

## ▀ ▀ ▀ 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 : Steven**

**Nom :** GOMES

**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/Blacklight059/kgb

Lien Drive du projet (si nécessaire) :

URL du site (si vous avez mis votre projet en ligne) : https://kgb-defi.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 commencer, j’ai écrit un scénario pour définir les rôles et les situations.

J’ai commencé par définir les tables et les liens entre elles avec un schéma, puis j’ai créé ma base de données avec toutes les tables.

J’ai créé les maquettes en commençant par la page d’accueil avec un header composer d’un logo et d’une partie pour accéder à la partie administration.

Dans la partie body, j’ai créé deux extraits d’articles avec une partie titre en noir, une partie extraite et une date de publication dans la partie basse à droite

J’ai créé ensuite la partie connexion ou inscription de l’administration avec deux formulaires distincts.

J’ai créé ensuite une page d’article avec le même titre, le type de mission une image de la cible et le contenu de la mission au complet.

J’ai commencé à créer l’interface du site avec les dossiers et les premiers fichiers.

J’ai commencé par le header avec l’intégration du logo et du menu avec Bootstrap.

J’ai organisé mon arborescence avec plusieurs dossiers dont le dossier public et le dossier views.

J’ai installé composer et altoRouter créer mon fichier Router dans le dossier src et le fichier index.php dans le dossier public pour avoir les routes du site pour les missions et la partie administration.

J’ai commencé par la partie administration

Avec une partie connexion et une partie inscription

Pour la partie connexion, j’ai créé un fichier admin puis le formulaire de connexion avec une partie email, mot de passe un bouton connexion et un bouton inscription, j’ai appliqué du css. Pour la partie inscription, j’ai réalisé la même chose avec les champs décrit dans l’énoncé.

J’ai créé ensuite un fichier Config.php pour ma connexion à la base de données en utilisant le système de namespace avec l’instance de POD et la class Config avec une fonction static et un system erreur en cas de problème de connexion.

Pour établir mon lien avec la base de données, j’ai utilisé session start et une condition pour savoir si les champs existent et ne sont pas vides, ensuite je vérifie si l’utilisateur est inscrit dans la base de données puis si l’email est au bon format puis si le mot de passe est correct.

Je renvoie une erreur si un des champs est mauvais avec un message que s’affiche grâce à login\_err transmit dans l’URL.

Si l’email et le mot de passe sont bons, je créé une session et je redirige vers la page landing.php.

Pour la page Landing.php, j’établie la connexion à la base de données et je vérifie si la session n’ai pas créé je redirige vers la page d’accueil sinon je récupère les données de l’utilisateur et j’affiche un espace membre avec le nom du pseudo.

J’affiche un bouton pour modifier le mot de passe et un bouton de déconnexion.

Pour la partie inscription, j’ai réalisé la même gestion des erreurs en modifiant les conditions d’acceptations pour s’adapter aux champs, l’insertion dans la base de données dans un fichier inscription\_traitement.php et dans un fichier inscription.php la partie front end avec le formulaire et le css.

Je suis passé par la création des autres dossiers : le public, le src et le dossier views ;

Pour le dossier public, j’ai créé la page index qui va me permettre d’afficher le header et d’utiliser le router afin d’avoir toutes les routes du site et de créer les url afficher dans le navigateur.

J’ai installé via composer Whoops pour mieux gérer mes erreurs sur les phases suivantes.

Dans le dossier views, j’ai créé 3 sous dossiers :

* Le dossier post pour les pages principales la page d’accueil, la page des missions et la page admin.
* Le dossier auth pour les fichiers qui permettent la gestion de l’administrateur.
* Le dossier admin pour la gestion des données de mission.

Pour le dossier src, j’ai créé le sous dossier table et models pour créer plusieurs fonctions pour éviter la répétition du code.

Pour ma page d’accueil situé dans le dossier post j’ai créé le fichier index.php, j’ai établi ma connexion à ma base de données grâce au fichier config.php et une variable pour le titre.

Dans le dossier Models, j’ai créé le fichier Post.php pour la class Post avec les getters et setters de chaque champ de la table missions en base de données et deux fonctions pour formatter la partie content de la mission à afficher comme extrait sur la page.

Ensuite dans le dossier src, j’ai créé le fichier PaginatedQuery.php pour le système de pagination en fonction du nombre de mission récupérer grâce à la fonction getItems ainsi que les fonctions pour gérer les pages suivantes et précédentes.

J’utilise ma class PaginatedQuery pour récupérer mes données dans l’ordre de date du début de mission et je compte via id le nombre de missions.

Pour la suite, j’ai créé une div avec un foreach qui récupère chaque missions ,puis j’affiche en titre la donnée slug grâce à getSlug puis la date du début de mission grâce à getDatedebut avec un format approprié puis un extrait du contenu de la mission avec la fonction getExerpt et j‘ajoute un bouton voir plus avec un lien vers la page de la mission grâce à la fonction getId et à mon système de router.

Pour les page suivantes et précédentes j’utilise mes fonctions PreviousLink et nextLink de la class PaginatedQuery.

J’ai créé le fichier mission.php pour faire apparaitre chaque mission, je récupère l’id et le slug de la précédente page, j’initialise pdo pour la connexion à la base de données, je récupère l’ensemble des données de la table missions et je fais le lien avec l’id de la variable.

J’ai décidé d’afficher le slug qui correspond au nom de code de la mission dans un h1 puis dans l’ordre qui suit :

* Cible
* Agent
* Contact
* Spécialité de la mission
* Date de fin de mission
* Type de mission
* Statut de la mission

Pour se faire, j’ai lié la ligne mission\_id des tables a afficher avec l’id de la mission. J’ai dû créer les class et les tables de chaque élément en copiant sur post et PostTable pour pouvoir établir les liens et récupérer les données nécessaires à l’affichage.

Et en drenier le contenu afficher au complet.

Je suis passé à la partie administration pour la création et la modification j’ai créé un dossier admin dans le dossier views avec un dossier pour chaque donnée à modifier. Pour chaque éléments j’ai créé ls fichiers :

* \_form pour la gestion du formulaire
* Delete pour la suppression d’une donnée
* Edit pour l’edition
* Index pour l’affichage
* New pour la création

J’ai commencé avec l’index, j’ai déterminé le titre dans une variable, j’initialise pdo et je vérifie si la session est créée sinon je redirige vers la page d’accueil.

Ensuite je créé un tableau avec un entête pour afficher avec l’id, le titre, et un bouton nouveau avec un lien vers la page new, puis j’affiche l’id, le slug et j’ajoute un bouton supprimer.

A la fin je rajoute les boutons suivants et précedent.

Pour Edit, je vérifie aussi si la session existe, je récupère les données de chaque tableau qui correspond aux id.

Pour le formulaire, je créé une class Form dans un dossier HTML dans le dossier src, pour pouvoir créer mes fonctions pour gérer les données.

Dans le fichier \_form.php j’ai juste intégrer ma fonction input avec les données à écrire et un select pour les données des autres tableaux.

Je créé une fonction pour modifier et mettre à jour les données.

Ensuite j’ai créé plusieurs conditions pour gérer les erreurs et le succès de la modification.

J’ai inscrit un h1 avec le nom de la mission, puis le formulaire.

Pour le fichier new.php, je vérifie toujours la session et je le fais pour chaque page en rapport avec l’administration. J’établi les liens avec les autres tables pour les listes ensuite je créé des conditions pour entrer les nouvelles données et je rajoute le formulaire.

Pour le delete.php, j’utilise la fonction delete et je retourne vers la page mission.

Pour la suite, j’ai copié le dossier pour chaque élément à gérer en administration.

J’ai créé une class Table générique je l’ai rendu abstrac et mes autres tables en final pour éviter les répétitions de code.

Pour la suite, j’ai modifié le css pour correspondre aux maquettes en ajoutant les couleurs et les mises en page et le menu pour qu’une partie apparaisse lors de la connexion de l’administrateur.

Pour la mise en ligne, j’ai utilisé Heroku et j’ai suivi les instructions sur leur site par le lien github.

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...

J’ai utilisé le site pour créer mon graphique avec lucid.app, Figma pour réaliser les maquettes, phpmyadmin pour la base de données, et virtualstudiocode pour le codage en me basant sur les cours Studi et les cours de Grafikart et la documentation php et heroku pour la mise en ligne..

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.

1. Informations complémentaires (*facultatif*)