Absolument. Voici un cahier des charges détaillé, rédigé sur la base du document de projet fourni.

# **Cahier des Charges : Sondage Bigscreen**

## **Présentation**

* **Nom du projet :** Sondage pour Bigscreen
* **Date :** 18 juin 2025
* **Auteurs :** [Noms du binôme]

## **1. Analyse Client**

Notre client, **Bigscreen**, est une entreprise spécialisée dans la réalité virtuelle (VR). 1 Elle développe et commercialise une application permettant de simuler un écran de cinéma géant en VR, où les utilisateurs peuvent regarder des films, des émissions de télévision ou jouer à des jeux vidéo, seuls ou à plusieurs. 2

Dans le but de préparer la future version de leur application, Bigscreen souhaite lancer un sondage en ligne pour recueillir les avis et les habitudes de leurs utilisateurs. 3 Notre mission est de prendre en charge la conception et le développement de cette plateforme de sondage. 4

Le projet se décompose en trois interfaces distinctes :

1. **Une page de sondage publique** où les utilisateurs répondent aux questions. 55
2. **Une page de restitution publique** permettant à un utilisateur de revoir les réponses qu'il a soumises. 66
3. **Une section d'administration privée** et sécurisée pour que Bigscreen puisse consulter et analyser l'ensemble des réponses collectées. 7777

## **2. Choix Technologiques**

Nos choix technologiques sont guidés par les contraintes du projet et les meilleures pratiques de développement web.

* **Côté Serveur (Back-end) :**
  + **Framework :** L'utilisation du framework **Laravel** est imposée. 8 Ce choix est pertinent pour sa robustesse, sa sécurité intégrée et son écosystème complet.
  + **Base de Données :** Nous utiliserons un système de gestion de base de données relationnelle (type MySQL)9. Les interactions avec la base de données seront gérées exclusivement via l'**ORM Eloquent** de Laravel pour la sécurité et la simplicité du code. 10
  + **Gestion de la BDD :** La structure de la base de données sera définie et versionnée grâce au système de **migrations** de Laravel, et des **seeders** seront créés pour initialiser les données de base (comme les questions du sondage). 11
* **Côté Client (Front-end) :**
  + **Fondations :** HTML5, CSS3, JavaScript.
  + **Interactivité et graphiques :** Pour l'affichage des graphiques sur le tableau de bord de l'administration, nous utiliserons la librairie **Chart.js** comme suggéré12, ou une alternative équivalente si nécessaire.
  + **Framework (Optionnel) :** Nous avons la liberté d'utiliser des frameworks front-end. 13 Le choix se portera sur une solution légère (comme Alpine.js ou Vue.js) si la complexité des interactions le justifie.
* **Outils de Développement :**
  + **Versionning :** Le code source sera versionné avec Git et hébergé sur **GitHub**, avec des commits réguliers de la part des deux membres du binôme. 14

## **3. Adresse Github**

L'ensemble du code source du projet sera disponible à l'adresse suivante :

https://github.com/[NomUtilisateur1]/[NomUtilisateur2]/bigscreen-survey

## **4. Déploiement**

Pour déployer le projet à partir du code source, un développeur devra suivre les étapes suivantes : 15

1. **Prérequis Serveur :**
   * PHP (version compatible avec le projet Laravel)
   * Serveur web (Nginx, Apache)
   * Composer 2.x
   * Node.js & NPM
   * Base de données MySQL
2. **Procédure d'Installation :**
   * Cloner le dépôt Github : git clone [URL\_DU\_PROJET]
   * Se déplacer dans le dossier du projet : cd bigscreen-survey
   * Installer les dépendances PHP : composer install --no-dev
   * Installer les dépendances front-end : npm install && npm run build
   * Créer le fichier d'environnement : cp .env.example .env
   * Configurer les variables d'environnement dans le fichier .env, notamment les accès à la base de données (DB\_DATABASE, DB\_USERNAME, DB\_PASSWORD).
   * Générer la clé d'application Laravel : php artisan key:generate
   * Lancer les migrations et les seeders pour créer les tables et les remplir avec les questions initiales : php artisan migrate --seed
   * Configurer le serveur web pour pointer vers le dossier /public du projet.
   * Assurer les bonnes permissions sur les dossiers storage et bootstrap/cache.

## **5. Méthode de Travail**

* **Méthodologie :** Nous adopterons une méthodologie **Agile**. Le projet sera découpé en petites itérations (sprints) axées sur le développement de fonctionnalités précises (ex: sprint 1 : base de données et authentification admin ; sprint 2 : formulaire de sondage, etc.). Cette approche nous permettra d'avoir un produit fonctionnel rapidement et de l'ajuster au fur et à mesure.
* **Travail en binôme :** Conformément aux exigences, les deux membres de l'équipe travailleront à la fois sur le back-end et le front-end. 16
* **Process de commit :** Chaque commit sur Github suivra une convention claire (ex: feat: add survey form validation, fix: correct chart display). 17 Chaque fonctionnalité majeure sera développée sur une branche dédiée avant d'être fusionnée dans la branche principale.

## **6. Outils Utilisés**

* **Logiciels :**
  + Visual Studio Code (Éditeur de code)
  + MySQL Workbench (ou équivalent, pour la modélisation de la base de données) 18
  + Navigateurs web (Chrome, Firefox) pour les tests
* **Outils de développement :**
  + Git & GitHub
  + Composer
  + Node.js (NPM)
  + Terminal (ou invite de commandes)
* **Services en ligne :**
  + GitHub

## **7. Évaluation du Temps de Travail**

| **Poste de Développement** | **Estimation (jours/homme)** | **Estimation (heures)** |
| --- | --- | --- |
| Conception et Cahier des Charges | 2 | 14 |
| Structure de la base de données (migrations, seeders) | 1.5 | 10.5 |
| Développement Back-end (Logique, Routes, Contrôleurs) | 4 | 28 |
| Développement de l'Administration (Login, CRUD, Sécurité) | 3 | 21 |
| Développement Front-end (Sondage, Page réponses) | 3 | 21 |
| Intégration des graphiques (Dashboard Admin) | 1.5 | 10.5 |
| Tests, Recettage et Débuggage | 2 | 14 |
| Préparation des livrables et de la soutenance | 2 | 14 |
| **Total** | **19** | **133** |

## **8. Liste Fonctionnelle**

### **Interface Publique**

* **Page Sondage :**
  + [F-01] Afficher une page unique avec les 20 questions du sondage. 19
  + [F-02] Chaque question doit afficher son numéro (ex: "Question 1/20"). 20
  + [F-03] Afficher les questions de type A avec des choix uniques (boutons radio).
  + [F-04] Afficher les questions de type B avec un champ de texte (limité à 255 caractères).
  + [F-05] Afficher les questions de type C avec un choix numérique de 1 à 5.
  + [F-06] Rendre la réponse à toutes les questions obligatoire. 21
  + [F-07] Appliquer une validation de format email uniquement pour la question 1. 22
  + [F-08] Lors de la soumission, enregistrer les 20 réponses en base de données.
  + [F-09] Après soumission, afficher un message de remerciement de la part de Bigscreen. 23
  + [F-10] Après soumission, générer et afficher une URL unique et sécurisée pour que l'utilisateur puisse consulter ses réponses. 24
* **Page Réponses :**
  + [F-11] Créer une page accessible via l'URL unique.
  + [F-12] L'URL ne doit pas contenir un identifiant simple (ex: ID numérique).
  + [F-13] Afficher la liste des 20 questions avec les réponses fournies par l'utilisateur. 25
  + [F-14] Afficher la date et l'heure de la soumission du sondage. 26

### **Interface d'Administration**

* **Accès et Sécurité :**
  + [A-01] Mettre en place une page de connexion accessible via l'URL /administration. 27
  + [A-02] Protéger l'ensemble de la section administration par une authentification (login/mot de passe). 28
  + [A-03] Gérer un administrateur unique.
* **Mise en Page :**
  + [A-04] Créer une mise en page fixe avec une barre de navigation latérale (menu) et une zone de contenu principal. 29
  + [A-05] La zone de contenu doit être scrollable verticalement si le contenu dépasse la hauteur de l'écran. 30
  + [A-06] Le menu de navigation doit contenir les liens : "Accueil", "Questionnaire", "Réponses".
* **Page d'Accueil (Dashboard) :**
  + [A-07] Afficher 3 graphiques de type "Pie chart" pour les questions 6 (marque casque), 7 (magasin d'application) et 10 (utilisation principale). 31
  + [A-08] Afficher 1 graphique de type "Radar chart" regroupant les données des questions 11 à 15 (notes de qualité). 32
* **Page Questionnaire :**
  + [A-09] Afficher la liste des 20 questions du sondage dans un tableau. 33
  + [A-10] Le tableau doit contenir 3 colonnes : Numéro, Corps de la question, Type (A, B, C). 34
  + [A-11] Les questions ne sont pas modifiables, ajoutables ou supprimables depuis cette interface. 35
* **Page Réponses :**
  + [A-12] Afficher l'ensemble des réponses soumises par tous les utilisateurs. 36
  + [A-13] Regrouper les réponses par utilisateur, chaque groupe étant présenté dans un tableau distinct. 37
  + [A-14] Chaque tableau doit contenir 3 colonnes : Numéro de la question, Corps de la question, Réponse de l'utilisateur. 38

## **9. Recettage**

| **ID** | **Fonctionnalité** | **Statut** |
| --- | --- | --- |
| **Partie Publique** |
| F-01 | Affichage page sondage | À développer |
| F-02 | Numérotation des questions | À développer |
| F-03 | Affichage questions type A | À développer |
| F-04 | Affichage questions type B | À développer |
| F-05 | Affichage questions type C | À développer |
| F-06 | Validation : toutes questions obligatoires | À développer |
| F-07 | Validation : format email | À développer |
| F-08 | Enregistrement des réponses | À développer |
| F-09 | Message de remerciement | À développer |
| F-10 | Génération URL unique | À développer |
| F-11 | Accès page réponses via URL | À développer |
| F-12 | Sécurité de l'URL réponses | À développer |
| F-13 | Affichage des réponses utilisateur | À développer |
| F-14 | Affichage date de soumission | À développer |
| **Partie Administration** |
| A-01 | Page de connexion admin | À développer |
| A-02 | Protection par authentification | À développer |
| A-03 | Gestion administrateur unique | À développer |
| A-04 | Mise en page fixe admin | À développer |
| A-05 | Scroll vertical du contenu | À développer |
| A-06 | Menu de navigation admin | À développer |
| A-07 | Affichage des 3 Pie charts | À développer |
| A-08 | Affichage du Radar chart | À développer |
| A-09 | Affichage tableau des questions | À développer |
| A-10 | Colonnes du tableau questions | À développer |
| A-11 | Non-administration des questions | À développer |
| A-12 | Affichage de toutes les réponses | À développer |
| A-13 | Regroupement des réponses par utilisateur | À développer |
| A-14 | Colonnes du tableau des réponses | À développer |

## **10. Diagramme de la Base de Données**

Le schéma de la base de données s'articulera autour de 4 tables principales pour assurer la flexibilité et l'intégrité des données. 39

* **users** (Pour l'administrateur)
  + id (Clé primaire, auto-incrément)
  + name (VARCHAR)
  + email (VARCHAR, unique)
  + password (VARCHAR)
  + created\_at, updated\_at (TIMESTAMPS)
* **questions** (Stockage des questions du sondage)
  + id (Clé primaire, auto-incrément)
  + title (VARCHAR) - Ex: "Votre adresse mail"
  + type (ENUM['A', 'B', 'C']) - Le type de question.
  + options (JSON, nullable) - Pour stocker les choix des questions de type A.
  + order (INTEGER) - Pour maintenir l'ordre des questions (1 à 20).
* **submissions** (Chaque entrée représente un sondage complété par un utilisateur)
  + id (Clé primaire, auto-incrément)
  + url\_token (VARCHAR, unique, indexé) - Le jeton sécurisé pour l'URL de consultation.
  + created\_at, updated\_at (TIMESTAMPS)
* **answers** (Stockage de chaque réponse individuelle)
  + id (Clé primaire, auto-incrément)
  + submission\_id (Clé étrangère vers submissions.id)
  + question\_id (Clé étrangère vers questions.id)
  + value (TEXT) - La réponse de l'utilisateur.

**Relations :**

* Une submission a plusieurs answers (Relation 1-N).
* Une question peut avoir plusieurs answers (Relation 1-N).

## **11. Wireframes**

Les wireframes décrivent la structure visuelle et la disposition des éléments sur les pages clés du projet. 40

* **Page Sondage :**
  + En-tête avec le logo Bigscreen.
  + Titre : "Sondage Bigscreen".
  + Introduction : "Merci de répondre à toutes les questions..." 41
  + Liste verticale des 20 questions.
  + Chaque question est dans un bloc distinct avec :
    - Titre (ex: "Question 1/20").
    - Intitulé de la question.
    - Zone de réponse (boutons radio, champ texte ou échelle numérique).
  + En bas de page, un bouton "Finaliser". 42
* **Page Réponses Utilisateur :**
  + En-tête avec le logo Bigscreen.
  + Titre : "Vos réponses au sondage Bigscreen".
  + Texte d'introduction avec la date de soumission. 43
  + Liste verticale des 20 questions.
  + Chaque question est dans un bloc distinct avec :
    - Titre (ex: "Question 1/20").
    - Intitulé de la question.
    - La réponse de l'utilisateur en texte simple dans un encadré. 44
* **Page d'Administration :**
  + Structure en deux colonnes. 45
  + **Colonne de gauche (fixe) :**
    - Logo.
    - Menu de navigation vertical avec les liens "Accueil", "Questionnaire", "Réponses".
  + **Colonne de droite (scrollable) :**
    - **Accueil :** Titre "Statistiques", suivi des 4 blocs de graphiques.
    - **Questionnaire :** Titre "Liste des questions", suivi du tableau des 20 questions.
    - **Réponses :** Titre "Résultats du sondage", suivi de la liste verticale des tableaux de réponses par utilisateur.

## **12. Documentation du Code**

La documentation du code sera intégrée directement dans les fichiers sources sous forme de commentaires et synthétisée dans ce document. 46

### **Back-end (Laravel)**

* **Contrôleurs :** Pour chaque méthode publique, une description de son rôle, de ses paramètres et de ce qu'elle retourne sera fournie.
* **Modèles (Eloquent) :** Chaque modèle décrira la table associée, ses champs ($fillable), et les relations avec d'autres modèles (ex: hasMany, belongsTo).
* **Migrations :** Chaque fichier de migration décrira les champs de la table qu'il crée ou modifie, incluant le nom du champ, son type et ses contraintes (nullable, unique, etc.).
* **Routes :** Le fichier de routes (web.php) sera commenté pour expliquer le rôle de chaque route et le middleware de protection appliqué.

### **Front-end**

* **Composants (si applicable) :** Si des composants (ex: Vue.js) sont utilisés, chacun sera documenté pour décrire son rôle, les props qu'il accepte et les événements qu'il émet.
* **Librairies Tierces :** La documentation inclura la liste des librairies front-end utilisées (ex: Chart.js) et la raison de leur utilisation.
* **Scripts JavaScript :** Les fonctions JavaScript complexes seront commentées pour expliquer leur logique, leurs paramètres et leur valeur de retour.