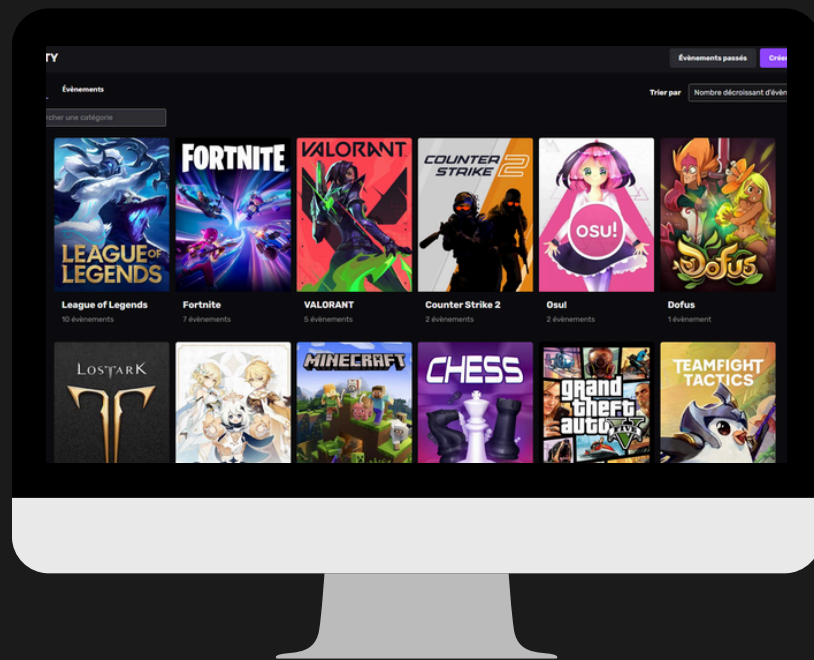




# PROJET WEB

## GAMEPARTY



# GAMEPARTY

GAMEPARTY est une plateforme qui permet d'organiser des événements de jeux vidéo. Que vous soyez un passionné de VALORANT, un stratège de League of Legends, un as de Fortnite, ou un amateur de tout autre jeu vidéo, GAMEPARTY saura satisfaire tous les goûts !

URL back-end : <https://github.com/Joelle13/event-manager-backend>

URL front-end : <https://github.com/juliehuyen/event-manager-frontend>

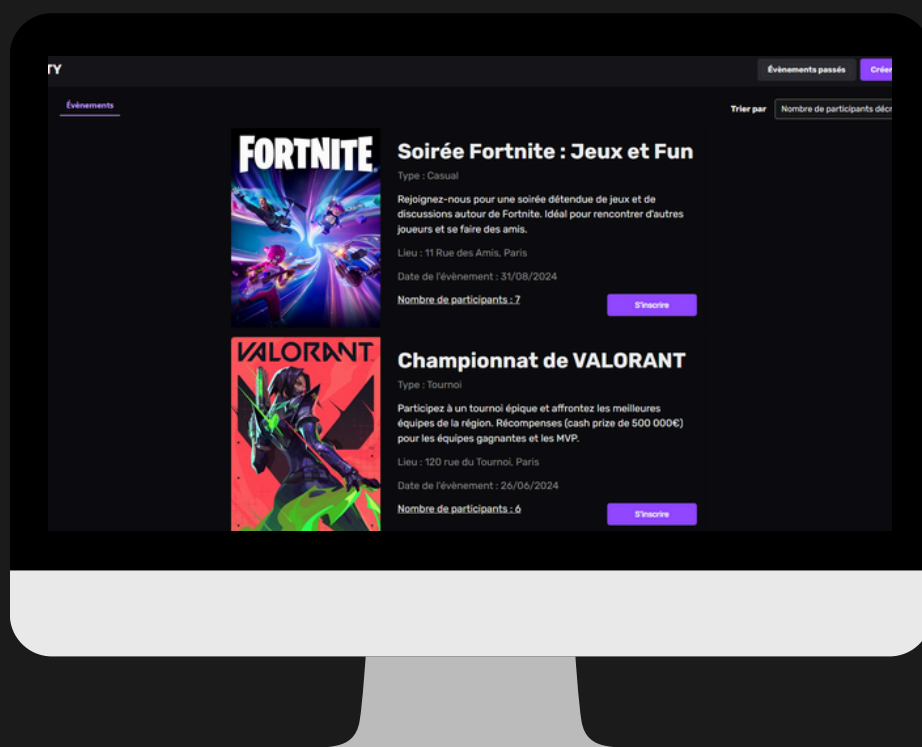
Notre projet rencontre un conflit de dépendances au niveau du front-end, ainsi pour pouvoir installer correctement les dépendances, il est indispensable d'ajouter le paramètre "**--legacy-peer-deeps**" à la commande "**npm install**"

Commande entière : **npm install --legacy-peer-deeps**

Dans ce rapport, nous vous présenterons dans un premier temps la documentation technique de notre projet, puis notre feedback concernant le cours de programmation web que vous avez réalisé en duo.

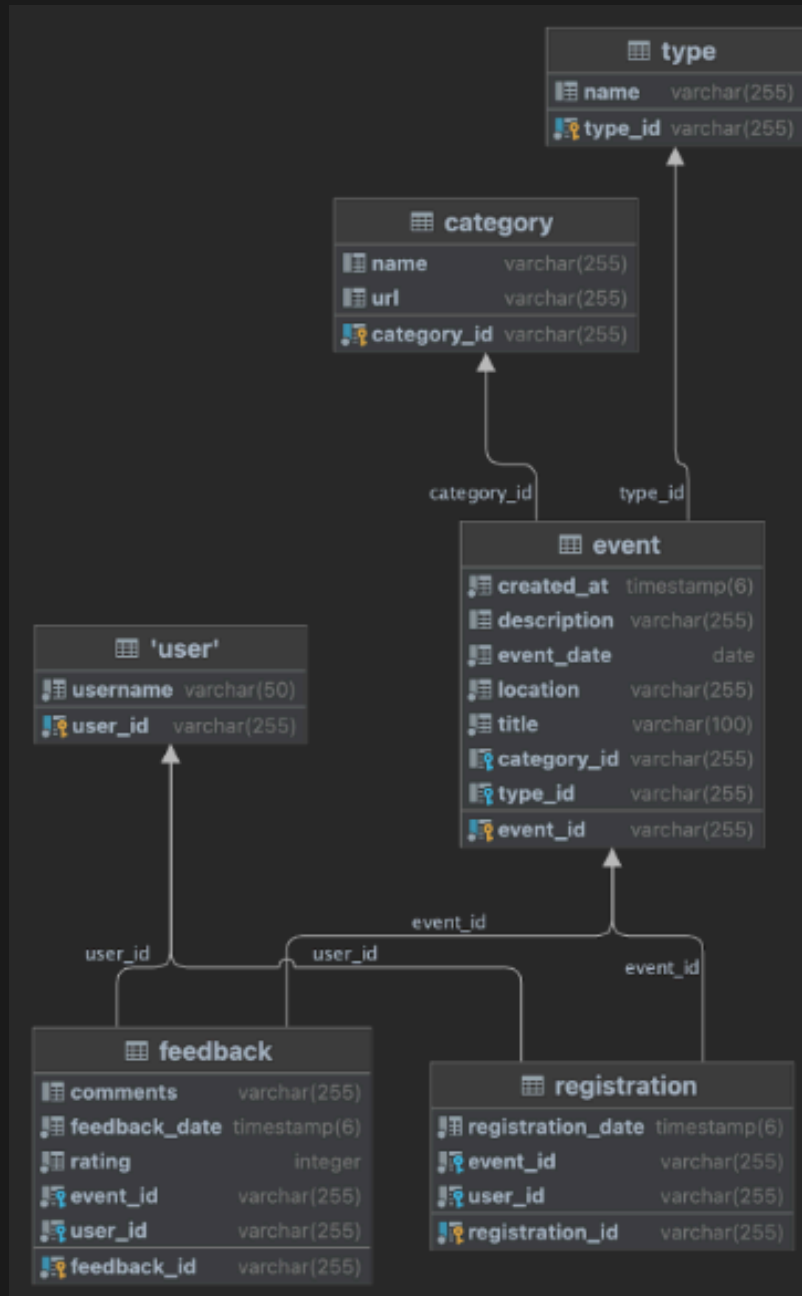
Un petit passage pour remercier vos enseignements et cours qui nous ont toutes les deux très plu, vous en saurez plus dans la partie concernée ! Merci Elie, merci Bilal pour nous avoir transmis vos connaissances sur la programmation web !

Bonne lecture à vous !



# DOCUMENTATION TECHNIQUE

## DIAGRAMME DE LA BASE DE DONNÉES



## CHOIX TECHNIQUES

Pour ce projet de gestion d'événements, nous avons décidé de rester très proches des pratiques et des technologies abordées en travaux pratiques (TP) afin de capitaliser sur les connaissances acquises en cours.

Nous avons utilisé Spring Boot pour construire notre application back-end. Spring Boot est un choix naturel grâce à sa robustesse et à sa capacité à simplifier le développement d'applications Java. Pour documenter et tester nos endpoints, nous avons intégré Swagger, ce qui nous permet de disposer d'une interface conviviale pour interagir avec notre API. Pour la gestion de la base de données, nous avons opté pour Neon SQL. Ce choix a été influencé par les retours positifs de nos camarades, notamment par rapport à Elephant SQL, qui était jugé moins performant et moins accessible.

Côté front-end, nous avons utilisé Angular, un framework que nous avons déjà manipulé en cours. Angular nous permet de créer des interfaces utilisateur dynamiques et réactives, améliorant ainsi l'expérience utilisateur de notre application. Pour le style, nous avons utilisé Bootstrap, une bibliothèque CSS populaire pour ses composants préconçus et réactifs. Cependant, nous avons ajouté notre propre CSS pour personnaliser l'apparence de notre application, lui donnant ainsi une identité visuelle unique. Cette approche nous a permis de bénéficier de la structure et des fonctionnalités de Bootstrap tout en ajoutant une touche personnelle à notre design.

Nous avons utilisé Maven et npm pour la gestion des dépendances et le déploiement continu. Ces outils nous permettent de maintenir une structure de projet propre.

## PROBLÈMES RENCONTRÉS

Lors du développement de notre projet, nous avons rencontré plusieurs difficultés, notamment en ce qui concerne l'intégration entre le back-end et le front-end. L'un des principaux défis a été la gestion des URLs, en particulier les différences entre les paramètres de requête (RequestParam) et les variables de chemin (PathVariable). Ces différences ont souvent conduit à des incohérences et des erreurs lors de l'appel aux services back-end depuis le front-end. Par exemple, les URLs générées pour les requêtes avec RequestParam nécessitaient des paramètres sous forme de clé-valeur dans la chaîne de requête, tandis que celles avec PathVariable exigeaient des segments spécifiques dans l'URL. Cette disparité a complexifié l'écriture des services dans Angular, car nous devions constamment adapter les formats d'URL et les méthodes d'appel pour chaque type de paramètre. Ce problème a nécessité une attention particulière pour assurer que chaque endpoint soit correctement adressé et que les données soient transmises de manière fiable entre le front-end et le back-end.

Un autre problème majeur que nous avons rencontré était lié à la création de notre base de données, en particulier avec une table nommée "User". Le mot "User" étant un mot réservé en SQL, cela a entraîné plusieurs complications lors de la création et de la manipulation de cette table. Lors des tests de nos endpoints, nous avons souvent rencontré des erreurs SQL en raison de ce conflit de nom. Pour résoudre ce problème, nous avons dû renommer notre table User en ajoutant des quotes pour montrer que nous ne voulions pas utiliser le mot réservé.

Le reste de nos problèmes apparaissait notamment dans le frontend. En effet, nous étions très à cheval sur l'esthétique de notre application, et nous voulions absolument respecter la maquette Figma établie en amont. Or nous étions considérablement restreintes par la bibliothèque de Bootstrap, car nous ne trouvions pas les composants souhaités. Nous avons également eu quelques problèmes de compréhension entre les padding et margin, que nous avons fini par résoudre. Enfin, un autre problème était de rendre notre application responsive : on remarquait qu'entre notre écran de bureau 1920 x 1080, tout allait bien mais que sur un MacBook Air, ça n'allait plus ! On a essayé au mieux de corriger cela, en remplaçant toutes les valeurs en dures par des pourcentages, même si nous ne sommes pas sûres du rendu sur d'autres résolutions.

# FEEDBACKS

Le cours de Programmation Web a été agréable à suivre tout au long du semestre. Les premières séances étaient néanmoins difficiles à suivre, car il s'agissait essentiellement de cours théoriques, même si nous comprenons que cette partie théorique était nécessaire avant de passer à la pratique avec les TPs.

Concernant les TPs, nous avons trouvé ça très rassurant de pouvoir les faire en cours, pas à pas, tout en suivant les slides et les explications/démonstrations au tableau. Le projet à réaliser était également très similaire aux TPs, ce qui nous a rassurées, étant des étudiantes n'ayant jamais fait de programmation web auparavant. Nous pensons aussi que les délais octroyés pour les TPs et le projet sont bien choisis : nous avons pu les terminer sans trop être pressées par le temps, ce qui est une bonne nouvelle !

Puis, en tant qu'étudiantes n'ayant jamais eu d'expérience en programmation web avant, nous avons apprécié ce cours et notamment le projet de fin de semestre, car nous avons pu monter en compétences, tout en ayant eu du plaisir à programmer. Créer "GAMEPARTY" nous a permis d'avoir plus confiance en nos compétences de programmation web, et nous en sommes fières aujourd'hui !

Enfin, les deux enseignants du cours étaient à l'écoute de nos questions et toujours disponibles. C'était très rassurant de savoir que leur porte était toujours ouverte si jamais nous avions une question à poser !

En conclusion, le cours de Programmation Web était très instructif et intéressant. Cette matière est probablement notre préférée du semestre voire de l'année, car ça change des cours assez théoriques, et pouvoir programmer notre propre application GAMEPARTY, a été une vraie partie de plaisir. Nous avons pu créer et personnaliser entièrement notre application selon nos idées et goûts, ce qui était très fun. Merci beaucoup pour ce cours et vos enseignements !