



UNIVERSITÉ
CAEN
NORMANDIE

Université de Caen
Normandie



IUT Grand Ouest Normandie

Pôle de Caen

Bachelor Universitaire de Technologie

Science des Données

Deuxième année

Ressource : Développement d'un composant d'une solution décisionnelle

Tableau de Bord Interactif pour l'Analyse de l'Économie Mondiale



Auteurs

Romain LESUEUR

Anthony YON

Mandir DIOP

Courteney SAINT-HUBERT

Année universitaire 2024-2025

SOMMAIRE

Introduction	3
1 Contexte et Objectifs	5
1.1 Contexte du projet	5
1.2 Objectifs	5
2 Méthodologie	6
2.1 Appropriation de la structure de la base de données	6
2.2 Conception de la visualisation	8
Livrable du projet	10
Contributions individuelles	13
Conclusion globale	20
Table des figures	21

Introduction

Présentation du projet

Ce projet a pour objectif de concevoir un outil interactif et intuitif permettant de visualiser et d'analyser des données économiques mondiales. Dans un contexte où les données économiques jouent un rôle crucial dans la prise de décisions stratégiques, ce tableau de bord vise à offrir une vue d'ensemble des principaux indicateurs économiques tels que le PIB, la croissance démographique, l'espérance de vie, le développement humain, et la consommation énergétique.

L'idée principale est de transformer des données complexes en visualisations claires et accessibles, facilitant ainsi la compréhension des grandes tendances économiques mondiales. Grâce à des graphiques dynamiques et une carte interactive, les utilisateurs peuvent explorer les données par région, par pays, ou encore par période, tout en mettant en lumière les disparités régionales et les opportunités de développement.

Ce projet ne se limite pas à une simple présentation des données. Il s'agit également d'un exercice technique et collaboratif, mobilisant des compétences en développement web, en gestion de base de données, et en visualisation de données. En combinant des technologies modernes comme Chart.js et D3.js, nous avons cherché à créer un outil performant, ergonomique, et adapté aux besoins des utilisateurs.

En somme, ce tableau de bord interactif se veut être un outil pédagogique et analytique, permettant à la fois de mieux comprendre les dynamiques économiques globales et de valoriser les compétences techniques acquises au cours de cette formation.

Stack technique imposée

Le projet utilise les technologies suivantes :

- **Frontend** : HTML, CSS, JavaScript (Chart.js pour les visualisations, D3.js pour la carte interactive)
- **Backend** : PHP
- **Base de données** : MySQL (gérée via phpMyAdmin avec XAMPP)
- **Outils de versionnement** : Git + GitHub

En outre, une contrainte importante était d'utiliser une structure basée sur le modèle MVC (Modèle-Vue-Contrôleur) pour organiser le projet. Cette approche a permis de séparer clairement la logique métier (modèle), la gestion des interactions utilisateur (contrôleur), et l'affichage des données (vue). Une autre contrainte imposée était de ne pas utiliser de classes, ce qui nous a poussés à structurer le code en utilisant des fonctions et des fichiers bien organisés pour maintenir la lisibilité et la modularité.

Répartition des tâches initiales et méthodologie de travail

- **Anthony YON :**
 - Création et maintien du code dans `/models/pays.php`.
 - Création et maintien du code dans `/views/dashboard.php`.
 - Product Owner de la partie `/controllers/indicateurs.php` concernant le dashboard principal.
 - Création et maintien du code concernant les headers/footers.
 - Création d'au moins un graphique sur le dashboard principal.
- **Mandir DIOP :**
 - Création et maintien du code dans `config.inc.php`.
 - Création et maintien du code dans `/models/indicateur.php`.
 - Création du MLD (Modèle Logique de Données).
 - Création et maintien du code dans `/views/infos_pays.php`.
 - Création et maintien du code dans `/controllers/indicateur.php` concernant `/views/infos_pays.php`.
 - Création d'au moins un graphique sur le dashboard principal.
- **Romain LESUEUR :**
 - Création et maintien du code dans `/models/data.php`.
 - Création et maintien du dossier `/tests/`.
 - Création et maintien du logger.
 - Assistance sur le développement des parties `/views/` et `/controllers/`.
 - Création et maintien du code dans `/controllers/indicateurs.php` concernant le dashboard principal.
 - Création et maintien des fichiers suivants : `DEV.md`, `README.md`, `.htaccess`, `/docs/howtogit.md`.
 - Création d'au moins un graphique sur le dashboard principal.
 - Supervision de tout le projet.
- **Courteney SAINT-HUBERT :**
 - Création et maintien du code dans `/models/indices.php`.
 - Création du MCD (Modèle Conceptuel de Données).
 - Création et maintien du code dans `/views/comparaison_pays.php`.
 - Création et maintien du code dans `/controllers/indicateurs.php` concernant `/views/comparaison_pays.php`.
 - Création d'au moins un graphique sur le dashboard principal.

En outre, le projet a été versionné sur GitHub, ce qui a grandement facilité la gestion collaborative et le suivi des modifications. Vous pouvez consulter le dépôt GitHub à l'adresse suivante : <https://github.com/EkiaND/DED>.

En tant que superviseur du projet (Romain), j'ai joué un rôle central dans la coordination et la validation des tâches. À chaque push, je réalisais une revue du code pour vérifier sa qualité et sa fonctionnalité. Si le code était fonctionnel, nous discussions ensemble des éventuelles améliorations possibles. En cas d'erreurs, je les expliquais en détail à l'auteur, que ce soit des erreurs logiques ou de code, souvent lors d'appels pour clarifier les points complexes. Ce processus a permis d'assurer une progression continue et une amélioration constante de la qualité du projet.

1 - Contexte et Objectifs

1.1 Contexte du projet

Dans un monde où l'économie évolue rapidement, les données économiques jouent un rôle crucial dans la compréhension des dynamiques globales et la prise de décisions stratégiques. Chaque jour, une quantité massive de données est produite, couvrant des aspects variés tels que la croissance démographique, le PIB, l'espérance de vie, ou encore la consommation énergétique. Cependant, ces données, bien que riches en informations, restent souvent complexes et difficiles à interpréter sans outils adaptés.

C'est dans ce contexte que la nécessité de développer des outils de visualisation clairs, interactifs et informatifs se fait sentir. Ces outils permettent de transformer des données brutes en informations exploitables, facilitant ainsi l'analyse et la compréhension des grandes tendances économiques mondiales. Ils offrent également la possibilité de mettre en lumière les disparités régionales, les opportunités de développement, et les relations entre différents indicateurs économiques.

En outre, l'évolution rapide des technologies numériques a ouvert de nouvelles perspectives pour la visualisation de données. Les outils modernes permettent désormais de créer des interfaces interactives et dynamiques, rendant les analyses plus accessibles et engageantes pour les utilisateurs. Ce projet s'inscrit dans cette dynamique, en cherchant à répondre aux besoins croissants d'analyse économique dans un monde de plus en plus interconnecté.

1.2 Objectifs

Ce projet vise à concevoir un tableau de bord interactif dédié à l'analyse de l'économie mondiale. Il permettra de visualiser les principaux indicateurs économiques (PIB, croissance démographique, espérance de vie, développement humain, consommation énergétique, etc.) afin de mettre en évidence les grandes tendances et les relations entre ces facteurs.

À travers des visualisations claires et dynamiques, cet outil offrira une compréhension approfondie des évolutions économiques globales, facilitant ainsi l'identification des dynamiques de croissance, des disparités régionales et des opportunités de développement. Les objectifs spécifiques incluent :

- Utiliser une base de données économique mondiale déjà fournie et préparée.
- Concevoir une interface visuelle intuitive et dynamique pour faciliter l'analyse.
- Permettre le filtrage des données par période, zone géographique (région ou pays), et indicateur économique.
- Mettre en évidence les tendances économiques majeures à travers des visualisations interactives.

2 - Méthodologie

2.1 Appropriation de la structure de la base de données

La base de données fournie a été analysée pour comprendre sa structure et ses relations. Cette étape a été facilitée par l'utilisation de schémas MCD (Modèle Conceptuel de Données) et MLD (Modèle Logique de Données).

Le Modèle Conceptuel de Données (MCD) et le Modèle Logique de Données (MLD) sont deux étapes essentielles dans la conception d'une base de données relationnelle. Voici une explication détaillée de leur structure et de leur rôle dans ce projet.

Modèle Conceptuel de Données (MCD)

Le MCD représente une vue abstraite et indépendante des données. Il met en évidence les entités principales, leurs attributs, et les relations entre elles. Dans ce projet, le MCD est défini comme suit :

- **Entité Region** : Représente les grandes régions géographiques. Elle contient un identifiant unique (`id_region`) et un nom (`nom_region`).
- **Entité Pays** : Représente les pays. Elle contient un code unique (`code_pays`), un nom (`nom_pays`), une référence à la région (`id_region`), et un groupe de revenu (`groupe_revenu`).
- **Entité Indicateur** : Contient les données économiques et sociales mesurées pour chaque pays et chaque année, comme le PIB, le taux de natalité, ou l'espérance de vie.
- **Entité IndiceDeveloppement** : Contient des indices de développement humain et des mesures d'inégalités, comme l'IDH, l'espérance de vie, ou les émissions de CO2.

Les relations entre ces entités sont les suivantes :

- Une région (**Region**) peut contenir plusieurs pays (**Pays**).
- Un pays (**Pays**) peut avoir plusieurs indicateurs (**Indicateur**) mesurés chaque année.
- Un pays (**Pays**) peut également avoir plusieurs indices de développement (**IndiceDeveloppement**) calculés chaque année.

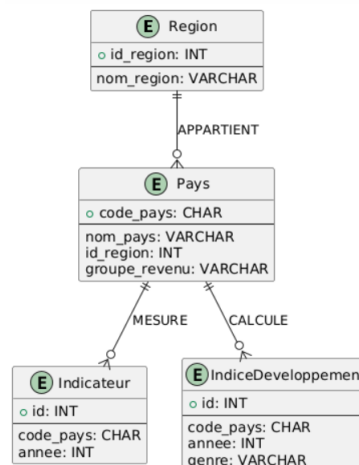


Fig. 1 - Modèle Conceptuel de Données (MCD)

La figure 1 illustre le MCD. On y voit les entités **Region**, **Pays**, **Indicateur**, et **IndiceDeveloppement**, ainsi que leurs relations. Par exemple, la relation **APPARTIENT** relie une région à plusieurs pays, tandis que les relations **MESURE** et **CALCULE** relient un pays à ses indicateurs et indices de développement respectivement.

Modèle Logique de Données (MLD)

Le MLD est une traduction du MCD en un modèle adapté à une base de données relationnelle. Il définit les tables, leurs colonnes, et les relations entre elles. Voici les principales tables du MLD :

- **Table regions** : Contient les colonnes `id_region` (clé primaire) et `nom_region`.
- **Table pays** : Contient les colonnes `code_pays` (clé primaire), `nom_pays`, `id_region` (clé étrangère vers `regions`), et `groupe_revenu`.
- **Table indicateurs** : Contient les colonnes `id` (clé primaire), `code_pays` (clé étrangère vers `pays`), `annee`, et divers indicateurs économiques comme `pib`, `esperance_vie`, ou `taux_chomage`.
- **Table indices_dvpt** : Contient les colonnes `id` (clé primaire), `code_pays` (clé étrangère vers `pays`), `annee`, et des indices de développement comme `idh`, `emissions_co2`, ou `inegalite_education`.

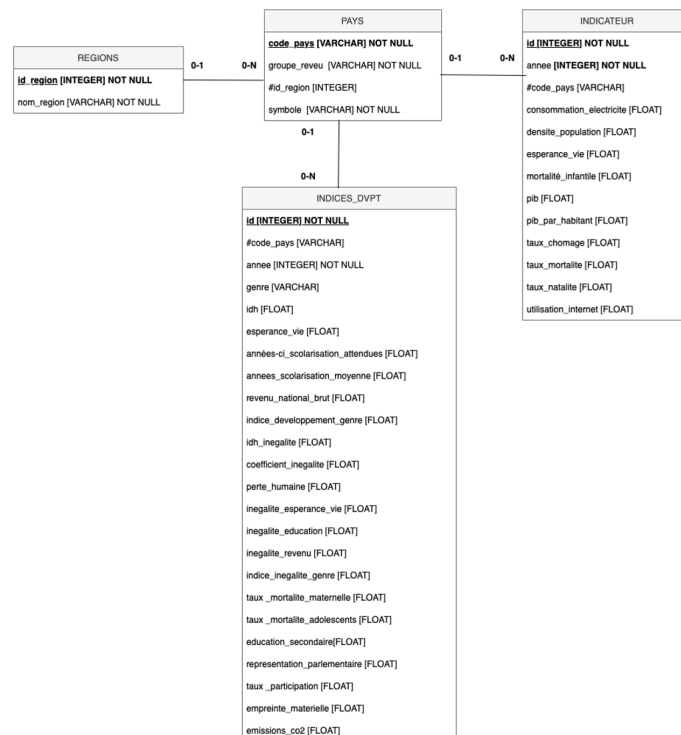


Fig. 2 - Modèle Logique de Données (MLD)

La figure 2 montre le MLD, où les entités du MCD sont traduites en tables relationnelles. Les relations sont implémentées à l'aide de clés primaires et de clés étrangères. Par exemple, la colonne `id_region` dans la table `pays` est une clé étrangère pointant vers la table `regions`, et la colonne `code_pays` dans les tables `indicateurs` et `indices_dvpt` est une clé étrangère pointant vers la table `pays`.

Conclusion

Le MCD et le MLD jouent un rôle fondamental dans la conception de la base de données de ce projet. Le MCD fournit une vue d'ensemble des données et de leurs relations, tandis que le MLD traduit cette vue en une structure exploitable par un système de gestion de base de données relationnelle. Ces modèles garantissent une organisation claire et cohérente des données, facilitant ainsi leur exploitation dans le tableau de bord interactif.

2.2 Conception de la visualisation

L'outil a été développé en utilisant Chart.js pour l'intégralité des graphiques, à l'exception de la carte interactive sur le dashboard, réalisée par Romain avec D3.js. Ces technologies ont permis de créer des visualisations dynamiques et interactives adaptées aux besoins du projet.

Pour garantir une organisation claire et modulaire, nous avons respecté la structure MVC (Modèle-Vue-Contrôleur) tout au long du développement. Voici une explication détaillée de cette structure et de son application dans le projet :

- **Modèle (Model)** : Le modèle est responsable de la gestion des données. Dans notre projet, les modèles sont implémentés dans les fichiers situés dans le dossier `/models/`. Ces fichiers interagissent directement avec la base de données MySQL pour récupérer, insérer ou mettre à jour les données. Par exemple, les données des indicateurs économiques ou des régions sont récupérées via des requêtes SQL dans les modèles, puis renvoyées sous forme de tableaux associatifs.
- **Vue (View)** : La vue est chargée de l'affichage des données. Dans notre projet, les fichiers situés dans le dossier `/views/` contiennent le code HTML et JavaScript nécessaire pour afficher les graphiques et la carte interactive. Par exemple, les graphiques générés avec Chart.js et la carte interactive réalisée avec D3.js sont intégrés dans les fichiers de vue. Ces fichiers reçoivent les données du contrôleur et les affichent de manière visuelle et interactive.
- **Contrôleur (Controller)** : Le contrôleur agit comme un intermédiaire entre le modèle et la vue. Les fichiers situés dans le dossier `/controllers/` reçoivent les requêtes des utilisateurs (par exemple, via des actions spécifiques dans l'URL), appellent les modèles pour récupérer les données nécessaires, et transmettent ces données aux vues. Par exemple, le fichier `indicateurs.php` dans `/controllers/` gère les requêtes pour récupérer les données des indicateurs et les renvoie sous forme de JSON pour être utilisées dans les graphiques.

Cette structure MVC a permis de séparer clairement les responsabilités, rendant le code plus lisible, modulaire et maintenable. Voici un exemple concret de son fonctionnement dans le cadre de la visualisation des données :

1. Lorsqu'un utilisateur sélectionne un indicateur et une année sur l'interface, une requête est envoyée au contrôleur (`indicateurs.php`) via une URL spécifique.
2. Le contrôleur appelle le modèle correspondant (par exemple, `indicateur.php`) pour récupérer les données nécessaires depuis la base de données.
3. Les données récupérées sont renvoyées au contrôleur, qui les formate en JSON et les transmet à la vue.
4. La vue utilise ces données pour générer un graphique interactif avec Chart.js ou mettre à jour la carte interactive avec D3.js.

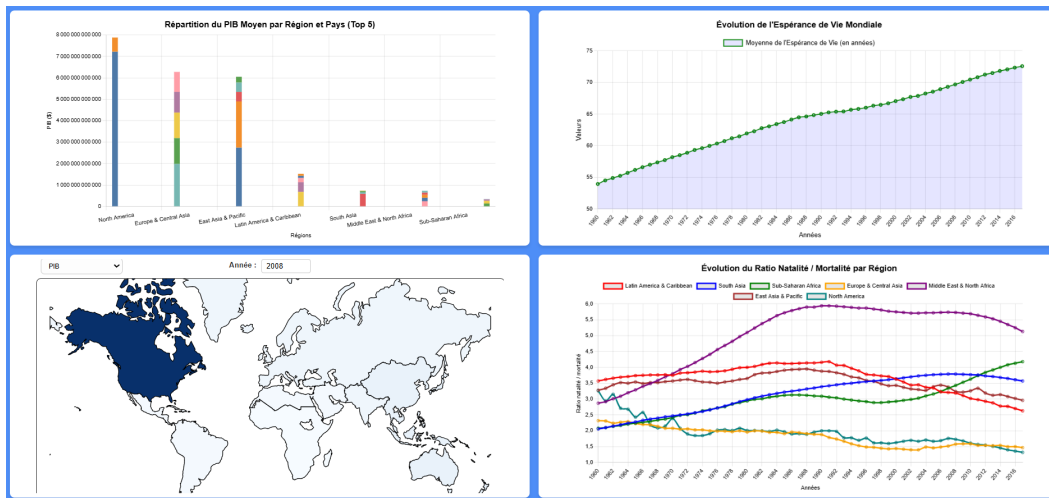


Fig. 3 - Architecture du tableau de bord interactif, intégrant Chart.js et D3.js dans une structure MVC

La figure ci-dessus illustre l'architecture du tableau de bord interactif, où chaque composant (modèle, vue, contrôleur) joue un rôle spécifique pour garantir une expérience utilisateur fluide et une gestion efficace des données.

Livrable du projet

Le livrable final est un tableau de bord interactif composé de trois pages principales :

— **Page des informations par pays :**

- Permet de sélectionner un pays, une année, et un indicateur.
- Affiche un tableau descriptif des différentes valeurs disponibles pour le pays et l'année sélectionnés.
- Affiche un graphique montrant l'évolution des données depuis l'année sélectionnée jusqu'à la plus récente dans la base de données pour le pays et l'indicateur choisis.



Fig. 4 - Page des informations par pays : tableau descriptif et graphique d'évolution

— **Dashboard principal :**

- Présente quatre graphiques interactifs affichant différents indicateurs économiques.
- Inclut une carte interactive permettant de changer l'année et l'indicateur affichés sur la carte.
- Permet une visualisation globale des données économiques à travers des graphiques et une carte.

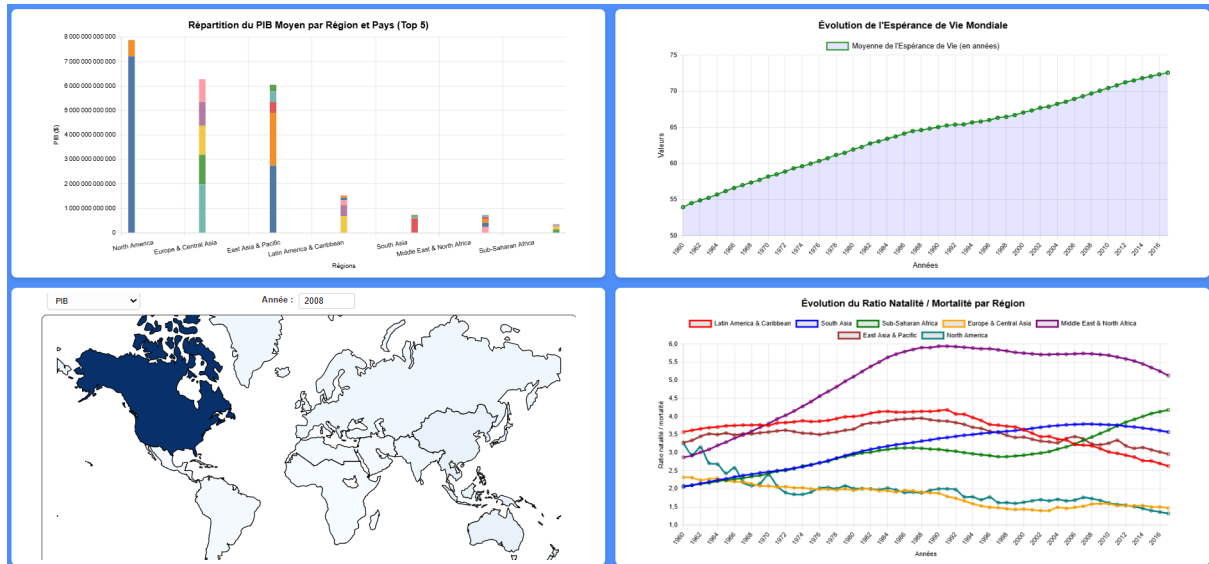


Fig. 5 - Page du dashboard : Graphique Chart.js et Map interactive avec D3.js

— **Page de comparaison entre pays :**

- Permet de sélectionner deux pays et un indicateur.
- Affiche deux indicateurs pertinents pour chaque pays sous forme de KPI (Key Performance Indicators) côte à côte.
- Affiche un graphique comparatif montrant les données associées aux deux pays pour l'indicateur sélectionné.

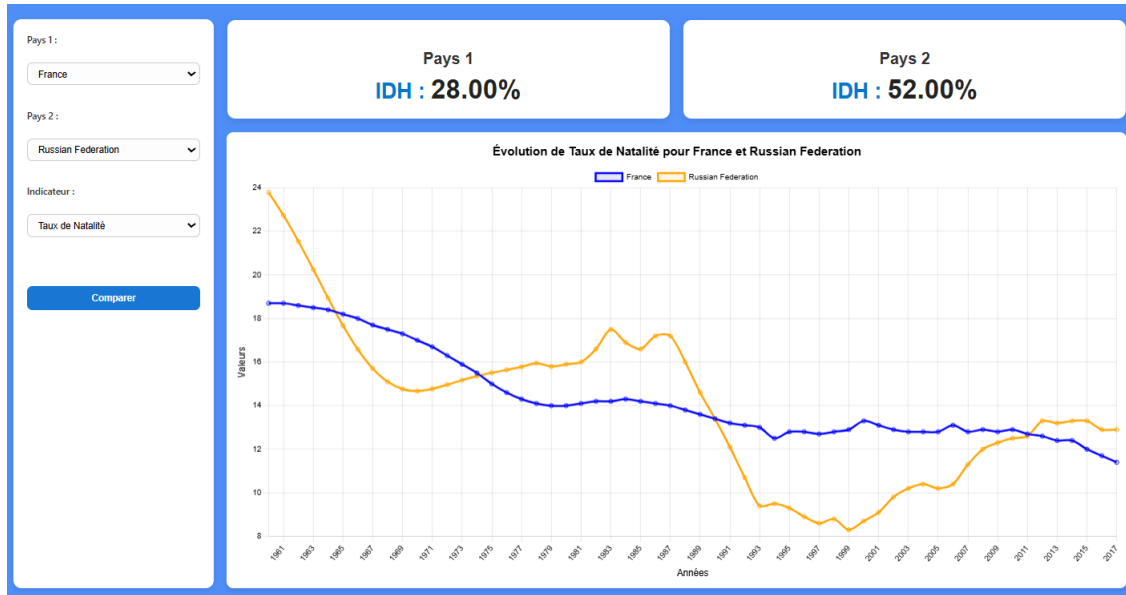


Fig. 6 - Page de comparaison entre pays : KPI et graphique comparatif

Contributions individuelles

Anthony YON

Tâches réalisées

Premièrement, dans la partie Backend, je me suis occupé de réaliser la connexion à la base de données dans `config.inc.php` (par la suite modifiée et perfectionnée par Romain). Ensuite, j'ai pu réaliser les fonctions du modèle `pays.php`, qui interroge directement la base de données afin de récupérer des informations sur les pays, comme son nom l'indique. J'ai donc créé les fonctions `getPays()`, `getDetailsPays()`, `getPaysParRegion()`, `getPaysParGroupeRevenu()`, `getRegions()` et `getNombrePaysParRegion()`.

Une fois mes tâches Backend réalisées, j'ai confectionné quelques-unes des actions dans le contrôleur `indicateurs.php`, qui m'ont servi pour confectionner le dashboard principal `dashboard.php`. Dans ce dernier, qui sert de page d'accueil de notre application en affichant des visuels concernant des données générales sur les données fournies, j'ai réalisé deux graphiques ("Évolution du PIB mondial par année" et "Évolution du ratio natalité/mortalité par région") qui ont nécessité la mise en place d'un code spécifique dans `graphiques.js`. J'ai aussi fait le `header.php` et le `footer.php`.

J'ai également été chargé de réaliser la page `comparaison_pays.php`. L'objectif était de créer une page interactive où l'on peut sélectionner deux pays et un indicateur afin d'afficher un graphique en courbe qui comparera l'indicateur au fil des années pour les deux pays. On retrouvera également sur cette page deux KPI.

Difficultés rencontrées

Je n'ai pas rencontré de difficultés particulières à la réalisation de la connexion à la base de données, excepté le fait de bien utiliser une base de données MySQL comme vu en cours. L'enjeu du modèle `pays.php`, comme pour les autres modèles, était d'optimiser le nombre de fonctions par rapport à nos besoins et la gestion des erreurs en cas de requête erronée.

C'est pour la partie contrôleur que ça s'est compliqué puisqu'avant de faire quoi que ce soit, il fallait bien avoir en tête l'architecture models-controllers-views. Au début, il a fallu que je refasse un point sur tous nos fichiers et leurs fonctions pour ensuite pouvoir coder sans risque. La gestion des erreurs silencieuses en PHP, notamment dans les requêtes SQL, a été un autre défi important, car la moindre faute de frappe pouvait entraîner des retours vides ou des erreurs JSON que j'ai pu détecter avec les fichiers tests ou dans la console. Il a aussi fallu bien structurer les réponses JSON pour qu'elles soient compatibles avec les appels JavaScript, notamment pour les graphiques ou les comparaisons entre pays.

Dans la partie Frontend, pour `dashboard.php`, la principale difficulté était de réussir à créer un visuel moderne et harmonieux qui donnerait envie à l'utilisateur de rester sur notre application. C'était globalement les mêmes défis pour `comparaison_pays.php`, en rajoutant la dimension interactive qu'il a fallu gérer dans `interactions.js`.

Méthodologie

Pour `pays.php`, j'ai commencé par tester les requêtes SQL dans phpMyAdmin. Une fois vérifiées, j'ai pu commencer à écrire mes fonctions en utilisant des requêtes paramétrées. L'utilisation de `mysqli_prepare()` m'a permis de rendre la requête plus sécurisée et performante. Par exemple, au final, la fonction `getPaysParGroupeRevenu()` prend en paramètres le groupe de revenu et retourne la liste des pays sous forme de tableaux associatifs.

Pour `indicateurs.php`, son but est de faire le lien entre les modèles et les vues en jouant le rôle d'interface API : il centralise les appels aux différentes fonctions du Backend en fonction de

ce dont nous avons besoin dans le Frontend, récupère les données traitées dans les modèles puis les renvoie au format JSON afin qu'elles puissent être exploitées dans le `dashboard.php`, etc.

Pour `header.php` et `footer.php`, j'ai simplement capitalisé mes connaissances en HTML et CSS pour, avec des classes et des identifiants, formater le bandeau correspondant.

Pour réaliser mes deux graphiques dans `dashboard.php`, j'ai utilisé Chart.js dans `graphiques.js`, comme vu en cours, avec le bloc de données, le bloc de configuration et le bloc d'affichage, tout cela renseigné dans une fonction pour l'utiliser dans le visuel final. Pour que cela fonctionne, il était primordial que mon contrôleur renvoie les données au format JSON.

Pour `comparaison_pays.php`, une partie de la méthodologie est la même que pour `dashboard.php`, à la différence près qu'ici j'avais à créer un formulaire HTML et notamment des listes déroulantes `Pays1` et `Pays2` avec les données récupérées par les requêtes dans le modèle `pays.php`. L'utilisation de `interactions.js` et `graphiques.js` a été requise pour que le graphique et les KPI s'adaptent bien aux pays sélectionnés.

Conclusion

Ce projet m'a permis de mettre en pratique de façon concrète l'architecture MVC et d'approfondir mes compétences aussi bien en PHP Backend qu'en JavaScript Frontend. J'ai particulièrement apprécié le travail d'interconnexion entre les modèles, le contrôleur et les vues, qui m'a permis de mieux comprendre les enjeux d'une application web dynamique. Ce fut un excellent exercice de rigueur, de logique et de collaboration, notamment pour assurer la cohérence des données du Backend jusqu'à leur affichage côté client. J'ai également apprécié le travail en groupe et notamment avec Romain, qui a su écouter et comprendre les problèmes que j'ai pu rencontrer pendant tout ce projet et m'accompagner dans la résolution de ces derniers.

Mandir DIOP

Tâches réalisées

- Création et maintien du code dans `config.inc.php`.

J'ai développé le fichier de configuration pour gérer la connexion à la base de données SQLite en utilisant le pattern Singleton, permettant une connexion unique et optimisée tout au long de l'exécution du script.

Cependant, ce code nécessite une adaptation selon le système d'exploitation utilisé (Mac ou Windows). Sous macOS, une authentification par mot de passe est requise pour se connecter à la base de données via phpMyAdmin avec MAMP.

- Création et maintien du code dans `/models/indicateur.php`.

Ce modèle assure l'interaction avec la base de données pour récupérer les informations sur les indicateurs concernant chaque pays.

Ce fichier contient notamment les fonctions :

- `getIndicateurs()`
- `getValeursIndicateur()`
- `getMoyenneIndicateurParRegion()`
- `getIndicateursParAnnee()`
- `getIndicateursParAnneePays()`

Ces fonctions permettent respectivement de récupérer dynamiquement la liste des indicateurs disponibles, les valeurs d'un indicateur pour un pays donné, la moyenne d'un indicateur par région, l'ensemble des indicateurs pour une année spécifique, ainsi que les données pour un pays et une année donnés. Ces fonctions facilitent l'accès et le traitement des données stockées dans la base pour les afficher sur l'interface web.

— Création du MLD (Modèle Logique de Données).

Quatre tables principales ont été conçues : REGIONS, PAYS, INDICATEUR et INDICES_DVPT, avec des relations de type 1-N pour représenter les données économiques par pays et par année.

1. Table REGIONS

La table REGIONS contient les identifiants et noms des grandes régions.

Clé primaire : id_region

Relation : Une région peut contenir plusieurs pays (relation 1-N). Sa clé primaire est utilisée comme clé étrangère dans la table PAYS (#id_region).

2. Table PAYS

Cette table contient les pays ainsi que leur appartenance régionale et leur groupe de revenu.

Clé primaire : code_pays

Relations :

- code_pays est référencée comme clé étrangère dans INDICES_DVPT et INDICATEUR.
- #id_region est une clé étrangère vers REGIONS(id_region).

3. Table INDICES_DVPT

Cette table contient des indicateurs de développement humain (IDH, espérance de vie, inégalités, etc.) par pays et par année.

Clé primaire : id

Clé étrangère : #code_pays référence la table PAYS(code_pays).

4. Table INDICATEUR

Elle stocke d'autres indicateurs économiques (PIB, taux de chômage, etc.).

Clé primaire : id

Clé étrangère : #code_pays référence également PAYS(code_pays).

Résumé des relations entre les tables

Clé source	Table cible	Champ cible	Type de relation
id_region	PAYS	#id_region	1 région → N pays
code_pays	INDICES_DVPT	#code_pays	1 pays → N indices développement
code_pays	INDICATEUR	#code_pays	1 pays → N indicateurs économiques

— Création et maintien du code dans /views/infos_pays.php.

Ce code crée une interface web interactive (informations sur les pays) permettant de sélectionner un pays, une année et un indicateur économique. Il affiche les informations générales du pays, sa région et un tableau regroupant plusieurs indicateurs (PIB, espérance de vie, etc.) de l'année choisie.

Un graphique (via Chart.js) qui illustre l'évolution d'un indicateur sélectionné en l'année choisie et 2018.

Le tout est mis à jour dynamiquement grâce à un script JavaScript (interaction_info.js). Cette vue permet une exploration simple et visuelle des données par pays et par année.

- Création et maintien du code dans `/controllers/indicateur.php` concernant `/views/infos_pays.php`.

Le contrôleur gère la logique de récupération des données à l'aide des fonctions définies dans `/models/inndicateur.php`, selon l'action demandée :

- (par exemple `?action=getMoyennePIBMondial`), il renvoie les résultats sous forme de JSON. Ce fichier permet d'alimenter dynamiquement les graphiques sur la page `infos_pays.php`
- Création du graphique en barre empilés (Top 5 des pays avec les grand PIB par régions) sur le dashboard principal. Ce graphique en barres empilées présente les 5 pays ayant le plus grand PIB dans chaque région. Les barres sont regroupées par région et chaque segment représente un pays avec sa contribution au PIB régional.

Difficultés rencontrées

- Gestion des valeurs manquantes dans la base de données (un nombre considérable de valeur manquantes concernant surtout les indicateurs).
- Optimisation de requêtes SQL pour la récupération et extractions des données.
- Adaptation du format des données à Chart.js.
- Maintien de la cohérence et de l'intuitivité de l'interface utilisateur en se basant sur les choix d'affichage.
- Rendre la page interactive

Méthodologie

- **Optimisation des requêtes SQL** : Construction dynamique des requêtes avec jointures internes optimisées et indexation appropriée. Extraction des métadonnées avec `INFORMATION_SCHEMA` pour adapter les requêtes à la structure réelle avec un remplacement des valeurs manquantes par des NA Lors de l'affichage des indicateurs sur la page les valeurs manquantes sont remplacés par indisponible.
- **Compatibilité avec Chart.js** : Transformation des résultats de requêtes en structures JSON uniformes via `array_map()` et fonctions de traitement dédiées. Séparation des libellés et valeurs en tableaux distincts facilitant l'intégration directe. Ensuite un chargement des données dans les listes déroulantes (pour le choix de l'utilisateur).
- **Cohérence de l'interface** : API REST standardisée avec points d'entrée homogènes et gestion d'erreurs cohérente. Les réponses JSON incluent systématiquement les métadonnées nécessaires (labels, unités) pour un affichage contextuel.
- **Interactivité** : Architecture asynchrone avec endpoints dédiés aux interactions dynamiques, permettant des recharges partielles sans rafraîchissement complet de la page pour afficher l'évolution des indicateurs choisies, la région ainsi que d'autres informations.

Conclusion

Le développement de ce tableau de bord économique m'a permis de renforcer mes compétences en PHP, SQL et JavaScript dans le cadre d'une architecture MVC. J'ai notamment appris à :

- Maîtriser le pattern Singleton pour la gestion des connexions à la base de données.
- Structurer et retourner les données sous forme de tableaux via les fonctions définies dans `/models/indicateur.php`.
- Concevoir un MCD et un MLD relationnel cohérents et optimisés.
- Sécuriser l'accès aux données grâce à l'utilisation de requêtes préparées et à la validation des entrées.

- Mettre en œuvre des visualisations dynamiques avec Chart.js à partir des données extraites de la base.
- Récupération des données da

Perspectives d'amélioration : une meilleure gestion des valeurs manquantes pourrait être envisagée. Par exemple, remplacer les données absentes par des valeurs issues de la moyenne ou de la médiane de l'indicateur concerné. Il serait également pertinent d'intégrer à terme des fonctionnalités d'analyse prédictive pour enrichir l'interprétation des données.

Courteney SAINT-HUBERT

Tâches réalisées

Difficultés rencontrées

Méthodologie

Conclusion

Romain LESUEUR

Tâches réalisées

Premièrement, j'ai créé et maintenu le code dans `/models/data.php`, ainsi que le dossier `/tests/` et le logger. J'ai également assisté au développement des parties `/views/` et `/controllers/`. J'ai conçu et maintenu le code dans `/controllers/indicateurs.php` pour le dashboard principal. En parallèle, j'ai rédigé et maintenu les fichiers `DEV.md`, `README.md`, `.htaccess`, et `/docs/howtogit.md`. J'ai aussi réalisé au moins un graphique sur le dashboard principal et supervisé l'ensemble du projet.

En tant que superviseur, j'ai également coordonné les tâches entre les membres de l'équipe, en m'assurant que chacun comprenne ses responsabilités et puisse avancer efficacement. J'ai pris en charge la validation des contributions, en effectuant des revues de code régulières pour garantir la qualité et la cohérence du projet. J'ai également fourni des exemples concrets pour guider mes coéquipiers, notamment en développant des fonctions et des graphiques de référence.

Difficultés rencontrées

La principale difficulté a été de travailler avec des personnes qui n'avançaient pas au même rythme ni au même niveau. Cela a nécessité une grande adaptabilité et une communication constante pour maintenir une dynamique de groupe positive. L'interdiction d'utiliser la POO a également été un défi, car cela limitait certaines optimisations et structurations du code. Enfin, l'exploration de D3.js pour la carte interactive a représenté un apprentissage important, car je n'avais jamais utilisé cette technologie auparavant.

Un autre défi a été de gérer les imprévus liés aux retards de certains membres, ce qui m'a poussé à reprendre certaines parties du projet pour respecter les délais. Cela a parfois créé une surcharge de travail, mais cela m'a également permis de renforcer mes compétences en gestion de projet et en résolution de problèmes.

Méthodologie

Pour ce projet, j'ai essayé de prévoir l'intégralité des fonctions nécessaires dans les modèles. J'ai rédigé le guide de développement et le `README.md` avant même de commencer à coder. Cependant, certaines fonctions dans les modèles ne sont pas utilisées, et d'autres ne sont pas mentionnées, ce qui reflète une planification initiale encore floue. J'ai décidé de développer d'abord les modèles, puis les vues, et enfin le contrôleur. Avec du recul, commencer par définir les vues aurait permis d'avoir une idée plus claire des besoins côté modèles.

J'ai donné ma vision pour chaque vue à développer, car certains membres étaient un peu perdus au début. Bien que je ne souhaitais pas imposer ma vision, elle a finalement été conservée. J'ai développé tous les tests et imposé une sortie uniforme sous forme de tableau associatif pour toutes les fonctions, afin de simplifier leur traitement dans les contrôleurs.

Les tests avaient pour but principal de fournir des exemples types et de visualiser les sorties via des pages de test accessibles sur le localhost (par exemple, `localhost/DED/tests/test_indicateur.php`). Concernant le dashboard, j'ai développé une fonction pour générer un graphique simple (évolution de l'espérance de vie), qui a servi d'exemple pour Anthony et Mandir dans leurs propres graphiques.

J'ai également réalisé une carte interactive avec D3.js, bien que Chart.js ait été imposé. Mon objectif était de reproduire une fonctionnalité vue dans une démo de l'entreprise Digdash, spécialisée en Business Intelligence. Cette carte permet une exploration approfondie des données. Cependant, des limitations sont apparues, notamment liées à la correspondance entre les fichiers GeoJSON et notre base de données. Par exemple, le Soudan du Sud n'apparaît pas dans la vue continentale mais est visible dans la vue régionale. Malgré ces imperfections, cette carte a été

bien accueillie, notamment par le RH de Digdash, ce qui m'a permis d'obtenir un stage chez eux.

Enfin, j'ai retouché le travail de chaque membre du groupe, non pas pour effacer leur contribution, mais pour leur montrer comment aller plus loin dans leur raisonnement. J'ai également pris le temps de documenter les parties critiques du projet pour faciliter leur compréhension et leur maintenance future.

Conclusion

J'ai grandement apprécié travailler avec Anthony, qui a été une belle surprise cette année. Ce projet m'a permis de travailler sur mon adaptabilité et de m'imposer comme chef de projet, ce qui m'a fait progresser en gestion de projet globale. Sur le plan technique, j'ai pu explorer l'utilisation de D3.js, ce qui a été une expérience enrichissante. Ce projet m'a également permis de renforcer mes compétences en communication et en leadership, des qualités essentielles pour mener à bien des projets collaboratifs complexes.

Conclusion globale

Ce projet a représenté une opportunité concrète de mettre en œuvre nos compétences techniques au sein d'un travail collaboratif. Malgré les contraintes de temps, les aléas extérieurs et les différences d'investissement entre les membres, nous sommes parvenus à aboutir à un résultat satisfaisant, stable et fonctionnel.

La répartition du travail s'est révélée inégale au fil des semaines. Certains membres ont dû prendre en charge une part importante du développement, incluant non seulement la création de fonctionnalités mais aussi la correction et l'harmonisation du code produit par d'autres. Ces ajustements, bien qu'invisibles en surface, ont été essentiels pour garantir la cohérence et la qualité globale du projet.

Anthony a démontré un engagement constant, assurant ses tâches tout en reprenant certaines parties initialement attribuées à d'autres. Il a su concilier ses responsabilités avec son stage, preuve d'un bon sens de l'organisation. Toutefois, il pourrait encore progresser en étant plus proactif et en n'attendant pas systématiquement les validations ou les consignes pour avancer.

Mandir a connu un démarrage difficile, pénalisé par des problèmes techniques et la charge de son propre stage. Néanmoins, il a su se mobiliser dans la seconde partie du projet, remplissant les tâches qui lui étaient confiées avec un accompagnement ponctuel. Pour les projets futurs, un renforcement de son autonomie et de sa communication serait bénéfique.

Enfin, la contribution de Courteney est malheureusement restée très limitée pour ne pas dire inexistante. Avec seulement 22 lignes de code inutilisables, son absence d'implication a eu un impact direct sur l'organisation collective. Ce manque de fiabilité a contraint les autres membres à compenser, ce qui a pesé sur la charge de travail globale.

Par ailleurs, le respect du calendrier initial n'a pas pu être assuré. La simultanéité avec d'autres projets à rendre, ainsi que les débuts de stage de certains membres, ont engendré des retards. Toutefois, grâce à une mobilisation progressive et à une bonne coordination en fin de parcours, nous avons su mener ce projet à son terme.

Nous avons également fait le choix de ne pas suivre aveuglément les instructions du projet. Bien que le but principal soit de nous évaluer sur nos capacités techniques, nous avons estimé que le sujet proposé était trop codifié et limitait l'aspect créatif. Par exemple, après avoir discuté avec Matthias et observé son livrable, j'ai constaté qu'il avait une page distincte pour chaque partie correspondant aux sections A, B, C et D. Cependant, nous avons choisi de combiner les parties A et B par exemple, car cela nous semblait plus logique et fluide. Je pense qu'il serait bénéfique que les sujets soient formulés de manière plus floue (sauf pour la stack technique), afin de pousser les étudiants à réfléchir davantage à la manière d'implémenter chaque partie selon leur propre approche. Cela encouragerait une plus grande diversité dans les livrables et stimulerait la créativité.

En somme, au-delà des compétences techniques mobilisées, ce projet a mis en lumière les enjeux humains d'un travail en équipe : l'importance de l'implication individuelle, la nécessité de la communication continue, et la gestion des imprévus. Il s'agit d'une expérience formatrice, riche en enseignements, qui nous permettra d'aborder nos futurs projets avec davantage de rigueur, de coordination et de maturité.

Table des figures

1	Modèle Conceptuel de Données (MCD)	6
2	Modèle Logique de Données (MLD)	7
3	Architecture du tableau de bord interactif, intégrant Chart.js et D3.js dans une structure MVC	9
4	Page des informations par pays : tableau descriptif et graphique d'évolution . . .	10
5	Page du dashboard : Graphique Chart.js et Map interactive avec D3.js	11
6	Page de comparaison entre pays : KPI et graphique comparatif	12