**Bigscreen**

Cahier des charges

Nom du projet: Bigscreen

Date : 12/09/2022 au 10/10/2022

Noms: FUSTIER Julien et MARCEAU Cyril

Analyse Client

Bigscreen est une entreprise travaillant dans la VR qui est un secteur en forte expansion et à besoin d’avoir un outil qui leur permettrait de mieux cibler leur clientèle d’où la réalisation d’une application de sondage leur permettant de connaître leur attente.

Choix technologiques

* Back-end: Concernant le back-end nous avons développé une API rest avec laravel avec l’ORM eloquent.
* Pour le système d'authentification nous avons utilisé “laravel sanctum”. Un système natif à laravel qui est compatible avec les SPA (Single page application).
* Front-end: Pour le front nous avons fait le choix d’utiliser [next.js](https://nextjs.org/) car ce framework front permet d’intégrer plus facilement les différentes routes de l’application, des middlewares ainsi que des gains de performance et de sécurité comparé à une application react.js ou l'on dois tout développer à la main.
* Pour les appels serveur côté front nous avons utilisé la librairie axios, un client HTTP basé sur les promesses. C’est une librairie simple à utiliser et qui permet de simplifier grandement le système des appels serveur.
* Pour l’intégration visuelle nous avons utilisé la librairie graphique “Ant Design créé pour react avec une bonne qualité d'expérience utilisateur et un design sobre et efficace.
* Pour l’affichage des graphiques côté front nous avons utilisé la librairie chart.js, celle-ci est simple d’utilisation et se base sur un design clair et efficace

Github

<https://github.com/cyrilmarceau/Bigscreen>

Déploiement

Changer les variables d’environnements (api/.env):

* (DB\_DATABASE, DB\_USERNAME et DB\_PASSWORD)

Installation des dépendances:

* $ cd api && composer install
* $ cd admin && npm i

Lancer les migrations:

* $ php artisan migrate:fresh –seed

Lancer le serveur de développement

* $ cd api / php artisan serve
* $ cd front / npm run dev

Identifiant de connexion

* Email: [admin@admin.fr](mailto:admin@admin.fr)
* Mot de passe: password

Méthode de travail

Nous utiliserons la méthodologie agile car elle permet de s’adapter à la charge de travail en délivrant un travail fonctionnel en continue et permet de mieux diviser le travail étape par étape.

Pour gérer le versionning nous utilisons la méthodologie git flow. Nous partons d’une branche main qui contient le code finalisé. A partir de celle-ci nous créons une branche par tâche de travail. Lorsque nous avons terminé la fonctionnalité du code nous faisons un merge dans la branche main. Ceci nous permet de travailler en équipe de façon efficace et structurée.

Nous utiliserons la convention de commit (détail [ici](https://www.conventionalcommits.org/en/v1.0.0/)) qui permet d’avoir une vue globale et simple des différents commits réalisés durant ce projet.

Outils utilisés

Pour l’écriture du code nous avons utilisé visual studio code, il s’agit d’un des éditeurs de texte les plus utilisé pour la programmation, spécialement adapté pour l’écriture de code et très riche grâce à son vaste système d'extension.

Afin de tester les différents appels API nous avons utilisé l’outil insomnia afin de pouvoir vérifier que nos requêtes fonctionnent correctement et renvoient les bonnes données.

Pour le versionning nous utilisons [github](https://github.com/cyrilmarceau/Bigscreen).

Concernant les différentes fonctionnalités du projet nous avons utilisés [jira](https://www.atlassian.com/fr/software/jira) qui permet d’avoir un aperçus des différents tickets à traiter durant le projet:

[**https://cyrilmarceau.atlassian.net/jira/software/projects/BIG/boards/2**](https://cyrilmarceau.atlassian.net/jira/software/projects/BIG/boards/2)

Pour la mise en forme du schéma de la base de données nous utiliserons DB diagram. Un outil spécialement adapté pour les développeurs qui permet de relier le code avec la représentation visuelle des tables.

Pour l’édition des documents de travail (cahiers des charges et slides de présentation orale) nous avons utilisé les outils google docs et google slides. Ils ont l’avantage d’être gratuits tout en étant complet et adapté au travail en équipe.

Pour le travail à distance nous avons utilisé discord, un logiciel de communication gratuit et très complet notamment par son système de partage d’écran sophistiqué.

Evaluation du temps de travail

| **Entrée** | **Sortie (jours)** |
| --- | --- |
| API - Migrations | 0.5 |
| API - Seeders + factories | 0.5 |
| API - Récupérer la liste des sondages | 0.1 |
| API - Récupérer 1 sondage | 0.1 |
| API - Récupérer la liste des questions | 0.1 |
| API - Authentification | 0.2 |
| API - Création d’un sondage | 0.3 |
| API - Graphique | 0.3 |
| FRONT - Layout | 0.2 |
| FRONT - Page authentification | 0.2 |
| FRONT - API Factory | 0.2 |
| FRONT - Formulaire des questions | 0.4 |
| FRONT - Afficher les questions | 0.1 |
| FRONT - Afficher les réponses d’un sondé | 0.2 |
| FRONT - Route protégé | 0.2 |

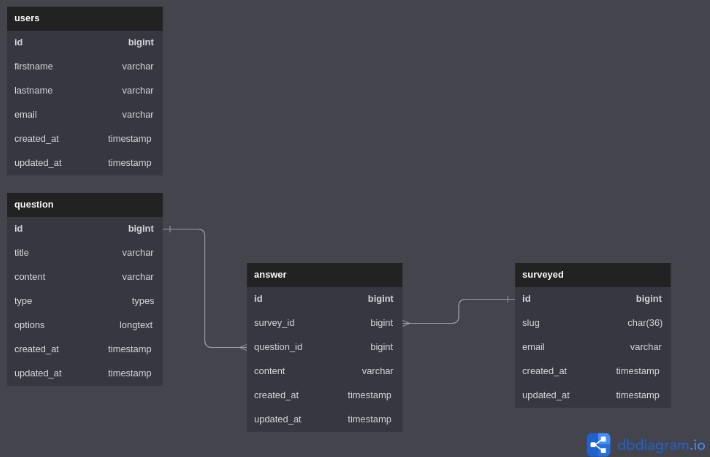
Liste fonctionnelle

| **FRONT** | **API** |
| --- | --- |
| FRONT - Layout | API - Migrations |
| FRONT - Page authentification | API - Seeders + factories |
| FRONT - API Factory | API - Récupérer la liste des sondages |
| FRONT - Formulaire des questions | API - Récupérer 1 sondage |
| FRONT - Afficher les questions | API - Récupérer la liste des questions |
| FRONT - Afficher les sondages | API - Authentification |
| FRONT - Afficher les réponses d’un sondé | API - Création d’un sondage |
| FRONT - Affichage graphique | API - Graphique |
| FRONT - Message confirmation sondage |  |
| FRONT - Affichage liste sondage |  |
| FRONT - Route protégé |  |

Recettage

| **Fonctionnalité** | **Status** |
| --- | --- |
| API - Migrations | Fonctionnelle |
| API - Seeders + factories | Fonctionnelle |
| API - Récupérer la liste des sondages | Fonctionnelle |
| API - Récupérer 1 sondage | Fonctionnelle |
| API - Récupérer la liste des questions | Fonctionnelle |
| API - Authentification | Fonctionnelle |
| API - Création d’un sondage | Fonctionnelle |
| API - Graphique | Fonctionnelle |
| FRONT - Layout | Fonctionnelle |
| FRONT - Page authentification | Fonctionnelle |
| FRONT - API Factory | Fonctionnelle |
| FRONT - Formulaire des questions | Fonctionnelle |
| FRONT - Afficher les questions | Fonctionnelle |
| FRONT - Afficher les réponses d’un sondé | Fonctionnelle |
| FRONT - Affichage graphique | Fonctionnelle |
| FRONT - Message confirmation sondage | Fonctionnelle |
| FRONT - Affichage liste sondage | Fonctionnelle |
| FRONT - Route protégé | Fonctionnelle |

Base de données:

****

**Table User:**

Informations concernant l’utilisateur admin

**Table Question:**

Contient les différentes questions définie au préalable

**Table Answer:**

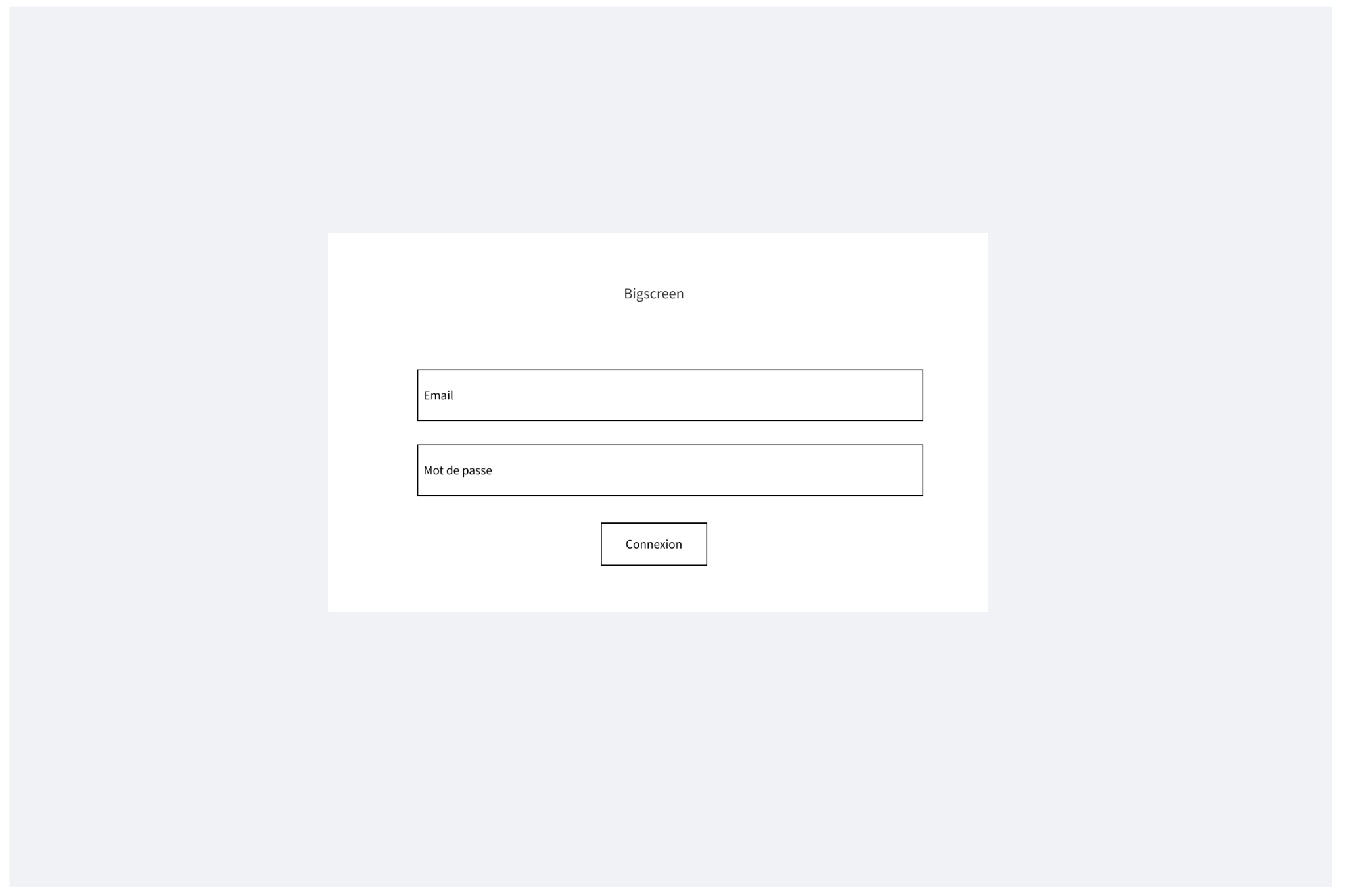
Contient les réponses et la relation vers la question

**Table Surveyed:**

Contient les différents sondé et la relations vers les réponses

Wireframe - administration

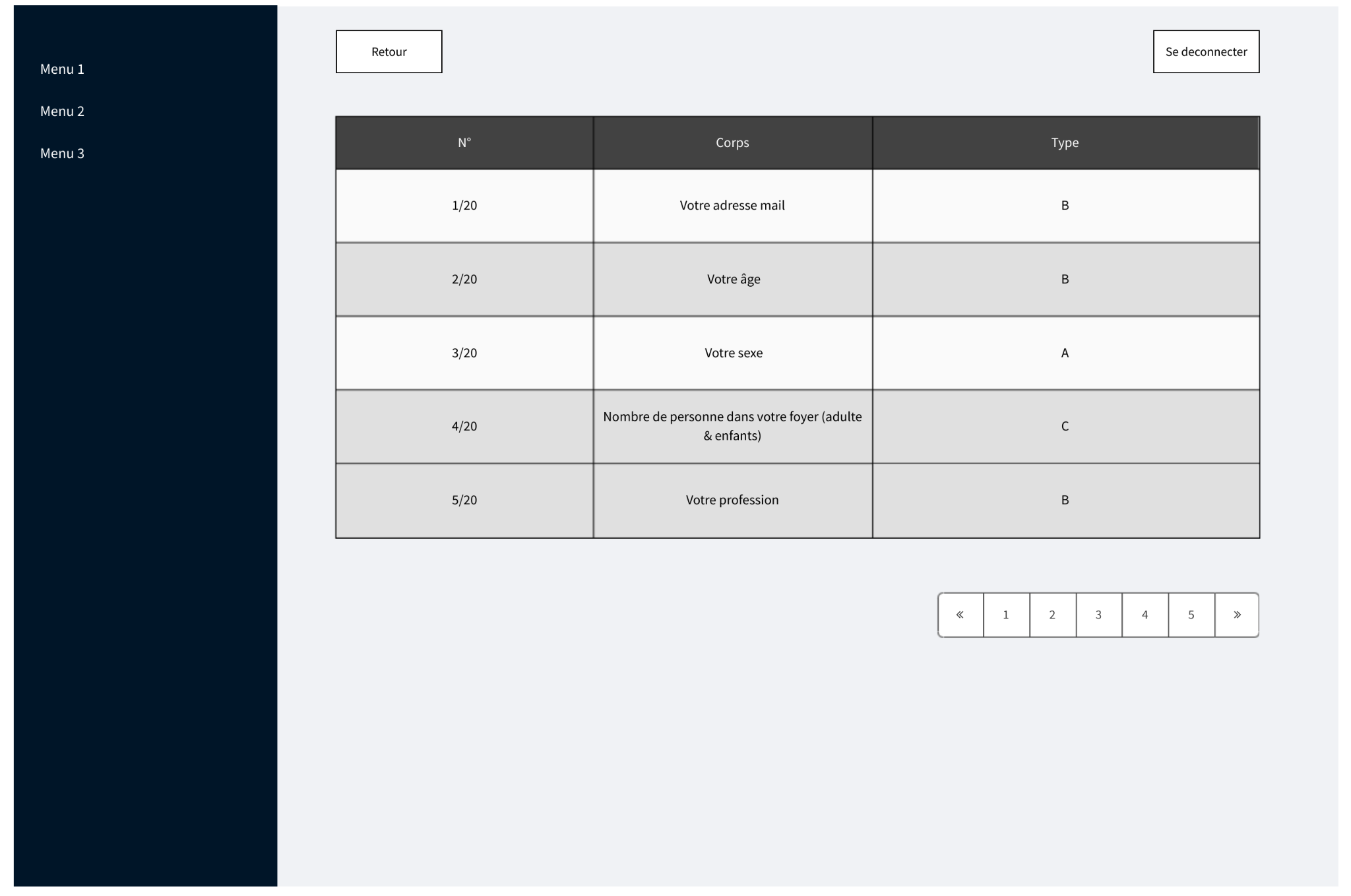
***Page d’authentification***

****

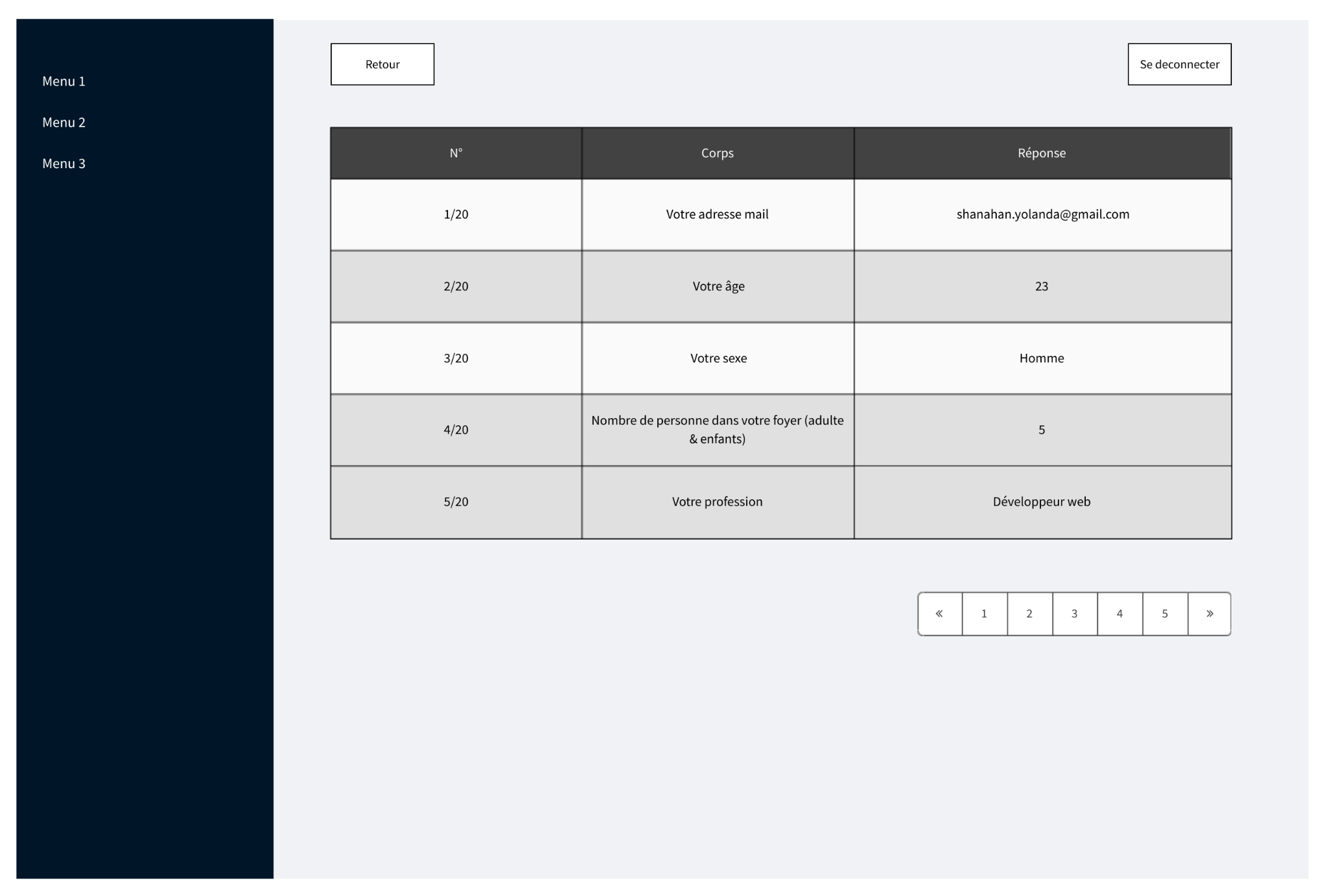
***Page d’accueil***

****

**Liste des questions**

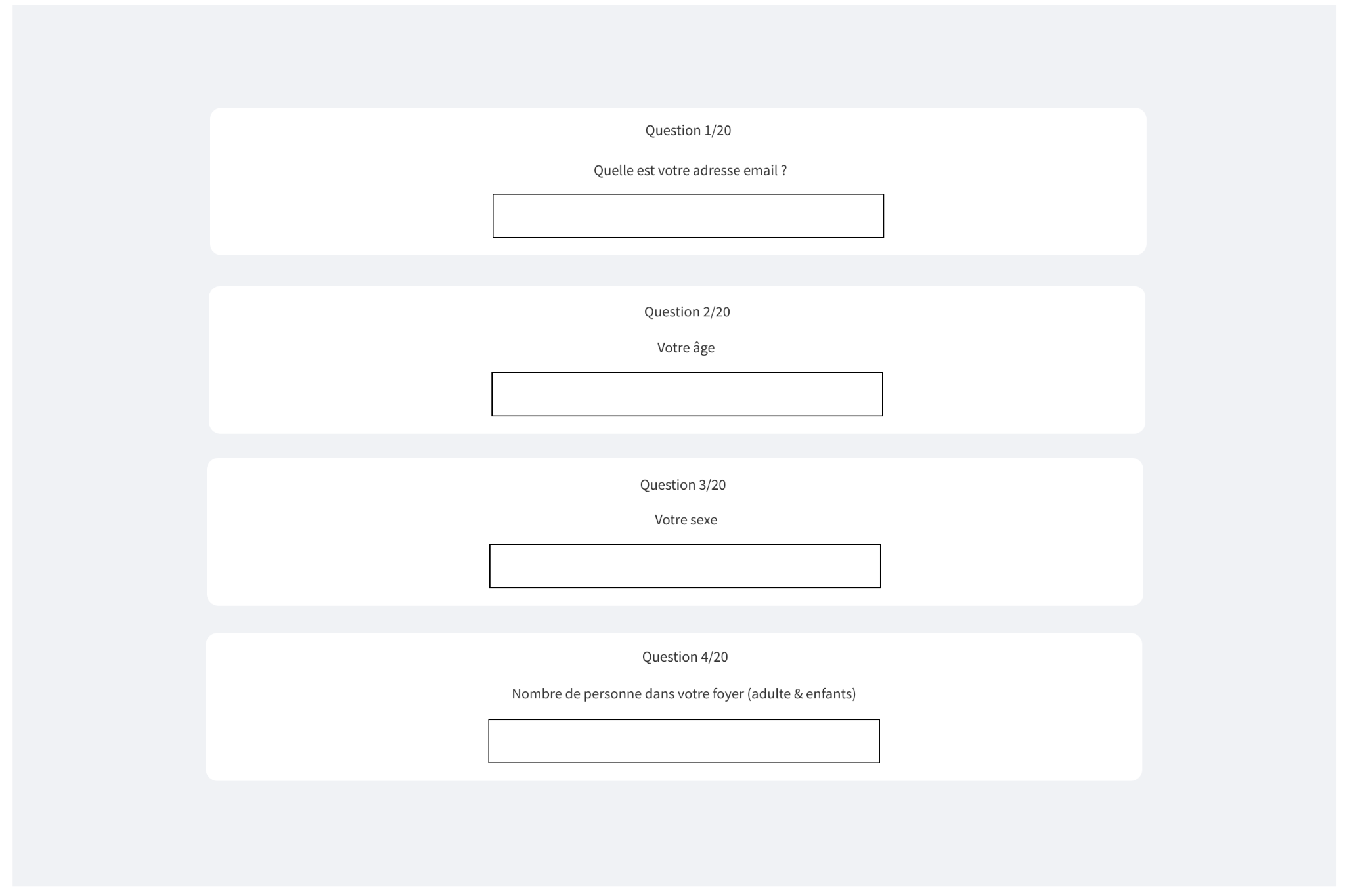
**

***Liste des réponses***

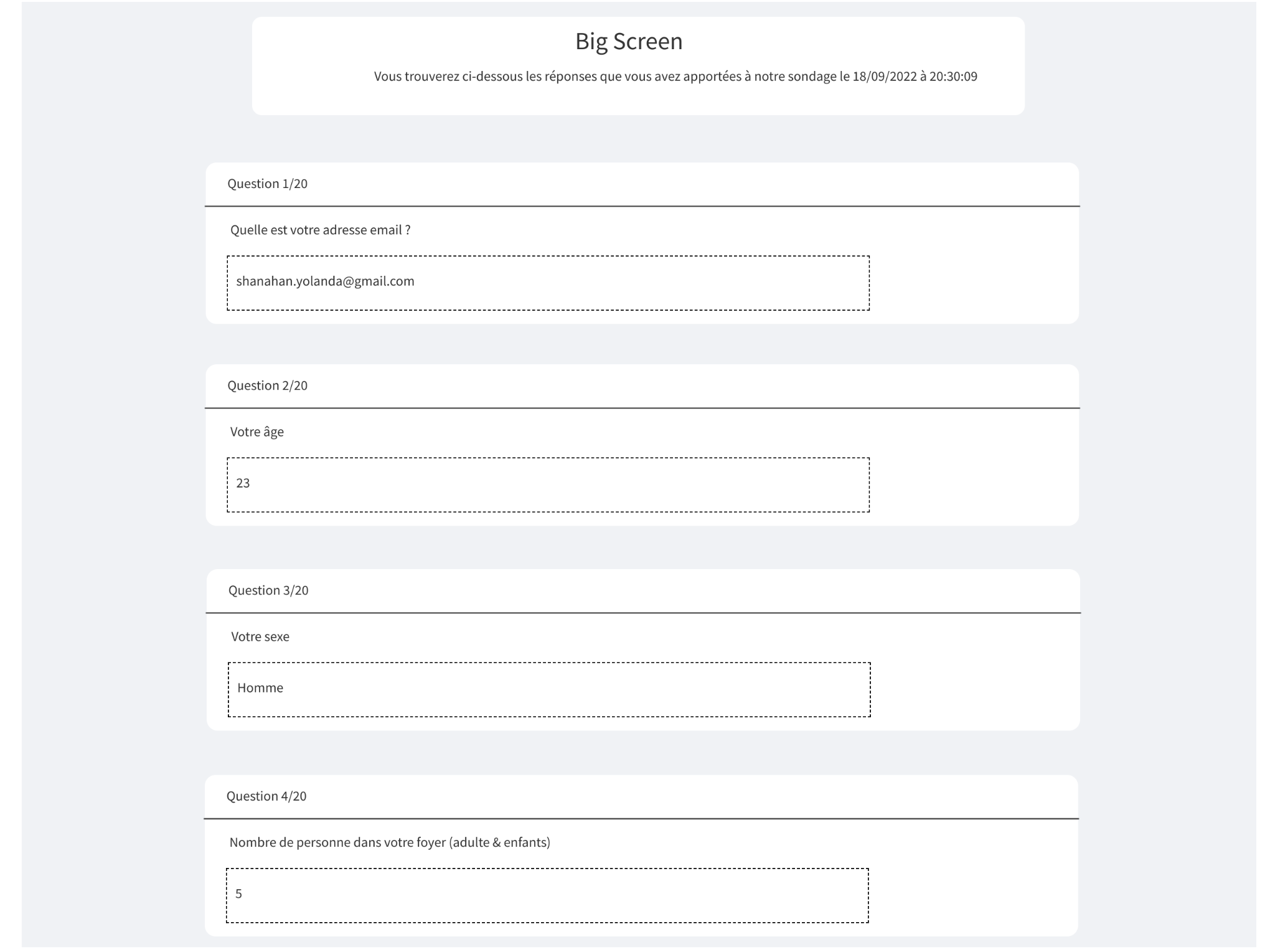
**

Wireframe - client

***Questionnaire***

**

***Réponses d’un sondé***

**

DOCUMENTATION API - ADMIN

Se connecter

***url***: <http://127.0.0.1:8000/api/login>

***type:*** POST

***Paramètres***:

{

"email": string,

"password": string,

}

200

{

"success": true,

"data": {

"token": "17|X4vufTR9YdlFgvY8jyqy3MrWLJ5M1laIkNZH4EyU",

"name": "Oliver Greenholt"

},

"message": "Bienvenue Oliver Greenholt"

}

404

{

"success": false,

"message": "Identifiant incorrect",

"data": {

"error": "unauthorised"

}

}

Se déconnecter

***url***: [http://127.0.0.1:8000/api/logout](http://127.0.0.1:8000/api/login)

***type:*** POST

***header***:

* Authorization: Bearer token
* Accept: application/json

200

{

"success": true,

"data": {

"data": null

},

"message": "Utilisateur déconnecté"

}

401

{

"message": "Unauthenticated."

}

Récupérer la liste des sondés

***url***: <http://127.0.0.1:8000/api/admin/surveyeds>

***type:*** GET

***header***:

* Authorization: Bearer token
* Accept: application/json

200

{

"success": true,

"data": [

{

"id": 1,

"slug": "444aa4e1-78a8-38c3-8535-43fcba8a3694",

"email": "kimberly59@gmail.com",

"created\_at": "2022-09-18T18:30:09.000000Z",

"updated\_at": "2022-09-18T18:30:09.000000Z",

"answers": [

{

"id": 1,

"content": "shanahan.yolanda@gmail.com",

"created\_at": "2022-09-18T18:30:09.000000Z",

"updated\_at": "2022-09-18T18:30:09.000000Z",

"question": {

"id": 1,

"title": "1/20",

"content": "Votre adresse mail",

"type": "B",

"options": null,

"created\_at": "2022-09-18T18:30:09.000000Z",

"updated\_at": "2022-09-18T18:30:09.000000Z"

}

},

],

"message": "Liste des sondés récupérés avec succès"

}

401

{

"message": "Unauthenticated."

}

Récupérer la liste des questions

***url***: <http://127.0.0.1:8000/api/admin/questions>

***header***:

* Authorization: Bearer token
* Accept: application/json

***type:*** GET

200

**{**

**"success": true,**

**"data": [**

**{**

**"id": 1,**

**"title": "1/20",**

**"content": "Votre adresse mail",**

**"type": "B",**

**"created\_at": "2022-09-18T18:30:09.000000Z",**

**"updated\_at": "2022-09-18T18:30:09.000000Z"**

**},**

**{**

**"id": 2,**

**"title": "2/20",**

**"content": "Votre âge",**

**"type": "B",**

**"created\_at": "2022-09-18T18:30:09.000000Z",**

**"updated\_at": "2022-09-18T18:30:09.000000Z"**

**}**

**],**

**"message": "Liste des questions récupérer avec succès"**

**}**

401

{

"message": "Unauthenticated."

}

Récupérer les statistiques pour le graphique

***url***: <http://127.0.0.1:8000/api/admin/charts>

***header***:

* Authorization: Bearer token
* Accept: application/json

***type:*** GET

200

**{**

**"success": true,**

**"data": [**

**{**

**"content": "Quel marque de casque VR utilisez vous ?",**

**"stats": {**

**"label": "Windows Mixed Reality",**

**"count": 1**

**},**

**"type": "pie"**

**},**

**{**

**"content": "Sur quel magasin d’application achetez vous des contenus VR ?",**

**"stats": {**

**"label": "Google Play",**

**"count": 1**

**},**

**"type": "pie"**

**},**

**{**

**"stats": [**

**{**

**"label": "Audio",**

**"count": 3.6**

**}**

**],**

**"type": "radar"**

**}**

**],**

**"message": null**

**}**

401

{

"message": "Unauthenticated."

}

DOCUMENTATION API - CLIENT

Récupérer la liste des questions

***url***: [http://127.0.0.1:8000/api/client/questions](http://127.0.0.1:8000/api/admin/charts)

***header***:

* Accept: application/json

***type:*** GET

200

**{**

**"success": true,**

**"data": [**

**{**

**"id": 1,**

**"title": "1/20",**

**"content": "Votre adresse mail",**

**"type": "B",**

**"created\_at": "2022-09-18T18:30:09.000000Z",**

**"updated\_at": "2022-09-18T18:30:09.000000Z"**

**},**

**{**

**"id": 2,**

**"title": "2/20",**

**"content": "Votre âge",**

**"type": "B",**

**"created\_at": "2022-09-18T18:30:09.000000Z",**

**"updated\_at": "2022-09-18T18:30:09.000000Z"**

**}**

**],**

**"message": "Liste des questions récupérer avec succès"**

**}**

Récupérer un sondage

***url***: [http://127.0.0.1:8000/api/client/surveyeds/](http://127.0.0.1:8000/api/surveyeds/c9de43b5-09e5-3391-a320-7dfc46d4608c)<uuid>

***header***:

* Accept: application/json

***type:*** GET

200

**{**

**"success": true,**

**"data": {**

**"id": 9,**

**"slug": "c9de43b5-09e5-3391-a320-7dfc46d4608c",**

**"email": "jruegckefr@hotmail.com",**

**"created\_at": "2022-09-16T09:36:25.000000Z",**

**"updated\_at": "2022-09-16T09:36:25.000000Z",**

**"answers": [**

**{**

**"id": 140,**

**"content": "32",**

**"created\_at": "2022-09-16T09:36:25.000000Z",**

**"updated\_at": "2022-09-16T09:36:25.000000Z",**

**"question": {**

**"id": 2,**

**"title": "2/20",**

**"content": "Votre âge",**

**"type": "B",**

**"options": null,**

**"created\_at": "2022-09-15T15:29:25.000000Z",**

**"updated\_at": "2022-09-15T15:29:25.000000Z"**

**}**

**}**

**]**

**},**

**"message": "Sondé récupéré avec succès"**

**}**

404

{

"success": false,

"message": "Aucun sondé n'a été trouvé"

}

Créer un sondage

***url***: <http://127.0.0.1:8000/api/client/surveyeds>

***header***:

* Content-Type: application/json

***type***: POST

***Paramètre***:

{

"email": string,

"questions": [{

"questionId": int<1-20>,

"content": string | int

}]

}

200

**{**

**"success": true,**

**"data": {**

**"slug": "82ab0e7d-157d-380e-b417-a37e965778f7",**

**"email": "test.fr@gmail.com",**

**"updated\_at": "2022-09-25T16:10:32.000000Z",**

**"created\_at": "2022-09-25T16:10:32.000000Z",**

**"id": 6**

**},**

**"message": "Sondé crée avec succès"**

**}**

400

{

"success": false,

"message": "Erreur de validation du formulaire",

"data": {

"email": [

"L'email est obligatoire"

]

}

}

400

{

"success": false,

"message": "Erreur de validation du formulaire",

"data": {

"email": [

"Vous devez correctement formater votre email"

]

}

}

422

**{**

**"success": false,**

**"message": "**Erreur enregistrement: Vous avez déjà répondu au sondage avec une adresse email identique**"**

**}**