

# Projet 11

## Ameliorer un projet existant en Python

Eleve : Yohan Billon

GitHub : [https://github.com/Billonyohan/P11\\_billion\\_yohan](https://github.com/Billonyohan/P11_billion_yohan)

Site : <https://purbeurreapp.herokuapp.com>

### **1/ Description du projet :**

Ce projet a pour objectif d'ajouter une fonctionnalité au projet 8 et de corriger un 'bug' dans un contexte professionnel, ainsi qu'écrire des tests pour cette nouvelle fonctionnalité.

### **2/ Entretien avec le client :**

Le client m'a demandé de corriger un dysfonctionnement avec le bouton de déconnexion qui s'affichait lorsque l'utilisateur n'était pas connecté, ainsi que d'ajouter comme fonctionnalité un formulaire de contact pour faciliter la communication avec ses clients.

Afin d'avoir plus de précisions sur le type de formulaire, je lui ai demandé quels champs il souhaitait et par quelle manière il voulait recevoir le formulaire.

Après cet entretien j'ai retenu comme informations que le client souhaitait un formulaire de contact avec trois champs (Nom d'utilisateur, email et message) et qu'il voulait le recevoir en email ainsi que le sauvegarder sur la base de données.

### **3/ Correction du bug :**

La correction du bug a été très simple à réaliser puisque j'ai simplement mis la balise contenant mon bouton dans la condition `{% if user.is_authenticated %}` (P11\_03\_demarche/boutton\_html.png et P11\_03\_demarche/boutton\_code.png).

### **4/ Ajout de la fonctionnalité :**

Pour l'ajout de la fonctionnalité j'ai tout d'abord créé un formulaire (P11\_03\_demarche/forms.png) avec les trois champs demandé par le client (Nom d'utilisateur, email et message (les deux derniers champs sont obligatoires)).

J'ai continué en créant un model (P11\_03\_demarche/models.png) avec les mêmes informations que le formulaire de contact afin de pouvoir plus tard les sauvegarder dans la base de données.

Une fois ces étapes faites, j'ai créé mon visuel (index.html) à partir du formulaire que j'ai créé (forms.py) via ma vue (views.py).

Views.py (P11\_03\_demarche/views.png) : je récupère mon formulaire ( `form = ContactForm(request.POST)` ) et je le renvoie à mon fichier html.

Index.html (P11\_03\_demarche/index.png) : j'ai créé une balise `<form>` dans laquelle j'ai ajouté une boucle `{% for field in form %}` ainsi que deux informations qui sont le nom du champ `{{ field.label_tag }}` et le champs `{{ field }}`.

Une fois le visuel fait, j'ai écrit la fonction qui permet de recevoir les données du formulaire et d'envoyer un mail de confirmation à l'utilisateur de l'application (P11\_03\_demarche/mail\_client.png) et un mail avec les données du formulaire à purbeurre (P11\_03\_demarche/mail\_purbeurre.png) ainsi que la sauvegarde du formulaire sur la base de données.

Pour l'envoi des mails j'ai utilisé le module EmailMessage.

Je vérifie d'abord que le formulaire est valide puis je récupère dans trois variables les données des champs du formulaire.

Ensuite j'ai créé deux variables dans lesquelles j'ai ajouté un message au format html, puis j'ai utilisé la fonction EmailMessage en précisant le sujet du message, le message reçu et l'adresse de réception du message et enfin j'ai utilisé la fonction send() pour envoyer les messages.

Pour la sauvegarde du formulaire dans la base de données j'ai utilisé la fonction create() dans django afin de créer une nouvelle ligne dans ma table contact avec les informations du formulaire puis j'ai utilisé la fonction save() afin de les sauvegarder.

Pour terminer, j'ai créé une variable 'info' que j'ai renvoyé à mon template une fois les étapes précédentes validées pour informer l'utilisateur que les données de son formulaire ont été validées (P11\_03\_demarche/confirmation.png).