



# I1 - ISCC

---

I-SCC-000

## ISCC PROJET 1

---

MyDevBlog



## SOMMAIRE

- DOSSIER DE TRAVAIL
- OBJECTIF
- CONSIGNES
- LES PAGES
  - FRONT-OFFICE
    - LES PAGES DU FRONT-OFFICE
  - BACK-OFFICE
    - LES PAGES DU BACK-OFFICE
- BASE DE DONEES
  - ARTICLES
  - UTILISATEURS
- MOCK-UPS

## DOSSIER DE TRAVAIL

---

Si vous n'avez pas préparé le repos de travail "ISCC-2020-MyDevBlog" pendant le Jour 01.

- Créez un nouveau repos "ISCC-2020-MyDevBlog" dans votre dossier "ISCC-2020".
  - Publiez le repos "ISCC-2020-MyDevBlog" sur GitHub.
  - **Invitez** les membres de votre équipe pédagogique de l'ISCC à collaborer avec vous sur votre repos "ISCC-2020-MyDevBlog".
- C'est dans ce dossier, que vous publierez vos travaux. N'oubliez pas d'envoyer régulièrement sur GitHub votre travail. Ceci pourra vous servir de sauvegarde en cas d'erreur ou de perte de données.

## OBJECTIF

---

L'objectif de ce premier projet est de mettre en application ce que vous aurez appris pendant les trois premières semaines de l'ISCC.

Durant le début de l'ISCC vous allez prendre en main, petit à petit, le HTML, le CSS, le PHP et le SQL. Chacun de ces langage est utile mais c'est bien l'alliance de ces langages qui vous permet de créer des sites internet dynamiques.

Pour ce premier projet vous allez créer un DevBlog.

Un DevBlog est généralement un blog très simple tenue par une équipe de développement pour partager l'avancement d'un projet ou pour partager les techniques et technologies qu'ils utilisent durant leur travail. C'est une pratique très courante dans le jeux vidéo par exemple, mais aussi chez la plupart des grosses entreprises d'informatique.



Pour ceux qui ne savent pas ce qu'est un DevBlog voici deux parfaits exemples : <https://developers.googleblog.com/> et <https://devblogs.microsoft.com/>

## CONSIGNES

---

Vous devez développer un site dynamique en utilisant les technologies HTML, CSS, PHP et SQL.

Le site que vous développez devra être votre DevBlog.

Ce site devra avoir une interface permettant d'ajouter des articles.

La page principale de ce DevBlog devra présenter rapidement l'équipe et le projet / entreprise qui est lié, ainsi que les derniers articles publiés.

Vous devrez permettre à l'utilisateur d'accéder la liste de tous les articles publier et de les consulter un à un.

Le site devra permettre à plusieurs utilisateur de se connecter et de publier des articles.



## LES PAGES

---

Votre DevBlog sera composé de 2 ensembles de pages :

- Les pages accessibles aux utilisateurs que nous appellerons le front-office.
- Les pages accessibles seulement aux administrateurs et auteurs du site que nous appellerons le back-office.

### FRONT-OFFICE

---

Le front-office est la partie du site accessible à tous. Le but d'un DevBlog étant de partager à tous un ensemble d'article, ce front-office regroupera l'ensemble des pages permettant de lister et afficher les articles. Toutes les pages du front-office doit contenir les éléments suivants :

- Un header contenant le titre du site.
- Une barre de navigation permettant l'accès à la page d'accueil, la liste des articles la page de contact.
- Un footer contenant les liens vers les autres sites que vous avez créés, votre profil gitHub et vos réseaux professionnels. Le footer doit également contenir un lien vers l'espace administration de votre site.

Le fichier "front.php" doit servir pour le routing du front-office un paramètre URL doit être utilisé pour la navigation entre les différentes pages.

### LES PAGES DU FRONT-OFFICE

La page d'**accueil** doit contenir :

- Une courte description de votre projet
- Liste les 5 derniers articles créés.

La page **articles** doit contenir :

- La liste de tous les articles.

Les pages **article** doit contenir

- l'affichage d'un article.

La page **contact** doit contenir :

- un formulaire de contact avec les champs suivants :
  - nom - prénom
  - e-mail
  - message

## BACK-OFFICE

Le back-office regroupe l'ensemble des pages permettant aux administrateurs d'ajouter du contenu au site et de contrôler l'accès au back-office

Toutes les pages du back-office doivent contenir les éléments suivants :

- Un header contenant le titre du site
- Une barre de navigation permettant l'accès à la page d'ajout d'articles, la liste des utilisateurs.

Le fichier "back.php" doit servir pour le routing du front-office un paramètre URL doit être utilisé pour la navigation entre les différentes pages.



L'accès au back office doit être limité aux utilisateurs connectés. Si un utilisateur n'est pas identifié, il doit être redirigé vers la page de connexion. La liste des utilisateurs doit être stocké en base de données.

## LES PAGES DU BACK-OFFICE

La page de **connexion** au back-office doit contenir :

- Un formulaire avec les champs suivants :
  - Login
  - Mot de passe

La page **ajout article** doit contenir :

- Un formulaire permettant d'ajouter tous les éléments nécessaires à la création d'un article.

La page **ajout utilisateur** doit contenir :

- Un formulaire permettant d'ajouter tous les éléments nécessaires à la création d'un utilisateur.

La page **utilisateurs** doit contenir :

- La liste de tous les utilisateurs et un bouton par utilisateur pour supprimer le compte utilisateur.



## **BASE DE DONEES**

---

### **ARTICLES**

---

Les articles doivent être stockés en base de données.

Un article est composé des éléments suivants :

- Un titre
- Une image
- Une date de publication
- Un auteur (doit être l'utilisateur qui crée l'article)
- Un contenu texte
- Un extrait de 300 caractères (l'extrait doit notamment être utilisé dans la page d'accueil)

### **UTILISATEURS**

---

Les utilisateur doivent être stockés en base de données.

Un utilisateur est composé des éléments suivants :

- Nom d'utilisateur
- Login
- Mot de passe

## **MOCK-UPS**

---

Des exemples visuels des pages du front-office vous sont fournis pour servir de référence :

- Page **Accueil**
- Page **Articles**
- Page **Article**

Le back-office étant accessible uniquement aux administrateurs vous êtes libre de le designer librement.