

Projet réalisé par :
Guillaume BONNET
Romain MUNOS

PROJET ISN PARDUS

Sous la direction de :
M. Turro

INTRODUCTION

QUEL EST LE BUT DU PROJET ?
POURQUOI EXISTE-IL ?
À QUI S'ADRESSE-T-IL ?

PRÉSENTATION DU PROJET

LANGAGES UTILISÉS



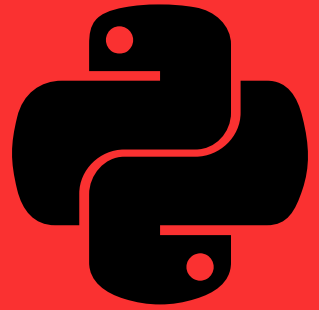
HTML 5



CSS 3



JavaScript



Python 3

BIBLIOTHÈQUES UTILISÉES



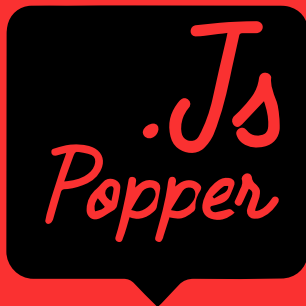
CSS 3



JavaScript



Jquery



PopperJS

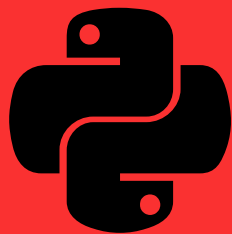


Bootstrap



Font Awesome

BIBLIOTHÈQUES UTILISÉES



Python 3



PILLOW

CAHIER DES CHARGES

Toutes les pages :

- En-tête :
 - Titre du site
 - Slogan du site
- Barre de navigation
 - Liens
 - Barre de recherche
- Pied de page
 - Titre du site
 - Slogan
 - Liens

Visiteur :

- Accueil :
 - Prévisualisation d'articles
- Catégories
 - Liste des catégories
- Article
 - Contenu de l'article
 - Détails sur l'article
 - Espace commentaire
- Contact

Utilisateur :

- Espace membre :
 - Inscription
 - Connexion
 - Paramètres
- Profil
 - Informations
 - Articles publiés
- + Pages accessibles au visiteur

Administrateur : Accès à l'administration (Gestion des articles, catégories, commentaires et messages via le formulaire de contact)

RÉPARTITION DES TÂCHES

NOMS	TÂCHES
Guillaume BONNET	Modèle HTML & CSS de la page d'accueil, des catégories, du filtre d'article par catégorie/date/recherche, du profil d'utilisateur
	JavaScript propre au projet
	Gestion des modèles Django
Romain MUNOS	Modèle HTML & CSS de la page d'article, de contact, de connexion, d'inscription et des paramètres utilisateurs
	Gestion des formulaires Django
Commun	Gestion de l'administration, des vues et de la distributions des URLs Django

CODE : LES MODÈLES

```
105
106 class Article(models.Model):
107     """
108     |   Modèle pour les articles
109     """
110
111     title = models.CharField(max_length=200, verbose_name="Titre")
112     slug = models.SlugField(max_length=200, verbose_name="Slug")
113     author = models.ForeignKey(User, on_delete=models.CASCADE)
114     date = models.DateTimeField(auto_now_add=True)
115     categories = models.ManyToManyField(Category)
116     image = models.ImageField(upload_to="article_images")
117     content = models.TextField(verbose_name="Contenu")
118
119     def __str__(self):
120         return self.title
121
122     def save(self, *args, **kwargs):
123         """
124         |   Surcharge de la méthode "save" :
125         |
126         |   Créé un slug (texte identificateur) en fonction du titre
127         |
128         |   Documentation Django sur la surcharge de la méthode save :
129         |   https://docs.djangoproject.com/fr/2.0/topics/db/models/#overriding-save-method
130         """
131
132         if not self.slug:
133             self.slug = slugify(self.title)
134         super().save()
135
136     class Meta:
137         verbose_name = "article"
138         ordering = ["-date"]
139
140
141 class Comment(models.Model):
142     """
143     |   Modèle pour les commentaires
144     """
```

À QUOI SERVENT-ILS ?

EXEMPLE DE MODÈLE : ARTICLE

```
106 class Article(models.Model):
107     """
108     |   Modèle pour les articles
109     """
110
111     title = models.CharField(max_length=200, verbose_name="Titre")
112     slug = models.SlugField(max_length=200, verbose_name="Slug", blank=True)
113     author = models.ForeignKey(User, on_delete=models.CASCADE, verbose_name="Auteur")
114     date = models.DateTimeField(auto_now_add=True, verbose_name="Date de création")
115     categories = models.ManyToManyField(Category, verbose_name="Catégories")
116     image = models.ImageField(upload_to=article_image, verbose_name="Image de fond", blank=True)
117     content = models.TextField(verbose_name="Contenu")
118
119     def __str__(self):
120         return self.title
121
122     def save(self, *args, **kwargs):
123         """
124         |   Surcharge de la méthode "save" :
125         |
126         |   Créé un slug (texte identificateur) en fonction du titre si ce dernier n'est pas déjà défini
127         |
128         |   Documentation Django sur la surcharge des méthodes de modèles :
129         |   https://docs.djangoproject.com/fr/2.0/topics/db/models/#overriding-model-methods
130         """
131
132         if not self.slug:
133             self.slug = slugify(self.title)
134         super().save()
135
136     class Meta:
137         verbose_name = "article"
138         ordering = ["-date"]
```

EXEMPLE DE MODÈLE : ARTICLE

```
title = models.CharField(max_length=200, verbose_name="Titre")
slug = models.SlugField(max_length=200, verbose_name="Slug", blank=True)
author = models.ForeignKey(User, on_delete=models.CASCADE, verbose_name="Auteur")
date = models.DateTimeField(auto_now_add=True, verbose_name="Date de création")
categories = models.ManyToManyField(Category, verbose_name="Catégories")
image = models.ImageField(upload_to=article_image, verbose_name="Image de fond", blank=True)
content = models.TextField(verbose_name="Contenu")
```

EXEMPLE DE MODÈLE : ARTICLE

```
def article_image(instance, file_name):
    """
    Fonction pour générer le chemin des images des articles

    Récupère l'instance et affecte la valeur à une variable avec la propriété "slug" de l'instance si elle est disponible,
    sinon affecte un UUID généré, puis retourne la variable
    """

    upload_to = "articles/"
    # Récupère l'extension du fichier (JPG, PNG, ...)
    file_type = file_name.split(".")[1]
    if instance.title:
        file_name = "{}.{}".format(instance.title, file_type)
    else:
        file_name = "{}.{}".format(uuid4().hex, file_type)
    # Défini le chemin complet depuis la racine du projet
    file_path = "uploads/" + upload_to + file_name
    # Vérifie si un fichier existe déjà et si oui, le supprime
    if os.path.isfile(file_path):
        os.remove(file_path)
    return os.path.join(upload_to, file_name)
```

EXEMPLE DE MODÈLE : ARTICLE

```
def save(self, *args, **kwargs):  
    """  
    Surcharge de la méthode "save" :  
  
    Créé un slug (texte identificateur) en fonction du titre si ce dernier n'est pas déjà défini  
  
    Documentation Django sur la surcharge des méthodes de modèles :  
    https://docs.djangoproject.com/fr/2.0/topics/db/models/#overriding-model-methods  
    """  
  
    if not self.slug:  
        self.slug = slugify(self.title)  
    super().save()
```

EXEMPLE DE MODÈLE : ARTICLE

```
def __str__(self):  
    return self.title
```

```
class Meta:  
    verbose_name = "article"  
    ordering = ["-date"]
```


EXEMPLE DE MODÈLE : ARTICLE

```
106 class Article(models.Model):
107     """
108     |   Modèle pour les articles
109     """
110
111     title = models.CharField(max_length=200, verbose_name="Titre")
112     slug = models.SlugField(max_length=200, verbose_name="Slug", blank=True)
113     author = models.ForeignKey(User, on_delete=models.CASCADE, verbose_name="Auteur")
114     date = models.DateTimeField(auto_now_add=True, verbose_name="Date de création")
115     categories = models.ManyToManyField(Category, verbose_name="Catégories")
116     image = models.ImageField(upload_to=article_image, verbose_name="Image de fond", blank=True)
117     content = models.TextField(verbose_name="Contenu")
118
119     def __str__(self):
120         return self.title
121
122     def save(self, *args, **kwargs):
123         """
124         |   Surcharge de la méthode "save" :
125         |
126         |   Créé un slug (texte identificateur) en fonction du titre si ce dernier n'est pas déjà défini
127         |
128         |   Documentation Django sur la surcharge des méthodes de modèles :
129         |   https://docs.djangoproject.com/fr/2.0/topics/db/models/#overriding-model-methods
130         """
131
132         if not self.slug:
133             self.slug = slugify(self.title)
134         super().save()
135
136     class Meta:
137         verbose_name = "article"
138         ordering = ["-date"]
```

CONCLUSION

POINTS POSITIFS
POINTS À AMÉLIORER
APPLICATION FLEXIBLE

**MERCI POUR
VOTRE ÉCOUTE**