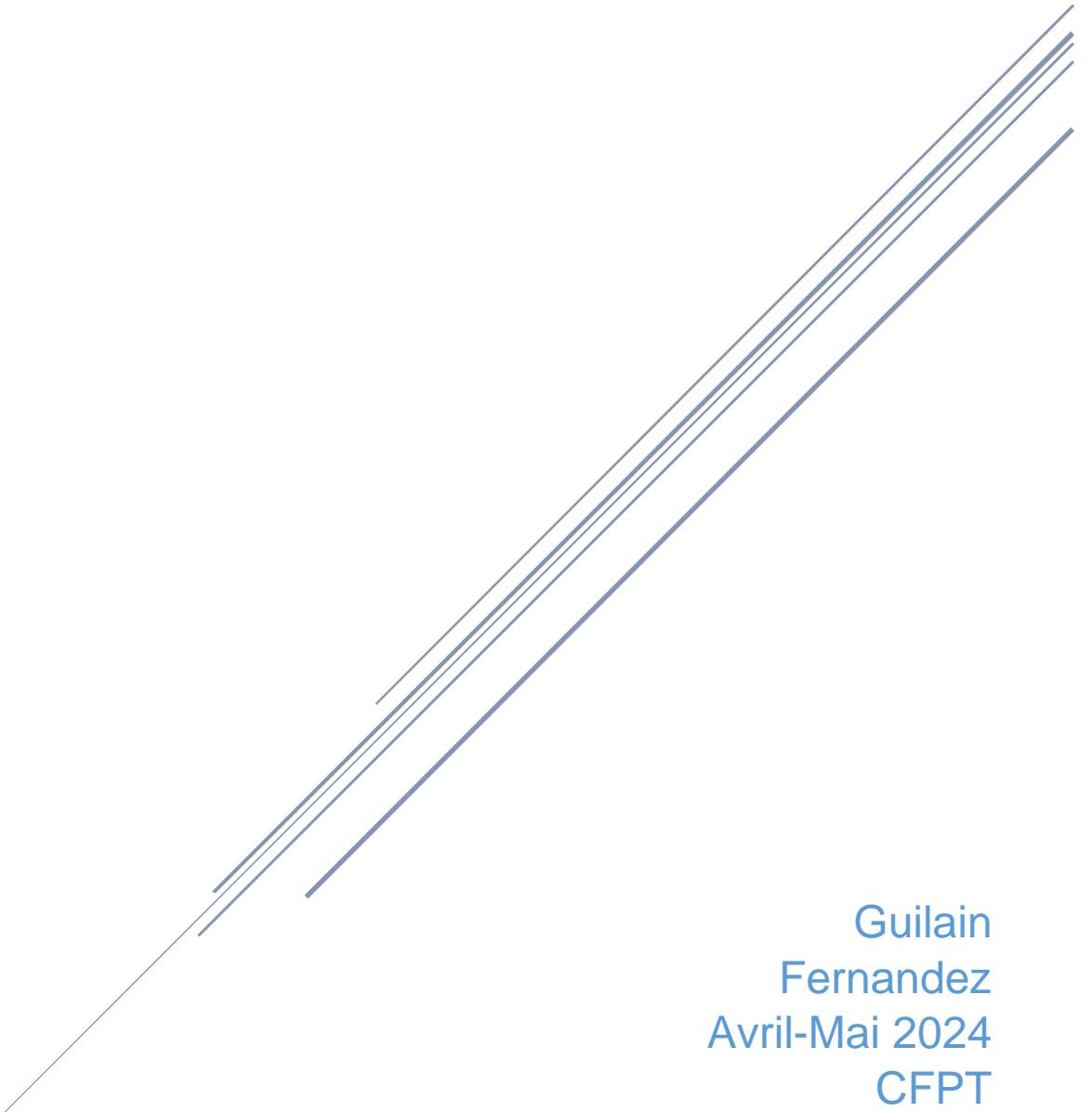


DOCUMENTATION TECHNIQUE

VoyageMémo



Guilain
Fernandez
Avril-Mai 2024
CFPT
Informatique

<i>Introduction</i>	2
<i>Organisation</i>	2
Méthodologie	2
<i>Analyse fonctionnelle</i>	3
<i>Inscription et Authentification</i>	3
<i>Publication d'un voyage et de ses étapes</i>	3
<i>Affichage des voyages</i>	3
<i>Détails des voyages</i>	3
<i>Profil</i>	4
<i>Analyse organique</i>	4
<i>Plan de navigation</i>	4
<i>Schéma</i>	5
<i>Structure du projet</i>	7
<i>Installation du projet</i>	14
<i>Outils utilisés</i>	14
<i>Modèle conceptuel de données</i>	15
<i>Dictionnaire de données</i>	15
<i>Test</i>	16
<i>Plan de tests</i>	16
<i>Rapport de tests</i>	18
<i>Conclusion</i>	23
<i>Difficultés rencontrées</i>	23
<i>Amélioration possible</i>	24
<i>Bilan personnel</i>	24
<i>Remerciements</i>	24
<i>Planification</i>	25
<i>Table des illustrations</i>	26
<i>Bibliographie</i>	27
<i>Glossaire</i>	27
<i>Annexe</i>	27

Introduction

Cette documentation va présenter toutes les informations importantes pour toute personne souhaitant reprendre le projet « VoyageMémo ».

J'ai réalisé ce projet lors de mon travail pratique individuel en 4ème année, dans le but de valider mon CFC en informatique. M. Garchery est mon maître d'apprentissage et la personne qui a rédigé le cahier des charges. Il est accompagné de deux experts, M. Monet et Mme. Travnjak.

Le projet est un site web qui permet aux utilisateurs de documenter et de partager les aventures de leurs voyages. Elle procure une plateforme personnelle pour enregistrer des souvenirs, des récits, des photos et des itinéraires de voyage.

Organisation

Méthodologie

La méthodologie que j'ai utilisée pour ce projet est la méthode en 6 étapes. J'ai sélectionné cette méthode car je la trouve pratique et facile à mettre en œuvre.

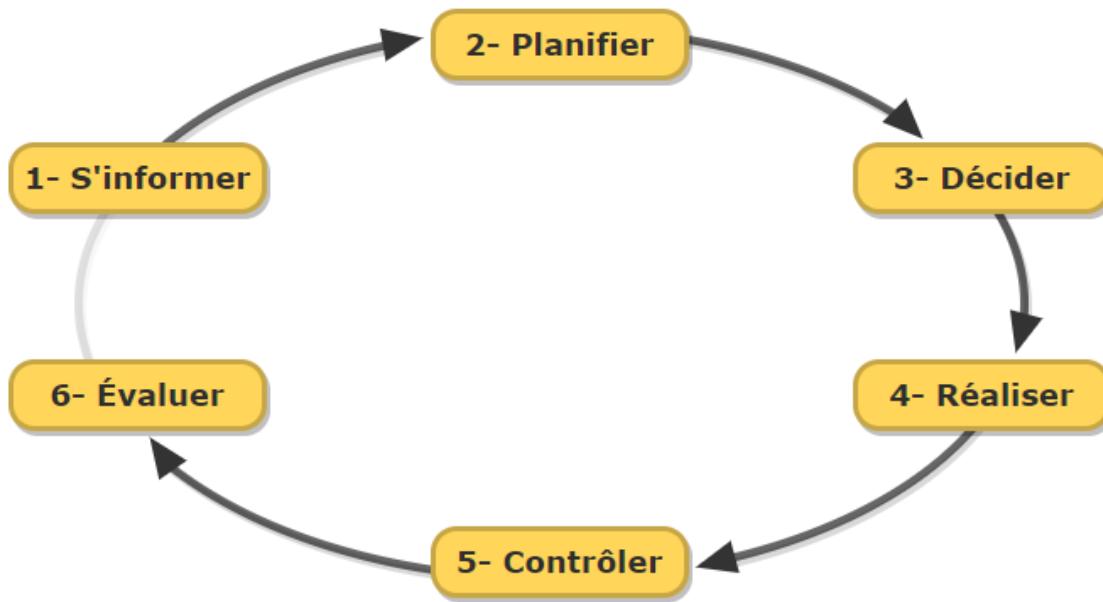


Figure 1.méthode en 6 étapes.

1- S'informer

J'ai commencé par prendre connaissance de l'énoncé de mon TPI. Puis j'ai rédigé des questions que j'ai envoyées à mon maître d'apprentissage pour mieux comprendre mon énoncé.

2- Planifier

Le premier jour du TPI, j'ai commencé par faire mon planning prévisionnel. J'y ai inclus toutes les tâches qui me semblaient importantes et je l'ai séparé en 5. S'informer, Planifier, Réalisation du projet, Tests et Documentation.

3- Décider

J'ai choisi les outils et programmes qui étaient adaptés au projet qui m'a été confié.

4- Réaliser

Cette étape est la réalisation du projet comme demandé dans le cahier de charges fournie et le respect au mieux du planning défini en début de projet.

5- Contrôler

J'ai élaboré un plan de tests afin de vérifier le bon fonctionnement de mon programme et de déterminer s'il y avait des corrections à apporter. J'ai aussi relu mon cahier des charges pour vérifier si j'avais respecté tous les points.

6- Evaluer

J'ai évalué mon travail en relisant le cahier des charges et en faisant une auto-évaluation pour me rendre compte si j'avais bien réussi mon travail et identifier les points d'amélioration.

Analyse fonctionnelle

Inscription et Authentification

Les utilisateurs peuvent s'inscrire sur le site en fournissant un nom d'utilisateur et un mot de passe.

Une fois inscrits, les utilisateurs peuvent se connecter avec leur identifiant.

Publication d'un voyage et de ses étapes

Les utilisateurs connectés peuvent créer un voyage en remplissant un formulaire et ajouter des étapes à ce voyage en remplissant un autre formulaire.

Ils peuvent également télécharger des photos en créant une étape.

Affichage des voyages

Les voyages sont affichés sur la page d'accueil.

Chaque voyage affiche son titre, sa date de début, sa date de fin, sa durée, son nombre d'activités et une image aléatoire qui lui est liée.

Détails des voyages

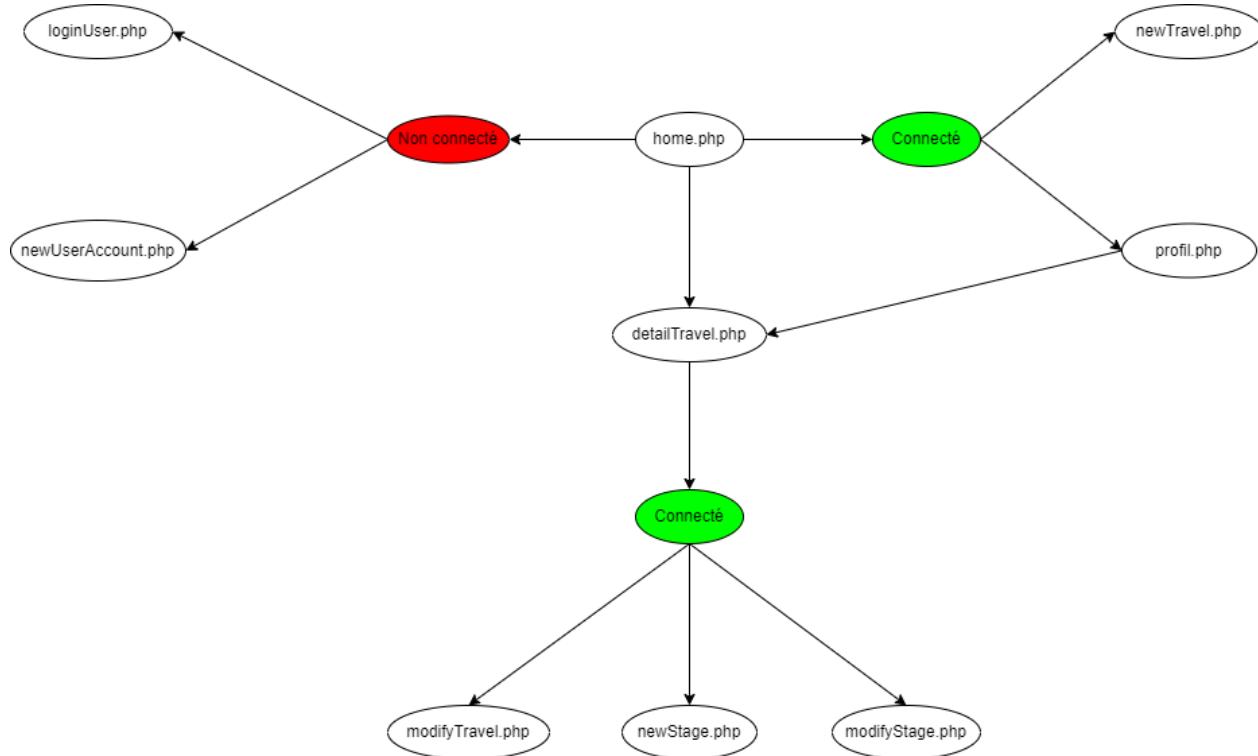
Les utilisateurs peuvent cliquer sur un voyage pour afficher les détails complets des étapes, y compris toutes les photos.

Profil

Les utilisateurs peuvent voir la liste de leurs propres voyages et les informations de leur compte utilisateur.

Analyse organique

Plan de navigation



Schéma

Accueil



1. Menu déroulant avec les redirections pour l'utilisateur.
2. Redirection vers le formulaire pour créer un nouveau voyage.
3. Example de card avec les informations d'un voyage et une redirection vers la page détail de ce voyage.
4. Logo du site web.

Détail

The wireframe shows a card for a travel step. At the top are three buttons: 'Nouvelle étape' (1), 'Modifier voyage' (2), and 'Supprimer voyage' (3). Below these is the title 'Titre voyage' and the subtitle 'Date Début / fin voyage'. The main content area contains fields for 'Titre', 'Date', 'Durée', 'Description', 'Activité', and 'Images'. At the bottom of the card are two buttons: 'Modifier' and 'Supprimer'.

1. Redirection vers le formulaire pour créer une nouvelle étape.
2. Redirection vers le formulaire pour modifier un voyage.
3. Bouton pour supprimer un voyage.
4. Example de card avec les informations d'une étape avec une redirection vers le formulaire de modification de l'étape et un bouton pour supprimer l'étape.

Profil

The wireframe shows a user profile section with 'Nom utilisateur' (1) and a 'Nouveau voyage' button. Below it is a list of travel entries with the header 'Nom voyage - Début - Fin - Durée- Détail->' (2).

1. Info de l'utilisateur
2. Liste des voyages créé par l'utilisateur avec une redirection vers le détail du voyage sélectionné.

Structure du projet

Pour ce projet, j'ai décidé d'utiliser l'architecture MVC afin d'organiser mon code et de séparer les différentes responsabilités de mon site web.

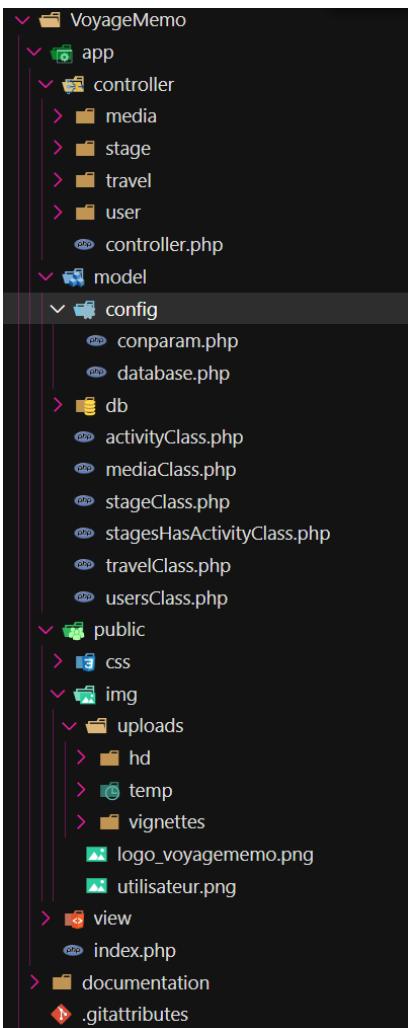


Figure 2. Arborescence des fichiers du site web

- VoyageMemo/app/controller/media/
Contient les fonctions pour redimensionner les images.

Prévisualisation d'un code PHP pour la redimensionnement d'images :

```

1 /**
2  * Redimensionne une image aux dimensions spécifiées tout en conservant son ratio d'aspect.
3  *
4  * @param string $originalpicture Le chemin de l'image source à redimensionner.
5  * @param int $width La largeur souhaitée de l'image redimensionnée.
6  * @param int $height La hauteur souhaitée de l'image redimensionnée.
7  * @return GdImage|bool Retourne une ressource image de l'image redimensionnée ou false en cas d'erreur.
8  */
9 function resize($originalpicture, $width, $height)
    
```

```
1  /**
2   * Découpe une image centrale selon les dimensions spécifiées et enregistre le résultat dans un fichier.
3   *
4   * @param mixed $sourceImage La ressource de l'image source à découper.
5   * @param string $destinationPath Le chemin de destination où enregistrer l'image découpée.
6   * @param int $width La largeur souhaitée de l'image découpée.
7   * @param int $height La hauteur souhaitée de l'image découpée.
8   * @return void
9  */
10 function pictureTrim($sourceImage, $destinationPath, $width, $height)
```

- *VoyageMemo/app/controller/stage/*
Contient le code source qui permet de gérer les étapes.
- *VoyageMemo/app/controller/travel/*
Contient le code source qui permet de gérer les voyages.
- *VoyageMemo/app/controller/user/*
Contient le code source qui permet de gérer les utilisateurs
- *VoyageMemo/app/model/config/*
Contient les fichiers de configuration de la base de donnée.
- *VoyageMemo/app/model/activityClass.php*

```
1  /**
2   * Récupère toutes les activités depuis la base de données.
3   *
4   * @return array|bool Retourne un tableau d'objets Activity contenant toutes les activités si trouvées,
5   *                     sinon retourne false en cas d'erreur ou si aucune activité n'est trouvée.
6  */
7 function getAllActivity()
```

```
1  /**
2   * Récupère les activités associées à un stage spécifique.
3   *
4   * @param int $stage L'identifiant du stage.
5   * @return array|bool Retourne un tableau d'objets Activity représentant les activités associées au stage, ou faux en cas d'erreur.
6  */
7 function getActivityByStageId($stage)
```

- *VoyageMemo/app/model/mediaClass.php*

```
● ● ●  
1  /**
2   * Crée un nouveau média dans la table 'media'.
3   *
4   * @param string $date La date d'ajout du média (au format YYYY-MM-DD).
5   * @param string $fileNameVignette Le nom du fichier vignette.
6   * @param string $fileNameHd Le nom du fichier HD.
7   * @param int $stageId L'identifiant du stage auquel le média est associé.
8   * @return bool Retourne vrai si la création du média est réussie, sinon retourne faux.
9   */
10  function createMedia($date, $fileNameVignette, $fileNameHd, $stageId)
```

```
● ● ●  
1  /**
2   * Récupère les médias associés à un stage spécifique.
3   *
4   * @param int $stageId L'identifiant du stage.
5   * @return array|bool Retourne un tableau contenant des objets Media représentant les médias associés au stage, ou faux en cas d'erreur.
6   */
7  function getMediaByStageId($stageId)
```

```
● ● ●  
1  /**
2   * Récupère les informations d'un média à partir de son identifiant.
3   *
4   * @param int $idMedia L'identifiant du média à récupérer.
5   * @return array|bool Retourne un tableau contenant un objet Media représentant le média correspondant à l'identifiant donné, ou faux en cas d'erreur.
6   */
7  function getMediaById($idMedia)
```

```
● ● ●  
1  /**
2   * Supprime un média de la table 'media' en fonction de son identifiant.
3   *
4   * @param int $id L'identifiant du média à supprimer.
5   * @return bool Retourne vrai si la suppression du média est réussie, sinon retourne faux.
6   */
7  function deleteMediaById($id)
```

- VoyageMemo/app/model/stageClass.php

```
1  /**
2   * Crée un nouveau stage dans la table 'stages'.
3   *
4   * @param string $title Le titre du stage.
5   * @param string $date La date du stage (au format YYYY-MM-DD).
6   * @param int $durationDays La durée du stage en jours.
7   * @param string $description La description du stage.
8   * @param int $travelsId L'identifiant du voyage auquel le stage est associé.
9   * @return bool Retourne vrai si la création du stage est réussie, sinon retourne faux.
10  */
11 function createStage($title, $date, $durationDays, $description, $travelsId)
```

```
1  /**
2   * Récupère les stages associés à un voyage spécifique.
3   *
4   * @param int $travel L'identifiant du voyage.
5   * @return array|bool Retourne un tableau contenant des objets Stage représentant les stages associés au voyage, ou faux en cas d'erreur.
6   */
7 function getStageByTravelId($travel)
```

```
1  /**
2   * Ajoute une activité à une étape dans la table 'stages_has_activity'.
3   *
4   * @param int $idStage L'identifiant du stage auquel ajouter l'activité.
5   * @param int $idActivity L'identifiant de l'activité à ajouter.
6   * @return bool Retourne vrai si l'ajout est réussi, sinon retourne faux.
7   */
8 function addActivity($idStage, $idActivity)
```

```
1  /**
2   * Supprime un stage de la table 'stages' en fonction de son identifiant.
3   *
4   * @param int $id L'identifiant du stage à supprimer.
5   * @return bool Retourne vrai si la suppression du stage est réussie, sinon retourne faux.
6   */
7 function deleteStageById($id)
```



```
1  /**
2   * Récupère les informations d'une étape à partir de son identifiant.
3   *
4   * @param int $idStage L'identifiant du stage à récupérer.
5   * @return array|bool Retourne un tableau contenant un objet Stage représentant le stage correspondant à l'identifiant donné, ou faux en cas d'erreur.
6   */
7  function getStageById($idStage)
```



```
1  /**
2   * Met à jour les informations d'une étape dans la table 'stages'.
3   *
4   * @param int $id L'identifiant du stage à mettre à jour.
5   * @param string $title Le titre du stage.
6   * @param string $date La date du stage (au format YYYY-MM-DD).
7   * @param int $durationDays La durée du stage en jours.
8   * @param string $description La description du stage.
9   * @return bool Retourne vrai si la mise à jour est réussie, sinon retourne faux.
10  */
11 function stageUpdate($id, $title, $date, $durationDays, $description)
```

- VoyageMemo/app/model/stagesHasActivityClass.php



```
1  /**
2   * Supprime l'association entre un stage et une activité dans la table 'stages_has_activity'.
3   *
4   * @param int $stageId L'identifiant du stage.
5   * @param int $activityId L'identifiant de l'activité.
6   * @return bool Retourne vrai si la suppression de l'association est réussie, sinon retourne faux.
7   */
8  function deleteStageActivityAssociationByStageIdAndActivityId($stageId, $activityId)
```

- VoyageMemo/app/model/travelClass.php



```
1  /**
2   * Crée un nouveau voyage dans la base de données avec le titre, le nombre d'affichage et l'ID de l'utilisateur spécifiés.
3   *
4   * @param string $title Le titre du voyage
5   * @param int $numberDisplay Le nombre d'affichages pour le voyage
6   * @param int $userId L'ID de l'utilisateur associé au voyage
7   * @return bool Retourne true si le voyage est créé avec succès, sinon false.
8   */
9  function createTravel($title, $numberDisplay, $userId)
```

```
● ● ●  
1  /**
2   * Récupère tous les voyages depuis la base de données.
3   *
4   * @return array|bool Retourne un tableau d'objets Travel contenant tous les voyages si trouvés,
5   *                     sinon retourne false en cas d'erreur ou si aucun voyage n'est trouvé.
6   */
7  function getAllTravel()
```

```
● ● ●  
1  /**
2   * Récupère les voyages associés à un utilisateur à partir de son identifiant dans la table 'travels'.
3   *
4   * @param int $userId L'identifiant de l'utilisateur.
5   * @return array|bool Retourne un tableau contenant des objets Travel représentant les voyages associés à l'utilisateur, ou faux en cas d'erreur.
6   */
7  function getTravelByUserId($userId)
```

```
● ● ●  
1  /**
2   * Récupère un voyage à partir de son identifiant dans la table 'travels'.
3   *
4   * @param int $id L'identifiant du voyage à récupérer.
5   * @return array|bool Retourne un tableau contenant un objet Travel représentant le voyage trouvé, ou faux si aucun voyage n'est trouvé ou en cas d'erreur.
6   */
7  function getTravelById($id)
```

```
● ● ●  
1  /**
2   * Supprime un voyage de la table 'travels' en fonction de son identifiant.
3   *
4   * @param int $id L'identifiant du voyage à supprimer.
5   * @return bool Retourne vrai si la suppression du voyage est réussie, sinon retourne faux.
6   */
7  function deleteTravelById($id)
```

```
● ● ●  
1  /**
2   * Met à jour le titre d'un voyage dans la table 'travels'.
3   *
4   * @param int $id L'identifiant du voyage à mettre à jour.
5   * @param string $title Le nouveau titre du voyage.
6   * @return bool Retourne vrai si la mise à jour du voyage est réussie, sinon retourne faux.
7   */
8  function travelUpdate($id, $title)
```

```
1  /**
2   * Met à jour le nombre de visite d'un voyage dans la table 'travels'.
3   *
4   * @param int $id L'identifiant du voyage à mettre à jour.
5   * @param string $title Le nouveau titre du voyage.
6   * @return bool Retourne vrai si la mise à jour du voyage est réussie, sinon retourne faux.
7   */
8  function travelNumberDisplayUpdate($id, $numberDisplay)
```

- VoyageMemo/app/model/usersClass.php

```
1  /**
2   * Crée un nouvel utilisateur dans la base de données avec le login et le mot de passe spécifiés.
3   *
4   * @param string $Login Le nom d'utilisateur du nouvel utilisateur
5   * @param string $password Le mot de passe du nouvel utilisateur
6   * @return bool Retourne true si l'utilisateur est créé avec succès, sinon false.
7   */
8  function createUser($login, $password)
```

```
1  /**
2   * Vérifie si un utilisateur existe dans la table 'users' en fonction de son nom d'utilisateur.
3   *
4   * @param string $Login Le nom d'utilisateur à vérifier.
5   * @return int/bool Retourne le nombre d'utilisateurs trouvés avec le nom d'utilisateur donné, ou faux en cas d'erreur.
6   */
7  function verifyUserExiste($login)
```

```
1  /**
2   * Récupère un utilisateur à partir de son nom d'utilisateur (Login) dans la table 'users'.
3   *
4   * @param string $Login Le nom d'utilisateur à rechercher.
5   * @return array/bool Retourne un tableau contenant un objet User représentant l'utilisateur trouvé, ou faux en cas d'erreur ou si aucun utilisateur n'est trouvé.
6   */
7  function getUser($login)
```

- VoyageMemo/app/model/db/
Contient la base de donnée.
- VoyageMemo/app/public/css/ :
Contient les fichiers CSS.
- VoyageMemo/app/public/img/
Contient les images du site web.
- VoyageMemo/app/view/
Contient les fichiers vus de l'application.
- VoyageMemo/app/index.php
Le point d'entrée principal de l'application.
- VoyageMemo/documentation/

Contient la documentation du projet.

Installation du projet

Programmes nécessaires

- Un serveur maria DB
- PHP GD
- PHP 8.1

Connexion à la base de donnée

Pour se connecter il faut aller dans le fichier comparam.php qui se trouve à cette emplacement « *VoyageMemo/app/model/config/* ». Une fois dans le fichier ouvert, il faut changer l'utilisateur et le mot de passe.

Outils utilisés

Bootstrap - 5.2.3

J'ai utilisé le Framework Bootstrap pour concevoir mon site web, et ce Framework assure que le site soit responsive.

GitHub

GitHub m'a permis de faire du versionning de mon projet. Il permet de publier le projet sur un serveur distant et de revenir aux anciennes versions si besoin.

ChatGPT - 3.5

ChatGPT m'a permis de corriger l'orthographe de mon projet. Il m'a aussi permis de commenter une partie de mon code et de m'aider lorsque j'étais bloqué sur un quelque chose.

Modèle conceptuel de données

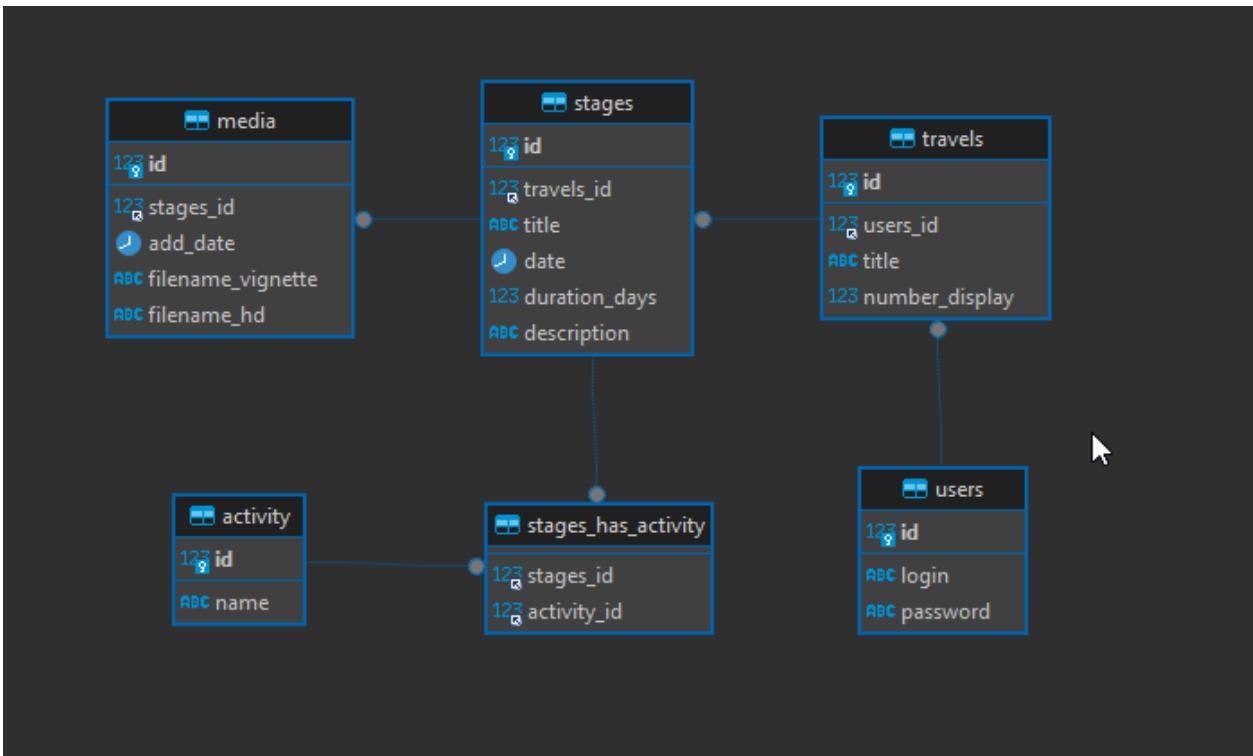


Figure 3. MVC de la base de données

Dictionnaire de données

users

Nom	Type	Taille	Null	Défaut
id	Int	11	NON	Auto-Incrément
login	Varchar	100	NON	
password	Varchar	90	NON	

travels

Nom	Type	Taille	Null	Défaut
id	Int	11	NON	Auto-Incrément
users_id	Int	11	NON	
title	Varchar	255	NON	
Number_display	Int	11	NON	

stages

Nom	Type	Taille	Null	Défaut
id	Int	11	NON	Auto-Incrément
travels_id	Int	11	NON	
title	Varchar	255	NON	

date	Date		OUI
duration_days	Int	2	OUI
description	tinytext		OUI

media

Nom	Type	Taille	Null	Défaut
id	Int	11	NON	Auto-Incrément
stages_id	Int	11	NON	
add_date	datetime		NON	
filename_vignette	Varchar	150	OUI	
filename_hd	Varchar	150	OUI	

activity

Nom	Type	Taille	Null	Défaut
Id	Int	11	NON	Auto-Incrément
Name	Varchar	45	NON	

travels

Nom	Type	Taille	Null	Défaut
stages_id	Int	11	NON	
activity_id	Int	11	NON	

Test

Plan de tests

N° test	Auteur	Scénario	Résultat attendu
T_001	GF	<u>Créer un utilisateur existant</u> <ol style="list-style-type: none"> 1. Choisir « créer un compte » en cliquant sur l'image de profile dans la barre de navigation 2. Saisir le login (Xayah) 3. Saisir un mot de passe (123456789) 4. Saisir la confirmation de mot de passe 5. Cliquer sur « s'inscrire » 	Un message d'erreur s'affiche.

T_002	GF	<u>Création d'un nouvel utilisateur</u> 1. Choisir « connexion » en cliquant sur l'image de profile dans la barre de navigation 2. Saisir le login 3. Saisir le mot de passe du compte 4. Cliquer sur « connexion »	Un message indiquant que le client a été créé avec succès et les informations du client sont ajouté à la bdd
T_003	GF	<u>Connexion d'un utilisateur</u> 1. Choisir « connexion » en cliquant sur l'image de profile dans la barre de navigation 2. Saisir le login (Xayah) 3. Saisir le mot de passe du compte (123456789) 4. Cliquer sur « connexion »	Redirection vers la page profile de l'utilisateur et met l'utilisateur dans la session.
T_004	GF	<u>Créer un voyage</u> 1. Choisir « nouveau voyage » dans la page accueil 2. Saisir le titre du voyage (Voyage Japon)	Les informations du voyage ont été ajouté à la bdd
T_005	GF	<u>Créer une étape (toutes les informations)</u> 1. Choisir « nouvelle étape » dans la page détail 2. Saisir le titre de l'étape 3. Saisir la description de l'étape 4. Saisir la date du début de l'étape 5. Saisir la date de fin de l'étape 6. Choisir une ou des activité(s) 7. Choisir une ou plusieurs image(s)	Affiche un message comme quoi l'étape a été ajouté et les informations de l'étape ont été ajouté à la bdd
T_006	GF	<u>Supprimer un voyage</u> 1. Choisir « supprimer voyage » dans la page détail	Supprime le voyage et tous ce qui est lié de la bdd et redirige vers la page profil.

T_007	GF	<u>Supprimer une étape</u> 1. Choisir « supprimer » dans la page détail	Supprime l'étape et tous ce qui est lié de la bdd et reste sur la page détail.
T_008	GF	<u>Modifier un voyage</u> 1. Choisir « Modifier voyage » dans la page détail 2. Saisir les informations à modifier 3. Valider les modification	Modifie les infos du voyage de la bdd.
T_009	GF	<u>Modifier une étape</u> 1. Choisir « Modifier » dans la page détail 2. Saisir les informations à modifier 3. Valider les modifications	Modifie les infos de l'étape de la bdd.
T_010	GF	<u>Déconnexion utilisateur</u> 2. Choisir « déconnexion » en cliquant sur l'image de profile dans la barre de navigation	La session est supprimé

Rapport de tests

	J-1	J-2	J-3	J-4	J-5	J-6	J-7	J-8	J-9	J-10	J-11
T_001											
T_002											
T_003											
T_004											
T_005											
T_006											
T_007											
T_008											
T_009											
T_010											

Légende :

Validé	
Echoué	
Non réalisé	

N° test	Testeur	Date	Résultat obtenu	OK
T_001	GF	25/04/2024 (j-2)	Je n'ai pas obtenu un message signalant qu'une erreur est survenue et l'utilisateur a été ajouté	Non
		29/04/2024 (J-3)	Le message d'erreur s'affiche et l'utilisateur n'est pas ajouté à la bdd	Oui
		30/04/2024 (J-4)	Le message d'erreur s'affiche et l'utilisateur n'est pas ajouté à la bdd	Oui
		02/30/2024 (J-5)	Le message d'erreur s'affiche et l'utilisateur n'est pas ajouté à la bdd	Oui
		06/30/2024 (J-6)	Le message d'erreur s'affiche et l'utilisateur n'est pas ajouté à la bdd	Oui
		07/30/2024 (J-7)	Le message d'erreur s'affiche et l'utilisateur n'est pas ajouté à la bdd	Oui
		08/30/2024 (J-8)	Le message d'erreur s'affiche et l'utilisateur n'est pas ajouté à la bdd	Oui
		13/30/2024 (J-9)	Le message d'erreur s'affiche et l'utilisateur n'est pas ajouté à la bdd	Oui
		14/30/2024 (J-10)	Le message d'erreur s'affiche et l'utilisateur n'est pas ajouté à la bdd	Oui
		15/04/2024 (j-11)	Le message d'erreur s'affiche et l'utilisateur n'est pas ajouté à la bdd	Oui
T_001	GF	25/04/2024 (j-2)	Un message indiquant que le client a été créé avec succès et les informations du client sont ajouté à la bdd.	Oui
		29/04/2024 (J-3)	Un message indiquant que le client a été créé avec succès et les informations du client sont ajouté à la bdd.	Oui
		30/04/2024 (J-4)	Un message indiquant que le client a été créé avec succès et les informations du client sont ajouté à la bdd.	Oui
		02/30/2024 (J-5)	Un message indiquant que le client a été créé avec succès et les informations du client sont ajouté à la bdd.	Oui
		06/30/2024 (J-6)	Un message indiquant que le client a été créé avec succès et les informations du client sont ajouté à la bdd.	Oui
		07/30/2024 (J-7)	Un message indiquant que le client a été créé avec succès et les informations du client sont ajouté à la bdd.	Oui

		08/30/2024 (J-8)	Un message indiquant que le client a été créé avec succès et les informations du client sont ajouté à la bdd.	Oui
		13/30/2024 (J-9)	Un message indiquant que le client a été créé avec succès et les informations du client sont ajouté à la bdd.	Oui
		14/30/2024 (J-10)	Un message indiquant que le client a été créé avec succès et les informations du client sont ajouté à la bdd.	Oui
		15/30/2024 (J-11)	Un message indiquant que le client a été créé avec succès et les informations du client sont ajouté à la bdd.	Oui
T_003	GF	25/04/2024 (j-2)	J'ai été redirigé vers la page profile et mis en session.	Oui
		29/04/2024 (J-3)	J'ai été redirigé vers la page profile et mis en session.	Oui
		30/04/2024 (J-4)	J'ai été redirigé vers la page profile et mis en session.	Oui
		02/05/2024 (J-5)	J'ai été redirigé vers la page profile et mis en session.	Oui
		06/05/2024 (J-6)	J'ai été redirigé vers la page profile et mis en session.	Oui
		07/05/2024 (J-7)	J'ai été redirigé vers la page profile et mis en session.	Oui
		08/05/2024 (J-8)	J'ai été redirigé vers la page profile et mis en session.	Oui
		13/05/2024 (J-9)	J'ai été redirigé vers la page profile et mis en session.	Oui
		14/05/2024 (J-10)	J'ai été redirigé vers la page profile et mis en session.	Oui
		15/05/2024 (j-11)	J'ai été redirigé vers la page profile et mis en session.	Oui
		29/04/2024 (J-3)	Le voyage a été ajouté dans la bdd.	Oui
		30/04/2024 (J-4)	Le voyage a été ajouté dans la bdd.	Oui
T_004	GF	02/05/2024 (J-5)	Le voyage a été ajouté dans la bdd.	Oui
		06/05/2024 (J-6)	Le voyage a été ajouté dans la bdd.	Oui
		07/05/2024 (J-7)	Le voyage a été ajouté dans la bdd.	Oui
		08/05/2024 (J-8)	Le voyage a été ajouté dans la bdd et un message confirmant la création du voyage s'affiche.	Oui

		13/05/2024 (J-9)	Le voyage a été ajouté dans la bdd et un message confirmant la création du voyage s'affiche.	Oui
		14/05/2024 (J-10)	Le voyage a été ajouté dans la bdd et un message confirmant la création du voyage s'affiche.	Oui
		15/05/2024 (j-11)	Le voyage a été ajouté dans la bdd et un message confirmant la création du voyage s'affiche.	Oui
T_005	GF	30/04/2024 (J-4)	Un message s'affiche confirmant l'ajout de l'étape et les informations de l'étape s'ajoutent dans la bdd.	Oui
		02/05/2024 (J-5)	Un message s'affiche confirmant l'ajout de l'étape et les informations de l'étape s'ajoutent dans la bdd.	Oui
		06/05/2024 (J-6)	Un message s'affiche confirmant l'ajout de l'étape et les informations de l'étape s'ajoutent dans la bdd.	Oui
		07/05/2024 (J-7)	Un message s'affiche confirmant l'ajout de l'étape et les informations de l'étape s'ajoutent dans la bdd.	Oui
		08/05/2024 (J-8)	Un message s'affiche confirmant l'ajout de l'étape et les informations de l'étape s'ajoutent dans la bdd.	Oui
		13/05/2024 (J-9)	Un message s'affiche confirmant l'ajout de l'étape et les informations de l'étape s'ajoutent dans la bdd.	Oui
		14/05/2024 (J-10)	Un message s'affiche confirmant l'ajout de l'étape et les informations de l'étape s'ajoutent dans la bdd.	Oui
		15/05/2024 (j-11)	Un message s'affiche confirmant l'ajout de l'étape et les informations de l'étape s'ajoutent dans la bdd.	Oui
T_006	GF	02/05/2024 (J-5)	Toutes les infos sont supprimées de la bdd, mais les médias ne sont pas supprimés des fichiers.	Non
		06/05/2024 (J-6)	Toutes les infos sont supprimées bdd / fichier.	Oui
		07/05/2024 (J-7)	Toutes les infos sont supprimées bdd / fichier.	Oui
		08/05/2024 (J-8)	Toutes les infos sont supprimées bdd / fichier.	Oui
		13/05/2024 (J-9)	Toutes les infos sont supprimées bdd / fichier.	Oui
		14/05/2024 (J-10)	Toutes les infos sont supprimées bdd / fichier.	Oui
		15/05/2024 (j-11)	Toutes les infos sont supprimées bdd / fichier.	Oui
T_007	GF	02/05/2024 (J-5)	Toutes les infos sont supprimées de la bdd, mais les médias ne sont pas supprimés des fichiers.	Non
		06/05/2024 (J-6)	Toutes les infos sont supprimées de la bdd, mais les médias ne sont pas supprimés des fichiers.	Oui
		07/05/2024 (J-7)	Toutes les infos sont supprimées de la bdd, mais les médias ne sont pas supprimés des fichiers.	Oui

		08/05/2024 (J-8)	Toutes les infos sont supprimées de la bdd, mais les médias ne sont pas supprimés des fichiers.	Oui
		13/05/2024 (J-9)	Toutes les infos sont supprimées de la bdd, mais les médias ne sont pas supprimés des fichiers.	Oui
		14/05/2024 (J-10)	Toutes les infos sont supprimées de la bdd, mais les médias ne sont pas supprimés des fichiers.	Oui
		15/05/2024 (j-11)	Toutes les infos sont supprimées de la bdd, mais les médias ne sont pas supprimés des fichiers.	Oui
T_008	GF	06/30/2024 (J-6)	Les infos ont été modifiée dans la bdd et un massage confirmant les modifications est affiché.	Oui
		07/30/2024 (J-7)	Les infos ont été modifiée dans la bdd et un massage confirmant les modifications est affiché.	Oui
		08/30/2024 (J-8)	Les infos ont été modifiée dans la bdd et un massage confirmant les modifications est affiché.	Oui
		13/30/2024 (J-9)	Les infos ont été modifiée dans la bdd et un massage confirmant les modifications est affiché.	Oui
		14/30/2024 (J-10)	Les infos ont été modifiée dans la bdd et un massage confirmant les modifications est affiché.	Oui
		15/04/2024 (j-11)	Les infos ont été modifiée dans la bdd et un massage confirmant les modifications est affiché.	Oui
T_009	GF	06/30/2024 (J-6)	Les nouvelle activité sélectionné ne se sauvegarde pas et les activités désélectionné ne se supprime pas	Non
		07/30/2024 (J-7)	Les infos ont été modifiée dans la bdd et un massage confirmant les modifications est affiché	Oui
		08/30/2024 (J-8)	Les infos ont été modifiée dans la bdd et un massage confirmant les modifications est affiché	Oui
		13/30/2024 (J-9)	Les infos ont été modifiée dans la bdd et un massage confirmant les modifications est affiché	Oui
		14/30/2024 (J-10)	Les infos ont été modifiée dans la bdd et un massage confirmant les modifications est affiché	Oui
		15/04/2024 (j-11)	Les infos ont été modifiée dans la bdd et un massage confirmant les modifications est affiché	Oui
T_010	GF	25/04/2024 (j-2)	Ma session a été supprimé.	Oui
		29/04/2024 (J-3)	Ma session a été supprimé.	Oui
		30/04/2024 (J-4)	Ma session a été supprimé.	Oui
		02/05/2024 (J-5)	Ma session a été supprimé.	Oui
		06/05/2024 (J-6)	Ma session a été supprimé.	Oui

	07/05/2024 (J-7)	Ma session a été supprimé.	Oui
	08/05/2024 (J-8)	Ma session a été supprimé.	Oui
	13/05/2024 (J-9)	Ma session a été supprimé.	Oui
	14/05/2024 (J-10)	Ma session a été supprimé.	Oui
	15/05/2024 (j-11)	Ma session a été supprimé.	Oui

Conclusion

Difficultés rencontrées

Gestion de la taille des images

J'ai mal compris les formules pour redimensionner les images données dans le cahier des charges, elles n'étaient pas en accord avec celles que j'avais imaginées. J'ai donc mis sur papier les formules pour les comparer et choisir celle qui me paraissait la meilleure.

Formule de redimensionnement des images :

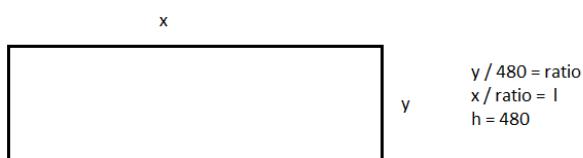
Largeur définit : 640
Hauteur définit : 480

x : largeur
y : hauteur

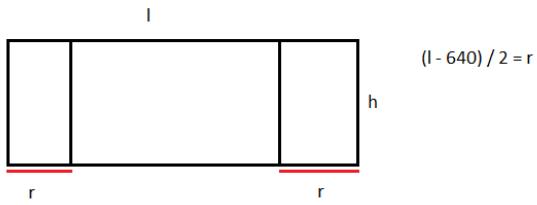
l : nouvelle largeur
h : nouvelle hauteur

r : bout à rogner

Largeur plus grande que la hauteur



Si non 4/3



Hauteur plus grande que la largeur

$x / 640 = \text{ratio}$
 $y / \text{ratio} = h$
 $l = 640$

Si non 4/3

$(h-480)/2 = r$

Figure 4. Formule pour redimensionner les images plus grande

Largeur définit : 640	x : largeur	I : nouvelle largeur	r : bout à rogner
Hauteur définit : 480	y : hauteur	h : nouvelle hauteur	
Largeur source plus petite que 640			Hauteur source plus petite que 480
$640 / x = \text{ratio}$			$480 / y = \text{ratio}$
$y * \text{ratio} = h$			$x * \text{ratio} = I$
$x = 640$			$h = 640$

Figure 5. Formule pour redimensionner les images plus petites

Une fois l'image plus petite redimensionnée on la rogne comme dans la figure 4.

Amélioration possible

Pour ce projet, je vois plusieurs améliorations possibles :

Comme première amélioration, ce serait de mettre en place un système de suppression de compte utilisateur, car actuellement, les utilisateurs qui créent un compte ne peuvent pas le supprimer.

Pour la seconde amélioration, ce serait d'améliorer la sécurité des comptes utilisateurs en ajoutant un système d'authentification à double facteur lors de la connexion et de la création de compte. On pourrait aussi ajouter des captchas sur les formulaires.

La troisième amélioration que je vois pour le site serait d'ajouter un système de compte administrateur. Les administrateurs pourraient ajouter des activités supplémentaires et supprimer des utilisateurs ou des voyages.

Je vois aussi des améliorations mineures qui pourraient améliorer l'expérience utilisateur, par exemple rajouter une fonction pour "liker" des voyages et un système de commentaires pour les voyages.

Bilan personnel

J'ai aimé réaliser ce projet. Il m'a permis de me rendre compte à quel point les projets dans la vie professionnelle ne se centrent pas seulement sur le développement. J'ai dû planifier des tâches et suivre un planning. Grâce à ce projet, j'ai pu mettre en pratique toutes les connaissances que j'ai acquises durant ma formation. Je suis satisfait de ce que j'ai produit, mais je pense que j'aurais pu faire mieux.

Remerciements

Je tiens à remercier :

- M. Garchery pour son soutien et son aide tout au long du projet.
- Leandro Antunes pour m'avoir aidé avec un problème de requête.

Planification

Prévisionnel

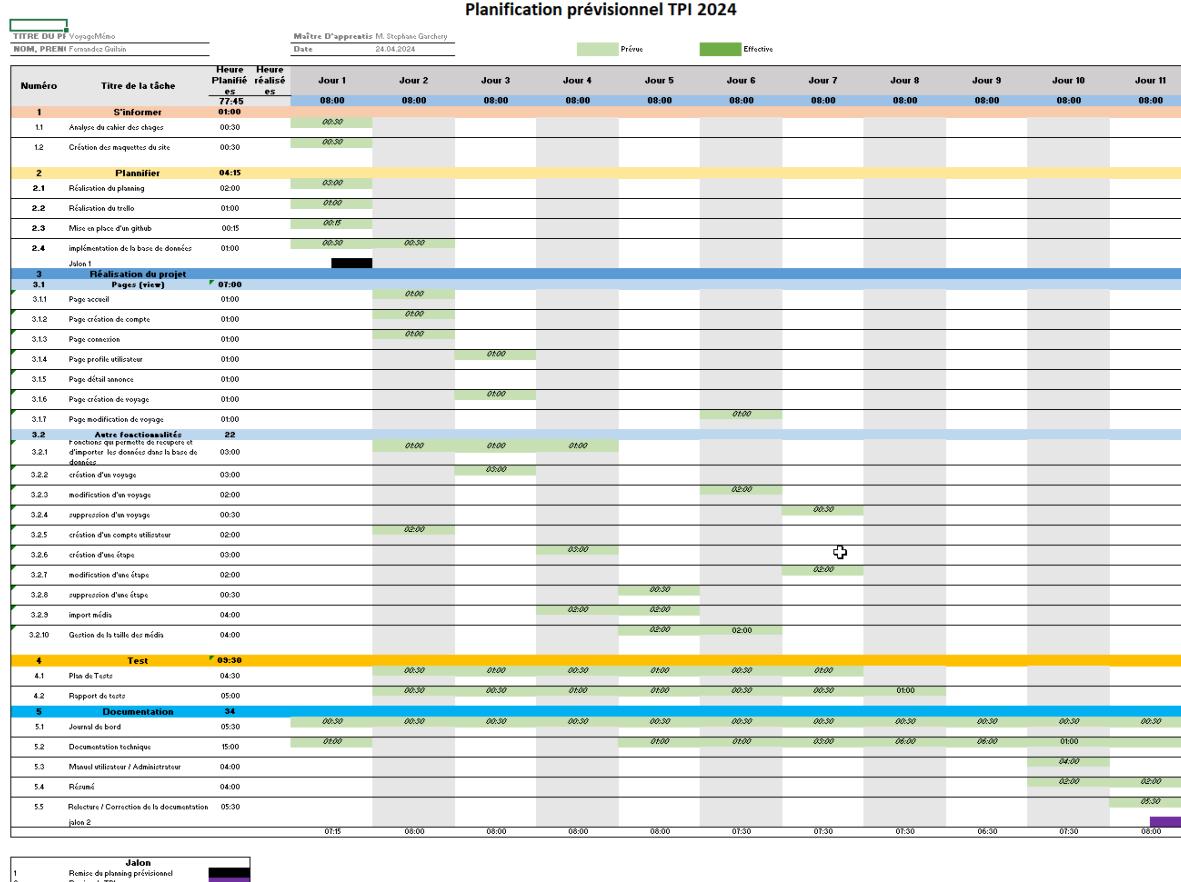


Figure 6. Planning prévisionnel

Effective

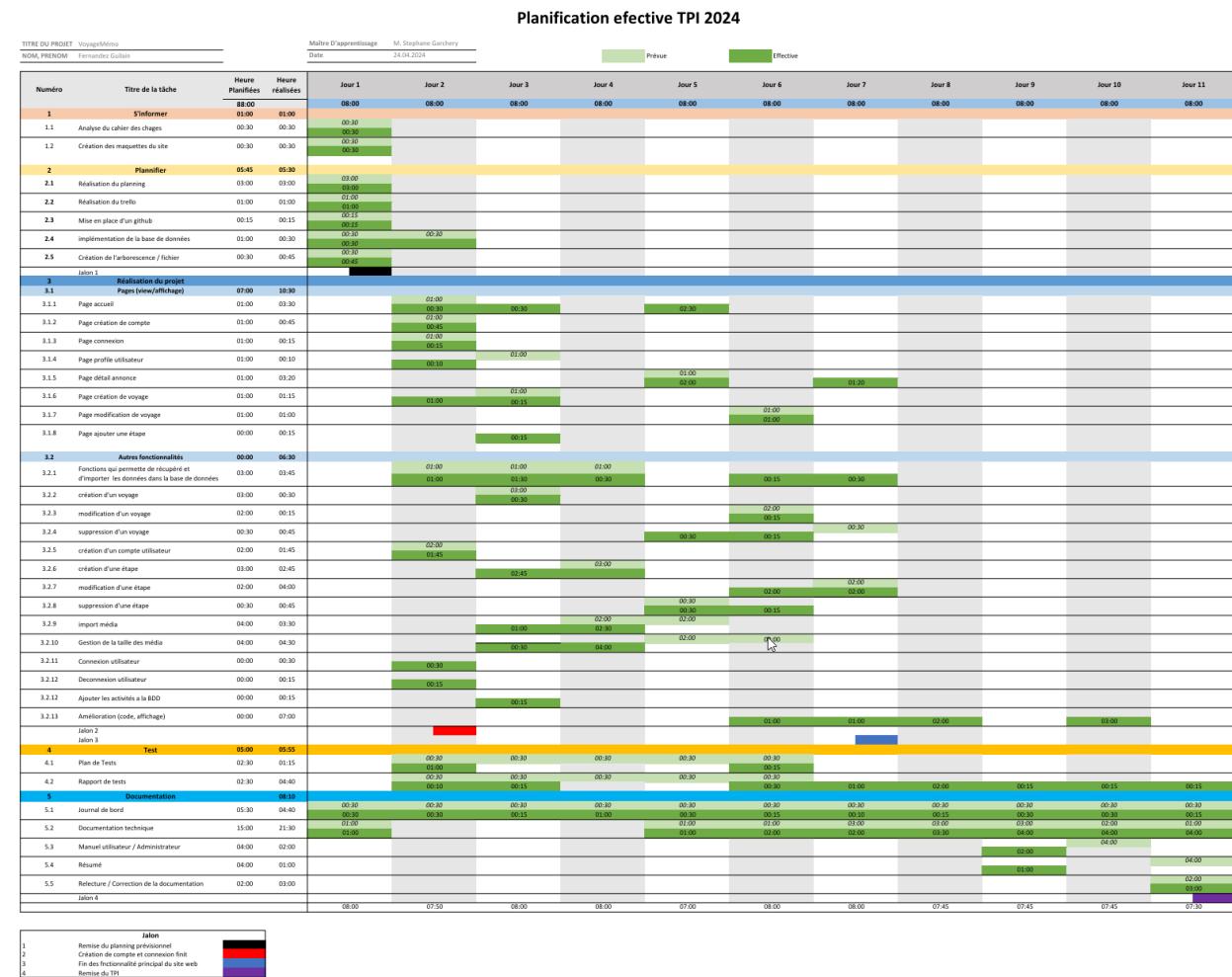


Figure 7. Planning effectif

Analyse

On peut remarquer que lors des trois premiers jours du projet, j'ai pris une très bonne avance car les premières tâches ont pris moins de temps que prévu. Cependant, une fois arrivé à la gestion des médias, j'ai perdu cet avantage, ce qui m'a fait respecter le calendrier de mon planning initial.

Table des illustrations

Figure 1.méthode en 6 étapes.....	2
Figure 2. Arborescence des fichiers du site web.....	7
Figure 3. MVC de la base de données	15
Figure 4. Formule pour redimensionner les images plus grande	23
Figure 5. Formule pour redimensionner les image plus petite.....	24
Figure 6. Planning prévisionnel	25

Bibliographie

<https://chatgpt.com/>

<https://www.php.net/>

<https://getbootstrap.com/docs/5.2/getting-started/introduction/>

<https://www.w3schools.com/>

<https://stackoverflow.com/>

<https://sql.sh/>

<https://developer.mozilla.org/fr/>

Glossaire

Identifiant : Terme utiliser pour désigner les informations de l'utilisateur.

Versionning : C'est un système qui permet de gérer les différentes versions d'un projet.

Bdd : Abréviation de « base de données ».

Authentification à double facteur : C'est une méthode permettant d'assurer l'authenticité de la personne derrière un compte. La personne doit présenter 2 preuve différente pour prouver son identité.

Annexe

En annexe de la documentation technique, j'ai ajouté le planning prévisionnel et effectif, car on a du mal à les lire.

Code source

Selected files

32 printable files

VoyageMemo/app/controller/media/mediaFunction.php
VoyageMemo/app/controller/stage/deleteStage.php
VoyageMemo/app/controller/stage/modifyStage.php
VoyageMemo/app/controller/stage/newStage.php
VoyageMemo/app/controller/travel/deleteTravel.php
VoyageMemo/app/controller/travel/modifyTravel.php
VoyageMemo/app/controller/travel/newTravel.php
VoyageMemo/app/controller/user/newAccount.php
VoyageMemo/app/controller/user/userFunction.php
VoyageMemo/app/controller/user/userLogin.php
VoyageMemo/app/controller/controller.php
VoyageMemo/app/controller/user/userDisconnection.php
VoyageMemo/app/model/config/comparam.php
VoyageMemo/app/model/config/database.php
VoyageMemo/app/model/db/dump-VoyageMemo-202405151452.sql
VoyageMemo/app/model/usersClass.php
VoyageMemo/app/model/activityClass.php
VoyageMemo/app/model/mediaClass.php
VoyageMemo/app/model/stageClass.php
VoyageMemo/app/model/stagesHasActivityClass.php
VoyageMemo/app/model/travelClass.php
VoyageMemo/app/public/css/all.css
VoyageMemo/app/view/home.php
VoyageMemo/app/view/loginUser.php
VoyageMemo/app/view/modifyStage.php
VoyageMemo/app/view/modifyTravel.php
VoyageMemo/app/view/newStage.php
VoyageMemo/app/view/newTravel.php
VoyageMemo/app/view/newUserAccount.php
VoyageMemo/app/view/profil.php
VoyageMemo/app/index.php
VoyageMemo/app/view/detailTravel.php

VoyageMemo/app/controller/media/mediaFunction.php

```
1 <?php
2
3 /**
4  * TITRE : VoyageMémo
5  * Auteur : Guilain Fernandez
6  * Description : la fonction qui vas créer une image hd a partir d'une image de base
7  * Date : 29/04/2024
8 */
9
10 /**
11  * Redimensionne une image aux dimensions spécifiées tout en conservant son ratio d'
12 aspect.
13 *
14 * @param string $originalpicture Le chemin de l'image source à redimensionner.
15 * @param int $width La largeur souhaitée de l'image redimensionnée.
16 * @param int $height La hauteur souhaitée de l'image redimensionnée.
17 * @return GdImage|bool Retourne une ressource image de l'image redimensionnée ou false en
18 cas d'erreur.
19 */
20 function resize($originalpicture, $width, $height)
21 {
    // Crée une image depuis l'image source
```

```

22     $sourceImage = imagecreatefromstring(file_get_contents($originalpicture));
23
24     // Obtient la largeur et la hauteur de l'image source
25     $sourceWidth = imagesx($sourceImage);
26     $sourceHeight = imagesy($sourceImage);
27
28     // Calcule les nouvelles dimensions tout en conservant le ratio d'aspect de l'image
29     // source
30     if ($sourceWidth > $sourceHeight) {
31         $ratioH = $sourceHeight / $height;
32         $newWidth = $sourceWidth / $ratioH;
33         $newHeight = $height;
34     } elseif ($sourceWidth < $sourceHeight) {
35         $ratioW = $sourceWidth / $width;
36         $newHeight = $sourceHeight / $ratioW;
37         $newWidth = $width;
38     } elseif ($sourceWidth < $width) {
39         $ratioW = $width / $sourceWidth;
40         $newHeight = $sourceHeight * $ratioW;
41         $newWidth = $width;
42     } elseif ($sourceHeight < $height) {
43         $ratioH = $height / $sourceHeight;
44         $newWidth = $sourceWidth * $ratioH;
45         $newHeight = $height;
46     }
47
48     // Convertit les nouvelles dimensions en entiers
49     $newWidth = intval($newWidth);
50     $newHeight = intval($newHeight);
51
52     // Crée une image vide avec les nouvelles dimensions
53     $newImage = imagecreatetruecolor($newWidth, $newHeight);
54
55     // Copie et redimensionne l'image source vers la nouvelle image
56     imagecopyresampled($newImage, $sourceImage, 0, 0, 0, 0, $newWidth, $newHeight,
57     $sourceWidth, $sourceHeight);
58
59     // Libère la mémoire de l'image source
60     imagedestroy($sourceImage);
61
62     // Retourne l'image redimensionnée
63     return $newImage;
64 }
65 /**
66 * Découpe une image centrale selon les dimensions spécifiées et enregistre le résultat
67 * dans un fichier.
68 *
69 * @param mixed $sourceImage La ressource de l'image source à découper.
70 * @param string $destinationPath Le chemin de destination où enregistrer l'image
71 * découpée.
72 * @param int $width La largeur souhaitée de l'image découpée.
73 * @param int $height La hauteur souhaitée de l'image découpée.
74 * @return void
75 */
76 function pictureTrim($sourceImage, $destinationPath, $width, $height)
77 {
78     // Obtient les dimensions de l'image source
79     $sourceWidth = imagesx($sourceImage);
80     $sourceHeight = imagesy($sourceImage);

```

```

79     if ($sourceWidth > $sourceHeight) {
80         // largeur
81         $cut = ($sourceWidth - $width) / 2;
82         $cut = intval($cut);
83         $newImage = imagecreatetruecolor($width, $sourceHeight); // Crée une nouvelle
image vide
84         $srcX = $cut;
85         $srcY = 0;
86     } else {
87         // hauteur
88         $cut = ($sourceHeight - $height) / 2;
89         $cut = intval($cut);
90         $newImage = imagecreatetruecolor($sourceWidth, $height); // Crée une nouvelle
image vide
91         $srcX = 0;
92         $srcY = $cut;
93     }
94
95     // Convertit les coordonnées en entiers
96     $srcX = intval($srcX);
97     $srcY = intval($srcY);
98
99     // Copie et redimensionne la zone centrale de l'image source dans la nouvelle image
100    imagecopyresampled($newImage, $sourceImage, 0, 0, $srcX, $srcY, $sourceWidth,
$sourceHeight, $sourceWidth, $sourceHeight); // Crée une band en copiant la partie
centrale de l'image
101
102    // Enregistre la nouvelle image découpée au format JPEG avec une qualité de 100
103    imagejpeg($newImage, $destinationPath, 100);
104
105    // Libère la mémoire de l'image source et de l'image découpée
106    imagedestroy($sourceImage);
107    imagedestroy($newImage);
108 }
109

```

VoyageMemo/app/controller/stage/deleteStage.php

```

1 <?php
2
3 /**
4  * TITRE : VoyageMémo
5  * Auteur : Guilain Fernandez
6  * Description : permet de supprimer une étape
7  * Date : 02/05/2024
8 */
9
10 // Déclaration des variables
11 $stageId = "";
12
13 if (isset($_POST['deleteStage'])) {
14     $stageId = filter_input(INPUT_POST, 'stageId', FILTER_DEFAULT);
15
16     // Pour chaque média associé à l'étape, supprime les fichiers de vignette et de haute
définition
17     foreach (getMediaByStageId($stageId) as $key => $value) {
18         unlink("./public/img/uploads/vignettes/" . $value->filenameVignette);
19         unlink("./public/img/uploads/hd/" . $value->filenameHd);
20     }
21     // Supprime l'étape de la base de données

```

```
22     deleteStageById($stageId);  
23 }  
24
```

VoyageMemo/app/controller/stage/modifyStage.php

```
1 <?php  
2  
3 /**  
4  * TITRE : VoyageMémo  
5  * Auteur : Guilain Fernandez  
6  * Description : Ce script permet de contrôler les informations entrées par l'utilisateur  
et de modifier l'étape sélectionnée du voyage en conséquence.  
7  * Date : 06/05/2024  
8  */  
9  
10 // Déclaration des variables  
11 $stageTitle = ""; // Titre de l'étape  
12 $stageDescription = ""; // Description de l'étape  
13 $startDate = "00-00-0000"; // Date de début de l'étape  
14 $endDate = "00-00-0000"; // Date de fin de l'étape  
15 $activity = array(); // Tableau des activités associées à l'étape  
16 $mediaEdit = array(); // Tableau des médias édités à supprimer  
17 $media = array(); // Tableau des nouveaux médias à ajouter  
18 $uploadOk = 1; // Indicateur pour le téléchargement de médias  
19 $destinationMediaVignette = "././public/img/uploads/vignettes/"; // Chemin du dossier où  
seront stockées les vignettes  
20 $destinationMediaHd = "././public/img/uploads/hd/"; // Chemin du dossier où seront  
stockées les images haute définition  
21 $destinationMediaTemp = "././public/img/uploads/temp/"; // Chemin du dossier où seront  
stockées les images temporaires  
22 $todayDate = date("Y-m-d"); // Date actuelle  
23 $ratio = 4 / 3; // Ratio d'aspect des images (4:3)  
24  
25 // Vérification si le formulaire de modification d'étape a été soumis  
26 if (isset($_POST['modifyStage'])) {  
27     EDatabase::beginTransaction(); // Début de la transaction  
28  
29     // Récupération des données du formulaire  
30     $stageTitle = filter_input(INPUT_POST, 'stageTitle', FILTER_DEFAULT);  
31     $stageDescription = filter_input(INPUT_POST, 'stageDescription', FILTER_DEFAULT);  
32     $startDate = filter_input(INPUT_POST, 'startDate', FILTER_DEFAULT);  
33     $endDate = filter_input(INPUT_POST, 'endDate', FILTER_DEFAULT);  
34     $stageId = filter_input(INPUT_POST, 'stageId', FILTER_DEFAULT);  
35     $activity = $_POST['activity'];  
36     $media = $_FILES['stageMedia'];  
37     $mediaEdit = $_POST['media'];  
38  
39     // Vérification que les champs obligatoires ne sont pas vides  
40     if ($stageTitle != "") {  
41         if ($startDate != null && $endDate != null) {  
42             if ($activity != null) {  
43                 $dateTimeStart = new DateTime($startDate);  
44                 $dateTimeEnd = new DateTime($endDate);  
45  
46                 // récupère les données de l'étape  
47                 $stage = getStageById($stageId);  
48  
49             // Vérification que la date de début est antérieure ou égale à la date de  
fin
```

```

50         if ($startDate <= $endDate) {
51             $durationDays = $endTime->diff($startTime);
52
53             // Mise à jour de l'étape du voyage dans la base de données
54             stageUpdate($stageId, $stageTitle, $startDate, $durationDays->days,
55             $stageDescription);
56
57             // Récupération des activités associées à cette étape
58             $stageActivity = getActivityByStageId($stageId);
59
60             $travel = getTravelById($stage[0]->travelId);
61             $numberDisplay = $travel[0]->numberDisplay;
62             $numberActivity = ($numberDisplay - count($stageActivity)) +
63             count($activity);
64
65             // Ajout des nouvelles activités sélectionnées par l'utilisateur
66             foreach ($activity as $key => $valueActivity) {
67                 foreach ($stageActivity as $key => $valueStageActivity) {
68                     if ($valueActivity != $valueStageActivity->id) {
69                         $addActivity = true;
70                     } else {
71                         $addActivity = false;
72                         break;
73                     }
74                 }
75                 if ($addActivity == true) {
76                     addActivity($stageId, $valueActivity);
77                 }
78             }
79             // Suppression des activités qui ont été désélectionnées par l'
utilisateur
80             foreach ($stageActivity as $key => $valueStageActivity) {
81                 foreach ($activity as $key => $valueActivity) {
82                     if ($valueStageActivity->id == $valueActivity) {
83                         $activityDelete = false;
84                         break;
85                     } else {
86                         $activityDelete = true;
87                     }
88                 }
89                 if ($activityDelete == true) {
90                     deleteStageActivityAssociationByStageIdAndActivityId($stageId,
$valueStageActivity->id);
91                 }
92             }
93
94             // Gestion des médias
95             if ($mediaEdit != null) {
96                 // Suppression des médias édités et de leurs enregistrements dans
la base de données
97                 foreach ($mediaEdit as $key => $value) {
98                     $deleteMedia = getMediaById($value);
99                     if (deleteMediaById($value)) {
100                         // Supprime les médias dans les fichiers
101                         unlink($destinationMediaHd . $deleteMedia[0]->filenameHd);
102                         unlink($destinationMediaVignette . $deleteMedia[0]->
filenameVignette);
103                     }
104             }

```

```

105 }
106
107 // Traitement des nouveaux médias ajoutés
108 try {
109     foreach ($media['error'] as $key => $v) {
110         if ($media['error'][$key] != 0) {
111             throw new Exception("file error", 1);
112             die;
113         }
114         $uniqid = uniqid(); // nom unique pour l'image
115         $imageFileType = strtolower(pathinfo($media["name"][$key],
116 PATHINFO_EXTENSION)); // récupère l'extension du fichier
117         $media["name"][$key] = $uniqid . "." . $imageFileType; // renomme l'image avec l'uniqid
118         $nameVignette = basename($uniqid . "_vignette." .
119 $imageFileType); // créer un nom pour la vignette avec l'uniqid
120         $nameHd = basename($uniqid . "_hd." . $imageFileType); // créer un nom pour l'image hd avec l'uniqid
121         $saveFileVignette = $destinationMediaVignette . $nameVignette;
122         $saveFileHd = $destinationMediaHd . $nameHd;
123
124         // Récupère les dimensions originales de l'image
125         $originalDimension = getimagesize($media["tmp_name"][$key]);
126         $originalWidth = $originalDimension[0];
127         $originalHeight = $originalDimension[1];
128
129         if ($originalDimension !== false) {
130             $uploadOk = 1;
131         } else {
132             $uploadOk = 0;
133         }
134
135         // Vérifie si le fichier a été téléchargé avec succès et déplacé vers le dossier de destination
136         if (move_uploaded_file($media["tmp_name"][$key],
137 $destinationMediaTemp . $media["name"][$key])) {
138             // Redimensionne l'image pour créer une vignette et une version HD
139             $vignette = resize($destinationMediaTemp . $media["name"] [
140 $key], 100, 75);
141             $hd = resize($destinationMediaTemp . $media["name"][$key],
142 640, 480);
143
144             // Vérifie si l'image respecte le ratio 4:3 ou 3:4
145             if (($originalHeight / $originalWidth) == $ratio ||
146 ($originalWidth / $originalHeight) == $ratio) {
147                 // Enregistre l'image redimensionnée au format JPEG
148                 imagejpeg($vignette, $saveFileVignette, 80);
149                 imagejpeg($hd, $saveFileHd, 100);
150
151                 // Libère la mémoire des nouvelles images
152                 imagedestroy($vignette);
153                 imagedestroy($hd);
154             } else {
155                 // Si l'image ne respecte pas le ratio attendu, la découpe pour obtenir une vignette et une version HD
156                 pictureTrim($vignette, $saveFileVignette, 100, 75);
157                 pictureTrim($hd, $saveFileHd, 640, 480);
158             }
159
160             // Supprime le fichier temporaire téléchargé
161             unlink($destinationMediaTemp . $media["name"][$key]);
162         }
163     }
164 }

```

```

157                                     // Crée une nouvelle entrée pour le média dans la base de
158 données
159 $stageId)) {
160
161     if (!createMedia($todayDate, $nameVignette, $nameHd,
162                     $uploadOk = 0;
163                     })
164     } else {
165         $uploadOk = 1;
166     }
167
168     if ($uploadOk == 0) {
169         EDatabase::rollBack();
170         $error = "<div class='bg-danger text-white text-center'>Une
erreure est survenue.</div>";
171     } else {
172         EDatabase::commit();
173         $error = "<div class='bg-success text-white text-center'>L'
étape a été créée.</div>";
174     }
175 }
176 } catch (\Throwable $th) {
177     EDatabase::commit();
178     $error = "<div class='bg-success text-white text-center'>Les
modifications ont été enregistrées. (Aucun média n'a été ajouté).</div>";
179 }
180 } else {
181     EDatabase::rollBack();
182     $error = "<div class='bg-danger text-white text-center'>La date de fin
est antérieure à la date de début.</div>";
183 }
184 } else {
185     $error = "<div class='bg-danger text-white text-center'>Il n'y a pas d'
activités sélectionné.</div>";
186 }
187 } else {
188     EDatabase::rollBack();
189     $error = "<div class='bg-danger text-white text-center'>L'étape n'a pas de titre.<
/div>";
190 }
191 }
```

VoyageMemo/app/controller/stage/newStage.php

```

1 <?php
2
3 /**
4  * TITRE : VoyageMémo
5  * Auteur : Guilain Fernandez
6  * Description : Ce script permet de contrôler les informations entrées par l'utilisateur
et d'ajouter une nouvelle étape au voyage sélectionné.
7  * Date : 29/04/2024
8  */
9
10 // Déclaration des variables
11 $stageTitle = "";
12 $stageDescription = "";
13 $startDate = "00-00-0000";
```

```

14 $endDate = "00-00-0000";
15 $activity = [];
16 $uploadOk = 1;
17 $destinationMediaVignette = "././public/img/uploads/vignettes/"; //chemin du dossier où l'on va stocker les vignettes
18 $destinationMediaHd = "././public/img/uploads/hd/"; //chemin du dossier où l'on va stocker les images haute définition
19 $destinationMediaTemp = "././public/img/uploads/temp/"; //chemin du dossier où l'on va stocker les images temporaires
20 $todayDate = date("Y-m-d"); // la date du jour
21 $ratio = 4 / 3;
22 $numberDisplay = 0;
23
24 if (isset($_POST['newStage'])) {
25     EDatabase::beginTransaction(); //début de la transaction
26
27     $stageTitle = filter_input(INPUT_POST, 'stageTitle', FILTER_DEFAULT);
28     $stageDescription = filter_input(INPUT_POST, 'stageDescription', FILTER_DEFAULT);
29     $startDate = filter_input(INPUT_POST, 'startDate', FILTER_DEFAULT);
30     $endDate = filter_input(INPUT_POST, 'endDate', FILTER_DEFAULT);
31     $activity = $_POST['activity'];
32     $media = $_FILES['stageMedia'];
33     $travelId = filter_input(INPUT_POST, 'travelId', FILTER_DEFAULT);
34
35     // Vérification que les champs ne sont pas vides
36     if ($stageTitle != "" && $travelId != "") {
37         if ($startDate != null && $endDate != null) {
38             if ($activity != null) {
39                 $startDateTime = new DateTime($startDate);
40                 $endDateTime = new DateTime($endDate);
41                 if ($startDate <= $endDate) {
42                     $durationDays = $endDateTime->diff($startDateTime);
43                     createStage($stageTitle, $startDate, $durationDays->days,
44 $stageDescription, $travelId);
45                     $lastid = Edatabase::lastInsertId();
46
47                     $travel = getTravelById($travelId);
48                     foreach ($travel as $key => $travelValue) {
49                         $numberDisplay += $travelValue->numberDisplay;
50                     }
51                     $numberActivity = $numberDisplay + count($activity);
52                     travelNumberDisplayUpdate($travelId, $numberActivity);
53                     if ($numberActivity != 0) {
54                         foreach ($activity as $key => $value) {
55                             addActivity($lastid, $value);
56                         }
57                     }
58                 try {
59                     foreach ($media['error'] as $key => $v) {
60                         if ($media['error'][$key] != 0) {
61                             throw new Exception("file error", 1);
62                             die;
63                         }
64                         $uniqid = uniqid(); //nom unique pour l'image
65                         $imageFileType = strtolower(pathinfo($media["name"][$key], PATHINFO_EXTENSION)); //récupère l'extention du fichier
66                         $media["name"][$key] = $uniqid . "." . $imageFileType; //renomme l'image avec l'uniqid
67                         $nameVignette = basename($uniqid . "_vignette." .
68 $imageFileType);
69                         $nameHd = basename($uniqid . "_hd." . $imageFileType);
70                         $saveFileVignette = $destinationMediaVignette . $nameVignette;

```

```

69         $saveFileHd = $destinationMediaHd . $nameHd;
70
71         // Récupère les dimensions originales de l'image
72         $originalDimension = getimagesize($media["tmp_name"][$key]);
73         $originalWidth = $originalDimension[0];
74         $originalHeight = $originalDimension[1];
75
76         if ($originalDimension !== false) {
77             $uploadOk = 1;
78         } else {
79             $uploadOk = 0;
80         }
81
82         // Vérifie si le fichier a été téléchargé avec succès et
83         // déplacé vers le dossier de destination
84         if (move_uploaded_file($media["tmp_name"][$key],
85             $destinationMediaTemp . $media["name"][$key])) {
86             // Redimensionne l'image pour créer une vignette et une
87             // version HD
88             $vignette = resize($destinationMediaTemp . $media["name"]
89             [$key], 100, 75);
90             $hd = resize($destinationMediaTemp . $media["name"][$key],
91             640, 480);
92
93             // Vérifie si l'image respecte le ratio 4:3 ou 3:4
94             if (($originalHeight / $originalWidth) == $ratio ||
95                 ($originalWidth / $originalHeight) == $ratio) {
96                 // Enregistre l'image redimensionnée au format JPEG
97                 imagejpeg($vignette, $saveFileVignette, 80);
98                 imagejpeg($hd, $saveFileHd, 100);
99
100            // Libère la mémoire des nouvelles images
101            imagedestroy($vignette);
102            imagedestroy($hd);
103        } else {
104            // Si l'image ne respecte pas le ratio attendu, la
105            // découpe pour obtenir une vignette et une version HD
106            pictureTrim($vignette, $saveFileVignette, 100, 75);
107            pictureTrim($hd, $saveFileHd, 640, 480);
108        }
109
110        // Supprime le fichier temporaire téléchargé
111        unlink($destinationMediaTemp . $media["name"][$key]);
112
113        // Crée une nouvelle entrée pour le média dans la base de
114        données
115        if (!createMedia($todayDate, $nameVignette, $nameHd,
116            $lastid)) {
117            $uploadOk = 0;
118        }
119    } else {
120        $uploadOk = 0;
121    }
122
123    if ($uploadOk == 0) {
124        EDatabase::rollBack();
125        $error = "<div class='bg-danger text-white text-center'>Une
126        erreur est survenue.</div>";
127    } else {
128        EDatabase::commit();
129        $error = "<div class='bg-success text-white text-center'>L'
130        étape a été créée.</div>";
131    }

```

```

121             } catch (\Throwable $th) {
122                 $error = "<div class='bg-danger text-white text-center'>Une erreur
est survenue.</div>";
123                 }
124             } else {
125                 $error = "<div class='bg-danger text-white text-center'>La date de fin
est antérieure à la date de début.</div>";
126                 }
127             } else {
128                 $error = "<div class='bg-danger text-white text-center'>Aucune activité
sélectionnée.</div>";
129                 }
130             } else {
131                 $error = "<div class='bg-danger text-white text-center'>Les champs dates sont
vides.</div>";
132                 }
133             } else {
134                 $error = "<div class='bg-danger text-white text-center'>L'étape n'a pas de titre.<
/div>";
135             }
136         }
137     }

```

VoyageMemo/app/controller/travel/deleteTravel.php

```

1 <?php
2
3 /**
4  * TITRE : VoyageMémo
5  * Auteur : Guilain Fernandez
6  * Description : Permet de supprimer un voyage
7  * Date : 202/05/2024
8 */
9
10 // Déclaration des variables
11 $travelId = "";
12
13 if (isset($_POST['deleteTravel'])) {
14     $travelId = filter_input(INPUT_POST, 'travelId', FILTER_DEFAULT);
15
16     // Parcourt tous les stages associés au voyage et supprime leurs médias
17     foreach (getStageByTravelId($travelId) as $key => $stageValue) {
18         foreach (getMediaByStageId($stageValue->id) as $key => $mediaValue) {
19             unlink("./public/img/uploads/vignettes/" . $mediaValue->filenameVignette);
20             unlink("./public/img/uploads/hd/" . $mediaValue->filenameHd);
21         }
22     }
23     // Supprime le voyage de la base de données
24     deleteTravelById($travelId);
25     // Redirige vers la page de profil après la suppression
26     header('Location: index.php?page=profil');
27 }
28

```

VoyageMemo/app/controller/travel/modifyTravel.php

```

1 <?php
2 /*

```

```

3   TITRE : VoyageMémo
4   Auteur : Guilain Fernandez
5   Description : la page qui permet de modifier un voyage
6   Date : 24/04/2024
7 */
8
9 if (isset($_POST['modifyTravel'])) {
10    $travelTitle = filter_input(INPUT_POST, 'travelTitle', FILTER_DEFAULT);
11    $travelId = filter_input(INPUT_POST, 'travelId', FILTER_DEFAULT);
12
13   // Vérification que les champs ne sont pas vides
14   if ($travelTitle != "") {
15      // Met à jour le titre du voyage dans la base de données
16      if (travelUpdate($travelId, $travelTitle)) {
17         $error = "<div class='bg-success text-white text-center'>Le voyage a bien été
modifié.</div>";
18      } else {
19         $error = "<div class='bg-danger text-white text-center'>Une erreur est survenue
et le voyage n'a pas pu être créé.</div>";
20      }
21   } else {
22      $error = "<div class='bg-danger text-white text-center'>Les champs n'ont pas été
remplis.</div>";
23   }
24 }
```

VoyageMemo/app/controller/travel/newTravel.php

```

1 <?php
2
3 /**
4  * TITRE : VoyageMémo
5  * Auteur : Guilain Fernandez
6  * Description : on va contrôler si toutes les informations entré par l'utilisateur son
bonne et après on créer un voyage
7  * Date : 29/04/2024
8 */
9
10 // Déclaration des variables
11 $travelTitle = "";
12
13 if (isset($_POST['newTravel'])) {
14    $travelTitle = filter_input(INPUT_POST, 'travelTitle', FILTER_DEFAULT);
15
16   // Vérification que les champs ne sont pas vides
17   if ($travelTitle != "") {
18      $userCreator = getUser($_SESSION["userLogin"]);
19      // Création du voyage dans la base de données
20      if (createTravel($travelTitle, 0, $userCreator[0]->id)) {
21         $error = "<div class='bg-success text-white text-center'>Le voyage a été créé<
/div>";
22      } else {
23         $error = "<div class='bg-danger'>Une erreur est survenue et le voyage n'a pas
pu être créé</div>";
24      }
25   } else {
26      $error = "<div class='bg-danger'>Les champs n'ont pas été remplis.</div>";
27   }
28 }
```

VoyageMemo/app/controller/user/newAccount.php

```
1 <?php
2 /**
3  * TITRE : VoyageMémo
4  * Auteur : Guilain Fernandez
5  * Description : on va controller si toutes les informations entré par l'utilisateur son
6  * bonne et après on créer son compte
7  * Date : 25/04/2024
8 */
9
10 // Déclaration des variables
11 $login = "";
12 $password = "";
13 $passwordValidate = "";
14
15 if (isset($_POST['newAccount'])) {
16     // Récupération des données du formulaire
17     $login = filter_input(INPUT_POST, 'login', FILTER_DEFAULT);
18     $password = filter_input(INPUT_POST, 'password', FILTER_DEFAULT);
19     $passwordValidate = filter_input(INPUT_POST, 'validatePsw', FILTER_DEFAULT);
20
21     // Vérification que les champs ne sont pas vides
22     if ($login != "" && $password != "" && $passwordValidate != "") {
23         // Vérification de la concordance des mots de passe
24         if (passwordCompare($password, $passwordValidate)) {
25             // Vérification de la longueur du mot de passe
26             if (passwordLength($password)) {
27                 // Vérification si l'utilisateur existe déjà
28                 if (!verifyUserExists($login)) {
29                     // Création de l'utilisateur
30                     if (createUser($login, $password)) {
31                         $error = "<div class='bg-success text-white text-center'>Le compte
32 a été créé.</div>";
33                     } else {
34                         $error = "<div class='bg-danger text-white text-center'>Une erreur
35 est survenue et le compte n'a pas pu être créé.</div>";
36                     }
37                 } else {
38                     $error = "<div class='bg-danger text-white text-center'>Le login existe
39 déjà.</div>";
40                 }
41             } else {
42                 $error = "<div class='bg-danger text-white text-center'>Les mots de passe ne
43 répondent pas.</div>";
44             }
45         } else {
46             $error = "<div class='bg-danger text-white text-center'>Les champs n'ont pas été
47 remplis.</div>";
48         }
50     }
51 }
```

VoyageMemo/app/controller/user/userFunction.php

```

1 <?php
2
3 /**
4 * TITRE : VoyageMémo
5 * Auteur : Guilain Fernandez
6 * Description : la page où il y a toutes les fonction concernant l'utilisateur
7 * Date : 25/04/2024
8 */
9
10 /**
11 * Compare deux mots de passe pour vérifier s'ils sont identiques.
12 *
13 * @param string $password Le premier mot de passe à comparer.
14 * @param string $passwordVerify Le deuxième mot de passe à comparer.
15 * @return bool Retourne vrai si les deux mots de passe sont identiques, sinon faux.
16 */
17 function passwordCompare($password, $passwordVerify)
18 {
19     if ($password == $passwordVerify) {
20         return true;
21     } else {
22         return false;
23     }
24 }
25
26 /**
27 * Vérifie si la longueur du mot de passe est d'au moins 8 caractères.
28 *
29 * @param string $password Le mot de passe à vérifier.
30 * @return bool Retourne vrai si le mot de passe a une longueur d'au moins 8 caractères,
31 * sinon faux.
32 */
33 function passwordLength($password)
34 {
35     $passwordSize = strlen($password);
36     if ($passwordSize >= 8) {
37         return true;
38     } else {
39         return false;
40     }
41 }
```

VoyageMemo/app/controller/user/userLogin.php

```

1 <?php
2
3 /**
4 * TITRE : VoyageMémo
5 * Auteur : Guilain Fernandez
6 * Description : on vérifie si l'utilisateur entre des données existante
7 * Date : 25/04/2024
8 */
9
10 // Déclaration des variables
11 $login = "";
12 $password = "";
13
```

```

14 if (isset($_POST['login'])) {
15     // Récupération des données du formulaire
16     $login = filter_input(INPUT_POST, 'userLogin', FILTER_DEFAULT);
17     $password = filter_input(INPUT_POST, 'password', FILTER_DEFAULT);
18
19     // Vérification que les champs ne sont pas vides
20     if ($login != "" && $password != "") {
21         // Vérifie si l'utilisateur existe dans la base de données
22         if (getUser($login)) {
23             $user = getUser($login);
24             // Vérifie si le mot de passe est correct
25             if (password_verify($password, $user[0]->password)) {
26                 $_SESSION["userLogin"] = $user[0]->login;
27                 $_SESSION["userId"] = $user[0]->id;
28                 // Redirige l'utilisateur vers la page de profil
29                 header('Location: index.php?page=profil');
30             }
31         } else {
32             $error = "<div class='bg-danger text-white text-center'>L'utilisateur n'existe pas.</div>";
33         }
34     } else {
35         $error = "<div class='bg-danger text-white text-center'>Les champs n'ont pas été remplis.</div>";
36     }
37 }
38

```

VoyageMemo/app/controller/controller.php

```

1 <?php
2
3 /**
4 * TITRE : VoyageMémo
5 * Auteur : Guilain Fernandez
6 * Description : le controller du projet
7 * Date : 24/04/2024
8 */
9
10 // Définit le fuseau horaire par défaut à utiliser.
11 date_default_timezone_set('UTC');
12
13 //démarrer une nouvelle session
14 session_start();
15
16 /* REQUIRES :*/
17 require_once './model/config/database.php';
18 require_once './model/usersClass.php';
19 require_once './model/activityClass.php';
20 require_once './model/travelClass.php';
21 require_once './model/stageClass.php';
22 require_once './model/mediaClass.php';
23 require_once './model/stagesHasActivityClass.php';
24
25 /* Gestion des pages */
26 if (!isset($_GET['page']) || $_GET['page'] == "home") {
27     require_once 'user/userDisconnection.php';
28     require_once './view/home.php';
29 }

```

```

30 | if (isset($_GET['page'])) {
31 |     $name = filter_input(INPUT_GET, 'page');
32 |     switch ($name) {
33 |         case 'modifyStage':
34 |             require_once 'media/mediaFunction.php';
35 |             require_once 'stage/modifyStage.php';
36 |             require_once './view/modifyStage.php';
37 |             break;
38 |         case 'newStage':
39 |             require_once 'media/mediaFunction.php';
40 |             require_once 'stage/newStage.php';
41 |             require_once './view/newStage.php';
42 |             break;
43 |         case 'modifyTravel':
44 |             require_once 'travel/modifyTravel.php';
45 |             require_once './view/modifyTravel.php';
46 |             break;
47 |         case 'newTravel':
48 |             require_once 'travel/newTravel.php';
49 |             require_once './view/newTravel.php';
50 |             break;
51 |         case 'detailTravel':
52 |             require_once 'user/userDisconnection.php';
53 |             require_once 'travel/deleteTravel.php';
54 |             require_once 'stage/deleteStage.php';
55 |             require_once './view/detailTravel.php';
56 |             break;
57 |         case 'profil':
58 |             require_once 'user/userDisconnection.php';
59 |             require_once './view/profil.php';
60 |             break;
61 |         case 'newUserAccount':
62 |             require_once 'user/userFunction.php';
63 |             require_once 'user/newAccount.php';
64 |             require_once './view(newUserAccount.php';
65 |             break;
66 |         case 'loginUser':
67 |             require_once 'user/userLogin.php';
68 |             require_once './view/loginUser.php';
69 |             break;
70 |         default:
71 |             require_once 'user/userDisconnection.php';
72 |             require_once './view/home.php';
73 |             break;
74 |     }
75 | }
76 |

```

VoyageMemo/app/controller/user/userDisconnection.php

```

1 <?php
2
3 /**
4  * TITRE : VoyageMémo
5  * Auteur : Guilain Fernandez
6  * Description : déconnecte l'utilisateur
7  * Date : 25/04/2024
8  */
9

```

```

10 | if (isset($_POST["disconnect"])) {
11 |     session_destroy();
12 |     // header('index.php?page=home');
13 |     // exit;
14 |
15 |

```

VoyageMemo/app/model/config/conparam.php

```

1 | <?php
2 | // Fichier de paramètre
3 |
4 | //db Config
5 | define( 'EDB_DBTYPE' , "mysql");
6 | define( 'EDB_DBNAME' , "VoyageMemo");
7 | define( 'EDB_HOST' , "localhost");
8 | define( 'EDB_PORT' , "3306");
9 | define( 'EDB_USER' , "guilain");
10| define( 'EDB_PASS' , "Super");

```

VoyageMemo/app/model/config/database.php

```

1 | <?php
2 | /**
3 | * TITRE : VoyageMémo
4 | * Page : database.php
5 | * Auteur :
6 | * Description : Page qui permet la connexion avec la base de donnée
7 | * Date : du 26.9.2023
8 | * Dernière modification : 19.10.2023
9 | *
10| */
11|
12| /**
13| * @remark Mettre le bon chemin d'accès à votre fichier contenant les constantes
14| */
15| require_once 'conparam.php';
16|
17| class EDatabase {
18| /**
19| * @var PDO The static PDO object instance created within getInstance()
20| */
21| private static $objInstance;
22| private function __construct()
23| {
24|     echo ".";
25| }
26| private static function getInstance() {
27|     if(!self::$objInstance){
28|         try{
29|
30|             $dsn = EDB_DBTYPE.':host='.$EDB_HOST.';port='.$EDB_PORT.';dbname='
31| .EDB_DBNAME;
32|             self::$objInstance = new PDO($dsn, EDB_USER, EDB_PASS, array('charset'=>'utf8'));
33|             self::$objInstance->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION)
34| ;
35|         }catch(PDOException $e ){

```

```

34         echo "EDatabase Error: ".$e;
35     }
36 }
37 return self::$objInstance;
38 }
39 final public static function __callStatic( $chrMethod, $arrArguments ) {
40     $objInstance = self::getInstance();
41     return call_user_func_array($arrArguments, $chrMethod, $arrArguments);
42 } # end method
43 }

```

VoyageMemo/app/model/db/dump-VoyageMemo-202405151452.sql

```

1 -- MySQL dump 10.13 Distrib 8.0.19, for Win64 (x86_64)
2 --
3 -- Host: localhost      Database: VoyageMemo
4 --
5 -- Server version      5.5.5-10.6.12-MariaDB-0ubuntu0.22.04.1
6
7 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
8 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
9 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
10 /*!50503 SET NAMES utf8mb4 */;
11 /*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
12 /*!40103 SET TIME_ZONE='+00:00' */;
13 /*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
14 /*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
15 /*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
16 /*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
17
18 --
19 -- Table structure for table `activity`
20 --
21
22 DROP TABLE IF EXISTS `activity`;
23 /*!40101 SET @saved_cs_client      = @@character_set_client */;
24 /*!50503 SET character_set_client = utf8mb4 */;
25 CREATE TABLE `activity` (
26     `id` int(11) NOT NULL AUTO_INCREMENT,
27     `name` varchar(45) NOT NULL,
28     PRIMARY KEY (`id`)
29 ) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
30 /*!40101 SET character_set_client = @saved_cs_client */;
31
32 --
33 -- Dumping data for table `activity`
34 --
35
36 LOCK TABLES `activity` WRITE;
37 /*!40000 ALTER TABLE `activity` DISABLE KEYS */;
38 INSERT INTO `activity` VALUES (1,'visites'),(2,'sportives'),(3,'découvertes'),(4,'relaxation / bien-être'),(5,'aventure'),(6,'shopping');
39 /*!40000 ALTER TABLE `activity` ENABLE KEYS */;
40 UNLOCK TABLES;
41
42 --
43 -- Table structure for table `media`
44 --
45

```

```

46 | DROP TABLE IF EXISTS `media`;
47 | /*!40101 SET @saved_cs_client      = @@character_set_client */;
48 | /*!50503 SET character_set_client = utf8mb4 */;
49 | CREATE TABLE `media` (
50 |   `id` int(11) NOT NULL AUTO_INCREMENT,
51 |   `stages_id` int(11) NOT NULL,
52 |   `add_date` datetime NOT NULL,
53 |   `filename_vignette` varchar(150) DEFAULT NULL,
54 |   `filename_hd` varchar(150) DEFAULT NULL,
55 |   PRIMARY KEY (`id`),
56 |   KEY `stages_id` (`stages_id`),
57 |   CONSTRAINT `media_ibfk_1` FOREIGN KEY (`stages_id`) REFERENCES `stages` (`id`) ON DELETE CASCADE
58 | ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
59 | /*!40101 SET character_set_client = @saved_cs_client */;
60 |
61 | --
62 | -- Dumping data for table `media`
63 | --
64 |
65 | LOCK TABLES `media` WRITE;
66 | /*!40000 ALTER TABLE `media` DISABLE KEYS */;
67 | /*!40000 ALTER TABLE `media` ENABLE KEYS */;
68 | UNLOCK TABLES;
69 |
70 | --
71 | -- Table structure for table `stages`
72 | --
73 |
74 | DROP TABLE IF EXISTS `stages`;
75 | /*!40101 SET @saved_cs_client      = @@character_set_client */;
76 | /*!50503 SET character_set_client = utf8mb4 */;
77 | CREATE TABLE `stages` (
78 |   `id` int(11) NOT NULL AUTO_INCREMENT,
79 |   `travels_id` int(11) NOT NULL,
80 |   `title` varchar(255) NOT NULL,
81 |   `date` date DEFAULT NULL,
82 |   `duration_days` int(2) DEFAULT NULL,
83 |   `description` tinytext DEFAULT NULL,
84 |   PRIMARY KEY (`id`),
85 |   KEY `travels_id` (`travels_id`),
86 |   CONSTRAINT `stages_ibfk_1` FOREIGN KEY (`travels_id`) REFERENCES `travels` (`id`) ON DELETE CASCADE
87 | ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
88 | /*!40101 SET character_set_client = @saved_cs_client */;
89 |
90 | --
91 | -- Dumping data for table `stages`
92 | --
93 |
94 | LOCK TABLES `stages` WRITE;
95 | /*!40000 ALTER TABLE `stages` DISABLE KEYS */;
96 | /*!40000 ALTER TABLE `stages` ENABLE KEYS */;
97 | UNLOCK TABLES;
98 |
99 | --
100 | -- Table structure for table `stages_has_activity`
101 | --
102 |
103 | DROP TABLE IF EXISTS `stages_has_activity`;

```

```

104 /*!40101 SET @saved_cs_client      = @@character_set_client */;
105 /*!50503 SET character_set_client = utf8mb4 */;
106 CREATE TABLE `stages_has_activity` (
107   `stages_id` int(11) NOT NULL,
108   `activity_id` int(11) NOT NULL,
109   KEY `stages_id` (`stages_id`),
110   KEY `activity_id` (`activity_id`),
111   CONSTRAINT `stages_has_activity_ibfk_1` FOREIGN KEY (`stages_id`) REFERENCES `stages`(`id`) ON DELETE CASCADE,
112   CONSTRAINT `stages_has_activity_ibfk_2` FOREIGN KEY (`activity_id`) REFERENCES `activity`(`id`) ON DELETE CASCADE
113 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
114 /*!40101 SET character_set_client = @saved_cs_client */;
115
116 --
117 -- Dumping data for table `stages_has_activity`
118 --
119
120 LOCK TABLES `stages_has_activity` WRITE;
121 /*!40000 ALTER TABLE `stages_has_activity` DISABLE KEYS */;
122 /*!40000 ALTER TABLE `stages_has_activity` ENABLE KEYS */;
123 UNLOCK TABLES;
124
125 --
126 -- Table structure for table `travels`
127 --
128
129 DROP TABLE IF EXISTS `travels`;
130 /*!40101 SET @saved_cs_client      = @@character_set_client */;
131 /*!50503 SET character_set_client = utf8mb4 */;
132 CREATE TABLE `travels` (
133   `id` int(11) NOT NULL AUTO_INCREMENT,
134   `users_id` int(11) NOT NULL,
135   `title` varchar(255) NOT NULL,
136   `number_display` int(11) DEFAULT NULL,
137   PRIMARY KEY (`id`),
138   KEY `users_id` (`users_id`),
139   CONSTRAINT `travels_ibfk_1` FOREIGN KEY (`users_id`) REFERENCES `users`(`id`) ON DELETE CASCADE
140 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
141 /*!40101 SET character_set_client = @saved_cs_client */;
142
143 --
144 -- Dumping data for table `travels`
145 --
146
147 LOCK TABLES `travels` WRITE;
148 /*!40000 ALTER TABLE `travels` DISABLE KEYS */;
149 /*!40000 ALTER TABLE `travels` ENABLE KEYS */;
150 UNLOCK TABLES;
151
152 --
153 -- Table structure for table `users`
154 --
155
156 DROP TABLE IF EXISTS `users`;
157 /*!40101 SET @saved_cs_client      = @@character_set_client */;
158 /*!50503 SET character_set_client = utf8mb4 */;
159 CREATE TABLE `users` (
160   `id` int(11) NOT NULL AUTO_INCREMENT,
161   `login` varchar(100) NOT NULL,

```

```

162 `password` varchar(90) NOT NULL,
163 PRIMARY KEY (`id`)
164 ) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
165 /*!40101 SET character_set_client = @saved_cs_client */;
166 --
167 --
168 -- Dumping data for table `users`
169 --
170
171 LOCK TABLES `users` WRITE;
172 /*!40000 ALTER TABLE `users` DISABLE KEYS */;
173 INSERT INTO `users` VALUES (1,'Xayah','
$2y$10$zjAj3ZrV3jQbpRe9GGnhS.nG6WCcQT2vIajVl8S0wJgx0J2vmG2MK');
174 /*!40000 ALTER TABLE `users` ENABLE KEYS */;
175 UNLOCK TABLES;
176 --
177 --
178 -- Dumping routines for database 'VoyageMemo'
179 --
180 /*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;
181
182 /*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
183 /*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
184 /*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
185 /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
186 /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
187 /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
188 /*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;
189
190 -- Dump completed on 2024-05-15 14:52:42
191

```

VoyageMemo/app/model/usersClass.php

```

1 <?php
2
3 /**
4  * TITRE : VoyageMémo
5  * Auteur : Guilain Fernandez
6  * Description : la class user
7  * Date : 25/04/2024
8  */
9 class User
10 {
11     public function __construct($InId = -1, $InLogin = "", $InPassword = "")
12     {
13         $this->id = $InId;
14         $this->login = $InLogin;
15         $this->password = $InPassword;
16     }
17
18     public $id;
19
20     public $login;
21
22     public $password;
23 }
24
25 /**

```

```

26 * Crée un nouvel utilisateur dans la base de données avec le login et le mot de passe
27 * spécifiés.
28 *
29 * @param string $login Le nom d'utilisateur du nouvel utilisateur
30 * @param string $password Le mot de passe du nouvel utilisateur
31 * @return bool Retourne true si l'utilisateur est créé avec succès, sinon false.
32 */
33 function createUser($login, $password)
34 {
35     // Requête SQL pour insérer un nouvel utilisateur avec le login et le mot de passe
36     // spécifiés
37     $sql = "INSERT INTO users (`login`, `password`) VALUES(:l, :p);";
38     $statement = EDatabase:::prepare($sql, array(PDO::ATTR_CURSOR => PDO::CURSOR_FWDONLY));
39     try {
40         $tab = array(':l' => $login, ':p' => password_hash($password, PASSWORD_DEFAULT));
41         foreach ($tab as $key => $value) {
42             $statement->bindValue($key, $value);
43         }
44         $statement->execute();
45         // Retourne true si l'insertion est réussie
46         return true;
47     } catch (PDOException $error) {
48         // En cas d'erreur, retourne false
49         return false;
50     }
51 /**
52 * Vérifie si un utilisateur existe dans la table 'users' en fonction de son nom d'
53 * utilisateur.
54 *
55 * @param string $login Le nom d'utilisateur à vérifier.
56 * @return int|bool Retourne le nombre d'utilisateurs trouvés avec le nom d'utilisateur
57 * donné, ou faux en cas d'erreur.
58 */
59 function verifyUserExiste($login)
60 {
61     // Requête SQL pour compter le nombre d'utilisateurs avec le nom d'utilisateur donné
62     $sql = "SELECT COUNT(*)
63             FROM users
64             WHERE `login` = :l";
65     $statement = EDatabase:::prepare($sql, array(PDO::ATTR_CURSOR => PDO::CURSOR_FWDONLY));
66     try {
67         $tab = array(":l" => $login);
68         foreach ($tab as $key => $value) {
69             $statement->bindValue($key, $value);
70         }
71         $statement->execute();
72         $data = $statement->fetch(PDO::FETCH_ASSOC, PDO::FETCH_ORI_NEXT);
73     } catch (PDOException $e) {
74         // En cas d'erreur, retourne faux
75         return false;
76     }
77     // Retourne le nombre d'utilisateurs trouvés avec le nom d'utilisateur donné
78     return $data['COUNT(*)'];
79 /**
80 * Récupère un utilisateur à partir de son nom d'utilisateur (login) dans la table 'users'
81 *

```

```

82 * @param string $login Le nom d'utilisateur à rechercher.
83 * @return array|bool Retourne un tableau contenant un objet User représentant l'
utilisateur trouvé, ou faux en cas d'erreur ou si aucun utilisateur n'est trouvé.
84 */
85 function getUser($login)
86 {
87     $userVerify = [];
88     // Requête SQL pour récupérer l'utilisateur avec le nom d'utilisateur donné
89     $sql = "SELECT `id`, `login`, `password`
90     FROM `users`
91     WHERE `login` = :1";
92     $statement = EDatabase::prepare($sql, array(PDO::ATTR_CURSOR => PDO::CURSOR_FWDONLY));
93     try {
94         $tab = array(":1" => $login);
95         foreach ($tab as $key Rightarrow $value) {
96             $statement->bindValue($key, $value);
97         }
98         $statement->execute();
99     } catch (PDOException $e) {
100         // En cas d'erreur, retourne faux
101         return false;
102     }
103     while ($row = $statement->fetch(PDO::FETCH_ASSOC, PDO::FETCH_ORI_NEXT)) {
104         // Ajoute un nouvel objet User au tableau $userVerify en utilisant les données de
la base de données
105         $userVerify[] = new User($row['id'], $row['login'], $row['password']);
106     }
107     // Retourne le tableau contenant un objet User représentant l'utilisateur trouvé, ou
faux en cas d'erreur ou si aucun utilisateur n'est trouvé
108     return $userVerify;
109 }
110

```

VoyageMemo/app/model/activityClass.php

```

1 <?php
2 /**
3 * TITRE : VoyageMémo
4 * Auteur : Guilain Fernandez
5 * Description : la class activités
6 * Date : 29/04/2024
7 */
8 class Activity
9 {
10     public function __construct($InId = -1, $InName = "")
11     {
12         $this->id = $InId;
13         $this->name = $InName;
14     }
15
16     public $id;
17
18     public $name;
19 }
20
21 /**
22 * Récupère toutes les activités depuis la base de données.
23 *
24 * @return array|bool Retourne un tableau d'objets Activity contenant toutes les activités
si trouvées,

```

```

25             * sinon retourne false en cas d'erreur ou si aucune activité n'est
26             trouvée.
27         */
28     function getAllActivity()
29     {
30         $allActivity = [];
31         // Requête SQL pour récupérer toutes les activités de la table 'activity'
32         $sql = "SELECT `id`, `name`
33             FROM `activity`";
34         $statement = EDatabase::prepare($sql, array(PDO::ATTR_CURSOR => PDO::CURSOR_FWDONLY));
35         try {
36             $tab = array();
37             foreach ($tab as $key => $value) {
38                 $statement->bindValue($key, $value);
39             }
40             $statement->execute();
41         } catch (PDOException $e) {
42             return false;
43         }
44         while ($row = $statement->fetch(PDO::FETCH_ASSOC, PDO::FETCH_ORI_NEXT)) {
45             // On crée l'objet Activity en l'initialisant avec les données provenant de la base
46             // de données
47             $allActivity[] = new Activity($row['id'], $row['name']);
48         }
49     }
50
51 /**
52 * Récupère les activités associées à un stage spécifique.
53 *
54 * @param int $stage L'identifiant du stage.
55 * @return array|bool Retourne un tableau d'objets Activity représentant les activités
56 * associées au stage, ou faux en cas d'erreur.
57 */
58 function getActivityByStageId($stage)
59 {
60     $stageActivity = [];
61     // Requête SQL pour sélectionner les activités associées au stage spécifié
62     $sql = "SELECT `activity`.`id`, `activity`.`name`
63         FROM `stages_has_activity`
64         INNER JOIN `activity` ON `stages_has_activity`.`activity_id` = `activity`.`id`
65         WHERE `stages_has_activity`.`stages_id` = :s";
66     $statement = EDatabase::prepare($sql, array(PDO::ATTR_CURSOR => PDO::CURSOR_FWDONLY));
67     try {
68         $tab = array(":s" => $stage);
69         foreach ($tab as $key => $value) {
70             $statement->bindValue($key, $value);
71         }
72         $statement->execute();
73     } catch (PDOException $e) {
74         return false;
75     }
76     while ($row = $statement->fetch(PDO::FETCH_ASSOC, PDO::FETCH_ORI_NEXT)) {
77         // Ajoute un nouvel objet Activity au tableau $stageActivity en utilisant les
78         // données de la base de données
79         $stageActivity[] = new Activity($row['id'], $row['name']);
80     }
81     // Retourne le tableau d'objets Activity représentant les activités associées au stage
82     return $stageActivity;

```

81 | }

VoyageMemo/app/model/mediaClass.php

```
1 <?php
2
3 /**
4  * TITRE : VoyageMémo
5  * Auteur : Guilain Fernandez
6  * Description : la class media
7  * Date : 25/04/2024
8  */
9 class Media
10 {
11     public function __construct($InId = -1, $InAddDate = "", $InFilenameVignette = "",
12     $InFilenameHd = "") {
13         $this->id = $InId;
14         $this->addDate = $InAddDate;
15         $this->filenameVignette = $InFilenameVignette;
16         $this->filenameHd = $InFilenameHd;
17     }
18
19     public $id;
20
21     public $addDate;
22
23     public $filenameVignette;
24
25     public $filenameHd;
26 }
27
28 /**
29  * Crée un nouveau média dans la table 'media'.
30  *
31  * @param string $date La date d'ajout du média (au format YYYY-MM-DD).
32  * @param string $fileNameVignette Le nom du fichier vignette.
33  * @param string $fileNameHd Le nom du fichier HD.
34  * @param int $stageId L'identifiant du stage auquel le média est associé.
35  * @return bool Retourne vrai si la création du média est réussie, sinon retourne faux.
36  */
37 function createMedia($date, $fileNameVignette, $fileNameHd, $stageId)
38 {
39     // Requête SQL pour insérer un nouveau média dans la table 'media'
40     $sql = "INSERT INTO media (`add_date`, `filename_vignette`, `filename_hd`,
41     `stages_id`) VALUES(:da, :fv, :fh, :si);";
42     $statement = EDatabase::prepare($sql, array(PDO::ATTR_CURSOR => PDO::CURSOR_FWDONLY));
43     try {
44         $tab = array(':da' => $date, ':fv' => $fileNameVignette, ':fh' => $fileNameHd,
45         ':si' => $stageId);
46         foreach ($tab as $key => $value) {
47             $statement->bindValue($key, $value);
48         }
49         $statement->execute();
50         // Retourne vrai si la création du média est réussie
51         return true;
52     } catch (PDOException $error) {
53         // En cas d'erreur, retourne faux
54         return false;
55     }
56 }
```

```

53     }
54 }
55
56 /**
57 * Récupère les médias associés à un stage spécifique.
58 *
59 * @param int $stageId L'identifiant du stage.
60 * @return array|bool Retourne un tableau contenant des objets Media représentant les
61 * médias associés au stage, ou faux en cas d'erreur.
62 */
63 function getMediaByStageId($stageId)
64 {
65     $stageMedia = [];
66     // Requête SQL pour sélectionner les médias associés au stage spécifié
67     $sql = "SELECT `id`, `add_date`, `filename_vignette`, `filename_hd`"
68     FROM `media`
69     WHERE `stages_id` = :si";
70     $statement = EDatabase:::prepare($sql, array(PDO::ATTR_CURSOR => PDO::CURSOR_FWDONLY));
71     try {
72         $tab = array(":si" => $stageId);
73         foreach ($tab as $key => $value) {
74             $statement->bindValue($key, $value);
75         }
76         $statement->execute();
77     } catch (PDOException $e) {
78         // En cas d'erreur, retourne faux
79         return false;
80     }
81     while ($row = $statement->fetch(PDO::FETCH_ASSOC, PDO::FETCH_ORI_NEXT)) {
82         // Ajoute un nouvel objet Media au tableau $stageMedia en utilisant les données de
83         // la base de données
84         $stageMedia[] = new Media($row['id'], $row['add_date'], $row['filename_vignette'],
85         $row['filename_hd']);
86     }
87     // Retourne le tableau contenant les objets Media représentant les médias associés au
88     // stage
89     return $stageMedia;
90 }
91
92 /**
93 * Récupère les informations d'un média à partir de son identifiant.
94 *
95 * @param int $idMedia L'identifiant du média à récupérer.
96 * @return array|bool Retourne un tableau contenant un objet Media représentant le média
97 * correspondant à l'identifiant donné, ou faux en cas d'erreur.
98 */
99 function getMediaById($idMedia)
100 {
101     $stageMedia = [];
102     // Requête SQL pour sélectionner les informations du média avec l'identifiant donné
103     $sql = "SELECT `id`, `add_date`, `filename_vignette`, `filename_hd`"
104     FROM `media`
105     WHERE `id` = :i";
106     $statement = EDatabase:::prepare($sql, array(PDO::ATTR_CURSOR => PDO::CURSOR_FWDONLY));
107     try {
108         $tab = array(":i" => $idMedia);
109         foreach ($tab as $key => $value) {
110             $statement->bindValue($key, $value);
111         }
112         $statement->execute();
113     } catch (PDOException $e) {

```

```

109         // En cas d'erreur, retourne faux
110     return false;
111 }
112 while ($row = $statement->fetch(PDO::FETCH_ASSOC, PDO::FETCH_ORI_NEXT)) {
113     // Ajoute un nouvel objet Media au tableau $stageMedia en utilisant les données de
114     // la base de données
115     $stageMedia[] = new Media($row['id'], $row['add_date'], $row['filename_vignette'],
116     $row['filename_hd']);
117 }
118 // Retourne le tableau contenant l'objet Media représentant le média correspondant à
119 // l'identifiant donné
120 return $stageMedia;
121 }
122 /**
123 * Supprime un média de la table 'media' en fonction de son identifiant.
124 *
125 * @param int $id L'identifiant du média à supprimer.
126 * @return bool Retourne vrai si la suppression du média est réussie, sinon retourne faux.
127 */
128 function deleteMediaById($id)
129 {
130     // Requête SQL pour supprimer le média avec l'identifiant donné
131     $sql = "DELETE
132         FROM `media`
133         WHERE `id` = :i";
134     $statement = EDatabase::prepare($sql, array(PDO::ATTR_CURSOR => PDO::CURSOR_FWDONLY));
135     try {
136         $tab = array(':i' => $id);
137         foreach ($tab as $key => $value) {
138             $statement->bindValue($key, $value);
139         }
140         $statement->execute();
141         // Retourne vrai si la suppression du média est réussie
142         return true;
143     } catch (PDOException $e) {
144         // En cas d'erreur, retourne faux
145         return false;
146     }
}

```

VoyageMemo/app/model/stageClass.php

```

1 <?php
2 /**
3 * TITRE : VoyageMémo
4 * Auteur : Guilain Fernandez
5 * Description : la class Stage
6 * Date : 25/04/2024
7 */
8 class Stage
9 {
10     public function __construct($InId = -1, $InTitle = "", $InDate = "", $InDurationDays =
11     "", $InDescription = "", $InTravelId = "")
12     {
13         $this->id = $InId;
14         $this->title = $InTitle;
15         $this->date = $InDate;
16         $this->durationDays = $InDurationDays;

```

```

16     $this->description = $InDescription;
17     $this->travelId = $InTravelId;
18 }
19
20 public $id;
21
22 public $title;
23
24 public $date;
25
26 public $durationDays;
27
28 public $description;
29
30 public $travelId;
31 }
32
33 /**
34 * Crée un nouveau stage dans la table 'stages'.
35 *
36 * @param string $title Le titre du stage.
37 * @param string $date La date du stage (au format YYYY-MM-DD).
38 * @param int $durationDays La durée du stage en jours.
39 * @param string $description La description du stage.
40 * @param int $travelsId L'identifiant du voyage auquel le stage est associé.
41 * @return bool Retourne vrai si la création du stage est réussie, sinon retourne faux.
42 */
43 function createStage($title, $date, $durationDays, $description, $travelsId)
44 {
45     // Requête SQL pour insérer un nouveau stage dans la table 'stages'
46     $sql = "INSERT INTO stages (`title`, `date`, `duration_days`, `description`,
`travels_id`) VALUES(:t, :da, :dd, :de, :ti);";
47     $statement = EDatabase:::prepare($sql, array(PDO:::ATTR_CURSOR => PDO:::CURSOR_FWDONLY));
48     try {
49         $tab = array(':t' => $title, ':da' => $date, ':dd' => $durationDays, ':de' =>
$description, ':ti' => $travelsId);
50         foreach ($tab as $key => $value) {
51             $statement->bindValue($key, $value);
52         }
53         $statement->execute();
54         // Retourne vrai si la création du stage est réussie
55         return true;
56     } catch (PDOException $error) {
57         // En cas d'erreur, retourne faux
58         return false;
59     }
60 }
61
62 /**
63 * Récupère les stages associés à un voyage spécifique.
64 *
65 * @param int $travel L'identifiant du voyage.
66 * @return array|bool Retourne un tableau contenant des objets Stage représentant les
stages associés au voyage, ou faux en cas d'erreur.
67 */
68 function getStageByTravelId($travel)
69 {
70     $travelStage = [];
71     // Requête SQL pour sélectionner les stages associés au voyage spécifié
72     $sql = "SELECT `id`, `title`, `date`, `duration_days`, `description`, `travels_id`
FROM `stages`"

```

```

74     WHERE `travels_id` = :1";
75     $statement = EDatabase::prepare($sql, array(PDO::ATTR_CURSOR => PDO::CURSOR_FWDONLY));
76     try {
77         $tab = array(":1" => $travel);
78         foreach ($tab as $key => $value) {
79             $statement->bindValue($key, $value);
80         }
81         $statement->execute();
82     } catch (PDOException $e) {
83         // En cas d'erreur, retourne faux
84         return false;
85     }
86     while ($row = $statement->fetch(PDO::FETCH_ASSOC, PDO::FETCH_ORI_NEXT)) {
87         // Ajoute un nouvel objet Stage au tableau $travelStage en utilisant les données
88         // de la base de données
89         $travelStage[] = new Stage($row['id'], $row['title'], $row['date'], $row['
duration_days'], $row['description'], $row['travels_id']);
90     }
91     // Retourne le tableau contenant les objets Stage représentant les stages associés au
voyage
92     return $travelStage;
93 }
94 /**
95 * Ajoute une activité à une étape dans la table 'stages_has_activity'.
96 *
97 * @param int $idStage L'identifiant du stage auquel ajouter l'activité.
98 * @param int $idActivity L'identifiant de l'activité à ajouter.
99 * @return bool Retourne vrai si l'ajout est réussi, sinon retourne faux.
100 */
101 function addActivity($idStage, $idActivity)
102 {
103     // Requête SQL pour insérer une nouvelle entrée dans la table 'stages_has_activity'
104     $sql = "INSERT INTO stages_has_activity (`stages_id`, `activity_id`) VALUES(:is ,:ia);
";
105     $statement = EDatabase::prepare($sql, array(PDO::ATTR_CURSOR => PDO::CURSOR_FWDONLY));
106     try {
107         $tab = array(':is' => $idStage, ':ia' => $idActivity);
108         foreach ($tab as $key => $value) {
109             $statement->bindValue($key, $value);
110         }
111         $statement->execute();
112         // Retourne vrai si l'ajout est réussi
113         return true;
114     } catch (PDOException $error) {
115         // En cas d'erreur, retourne faux
116         return false;
117     }
118 }
119 /**
120 * Supprime un stage de la table 'stages' en fonction de son identifiant.
121 *
122 * @param int $id L'identifiant du stage à supprimer.
123 * @return bool Retourne vrai si la suppression du stage est réussie, sinon retourne faux.
124 */
125 function deleteStageById($id)
126 {
127     // Requête SQL pour supprimer le stage avec l'identifiant donné
128     $sql = "DELETE
129     FROM `stages`"

```

```

131     WHERE `id` = :i";
132     $statement = EDatabase::prepare($sql, array(PDO::ATTR_CURSOR => PDO::CURSOR_FWDONLY));
133     try {
134         $tab = array(':i' => $id);
135         foreach ($tab as $key => $value) {
136             $statement->bindValue($key, $value);
137         }
138         $statement->execute();
139         // Retourne vrai si la suppression du stage est réussie
140         return true;
141     } catch (PDOException $e) {
142         // En cas d'erreur, retourne faux
143         return false;
144     }
145 }
146
147 /**
148 * Récupère les informations d'une étape à partir de son identifiant.
149 *
150 * @param int $idStage L'identifiant du stage à récupérer.
151 * @return array|bool Retourne un tableau contenant un objet Stage représentant le stage
correspondant à l'identifiant donné, ou faux en cas d'erreur.
152 */
153 function getStageById($idStage)
154 {
155     $travelStage = [];
156     // Requête SQL pour sélectionner les informations du stage avec l'identifiant donné
157     $sql = "SELECT `id`, `title`, `date`, `duration_days`, `description`, `travels_id`"
158     FROM `stages`
159     WHERE `id` = :i";
160     $statement = EDatabase::prepare($sql, array(PDO::ATTR_CURSOR => PDO::CURSOR_FWDONLY));
161     try {
162         $tab = array(":i" => $idStage);
163         foreach ($tab as $key => $value) {
164             $statement->bindValue($key, $value);
165         }
166         $statement->execute();
167     } catch (PDOException $e) {
168         return false;
169     }
170     while ($row = $statement->fetch(PDO::FETCH_ASSOC, PDO::FETCH_ORI_NEXT)) {
171         // Ajoute un nouvel objet Stage au tableau $travelStage en utilisant les données
de la base de données
172         $travelStage[] = new Stage($row['id'], $row['title'], $row['date'], $row['
duration_days'], $row['description'], $row['travels_id']);
173     }
174     // Retourne le tableau contenant l'objet Stage représentant le stage correspondant à
l'identifiant donné
175     return $travelStage;
176 }
177
178 /**
179 * Met à jour les informations d'une étape dans la table 'stages'.
180 *
181 * @param int $id L'identifiant du stage à mettre à jour.
182 * @param string $title Le titre du stage.
183 * @param string $date La date du stage (au format YYYY-MM-DD).
184 * @param int $durationDays La durée du stage en jours.
185 * @param string $description La description du stage.
186 * @return bool Retourne vrai si la mise à jour est réussie, sinon retourne faux.
187 */

```

```

188 function stageUpdate($id, $title, $date, $durationDays, $description)
189 {
190     // Requête SQL pour mettre à jour les informations du stage
191     $sql = "UPDATE stages
192         SET `title` = :ti, `date` = :da, `duration_days` = :du, `description` = :de
193         WHERE `id` = :i";
194     $statement = EDatabase::prepare($sql, array(PDO::ATTR_CURSOR => PDO::CURSOR_FWDONLY));
195     try {
196         $tab = array(':ti' => $title, ':i' => $id, ':da' => $date, ':du' => $durationDays,
197 ':de' => $description);
198         foreach ($tab as $key => $value) {
199             $statement->bindValue($key, $value);
200         }
201         $statement->execute();
202         // Retourne true si l'insertion est réussie
203         return true;
204     } catch (PDOException $error) {
205         // En cas d'erreur, retourne false
206         return false;
207     }
208 }
```

VoyageMemo/app/model/stagesHasActivityClass.php

```

1 <?php
2 /**
3 * TITRE : VoyageMémo
4 * Auteur : Guilain Fernandez
5 * Description : la class StageHasActivity
6 * Date : 06/05/2024
7 */
8 class StageHasActivity
9 {
10     public function __construct($InStageId = "", $InActivityId = "")
11     {
12         $this->stagesId = $InStageId;
13         $this->activityId = $InActivityId;
14     }
15
16     public $stagesId;
17
18     public $activityId;
19 }
20
21 /**
22 * Supprime l'association entre un stage et une activité dans la table 'stages_has_activity'.
23 *
24 * @param int $stageId L'identifiant du stage.
25 * @param int $activityId L'identifiant de l'activité.
26 * @return bool Retourne vrai si la suppression de l'association est réussie, sinon
retourne faux.
27 */
28 function deleteStageActivityAssociationByStageIdAndActivityId($stageId, $activityId)
29 {
30     // Requête SQL pour supprimer l'association entre le stage et l'activité
31     $sql = "DELETE
32         FROM `stages_has_activity`
33         WHERE `stages_id` = :si
```

```

34     AND `activity_id` = :ai";
35     $statement = EDatabase:::prepare($sql, array(PDO::ATTR_CURSOR => PDO::CURSOR_FWDONLY));
36     try {
37         $tab = array(':si' => $stageId, ':ai' => $activityId);
38         foreach ($tab as $key => $value) {
39             $statement->bindValue($key, $value);
40         }
41         $statement->execute();
42         // Retourne vrai si la suppression de l'association est réussie
43         return true;
44     } catch (PDOException $e) {
45         // En cas d'erreur, retourne faux
46         return false;
47     }
48 }

```

VoyageMemo/app/model/travelClass.php

```

1 <?php
2
3 /**
4  * TITRE : VoyageMémo
5  * Auteur : Guilain Fernandez
6  * Description : la class des voyage
7  * Date : 29/04/2024
8  */
9 class Travel
10 {
11     public function __construct($InId = -1, $InTitle = "", $InNumberDisplay, $InUserId)
12     {
13         $this->id = $InId;
14         $this->title = $InTitle;
15         $this->numberDisplay = $InNumberDisplay;
16         $this->userId = $InUserId;
17     }
18
19     public $id;
20
21     public $title;
22
23     public $numberDisplay;
24
25     public $userId;
26 }
27
28 /**
29 * Crée un nouveau voyage dans la base de données avec le titre, le nombre d'affichage et
l'ID de l'utilisateur spécifiés.
30 *
31 * @param string $title Le titre du voyage
32 * @param int $numberDisplay Le nombre d'affichages pour le voyage
33 * @param int $userId L'ID de l'utilisateur associé au voyage
34 * @return bool Retourne true si le voyage est créé avec succès, sinon false.
35 */
36 function createTravel($title, $numberDisplay, $userId)
37 {
38     // Requête SQL pour insérer un nouveau voyage dans la table 'travels'
39     $sql = "INSERT INTO travels (`title`, `number_display`, `users_id`) VALUES(:t, :nd,
:u);";

```

```

40     $statement = EDatabase:::prepare($sql, array(PDO::ATTR_CURSOR => PDO::CURSOR_FWDONLY));
41     try {
42         // Tableau des paramètres à lier à la requête
43         $tab = array(':t' => $title, ':nd' => $numberDisplay, ':u' => $userId);
44         foreach ($tab as $key => $value) {
45             $statement->bindValue($key, $value);
46         }
47         $statement->execute();
48         // Retourne true si l'insertion est réussie
49         return true;
50     } catch (PDOException $error) {
51         // En cas d'erreur, retourne false
52         return false;
53     }
54 }
55
56 /**
57 * Récupère tous les voyages depuis la base de données.
58 *
59 * @return array|bool Retourne un tableau d'objets Travel contenant tous les voyages si
60 * trouvés,
61 *                     sinon retourne false en cas d'erreur ou si aucun voyage n'est
62 * trouvé.
63 */
64 function getAllTravel()
65 {
66     $allTravel = [];
67     // Requête SQL pour récupérer tous les voyages de la table 'travels'
68     $sql = "SELECT `id`, `title`, `number_display`, `users_id`"
69     FROM `travels`";
70     $statement = EDatabase:::prepare($sql, array(PDO::ATTR_CURSOR => PDO::CURSOR_FWDONLY));
71     try {
72         $tab = array();
73         foreach ($tab as $key => $value) {
74             $statement->bindValue($key, $value);
75         }
76         $statement->execute();
77     } catch (PDOException $e) {
78         // En cas d'erreur, retourne false
79         return false;
80     }
81     while ($row = $statement->fetch(PDO::FETCH_ASSOC, PDO::FETCH_ORI_NEXT)) {
82         // Crée un objet Travel avec les données récupérées de la base de données
83         $allTravel[] = new Travel($row['id'], $row['title'], $row['number_display'],
84         $row['users_id']);
85     }
86     // Retourne le tableau de tous les voyages (peut être vide si aucun voyage n'est
87     trouvé)
88     return $allTravel;
89 }
90
91 /**
92 * Récupère les voyages associés à un utilisateur à partir de son identifiant dans la
93 * table 'travels'.
94 *
95 * @param int $userId L'identifiant de l'utilisateur.
96 * @return array|bool Retourne un tableau contenant des objets Travel représentant les
97 * voyages associés à l'utilisateur, ou faux en cas d'erreur.
98 */
99 function getTravelByUserId($userId)
100 {

```

```

95     $allTravel = [];
96     // Requête SQL pour récupérer les voyages associés à l'utilisateur avec l'identifiant
97     // donné
98     $sql = "SELECT `id`, `title`, `number_display`
99     FROM `travels`
100    WHERE `users_id` = :ui";
101   $statement = EDatabase::prepare($sql, array(PDO::ATTR_CURSOR => PDO::CURSOR_FWDONLY));
102   try {
103     $tab = array(':ui' => $userId);
104     foreach ($tab as $key => $value) {
105       $statement->bindValue($key, $value);
106     }
107     $statement->execute();
108   } catch (PDOException $e) {
109     // En cas d'erreur, retourne faux
110     return false;
111   }
112   while ($row = $statement->fetch(PDO::FETCH_ASSOC, PDO::FETCH_ORI_NEXT)) {
113     // Ajoute un nouvel objet Travel au tableau $allTravel en utilisant les données de
114     // la base de données
115     $allTravel[] = new Travel($row['id'], $row['title'], $row['number_display'],
116     $row['users_id']);
117   }
118 }
119 /**
120 * Récupère un voyage à partir de son identifiant dans la table 'travels'.
121 *
122 * @param int $id L'identifiant du voyage à récupérer.
123 * @return array|bool Retourne un tableau contenant un objet Travel représentant le voyage
124 trouv , ou faux si aucun voyage n'est trouvé ou en cas d'erreur.
125 */
126 function getTravelById($id)
127 {
128   $allTravel = [];
129   // Requête SQL pour récupérer le voyage avec l'identifiant donné
130   $sql = "SELECT `id`, `title`, `number_display`, `users_id`
131   FROM `travels`
132   WHERE id = :i";
133   $statement = EDatabase::prepare($sql, array(PDO::ATTR_CURSOR => PDO::CURSOR_FWDONLY));
134   try {
135     $tab = array(':i' => $id);
136     foreach ($tab as $key => $value) {
137       $statement->bindValue($key, $value);
138     }
139     $statement->execute();
140   } catch (PDOException $e) {
141     // En cas d'erreur, retourne false
142     return false;
143   }
144   while ($row = $statement->fetch(PDO::FETCH_ASSOC, PDO::FETCH_ORI_NEXT)) {
145     // Ajoute un nouvel objet Travel au tableau $allTravel en utilisant les données de
146     // la base de données
147     $allTravel[] = new Travel($row['id'], $row['title'], $row['number_display'],
148     $row['users_id']);
149   }
150   // Retourne le tableau contenant un objet Travel représentant le voyage trouvé, ou
151   // faux si aucun voyage n'est trouvé ou en cas d'erreur
152   return $allTravel;

```

```

149 }
150
151 /**
152 * Supprime un voyage de la table 'travels' en fonction de son identifiant.
153 *
154 * @param int $id L'identifiant du voyage à supprimer.
155 * @return bool Retourne vrai si la suppression du voyage est réussie, sinon retourne
156 faux.
157 */
158 function deleteTravelById($id)
159 {
160     // Requête SQL pour supprimer le voyage avec l'identifiant donné
161     $sql = "DELETE
162         FROM `travels`
163         WHERE `id` = :i";
164     $statement = EDatabase::prepare($sql, array(PDO::ATTR_CURSOR => PDO::CURSOR_FWDONLY));
165     try {
166         $tab = array(':i' => $id);
167         foreach ($tab as $key => $value) {
168             $statement->bindValue($key, $value);
169         }
170         $statement->execute();
171         // Retourne vrai si la suppression du voyage est réussie
172         return true;
173     } catch (PDOException $e) {
174         // En cas d'erreur, retourne faux
175         return false;
176     }
177 }
178 /**
179 * Met à jour le titre d'un voyage dans la table 'travels'.
180 *
181 * @param int $id L'identifiant du voyage à mettre à jour.
182 * @param string $title Le nouveau titre du voyage.
183 * @return bool Retourne vrai si la mise à jour du voyage est réussie, sinon retourne
184 faux.
185 */
186 function travelUpdate($id, $title)
187 {
188     // Requête SQL pour mettre à jour le titre du voyage dans la table 'travels'
189     $sql = "UPDATE
190         travels
191         SET `title` = :t
192         WHERE `id` = :i";
193     $statement = EDatabase::prepare($sql, array(PDO::ATTR_CURSOR => PDO::CURSOR_FWDONLY));
194     try {
195         // Tableau des paramètres à lier à la requête
196         $tab = array(':t' => $title, ':i' => $id);
197         foreach ($tab as $key => $value) {
198             $statement->bindValue($key, $value);
199         }
200         $statement->execute();
201         // Retourne vrai si la mise à jour du voyage est réussie
202         return true;
203     } catch (PDOException $error) {
204         // En cas d'erreur, retourne faux
205         return false;
206     }

```

```

207 /**
208  * Met à jour le nombre de visite d'un voyage dans la table 'travels'.
209  *
210  * @param int $id L'identifiant du voyage à mettre à jour.
211  * @param string $title Le nouveau titre du voyage.
212  * @return bool Retourne vrai si la mise à jour du voyage est réussie, sinon retourne
213 faux.
214 */
215 function travelNumberDisplayUpdate($id, $numberDisplay)
216 {
217     // Requête SQL pour mettre à jour le nombre de visite du voyage dans la table 'travels'
218     $sql = "UPDATE
219         travels
220         SET `number_display` = :nd
221         WHERE `id` = :i";
222     $statement = EDatabase:::prepare($sql, array(PDO::ATTR_CURSOR => PDO::CURSOR_FWDONLY));
223     try {
224         // Tableau des paramètres à lier à la requête
225         $tab = array(':nd' => $numberDisplay, ':i' => $id);
226         foreach ($tab as $key => $value) {
227             $statement->bindValue($key, $value);
228         }
229         $statement->execute();
230         // Retourne vrai si la mise à jour du voyage est réussie
231         return true;
232     } catch (PDOException $error) {
233         // En cas d'erreur, retourne faux
234         return false;
235     }
236 }
237

```

VoyageMemo/app/public/css/all.css

```

1 /**
2  * TITRE : VoyageMémo
3  * Auteur : Guilain Fernandez
4  * Description : le css qui s'appliquera a toute les pages
5  * Date : 25/04/2024
6 */
7
8 {
9     margin: 0;
10    padding: 0;
11}
12
13 body {
14     width: 100vw;
15     height: 100vh;
16}
17
18 nav {
19     background-color: #3cdbd1;
20     height: 8vh;
21}
22
23 nav img{

```

```

24     width: 60px;
25     height: 60px;
26   }
27   .card{
28     border: none;
29     background-color: □ aliceblue
30   }
31   .card img{
32     width: 100;
33     height: 75;
34   }
35
36   .pictureList{
37     height: 15vh;
38   }
39
40   .modifyStage{
41     height: 90%;
42   }
43
44   .dropdown-menu>li {
45     margin: 5px;
46   }
47
48   .password-field {
49     position: relative;
50   }
51
52 /*css pour icone voir le mots de passe*/
53   .password-field .password-icon {
54     display: flex;
55     position: absolute;
56     top: 50%;
57     right: 20px;
58     transform: translateY(-50%);
59     width: 20px;
60     color: ■ black
61     transition: all 0.2s;
62   }
63
64 /*css pour icone randome mots de passe*/
65   .password-field .random-icon {
66     display: flex;
67     position: absolute;
68     top: 50%;
69     right: 50px;
70     transform: translateY(-50%);
71     width: 20px;
72     color: ■ black
73     transition: all 0.2s;
74   }
75
76   .password-field .password-icon .bi-eye-slash {
77     display: none;
78   }
79

```

```

1 <!--
2     TITRE : VoyageMémo
3     Auteur : Guilain Fernandez
4     Description : la page accueil du site web
5     Date : 24/04/2024
6 -->
7
8
9 <header>
10    <nav class="navbar navbar-expand-lg">
11        <div class="container">
12            <a class="navbar-brand" href="#">
13                
15            </a>
16            <ul class="navbar-nav">
17                </ul>
18                <div class="dropdown">
19                    <a class="nav-link" href="#" role="button" data-bs-toggle="dropdown" aria-
expanded="false">
20                        
21                    </a>
22                    <ul class="dropdown-menu dropdown-menu-end">
23                        <?php
24                        if ($_SESSION['userLogin'] != "") {
25                            ?>
26                            <li><em><?= $_SESSION['userLogin'] ?></em></li>
27                            <li>
28                                <hr class="dropdown-divider">
29                            </li>
30                            <li><a class="nav-link link-primary" href="index.php?page=profil">
31                                Profil</a></li>
32                            <li>
33                                <hr class="dropdown-divider">
34                            </li>
35                            <li>
36                                <form action="" method="post">
37                                    <input class="btn btn-primary" type="submit" name="
disconnect" value="Déconnexion">
38                                </form>
39                            </li>
40                        <?php
41                        } else {
42                            ?>
43                            <li><a class="nav-link link-primary" href="index.php?page=
loginUser">Connexion</a></li>
44                            <li><a class="nav-link link-primary" href="index.php?page=
newUserAccount">Créer un compte</a></li>
45                        <?php
46                        }
47                    </li>
48                </ul>
49            </div>
50        </nav>
51    </header>
52    <main class="m-3">
53        <div class="container">
54            <?php
55            if ($_SESSION['userLogin'] != "") {

```

```

56      ?>
57      <div class="position-relative">
58          <a class="btn btn-primary position-absolute top-0 end-0" href="
59 index.php?page=newTravel">Nouveau voyage</a>
60      </div>
61      <?php
62      }
63      ?>
64      <div class="row">
65          <?php
66          // Récupération de tous les voyages
67          $allTravel = getAllTravel();
68          foreach ($allTravel as $travelKey => $travelValue) {
69          ?>
70          <div class="col-sm-12 col-md-4 col-lg-3">
71              <div class="card mt-5">
72                  <div class="card-body">
73                      <h4 class="card-title text-center"><?= $travelValue->title ?>
74                  /h4>
75                      <?php
76                      // Récupération des étapes du voyage
77                      $travelStage = getStageByTravelId($travelValue->id);
78                      // réinitialise les variables a chaque nouveau voyage
79                      $stageMediaThumbnail = array();
80                      $activity = "";
81                      $travelDuration = 0;
82                      $travelStartDate = new DateTime("0000-00-00");
83                      $travelEndDate = new DateTime("0000-00-00");
84                      foreach ($travelStage as $stageKey => $stageValue) {
85                          // Récupération des activités et des médias des étapes
86                          $stageActivity = getActivityById($stageValue->id);
87                          $stageMedia = getMediaById($stageValue->id);
88                          $travelDuration = $travelDuration += $stageValue->
89                          durationDays;
90
91                          foreach ($stageMedia as $key => $stageMediaValue) {
92                              //on met le nom des vignettes de chaque étape dans un
93                              tableau
94                              filenameVignette);
95                          }
96
97                          foreach ($stageActivity as $key => $stageActivityValue) {
98                              $activity = $activity . '<p class="card-text col-6">' .
99                              $stageActivityValue->name . '</p>';
100                         }
101
102                         if ($travelStartDate == new DateTime("0000-00-00")) {
103                             $travelStartDate = $stageValue->date;
104                             $travelStartDate = new DateTime($travelStartDate);
105                         }
106
107                         // Sélection aléatoire d'une vignette pour le voyage
108                         if ($stageMediaThumbnail != null) {
109                             $thumbnail = array_rand($stageMediaThumbnail, 1);
110                         }
111
112                         $travelDurationInterval = new DateInterval("P" .
113                         $travelDuration . "D");

```

```

111         $startDateFormat = $travelStartDate->format("d/m/Y");
112         $travelEndDate = $travelStartDate->
113             add($travelDurationInterval);
114             $endDateFormat = $travelEndDate->format("d/m/Y");
115
116             if ($travelStartDate == new DateTime("0000-00-00")) {
117                 $startDateFormat = "00/00/0000";
118             }
119
120             if ($travelEndDate == new DateTime("0000-00-00")) {
121                 $endDateFormat = "00/00/0000";
122             }
123         ?>
124         <p class="card-text">Début : <em><?= $startDateFormat ?></em><
125 /p>
126     ?> jours</em></p>
127     numberDisplay?></p>
128
129
130
131         <div class="row">
132             <?= $activity ?>
133         </div>
134         <div class="mt-3">
135             <?php if ($stageMediaThumbnail != null) { ?>
136                 
138                 <?php } ?>
139             </div>
140             <div class="text-end"><a class="card-link" href="index.php?
141 page=detailTravel&id=<?= $travelValue->id ?>">Détail-></a></div>
142         </div>
143     </div>
144 </main>

```

VoyageMemo/app/view/loginUser.php

```

1 <!--
2     TITRE : VoyageMémo
3     Auteur : Guilain Fernandez
4     Description : la page qui permet de se connecter à son compte
5     Date : 24/04/2024
6 -->
7
8 <main class="container position-relative w-100 h-100">
9     <div class="row text-bg-info justify-content-center w-75 h-50 position-absolute top-50
10        start-50 translate-middle">
11            <div class="col-md-9">
12                <h1 class="text-center text-white">Connexion</h1>
13                <form class="h-75 d-flex flex-column justify-content-around" action="" method="
14 post">
15                    <input type="text" class="form-control" name="userLogin" id="userLogin"
16 placeholder="Entrez votre identifiant">
17                    <input type="password" class="form-control" name="password" id="password"
18 placeholder="Entrer votre mot de passe">

```

```

15             <button type="submit" name="login" class="btn btn-primary">Se connecter<
16         <?= $error ?>
17     </form>
18     <div class="d-flex justify-content-between">
19         <a class="text-end text-white" href="index.php?page=newUserAccount">Créer
un compte</a>
20         <a class="text-white" href="index.php">Accueil</a>
21     </div>
22 </div>
23 </div>
24 </main>

```

VoyageMemo/app/view/modifyStage.php

```

1 <!--
2     TITRE : VoyageMémo
3     Auteur : Guilain Fernandez
4     Description : la page qui permet de modifier une étape
5     Date : 05/06/2024
6 -->
7
8 <?php
9 if ($_SESSION['userLogin'] == "") {
10     header('Location: index.php?page=home');
11 }
12 $stage = getStageById($_GET["id"]);
13 $stageMedia = getMediaByStageId($stage[0]->id);
14 $date = date_create($stage[0]->date);
15 $durationToString = strval($stage[0]->durationDays);
16 date_modify($date, '+' . $durationToString . 'days');
17 $endDate = $date->format("Y-m-d");
18 ?>
19 <main class="container position-relative vw-100 vh-100 p-0">
20     <div class="row text-bg-info justify-content-center w-75 h-75 position-absolute top-50
start-50 translate-middle">
21         <div class="col-md-9 p-0">
22             <h2 class="text-center text-white">Modifier l'étape d'un voyage</h2>
23             <form class="modifyStage d-flex flex-column justify-content-around" action="">
24                 <input type="text" class="form-control" name="stageTitle" id="stageTitle"
value="<?= $stage[0]->title ?>" placeholder="Titre de l'étape">
25                 <textarea class="form-control" name="stageDescription" placeholder="Description"
rows="5"><?= $stage[0]->description ?></textarea>
26                 <div class="input-group">
27                     <span class="input-group-text">Début</span>
28                     <input type="date" class="form-control" name="startDate" value="<?=
$stage[0]->date ?>">
29                     <span class="input-group-text">Fin</span>
30                     <input type="date" class="form-control" name="endDate" value="<?=
$endDate ?>">
31                 </div>
32                 <select class="form-select form-select-lg" multiple aria-label="Multiple
select activity" name="activity[]" id="activity">
33                     <option disabled>Choisis tes activités</option>
34                     <?php
35                         $allActivity = getAllActivity();
36                         $stageActivity = getActivityByStageId($stage[0]->id);
37
38                     <foreach ($allActivity as $ActivityValue) {
39                         $selected = false;

```

```

40         foreach ($stageActivity as $stageActivityValue) {
41             if ($stageActivityValue->id == $ActivityValue->id) {
42                 $selected = true;
43                 break;
44             }
45         }
46     ?>
47     <option <?= $selected ? 'selected' : '' ?> value=<?=
$ActivityValue->id ?><?= $ActivityValue->name ?></option>
48         <?php
49     }
50     ?>
51     </select>
52     <div class="pictureList row overflow-auto d-flex justify-content-center">
53         <?php
54         foreach ($stageMedia as $keyMedia => $valueMedia) {
55             ?
56             <div class="card col-5 col-md-4 col-lg-3 col-xl-2 m-1">
57                 
58                 <p class="text-center m-1">Supprimer</p>
59                 <input class="text-center m-1" type="checkbox" name="media[]" value=<?= $valueMedia->id ?>">
60             </div>
61             <?php
62             ?
63             ?
64             </div>
65             <input class="form-control" type="file" name="stageMedia[]" id="stageMedia"
accept="image/png, image/jpg, image/jpeg" multiple>
66             <input type="hidden" name="stageId" value=<?= $_GET["id"] ?>">
67             <button type="submit" name="modifyStage" class="btn btn-primary">Modifier
l'étape</button>
68             <?= $error ?>
69         </form>
70         <div class="d-flex justify-content-between">
71             <a class="text-end text-white" href="index.php?page=profil">Profil</a>
72             <a class="text-white" href="index.php">Accueil</a>
73         </div>
74     </div>
75     </div>
76 </main>

```

VoyageMemo/app/view/modifyTravel.php

```

1  <!--
2      TITRE : VoyageMémo
3      Auteur : Guilain Fernandez
4      Description : la page qui permet de modifier un voyage
5      Date : 24/04/2024
6  -->
7
8  <?php
9  if ($_SESSION['userLogin'] == "") {
10     header('Location: index.php?page=home');
11 }
12 $travel = getTravelById($_GET["id"]);
13 ?>
14 <main class="container position-relative w-100 h-100">

```

```

15 <div class="row text-bg-info justify-content-center w-75 h-25 position-absolute top-50
16 start-50 translate-middle">
17     <div class="col-md-9">
18         <h1 class="text-center text-white">Modifier le voyage</h1>
19         <form class="h-75 d-flex flex-column justify-content-around" action="" method="post">
20             <input type="text" class="form-control" name="travelTitle" id="travelTitle" value="<?= $travel[0]->title?>">
21             <input type="hidden" name="travelId" value="<?= $_GET["id"]?>">
22             <button type="submit" name="modifyTravel" class="btn btn-primary">Modifier
le nouveau voyage</button>
23             <div class="text-danger" id="error"><?= $error ?></div>
24         </form>
25         <div class="d-flex justify-content-between">
26             <a class="text-end text-white" href="index.php?page=profil">Profil</a>
27             <a class="text-white" href="index.php">Accueil</a>
28         </div>
29     </div>
30 </main>

```

VoyageMemo/app/view/newStage.php

```

1 <!--
2     TITRE : VoyageMémo
3     Auteur : Guilain Fernandez
4     Description : la page qui permet d'ajouter des étapes aux voyages'
5     Date : 29/04/2024
6 -->
7 <?php
8 if ($_SESSION['userLogin'] == "") {
9     header('Location: index.php?page=home');
10 }
11 ?>
12 <main class="container position-relative w-100 h-100">
13     <div class="row text-bg-info justify-content-center w-75 h-75 position-absolute top-50
start-50 translate-middle">
14         <div class="col-md-9">
15             <h2 class="text-center text-white">Nouvelle étape d'un voyage</h2>
16             <form class="h-75 d-flex flex-column justify-content-around" action="" method="post" enctype="multipart/form-data">
17                 <?php
18                     $travel = getTravelById($_GET["id"]);
19                     ?>
20                     <input type="text" class="form-control" value="<?= $travel[0]->title ?>" disabled="disabled">
21                     <input type="hidden" class="form-control" name="travelId" value="<?= $travel[0]->id ?>">
22                     <input type="text" class="form-control" name="stageTitle" placeholder="Titre de l'étape">
23                     <textarea class="form-control" name="stageDescription" placeholder="Description" id="" rows="5"></textarea>
24                     <div class="input-group">
25                         <span class="input-group-text">Début</span>
26                         <input type="date" class="form-control" name="startDate">
27                         <span class="input-group-text">Fin</span>
28                         <input type="date" class="form-control" name="endDate">
29                     </div>
30                     <select class="form-select form-select-lg" multiple aria-label="Multiple
select activity" name="activity[]">
31                         <option disabled>Choisis tes activités</option>

```

```

32             <?php
33                 $allActivity = getAllActivity();
34                 foreach ($allActivity as $key => $value) {
35                     ?>
36                         <option value="<?= $value->id ?>"><?= $value->name ?></option>
37                     <?php
38                     }
39                     ?>
40                 </select>
41                 <input class="form-control" type="file" name="stageMedia[]" accept="image/png, image/jpg, image/jpeg" multiple>
42                     <button type="submit" name="newStage" class="btn btn-primary">Créer la
nouvelle étape</button>
43                     <?= $error ?>
44                 </form>
45                 <div class="d-flex justify-content-between">
46                     <a class="text-end text-white" href="index.php?page=profil">Profil</a>
47                     <a class="text-white" href="index.php">Accueil</a>
48                 </div>
49             </div>
50         </div>
51     </main>

```

VoyageMemo/app/view/newTravel.php

```

1      <!--
2          TITRE : VoyageMémo
3          Auteur : Guilain Fernandez
4          Description : la page qui permet de créer un voyage
5          Date : 24/04/2024
6      -->
7      <?php
8      if ($_SESSION['userLogin'] == "") {
9          header('Location: index.php?page=home');
10     }
11     ?>
12     <main class="container position-relative w-100 h-100">
13         <div class="row text-bg-info justify-content-center w-75 h-50 position-absolute top-50
start-50 translate-middle">
14             <div class="col-md-9">
15                 <h1 class="text-center text-white">Nouveau voyage</h1>
16                 <form class="h-75 d-flex flex-column justify-content-around" action="" method="post">
17                     <input type="text" class="form-control" name="travelTitle" id="travelTitle"
placeholder="Titre du voyage">
18                     <button type="submit" name="newTravel" class="btn btn-primary">Créer le
nouveau voyage</button>
19                     <div class="text-danger" id="error"><?= $error ?></div>
20                 </form>
21                 <div class="d-flex justify-content-between">
22                     <a class="text-end text-white" href="index.php?page=profil">Profil</a>
23                     <a class="text-white" href="index.php">Accueil</a>
24                 </div>
25             </div>
26         </div>
27     </main>

```

VoyageMemo/app/view/newUserAccount.php

```

1 <!--
2   TITRE : VoyageMémo
3   Auteur : Guilain Fernandez
4   Description : la page qui permet de créer un nouveau compte
5   Date : 24/04/2024
6 -->
7   <main class="container position-relative w-100 h-100">
8     <div class="row text-bg-info justify-content-center w-75 h-50 position-absolute top-50 start-50 translate-middle">
9       <div class="col-md-9">
10         <h1 class="text-center text-white">Créer un compte</h1>
11         <form class="h-75 d-flex flex-column justify-content-around" action="" method="post">
12           <input type="text" class="form-control" name="login" id="login" placeholder="Entrez votre identifiant">
13           <div class="password-field">
14             <input type="password" value="" class="form-control" name="password" id="password" placeholder="Entrez votre mot de passe">
15             <div class="random-icon" onclick="randomString()" data-bs-toggle="tooltip" data-bs-html="true" title="Générer un mot de passe">
16               <svg xmlns='http://www.w3.org/2000/svg' width='20' height='20' fill='currentColor' class='bi bi-shuffle' viewBox='0 0 16 16'>
17                 <path fill-rule='evenodd' d='M0 3.5A.5.5 0 0 1 .5 3H1c2.202 0 3.827 1.24 4.874 2.418A9.552.865 1.102 1.126 1.532.26-.43.636-.98 1.126-1.532C9.173 4.24 10.798 3 13 3v1c-1.798 0-3.173 1.01-4.126 2.082A9.6 9.6 0 0 0 7.556 8a9.6 9.6 0 0 0 1.317 1.918C9.828 10.99 11.204 12 13 12v1c-2.202 0-3.827-1.24-4.874-2.418A10.6 10.6 0 0 1 7 9.05c-.26.43-.636.98-1.126 1.532C4.827 11.76 3.202 13 1 13H.5a.5.5 0 0 1 0-1H1c1.798 0 3.173-1.01 4.126-2.082A9.6 9.6 0 0 0 6.444 8a9.6 9.6 0 0 0-1.317-1.918C4.172 5.01 2.796 4 1 4H.5a.5.5 0 0 1-5-.5' />
18               <path d='M13 5.466V1.534a.25.25 0 0 1 .41-.192l2.36 1.966c.12.1.12.284 0 .384l-2.36 1.966a.25.25 0 0 1 -.41-.192m0 9v-3.932a.25.25 0 0 1 .41-.192l2.36 1.966c.12.1.12.284 0 .384l-2.36 1.966a.25.25 0 0 1 -.41-.192' />
19             </svg>
20           </div>
21           <div class="password-icon" onclick="pswVisibility()">
22             <svg id="eye" xmlns='http://www.w3.org/2000/svg' width="20" height="20" fill="currentColor" class="bi bi-eye" viewBox='0 0 16 16'>
23               <path d="M16 8s-3-5.5-8-5.5S0 8 0 8s3 5.5 8 5.5S16 8 16 8M1.173 8a13 13 0 0 1 1.66-2.043C4.12 4.668 5.88 3.5 8 3.5s3.879 1.168 5.168 2.457A13 13 0 0 1 14.828 8q-.086.13-.195.288c-.335.48-.83 1.12-1.465 1.755C11.879 11.332 10.119 12.5 8 12.5s-3.879-1.168-5.168-2.457A13 13 0 0 1 1.172 8z" />
24               <path d="M8 5.5a2.5 2.5 0 1 0 0 5 2.5 2.5 0 0 0 0-5M4.5 8a3.5 3.5 0 1 1 7 0 3.5 3.5 0 0 1-7 0" />
25             </svg>
26             <svg id="eyeOff" xmlns='http://www.w3.org/2000/svg' width="20" height="20" fill="currentColor" class="bi bi-eye-slash" viewBox='0 0 16 16'>
27               <path d="M13.359 11.238C15.06 9.72 16 8 16 8s-3-5.5-8-5.5a7 7 0 0 2.79.588l.77.771A6 6 0 0 1 8 3.5c2.12 0 3.879 1.168 5.168 2.457A13 13 0 0 1 14.828 8q-.086.13-.195.288c-.335.48-.83 1.12-1.465 1.755q-.247.248-.517.486z" />
28               <path d="M11.297 9.176a3.5 3.5 0 0 0-4.474-4.474l.823.823a2.5 2.5 0 0 1 2.829 2.829zm-2.943 1.299.822.822a3.5 3.5 0 0 1-4.474-4.474l.823.823a2.5 2.5 0 0 0 2.829 2.829" />
29               <path d="M3.35 5.47q-.27.24-.518.487A13 13 0 0 0 1.172 81.195.288c.335.48.83 1.12 1.465 1.755C4.121 11.332 5.881 12.5 8 12.5c.716 0 1.39-.133 2.02-.361.77.772A7 7 0 0 1 8 13.5C3 13.5 0 8 0 8s.939-1.721 2.641-3.238l.708.709zm10.296 8.884-12-12 .708-.708 12 12z" />
30             </svg>
31           </div>
32         </div>
33         <input type="password" class="form-control" name="validatePsw" id="validatePsw" placeholder="Confirmez votre mot de passe">
34         <button type="submit" name="newAccount" class="btn btn-primary">S'inscrire</button>
35         <div class="text-danger" id="error"><?= $error ?></div>
36       </form>
37       <div class="d-flex justify-content-between">
```

```

38         <a class="text-end text-white" href="index.php?page=loginUser">Connexion<
39     /a>
40             <a class="text-white" href="index.php">Accueil</a>
41         </div>
42     </div>
43 </main>
44
45 <script>
46     var eye = document.getElementById("eye");
47     var eyeoff = document.getElementById("eyeOff");
48     var pswType = document.getElementById("password");
49
50     function pswVisibility() {
51         //quand on clique on m'est le type de l'input en text
52         if (password.type == "password") {
53             eye.style.display = "none";
54             eyeoff.style.display = "block";
55             pswType.type = "text";
56         }
57         //quand on clique on mest le type de l'input en password
58         else {
59             eyeoff.style.display = "none";
60             eye.style.display = "block";
61             pswType.type = "password";
62         }
63     }
64
65     function randomString() {
66         let result = '';
67         let length = 15;
68         const characters = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789%/?!*#';
69         const charactersLength = characters.length;
70         let counter = 0;
71         while (counter < length) {
72             result += characters.charAt(Math.floor(Math.random() * charactersLength));
73             counter += 1;
74         }
75         pswType.value = result;
76     }
77 </script>

```

VoyageMemo/app/view/profil.php

```

1 <!--
2     TITRE : VoyageMémo
3     Auteur : Guilain Fernandez
4     Description : la page profile de l'utilisateur connecter
5     Date : 24/04/2024
6 -->
7 <?php
8 if ($_SESSION['userLogin'] == "") {
9     header('Location: index.php?page=home');
10 }
11 ?>
12 <header>
13     <nav class="navbar navbar-expand-lg">
14         <div class="container">

```

```

15      <a class="navbar-brand" href="index.php?page=home">
16          
17      </a>
18      <ul class="navbar-nav">
19          <li><a class="nav-link fs-3" href="index.php?page=home">Accueil</a></li>
20      </ul>
21      <div class="dropdown">
22          <a class="nav-link" href="#" role="button" data-bs-toggle="dropdown" aria-
expanded="false">
23              
24          </a>
25          <ul class="dropdown-menu dropdown-menu-end">
26              <?php
27              if ($_SESSION['userLogin'] != "") {
28                  ?>
29                  <li><em><?= $_SESSION['userLogin'] ?></em></li>
30                  <li>
31                      <hr class="dropdown-divider">
32                  </li>
33                  <li><a class="nav-link link-primary" href="index.php?page=profil">
34                      Profil</a></li>
35                  <li>
36                      <hr class="dropdown-divider">
37                  </li>
38                  <li>
39                      <form action="" method="post">
40                          <input class="btn btn-primary" type="submit" name="
disconnect" value="Déconnexion">
41                      </form>
42                  </li>
43              } else {
44                  ?>
45                  <li><a class="nav-link link-primary" href="index.php?page=
loginUser">Connexion</a></li>
46                  <li><a class="nav-link link-primary" href="index.php?page=
newUserAccount">Créer un compte</a></li>
47                  <?php
48                  }
49                  ?>
50
51          </ul>
52      </div>
53  </div>
54 </nav>
55 </header>
56 <main class="m-3">
57     <div class="container">
58         <?php
59         if ($_SESSION['userLogin'] != "") {
60             ?>
61             <div class="position-relative">
62                 <div>Nom utilisateur: <em><?= $_SESSION['userLogin'] ?></em></div>
63                 <button class="btn btn-primary position-absolute top-0 end-0">
64                     <a class="nav-link" href="index.php?page=newTravel">Nouveau voyage</a>
65                 </button>
66             </div>
67             <?php
68             }
69             ?>
70         <div class="row">

```

```

71 <table class="table table-hover mt-5">
72     <thead>
73         <tr>
74             <th scope="col"></th>
75             <th scope="col">Début</th>
76             <th scope="col">Fin</th>
77             <th scope="col">Durée</th>
78             <th scope="col"></th>
79
80         </tr>
81     </thead>
82     <?php
83     // Récupération de tous les voyages
84     $allTravel = getTravelByUserId($_SESSION['userId']);
85     foreach ($allTravel as $travelKey => $travelValue) {
86     ?>
87         <tbody>
88             <th scope="col"><?= $travelValue->title ?></th>
89             <?php
90                 // Récupération des étapes du voyage
91                 $travelStage = getStageByTravelId($travelValue->id);
92                 // réinitialise les variables a chaque nouveau voyage
93                 $travelDuration = 0;
94                 $travelStartDate = new DateTime("0000-00-00");
95                 $travelEndDate = new DateTime("0000-00-00");
96                 foreach ($travelStage as $stageKey => $stageValue) {
97                     // Récupération des activités et des médias des étapes
98                     $travelDuration += $travelDuration ->
durationDays;
99
100                     if ($travelStartDate == new DateTime("0000-00-00")) {
101                         $travelStartDate = $stageValue->date;
102                         $travelStartDate = new DateTime($travelStartDate);
103                     }
104                 }
105
106                 $travelDurationInterval = new DateInterval("P" . $travelDuration .
"D");
107                 $startDateFormat = $travelStartDate->format("d/m/Y");
108                 $travelEndDate = $travelStartDate->add($travelDurationInterval);
109                 $endDateFormat = $travelEndDate->format("d/m/Y");
110
111                 if ($travelStartDate == new DateTime("0000-00-00")) {
112                     $startDateFormat = "00/00/0000";
113                 }
114
115                 if ($travelEndDate == new DateTime("0000-00-00")) {
116                     $endDateFormat = "00/00/0000";
117                 }
118             ?>
119             <td><?= $startDateFormat ?></td>
120             <td><?= $endDateFormat ?></td>
121             <td><?= $travelDuration ?> jours</td>
122             <td><a href="index.php?page=detailTravel&id=<?= $travelValue->id ?
>">détail-></a></td>
123             </tbody>
124             <?php
125         }
126     ?>
127     </table>
128 </div>

```

```
129     </div>
130 </main>
131
132 </html>
```

VoyageMemo/app/index.php

```
1 <!--
2     TITRE : VoyageMémo
3     Auteur : Guilain Fernandez
4     Description : Page index.php du projet
5     Date : 24/04/2024
6 -->
7 <!DOCTYPE html>
8 <html lang="fr-ch">
9
10 <head>
11     <meta charset="UTF-8">
12     <meta name="viewport" content="width=device-width, initial-scale=1.0">
13     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-rbsA2VBKQhgGwzxH7pPCaAq046MgnOM80zW1RWuH61DGLwZJEdK2Kadq2F9CUG65" crossorigin="anonymous">
14     <link href="./public/css/all.css" rel="stylesheet" />
15     <title>VoyageMémo</title>
16 </head>
17
18 <body>
19     <?php require_once './controller/controller.php'; ?>
20 </body>
21 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-kenU1KFdBi4zVF0s0G1M5b4hcpxyD9F7jL+jjXkk+Q2h455rYXK/7HAuoJl+0I4" crossorigin="anonymous"></script>
22
23 </html>
```

VoyageMemo/app/view/detailTravel.php

```
1 <!--
2     TITRE : VoyageMémo
3     Auteur : Guilain Fernandez
4     Description : la page des détails des voyages
5     Date : 24/04/2024
6 -->
7
8 <header>
9     <nav class="navbar navbar-expand-lg">
10         <div class="container">
11             <a class="navbar-brand" href="index.php?page=home">
12                 
13             </a>
14             <ul class="navbar-nav">
15                 <li><a class="nav-link fs-3" href="index.php?page=home">Accueil</a></li>
16             </ul>
17             <div class="dropdown">
18                 <a class="nav-link" href="#" role="button" data-bs-toggle="dropdown" aria-expanded="false">
19                     
20                 </a>
```

```

21         <ul class="dropdown-menu dropdown-menu-end">
22             <?php
23             if ($_SESSION['userLogin'] != "") {
24             ?>
25                 <li><em><?= $_SESSION['userLogin'] ?></em></li>
26                 <li>
27                     <hr class="dropdown-divider">
28                 </li>
29                 <li><a class="nav-link link-primary" href="index.php?page=profil">
30                     Profil</a></li>
31                     <li>
32                         <hr class="dropdown-divider">
33                     </li>
34                     <li>
35                         <form action="" method="post">
36                             <input class="btn btn-primary" type="submit" name="
disconnect" value="Déconnexion">
37                         </form>
38                     </li>
39             } else {
40             ?>
41                 <li><a class="nav-link link-primary" href="index.php?page=
loginUser">Connexion</a></li>
42                 <li><a class="nav-link link-primary" href="index.php?page=
newUserAccount">Créer un compte</a></li>
43                     <?php
44                     }
45             ?>
46
47             </ul>
48         </div>
49     </div>
50 </header>
51 <main class="mt-5 d-flex justify-content-center">
52     <div class="container m-0 p-0">
53         <?php
54             // Récupération du voyage sélectionné
55             $travel = getTravelById($_GET["id"]);
56             if ($_SESSION['userId'] == $travel[0]->userId) {
57             ?>
58                 <div class="position-relative d-flex justify-content-evenly">
59                     <a class="btn btn-primary" href="index.php?page=newStage&id=<?= $travel[0]
->id ?>">Nouvelle étape</a>
60                     <a class="btn btn-primary" href="index.php?page=modifyTravel&id=<?
$travel[0]->id ?>">Modifier voyage</a>
61                     <form action="" method="post">
62                         <input type="hidden" name="travelId" value="<?= $_GET["id"] ?>">
63                         <button type="submit" name="deleteTravel" class="btn btn-primary">
64                         Supprimer voyage</button>
65                         </form>
66                 </div>
67             <?php
68             }
69             ?>
70             <h1 class="text-center"><?= $travel[0]->title ?></h1>
71             <?php
72                 // Récupération des étapes du voyage
73                 $travelStage = getStageByTravelId($travel[0]->id);
74
75                 $travelStartDate = new DateTime($travelStage[0]->date);

```

```

76     $travelEndDate = new DateTime("0000-00-00");
77     foreach ($travelStage as $key => $value) {
78         $travelDuration += $value->durationDays;
79     }
80
81     $travelDurationInterval = new DateInterval("P" . $travelDuration . "D");
82     $startDateFormat = $travelStartDate->format("d/m/Y");
83     $travelEndDate = $travelStartDate->add($travelDurationInterval);
84     $endDateFormat = $travelEndDate->format("d/m/Y");
85     ?>
86     <div class="text-center"><?= $startDateFormat . " - " . $endDateFormat ?></div>
87     <?php
88     foreach ($travelStage as $stageKey => $stageValue) {
89         // Récupération des activités et des médias des étapes
90         $stageActivity = getActivityByStageId($stageValue->id);
91         $stageMedia = getMediaByStageId($stageValue->id);
92
93         if ($stageValue->durationDays == "") {
94             $stageDuration = "1";
95         } else {
96             $stageDuration = $stageValue->durationDays;
97         }
98         $date = new DateTime($stageValue->date);
99         $date = $date->format("d/m/Y");
100
101     ?>
102     <div class="card mt-5">
103         <div class="card-body">
104             <h3 class="card-title text-center"><?= $stageValue->title ?></h3>
105             <p class="card-text text-center m-0"><em><?= $date ?></em></p>
106             <p class="card-text text-center"><em>(<?= $stageDuration ?> jours)</em></p>
107             <p class="card-text text-center"><?= $stageValue->description ?></p>
108             <h5 class="card-text fw-bold text-center">Activités</h5>
109             <div class="d-flex flex-row justify-content-evenly">
110                 <?php
111                 foreach ($stageActivity as $key => $stageActivityValue) {
112                     ?>
113                     <p class="card-text">
114                         <?= $stageActivityValue->name ?>
115                     </p>
116                     <?php
117                     ?
118                     ?>
119                 </div>
120                 <h5 class="card-text fw-bold text-center">Images</h5>
121                 <?php
122                 // Vérifie si un formulaire a été soumis
123                 if (isset($_POST['swap'])) {
124                     // Récupère l'ID de l'image cliquée à partir des données du
125                     formulaire
126                     $clickedImageID = $_POST['swap'];
127
128                     // Vérifie l'ID de l'image cliquée pour déterminer si elle doit
129                     être affichée en grand ou en petit
130                     foreach ($stageMedia as $keyMedia => $valueMedia) {
131                         if ($valueMedia->id == $clickedImageID) {
132                             // Si l'ID de l'image correspond à celle cliquée, inverse
le statut d'affichage
133                             $valueMedia->isBig = !$valueMedia->isBig;
134                         }

```

```

133
134 }
135
136 // Affiche la liste des images avec leurs boutons pour afficher en
137 grand
138 ?>
139 <div class="row">
140     <?php foreach ($stageMedia as $keyMedia => $valueMedia) { ?>
141         <div class="col-md-3 position-relative" style="text-align: center;"><!-- Ajout de la
142 classe position-relative pour positionner l'image cliquée -->
143             <form action="" method="post">
144                 <input type="hidden" name="swap" value="<?= $valueMedia->id ?>" />
145                 <button class="btn" type="submit">
146                     
147                 </button>
148             </form>
149             <?php if ($valueMedia->isBig) { ?>
150                 <div class="position-absolute top-0 start-0 w-100 h-100" style="z-index: 10;"><!-- Utilisation de z-index pour afficher l'image en grand au-
151                     dessus des autres -->
152                     
153             </div>
154         </?php ?>
155     </div>
156     <?php
157         if ($_SESSION['userId'] == $travel[0]->userId) {
158             <div class="mt-3 text-end d-flex justify-content-end">
159                 <a class="btn btn-primary me-2" href="index.php?page=modifyStage&id=<?= $travelStage[0]->id ?>">Modifier</a>
160                 <form action="" method="post">
161                     <input type="hidden" name="stageId" value="<?= $stageValue->id ?>" />
162                     <button type="submit" name="deleteStage" class="btn btn-primary">Supprimer</button>
163                 </form>
164             </div>
165         <?php
166         }
167     </?php ?>
168     </div>
169 </div>
170 <?php
171     }
172 ?>
173 </div>
174 </main>
```