**Documentation technique**

1. Introduction

L’annuaire de beauté est un outil très utile pour les internautes à la recherche des prestataires de services de beauté.

1. Technologies utilisées

* MySQL comme base de données de l’annuaire
* Twig est le générateur de Template
* Apache est le serveur web qui a été utilisé en local pour implémenter ce
* Symfony 5.4 est le Framework qui a été utilisé pour ce projet
* Webpack de Symfony pour gérer le CSS avec le préprocesseur SASS

1. Une image contenant texte

   Description générée automatiquementUne image contenant texte

   Description générée automatiquementArchitecture : Base de données

Model conceptuel :

Une image contenant table

Description générée automatiquementUne image contenant texte

Description générée automatiquement

Une image contenant table

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

Une image contenant table

Description générée automatiquementUne image contenant table

Description générée automatiquement

Une image contenant table

Description générée automatiquementUne image contenant table

Description générée automatiquementUne image contenant texte

Description générée automatiquementUne image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

Une image contenant table

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

Une image contenant table

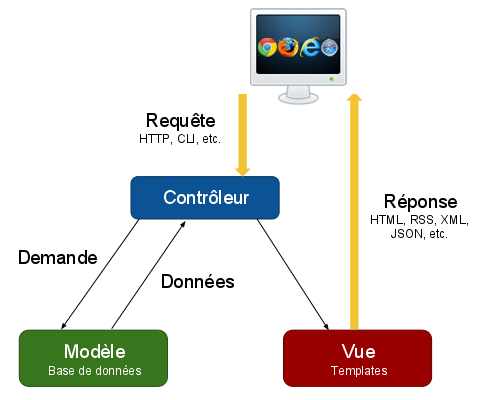
Description générée automatiquement

Une image contenant table

Description générée automatiquementUne image contenant table

Description générée automatiquement

1. Architecture Modèle : MVC



/src

/src et bd

/Templates

* **Le Modèle**

Ici nous avons la base de données via MySQL qui fournit des réponses aux requêtes reçues des contrôleurs

Le fichier .**env** sert à lier l’application à la base de données et est configuré au préalable .

* **Les Vues**

Twig est la moteur de Template utilisé pour l’affichage et le langage Twig est celui qui traite les données reçues des contrôleurs. Dans l’arborescence , le dossier Template est celui qui contient tous les Templates

* **Les Contrôleurs**

Avec l’orienté objet de PHP , La gestion , la redirection ainsi que le traitement des demandes est géré par différents fichiers se trouvant dans différents dossiers  ainsi :

* Les contrôleurs se trouvent dans le dossier **Controller** inclut dans le dossier **/SRC**
* Les requêtes DQL se trouvent dans la dossier **/Repository**
* Le dossier **Security** sert a gérer la connexion et déconnexion des utilisateurs
* Le dossier **Classe** contient les classes crées qui regorgent des fonctions amener à gérer le traitement des images , l’intégration des communes , code postaux et provinces ainsi la gestion partielle des mails .

1. Fonctionnalités

* **Connexion et déconnexion**

Afin de gérer la connexion et déconnexion , ces entités sont concernées :

* Le fichier SecurityController.php du dossier **/src/Controller**
* Le fichier login.html.twig du dossier **templates/security**
* Le dossier form se trouvant dans le dossier **/src**
* Le fichier **security.yaml** ayant pour chemin **/config/packages/** qui permet de faire des configurations spécifiques
* **Gestion des utilisateurs** ( internautes et prestataires et administrateurs)

Dans la gestion , nous avons :

* La préinscription
* L’inscription
* L’affichage
* L’ Ajout
* La Modification

Les entités qui s’occupent de la gestion sont les suivants :

* Le dossier /**Controller** se trouvant dans **/src**
* Le dossier /**Form** se trouvant dans le dossier **/src**
* Le dossier /**Classes** se trouvant dans le dossier **/src**
* Le dossier prestataire, administrateur, inscription se trouvant dans le dossier /**templates**
* Le dossier **/Repository** pour les requêtes dql se trouvant dans le dossier **/src**
* **La recherche de prestataires**

C’est au travers des dossiers suivant que sont gérés les recherches :

* Le dossier **/controller** se trouvant dans **/src**
* Le dossier **/Form** se trouvant dans le dossier **/src**
* Le dossier **/prestataire** se trouvant dans le dossier **/templates**
* Le dossier **/Repository** pour les requêtes dql se trouvant dans le dossier /src
* **Le mailing**
* Le dossier **/src** et le dossier **/Classes** contiennent des fichiers et des routes qui gèrent l’envoi des emails
* **L’affichage de la carte**
* Les fichiers de gestion se trouvent dans le dossier **/js** inclut dans dossier **/public** .
* Le dossier **/prestataire**, se trouvant dans le dossier **/templates** permet l’affichage .
* **CSS et javascript**

Les fichiers javascript et ceux pour le style sont gérées par les entités suivantes :

* **CSS**

Le dossier **/assets** gèrent les fichiers du préprocesseur SASS et contient tous les fichiers de style .

* **Javascript**
* Les fichiers de gestion se trouvent dans le dossier **/js** inclut dans dossier **/public** .
* Le fichier **HomeController.php** inclut dans le dossier **/src/Controller** contient des routes qui gèrent les requêtes Ajax .

1. Configurations

* **La base de données**

Dans le fichier .env , décommenter la ligne afin de choisir votre base de données et définissez y le nom de votre base de données

* **Le Mailing**
* Dans le fichier **.env** configurer le smtp du service web choisit
* Dans le fichier **messenger.yawl** inclut dans **/config/packages** , décommenter cette ligne :
* *Symfony\Component\Mailer\Messenger\SendEmailMessage: async*