

## ▀ ▀ ▀ Modèle de copie :

**Créer une application web**

**GDWFSCAWEXAIII1A**

**Ceci est un modèle de copie. N’oubliez pas de renseigner vos prénom/nom, ainsi que le nom et le lien vers le projet.**

**Vous pouvez bien sûr agrandir les cadres pour répondre aux questions sur la description du projet si nécessaire.**

**Prénom :**Cédric

**Nom :** Phung

**ATTENTION ! PENSEZ À RENSEIGNER VOS NOM ET PRÉNOM DANS LE TITRE DE VOS FICHIERS / PROJETS !**

Nom du projet : KGB

Lien Github du projet : **https://github.com/ccphung/kgb**

Lien Drive du projet (si nécessaire) :

URL du site (si vous avez mis votre projet en ligne) : **https://kgb-2024-04118c8f1e45.herokuapp.com/**

**Description du projet**

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions. Dans cette rubrique, le jury cherche à voir comment vous procédez : comment vous organisez votre travail, comment vous réalisez concrètement la tâche ou l’opération pas à pas.

Utiliser un langage professionnel. Employez le « je », car vous parlez en votre nom. Vous pouvez écrire au temps présent.

Pour se connecter :

Identifiant : cedric@kgb.com

Mot de passe : password

**Mise en place du modèle MVC :**

Je commence par mettre en place la structure du projet en suivant le modèle MVC

Je crée donc un dossier controllers, models et views.

**Création du dossier app**

Dans le dossier app je crée un Controller un Model principal.

Dans le Controller principal j’instancie le modèle qui lui est associé et crée une fonction qui va générer la vue.

Dans le Model principal, je mets en place la connexion à la BDD ainsi que d’autres fonctions qui pourront me servir pour différentes entités.

**Création du dossier public**

Dans le dossier public je crée un dossier css, js et images qui seront composés des fichiers de style, de mes scripts et des différentes images intégrées au site.

**Création du dossier core**

Je crée dans ce dossier le fichier Form.php qui inclus une fonction de création de formulaire et de vérification.

**Création du Router:**

Le premier controller que je crée est le Router, composé de deux fonctions : addRoute et getHandler.  
La première fonction permet d’ajouter les différentes routes et la deuxième me permet de les gérer.

**Mise en place d’un template**  
Je crée le fichier default.php qui sera le template de base de la vue.

Ce fichier comprend le head (inclusion des fonts de google, bootstrap, feuilles de style et javascript) ainsi que la nav.

**Mise en place de la partie authentification :**

Création de login/index.php (view) AuthController (controller) et de User.php (model).

Login/index.php : création d’un formulaire de connexion. La route est ajoutée à la page index.php qui gère les routes via le Router.

AuthController : récupère les éléments du formulaire (email et mot de passe).

User.php : vérifie que les éléments sont présents et corrects. Si oui, redirection vers la page d’accueil et création de la session.

**Mise en place des agents, cibles, contacts et missions :**

Création des différentes vues avec DataTable qui permet d’y inclure une pagination (que j’ai désactivé car je l’avais créée avant d’inclure DataTable), une barre de recherche, des tris, tout ça sous forme de tableau.

J’affiche les différentes données : le model va chercher les données en base de données via une requête SQL et le contrôleur envoie les données à la vue.

Pour les missions, j’ai également créé une page détail permettant d’afficher le détail pour la mission en question.

**Mise en place de formulaires d’ajout:**

Lorsque l’utilisateur est connecté (vérification de la session, s’il y a bien un user dans les données de sessions et si elles ne sont pas nulles), alors il a accès à un formulaire permettant d’ajouter un agent/mission/contact/cible.  
  
Je crée un formulaire avec ma fonction de création dédiée. Pour les missions, l’affichage de certains éléments sont conditionné par les données choisies. Cela est géré avec Ajax via JQuery.  
Par exemple, l’affichage des agents déprendra de la spécialité requise. Ainsi, à chaque changement de la spécialité, une requête POST est envoyée avec l’id de la spécialité. Le controlleur gère la requête qui appelle la fonction en charge de requêter les agents dans le model puis envoi les données à la vue.

Je vérifie que les données ne sont pas vides et ensuite, j’insère en BDD.  
Controlleur : création du formulaire et envoi à la vue

Model : Requête SQL qui permet d’insérer une nouvelle instance

Vue : Affiche le formulaire et un message de confirmation si tout s’est bien passé

**Mise en place de la partie Admin :**

Création d’une page Admin permettant de modifier/supprimer.

Page d’accueil : permet d’accéder aux missions, agents, contacts et cibles

Page de missions/agents/contacts/cibles : affiche la liste des éléments avec en plus un bouton « modifier » pour l’élément à supprimer ou à modifier.

Page modification : je recrée le même formulaire que celui pour l’ajout d’un élément. Dans les inputs, je récupère les données en BDD pour les afficher en tant que value. Les checkbox sont cochées sur les éléments qui étaient sélectionnées. Par exemple, si l’Agent avec l’id 3 avait pour spécialité « informatique » et « armes à feu » alors en ouvrant le formulaire ces deux éléments seront déjà cochées.

Enfin un bouton « supprimer » est ajouté en bas de la page, celui-ci permet d’envoyer une requête POST qui demande la suppression des éléments dans toutes les tables dans le bon ordre afin de ne pas avoir de blocage lié aux clés étrangères.  
Par exemple, pour la suppression d’un agent :

- Il faut supprimer l’agent de la table d’association agent\_mission

- Puis il faut supprimer l’agent de la table d’association agent\_specialty

- Ensuite, suppression dans la table agents

- Enfin suppression dans la table persons.

1. Précisez les moyens utilisés. Expliquez tout ce dont vous avez eu besoin pour réaliser vos tâches : langages de programmation, frameworks, outils, logiciels, documentations techniques, etc...

Langage de programmation : PHP 8, CSS, HTML

Librairie : Bootstrap, JQuery

Tuto Youtube, documentation PHP, Stakeoverflow, documentation DataTable

1. Contexte. Les noms des organismes, entreprises ou associations, dans lesquels vous avez exercé vos pratiques

NB: Pour le cas des exercices et évaluations demandées sur la plateforme Studi, il s'agit de...Studi.

Studi

1. Informations complémentaires (*facultatif*)