* **Ton système de panier repose sur les sessions. Que se passe-t-il si je me connecte sur deux appareils en même temps ? Comment gérer un panier persistant ?**

Une session est lié a un navigateur donc la session sera absente sur le 2 pc  
En bdd

* **Ton panier est stocké en session. Est-ce scalable ? Serais-tu capable de passer à une version stockée en base ?**

Non car justement il est stocké en session donc dépendant du pc

Oui en liant le panier a l utilisateur par le biais d une table dédiée.

* **Tu as un CartService, un EmailService, un OrderExportPdfService... Mais pas d’interface. Pourquoi ?**

Quand j ai crée mes services je ne connaissais pas le système d interface. Quand je me suis rendue compte que j avais des services d emails quasi similaires je me suis interrogée sur la possibilité de mieux les organiser. C’est la que j’ai eu connaissance des interfaces. Le projet actuel étant de petite envergure je n’en ai pas besoin mais il serait intéréssant de les mettre en place pour une meilleure évolution du projet

* **Tu utilises DomPDF. Est-ce le meilleur choix ? En connais-tu d’autres ? Et pour une API REST, comment ferais-tu ?**

Il est simple et suffisant pour générer des pdf dans un petit projet. J’ai vu que snappy serait plus puissant.

Je générerais le pdf coté serveur a partir des données json reçues et le retournerais en réponse. Le service rest ne reverrait pas de html mais le pdf directement.  
Dom ou Snappy peut faire ce travail (snappy plus adapté pour api, plus rapide)

* **Tu parles de Doctrine, mais quels types de jointures utilises-tu concrètement (LEFT JOIN, INNER JOIN...) ?**

J’utilise uniquement inner join car j avais besoin de recuperer des données liées entre 2 entités

* **Tu as implémenté un rôle ROOT. Est-ce vraiment utile dans un projet réel ? Quels sont les risques de ce rôle ?**

Oui c’est utile dans un projet réel car il faut bien quelqu’un au dessus des adminstrateurs on aurait pu aussi mettre un gestionnaire des admins.

Etant donné qu il a accès a tout, le compte risque d’etre compromis. Il faudrait si d aventure on offre encore plus de responsabilités protégé d’avantage ce compte.

* **Comment sécurises-tu le back-office ? As-tu mis en place une route protégée ? Un Firewall personnalisé ?**

J ai mis en place des vérification de connexion avec isGranted et également un firewall personnalisé. Les routes nommées backOffice ne sont disponibles uniquement pour les roles d admin et de root a l aide d un firewall.

* **Ton code est-il documenté ? As-tu respecté les PSR (PSR-1/PSR-12) ?**

Mon code possede des commentaires mais il ne sont pas conforme au psr 5 et 12 et n utilise pas le @param ou @return au dessus de la méthode par exemple

L’indentation est conforme au psr 12 avec 4 espaces prévu

1. **Pourquoi avoir utilisé des services pour le panier et l’envoi d’e-mails ? En quoi cela respecte-t-il le principe SOLID ?**

principe de responsabilité unique (Single Responsibility Principle)

Chaque classe ou service a une responsabilité claire et unique.  
Par exemple, le CartService ne gère que les opérations liées au panier, et l’EmailService uniquement l’envoi de mails.

1. **Ton système de panier repose sur une clé composée (exhibitionId\_ticketId). Quels sont les risques de cette méthode ? Que ferais-tu pour la rendre plus robuste ?**

Si un autre critére devrait etre ajouté cela rendrait ma méthode plus fragile.

Je pourrais stocker directement l’objet sous forme de tableau associatif ex exhibitionId => 12

1. **Peux-tu expliquer ce que fait la méthode groupCartByExhibition() dans le détail ? Quels principes POO ou d’algorithmie sont appliqués ?**

La méthode groupCartByExhibition() organise les éléments du panier en les regroupant par exposition (exhibitionId). Concrètement, elle parcourt la liste des articles et crée un tableau où chaque clé correspond à une exposition, contenant les tickets associés. Cela facilite l’affichage et la gestion des données.

En POO, cette méthode améliore la **responsabilité unique** en déléguant la gestion du regroupement au service dédié, respectant ainsi le principe **Single Responsibility** du SOLID.

1. **As-tu utilisé un EventSubscriber ou un EventListener dans ton projet ? Pourquoi, ou pourquoi pas ?**

Mon projet est simple et de petite envergure les services suffisent largement pour répondre aux besoins fonctionnels. Mais il serait en effet intéréssant de mettre en place des interfaces si le projet évolue.

1. **Pourquoi as-tu utilisé Dompdf et non une autre solution comme TCPDF ou API externe ? Quelles sont les limites de Dompdf ?**

Il est simple et suffisant pour générer des pdf dans un petit projet. J’ai vu que snappy serait plus puissant. Une API externe pourrait aussi être envisagée pour déléguer la génération de PDF hors du serveur.

**⚙️ Symfony et Doctrine**

1. **Peux-tu m’expliquer la différence entre QueryBuilder et DQL ? Dans quels cas les as-tu utilisés ?**

DQL (Doctrine Query Language) est un langage de requêtes écrit sous forme de chaînes de caractères qui ressemble au SQL mais pour les entités Doctrine. Le QueryBuilder est une API PHP fluide permettant de construire ces requêtes de façon dynamique et sécurisée sans écrire directement du texte.

Je privilégie le QueryBuilder, car il permet de construire les requêtes DQL de manière dynamique et sécurisée par le biais des requetes preparées.

1. **Comment sont gérées les tables associatives dans Doctrine ? As-tu utilisé des entités intermédiaires ? Pourquoi ?**

Doctrine crée une entité intermédiaire pour les tables associatives, cela permet en plus des clé étrangéres d ajouter des attributs supplémentaires

1. **Tu parles d'un système de rôles (ROLE\_USER, ROLE\_ADMIN, ROLE\_ROOT). Comment sont-ils attribués et gérés dans Symfony ?**

Le role user est attribué par défaut lors de la création d’un compte.

Le role root est attribué directement en base de donnée de facçon manuelle.

Le role admin est attribué uniquement par root qui a les droits nécessaires

1. **Pourquoi utiliser AbstractController ? Quelles méthodes utilitaires cela t’apporte-t-il ?**

AbstractController fournit une base prête à l’emploi pour les contrôleurs Symfony, avec des méthodes utilitaires intégrées.

Render() redirectToRoute() addflash find() …

1. **Comment gères-tu la validation des données dans un FormType ? Peux-tu citer des contraintes spécifiques que tu as utilisées ?**

Par le biais de contraintes assert, j ai utilisé longueur maximale, type de date, regex

**🔐 Sécurité**

1. **Comment le token CSRF est-il généré et vérifié dans Symfony ? Peux-tu me montrer le code dans un formulaire Twig ?**

Il est généré automatiquement lorsqu’il s agit du login regisster et logout.  
Sinon j ai rajouté la csrf protection dans la configuration d option dans les formulaires.

1. **Tu as mis une regex sur le mot de passe. Pourquoi est-ce utile si tu hashes le mot de passe avec bcrypt ?**

La regex met en place une sécurité avant le hashage ce qui devient compliqué de le deviner et le rend plus fort. Le hashage lui permet la sécurité du mot de passe lor de la sauvegarde

1. **Pourquoi as-tu utilisé une protection rel="noopener noreferrer" ? Qu’est-ce que window.opener exactement ?**

Sans noopener la nouvelle page ouvert a la possibilité de communiqué avec window opener. Et un pirate pourrait s’en servir pour modifier le contenu de ma page ou rediriger l utilisateur. Le fait de mettre noopener empeche les pages de communiquer et donc au pirate de modifier la page parente.

1. **Dans ton système d’upload, comment vérifies-tu que l’utilisateur n’envoie pas un fichier malveillant malgré l’extension correcte ?**

 je vérifie aussi le type MIME réel du fichier côté serveur. Je limite la taille des fichiers et je renomme les fichiers pour éviter l’exécution de scripts. Pour encore plus de sécurité il serait judicieux de déplacer mes fichiers dans le dossier asset.