



Projet Professionnel

IMT_ATLANTIQUE



AFPA | DWWM

SIDER | NOV 2019 | JUIL 2020

Table des matières

L'entreprise d'accueil	4
Cahier des charges	5
Présentation générale du projet	5
Les contraintes des LABS	5
Les contraintes des RESSOURCES	6
Authentification	6
Mise ne place d'un panel d'administration avec LARAVEL et VOYAGER pour gérer les profils d'utilisateurs	7
Ergonomie, fluidité	7
Respect de la charte IMT Atlantique	7
Sauvegarde	7
SOFTWARES utilisés	8
TAIGA (https://taiga.io/)	8
RENDEZ-VOUS (https://rendez-vous.renater.fr/home/)	8
Présentation et choix de FRAMEWORK et IDE	9
LARAVEL (https://laravel.com/)	9
VOYAGER (https://voyager.devdojo.com/)	9
VISUAL STUDIO CODE (https://code.visualstudio.com/)	9
BALSAMIQ (https://upload.wikimedia.org/wikipedia/commons/4/47/Profilewireframe.png)	9
Développement de l'application	10
Maquetter une application	10
Installation de l'environnement de travail	11
Développement de la partie FRONT-END	12
Créations de nos vues	14
Test	15
Création d'un système d'authentification	15
Respect de la charte de l'entreprise sur nos différentes vues	16
Installation de voyager	17
Développement de la partie BACKEND en MVC et POO	18
Création de la base de données	18
Création de nos contrôleurs	21
Créations des autres contrôleurs	24
Création de nos routes	25
Tests de notre application	25

Sauvegarde.....	26
Fonctionnalité hors cahier des charges.....	27
Recherche et veille.....	29
Synthèse du projet.....	29
Remerciements	29
Annexe 1 : Tests formulaire.....	30
Annexe 2 : Réinitialisation du mot de passe.....	31
Annexe 3 : Se déconnecter	31

L'entreprise d'accueil



Situé en périphérie de la région de Brest, **IMT Atlantique** est une grande école d'ingénieurs généralistes et un centre de recherche reconnus internationalement.

Spécialisé dans l'enseignement, l'innovation et la recherche dans les domaines de l'Énergie, le numérique et l'environnement

IMT Atlantique, nouvelle née parmi les grandes écoles d'ingénieurs généralistes, conjugue numérique, énergie et environnement pour mieux répondre aux besoins d'un monde de plus en plus interconnecté, et vous préparera à mettre le cap sur un avenir prometteur.

IMT Atlantique offre un programme d'enseignement de haut niveau et un panel de possibilités : l'élève ingénieur, quel que soit son profil initial, ses projets ou ses attentes, est accompagné vers un large choix de métiers et de parcours professionnels en France comme à l'international.

<https://www.imt-atlantique.fr/fr/l-ecole/campus/campus-de-brest>

J'ai été accueilli par Emanuel Braux : ingénieur système, responsable fonctionnel et technique Cloud Privé et infrastructure Web, formateur Cloud et Openstack.

Il a été mon référent pendant toute la durée du stage.

Cahier des charges

Présentation générale du projet

Les « EC » (enseignants-chercheurs) d'IMT Atlantique ont besoin d'accéder à des ressources dans le cadre de leurs activités :

- Travaux pratiques (environnement de travail, infrastructure, outils, ...)
- Projets (serveur dédié, service spécifique, ...)
- Calculs et simulation.

Ils font régulièrement appel aux services de la « DISI » (Direction Informatique) qui met à leur disposition les ressources dont ils ont besoin, en s'appuyant sur un ensemble de technologies et d'outils (Openstack, Jupyterhub, ...).

Les demandes se font actuellement par mail, ou via l'outil de support de la DISI.

De nombreux échanges sont nécessaires entre les EC et la DISI, pour obtenir l'ensemble des informations permettant la création de ces ressources, ce qui entraîne une perte de temps de part et d'autre. Bien souvent l'historique de ces échanges est perdu. Il est donc difficile de planifier l'utilisation des ressources, et il est courant que des ressources ne soient libérées que bien après qu'elles ne soient plus utilisées.

L'objectif de ce projet est de proposer une application accessible en ligne, permettant de gérer les demandes de ressources.

L'application doit permettre aux EC de déposer des demandes de création de LAB, de suivre l'évolution de leur demande, et d'accéder aux données utiles une fois un LAB déployé.

Elle doit permettre à la DISI de suivre les demandes en cours, de les valider, puis de gérer le cycle de vie des LAB. Le tout en interaction avec les EC.

L'application n'aura pas pour objectif le déploiement des ressources, mais uniquement de gérer les demandes, et de permettre aux EC et à la DISI de gérer le cycle de vie des ressources.

Les termes suivants seront utilisés dans la suite de ce document :

- « DISI » : la Direction Informatique
- « EC » : les enseignants-chercheurs d'IMT Atlantique seront.
- « LAB » représente tout type de ressource à déployer. Il correspond à une activité demandée par un EC et ressources
- « RESSOURCE » : tout type de ressource à déployer.

Les contraintes des LABS

- ❖ Un LAB doit donc comporter une date de début et une date de fin
- ❖ Il est identifié par un nom unique qui est obligatoire.
Le nom du LAB sera celui de la demande de création de LAB. L'unicité et le format du nom doivent être validés au moment de la création de la demande. La validation du nom doit être la première étape de la création de la demande pour éviter qu'un autre utilisateur ne puisse créer un autre LAB en même temps avec le même nom.
La longueur du nom devra être comprise entre 3 et 20 caractères. Il ne devra pas contenir de caractères spéciaux ou d'espaces : uniquement des lettres minuscules / majuscule sans accents, des chiffres, et les caractères '-'

- ❖ Une demande de LAB peut être créée à partir de plusieurs endroits dans l'application. En fonction de l'endroit à partir duquel la demande de création d'un LAB est faite des informations devront être préremplies.
- ❖ Depuis la page d'accueil, exemple de demandes de LABS
Un EC souhaite donner un cours de développement WEB. Il sait qu'il va donner cet enseignement sur novembre/décembre. Il ne sait pas encore quand les sessions de TP seront programmées. Il ne sait pas encore le nombre d'élèves qui seront inscrits. Cependant, il est sûr que ce TP sera donné. Il souhaite un environnement Apache/PHP/Mysql. Il va donc créer une demande de LAB, en indiquant comme nom pour le LAB le nom de son module d'enseignement.
Il va indiquer la période du premier novembre au 31 décembre en précisant que la date n'est pas définitive.
Il va indiquer que c'est un LAB en mode « Multi-LAB », et si possible donner une estimation du nombre de LABs enfants souhaité.
Il va rechercher dans le catalogue une RESSOURCE de référence correspondant à PHP/Mysql, et consulter ses caractéristiques
- ❖ Modifier une demande de LAB
Une fois une demande créée, les gestionnaires doivent pouvoir y accéder pour la modifier.
- ❖ Consulter la liste des LAB pour les « gestionnaires »
Les gestionnaires doivent pouvoir afficher la liste de leurs LAB, actifs ou passés, et accéder aux informations y étant associées.

Les contraintes des RESSOURCES

Elle est identifiée par un nom unique : chaque RESSOURCE devra être identifiée précisément. Il est donc nécessaire que chaque ressource soit identifiée par un nom unique.

La longueur du nom devra être comprise entre 3 et 30 caractères.

Il ne devra pas contenir de caractères spéciaux ou d'espaces : uniquement des lettres minuscules/majuscule sans accents, des chiffres, et les caractères '-' et '_'.

Elle est rattachée à un LAB : il est donc nécessaire de définir une liste de RESSOURCES disponibles, sous forme d'un catalogue

Consulter le catalogue des RESSOURCES de référence Les gestionnaires doivent pouvoir afficher la liste des RESSOURCES de référence, afficher les caractéristiques, filtrer les RESSOURCES en fonction de mots clé et critères.

Authentification

Les accès à l'application seront soumis à authentification. Lorsqu'un utilisateur accédera à l'application sans être authentifié, il sera redirigé vers une page d'information. Cette page indiquera les conditions d'accès à l'application, un moyen de contact avec les administrateurs de l'application ainsi qu'un espace permettant de gérer l'authentification comprenant l'authentification par login / mot de passe et la possibilité réinitialiser le mot de passe.

Une fois authentifié, l'utilisateur devra pouvoir se déconnecter, se reconnecter et changer son mot de passe
Une fois authentifié, l'utilisateur aura accès à différentes informations en fonction de son profil

Donner accès aux Informations des utilisateurs pour les « administrateurs ». Ils doivent pouvoir accéder aux informations personnalisées de chaque utilisateur. L'indication de cette possibilité d'accès doit figurer dans l'interface affichant les données personnalisées.

Mise ne place d'un panel d'administration avec LARAVEL et VOYAGER pour gérer les profils d'utilisateurs

❖ Les différents profils d'utilisateurs

Les EC doivent pouvoir effectuer des demandes de LAB, les modifier, et les valider. Ils doivent pouvoir avoir accès à leurs demandes en cours, leur demandes passées, et les LABs actifs les concernant. Ils doivent également pouvoir avoir accès aux informations fournies aux utilisateurs des LABs. Le profil correspondant sera « gestionnaire ». Il sera principalement attribué au EC

Les élèves et les EC doivent pouvoir accéder aux informations utiles pour pouvoir travailler avec les LABs auxquels ils peuvent avoir accès. Par exemple les identifiants nécessaires pour pouvoir accéder aux RESSOURCES. Le profil correspondant sera « utilisateur ». Il sera attribué par défaut à toute personne accédant à l'application (un élève, un EC, un membre de la DISI, ...)

❖ La DISI sera par défaut administrateur

Ergonomie, fluidité

La navigation devra être simple, cohérente et intuitive, aucun élément superflu ne devra apparaître, pas plus de trois polices différentes seront utilisées.

L'application devra être si possible accessible depuis n'importe quel terminal, notamment mobiles et tablettes.

Respect de la charte IMT Atlantique

Les standards de la chartre graphiques indiqués par la direction de la communication d'IMT Atlantique doivent être respectés.

<https://www.imt-atlantique.fr/fr/l-ecole/logotypes-imt-atlantique>

Sauvegarde

On sécurise notre application par un dépôt sur git hub :

<https://github.com/>

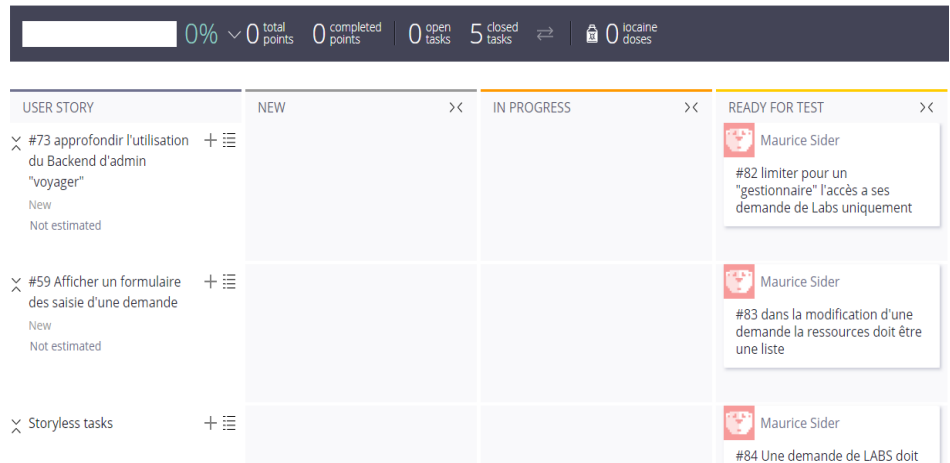
SOFTWARES utilisés

TAIGA (<https://taiga.io/>)

Il s'agit d'un outil de gestion de projet en méthodologie agile, que j'ai utilisé pendant la durée du stage. J'utilisais ce support quotidiennement pour suivre, et mettre à jour mes tâches.

Ce support m'a aussi permis de découvrir un nouveau langage de balisage MARKDOWN.

LIVRAISON D'UN V1 DE L'APPLICATION CLOUDLABS-APPLABS-MS 25 MAY 2020-29 MAY 2020



RENDEZ-VOUS (<https://rendez-vous.renater.fr/home/>)

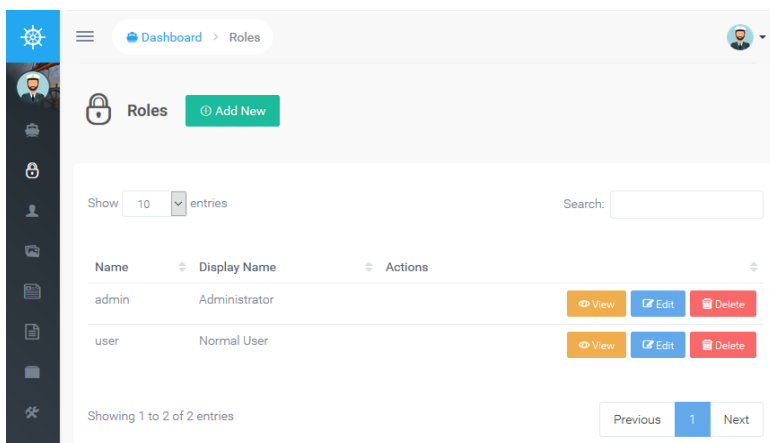
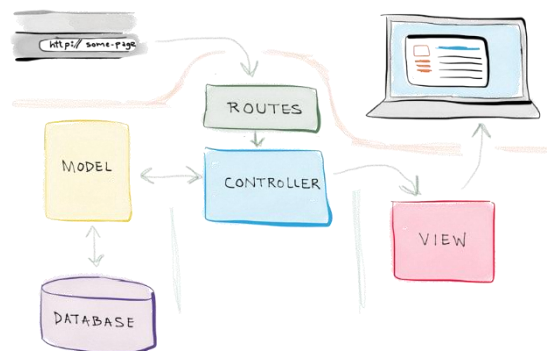
C'est un support de Visioconférence que j'ai utilisé comme contact visuel avec mon référent de stage.



Présentation et choix de FRAMEWORK et IDE

LARAVEL (<https://laravel.com/>)

C'est un Framework php en MVC et développer en POO que j'ai utilisé pour le projet



VOYAGER

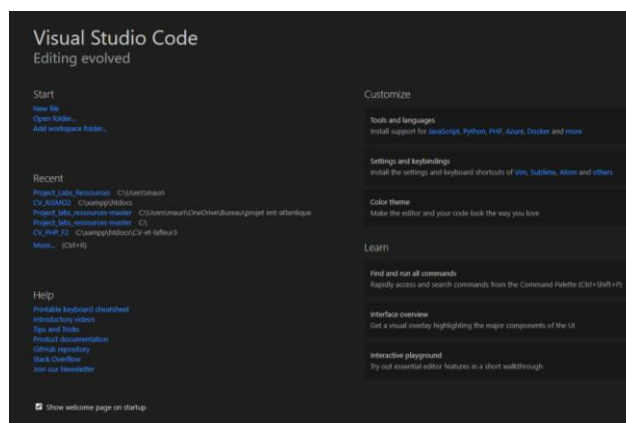
(<https://voyager.devdojo.com/>)

C'est un outil Backend pour créer le système d'administration que j'ai utilisé pour le projet

VISUAL STUDIO CODE

(<https://code.visualstudio.com/>)

Est un éditeur de texte que j'ai utilisé pour développer le projet



BALSAMIQ

(<https://upload.wikimedia.org/wikipedia/commons/4/47/Profilewireframe.png>)

C'est un outil qui va me permettre de maquetter mon projet



Développement de l'application

Maquetter une application

Réalisation de la maquette « page d'accueil », elle comprend :

- 2 menus déroulants (comprenant un logo, des boutons (admin, ressources, demande, Home, se déconnecter))
- Un logo principal au centre avec un message d'accueil au-dessus
- Deux liens vers les vues create et catalogue



Réalisation de la maquette « page demande » (formulaire de demande), elle comprend :

- Un logo en haut à gauche
- Un formulaire avec 4 champs (labs, ressources, date début et date fin)
- Un bouton envoyer

Maquette du formulaire de demande de labs. En haut à gauche, un logo. Le titre est "Demande de labs". Les champs sont : NOM DU LAB (champ de texte), RESSOURCES DISPONIBLES (menu déroulant avec "Ressources disponibles" sélectionné), DATE DE DEBUT (champ de texte), DATE DE FIN (champ de texte). En bas, un bouton "Envoyer".

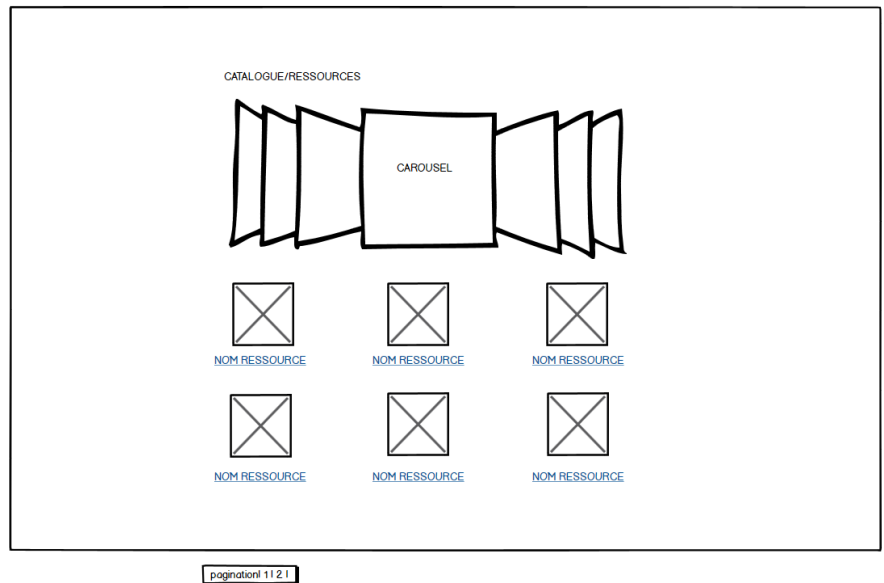
Réalisation de la maquette « Page Edit » (formulaire de modification), elle comprend :

- Un logo en haut à gauche
- Un formulaire avec 4 champs (labs, ressources, date début et date fin)
- Des boutons « valider » et « supprimer »

Maquette du formulaire de modification de demande. En haut à gauche, un logo. Le titre est "MODIFIER VOTRE DEMANDE". Les champs sont : NOM DU LAB (champ de texte), RESSOURCES DISPONIBLES (menu déroulant avec "Ressources disponibles" sélectionné), DATE DE DEBUT (champ de texte), DATE DE FIN (champ de texte). En bas, deux boutons : "VALIDER VOTRE DEMANDE" et "SUPPRIMER VOTRE DEMANDE".

Réalisation de la maquette « page catalogue » (ressources), elle comprend :

- Un carrousel
- Des images correspondant aux ressources disponibles
- Un bouton pagination



Installation de l'environnement de travail

- ❖ Utilisation de l'éditeur de texte : VISUAL STUDIO CODE
- ❖ Installation du gestionnaire de DEPENDANCE COMPOSER
 - Télécharger le COMPOSER-Setup.exe sur [getComposer.org](https://getcomposer.org) et exécuter
 - Tester l'installation en ouvrant une ligne de commande CMD et taper Composer
- ❖ Installation de LARAVEL avec COMPOSER (utilisation de l'outil en ligne de commande « artisan »)
- ❖ Création du projet LARAVEL avec COMPOSER dans la console :
Composer « create-project --prefer-dist Laravel/Laravel Project_Labs_Ressources »
- ❖ Création de la base de données « imt-labs » dans PhpMyadmin
- ❖ Configuration de notre fichier de connexion à la base de données : « .env »
- ❖ Dans la console de notre éditeur on vérifie l'installation avec la commande suivante

```
DEBUG CONSOLE  PROBLEMS 4 OUTPUT  TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Testez le nouveau système multiplateforme PowerShell https://aka.ms/pscore6

PS F:\UwAmp\www\Project_Labs_Ressources> php artisan serve
```

Laravel

DOCS LARACASTS NEWS BLOG NOVA FORGE VAPOR GITHUB

La page ci-dessus s'affiche, l'installation s'est bien passée. Nous aurons donc une architecture trois tiers (client, serveur, base de données).

Développement de la partie FRONT-END

- ❖ Réalisation de notre page d'accueil en conformité avec notre maquette
- ❖ Import de BOOTSTRAP (Framework CSS comportant des librairies) et de JQUERY (Framework JAVASCRIPT qui dynamisera notre application)
- ❖ Importation des librairies :
 - Bootstrap (téléchargement du package sur Bootstrap.com)

Nom	Taille	Compressé	Type	Modifié	CRC32
..			Dossier de fichiers		
js	1 676 633	387 147	Dossier de fichiers	12/05/2020 18:52	
css	2 025 237	337 792	Dossier de fichiers	12/05/2020 18:52	

On copie colle le fichier bootstrap.min.css dans le dossier public\css

Nom	Taille	Compressé	Type	Modifié	CRC32
..			Dossier de fichiers		
bootstrap-reboo...	32 286	8 074	Fichier MAP	12/05/2020 18:52	3324CC37
bootstrap-reboo...	3 879	1 568	JetBrains PhpStorm	12/05/2020 18:52	4A097127
bootstrap-reboo...	77 337	17 191	Fichier MAP	12/05/2020 18:52	504FA19E
bootstrap-reboo...	4 740	1 680	JetBrains PhpStorm	12/05/2020 18:52	CFD838A9
bootstrap-grid...	115 803	14 550	Fichier MAP	12/05/2020 18:52	2B568585
bootstrap-grid...	50 995	6 192	JetBrains PhpStorm	12/05/2020 18:52	A574B80D
bootstrap-grid.c...	158 858	27 819	Fichier MAP	12/05/2020 18:52	10D78F95
bootstrap-grid.css	67 959	7 065	JetBrains PhpStorm	12/05/2020 18:52	0344097C
bootstrap.min.c...	646 432	103 637	Fichier MAP	12/05/2020 18:52	DD57D063
bootstrap.min.css	160 403	23 668	JetBrains PhpStorm	12/05/2020 18:52	146CDF3F
bootstrap.css.m...	508 232	100 530	Fichier MAP	12/05/2020 18:52	0D8A3F6B
bootstrap.css	198 313	25 818	JetBrains PhpStorm	12/05/2020 18:52	E4CEF881

On copie colle le fichier Bootstrap.min.js dans le dossier js\bootstrap.min.js

Nom	Taille	Compressé	Type	Modifié	CRC32
..			Dossier de fichiers		
bootstrap.min.js...	191 486	46 561	Fichier MAP	12/05/2020 18:52	289A3490
bootstrap.min.js	60 174	14 770	JetBrains PhpStorm	12/05/2020 18:52	731CFCE2
bootstrap.js.map	253 935	57 553	Fichier MAP	12/05/2020 18:52	481E8C40
bootstrap.js	136 323	25 300	JetBrains PhpStorm	12/05/2020 18:52	840C724A
bootstrap.bundl...	315 300	82 144	Fichier MAP	12/05/2020 18:52	C7D94C2A
bootstrap.bundl...	81 084	21 608	JetBrains PhpStorm	12/05/2020 18:52	500EB09E
bootstrap.bundl...	409 130	90 890	Fichier MAP	12/05/2020 18:52	A2B3A59F
bootstrap.bundl...	229 201	48 321	JetBrains PhpStorm	12/05/2020 18:52	0862F2F2

- JQUERY

Copie colle le code JQUERY sur jquery.com et colle dans le fichier jquery.js dans le dossier public\js

❖ Choix du Template :

- BOOSTWACH Pour l'interface utilisateur
Choix d'un thème sur Bootstrap.com
Télécharger et copier le fichier bootstrap.min.css
dans le dossier public\css



❖ Réalisation d'une interface utilisateur Web dynamique

- Utilisation des fonctionnalités du Framework LARAVEL pour dynamiser nos pages.
- Création de notre Squelette (layout), toutes nos vues en hériteront
- Respect de la charte de l'entreprise sur nos différentes vues
- Création d'un dossier « layout » dans resources/views/app.blade.php de notre projet

```

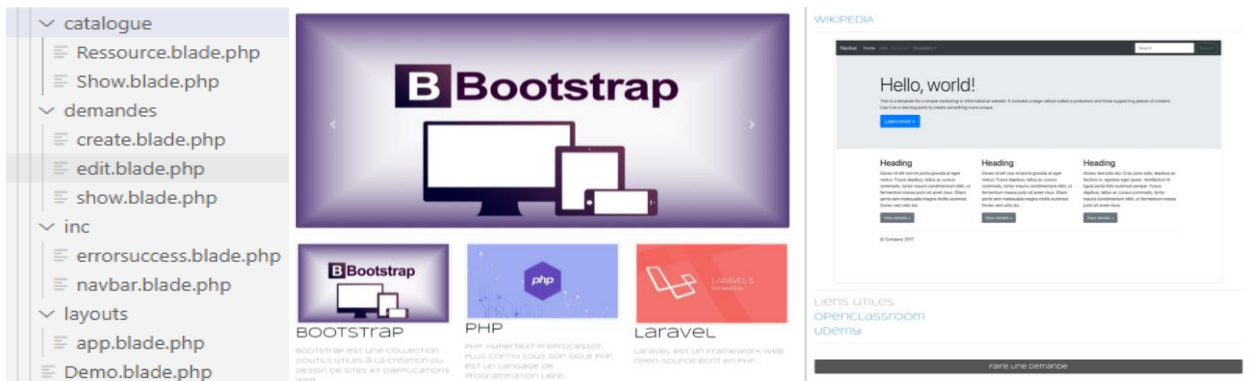
app.blade.php x
resources > views > layouts > app.blade.php > html > body > div.container
1 <!doctype html>
2 <html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
3 <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1">
6     <!-- CSRF Token -->
7     <meta name="csrf-token" content="{{ csrf_token() }}">
8     <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
9
10    <link rel="stylesheet" href="{{ asset('css/app.css') }}">
11    <link rel="stylesheet" href="{{ asset('css/css.css') }}">
12
13    @yield('stylesheets')
14    <title{{ config('app.name', 'LABS_RESSOURCES') }}</title>
15
16 </head>
17 <body>
18
19     @include('inc.navbar')
20
21     <div class="container">
22
23         @yield('slider')
24
25         @include('inc.errorsuccess')
26
27         @yield('content')<!--permet d'afficher le contenu -->
28     </div>
29
30     <script src="{{ asset('js/jquery.js') }}"></script>
31     <script src="{{ asset('js/bootstrap.min.js') }}"></script>
32     @yield('javascripts')
33 </body>
34 </html>
35
36

```

Créations de nos vues

❖ Dans le dossier catalogue

- Exemple pour la « page Ressource » affichera toutes les ressources disponibles dynamiquement
- La vue « Show » affichera des infos et des liens sur les ressources

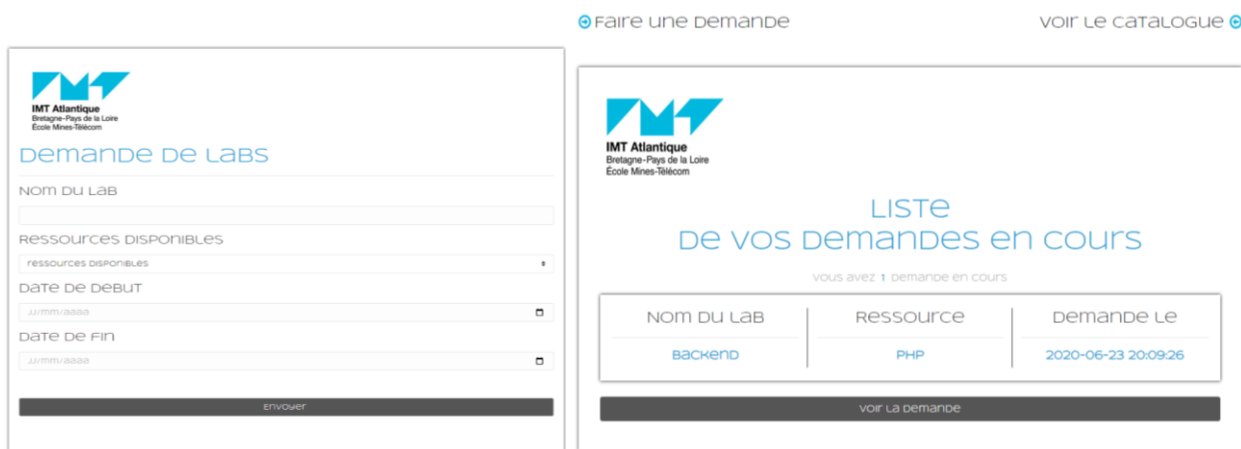


Exemple de page Resource

Exemple de vue Show

❖ Dans le dossier demande

- La vue « create » affichera un formulaire de demande
- La vue « show » affichera la demande



❖ Dans le dossier Inc

- La vue « errorsucces » affichera les messages d'erreurs ou de succès
- La vue « nav bar » affichera deux « dropdowns »



Test

J'ai testé le HTML avec le validateur W3C. Je me suis assuré de la cohérence de la balise « alt » de l'image pour le référencement SEO

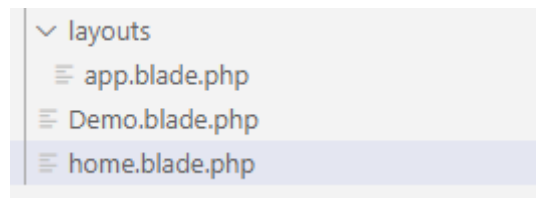
J'ai testé le responsive sur plusieurs supports (mobile, ordinateur portable, tablette).

Création d'un système d'authentification

On sécurise notre application, tout utilisateur devra s'authentifier pour y accéder. Avec un système d'authentification généré par LARAVEL :

- ❖ Dans la console
 - Composer require laravel/ui
 - Php artisan ui vue –auth

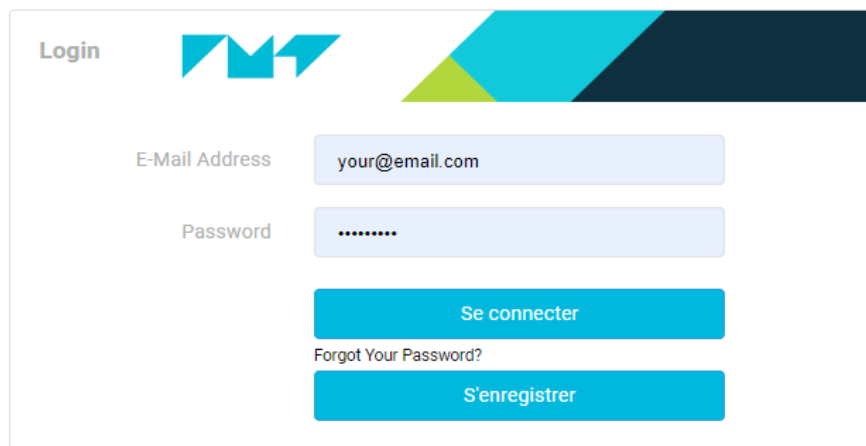
Cette commande nous génère une vue Home .blade.php



- ❖ Dans notre fichier web.php, des « routes » ont été générées :

```
////////////////////////////////AUTHENTIFICATION////////////////////////////////  
Auth::routes();
```

- ❖ On obtient cette vue « login » :

A screenshot of a web application's login page. The page has a header with the word 'Login' and a logo consisting of three blue and green geometric shapes. Below the header, there are two input fields: 'E-Mail Address' with the placeholder text 'your@email.com' and 'Password' with masked characters '.....'. Below these fields are two buttons: 'Se connecter' (blue) and 'S'enregistrer' (blue). Above the 'S'enregistrer' button is a link that says 'Forgot Your Password?'. The entire form is enclosed in a light gray border.

Respect de la charte de l'entreprise sur nos différentes vues

Mise en place d'un panel d'administration avec LARAVEL VOYAGER <https://voyager.devdojo.com/> qui nous permettra de :

- Gérer nos profils utilisateurs
- Créer nos tables et nos models

Modification de notre URL dans le fichier « .env » et installation de VOYAGER via la console :

```
.env
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:B416j0DTHZ1JJ7hOV+YfSBHo4YsbN8hR5UnAmVX+Ac4=
4 APP_DEBUG=true
5 APP_URL=http://localhost:8000
6
7 LOG_CHANNEL=stack
8
9 DB_CONNECTION=mysql
10 DB_HOST=127.0.0.1
11 DB_PORT=3306
12 DB_DATABASE=imt_labs
13 DB_USERNAME=root
14 DB_PASSWORD=root
15
16 BROADCAST_DRIVER=log
17 CACHE_DRIVER=file
18 QUEUE_CONNECTION=sync
19 SESSION_DRIVER=file
20 SESSION_LIFETIME=120
21
22 REDIS_HOST=127.0.0.1
23 REDIS_PASSWORD=null
24 REDIS_PORT=6379
25
26 MAIL_MAILER=smtp
27 MAIL_HOST=smtp.mailtrap.io
28 MAIL_PORT=2525
29 MAIL_USERNAME=null
30 MAIL_PASSWORD=null
31 MAIL_ENCRYPTION=null
32 MAIL_FROM_ADDRESS=null
33 MAIL_FROM_NAME="${APP_NAME}"
```

DEBUG CONSOLE PROBLEMS OUTPUT TERMINAL

Windows PowerShell

Copyright (C) Microsoft Corporation. Tous droits réservés.

Testez le nouveau système multiplateforme PowerShell <https://aka.ms/pscore6>

PS C:\Users\mauri\Project_Labs_Ressources> composer require tcg/voyager[]

Installation de voyager

DEBUG CONSOLE PROBLEMS OUTPUT TERMINAL

Windows PowerShell

Copyright (C) Microsoft Corporation. Tous droits réservés.

Testez le nouveau système multiplateforme PowerShell <https://aka.ms/pscore6>

```
PS C:\Users\mauri\Project_Labs_Ressources> php artisan voyager:install
```

Création d'un profil administrateur pour se connecter à notre système d'administration :

DEBUG CONSOLE PROBLEMS OUTPUT TERMINAL

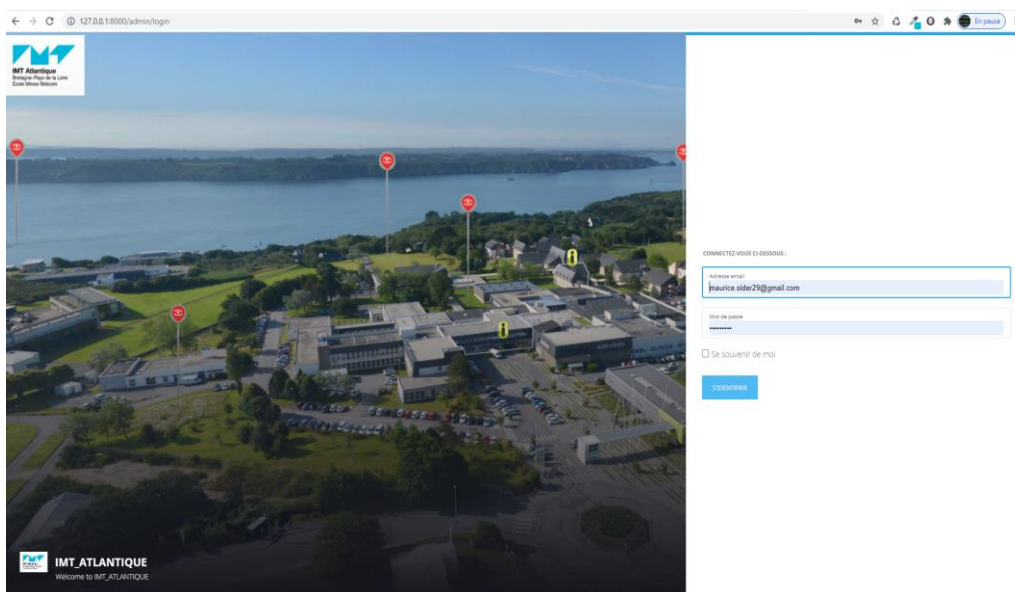
Windows PowerShell

Copyright (C) Microsoft Corporation. Tous droits réservés.

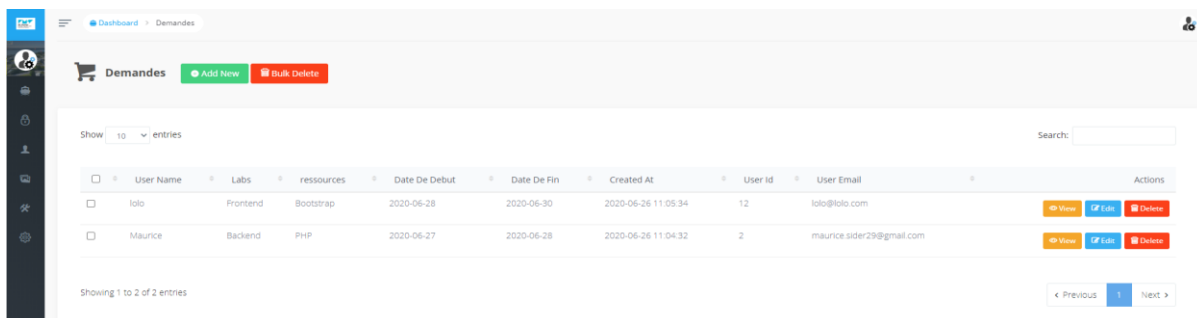
Testez le nouveau système multiplateforme PowerShell <https://aka.ms/pscore6>

```
PS C:\Users\mauri\Project_Labs_Ressources> php artisan voyager:admin maurice.sider29@gmail.com --create
```

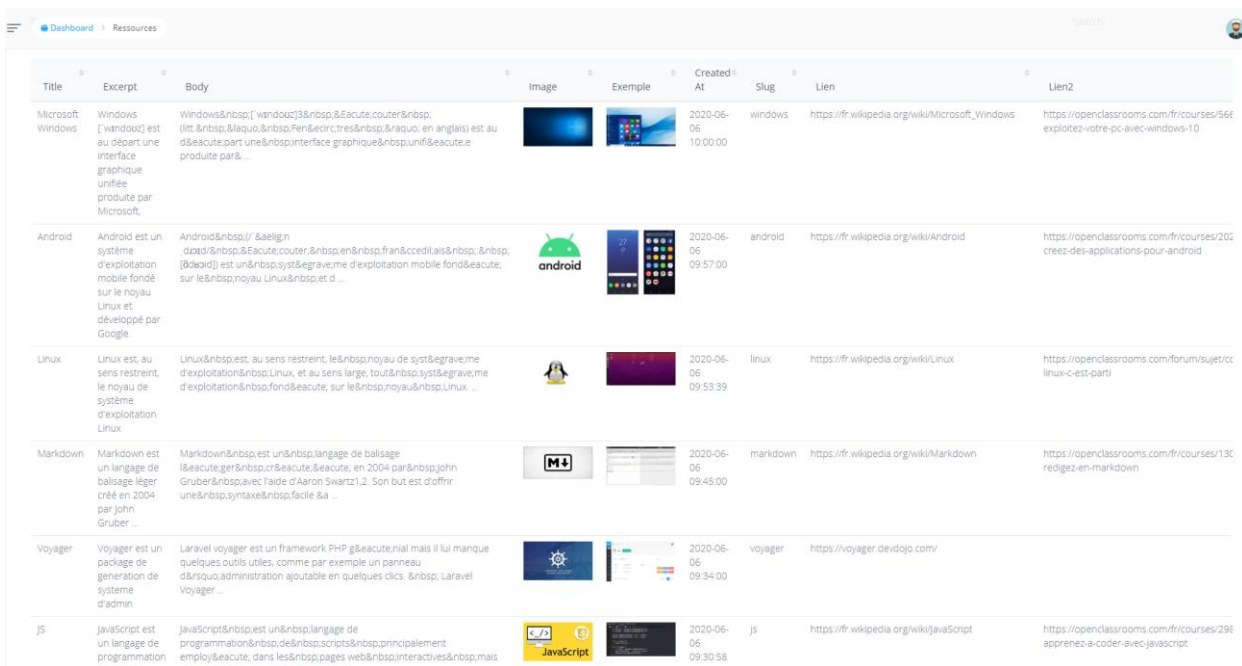
On peut maintenant se connecter suivant avec l'URL suivante « .../admin »















La DISI qui est admin par défaut a désormais accès à toutes les demandes effectués par les EC.

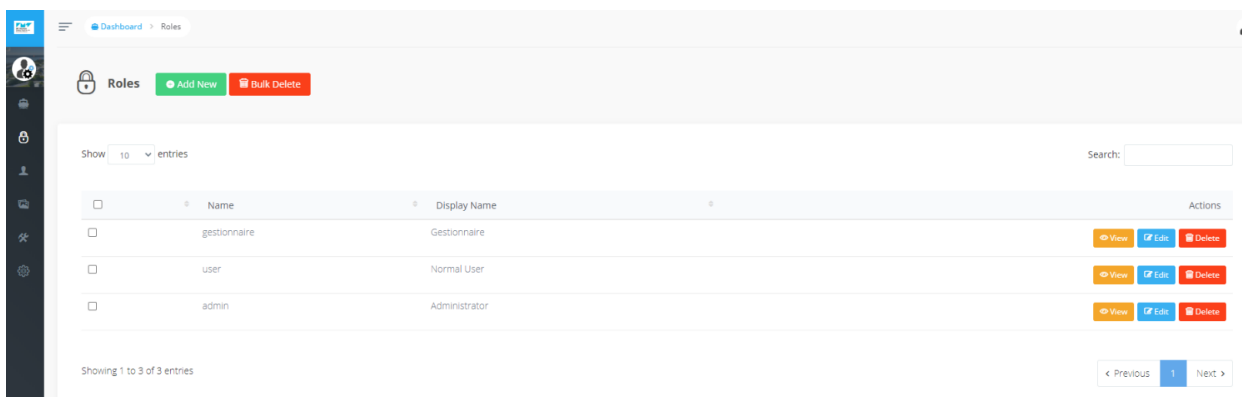


On ajoute des ressources :



Title	Excerpt	Body	Image	Exemple	Created At	Slug	Lien	Lien2
Microsoft Windows	Windows [window] est au départ une interface graphique unifiée produite par Microsoft.	Windows est un système d'exploitation pour ordinateurs personnels et professionnels. En anglais, c'est aussi le nom de la famille de produits de Microsoft.			2020-06-06 10:00:00	windows	https://fr.wikipedia.org/wiki/Microsoft_Windows	https://openclassrooms.com/fr/courses/566-exploitez-votre-pc-avec-windows-10
Android	Android est un système d'exploitation mobile fondé sur le noyau Linux et développé par Google.	Android est un système d'exploitation pour appareils mobiles. Il est basé sur le noyau Linux et développé par Google.			2020-06-06 09:57:00	android	https://fr.wikipedia.org/wiki/Android	https://openclassrooms.com/fr/courses/202-creez-des-applications-pour-android
Linux	Linux est, au sens restreint, le noyau de système d'exploitation.	Linux est, au sens restreint, le noyau de système d'exploitation. Au sens large, tout système d'exploitation basé sur le noyau Linux est dit Linux.			2020-06-06 09:53:39	linux	https://fr.wikipedia.org/wiki/Linux	https://openclassrooms.com/forum/sujet/cc-linux-c-est-parti
Markdown	Markdown est un langage de balisage léger créé en 2004 par John Gruber.	Markdown est un langage de balisage léger créé en 2004 par John Gruber. Il est utilisé pour écrire du contenu web.			2020-06-06 09:45:00	markdown	https://fr.wikipedia.org/wiki/Markdown	https://openclassrooms.com/fr/courses/130-redigez-en-markdown
Voyager	Voyager est un package de génération de système d'admin.	Laravel voyager est un framework PHP qui facilite la création de systèmes d'administration.			2020-06-06 09:34:00	voyager	https://voyager.devdojo.com/	
JS	JavaScript est un langage de programmation.	JavaScript est un langage de programmation web. Il est utilisé pour créer des applications web interactives.			2020-06-06 09:30:58	js	https://fr.wikipedia.org/wiki/JavaScript	https://openclassrooms.com/fr/courses/296-apprenez-a-coder-avec-javascript

On gère les accès à notre système :



	Name	Display Name	Actions
<input type="checkbox"/>	gestionnaire	Gestionnaire	View Edit Delete
<input type="checkbox"/>	user	Normal User	View Edit Delete
<input type="checkbox"/>	admin	Administrator	View Edit Delete

Développement de la partie BACKEND en MVC et POO

Création de la base de données

Dans phpMyAdmin (application web pour les systèmes de gestion de base de données)

Création de la BDD au nom de : imt-labs

Dans le fichier « .env » de notre projet, je configure ma connexion à la base de données en renseignant les champs :

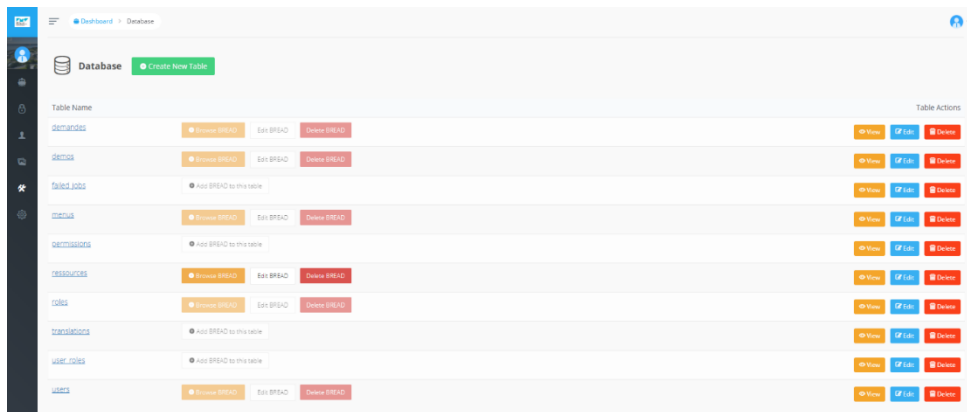
```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=imt_labs
DB_USERNAME=root
DB_PASSWORD=
```

Avec LARAVEL, on peut créer des tables d'une base de données avec son système de migration :

- Dans la console : Exemple : `php artisan make : migration create-demande_table`
- Et créer aussi des modèles : Exemple : `php artisan make : model demande`

Une table est représentée par une class qui étend de la class model.

ELOQUENT est l'ORM (Object Relationnel Mapping) de LARAVEL qui converti les données relationnelles d'une « bdd » en objet, immunise aussi contre les injections « Sql ».



Cependant j'ai utilisé le Framework VOYAGER qui génère un panel d'administration pour créer mes Models et mes Tables que je détaillerai à l'issue de ma partie backend.

J'ai créée trois tables : Demandes, Ressources et Demo

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI		auto_increment
created_at	timestamp	YES			
updated_at	timestamp	YES			
user_id	integer	NO			
Labs	varchar	YES			
Date_de_debut	date	YES			
Date_de_fin	date	YES			
user_name	varchar	YES			
user_email	varchar	YES			
ressource_id	integer	NO			
nb_demande	integer	YES			

Field	Type	Null	Key	Default	Extra
id	integer	NO	PRI		auto_increment
title	varchar	YES			
excerpt	varchar	YES			
body	longtext	YES			
image	varchar	YES			
exemple	varchar	YES			
created_at	timestamp	YES			
updated_at	timestamp	YES			
slug	varchar	YES			
lien	varchar	YES			
lien2	varchar	YES			
lien3	varchar	YES			
nb_visite	integer	YES		0	

La table demande

La table ressources

La table demo

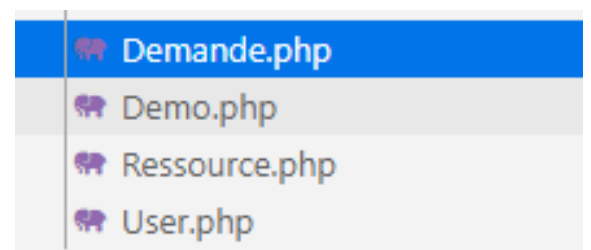
demos					
Field	Type	Null	Key	Default	Extra
id	integer	NO	PRI		auto_increment
image	varchar	YES			
created_at	timestamp	YES			
updated_at	timestamp	YES			
description	longtext	YES			

Nos trois « models » (Demande, Ressource et Demo) ont été créées dans notre projet en même temps que nos tables :

La table et le Model User sont créés par défaut avec LARAVEL à la création du projet

Relationship (User et Demande)

Relation « one to many » entre la table « User » et la table « Demande »



❖ Le model Demande

```
Demande.php > ...  
<?php
```

Une Demande appartient à
aucun ou à un utilisateur

```
namespace App;  
  
use Illuminate\Database\Eloquent\Model;  
  
class Demande extends Model  
{  
    public function user(){  
        return $this->belongsTo('App\User');  
        //chaque demande doit etre relié a notre utilisateur  
        // ici une chaque demande appartient a un seul utilisateur  
    }  
}
```

❖ Le Model User

```
public function demande()  
{  
    return $this->hasMany(\App\Demande::class);  
    //Un utilisateur peut avoir une ou plusieurs demandes  
}
```

Et un utilisateur peut avoir Zéro ou plusieurs demandes :

- Relationship (User et Demande)
- Relation one to many entre la table Ressource et la table Demande

❖ Le model Ressource

Une Ressource peut avoir zéro ou plusieurs demandes

```
app > DemandeController.php RessourcesController.php Ressource.php X .env
app > Ressource.php > ...
1  <?php
2
3  namespace App;
4
5  use App\Demande;
6  use Illuminate\Database\Eloquent\Model;
7
8
9  class Ressource extends Model
10 {
11     protected $fillable = [
12         'title', 'image', 'message',
13     ];
14
15     public function demande(){
16
17         return $this->hasMany(\App\Demande::class);
18     }
19 }
20
21
22
```

❖ Le Model Demande

Une Demande concerne zéro ou une ressource

```
app > Demande.php > Demande
1  <?php
2
3  namespace App;
4
5
6  use Illuminate\Database\Eloquent\Model;
7
8  class Demande extends Model
9  {
10     public function user(){
11
12         return $this->belongsTo('App\User');
13         //chaque demande doit etre relié a notre utilisateur
14         // ici une chaque demande appartient a un seul utilisateur
15     }
16
17
18
19
20     public function ressource(){
21
22         return $this->belongsTo(\App\Ressource::class);
23     }
24
25 }
```

Création de nos contrôleurs

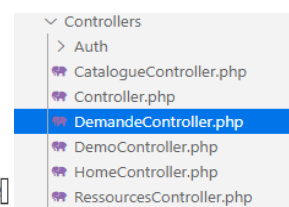
Exemple pour la création du controller « Demande »

❖ Dans la console :

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Testez le nouveau système multiplateforme PowerShell https://aka.ms/pscore6

PS C:\Users\mauri\Project_Labs_Ressources> php artisan make:controller DemandeController --resource[]
```



Cette commande va nous générer un C.R.U.D. (Create, Read, Update, Delete) dans le fichier « DemandeController ».

Et une route « ressource » dans le fichier web « php ».

- Une fonction CREATE qui retournera la vue « create » avec la liste des titres de nos ressources

```
public function create()  
  
    $ressources = Ressource::all();//avant le pluck on nous renvoyé un tableau indexé  
    $ressources_rebuild = array();  
    foreach($ressources as $ressource):  
        $ressources_rebuild[$ressource['id']] = $ressource['title'];//transformé en tableau associatif  
    endforeach;  
  
    return view('demandes.create')->with(['ressources'=> $ressources_rebuild,]);
```

- à la page « home » si « success ».

```
public function store(Request $request){ //pour notre mise a jour on copie la fonction
store et on la met dans la fonction update */
{

    $this->validate($request,[

        'Labs'=>'required|alpha|between:3,20',
        'Date_de_debut'=>'required|date|after:now',
        'Date_de_fin'=>'required|date|after:Date_de_debut',
        'ressources'=>'required'

    ]);

    date_default_timezone_set('Europe/Paris');

    $demande=new Demande;
    $test_labs = Demande::where('Labs', $request->input('Labs'))
                        ->get()
                        ->count();

    if($test_labs == 0){
        $demande->Labs=trim($request->input('Labs'));
        $demande->ressource_id=$request->input('ressources');
        $demande->Date_de_debut=$request->input('Date_de_debut');
        $demande->Date_de_fin=$request->input('Date_de_fin');
        $demande->user_name=auth()->user()->name;
        $demande->user_email=auth()->user()->email;
        $demande->created_at=time();

    } else {
        return redirect('/demande/create')
        ->with('error',"Ce nom existe déjà !");
    }

    $demande->user_id=auth()->user()->id;// grace a la fonction auth on recupere l'id et l'utilisateur
    $demande->save();

    return redirect('/home') ->with('success', "votre demande a été envoyé !");
}
```



```

public function show($id)// notre fonction show nous la copions dans la fonction edit
{
    $demande=Demande::find($id);

    $demande->nb_demande++;
    $demande->save();
    $ressource=Ressource::find($id);

    return view ('demandes.show') ->with(['demande'=>$demande,'ressource'=> $ressource ]);
}

```

Cette fonction retournera la vue « show » avec la demande et mes ressources choisies correspondant à « l'id » de l'utilisateur.

- La fonction EDIT retourne le formulaire avec les champs renseignés

```

public function edit($id)
{
    $demande=Demande::find($id);
    $ressources = Ressource::all();//avant le pluck on nous renvoyé un tableau indexé
    $ressources_rebuild = array();

    foreach($ressources as $ressource):
        $ressources_rebuild[$ressource['id']] = $ressource['title'];//transformé en tableau associatif
    endforeach;

    return view ('demandes.edit') ->with (['ressources' => $ressources_rebuild,'demande' => $demande ]);
}

```

- La fonction UPDATE : Cette fonction met à jour notre formulaire et redirige vers la page « home »

```

public function update(Request $request, $id)
{
    $this->validate($request,[
        'Labs'=>'required|alpha|between:3,20',
        'Date_de_debut'=>'required|date|after:now',
        'Date_de_fin'=>'required|after:Date_de_debut',
        'ressource'=>'required'
    ]);
    $test_labs = Demande::where('Labs', $request->input('Labs'))
    ->get()
    ->count();
    if($test_labs == 0){
        $demande=Demande::find($id);//on va chercher la demande que l'on veut modifier
        $demande->ressource_id=$request->input('ressource');
        $demande->Labs=trim($request->input('Labs'));
        $demande->Date_de_debut=$request->input('Date_de_debut');
        $demande->Date_de_fin=$request->input('Date_de_fin');
        $demande->save();
    } else {
        return redirect('/demande/create')
        ->with('error',"Ce nom existe déjà !");
    }

    return redirect('/home')
    ->with ('success', "Votre demande a bien été modifié !");
}

```

- La fonction DELETE supprime la demande

```

public function destroy($id)
{
    $demande=Demande::find($id);//on recupere lid de la demande
    $demande->delete();//on supprime la demande
    return redirect('/home')
    ->with('success', "Votre demande a bien été supprimée !");
}

/**
 * Show the form for creating a new resource.
 *
 * @return \Illuminate\Http\Response
 */
public function create_from($id)
{
    $ressources = Ressource::All();//avant le pluck on nous renvoyé un tableau indexé
    $ressources_rebuild = array();
    foreach($ressources as $ressource):
        $ressources_rebuild[$ressource['id']] = $ressource['title'];//transformé en tableau associatif
    endforeach;
    $ressource_selected = strip_tags($id);//retourne une chaine str apres avoir supprimé tous les balises php et html
    dd($ressource_selected);

    return view ('demandes.create')->with(['success','Votre ressources a été sélectionné','ressources'=> $ressources_rebuild,'ressource_s

```

Créations des autres contrôleurs

❖ DemoControlleur :

- Une fonction « construct » qui sera exécutée avant toutes les autres. Dans celle-ci, il y aura un middleware ayant en paramètre « auth ». La page « ressource » ne sera accessible qu'aux utilisateurs authentifiés.
- Elle contiendra une fonction avec une requête qui ira récupérer l tous les infos dans la table et retournera la vue demo

❖ RessourcesControlleur

- Une fonction « construct » qui sera exécutée avant toutes les autres. Dans celle-ci, il y aura un middleware ayant en paramètre « auth ». La page « ressource » ne sera accessible qu'aux utilisateurs authentifiés.
- Une fonction index avec une requête qui ira récupérer dans la table ressource les dernières ressources créées par ordre descendant et retournera la vue ressource
- Une fonction show avec une requête qui retournera le « slug » (ou « Friendly URL »). Le « slug » est la partie d'une URL qui identifie le titre d'un article. Il correspondant à la ressource et retournera la vue show

❖ HomeControleur contiendra

- Une fonction « construct » qui sera exécutée avant toutes les autres. Dans celle-ci, il y aura un middleware ayant en paramètre « auth ». La page « ressource » ne sera accessible qu'aux utilisateurs authentifiés.
- Une fonction index avec une requête qui récupèrera « l'id » de l'utilisateur identifié et retournera la vue « Home » avec ses demandes

Création de nos routes

Dans le fichier web.php

```
//////////DEMANDE//////////
Route::post('/demande', 'DemandeController@create');
Route::get('/demande/create/{id}', 'DemandeController@create_from');
Route::resource('demande', 'DemandeController');

//////////LABS//////////
Route::get('/catalogue', 'RessourcesController@index');
Route::get('/catalogue/{slug}', 'RessourcesController@show');

//////////AUTHENTIFICATION//////////
Auth::routes();

//////////HOME(dashboard)//////////
Route::get('/', 'HomeController@index');
Route::get('/home', 'HomeController@index')->name('home');
Route::get('/all', 'DemandeController@index');

//////////DEMO APPLICATION//////////
Route::get('/Demo', 'DemoController@demo');
```

Tests de notre application

On vérifie la cohérence de notre code par rapport aux demandes du cahier des charges.

❖ Test formulaire de demande

Un nom de labs (voir annexe) :

- Est requis
- Doit être unique
- Doit comporter entre 3 et 20 caractères
- Ne doit pas contenir d'espace

Une date de début (voir annexe) :

- Ne doit pas être inférieure ou égale à la date du jour
- Est requise

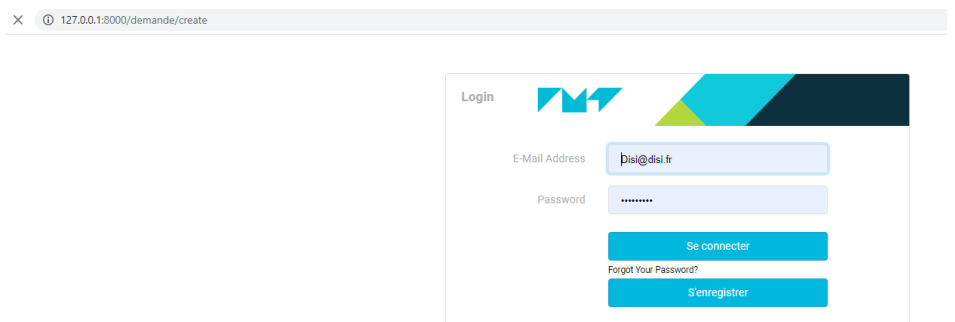
Une date de fin (voir annexe) :

- Ne doit pas être inférieure ou égale à la date du jour et à la date de début
- Est requise

❖ Test de connexion, déconnexion, changer de mot de passe

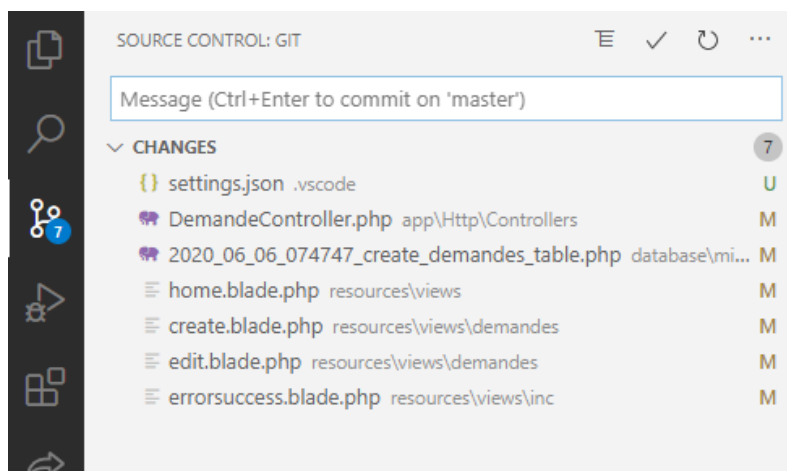
- Test d'accès de notre application dans l'URL

Exemple : Si un utilisateur non identifié tente un accès à notre formulaire de demande via l'URL, il sera renvoyé vers la page login



Sauvegarde

Dépôt de l'application sur GitHub en mode privée avec l'extension git installée sur l'éditeur de texte Visual studio code ou via la console en ligne de commande.



L'application a été livrée en version 1 mais n'as pas encore été déployée !

Fonctionnalité hors cahier des charges

J'ai pris l'initiative de contacter la DISI via l'envoi d'un mail. J'ai utilisé l'outil d'envoi de Mail MAILTRAP pour tester l'envoi de de courriel pendant la phase de développement : <https://mailtrap.io/signin>

❖ Configuration de notre fichier « .env »

On renseigne les champs ci-dessous avec les informations récupérées sur MAILTRAP après avoir créé un compte

```
25  
26 MAIL_MAILER=smtp  
27 MAIL_HOST=smtp.mailtrap.io  
28 MAIL_PORT=2525  
29 MAIL_USERNAME=dee8b4bf2b64c4  
30 MAIL_PASSWORD=ecf117f88bfa13  
31 MAIL_ENCRYPTION=tls  
32 MAIL_FROM_ADDRESS="imt@labs.com"  
33 MAIL_FROM_NAME="Admin"  
34
```

❖ Création d'une class mail

Dans cette class, on renseigne la fonction BUILD Elle retournera la vue avec les données

```
Windows PowerShell  
Copyright (C) Microsoft Corporation. Tous droits réservés.  
  
Testez le nouveau système multiplateforme PowerShell https://aka.ms/pscore6  
  
PS C:\Users\mauri\Project_Labs_Ressources> php artisan make:mail ContactMail
```

```
1  
2 public function build()  
3 {  
4     return $this->markdown('emails.contact.contact-form', $this->data);  
5     //retourn la vue contact qui se trouve dans le dossier view/email/contact  
6 }
```

❖ Création de notre vue CREATE avec un formulaire d'envoi

Elle comprendra une fonction CREATE qui retournera la vue CREATE contenant le formulaire



❖ Création d'un controller

Elle contient un fonction store qui récupèrera les données :

- Un mail ::to
- Mail::to('test@test.com')->send(new ContactMail(\$data));
return redirect('/home')->with('success', "Votre message a bien été envoyé !")

Elle comprend :

- L'adresse de l'administrateur
- L'envoi avec la méthode « send »
- La création d'une instance de la classe Contact avec la transmission des données

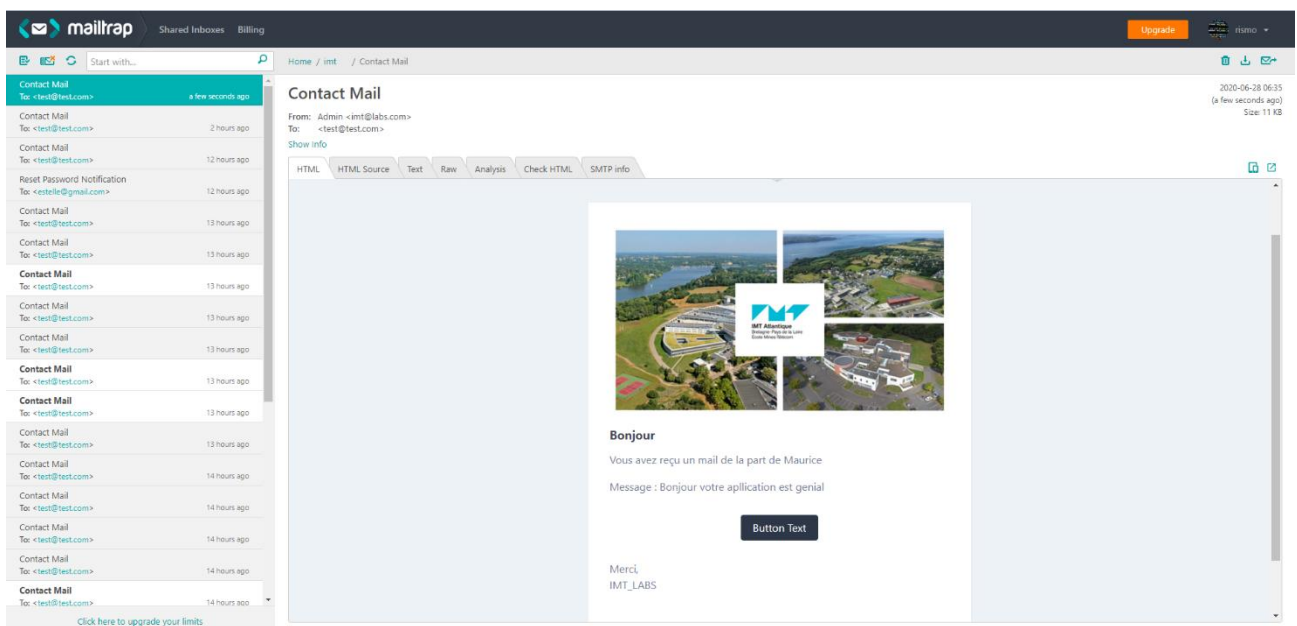
❖ Creation de notre vue contenant le mail

```
✓ emails \ contact
  contact-form.blade.php
```

Notre mail en MARKDOWN (qui est un langage de balisage) qui s'affichera dans MAILTRAP

```
1  -
2  @component('mail::message')
3
4  ![imt](https://www.imt-atlantique.fr/sites/default/files/users/user336/IMT%20Atlantique%20-%203%20campus.)
5  # Bonjour
6
7  Vous avez reçu un mail de la part de {{Auth::user()->name}}
8
9  Message :
10 @foreach ($data as $d)
11 |   {{ $d }}
12 @endforeach
13
14 @component('mail::button', ['url' => ''])
15 Button Text
16 @endcomponent
17
18 Merci,<br>
19 {{ config('app.name') }}
20 @endcomponent
21
```

Affichage sur MAILTRAP



Recherche et veille

Pour ma veille, mes recherches et mes solutions pour résoudre les problèmes, j'ai utilisé les sites suivants :

<https://laravel.com/>

<https://laravel.sillo.org/>

<https://openclassrooms.com/fr/>

<https://getbootstrap.com/>

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

<https://www.grafikart.fr/>

<https://voyager.devdojo.com/>

Synthèse du projet

Ce projet m'a été proposé par mon référent de stage Emanuel chez **IMT Atlantique**.

Vu la période particulière de confinement, ce stage s'est déroulé entièrement en télétravail. J'ai tout de même suivi la vie de l'entreprise grâce à un accès à la messagerie de l'entreprise.

Un outil de gestion de projet (en mode Agile) a été utilisé avec des tâches qui m'étaient allouées. Je faisais régulièrement le point sur l'avancée du projet avec mon référent en visioconférence.

Remerciements

Je remercie **IMT Atlantique** et tout particulièrement mon référent EMMANUEL BRAUX, pour sa confiance accordée pour la réalisation de ce projet qui m'a permis de mettre en application mes connaissances, de découvrir de nouveaux outils, environnement de travail et des méthodes de travail.

Ce fut une bonne expérience dans laquelle j'ai pu m'épanouir.

Annexe 1 : Tests formulaire

❖ Labs

Ce nom existe déjà !



IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Demande De Labs

NOM DU LAB

Backend

The labs field is required.



IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Demande De Labs

NOM DU LAB

The labs must be between 3 and 20 characters.



IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Demande De Labs

NOM DU LAB

Ba

The labs may only contain letters.



IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Demande De Labs

NOM DU LAB

Ba C

❖ Dates

The date de debut must be a date after now.

The date de fin must be a date after date de debut.



IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Demande De Labs

NOM DU LAB

Backend

RESSOURCES DISPONIBLES

PHP

DATE DE DEBUT

24/06/2020

DATE DE FIN

23/06/2020

Envoyer

Annexe 2 : Réinitialisation du mot de passe

Login

E-Mail Address

Password

[Se connecter](#)

[Forgot Your Password?](#)

[S'enregistrer](#)

Reset Password

E-Mail Address

[Send Password Reset Link](#)

Reset Password

We have emailed your password reset link!

E-Mail Address

[Send Password Reset Link](#)

Annexe 3 : Se déconnecter

MANUEL

[Faire une demande](#) [voir le catalogue](#)

comment faire une demande

BONJOUR
MANUEL



vous n'avez aucune demande

IMT Atlantique
Bretagne-Pays de la Loire
Ecole Mines-Télécom

Tableau de bord

[Admin](#)

[Se déconnecter](#)