

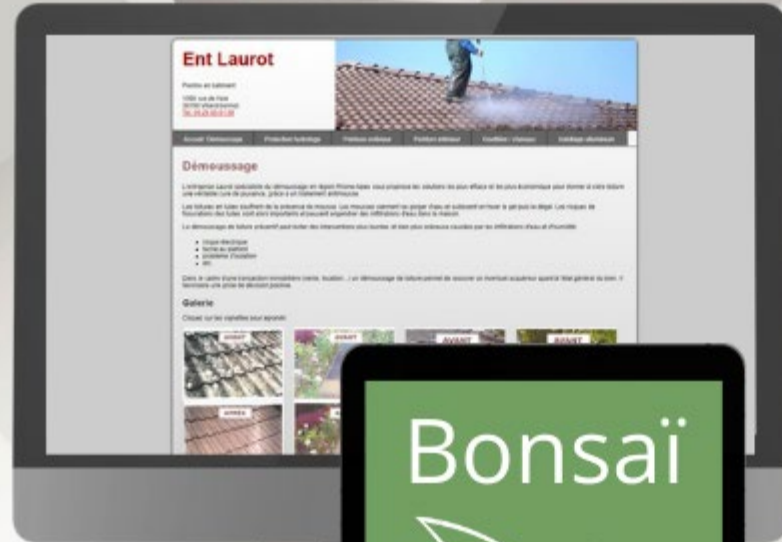


**epsi**

**MA &  
TMA**



laurot.com



je-code.com



pictionAi.fr



LuxAeterna.fr



# Thibault Vincent

Formateur permanent EPSI

[thibault.vincent@campus-cd.com](mailto:thibault.vincent@campus-cd.com)

*Toujours à votre disposition pour des compléments de cours, suivi de projet, demandes diverses.*

- ❖ Formateur en conception d'applications depuis 2015 (écoles d'ingénieur, universités, instituts)
- ❖ Ingénieur développement depuis 2010 (Sopra, Toyota, Eurotunnel etc.).

Illustration : Exemples de sites créés...  
... et toujours en fonctionnement !





# Dimension pédagogique

- ◆ Décalage entre le système universitaire et professionnel:
  - ◆ Beaucoup de projets réalisés dans un cadre scolaire sont des travaux faits « from scratch »,
  - ◆ Alors que vous aurez quasi systématiquement une base de code existante. Et parfois assez monstrueuse.
- ◆ Cela pose deux problèmes majeurs:
  - ◆ La difficulté pour un jeune diplômé ou avec peu d'expérience à reprendre le code édité par des tiers,
  - ◆ La tentation de reprendre un travail « from scratch » au lieu de modifier l'existant.
- ◆ C'est ce que va s'efforcer de résoudre ce cours en vous habituant :
  - ◆ A faire un code facilement éditable par un tiers,
  - ◆ À vous habituer à travailler sur des grosses bases de code difficilement éditables.


C'est donc un cours important dans votre montée en compétence.



# Objectifs globaux

- ◇ Développer une application maintenable
- ◇ Améliorer progressivement son approche
  - ◇ En développant sa manière de communiquer
  - ◇ Avec l'utilisation d'outils
  - ◇ En codant proprement
- ◇ En alternant plusieurs vagues successives de:
  - ◇ Modification d'application à partir d'une base ou from scratch faite par vous-même (Maintenance applicative)
  - ◇ Reprise d'application existante faite par d'autres (Tierce Maintenance Applicative)






# Programme de la (Tierce) Maintenance Applicative

## ◇ Durée du module

- ◇ MA : 20h
- ◇ TMA : 20h

## ◇ Étapes

- ◇ **Cours 1**  
MA : Démonstration de la démarche de conception. Orienté MVP : prioriser les fonctionnalités, faire simple en termes de design au début.  
Objectif : développer un travail de HTML et PHP sur 2 pages  
Mise en ligne de vos travaux sur GIT.
- ◇ **Cours 2**  
MA : Reprise du travail précédent. Développer une modification CSS, HTML, PHP impliquant la structure générale.
- ◇ **Cours 3**  
TMA par groupe de 4 (dans l'ordre alphabétique).  
Ajout de fonctionnalité sur un travail existant.
- ◇ **Cours 4**  
TMA : Ajout de fonctionnalité sur un travail existant plus complexe.
- ◇ **A partir du cours 5**  
MA : Développement d'un projet complexe.
- ◇ **Partie 6 (en plusieurs cours)**  
TMA : Ajout de fonctionnalité sur un autre travail existant plus complexe.
- ◇ **Partie 7 (en plusieurs cours)**  
TMA collective sur une autre application.



# Programme d'aujourd'hui

1. Présentation du projet fil rouge
2. Première définition (MA, TMA)
3. Survol des outils qui seront utilisés (méthodologie, organisation, testing, CI/CD/Devops)
4. Travail attendu
5. CODING !





# Présentation du projet fil rouge


Cf navigateur



# Premières définitions


MA, TMA





# (Tierce) Maintenance Applicative

- ◇ Pourvoir au bon fonctionnement d'une application.
- ◇ **Activité de :**
  - ◇ Corriger
  - ◇ Faire évoluer
- ◇ **Avec pour objectif de :**
  - ◇ Conserver un haut niveau de qualité et disponibilité
  - ◇ Répondre aux besoins évolutifs des métiers.
- ◇ **Exemples concrets :**
  - ◇ Mise à jour de logiciels
  - ◇ Résolution de bugs
  - ◇ Maintenance de la sécurité
  - ◇ Support utilisateur



# Première approche des outils abordés

Méthodologie de projet,  
organisation, testing, CI/CD,  
Devops





# Méthodologie de projet

- ◇ Structurer les différentes phases et tâches d'un projet
- ◇ **Activité de :**
  - ◇ Planifier
  - ◇ Gérer
  - ◇ Contrôler
- ◇ **En fonction des :**
  - ◇ Objectifs
  - ◇ Rôles
  - ◇ Ressources
  - ◇ Risques
- ◇ **Exemples de méthodologies de projet :**
  - ◇ Cycle en V
  - ◇ Cascade
  - ◇ Agile



# Organisation

- ◇ Conception : Merise, MCD, MLD, UML
- ◇ Documentation : CDC, Markdown, commentaire
- ◇ Suivi de projet / Ticketing : Trello, JIRA, Notion





# Testing

- ◇ Catégorie de test :
  - ◇ TU (Test Unitaires) : de fonctions, méthodes, classes..
  - ◇ 2E2 (Test End to End) : de fonctionnalités
  - ◇ Intégration : de mise en production
- ◇ Méthodologie :
  - ◇ TDD (Test Driven Development) : rouge vert refactor
- ◇ Mesure de la qualité / monitoring :
  - ◇ SonarQube



# CI / CD, Devops

- ❖ Intégration Continue / Déploiement Continu, Devops : automatisation du processus entre le développement et le déploiement, en passant par les tests automatisés.
- ❖ Virtualisation / Conteneur : Docker
- ❖ Orchestration : Kubernetes
- ❖ Workflow : Github actions, Jenkins



# Travail attendu

Version 1





# Instructions

- ◇ Réaliser un site internet d'upload de fichier :
  - ◇ Front : formulaire HTML : bouton « parcourir ses documents » et bouton « envoyer »
  - ◇ Back : validation PHP, dépôt sur le serveur, restriction de format de fichiers
- ◇ Orienté MVP : prioriser les fonctionnalités, faire simple en termes de design au début.
- ◇ Objectif : développer un travail de HTML et PHP sur 2 pages
- ◇ Rendu : à déposer sur votre espace git