

# DOC Technique

- Création d'un éditeur de sondages.
- Création d'un transfert de sondage pour que les clients puissent répondre.
- Création de deux types d'analyseur de résultats pour abonnement pro et gratuit.
- Supplément

Création d'un éditeur de sondages. (editor.php)

- Possibilité de créer des sondages, de les supprimer et de les partager pour le remplir.
- Possibilité de créer des questions, de les modifier et de les remplir.
- Chaque questions peuvent être optionnel ou non ainsi qu'être à choix multiple ou unique.
- Permet de créer, supprimer ou modifier des réponses à des questions.

-Création d'une page pour répondre aux sondages (viewer.php)

- Les questions et les réponses s'affichent
- Si la question est obligatoire une étoile rouge apparaît à côté de la question
- Si la question est à choix unique les réponses auront des boutons radios sinon des checkboxs
- Une fois les réponses envoyé, si il n'y a pas d'erreur, les réponses sont inscrits dans la DB

Et l'utilisateur est redirigé sur une page de remerciement

-Création de deux types d'analyseur

-Le gratuit : (analytics.php)

Chaque question s'affiche avec le nombre de personne ayant répondu  
Ainsi que chaque réponse avec le nombre de personne ayant choisi cette  
réponse

-Le professionnel : (analyticsPro.php)

Comporte la même chose que le gratuit

Chaque questions possèdent un graphique

Avec la possibilité de changer le type de graphique

-Supplément

-Système d'abonnement

Un script php est lancé sur le serveur linux tous les jours qui vérifie la date  
d'abonnement de tous les utilisateurs, si l'abonnement remonte à un mois  
l'abonnement passe en gratuit et un mail est envoyé à l'utilisateur.

-Hébergement

-Le site est hébergé sur un VPS linux : <https://www.danganronpa-online.games/>

Explication du php :

Editeur :

Chaque question s'affiche via un foreach ainsi que les réponses disponibles

Les boutons s'affichent en même temps

Lorsqu'on appuie sur un bouton pour supprimer, une alertbox s'affiche pour confirmer

Si accepter alors une requête ajax est effectué avec l'id de l'élément

Puis il est supprimé (si c'est une question les réponses aussi et pareil pour le sondage)

Lorsqu'on appuie sur un bouton pour modifier, si c'est une réponse un alertbox avec un prompt d'affiche pour rentrer le texte (si le champ est vide cela renvoie une alertbox avec une erreur) et cela envoie une requête ajax pour modifier la réponse dans la db.

Si c'est une question alors un popup créer via ajax s'affiche, une première requête ajax est envoyé avec l'id de la question à php/ajax.php pour que le popup se créer avec un formulaire, ensuite la data de l'ajax.php est envoyé dans le popup pour que l'utilisateur puisse la voir, une fois les champs remplis (si le champ est vide cela renvoie une alertbox avec une erreur) une seconde requête ajax est envoyé à updateQ.php, qui vérifie si le champ n'est pas vide (cela renvoie une alertbox avec une erreur) et si c'est bon alors la question est modifiée dans la db.

Lorsqu'on appuie sur ajouter, cela revient au même mais pour la question on vérifie aussi le type d'abonnement et s'il lui permet d'en créer une autre alors on l'ajoute dans la db.

Lorsqu'on créer un survey, un alertbox avec un prompt d'affiche pour rentrer le texte (si le champ est vide cela renvoie une alertbox avec une erreur) s'il n'y a pas d'erreur alors une requête ajax est envoyé où l'on vérifie aussi si son abonnement lui permet d'en créer un autre si c'est le cas alors on l'ajoute dans la db sinon l'utilisateur est envoyé sur la page AlertLimit.html

Viewer :

Chaque question s'affiche via un foreach ainsi que les réponses disponibles

Lorsque l'utilisateur envoie ces réponses, l'id du survey est envoyé ensuite on récupère toutes les questions du survey puis toutes les réponses possibles des questions et ensuite les ids des questions avec au moins une réponse et on vérifie si les questions obligatoires sont bien répondues sinon une erreur est envoyée et s'il n'y a pas d'erreur alors on inscrit les résultats dans la db.

Donc on augmente le total de réponse des questions et des réponses.

Analytics :

Chaque question s'affiche avec le nombre de personne ayant répondu via un foreach

Ainsi que chaque réponse avec le nombre de personne ayant choisi cette réponse via un autre foreach

Si c'est la version pro alors on affiche pour chaque question un graphique via un script de canvasjs et un foreach, ainsi qu'un script pour changer le type de graphique via un select