

# Application de gestion des trésors d'art du Bénin by Malthus AMETPE

---

Cette solution Laravel de Thus permet de gérer les œuvres d'art restituées au Bénin, en distinguant les trésors royaux et les œuvres d'art contemporaines.

## Sommaire

---

1. [Configuration initiale](#)
2. [Structure de la base de données](#)
3. [Migrations](#)
4. [Modèles](#)
5. [Seeders](#)
6. [Contrôleurs](#)
7. [Routes](#)
8. [Vues](#)
9. [JavaScript pour confirmation de suppression](#)

# Configuration initiale

---

## Installation de Laravel

```
composer create-project laravel/laravel tresor-benin "10.*"  
cd tresor-benin
```

## Configuration de la base de données (.env)

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

## Installation de Bootstrap

```
npm install bootstrap
```

Configurez Bootstrap dans `resources/js/app.js` :

```
import 'bootstrap';  
import '../css/app.css';
```

## Structure de la base de données

---

La base de données `tresor` contient trois tables:

- `categories` : stocke les catégories d'œuvres
- `artistes` : enregistre les informations sur les artistes
- `oeuvres` : contient les détails des œuvres avec des clés étrangères vers artistes et catégories

## Migrations

---

### Création des migrations

```
php artisan make:migration create_categories_table  
php artisan make:migration create_artistes_table
```

```
php artisan make:migration create_oeuvres_table
```

## Migration pour les catégories

```
// database/migrations/xxxx_create_categories_table.php
public function up()
{
    Schema::create('categories', function (Blueprint $table) {
        $table->id('idCategorie');
        $table->string('nomCategorie');
        $table->timestamps();
    });
}
```

## Migration pour les artistes

```
// database/migrations/xxxx_create_artistes_table.php
public function up()
{
    Schema::create('artistes', function (Blueprint $table) {
        $table->id('idArtiste');
        $table->string('nom');
    });
}
```

```
        $table->string('prenom');
        $table->string('telephone')->nullable();
        $table->timestamps();
    });
}
```

## Migration pour les œuvres

```
// database/migrations/xxxx_create_oeuvres_table.php
public function up()
{
    Schema::create('oeuvres', function (Blueprint $table) {
        $table->id('idOeuvre');
        $table->string('nom');
        $table->text('description')->nullable();
        $table->integer('annee')->nullable();
        $table->unsignedBigInteger('idArtiste')->nullable();
        $table->unsignedBigInteger('idCategorie');
        $table->timestamps();

        $table->foreign('idArtiste')->references('idArtiste')->on('artistes')->onDelete('set null');
        $table->foreign('idCategorie')->references('idCategorie')->on('categories')->onDelete('cas
```

```
});  
}
```

## Exécution des migrations

```
php artisan migrate
```

## Modèles

---

### Modèle Categorie

```
// app/Models/Categorie.php  
namespace App\Models;  
  
use Illuminate\Database\Eloquent\Factories\HasFactory;  
use Illuminate\Database\Eloquent\Model;  
  
class Categorie extends Model  
{  
    use HasFactory;  
}
```

```
protected $primaryKey = 'idCategorie';
protected $fillable = ['nomCategorie'];

public function oeuvres()
{
    return $this->hasMany(Oeuvre::class, 'idCategorie', 'idCategorie');
}
}
```

## Modèle Artiste

```
// app/Models/Artiste.php
namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Artiste extends Model
{
    use HasFactory;

    protected $primaryKey = 'idArtiste';
    protected $fillable = ['nom', 'prenom', 'telephone'];
}
```

```

public function oeuvres()
{
    return $this->hasMany(Oeuvre::class, 'idArtiste', 'idArtiste');
}

public function nomComplet()
{
    return $this->prenom . ' ' . $this->nom;
}
}

```

## Modèle Oeuvre

```

// app/Models/Oeuvre.php
namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Oeuvre extends Model
{
    use HasFactory;

    protected $primaryKey = 'idOeuvre';
}

```



```
protected $fillable = ['nom', 'description', 'annee', 'idArtiste', 'idCategorie'];

public function categorie()
{
    return $this->belongsTo(Categorie::class, 'idCategorie', 'idCategorie');
}

public function artiste()
{
    return $this->belongsTo(Artiste::class, 'idArtiste', 'idArtiste');
}
}
```

## Seeders

---

### Seeder pour les catégories

```
// database/seeders/CategorieSeeder.php
namespace Database\Seeders;

use App\Models\Categorie;
use Illuminate\Database\Seeder;
```

```
class CategorieSeeder extends Seeder
{
    public function run()
    {
        Categorie::create(['nomCategorie' => 'Trésors royaux']);
        Categorie::create(['nomCategorie' => 'Arts contemporains']);
    }
}
```

## Seeder pour les artistes

```
// database/seeder/ArtisteSeeder.php
namespace Database\Seeders;

use App\Models\Artiste;
use Illuminate\Database\Seeder;

class ArtisteSeeder extends Seeder
{
    public function run()
    {
        Artiste::create([
            'nom' => 'TOKOUDAGBA',
        ]);
    }
}
```

```

        'prenom' => 'Cyprien',
        'telephone' => null
    ]);

    Artiste::create([
        'nom' => 'PEDE',
        'prenom' => 'Apollinaire',
        'telephone' => null
    ]);
}
}

```

## Seeder pour quelques œuvres initiales

```

// database/seeder/OeuvreSeeder.php
namespace Database\Seeders;

use App\Models\Oeuvre;
use App\Models\Artiste;
use App\Models\Categorie;
use Illuminate\Database\Seeder;

class OeuvreSeeder extends Seeder
{

```

```
public function run()
{
    $categorieRoyale = Category::where('nomCategorie', 'Trésors royaux')->first();
    $categorieContemporaine = Category::where('nomCategorie', 'Arts contemporains')->first();
    $artistePede = Artiste::where('nom', 'PEDE')->first();

    Oeuvre::create([
        'nom' => 'Trône d\'apparat du Roi GHEZO',
        'description' => 'Trône royal provenant du palais d\'Abomey',
        'annee' => null,
        'idArtiste' => null,
        'idCategorie' => $categorieRoyale->idCategorie
    ]);

    Oeuvre::create([
        'nom' => 'Ekelodjouti',
        'description' => 'Œuvre contemporaine',
        'annee' => 2018,
        'idArtiste' => $artistePede->idArtiste,
        'idCategorie' => $categorieContemporaine->idCategorie
    ]);
}
}
```

# Mettre à jour le DatabaseSeeder principal

```
// database/seeder/DatabaseSeeder.php
namespace Database\Seeders;

use Illuminate\Database\Seeder;

class DatabaseSeeder extends Seeder
{
    public function run()
    {
        $this->call([
            CategorieSeeder::class,
            ArtisteSeeder::class,
            OeuvreSeeder::class
        ]);
    }
}
```

## Exécution des seeders

```
php artisan db:seed
```

# Contrôleurs

---

## Contrôleur pour les œuvres

```
// app/Http/Controllers/OeuvreController.php
namespace App\Http\Controllers;

use App\Models\Oeuvre;
use App\Models\Artiste;
use App\Models\Categorie;
use Illuminate\Http\Request;

class OeuvreController extends Controller
{
    public function index()
    {
        $oeuvres = Oeuvre::with(['categorie', 'artiste'])->get();
        return view('oeuvres.index', compact('oeuvres'));
    }

    public function create()
    {
        $categories = Categorie::all();
    }
}
```

```
    $artistes = Artiste::all();
    return view('oeuvres.create', compact('categories', 'artistes'));
}

public function store(Request $request)
{
    $validatedData = $request->validate([
        'nom' => 'required|string|max:255',
        'description' => 'nullable|string',
        'annee' => 'nullable|integer',
        'idArtiste' => 'nullable|exists:artistes,idArtiste',
        'idCategorie' => 'required|exists:categories,idCategorie',
    ]);

    Oeuvre::create($validatedData);

    return redirect()->route('oeuvres.index')->with('success', 'Œuvre ajoutée avec succès');
}

public function edit(Oeuvre $oeuvre)
{
    $categories = Categorie::all();
    $artistes = Artiste::all();
    return view('oeuvres.edit', compact('oeuvre', 'categories', 'artistes'));
}
```

```
public function update(Request $request, Oeuvre $oeuvre)
{
    $validatedData = $request->validate([
        'nom' => 'required|string|max:255',
        'description' => 'nullable|string',
        'annee' => 'nullable|integer',
        'idArtiste' => 'nullable|exists:artistes,idArtiste',
        'idCategorie' => 'required|exists:categories,idCategorie',
    ]);

    $oeuvre->update($validatedData);

    return redirect()->route('oeuvres.index')->with('success', 'Œuvre mise à jour avec succès');
}

public function destroy(Oeuvre $oeuvre)
{
    $oeuvre->delete();
    return redirect()->route('oeuvres.index')->with('success', 'Œuvre supprimée avec succès');
}
}
```

## Routes

---



```
// routes/web.php
use App\Http\Controllers\OeuvreController;
use Illuminate\Support\Facades\Route;

Route::get('/', function () {
    return redirect()->route('oeuvres.index');
});

Route::get('/oeuvres', [OeuvreController::class, 'index']->name('oeuvres.index'));
Route::get('/oeuvres/create', [OeuvreController::class, 'create']->name('oeuvres.create'));
Route::post('/oeuvres', [OeuvreController::class, 'store']->name('oeuvres.store'));
Route::get('/oeuvres/{oeuvre}/edit', [OeuvreController::class, 'edit']->name('oeuvres.edit'));
Route::put('/oeuvres/{oeuvre}', [OeuvreController::class, 'update']->name('oeuvres.update'));
Route::delete('/oeuvres/{oeuvre}', [OeuvreController::class, 'destroy']->name('oeuvres.destroy'));
```

## Vues

---

### Layout principal avec Bootstrap

```
<!-- resources/views/layouts/app.blade.php -->
<!DOCTYPE html>
```

```
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Trésors du Bénin</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="styl
</head>
<body>
  <nav class="navbar navbar-expand-lg navbar-dark bg-dark mb-4">
    <div class="container">
      <a class="navbar-brand" href="{{ route('oeuvres.index') }}">Trésors du Bénin</a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target=
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav">
          <li class="nav-item">
            <a class="nav-link" href="{{ route('oeuvres.index') }}">Liste des œuvres</
          </li>
          <li class="nav-item">
            <a class="nav-link" href="{{ route('oeuvres.create') }}">Ajouter une œuvre
          </li>
        </ul>
      </div>
    </div>
  </nav>
```

```
<div class="container">
    @if(session('success'))
        <div class="alert alert-success">
            {{ session('success') }}
        </div>
    @endif

    @yield('content')
</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></s
@yield('scripts')
</body>
</html>
```

## Page index.php - Liste des œuvres

```
<!-- resources/views/oeuvres/index.blade.php -->
@extends('layouts.app')

@section('content')
    <h1>Liste des œuvres</h1>
    <a href="{{ route('oeuvres.create') }}" class="btn btn-primary mb-3">Ajouter une œuvre</a>
```

```

<table class="table table-striped">
  <thead>
    <tr>
      <th>Œuvre</th>
      <th>Auteur</th>
      <th>Année</th>
      <th>Catégorie</th>
      <th>Actions</th>
    </tr>
  </thead>
  <tbody>
    @foreach($oeuvres as $oeuvre)
      <tr>
        <td>{{ $oeuvre->nom }}</td>
        <td>{{ $oeuvre->artiste ? $oeuvre->artiste->nomComplet() : 'Non spécifié' }}</td>
        <td>{{ $oeuvre->annee ?: 'Non spécifié' }}</td>
        <td>{{ $oeuvre->categorie->nomCategorie }}</td>
        <td>
          <div class="btn-group">
            <a href="{{ route('oeuvres.edit', $oeuvre) }}" class="btn btn-sm btn-w
            <form action="{{ route('oeuvres.destroy', $oeuvre) }}" method="POST" c
              @csrf
              @method('DELETE')
              <button type="submit" class="btn btn-sm btn-danger">Supprimer</but
            </form>
          </td>
        </tr>
      @endforeach
    </tbody>
  </table>

```

```
        </div>
    </td>
</tr>
@endforeach
</tbody>
</table>
@endsection

@section('scripts')
<script>
    document.addEventListener('DOMContentLoaded', function() {
        const deleteForms = document.querySelectorAll('.delete-form');
        deleteForms.forEach(form => {
            form.addEventListener('submit', function(e) {
                e.preventDefault();
                if (confirm('Êtes-vous sûr de vouloir supprimer cette œuvre ?')) {
                    this.submit();
                }
            });
        });
    });
</script>
@endsection
```

## Page create.php - Ajouter une œuvre

```
<!-- resources/views/oeuvres/create.blade.php -->
@extends('layouts.app')

@section('content')
    <h1>Ajouter une œuvre</h1>

    <form action="{{ route('oeuvres.store') }}" method="POST">
        @csrf

        <div class="mb-3">
            <label for="nom" class="form-label">Nom de l'œuvre</label>
            <input type="text" class="form-control @error('nom') is-invalid @enderror" id="nom" name="nom" @error('nom')
                <div class="invalid-feedback">{{ $message }}</div>
            @enderror
        </div>

        <div class="mb-3">
            <label for="description" class="form-label">Description</label>
            <textarea class="form-control @error('description') is-invalid @enderror" id="description" name="description" @error('description')
                <div class="invalid-feedback">{{ $message }}</div>
            @enderror
        </div>
    </form>
</div>
```

```
</div>
```

```
<div class="mb-3">
```

```
  <label for="annee" class="form-label">Année</label>
```

```
  <input type="number" class="form-control @error('annee') is-invalid @enderror" id="ann
```

```
  <small class="form-text text-muted">Laisser vide pour les trésors royaux sans année co  
  @error('annee')
```

```
    <div class="invalid-feedback">{{ $message }}</div>
```

```
  @enderror
```

```
</div>
```

```
<div class="mb-3">
```

```
  <label for="idCategorie" class="form-label">Catégorie</label>
```

```
  <select class="form-select @error('idCategorie') is-invalid @enderror" id="idCategorie
```

```
    <option value="">Sélectionner une catégorie</option>
```

```
    @foreach($categories as $categorie)
```

```
      <option value="{{ $categorie->idCategorie }}" {{ old('idCategorie') == $catego
```

```
        {{ $categorie->nomCategorie }}>
```

```
      </option>
```

```
    @endforeach
```

```
  </select>
```

```
  @error('idCategorie')
```

```
    <div class="invalid-feedback">{{ $message }}</div>
```

```
  @enderror
```

```
</div>
```

```

<div class="mb-3">
    <label for="idArtiste" class="form-label">Artiste</label>
    <select class="form-select @error('idArtiste') is-invalid @enderror" id="idArtiste" na
        <option value="">Non spécifié</option>
        @foreach($artistes as $artiste)
            <option value="{{ $artiste->idArtiste }}" {{ old('idArtiste') == $artiste->idA
                {{ $artiste->nomComplet() }}
            </option>
        @endforeach
    </select>
    <small class="form-text text-muted">Laisser vide pour les trésors royaux sans artiste
    @error('idArtiste')
        <div class="invalid-feedback">{{ $message }}</div>
    @enderror
</div>

<button type="submit" class="btn btn-primary">Enregistrer</button>
<a href="{{ route('oeuvres.index') }}" class="btn btn-secondary">Annuler</a>
</form>
@endsection

```

Page edit.php - Modifier une œuvre



```
<!-- resources/views/oeuvres/edit.blade.php -->
@extends('layouts.app')

@section('content')
    <h1>Modifier l'œuvre</h1>

    <form action="{{ route('oeuvres.update', $oeuvre) }}" method="POST">
        @csrf
        @method('PUT')

        <div class="mb-3">
            <label for="nom" class="form-label">Nom de l'œuvre</label>
            <input type="text" class="form-control @error('nom') is-invalid @enderror" id="nom" name="nom"
            @error('nom')
                <div class="invalid-feedback">{{ $message }}</div>
            @enderror
        </div>

        <div class="mb-3">
            <label for="description" class="form-label">Description</label>
            <textarea class="form-control @error('description') is-invalid @enderror" id="description" name="description"
            @error('description')
                <div class="invalid-feedback">{{ $message }}</div>
            @enderror
        </div>
    </form>
</div>
```

```
<div class="mb-3">
  <label for="annee" class="form-label">Année</label>
  <input type="number" class="form-control @error('annee') is-invalid @enderror" id="annee" value="{{ old('annee') }}"
  <small class="form-text text-muted">Laisser vide pour les trésors royaux sans année co
  @error('annee')
    <div class="invalid-feedback">{{ $message }}</div>
  @enderror
</div>
```

```
<div class="mb-3">
  <label for="idCategorie" class="form-label">Catégorie</label>
  <select class="form-select @error('idCategorie') is-invalid @enderror" id="idCategorie" value="{{ old('idCategorie') }}"
    <option value="">Sélectionner une catégorie</option>
    @foreach($categories as $categorie)
      <option value="{{ $categorie->idCategorie }}" {{ old('idCategorie', $oeuvre->idCategorie) == $categorie->idCategorie ? 'selected' : '' }}>
        {{ $categorie->nomCategorie }}
      </option>
    @endforeach
  </select>
  @error('idCategorie')
    <div class="invalid-feedback">{{ $message }}</div>
  @enderror
</div>
```

```
<div class="mb-3">
```

```

<label for="idArtiste" class="form-label">Artiste</label>
<select class="form-select @error('idArtiste') is-invalid @enderror" id="idArtiste" na
    <option value="">Non spécifié</option>
    @foreach($artistes as $artiste)
        <option value="{{ $artiste->idArtiste }}" {{ old('idArtiste', $oeuvre->idArtis
            {{ $artiste->nomComplet() }}
        </option>
    @endforeach
</select>
<small class="form-text text-muted">Laisser vide pour les trésors royaux sans artiste
    @error('idArtiste')
        <div class="invalid-feedback">{{ $message }}</div>
    @enderror
</div>

<button type="submit" class="btn btn-primary">Mettre à jour</button>
<a href="{{ route('oeuvres.index') }}" class="btn btn-secondary">Annuler</a>
</form>
@endsection

```

## JavaScript pour confirmation de suppression

---

Le JavaScript pour la confirmation de suppression est déjà inclus dans la vue index.blade.php.

# Instructions d'installation et d'utilisation

---

1. Cloner le projet
2. Exécuter `composer install`
3. Configurer le fichier .env pour la base de données
4. Exécuter `php artisan key:generate`
5. Exécuter `php artisan migrate --seed`
6. Exécuter `npm install && npm run dev`
7. Démarrer le serveur avec `php artisan serve`
8. Accéder à l'application via <http://localhost:8000>