|  |  |
| --- | --- |
| WEBA – Test 2 | |
| Nom et prénom |  |

|  |  |
| --- | --- |
| Modalités | |
| Type de travail | Individuel |
| Durée | 2 périodes, 1h30 |
| Fichiers sources | Fichier db.sql  Dossier « sources » |
| À rendre | Dans le répertoire : Teams/ETML\_SIG-21-23\_Teams/Fichiers/Gander/TE2/<votre prénom> <votre nom de famille>/ :   * Votre code source final   À imprimer : la première page de ce document, complétée avec votre nom et prénom |
| Moyens d’aide | Tout, sauf vos collègues |
| Nombre total de points | 40 |

|  |  |
| --- | --- |
| Résultat | |
| Points obtenus |  |
| Note |  |

# Contexte

Le stagiaire de votre entreprise a commencé le code source d’un blog. Vous êtes chargé de le terminer.

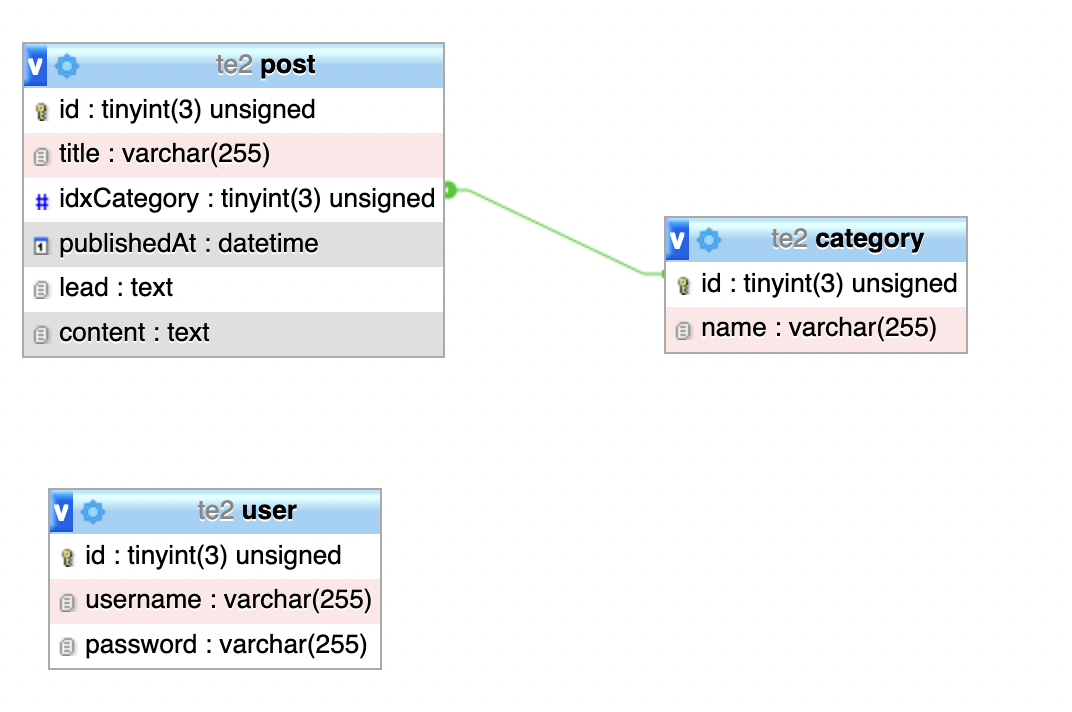
Le blog devra permettre de lister les articles (posts en anglais) et afficher les détails d’un article. Un menu permettra d’ajouter un article.

Note : En temps normal, l’ajout d’un article devrait être protégé par un mot de passe. Dans le cadre de ce test, nous n’allons pas le faire.

**Base de données**

Récupérez les fichiers sources. Exécutez le script SQL pour initialiser la base de données et quelques données. Vous n’aurez pas besoin de modifier la base de données dans le cadre de ce test.

La base de données est composée des tables suivantes représentant respectivement, un article (post) avec id, titre, lien vers une catégorie (clé étrangère), date et heure de publication, texte court (lead) et texte long (content) ; et une catégorie avec id et nom.



**Code source**

Le code source initial fourni suit la structure MVC. Le fichier *index.php* représente le « routeur » et est chargé d’appeler le contrôleur en fonction de l’URL.

Pour l’ensemble de ce test, il est demandé de respecter la structure MVC fournie initialement.

Note : Si vous affichez la page d’accueil du site après avoir importé les fichiers sources, vous aurez un message d’erreur. Modifiez les accès à votre base de données dans le fichier Database.php.

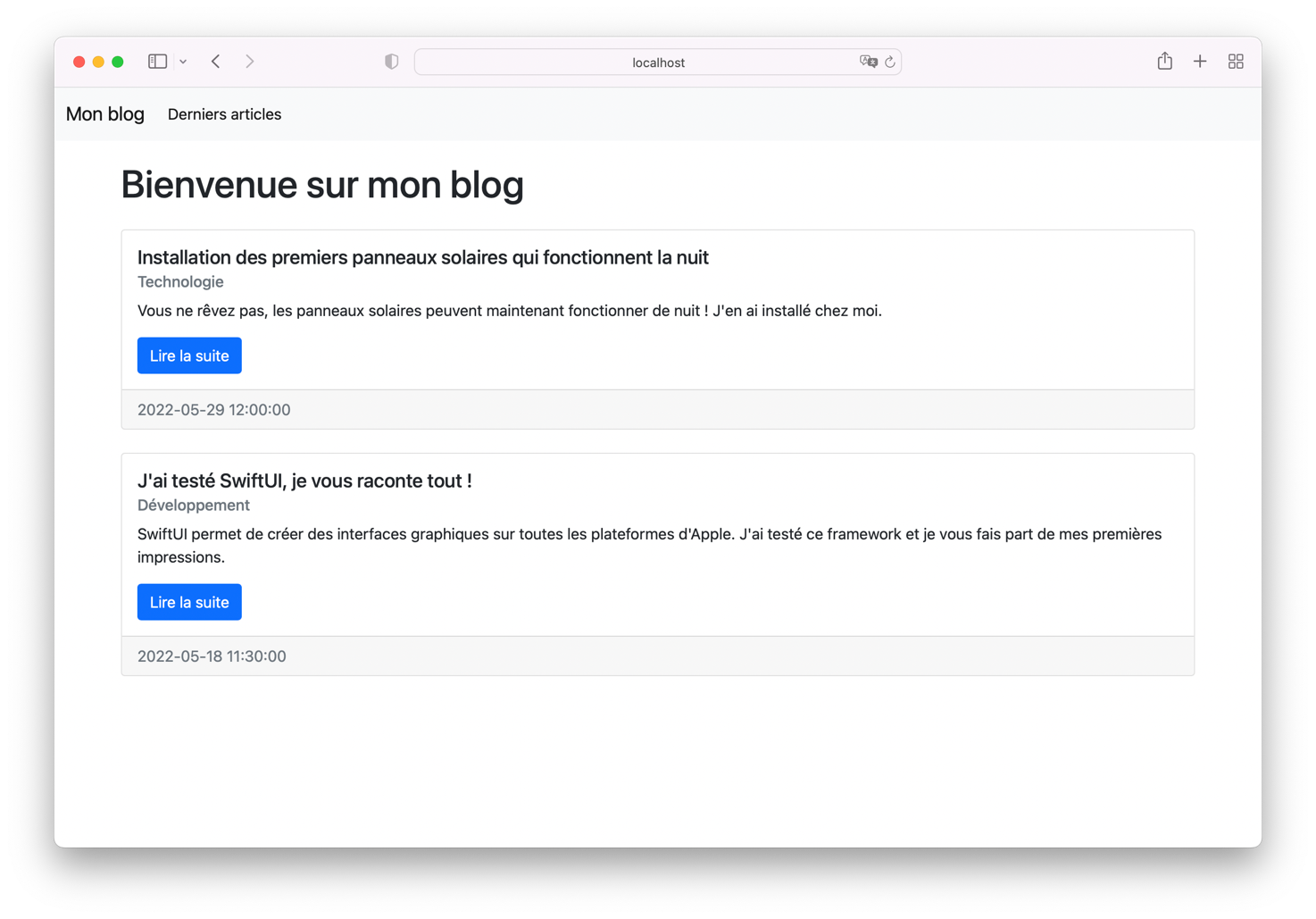
# Consigne

Complétez le code source du blog pour respecter le cahier des charges ci-dessous. Celui-ci est séparé en thèmes, il est conseillé de les faire dans cet ordre. Pour chaque partie, lisez attentivement les indications et contraintes.

Le code source final, incluant toutes les parties, est à rendre en fin de test.

# Affichage des articles

La page d’accueil doit lister les articles, du plus récent au plus ancien avec l’affichage suivant :



Indications et contraintes :

* Le routeur (« index.php ») est déjà fait pour cette partie. Vous n’avez rien à modifier.
* La fonction du contrôleur est à compléter.
* Dans le fichier Database.php, une fonction *getPosts()* est déjà déclarée. Vous devez la compléter.
* Une vue est en partie faite. Il s’agit du fichier list.php. Complétez-le.

# Détails d’un article

Lorsqu’on clique sur « Lire la suite », sur un article de la page d’accueil, une nouvelle page affiche les détails de l’article avec l’affichage suivant :

Une image contenant texte

Description générée automatiquement

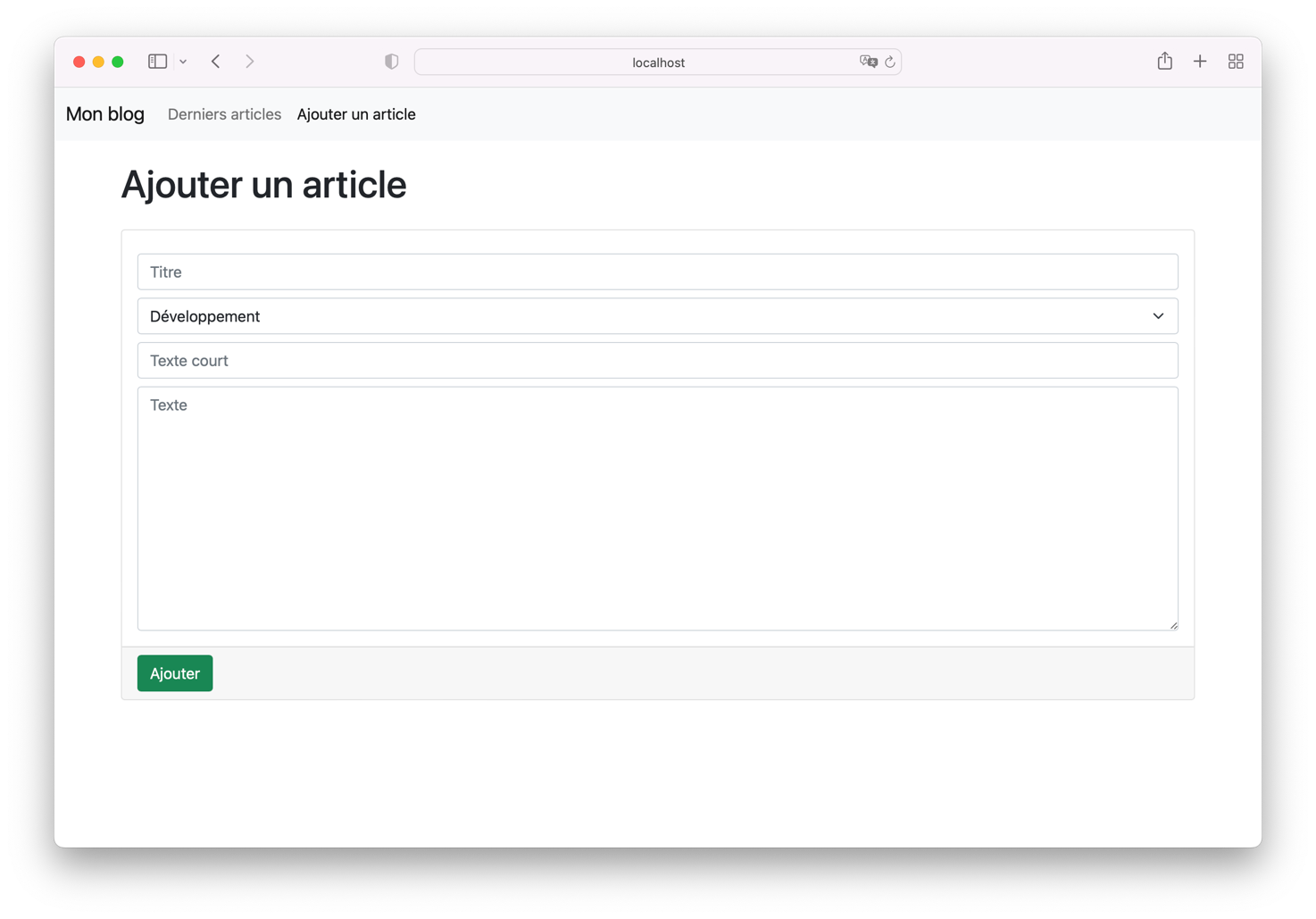
Indications et contraintes :

* Le routeur (« index.php ») est déjà fait pour cette partie. Vous n’avez rien à modifier.
* La fonction du contrôleur est à compléter.
* Dans le fichier Database.php, une fonction *getPostDetails()* est déjà déclarée. Vous devez la compléter.
* Une vue est en partie faite. Il s’agit du fichier details.php. Complétez-le.

# Ajouter un article

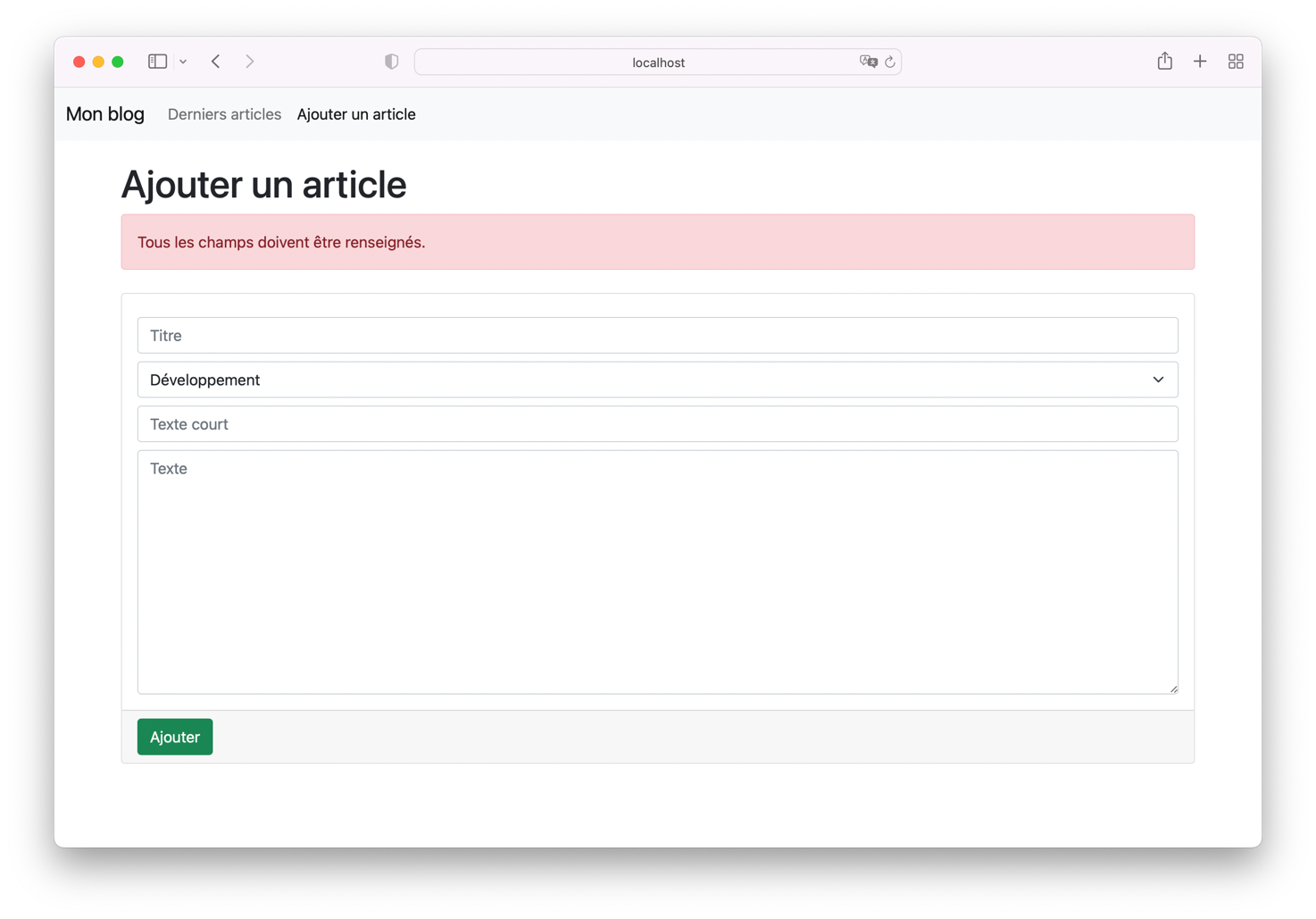
Ajouter un bouton « Ajouter un article » dans le menu.

En cliquant « Ajouter un article », une nouvelle page affiche le formulaire suivant :

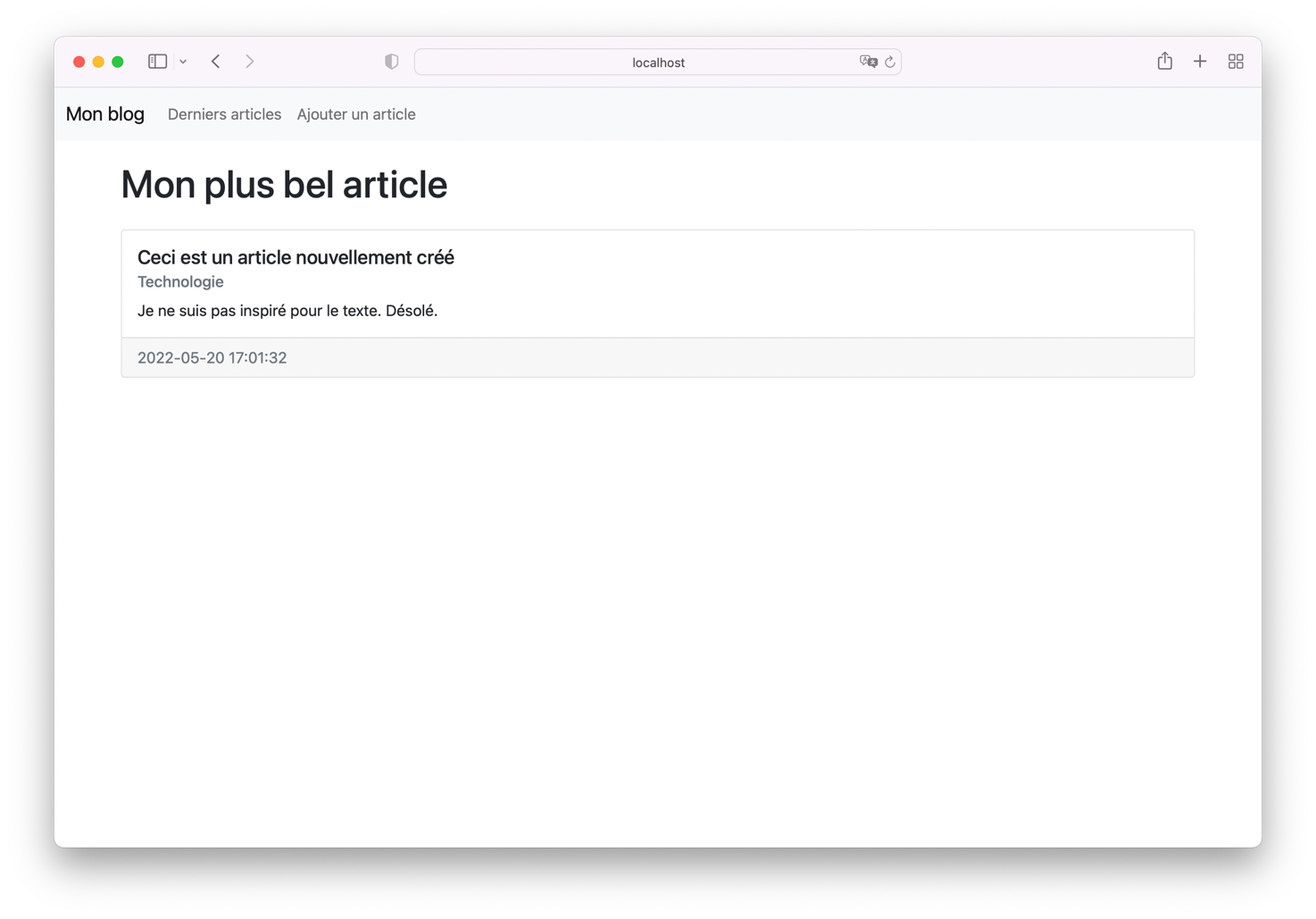


Les champs sont : le titre, une liste déroulante avec les catégories, le texte court, le texte long. La date et heure de publication est renseignée automatiquement au moment de la sauvegarde dans la base de données.

En cas d’erreur lors de l’ajout, un message doit être affiché avec le visuel suivant :



Lorsque l’article a été ajouté avec succès, la page de détails de l’article est affichée.



Indications et contraintes :

* Dans le fichier Database.php, une fonction *addNewPost()* est déjà déclarée. Vous devez la compléter.
* Une vue est en partie faite. Il s’agit du fichier addpost.php. Complétez-le.
* La liste déroulante doit afficher les catégories par ordre alphabétique.
* Tous les champs du formulaire sont obligatoires.
* Le message d’erreur affiché au-dessus du formulaire doit indiquer si un champ est manquant ou vide. En cas d’autres erreurs, un message générique est accepté.
* Le texte du menu « Ajouter un article » doit être en noir lorsqu’on est sur la page du formulaire.