
Rapport de Développement

Étude de Cas

Nicolas WEINKOPF, Daryll SINDZE, Rémy JEROME - 5 avril 2016

Projet Clique à BAYANGAM

Sommaire

1. État du projet	3
1.1.État actuel de l'avancement du projet.....	3
1.2.Design graphique du site web.....	3
2. Implémentation de la base de données.....	5
2.1.Entités	5
2.2.Les vues	11

1. État du projet

1.1.État actuel de l'avancement du projet

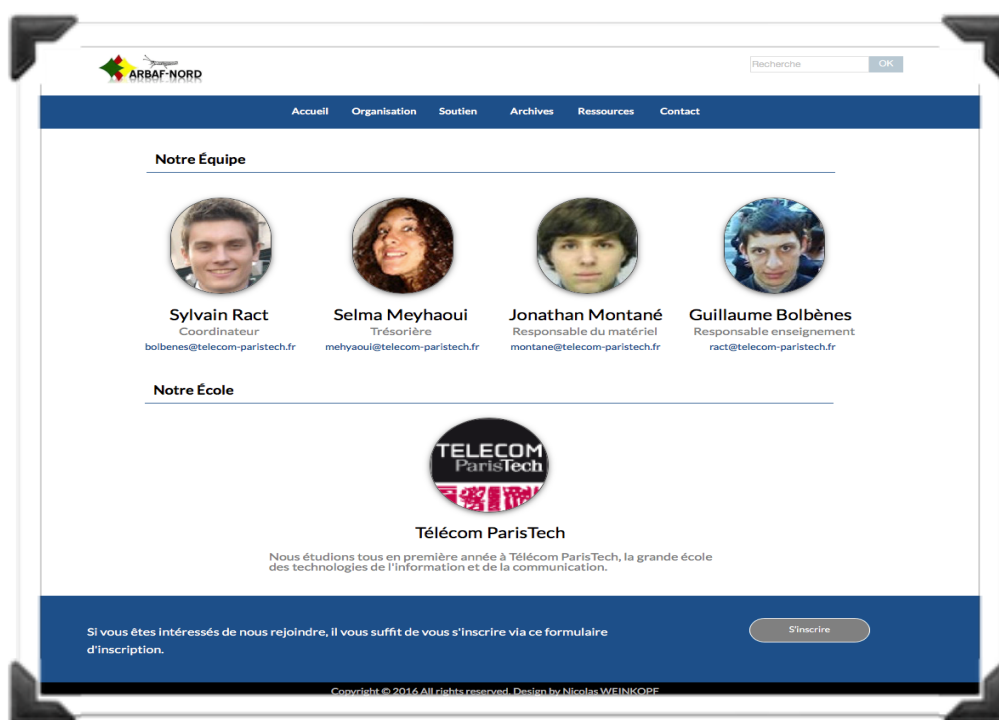
Actuellement, le site web est en cours de développement. En effet, nous avons fini la phase de conception et la réalisation de la maquette.

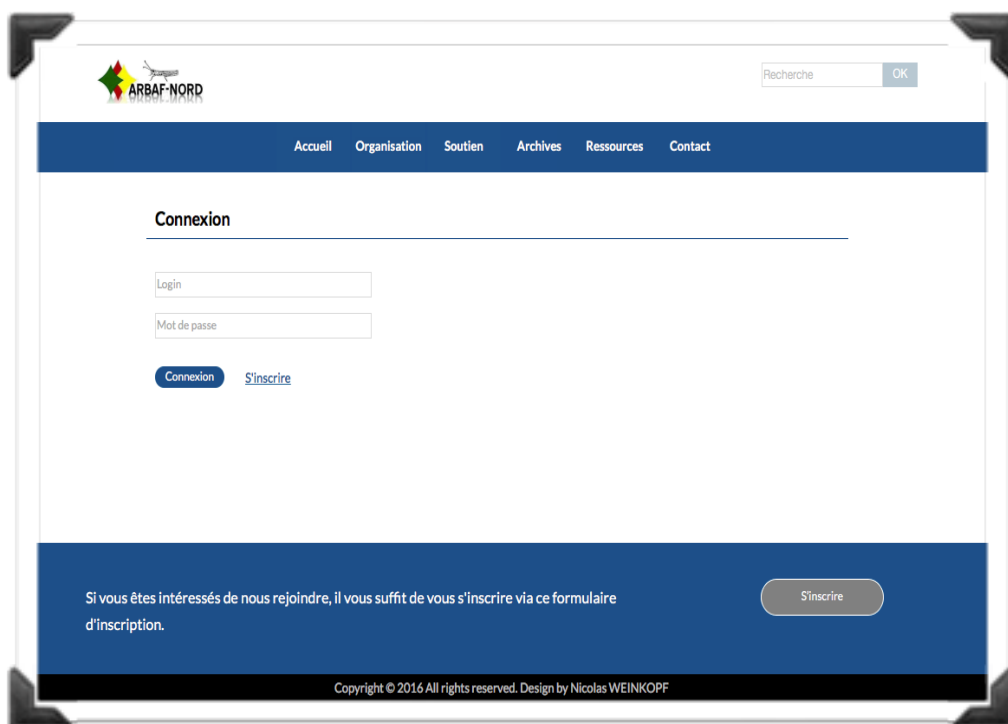
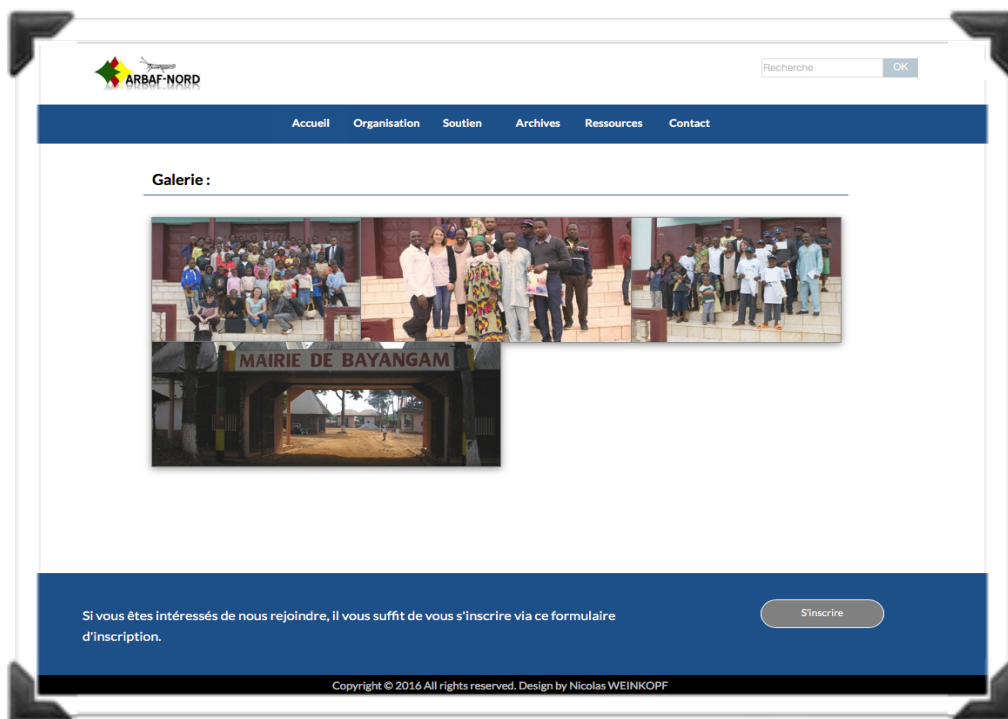
Nous avons implémenté la couche métier de notre application web, ceci nous permet de manipuler les informations de la base de données afin d'avoir un site web dynamique à l'aide de Symfony et de l'Ajax.

1.2.Design graphique du site web

Nous avons implémenter le graphique du site web en utilisant le framework Bootstrap, disponible sur Github <https://github.com/twbs/bootstrap>. Il est utilisé pour des grands sites web comme Twitter.

Le site web est design responsive. En effet, Bootstrap permet d'avoir une optimisation du design avec la majorité des navigateurs Internet.





2. Implémentation de la base de données

2.1. Entités

Les classe php sont appelées « entité ». Ce sont des classes basiques contenant des données et remplissent les besoins métiers de notre application. Ces classes seront par la suite persistées en base de données dans le code de notre application.

Articles :

```
<?php

namespace wp\AssociationBundle\Entity;

use Doctrine\ORM\Mapping as ORM;

class Articles
{
    private $id;
    private $auteur;
    private $titre;
    private $article;
    private $dateAjout;
    private $dateEvenement;
    public function getId()
    {
        return $this->id;
    }

    public function setAuteur($auteur)
    {
        $this->auteur = $auteur;

        return $this;
    }

    public function getAuteur()
    {
        return $this->auteur;
    }

    public function setTitre($titre)
    {
        $this->titre = $titre;

        return $this;
    }
}
```

```

public function getTitre()
{
    return $this->titre;
}

public function setArticle($article)
{
    $this->article = $article;

    return $this;
}

public function getArticle()
{
    return $this->article;
}

public function setDateAjout($dateAjout)
{
    $this->dateAjout = $dateAjout;

    return $this;
}

public function getDateAjout()
{
    return $this->dateAjout;
}

public function setDateEvenement($dateEvenement)
{
    $this->dateEvenement = $dateEvenement;

    return $this;
}

public function getDateEvenement()
{
    return $this->dateEvenement;
}
}

```

Blog :

```

<?php

namespace wp\AssociationBundle\Entity;

```

```
use Doctrine\ORM\Mapping as ORM;

class Blog
{
    private $id;
    private $titre;
    private $auteur;
    private $article;
    private $dateAjout;
    private $dateEvenement;
    private $photo;
    private $video;
    public function getId()
    {
        return $this->id;
    }

    public function setTitre($titre)
    {
        $this->titre = $titre;

        return $this;
    }

    public function getTitre()
    {
        return $this->titre;
    }

    public function setAuteur($auteur)
    {
        $this->auteur = $auteur;

        return $this;
    }

    public function getAuteur()
    {
        return $this->auteur;
    }

    public function setArticle($article)
    {
        $this->article = $article;

        return $this;
    }

    public function getArticle()
```

```

{
    return $this->article;
}

public function setDateAjout($dateAjout)
{
    $this->dateAjout = $dateAjout;

    return $this;
}

public function getDateAjout()
{
    return $this->dateAjout;
}

public function setDateEvenement($dateEvenement)
{
    $this->dateEvenement = $dateEvenement;

    return $this;
}

public function getDateEvenement()
{
    return $this->dateEvenement;
}

public function setPhoto($photo)
{
    $this->photo = $photo;

    return $this;
}

public function getPhoto()
{
    return $this->photo;
}

public function setVideo($video)
{
    $this->video = $video;

    return $this;
}

public function getVideo()
{
    return $this->video;
}

```



```
}  
}
```

TimeLine :

```
<?php  
  
namespace wp\AssociationBundle\Entity;  
  
use Doctrine\ORM\Mapping as ORM;  
  
class TimeLine  
{  
    private $id;  
    private $annee;  
    private $commentaire;  
    public function getId()  
    {  
        return $this->id;  
    }  
  
    public function setAnnee($annee)  
    {  
        $this->annee = $annee;  
  
        return $this;  
    }  
  
    public function getAnnee()  
    {  
        return $this->annee;  
    }  
  
    public function setCommentaire($commentaire)  
    {  
        $this->commentaire = $commentaire;  
  
        return $this;  
    }  
  
    public function getCommentaire()  
    {  
        return $this->commentaire;  
    }  
}
```

Auteur :

```
<?php

namespace wp\AssociationBundle\Entity;

use Doctrine\ORM\Mapping as ORM;

class Auteur
{
    private $id;
    private $nom;
    private $prenom;

    public function getId()
    {
        return $this->id;
    }

    public function setNom($nom)
    {
        $this->nom = $nom;

        return $this;
    }

    public function getNom()
    {
        return $this->nom;
    }

    public function setPrenom($prenom)
    {
        $this->prenom = $prenom;

        return $this;
    }

    public function getPrenom()
    {
        return $this->prenom;
    }
}
```

2.2.Les vues

La base de donnée de Symfony est gérée à l'aide de Doctrine. Pour que Doctrine soit capable de faire ça, nous ajoutons des « métadonnées », ou configurations expliquant à Doctrine comment les classes et leurs propriétés doivent être implémentée avec la base de données. Nous avons choisi le format de base fourni par Symfony avec les annotations.

```
--
-- Structure de la table `articles`
--

CREATE TABLE `articles` (
  `id` int(11) NOT NULL,
  `Auteur` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
  `Titre` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
  `Article` longtext COLLATE utf8_unicode_ci NOT NULL,
  `DateAjout` datetime NOT NULL,
  `DateEvenement` date NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

-----

--
-- Structure de la table `auteur`
--

CREATE TABLE `auteur` (
  `id` int(11) NOT NULL,
  `Nom` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
  `Prenom` varchar(255) COLLATE utf8_unicode_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

-----

--
-- Structure de la table `blog`
--

CREATE TABLE `blog` (
  `id` int(11) NOT NULL,
  `Titre` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
  `Auteur` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
  `Article` longtext COLLATE utf8_unicode_ci NOT NULL,
  `DateAjout` datetime NOT NULL,
  `DateEvenement` date NOT NULL,
  `Photo` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
  `Video` varchar(255) COLLATE utf8_unicode_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

```
--  
-- Structure de la table `time_line`  
--  
  
CREATE TABLE `time_line` (  
  `id` int(11) NOT NULL,  
  `Annee` date NOT NULL,  
  `Commentaire` varchar(255) COLLATE utf8_unicode_ci NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```