événements auxquels l'utilisateur n'a pas participé

```
--GET events user didn't sign up to that aren't done yet
--or still haven't happened
(SELECT event_id AS event_id FROM event
WHERE event_id NOT IN (
SELECT DISTINCT event_id FROM event
NATURAL JOIN participated
NATURAL JOIN users
WHERE user_email = 'EMAIL')
AND event_date + event_duration > CURRENT_DATE) AS notparticipating
```

Dans cette requête, on lie les tables event (pour obtenir les informations d'événement mais surtout les événements n'apparaissent pas dans la liste de ceux auxquels l'utilisateur a participé), participated (pour obtenir les événements auxquels l'utilisateur a participé) et users (pour prendre les informations seulement de l'utilisateur avec le mail EMAIL. On déterminera dans un premier temps les événements auxquels l'utilisateur a participé (voir sous requête entre parenthèses) puis on prendra tous ceux n'apparaissant pas dans cette sous requête.

- Enfin, on ajoute les lignes suivantes:

```
WHERE category.category_id = favcategories.cat_id
AND event.event_id = notparticipating.event_id
```

Afin de lier les 2 requêtes ensemble (qui deviennent donc des sous requêtes). Ceci permet donc d'obtenir des événements avec les mêmes catégories que d'autres auxquels l'utilisateur s'est inscrit mais pour lesquels il n'est pas inscrit.

➤ **Liste des événements évaluable :**

Sur la page d'accueil du site, l'utilisateur a un lien le redirigeant vers tous les événements qu'il peut évaluer. La requête qui nous permet de récupérer la liste des événements qu'un utilisateur peut évaluer se compose en 3 parties, dont 1 sous-requête :

```
• SELECT event_id
•     date_of_participation
• FROM participated
• WHERE user_id = 'USER_ID'
• AND date_of_participation < 'TODAY'
•
•
```

```
SELECT DISTINCT event_name,
                event_id
FROM event
NATURAL JOIN
(
    SELECT event_id,
           date_of_participation
    FROM participated
    WHERE user_id = $user_id
    AND date_of_participation < '$today'

    EXCEPT

    SELECT event_id,
           date_of_rating AS date_of_participation
    FROM rated
    WHERE user_id = $user_id
) AS canrate_event;"
```

➤ **Le compte admin**

➤ **Répartition du travail**

Une page dédiée sur le site présente la répartition des tâches pour la création du site (voir </about.php>).