



Gestion des notes

NOTEBLADE

Presented by: Abdelljabar Hamri



Table des Matières

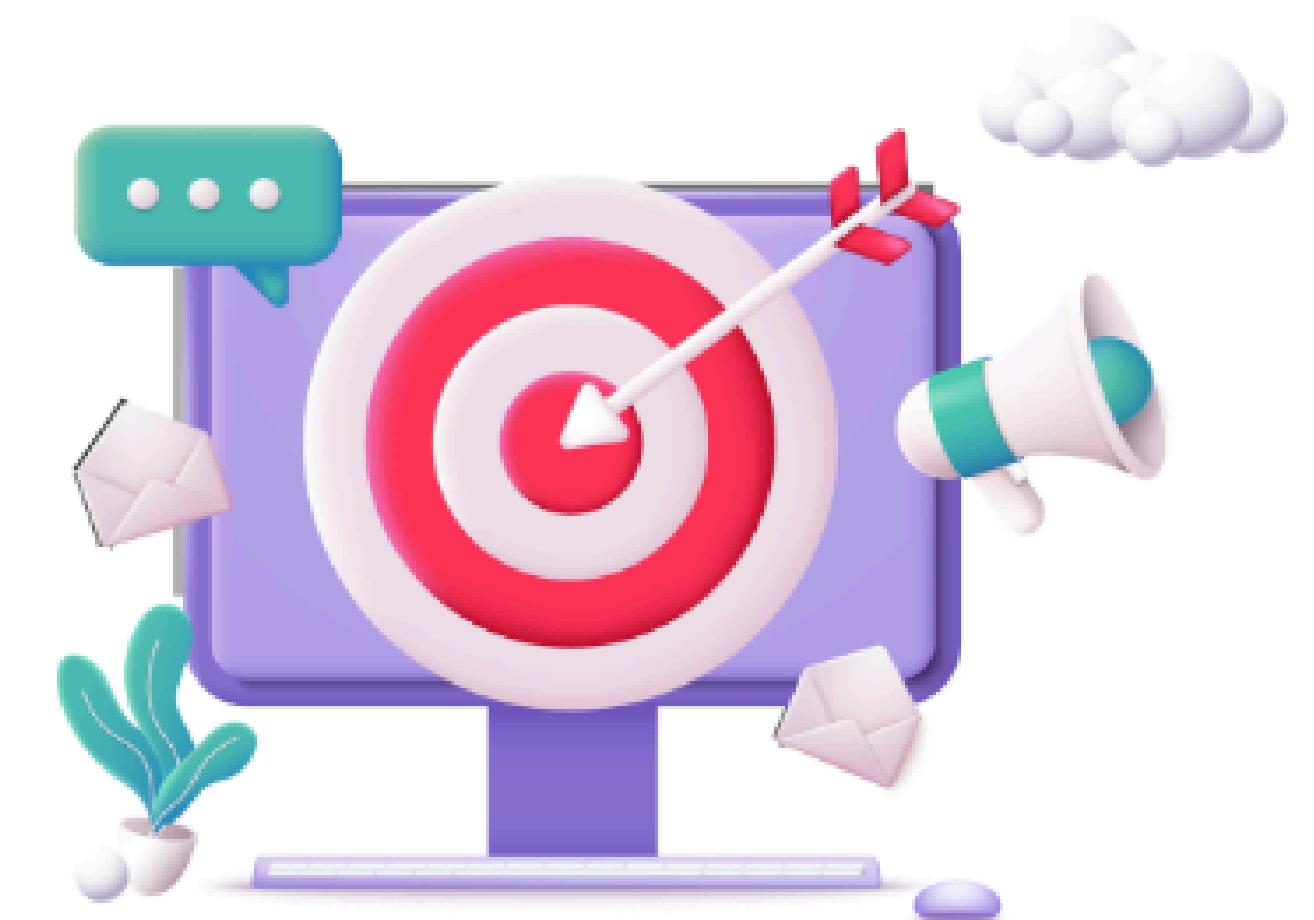
- *Introduction*
- *Spécifications Fonctionnelles*
- *Vue globale du projet*
- *Stack Technique*
- *Architecture du Système*
- *SécuritéClients*



[Se connecter](#)[Register](#)

Introduction

- NoteBlade est une application web de gestion de notes conçue pour améliorer la productivité des utilisateurs. Elle permet de créer, organiser et gérer des notes de manière simple et efficace.



Spécifications Fonctionnelles



Authentification des Utilisateurs



Gestion des Notes

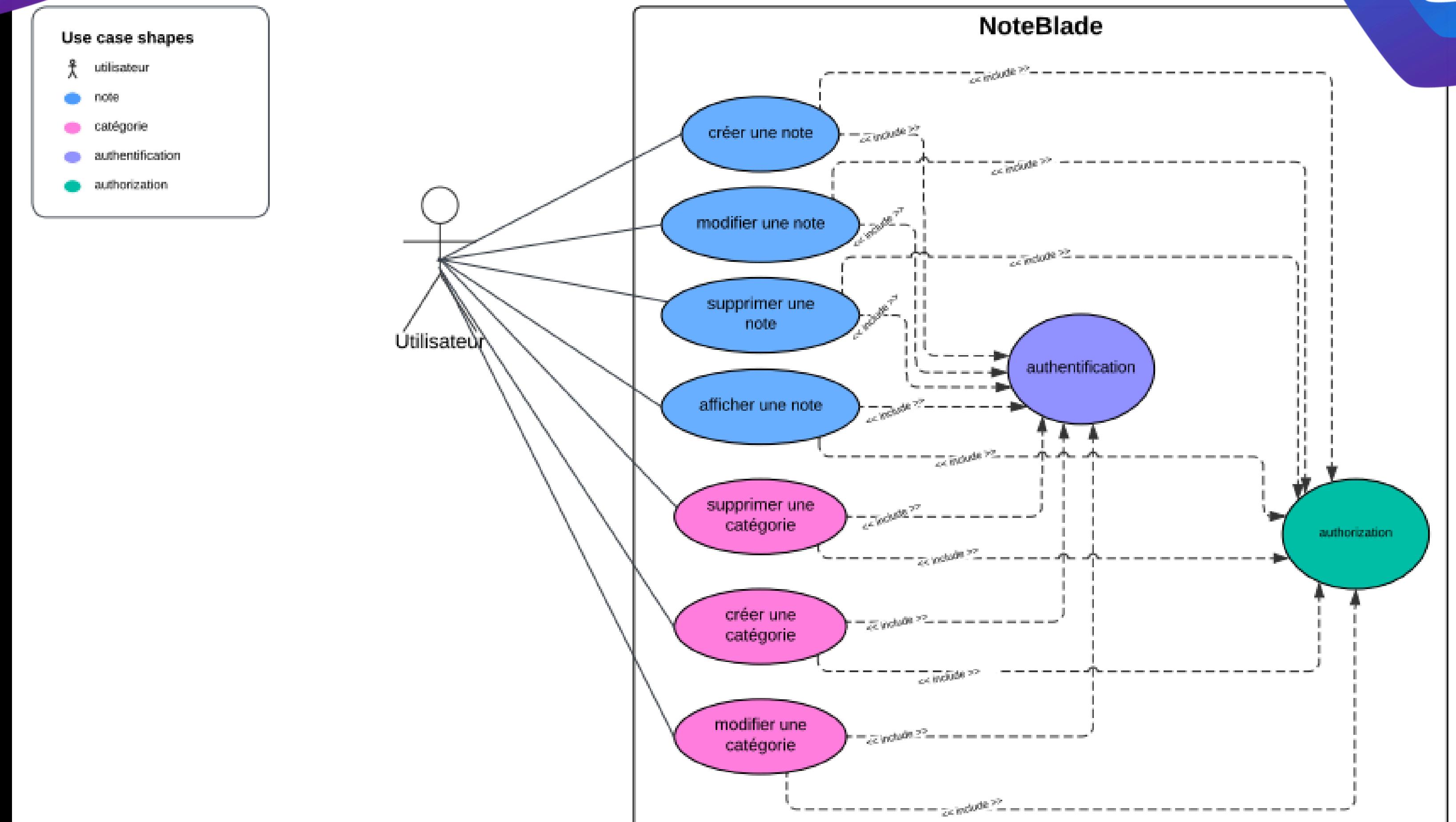


Gestion des Catégories



Recherche et Pagination

Vue globale du projet



STACK TECHNIQUE



Langage de Programmation

PHP : Le choix de PHP s'explique par sa maturité et sa popularité pour le développement web.



Base de Données

MySQL : MySQL est un SGBD open-source reconnu pour sa performance



Outils de Développement

Artisan Console - Eloquent ORM - Blade Templating Engine



Framework

Laravel : Laravel est choisi pour sa simplicité



Serveur d'Application

Apache : Apache est un serveur web largement utilisé qui offre stabilité et sécurité.

Architecture du Système

MVC (Modèle-Vue-Contrôleur) de Laravel. Cela permettra une séparation claire des préoccupations et une maintenance facilitée.



Modèle de Données

Utilisateur	Note	Catégorie
<ul style="list-style-type: none">- id : int- name : string- email : string- password : string	<ul style="list-style-type: none">+ id : int+ title : string+ contenu : string+ priority : string+ la date : date	<ul style="list-style-type: none">+ id : int+ name : string
<ul style="list-style-type: none">+ register()+ login()+ logout()	<ul style="list-style-type: none">+ créerNote()+ modifierNote()+ supprimerNote()+ rechercheNote()	<ul style="list-style-type: none">+ créerCategory()+ modifierCategory()+ supprimerCategory()+ rechercheCategory()

Architecture du Système

MVC (Modèle-Vue-Contrôleur) de Laravel. Cela permettra une séparation claire des préoccupations et une maintenance facilitée.



Routes

Authentification

```
Route::middleware('guest')->group(function () {
    Route::get('register', [RegisteredUserController::class, 'create'])
        ->name('register');

    Route::post('register', [RegisteredUserController::class, 'store']);

    Route::get('login', [AuthenticatedSessionController::class, 'create'])
        ->name('login');

    Route::post('login', [AuthenticatedSessionController::class, 'store']);

    Route::get('forgot-password', [PasswordResetLinkController::class, 'create'])
        ->name('password.request');

    Route::post('forgot-password', [PasswordResetLinkController::class, 'store'])
        ->name('password.email');

    Route::get('reset-password/{token}', [NewPasswordController::class, 'create'])
        ->name('password.reset');

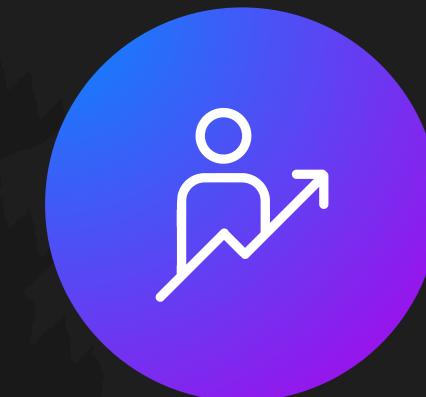
    Route::post('reset-password', [NewPasswordController::class, 'store'])
        ->name('password.store');
});
```

Notes

```
// note
Route::middleware('auth')->group(function () {
    Route::post('/note/store', [NoteController::class, 'store'])->name('note.store');
    Route::get('/note/show/{id}', [NoteController::class, 'show'])->middleware('Category');
    Route::get('/note/destroy/{id}', [NoteController::class, 'destroy'])->middleware('ShowNote');
    Route::get('/note/afficher/{id}', [NoteController::class, 'afficher'])->middleware('ShowNote');
    Route::get('/note/modifier/{id}', [NoteController::class, 'modifier'])->middleware('modifier');
    Route::post('/note/modifier', [NoteController::class, 'modifier_note'])->name('note.modifier');
});
```

Architecture du Système

MVC (Modèle-Vue-Contrôleur) de Laravel. Cela permettra une séparation claire des préoccupations et une maintenance facilitée.



Routes

Catégories

```
//category
Route::middleware('auth')->group(function () {
    Route::post('/category/store',[categoryController::class,'store'])->name('category.store');
    Route::get('/category/destroy/{id}',[categoryController::class,'destroy'])->middleware('Category');
});
```

Architecture du Système

MVC (Modèle-Vue-Contrôleur) de Laravel. Cela permettra une séparation claire des préoccupations et une maintenance facilitée.



Contrôleurs

NoteController

Gère les opérations CRUD pour les notes.

CategoryController

Gère les opérations CRUD pour les catégories.

- ✓ Controllers
- > Auth
 - php categoryController.php
 - php Controller.php
 - php dashboardController.php
 - php HomeController.php
 - php NoteController.php
 - php ProfileController.php

Sécurité

Authentification



Laravel Breeze

Utilisation du système
d'authentification de
Laravel.

Protection des Routes



Middleware pour protéger
les routes sensibles





NoteBlade

MERCI

Pour regarder cette présentation



+0657892108



abdelljabarhamri06@gmail.com



abdelljabar.rf.gd