Application web de gestion des Projets

Guide et documentation

**Développée par : Abdessamad et Sofiane DERRAZ - 2019**

**Entreprise : ISOPROM**

Sommaire :

[Sommaire : 2](#_Toc19092284)

[Table des Figures : 3](#_Toc19092285)

[Introduction : 4](#_Toc19092286)

[I. Page d’Installation 5](#_Toc19092287)

[II. Page d'accueil principale (après la première inscription) 7](#_Toc19092288)

[III. Page de Login / authentification 9](#_Toc19092289)

[IV. Page d’accueil 10](#_Toc19092290)

[V. Interface Admin et Gérant 11](#_Toc19092291)

[VI. Interface Chef de Projet 12](#_Toc19092292)

[VII. Task Index : 13](#_Toc19092293)

[VIII. Tasks\edit : 14](#_Toc19092294)

[IX. Interface Employé 17](#_Toc19092295)

[X. DataTimePicker / Notification : 19](#_Toc19092296)

[XI. Controle d’Accès : 21](#_Toc19092297)

Table des Figures :

[Figure 1. Admin authentification 5](#_Toc524464566)

[Figure 2. HomeController index 7](#_Toc524464567)

[Figure 3. Page Login 8](#_Toc524464568)

[Figure 4. Interface Admin et Gérant 10](#_Toc524464569)

[Figure 5. Interface Chef de Projet 11](#_Toc524464570)

[Figure 6. Tasks\index : 12](#_Toc524464571)

[Figure 7. Liste des Taches 12](#_Toc524464572)

[Figure 8. Code « Modifier une Tâche » 13](#_Toc524464573)

[Figure 9. Modifier une Tâche(top) 13](#_Toc524464574)

[Figure 10. Modifier une Tâche(bottom) 15](#_Toc524464575)

[Figure 11. Interface Employé 17](#_Toc524464576)

[Figure 12. Tache employée 18](#_Toc524464577)

[Figure 13. Date Time Picker : 19](#_Toc524464578)

[Figure 14. Notification : 20](#_Toc524464579)

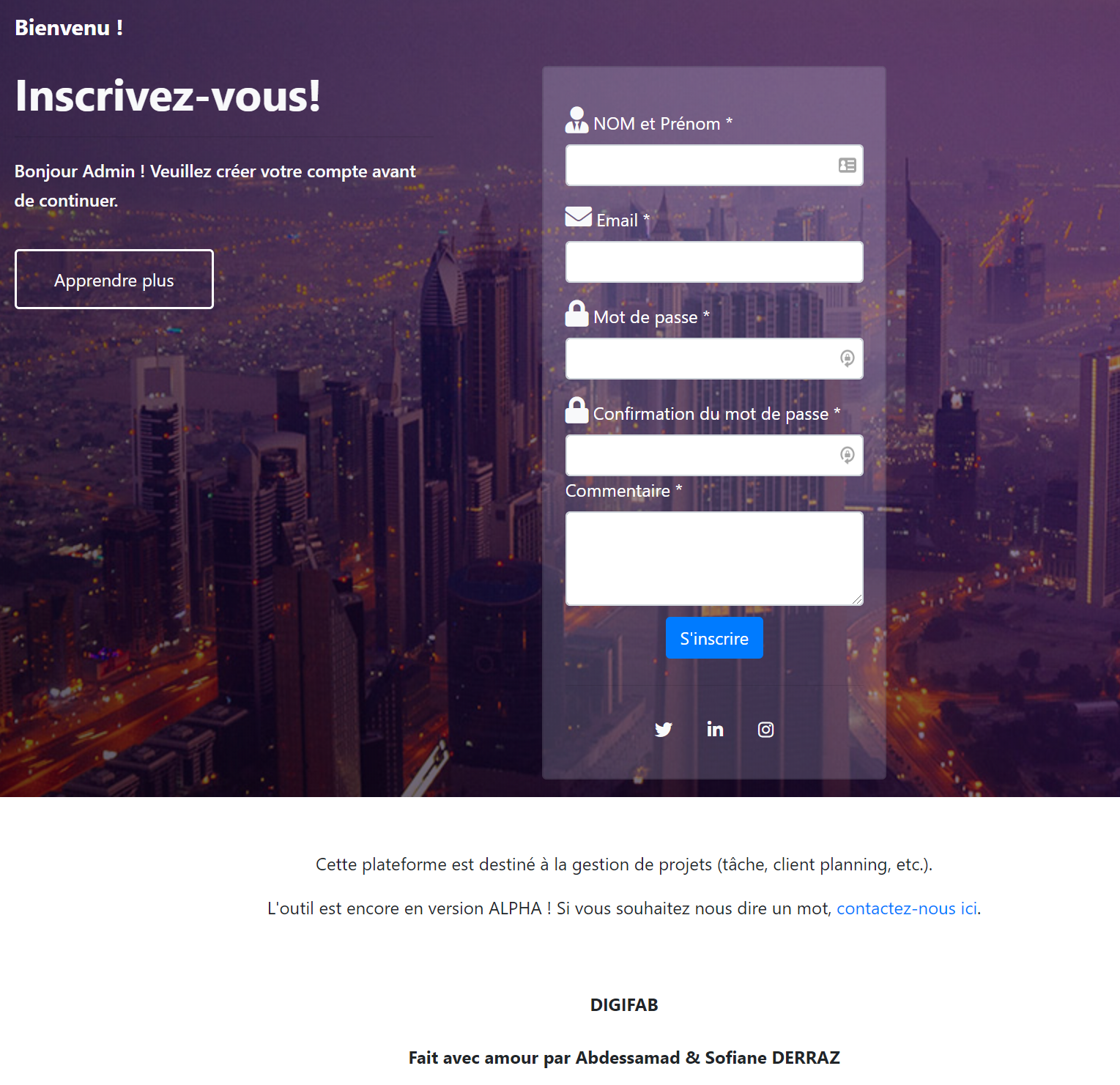
[Figure 15. Code Notification (view) 20](#_Toc524464580)

[Figure 16. Code Notification (Controller) 20](#_Toc524464581)

Introduction :

Cette application web a été conçue et développée pour but principal de permettre la gestion des projets et taches au sein d’une entreprise ou société, en effet la plateforme offre une Gestion des employés représentant les utilisateurs de l’application, avec une possibilité de suivis des taches de chaque employé, ainsi que les projets de chaque chef de projet.Page d’Installation

Lors de votre premier accès à l’application par l’URL : *www……….com*, une page d’inscription d’Admin seras mise à votre disposition.

1. ****Admin authentification

Une fois inscrit, l’admin a accès à presque toutes les actions de l’application.

* **Notes sur le code :**

View :

**AdminRegistration:**

Se situe dans le fichier (views/AdminRegistration.blade.php), représente le Template pour la création du premier utilisateur Admin.

Controller :

**HomeController :**

Se situe dans le fichier (Controllers/HomeController), l'une de ses utilisations, est la création administrateur (adminRegisterIndex and firstAdminStore)

1. Page d'accueil principale (après la première inscription)

* **Notes sur le code :**

Views :

**Layout\Structure :**

Se situe dans le fichier (views/layouts/structure.blade.php), représente le Template principal que tous les autres vus vont hériter (étendre).

JS : Liens CSS :

[../jquery.min.js](https://code.jquery.com/jquery-3.3.1.min.js) [*../css/bootstrap.min.css*](https://stackpath.bootstrapcdn.com/bootstrap/4.1.2/css/bootstrap.min.css)

[../umd/popper.min.js](https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js) *..[/css/jquery.dataTables.css](https://cdn.datatables.net/1.10.19/css/jquery.dataTables.css)*

[../js/bootstrap.min.j](https://stackpath.bootstrapcdn.com/bootstrap/4.1.2/js/bootstrap.min.js) *../*[*css/font-awesome.min.css*](https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css)

Controllers :

* **HomeController :**

Lors de l’accès à l’application, la fonction **index ()** qui se trouve dans app/Http/Controllers/HomeController.php seras exécuté.

**Fonction Index () :**

Cette fonction va premièrement vérifier si la base de données des utilisateurs est vide en appelant la fonction **CheckFirstRegistration(),** dans le cas vrai, une redirection vers la route 'admin.register.index' sera effectuée.

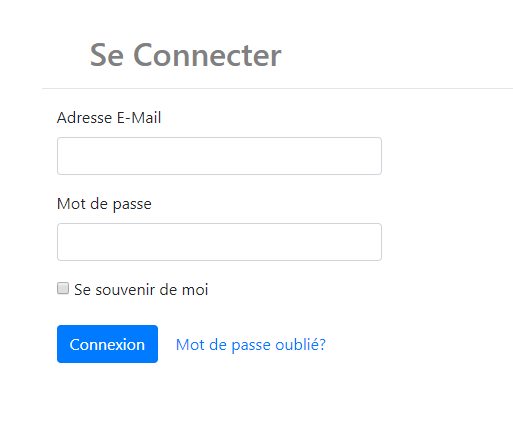
Dans le cas contraire, la vue 'index' est appelée tout en envoyant les variables suivantes :

$LastFewProjects : les 3 derniers projets triés par état en-cours.

$LastFewTasks : les 3 dernières Taches triées par état en-cours.

1. HomeController index
2. Page de Login / authentification

Une fois la base de données des utilisateurs est remplie, l’accès à l’application sera automatiquement initié par une page d’authentification.

1. Page Login

Cette page va étendre la structure ***structure.blade.php*** (voir Layout Structure) donc hérite ses scripts et liens css.

L’utilisateur ayant oublié son mot de passe pourra choisir l’option *(Mot de passe oublié ?)* et ensuite sera mené à insérer une adresse email valide, pour recevoir le lien qui le mènera ensuite à la page de réinitialisation du mot de passe. (Action en cours de modification...)

1. Page d’accueil

L’utilisateur authentifié aura accès à la page d’accueil, celle-ci contient une barre de menu en haut, un calendrier à gauche et Les Derniers Projets et Taches (en-cours) aux milieux.

* **Notes sur le code :**

**Le calendrier :**

-le code HTML du calendrier se trouve dans :

(views/Calendars/mycalendar.blade.php), ce fichier utilise l’objet $month de classe (app\Calendar\Month)

-Son css défini dans le Template :

*{{asset ('/css/calendar.css')}}*

-La classe Month définit les éléments suivant :

* **Variables globales :**

$days : tous les jours de la semaine

$months : Tout les mois de l’année par nom

$month et $year : par défaut =null : Le mois compris entre 1 et 12 plus l’année entière passée dans l’URL de la page.

* **Fonctions de la classe Month :**

**f.GetStartingDay() :** retourne le premier jour du mois courant sous format (Y-M-01)

**f.ToString() :** retourne le mois en toutes lettres (ex.: Janvier 2018)

**f.GetWeeks() :** retourne le nombre de semaines du mois

**f.WithinMonth($date) :** vérifie si le Jour ($date) est inclus dans le mois courant !

**f. Inweekend ($date) :** vérifie si le jour est en samedi ou dimanche

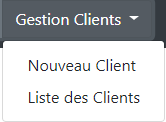
**f. Isfreeday($date)** : vérifie si le jour est une fête nationale est retourné le titre de la fête (pour l’année 2018 – 2019 - 2020)

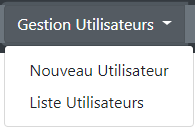
**f.Iscurrentdate($date) :** vérifie si la $date est le jour courant

**f.nextMonth() :** renvois le mois suivant sous format (Objet de Classe Month).

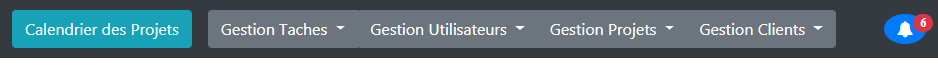
**f.previousMonth() :** renvois le mois précédent.

1. ***Interface Admin et Gérant***
2. Interface Admin et Gérant









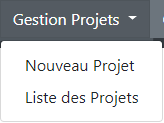
Nom

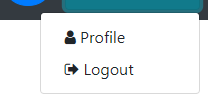
***Notifications :***

Derniers Projets créés

Ou modifiés







NomAdmin

En accédant à un projet, l’Admin et le Gérant auront le droit de modifier et clôturer un projet.

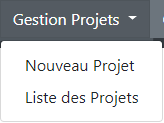
NB : Une fois le Projet clôturé, aucune modification n’est permise concernant le projet, ses taches et les membres concernés.

1. ***Interface Chef de Projet***
2. Interface Chef de Projet



Nom





Liste des Projets de ce Chef de Projet

Lister les taches de cet utilisateur

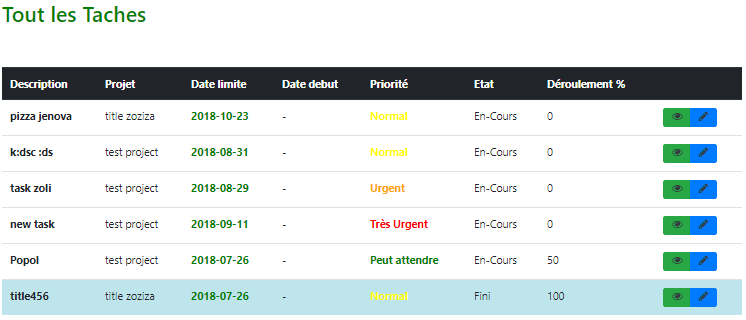
-Le Chef de Projet a accès aux Projets, il peut les modifier, mais ne peut pas clôturer un Projet.

-Le Chef de Projet à accès aux Tâches, il peut les modifier, ajouter ou supprimer des membres, et peu valider la tâche.

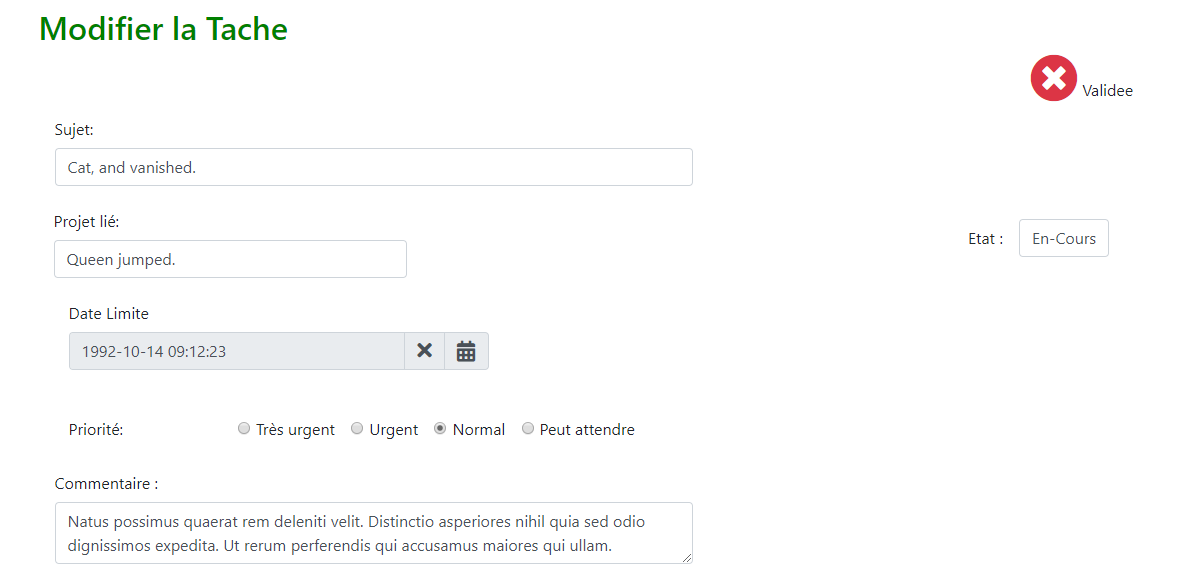
1. ***Task Index :***
2. Tasks\index :



Chaque tâche peut avoir plusieurs employés qui y travaillent, en même temps, chaque relation "task\_user" a une date de début et une date de fin, pour décider de la date de début à afficher sur une tâche, nous avons sélectionné les plus petits.

1. Liste des Taches
2. Tasks\edit :
3. Code « Modifier une Tâche »

À partir du code ci-dessus, nous avons renvoyé toutes les données requises pour les deux tables, y compris le nombre de tâches pour chaque employé.

1. Modifier une Tâche(top)

Pour valider une tâche, il faut cliquer sur le bouton circulaire rouge pour enregistrer les modifications.

Comme vous pouvez le voir sur les figures, il y a deux tableaux, un pour les employés actuels travaillant sur la tâche en cours et un pour les autres employés.

1. Modifier une Tâche(bottom)

Pour joindre un employé à une tâche, vous devez le sélectionner en cliquant sur le bouton (bleu) + situé à côté.

Pour détacher un employé d’une tâche, vous devez le sélectionner en cliquant sur le bouton (rouge) X situé à côté.

Vous pouvez attacher ou détacher autant de fois que possible, mais aucune modification n'est enregistrée à moins de cliquer sur le bouton d'enregistrement (vert).

* **Notes sur le code :**

View :

{{Asset (‘Resposive\_Task\_User\_Select.js’)}}

addRow(v, isAdd, name, tcount) *création d’un élément avec les donnes nécessaire.*

addTo(v, name, tcount) *ajouter un élément a lune des tables*.

removeFrom(v, name, tcount) *retirer un élément a lune des tables.*

Controllers :

**HomeController :**

Se situe dans le fichier (Controllers/TaskController), ‘task update’

1. ***Interface Employé***

Nom

Afficher tous les Projets

Lister les tâches attribuées à cet employé

1. Interface Employé

* **Notes sur le code :**

Controllers :

* **TaskController :**

**Fonction MyTasks ($id):** cette fonction prend en paramètre l’id d’utilisateur courant (l’employé).

Retourne la vue ‘tasks.mesTaches’ avec les variables :

Le tableau $ts : stockant toutes les tâches de cet employé

$t->project\_title : retourne le titre du projet relié à chaque tache

**//** les 2 variables si dessous ne sont pas encore pris en charge ils seront traités dans une prochaine version de l’application **//**

**$s\_d :** la date du début de tache(en cours de traitement)

**$f\_d :** la date de fin de tâche (en cours de traitement)

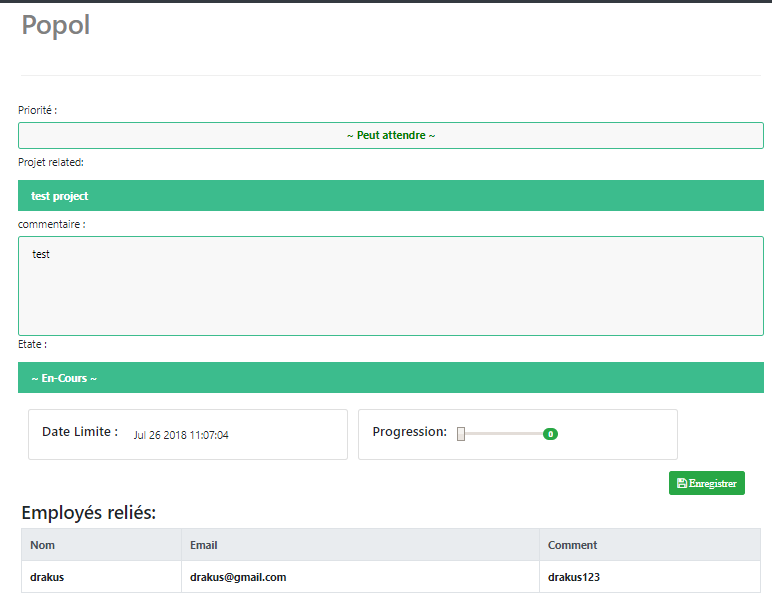
s

Views :

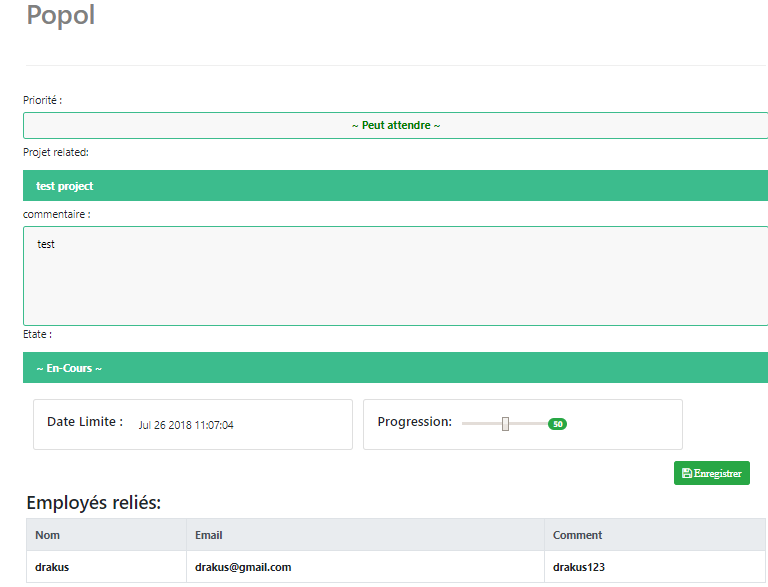
* **Task\show :**

En accédant à une de ses tâches, l’employé peut modifier le pourcentage de progression dans la tache tant que le projet relié à cette tâche n’est pas clôturé.

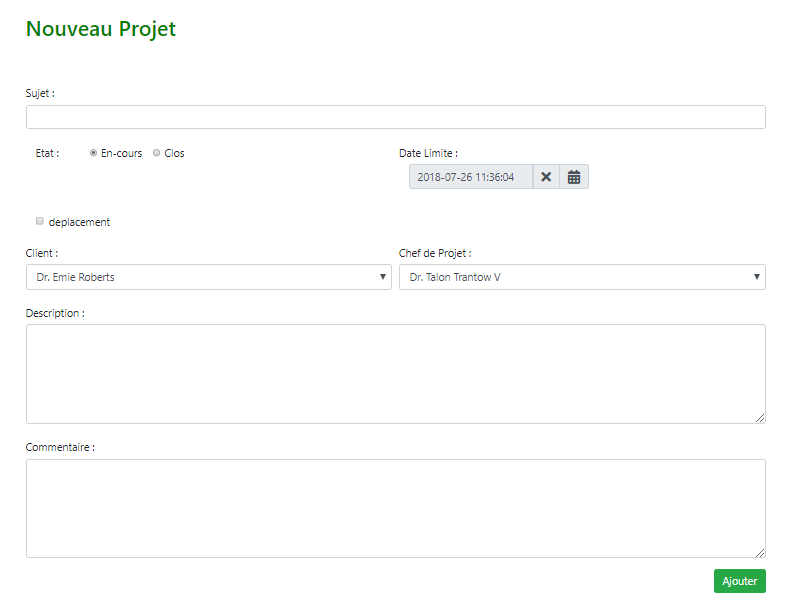
1. Tache employée



**Task Name**



1. DataTimePicker / Notification :

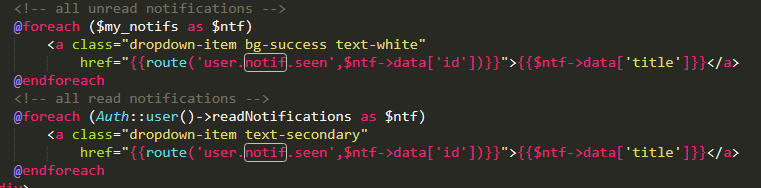
1. Date Time Picker :

Pour sélectionner un datetime, nous pouvons cliquer n'importe où sur le datetimepicker pour afficher le calendrier.

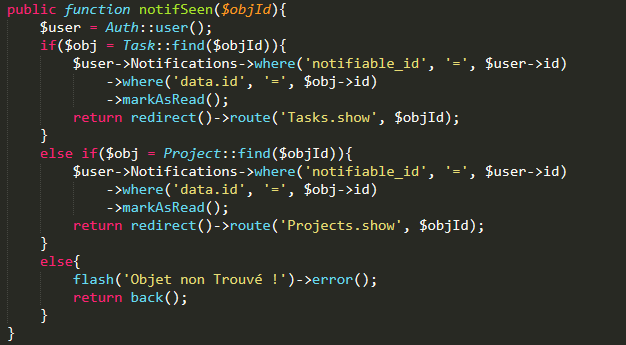
Sélectionner une date affichera le sélecteur de temps, alors la datetimepicker changera de valeur

Pour plus d’information sur le datatimepicker :

https://github.com/smalot/bootstrap-datetimepicker

1. Notification :
2. Code Notification (view)

Comme vous l'avez peut-être remarqué, le code ci-dessous est loin d'être parfait (mauvaise condition « if »), une version fixe sera disponible au plus vite possible.

1. Code Notification (Controller)

Tout ce code a été créé pour afficher les notifications actuelles de l'utilisateur actuel et pour changer l'état de chaque notification, après avoir cliqué dessus, il affiche la liste des notifications actuelles, pour pouvoir ensuit clinker sur une pour l’afficher.

1. Controle d’Accès :

Le contrôle d’accès se fait via les Policies se trouvant dans le dossier (app/Policies), qui représentent des classes organisant les autorisations pour toutes les actions des différents modèles (Task, Project, User et Client).