

PLAN DE BATAILLE : MODULE 6 (DASHBOARD REACT)

À : [Nom du Coéquipier]

De : [Votre Nom]

Date : 17 Novembre 2025

Objet : Instructions Finales pour l'Intégration du Dashboard Média-Scan

Salut [Nom du coéquipier],

Voici le plan d'intégration final pour notre projet Média-Scan. C'est le dernier jour, donc nous devons être efficaces et précis.

1. Notre Architecture (Approche Frugale)

Pour gagner du temps et garantir une démo stable, nous n'allons PAS construire une API Python backend (Pas de FastAPI, pas de Node.js).

À la place, mes scripts Python (Modules 2, 3, 4, 5) vont générer 5 fichiers JSON statiques.

Ton application React n'aura qu'à lire (fetch) ces 5 fichiers JSON comme s'ils étaient une API. Tu peux les placer dans ton dossier public/data/ de React pour y accéder facilement (ex: `fetch('/data/influence_ranking.json')`).

C'est une approche 100% frugale (critère d'évaluation), robuste, et ultra-rapide.

2. Les 5 Fichiers de Données (Le Contrat d'Intégration)

Je m'occupe de générer les 5 fichiers suivants. Ils seront dans un dossier /output/ que je te transmettrai. Tu devras les placer dans /public/data/ dans ton projet React.

A. influence_ranking.json

•

Contient : Le classement final des médias. C'est une liste d'objets.

•

•

Structure (exemple) :

•

JSON

•

[

{

 "media": "Lefaso.net",
 "score_influence_total": 85.2,
 "audience_score": 28.5,
 "engagement_score": 38.0,
 "regularity_score": 15.7,
 "diversity_score": 3.0,
 "audience_raw": 500000,
 "engagement_raw": 150.5,
 "regularity_raw": 12.0,
 "diversity_raw": 3

},

{

 "media": "Sidwaya",
 "score_influence_total": 72.1,

...

}

]

•

B. monitoring_themes_distribution.json

•

Contient : Les données pour les graphiques "Camembert" (Pie Charts) et les barres.

•

•

Structure :

•

JSON

•

{

"global": [

{ "theme": "Politique", "count": 1500 },

{ "theme": "Sécurité", "count": 800 },

{ ... }

],

"by_media": {

"Lefaso.net": {

"Politique": 500,

"Économie": 200,

...

```
    },  
  
    "Sidwaya": { ... }  
  
}  
  
}  
  
•
```

C. monitoring_trends.json

```
•  
  
Contient : Les données pour les graphiques d'évolution (Line  
Charts) : posts par jour.
```

```
•  
•  
  
Structure :
```

```
JSON  
•  
  
•
```

```
{  
  
    "Lefaso.net": {  
  
        "dates": ["2025-11-01", "2025-11-02", ...],  
  
        "counts": [5, 8, ...]  
  
    },  
  
    "Sidwaya": {
```

```
        "dates": ["2025-11-01", "2025-11-02", ...],  
        "counts": [3, 4, ...]  
    }  
}
```

•

D. monitoring_alerts.json

•

Contient : Les alertes pour inactivité ou pics d'engagement.

•

•

Structure :

•

JSON

•

[

{

```
    "type": "inactivite",  
    "media": "Média X",  
    "message": "Aucune publication détectée depuis 8 jours.",  
    "date": "2025-11-17T06:30:00Z"  
,
```

```
{  
    "type": "pic_engagement",  
    "media": "Lefaso.net",  
    "message": "Pic d'engagement détecté sur un post (540 réactions)",  
    "post_link": "http://facebook.com/...",  
    "date": "2025-11-16T14:00:00Z"  
}  
]  
•
```

E. sensitive_alerts.json

```
•  
•  
Contient : Les alertes pour les commentaires toxiques/haineux détectés (Module 5).
```

Structure :

```
•  
•  
JSON  
•  
[
```

```
{  
    "comment_id": 123,  
    "post_id": 456,  
    "media_page": "Lefaso.net",  
    "comment_text": "Sale idiot ferme ta bouche",  
    "true_category": "toxic",  
    "model_label": "toxic",  
    "model_score": 0.95  
},  
{ ... }  
]  
•
```

3. Ce que tu dois préparer dans le Dashboard (React)

Voici les 5 composants React que tu dois créer pour afficher ces données. Utilise Recharts ou Plotly.js pour les graphiques, car ils sont demandés.

A. Composant 1 : Le Classement (Tableau d'influence)

•

Page : Page d'accueil / Dashboard principal.

-
-

Données : influence_ranking.json

-
-

UI : Un tableau (utilise Material-UI Table, Ant-Design Table ou react-table).

-
-

Colonnes à afficher :

-

1.

Rang (#)

2.

3.

Média

4.

5.

Score d'influence (Total) (sur 100)

6.

7.

Score Audience (sur 30)

8.

9.

Score Engagement (sur 40)

10.

11.

Score Régularité (sur 20)

12.

13.

Score Diversité (sur 10)

14.

•

Fonctionnalité : Le tableau doit être triable en cliquant sur l'en-tête "Score d'Influence".

•

B. Composant 2 : Le Centre d'Alertes (Très important pour le CSC)

•

Page : Page d'accueil ou onglet "Alertes".

•

•

Données : monitoring_alerts.json ET sensitive_alerts.json (tu devras fusionner les deux listes en React avec Promise.all et .concat()).

•

•

UI : Une liste de cartes (Cards).

•

•

Logique :

•

○

Pour chaque alerte de monitoring_alerts.json (type "inactivite" ou "pic_engagement") \rightarrow Affiche une Carte d'Alerte JAUNE (Avertissement).

○

○

Pour chaque alerte de sensitive_alerts.json (toxicité) \rightarrow Affiche une Carte d'Alerte ROUGE (Critique), en montrant le media_page et le comment_text.

o

C. Composant 3 : Analyse des Thématiques (Graphiques)

•

Page : Onglet "Analyse Thématique".

•

•

Données : monitoring_themes_distribution.json

•

•

UI :

•

1.

Graphique 1 (Pie Chart) : Un "camembert" affichant la Distribution Globale (utilise la clé "global" du JSON).

2.

3.

Graphique 2 (Bar Chart) : Un menu déroulant (<select>) pour sélectionner un média. Quand on choisit un média, un graphique à barres montre sa distribution de thèmes (utilise la clé "by_media").

4.

D. Composant 4 : Tendance des Publications (Graphique)

•

Page : Onglet "Activité" ou sur la page d'accueil.

•

•

Données : monitoring_trends.json

•

•

UI : Un graphique linéaire (Line Chart).

•

•

Logique : Le graphique doit montrer l'évolution du nombre de publications par jour. Idéalement, ajoute des checkboxes pour permettre à l'utilisateur de sélectionner/désélectionner les médias à comparer sur le graphique. Tu devras transformer les données de monitoring_trends.json en un format adapté pour Recharts/Plotly.

•

E. Composant 5 : Export (Bonus)

•

Page : Sur le composant "Classement" (Composant A).

-
-

Données : influence_ranking.json

-
-

UI : Un bouton "Exporter en CSV / Excel".

-
-

Logique : Utilise une librairie simple comme react-csv ou xlsx.

Quand l'utilisateur clique, cela convertit le JSON du classement en fichier CSV et le télécharge. (Oublie le PDF, c'est trop long à implémenter).

-

4. Résumé

Concentre-toi sur la création de ces composants React "vides" et sur la logique pour fetcher (useEffect) et afficher les données des 5 fichiers JSON.

Dès que j'ai fini d'entraîner mes modèles et de générer les fichiers, je te les envoie. Tu les mettras dans ton dossier public/data/ et le dashboard devrait s'allumer instantanément.

On se synchronise dans quelques heures. Bon courage !