
SAÉ THALES n°24 Traitement de données

Enseignant de la matière : Philippe Cam

Déroulé du projet :

- Séances libres de 1.5h ou 3h répartis dans l'emploi du temps
- Avancement maison

Objectif du projet :

Réaliser un produit capable d'extraire les trames d'un module spatial et être capable de les lire de manière ergonomique sur une interface web.

Sommaire n°1

1 / Introduction du projet

Problématique et exigences -----	4
Compétences mises en oeuvres -----	4
Planification et répartition des tâches -----	5
Analyse des risques -----	6

2 / Thales Alenia Space

Présentation de l'entreprise -----	8
------------------------------------	---

3 / Conception de l'extracteur

Analyses et conception -----	9
Réalisation du code python -----	9
Fonctionnement de l'extracteur -----	10
Diagramme d'utilisation de l'extracteur -----	15
Exécution des tests -----	16

Sommaire n°2

4 / Conception de l'interface Web V1

Configuration du Raspberry Pi 3 -----	17
Analyses et conception -----	18
Réalisation du site web -----	20
Présentation de la mise en page -----	21
Screenshots du site Web -----	24

5 / Conception de l'interface Web V2

Changement de la mise en page -----	27
Base de données du site web -----	29
Screenshots du site Web -----	35

6 / Conclusion générale

Réponse de nos outils à la problématique -----	41
Retour d'expérience -----	43

Introduction du projet

Problématique et exigences :

Ce projet de SAE se situe dans le cadre de la création de nouveaux outils pour répondre aux besoins de l'entreprise Thales Alenia Space. L'objectif est de réaliser différents outils capables d'extraire les trames d'un module spatial ayant réalisé des enregistrements fonctionnels afin de les décoder et de les envoyer sur la base de données reliant les différents outils nécessaires afin de répondre aux besoins exigés.

La base données s'occupera d'intercepter les trames afin de pouvoir les envoyer sur l'interface Web de manière ordonnée en indiquant sa taille en octets, son nom, sa date d'enregistrement ainsi que sa visualisation en binaire.

Notre solution devra répondre aux besoins des utilisateurs à savoir pouvoir interagir avec les trames de manière à les différencier en fonction de leurs différentes caractéristiques (Taille, Nom, Date) et pouvoir les supprimer.

Compétences mises en œuvres :

Afin de réaliser l'extracteur, la base de données et le site web, nous avons besoin de différents langages orientés objet, manipulation et web.

Des capacités rédactionnelles sont également exigées afin de rédiger une documentation expliquant en détail l'utilisation du site ainsi que le fonctionnement du programme python et un rapport complet présentant le projet dans sa globalité avec tous les critères.

Les différents langages utilisées dans la réalisation de ce projet :

Interface Web : Html, Css, Php, Javascript

Extracteur de trames : Python, SQL

Planification et répartition des tâches :

Rôle attribué aux participants du projet :

Jeremy Gardas	Objectif attribué : Réalisation de l'extracteur + base de données Compétence mise en œuvre : Maîtrise des langages Python et SQL ainsi que la compréhension de la problématique en question.
Evan Valerino	Objectif attribué : Réalisation du rapport + documentation Compétence mise en œuvre : Connaissance des différentes étapes du projet afin de l'expliquer en détails et simplifier l'utilisation de notre solution aux utilisateurs. Maîtrise des langages Web pour réaliser un site de documentation.
Adrien Cambier	Objectif attribué : Réalisation de l'interface web Compétence mise en œuvre : Maitrise des langages Javascript, Html, Css et Php afin de réaliser un site web affichant les enregistrements ainsi que relier la documentation.

Analyse des risques :

1 / Raspberry :

- Problème de performance : Le Raspberry rencontre des difficultés lors de l'extraction de données, ce qui limite son utilisation.
- Problème d'affichage : Des problèmes graphiques surviennent lors de l'utilisation du Raspberry.
- Solution pour Raspberry : Une solution potentielle consisterait à développer directement le code sur le Raspberry, plutôt que de le développer sur une plateforme Linux puis de le tester ultérieurement sur le Raspberry. Cela permettrait de mieux prendre en compte les spécificités et les limitations du Raspberry dès le départ.

2 / Equipiers :

- Problème d'organisation des tâches : Il peut y avoir des situations où les membres de l'équipe ne sont pas clairement informés des tâches qui leur sont assignées, des délais à respecter ou des responsabilités qui leur sont assignés.
- Solution pour l'équipier : Il est important de maintenir une communication régulière et transparente avec l'équipe. Tenir les membres de l'équipe informés de l'avancement des tâches, partager les obstacles rencontrés et solliciter de l'aide si nécessaire peut contribuer à améliorer l'organisation des tâches et à éviter les retards.

3 / Fichiers :

- Les données du fichier peuvent changer et corrompre l'ensemble du système.

- Solution pour le fichier : Pour faire face à ce risque, il est recommandé de concevoir notre extracteur de données de manière à ce qu'il soit adaptatif. Cela signifie que l'extracteur doit être en mesure de gérer les changements de format, de structure ou de contenu des fichiers sans compromettre l'intégrité du système. En anticipant les éventuels changements et en développant une solution flexible, nous minimisons les risques de corruption des données.

En prenant en compte ces différentes solutions, nous pouvons réduire les risques liés à l'utilisation du Raspberry, à l'organisation des tâches au sein de l'équipe et à la gestion des fichiers, ce qui contribuera à la réussite de notre projet.

Thales Alenia Space :

Présentation de l'entreprise Thales :

Thales est une entreprise multinationale française spécialisée dans les technologies de l'information et de la communication, la sécurité et la défense. Fondée en 1893, Thales est présente dans plus de 56 pays à travers le monde et emploie plus de 83 000 personnes.

Les domaines d'activité de l'entreprise :

Thales est une entreprise internationale leader dans les domaines des technologies de l'information et de la communication, de la sécurité et de la défense. Elle propose des solutions en cybersécurité, communication satellite, sécurité numérique et cloud computing. Elle fournit également des systèmes critiques tels que le contrôle aérien, la gestion du trafic ferroviaire et la sécurité publique. Thales conçoit des systèmes de transport comme la signalisation ferroviaire et les équipements de transport en commun. Dans le domaine de la défense et de la sécurité, Thales fournit des systèmes de communication sécurisés, des protections contre les cyberattaques et des systèmes de surveillance. Elle est également présente dans l'aérospatiale et l'espace avec des solutions de communication par satellite, de navigation et de surveillance. Thales développe des solutions de sécurité et de surveillance pour les infrastructures critiques, les gouvernements, les villes intelligentes et les entreprises.

Conception de l'extracteur

Analyses et conception :

Le code python en général doit être capable de lire les enregistrements venant d'un module spatial afin de déchiffrer les trames et de les enregistrer dans une base de données créée afin de relier l'interface web à l'extracteur de trames.

Ces trames enregistrées dans des tables seront différencierées par leur date ainsi que leur nom d'enregistrement et leur taille en octets.

Réalisation du code python :

L'extracteur de trames doit être réalisé en langage python :

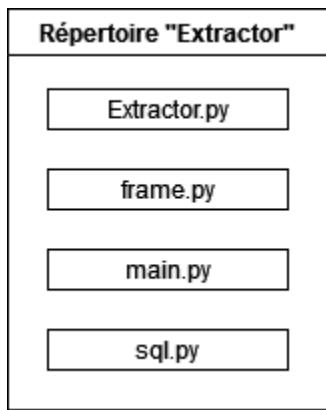
- Python est un langage de programmation interprété, multi paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet.

Les fichiers enregistrées sous l'extension .py sont rangées dans le même répertoire afin de faciliter l'accès entre les différents fichiers composants l'extracteur. Des fonctions sont réalisées dans différents fichiers afin de faciliter sa lecture et sa modification.

Ainsi, la fonction principale fait appel à ces fonctions privées enregistrées dans différents fichiers afin de rendre opérationnel l'extracteur.

Fonctionnement de l'extracteur :

L'extracteur est composé de 4 fichiers différents pour fonctionner :



Le fichier extractor.py est celui qui fera appel à toutes les fonctions réparties au sein du fichier même ainsi que frame.py et sql.py.

Les fichiers frame.py et sql.py sont uniquement des fichiers de type “Class” afin de respectivement créer l'affichage des trames et la base de données qui accueillera tous les enregistrements différenciés par leur nom, taille en octets et date.

Voici un exemple de comment faire pour appeler une Class :

Fic sql.py -> Class SQL: #Création de la class SQL

Fic extractor.py -> SQL.<nom de la fonction>(<option>) #Exécute une fonction de la class SQL

Le fichier main.py quant à lui est uniquement destiné à exécuter le fichier extractor.py et renvoie 0 ou 1 en fonction de si l'extracteur est en cours d'exécution.

L'extracteur comporte plusieurs méthodes, voici à quoi elles servent et pourquoi nous les utilisons.

- run
- parse_cmd_line_args
- extract_frames_from_file_and_add_them_to_db
- insert_frames_inti_db
- calculate_frame_date
- calculate_packet_date

Méthode run :

Cette méthode exécute l'extracteur. Elle commence par appeler la méthode `parse_cmd_line_args()` pour analyser les arguments de ligne de commande. Ensuite, elle extrait les trames Ethernet à partir d'un fichier binaire et les ajoute à la base de données SQL.

Méthode parse cmd line args :

Cette méthode analyse les arguments de ligne de commande passés au programme. Elle utilise le module argparse pour faciliter cette tâche. Elle vérifie également la validité des arguments et affiche des messages d'erreur en cas de problème.

Méthode extract_frames_from_file_and_add_them_to_db :

La méthode est utilisée pour extraire les trames du fichier binaire et les ajouter à la base de données SQL.

Elle commence par ouvrir le fichier binaire en utilisant la bibliothèque bitstring et lit le contenu du fichier bit par bit.

Ensuite, une boucle est utilisée pour extraire chaque trame :

La méthode self.current_frame = frame.Frame() crée une instance de la classe Frame à chaque itération pour stocker les informations de la trame actuelle.

La méthode self.find_test_name_and_execution_date() est appelée pour rechercher le nom du test et la date d'exécution, probablement à partir des données de la trame.

La méthode lit ensuite les champs de l'en-tête de la trame, qui sont stockés dans le dictionnaire self.current_frame.fields. Les valeurs sont converties en hexadécimal pour une meilleure lisibilité.

Ensuite, les champs connus (MACs, IPs, ...) sont lus. La méthode parcourt les champs à partir de l'indice 7 du dictionnaire self.current_frame.fields et lit les données correspondantes à partir du flux de bits. Les valeurs sont à nouveau converties en hexadécimal.

Si la taille totale des bits lus est inférieure à la taille de la trame, cela signifie qu'il reste un champ de message à lire. La méthode lit donc les bits restants pour atteindre la fin de la trame.

Diverses méthodes (`self.calculate_frame_date()`, `self.calculate_packet_date()`, `self.calculate_MACs_and_IPs()`, `self.calculate_msg_type()`) sont ensuite appelées pour effectuer des modifications supplémentaires sur les valeurs des champs et les rendre plus lisibles.

Ensuite, la méthode supprime le préfixe "0x" des valeurs hexadécimales des champs.

Enfin, la méthode `self.insert_frames_into_db(self.current_frame)` est appelée pour insérer la trame actuelle dans la base de données SQL.

En cas d'erreur pendant le processus d'extraction des trames, une exception est levée et le message d'erreur est affiché. La méthode renvoie `False` en cas d'erreur.

À la fin, la méthode renvoie `True` pour indiquer le succès de l'extraction des trames.

Méthode `insert_frames_into_db` :

Cette méthode est utilisée pour insérer les trames dans la base de données SQL.

Elle commence par ouvrir la base de données en appelant la méthode `open_db()` de l'objet `self.sql_db`.

Ensuite, elle appelle la méthode `insert_frame(frame)` de l'objet `self.sql_db` pour insérer la trame dans la base de données.

Méthode calculate_frame_date :

Cette méthode est utilisée pour calculer la date de la trame (depuis le 1er janvier 1970 00:00:00).

Elle utilise la valeur du champ "frame_date" de l'objet self.current_frame qui est supposée être un nombre hexadécimal représentant les secondes écoulées depuis une certaine date.

Elle calcule la date en utilisant l'objet datetime.datetime et ajoute le résultat au champ "frame_date" dans le format "AAAA-MM-JJ HH:MM:SS.ssssss".

Méthode calculate_packet_date(self) :

Cette méthode est utilisée pour calculer la date du paquet (depuis le 1er janvier 2000 12:00:00).

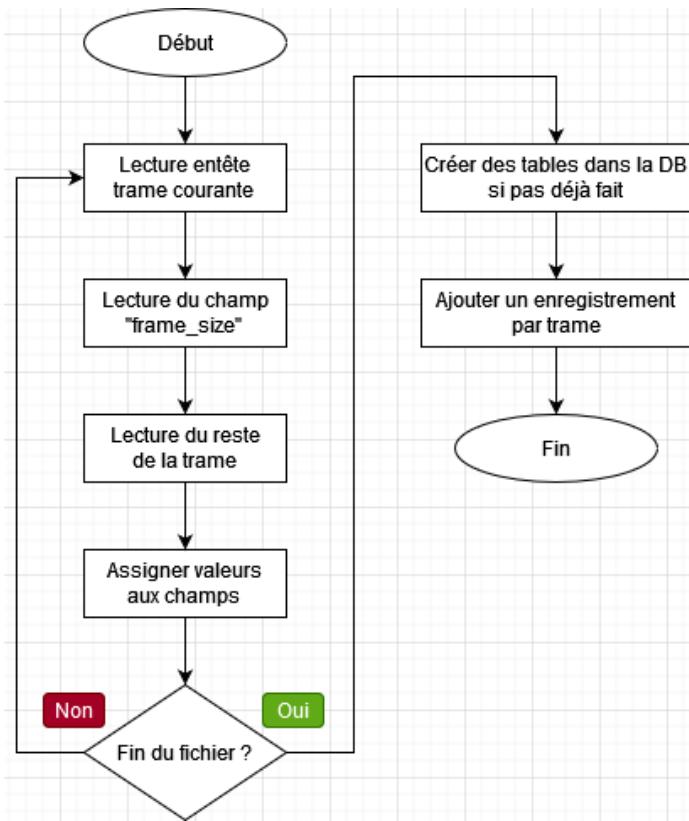
Elle vérifie d'abord si tous les champs requis (field_33, field_34, field_35) ont une valeur dans l'objet self.current_frame. Si l'un d'eux est manquant, la méthode se termine sans rien faire.

Elle utilise les valeurs des champs field_33, field_34 et field_35 de l'objet self.current_frame qui sont supposées être des nombres hexadécimaux représentant des secondes et des fractions de seconde.

Elle calcule la date en utilisant l'objet datetime.datetime et ajoute le résultat à l'attribut packet_date de l'objet self.current_frame dans le format "AAAA-MM-JJ HH:MM:SS.ssssss".

Diagramme d'utilisation de l'extracteur :

Schéma synthétisant les différentes étapes clés :



Lors de l'exécution de l'outil, la trame est lue en commençant par l'entête pour ensuite déchiffrer son contenu et l'assigner dans une table SQL réalisée via le programme python.

Exécution des tests :

Tableau récapitulatif des anomalies relevés durant les tests :

Rapport Anomalie n°1		
Titre : Date_issue	Jérémy Gardas	17/03/23
Observation	Num 01 Mauvais affichage des dates de chaque enregistrement. Date calculée à partir de l'année 1970.	
Analyse	Num 01 Modification du fichier extractor.py. Calcul de date à réaliser pour remédier au problème d'affichage.	
Correction	Num 02 Modification de la fonction calculate_frame_date afin d'ajouter le temps nécessaire pour obtenir une date calculé à partir de l'an 0	

Conception de l'interface Web V1

Configuration du Raspberry Pi 3 :

Afin d'héberger le site web ainsi que l'extracteur, le Raspberry fera office d'hébergeur afin de simuler l'utilisation de notre solution sur une faible configuration (1 Go de RAM).

Un OS (Système d'exploitation) doit être installé sur la carte SD du Raspberry afin de pouvoir lancer la machine, pour cela il faut se rendre sur le lien ci-dessous et procéder à l'installation détaillée.

<https://www.raspberrypi.com/software/>

Lorsque le Raspberry sera fonctionnel, il faudra télécharger le paquet SQL afin de pouvoir héberger son site web PHP sur sa machine accessible en localhost.

Commandes Debian (Terminal de Commande) :

```
#apt-get update  
  
#apt-get install php php-sqlite3
```

Après cela, il suffira d'injecter les fichiers de l'extracteur python ainsi que le code du site web afin de pouvoir accéder à notre site web php. Il faut taper le mot clé "localhost" dans un navigateur internet afin d'accéder à l'interface web.

Analyses et conception :

Le site internet devra comporter 3 rubriques :

- La page d'accueil
- La documentation
- L'utilisation Sniffer

1 | La page d'accueil devra comporter des informations permettant de résumer en quelques mots l'utilité du site web. Celle-ci expliquera le cadre du développement de ce site web ainsi que la fonction de notre solution brièvement. Un menu sera réalisé afin de relier les 3 différentes rubriques à savoir : Home, Documentation, Sniffer Ethernet.

2 | La documentation est une seconde page du site web que l'on pourra accéder via le menu réalisé au préalable pour faciliter son accessibilité.

Le rôle de cette dernière est d'accéder à une page expliquant pourquoi nous avons réalisé cette rubrique nommée documentation ainsi qu'un bouton d'accès pour télécharger la documentation sous différents formats PNG, PDF etc...

Le fichier de documentation téléchargé doit expliquer en détail l'utilisation du site web ainsi que de l'extracteur de trames. En outre, elle se réfère à un manuel d'utilisation pour simplifier l'expérience utilisateur.

3 | La dernière rubrique qui compose le site sera nommée Sniffer Ethernet et son utilité est la plus importante des 3 rubriques car c'est à partir de celle-ci que nous pourrons interagir avec l'extracteur de trames afin de visualiser les différents enregistrements.

La page se présente avec un menu de recherche permettant de rechercher les enregistrements en fonction de leur nom, de leur taille en octets ou bien de leur date.

Afin de pouvoir rechercher ces différentes trames, un bouton afin de lancer la capture des trames sera réalisé. Lors de son arrivée sur la page, l'utilisateur devra ainsi actionner la capture des trames et ensuite lancer une recherche si celui-ci désire une trame en particulier. Dernière addition que la page devra proposer et la modification du nom d'une trame.

Le site web doit pouvoir enregistrer la modification nominale d'une trame et ainsi changer cette dernière sur la base de données reliant l'interface web à l'extracteur de trames réalisé en python. Une particularité à prendre en compte et que si le nombre de tailles est démesuré, le site doit pouvoir limiter la capture de trame à un certain nombre et proposer une fonctionnalité afin de lire plus de trames si l'utilisateur le souhaite.

Pour l'aspect visuel du site, étant donné que nous parlons de trame venant de module spatial et que l'entreprise en question nommé Thales est une entreprise travaillant dans le domaine spatial, les couleurs les plus présentes seront le noir et le bleu pour s'appuyer sur les couleurs du logo de l'entreprise ainsi qu'évoquer l'idée de l'univers.

Le texte rédigé sur les pages sera principalement en blanc afin de faciliter sa lecture étant donné que le fond sera de couleurs sombres.

Réalisation du site web :

Le site web doit être réalisé en langage html, css, php et javascript :

- Html : Le HyperText Markup Language, généralement abrégé HTML ou, dans sa dernière version, HTML5, est le langage de balisage conçu pour représenter les pages web
- Css : Les feuilles de style en cascade, généralement appelées CSS de l'anglais Cascading Style Sheets, forment un langage informatique qui décrit la présentation des documents HTML et XML. Ce langage conçoit l'aspect visuel du site afin d'offrir une meilleure ergonomie.
- Php : Hypertext Preprocessor, plus connu sous son sigle PHP, est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet.
- JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web. Avec les langages HTML et CSS, JavaScript est au cœur des langages utilisés par les développeurs web.

Afin de profiter des fonctionnalités qu'offre le langage php, le site doit être hébergé sur une machine en local durant sa réalisation avant le rendu final ou celui-ci sera hébergé sur un raspberry pour effectuer la simulation de notre solution sur une configuration rudimentaire.

Le site est accessible sur OneDrive durant sa réalisation afin de permettre aux différents participants du projet (Evan, Jeremy, Adrien) d'accéder au site et d'apporter des modifications. Cela est également un filet de sécurité afin de pouvoir enregistrer les différents fichiers .html .css .php etc... sans qu'elles puissent être perdues.

Présentation de la mise en page :

L'entête de la page :

Cette entête sera présente dans les trois différentes pages que compose le site Web (Home, Documentation, Sniffer Ethernet).

En bref, les éléments sont contenus dans une balise `<nav>` qui sert à contenir le nom de l'entreprise destiné “Thales” ainsi qu'un menu permettant d'accéder aux autres pages.

Le menu est affiché horizontalement mais lorsque la taille de la fenêtre diminue, un menu “hamburger” s'affiche afin de ranger le menu verticalement.

Cela se réalise en css notamment avec le code :

```
@media screen and (max-width: ??em){ *contenu css* }
```

Les éléments sont positionnés en “relative” car c'est la méthode de positionnement css la plus efficace pour la réalisation d'un site web responsive.

Afin d'accéder aux autres pages, le code html des rubriques Home, Documentation, Sniffer Ethernet, sera composé de la manière suivante :

```
<ul>
    <li href="fichier.html">Home</li>
    ...
</ul>
```

L'élément “href” présent dans la balise permet de rediriger vers une autre page ou bien un élément de la page avec le nom de l'id ex : href="#title"

La page d'accueil “Home” :

Le corps de la page est composé de `<div>` alternant entre la couleur bleu et l'image de fond “galaxie”. Ces div contiennent du texte et des images qui sont centrées verticalement et horizontalement.

Exemple de code css pour centrer les éléments :

```
div{
    display:flex;
    justify-content:center;
    align-items:center;
}
```

Sur cette page, il y a 3 rubriques : Introduction, Documentation, Sniffer Ethernet.

La documentation :

Cette page est destinée à la documentation de notre projet Sniffer.

Une balise <bouton> redirige vers une autre page web qui contiendra le manuel d'utilisation de notre solution Sniffer Ethernet ainsi qu'une explication détaillée de l'extracteur de trames Python.

Dans cette page dédiée à la documentation, un menu en position fixed sera placé à gauche afin de pouvoir naviguer plus facilement dans la page.

Le thème principal du site est repris en prenant l'image de "galaxie" en fond et la couleur bleu sombre rappelant le logo Thales.

L'outil "Sniffer Ethernet" :

Cette page est destinée à l'outil Sniffer Ethernet. Des boutons sont installés afin d'interagir avec la base de données SQL qui enregistre les trames de la forme suivante :

```
<form method="post" action="fichier.php">  
    <input type="text" name="testName" placeholder="Chercher par nom">  
    <input type="submit" value="Rechercher">  
</form>
```

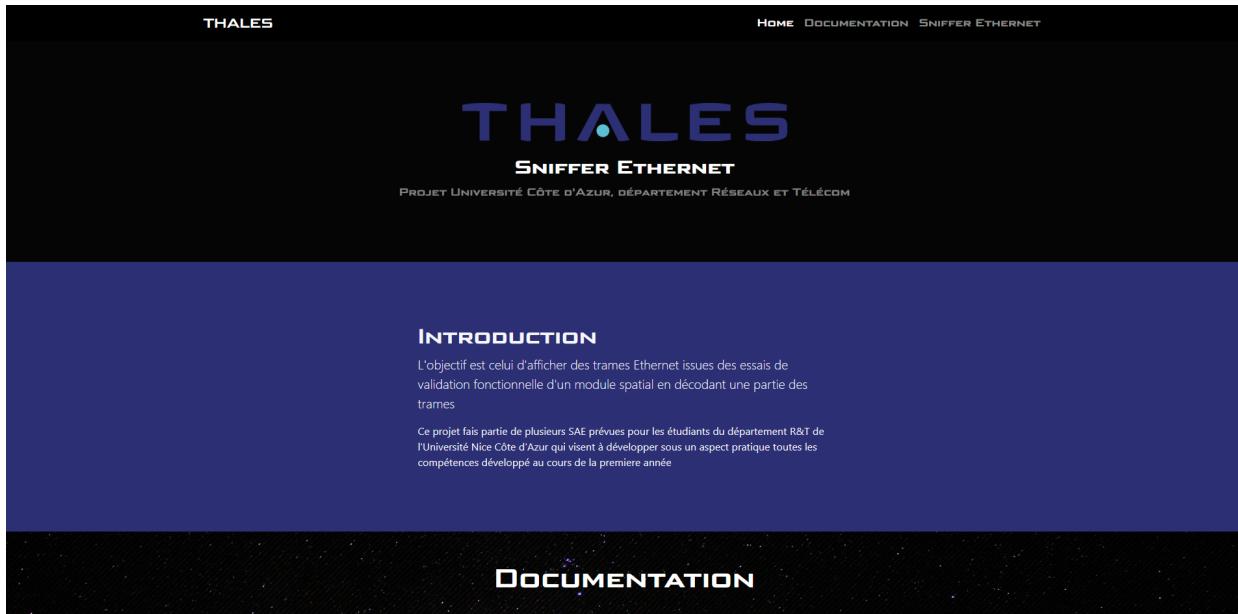
Dans cet exemple, une balise <form> contient des balises <input> afin d'envoyer des informations vers le fichier.php.

La balise <input> peut être de type "text", "date" ou bien "submit".

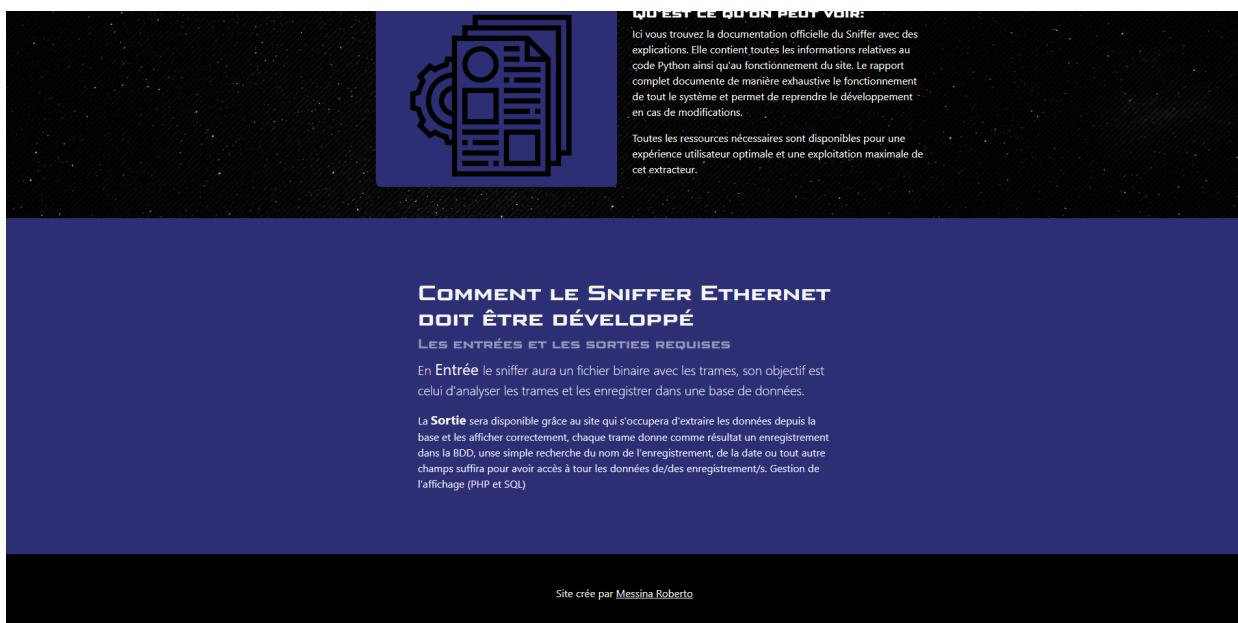
Ce sont ces input qui vont permettre d'interagir avec la base de données SQL car ils effectueront des recherches en fonction des critères définis.

Screenshots du site Web :

Page d'accueil “Home” :

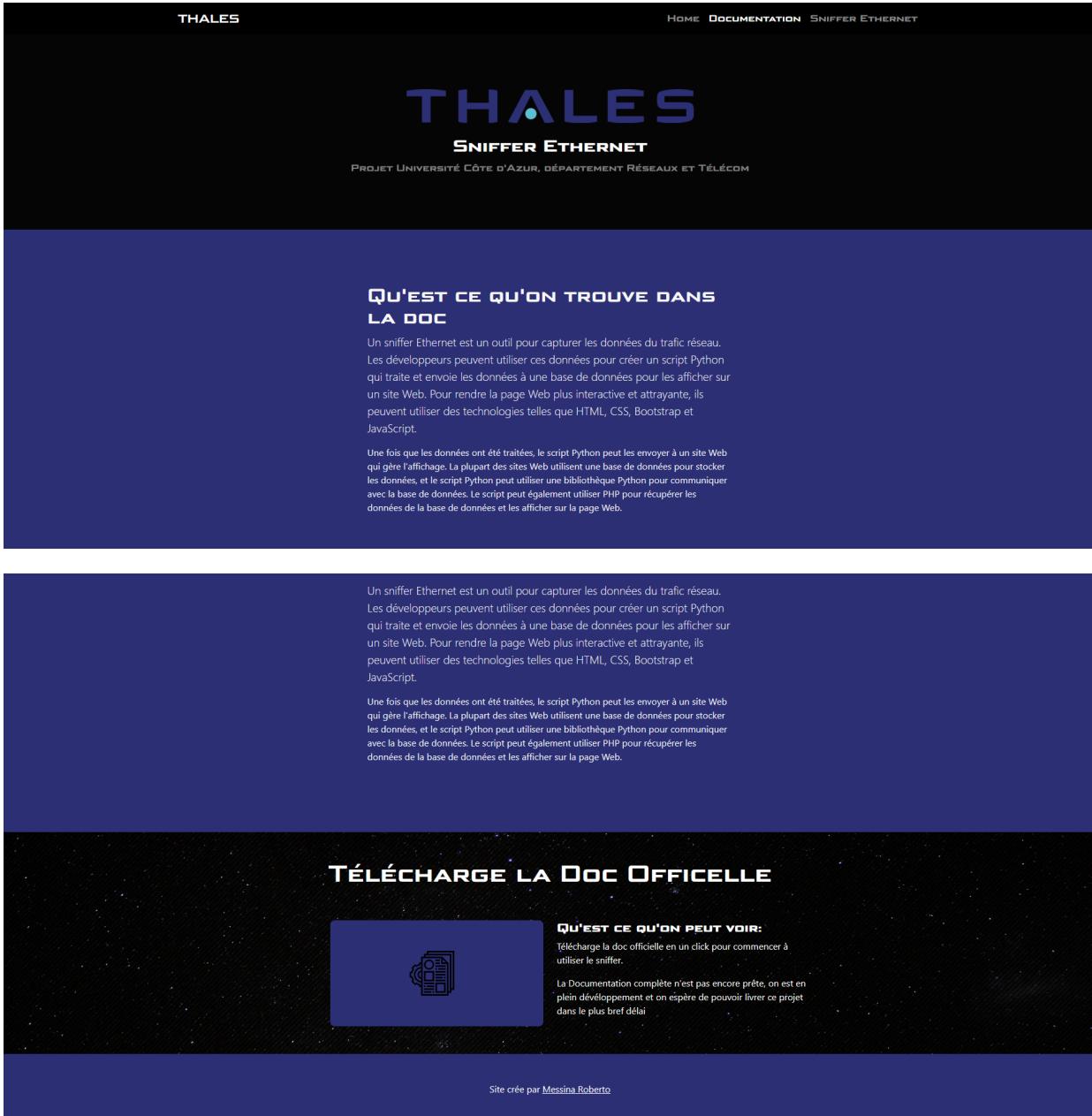


The screenshot shows the homepage of the project. At the top, there is a dark header with the THALES logo and a navigation bar with links to HOME, DOCUMENTATION, and SNIFFER ETHERNET. The main content area has a dark blue background. It features the THALES logo and the text "SNIFFER ETHERNET" below it, followed by "PROJET UNIVERSITÉ Côte d'Azur, DÉPARTEMENT RÉSEAUX ET TÉLÉCOM". Below this, there is a section titled "INTRODUCTION" with some descriptive text and a small image of a document with gears.



The screenshot shows the Documentation page. It features a dark blue header with the word "DOCUMENTATION" in white. Below this is a section titled "QU'EST CE QU'ON PEUT VOIR?" with a gear and document icon. It contains text about the official documentation and code. Another section titled "COMMENT LE SNIFFER ETHERNET DOIT ÊTRE DÉVELOPPÉ" follows, with a sub-section "LES ENTRÉES ET LES SORTIES REQUISES". It describes the input and output requirements for the Sniffer. The bottom of the page includes a footer with the text "Site créé par Messina Roberto".

La page documentation :



The screenshot shows a web page with a dark blue header containing the Thales logo and the text "SNIFFER ETHERNET". Below the header, a large blue section contains the heading "QU'EST CE QU'ON TROUVE DANS LA DOC". It includes two paragraphs of text about sniffing and a "TÉLÉCHARGE LA Doc OFFICIELLE" button.

THALES
SNIFFER ETHERNET
PROJET UNIVERSITÉ CÔTE D'AZUR, DÉPARTEMENT RÉSEAUX ET TÉLÉCOM

QU'EST CE QU'ON TROUVE DANS LA DOC

Un sniffer Ethernet est un outil pour capturer les données du trafic réseau. Les développeurs peuvent utiliser ces données pour créer un script Python qui traite et envoie les données à une base de données pour les afficher sur un site Web. Pour rendre la page Web plus interactive et attrayante, ils peuvent utiliser des technologies telles que HTML, CSS, Bootstrap et JavaScript.

Une fois que les données ont été traitées, le script Python peut les envoyer à un site Web qui gère l'affichage. La plupart des sites Web utilisent une base de données pour stocker les données, et le script Python peut utiliser une bibliothèque Python pour communiquer avec la base de données. Le script peut également utiliser PHP pour récupérer les données de la base de données et les afficher sur la page Web.

TÉLÉCHARGE LA Doc OFFICIELLE

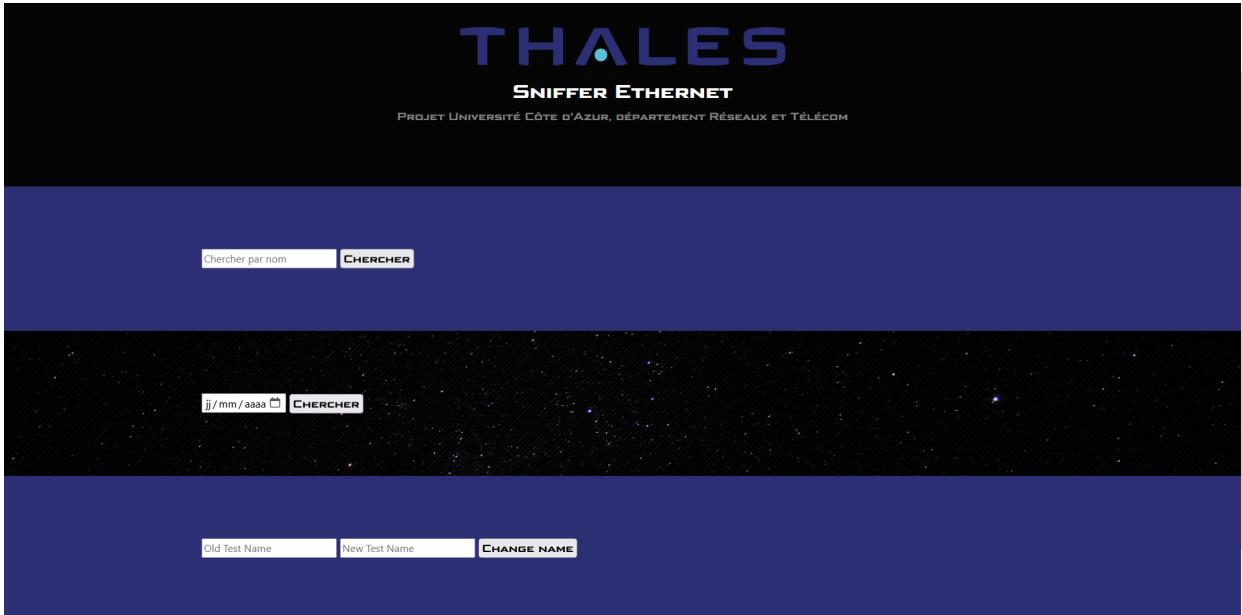
QU'EST CE QU'ON PEUT VOIR:
Télécharge la doc officielle en un click pour commencer à utiliser le sniffer.

La Documentation complète n'est pas encore prête, on est en plein développement et on espère de pouvoir livrer ce projet dans le plus bref délai

Site créé par Messina Roberto

Ceci est la page destinée à accéder à la documentation, vous pouvez retrouver une image de la page Web documentation dans la version 2 du site.

L'outil Sniffer "Sniffer Ethernet" :



La page "Sniffer Ethernet" nécessitait encore du travail afin d'afficher les enregistrements des bases de données.
Cependant, dû au changement de template de la page web, nous avons donc continué le code php sur la deuxième version du site.

Conception de l'interface Web V2

Changement de la mise en page :

Afin de rendre le site web plus lisible et simple d'utilisation, nous avons réduit le contenu textuel afin de ne laisser que l'essentiel et de mieux gérer la gestion du vide.

Dans la version précédente, beaucoup d'espace était disponible et non utilisé, nous avons donc palier à ce problème.

Désormais, le site web est composé de 4 rubriques :

- Home
- Explorer
- Transfer Functions
- Documentation

1 | Nous avons gardé une page d'accueil reprenant le même principe que dans l'ancienne version à savoir résumer en quelques mots l'utilité de notre solution et comment elle fonctionne tout en gardant un menu navigable en position fixed.

La page Home est la page d'accueil de notre projet qui introduira en quelques mots notre solution ainsi que le cadre dans lequel il est réalisé.

2 | La page Explorer sert à exécuter l'outil en lisant les enregistrements présents dans les bases de données réalisées au préalable afin de les trier par Date, Nom et taille en octets.

3 | La page Transfer Functions est destinée uniquement à la modification nominale des enregistrements. On sélectionne l'enregistrement souhaité et on modifie depuis l'interface Web son nom qui s'affiche de la sorte dans la BDD.

4 | La page documentation est directement le site de documentation à savoir le manuel et non une page intermédiaire pour relier la doc afin de rendre plus facile l'accès et d'éviter les contenus textuels facultatifs.

À noter que pour générer le code python des fichiers lourds à savoir (extractor.py, sql.py, frame.py) un code php a dû être réalisé afin de lire chaque ligne des fichiers et ajouter une balise
 à la fin de chaque ligne pour faciliter la lecture du code dans la documentation. L'avantage de cette solution est que l'affichage prend en compte la modification des fichiers pythons étant donné qu'elle affiche directement le contenu venant de ces fichiers.

Exemple de code pour la lecture des fichiers par ligne :

```
<?php
$fichier = fopen("extractor/extractor.py", "r");
if ($fichier) {
    while (($ligne = fgets($fichier)) !== false) {
        echo $ligne . "<br>";
    }
    fclose($fichier);
} else {
    echo "Impossible d'ouvrir le fichier.";
}
?>
```

On ouvre le fichier python et on définit une boucle qui affiche les lignes du code proprement sur la page web de documentation.

Base de données du site web :

Injection SQL et XSS :

- SQL est un langage informatique normalisé servant à exploiter des bases de données relationnelles. La partie langage de manipulation des données de SQL permet de rechercher, d'ajouter, de modifier ou de supprimer des données dans les bases de données relationnelles. Outre le langage de manipulation des données.

- Le cross-site scripting (XSS) est un type de faille de sécurité des sites web permettant d'injecter du contenu dans une page, provoquant ainsi des actions sur les navigateurs web visitant la page.

On prépare les requêtes SQL avant de les exécuter :

```
public function run_sql_query($query, $params_to_bind = array())  
{  
    $statement = $this->dbh->prepare($query);  
    foreach ($params_to_bind as $param => $value)  
    {  
        $statement->bindValue($param, $value);  
    }  
    $statement->execute();  
    return $statement->fetchAll(PDO::FETCH_ASSOC);  
}
```

En effet, les tables SQL sont déjà générées afin d'éviter aux utilisateurs d'injecter leur propre table pouvant altérer la sécurité du site web.

Différents types de tables SQL :

Le langage SQL possède différents modèles de tables notés de la forme xNF où x représente un entier naturel partant de 1.

1 | Première forme normale (1NF) : La première forme normale (1NF) est le niveau initial de normalisation dans le modèle relationnel. Pour qu'une table soit en 1NF, elle doit respecter les règles suivantes :

- Les données doivent être organisées en tables ayant des lignes et des colonnes.
- Chaque colonne doit contenir une seule valeur atomique (non divisible).
- Chaque ligne doit être unique et identifiable par une clé primaire.

L'objectif principal de la 1NF est d'éliminer les redondances de données et de garantir l'intégrité des données.

2 | Deuxième forme normale (2NF) : La deuxième forme normale (2NF) s'appuie sur la 1NF et ajoute une condition supplémentaire :

- Toutes les colonnes non clés d'une table doivent dépendre entièrement de la clé primaire.

En d'autres termes, chaque colonne non clé doit être fonctionnellement dépendante de la totalité de la clé primaire, et non d'une partie seulement. Si une dépendance partielle existe, il convient de séparer les données en plusieurs tables pour éviter les redondances.

3 | Troisième forme normale (3NF) : La troisième forme normale (3NF) renforce les principes de la 2NF et ajoute une nouvelle condition :

- Aucune colonne non clé ne doit dépendre transitivement d'une autre colonne non clé.

Cela signifie qu'une table en 3NF doit éviter les dépendances transitives, où une colonne non clé dépend indirectement d'une autre colonne non clé via une colonne intermédiaire.

Visualisation des tables SQL :

La base de données est composée de 4 différentes tables SQL.

- field_names pour définir le changement des colonnes sur le site
- transfert_functions pour changer le nom des enregistrements
- tests pour définir un test de trames
- frames pour afficher les enregistrements avec leurs critères (date, nom etc...)

field_names (old VARCHAR(50), new VARCHAR(50))	transfert_functions (field_name VARCHAR(50), brut VARCHAR(50), label VARCHAR(50))
---	---

<pre>tests (id INTEGER, name TEXT, execution_date TEXT, PRIMARY KEY("id" AUTOINCREMENT))</pre>	<pre>frames (test_id INTEGER, frame_date TEXT, bench_3 TEXT, bench_5 TEXT, frame_size TEXT, MAC_dest TEXT, MAC_src TEXT, IP_src TEXT, IP_dest TEXT, field_1 TEXT (field 1 jusqu'à 35), packet_date TEXT, msg_type TEXT)</pre>
--	---

En SQL, différents termes font références au type de données pouvant être utilisé pour être rangés dans la base de données.

Dans le cas ci-dessus, nous avons utilisé 3 types de données différents à savoir VARCHAR, TEXT et INTEGER.

PRIMARY KEY est à part car elle définit une clé pour différencier les tables.

1 | VARCHAR est une abréviation de "variable character". C'est un type de données utilisé pour stocker des chaînes de caractères de longueur variable. Lorsque vous déclarez une colonne comme VARCHAR, vous devez spécifier la longueur maximale de la chaîne de caractères qu'elle peut contenir.

Par exemple, VARCHAR(50) signifie que la colonne peut stocker une chaîne de caractères d'une longueur maximale de 50 caractères.

2 | TEXT est un type de données utilisé pour stocker des chaînes de caractères de longueur variable plus grandes que ce que VARCHAR peut contenir. Contrairement à VARCHAR, vous n'avez pas besoin de spécifier une longueur maximale pour une colonne de type TEXT.

Elle peut stocker de grandes quantités de texte, jusqu'à la limite de taille autorisée par le système de gestion de base de données. Par exemple, vous pouvez utiliser TEXT pour stocker des descriptions longues ou des articles de blog.

3 | INTEGER est un type de données utilisé pour stocker des nombres entiers. Il peut contenir des nombres positifs et négatifs, sans décimales. Les valeurs INTEGER peuvent varier en taille en fonction du nombre de bits utilisés pour les stocker, mais généralement, elles sont stockées sur 4 octets (32 bits) dans la plupart des systèmes de gestion de base de données.

On peut effectuer des opérations mathématiques telles que l'addition, la soustraction et la multiplication sur des colonnes INTEGER.

Méthodes \$ GET et \$ POST :

Ces méthodes sont utilisées dans les différents code php du site afin de pouvoir interagir avec la base de données ou différentes fonctionnalités du site web.

1 | results.php :

- `$_GET['id']` correspond à l'identifiant du test
- `$_GET['page']` correspond au numéro de la page
- `$_GET['limit']` correspond au nombre maximum d'enregistrements / page

2 | transfert_functions :

- `$_POST['field_names']` correspond au contenu des champs du tableau sélectionnés
- `$_POST['brut']` correspond à la valeur recherché dans le tableau
- `$_POST['label']` correspond à la valeur modifiée de brut

3 | update_field_names.php

- `$_POST['field_name']` correspond au nom des champs (frame_date, bench...)

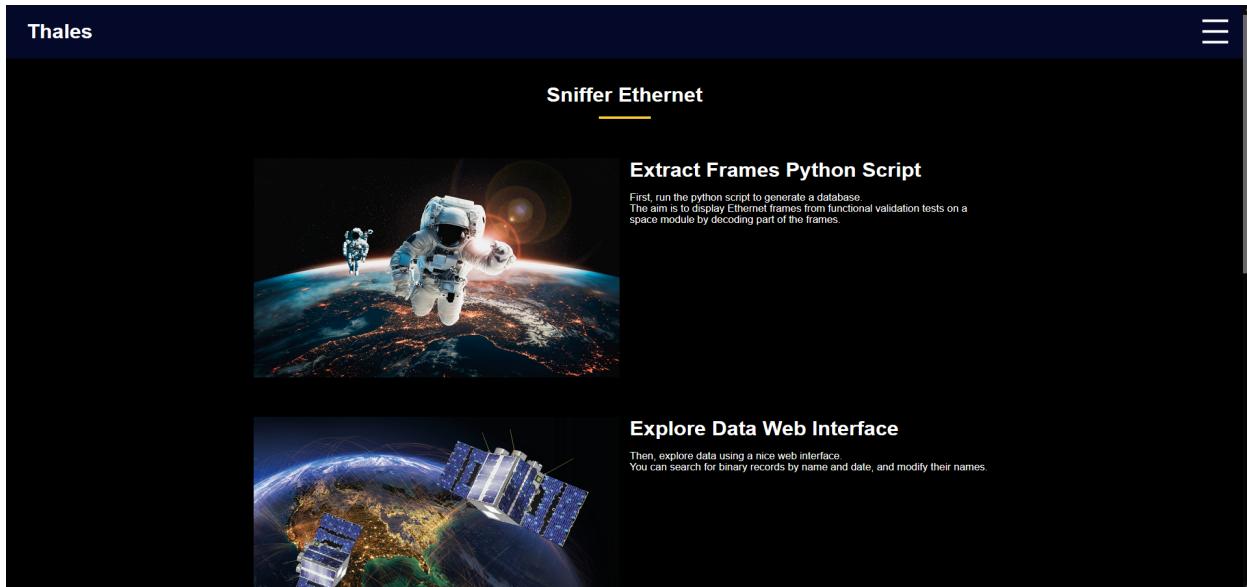
La méthode GET s'affiche dans l'URL lorsqu'elle est exécutée, alors que la POST est transparente. Quand on veut que l'usager puisse modifier facilement les valeurs, on met du `$_GET` dans le cas inverse on mettra du `$_POST`.

exemple de lien : `localhost/results.php?id=1&limit=1&page=3`

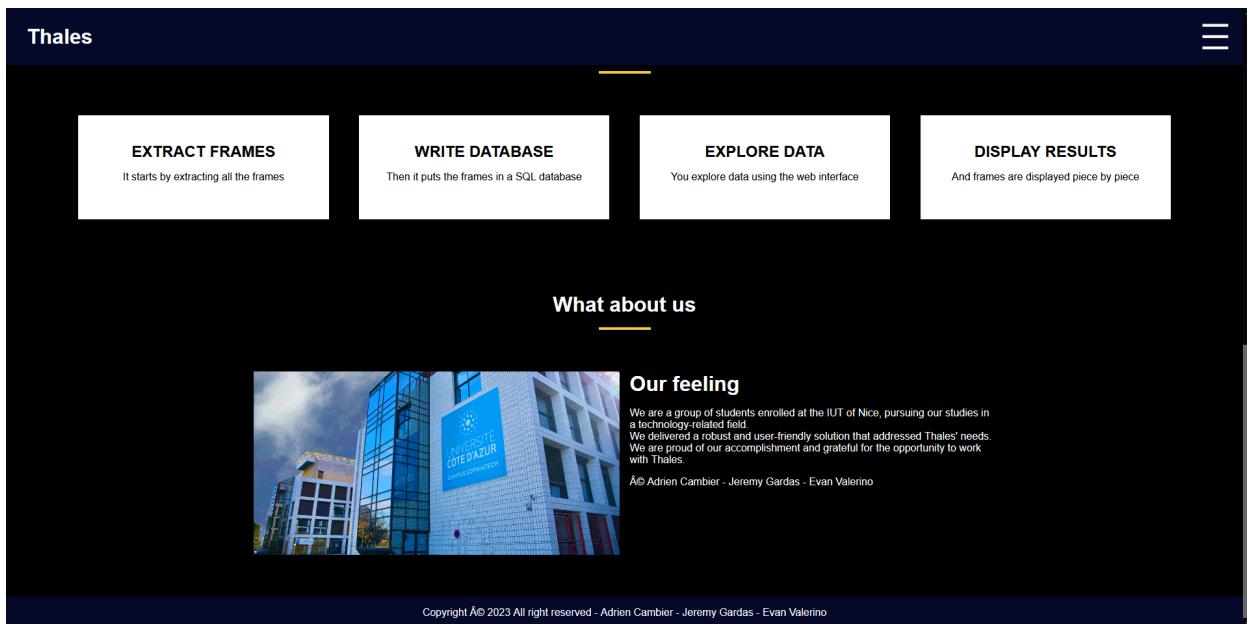
La méthode GET s'affiche après le caractère "?" et les méthodes sont séparées par le caractère "&".

Screenshots du site Web :

Page d'accueil "Home" :

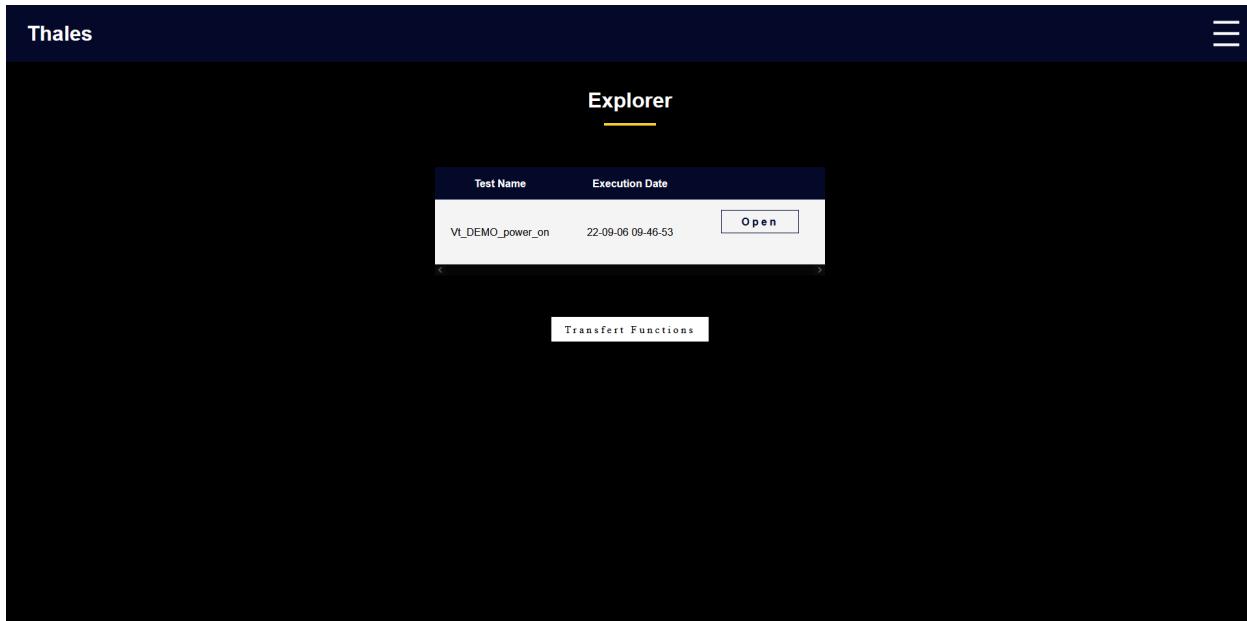


The screenshot shows a dark-themed web page for 'Sniffer Ethernet'. At the top, there is a navigation bar with the Thales logo on the left and a three-line menu icon on the right. Below the header, the title 'Sniffer Ethernet' is displayed above a horizontal line. To the left, there is a large image of two astronauts floating in space above Earth at night. To the right of the image, the text 'Extract Frames Python Script' is shown, followed by a brief description: 'First, run the python script to generate a database. The aim is to display Ethernet frames from functional validation tests on a space module by decoding part of the frames.' Below this section, another image shows a satellite in orbit around Earth. To its right, the text 'Explore Data Web Interface' is displayed, with a note: 'Then, explore data using a nice web interface. You can search for binary records by name and date, and modify their names.'

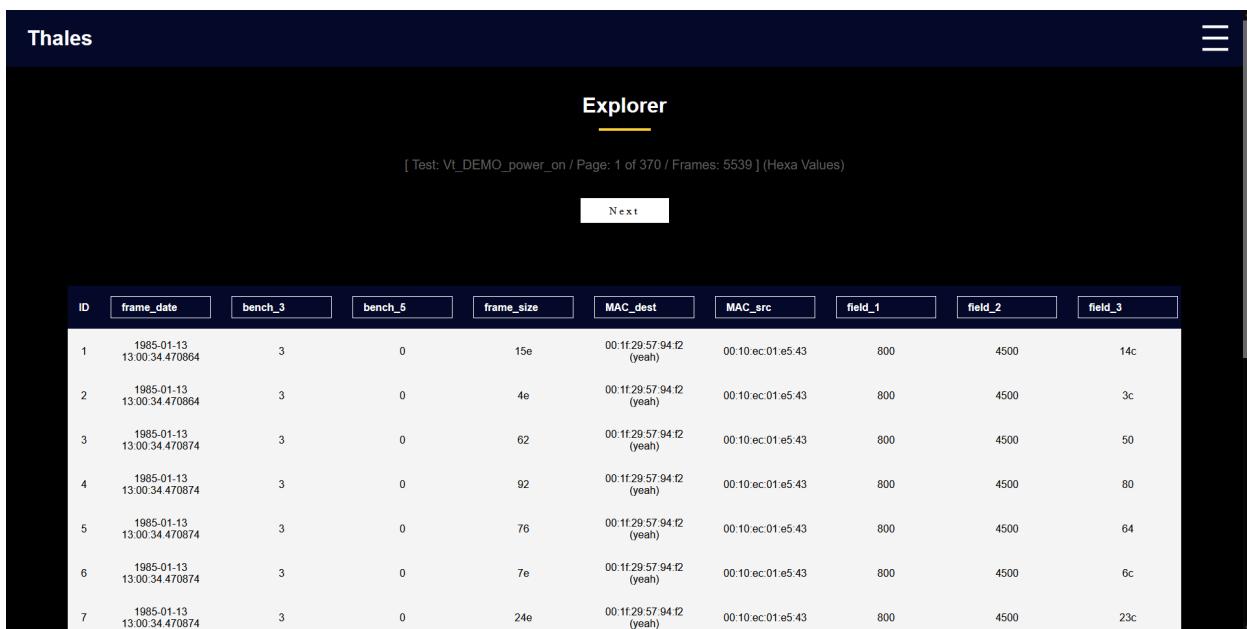


The screenshot shows a dark-themed web page with a navigation bar at the top featuring the Thales logo and a three-line menu icon. Below the header, there are four white rectangular boxes arranged horizontally, each representing a step in the process: 'EXTRACT FRAMES' (described as extracting all the frames), 'WRITE DATABASE' (described as putting the frames in a SQL database), 'EXPLORE DATA' (described as exploring data using the web interface), and 'DISPLAY RESULTS' (described as displaying frames piece by piece). Below these boxes, the heading 'What about us' is centered above a horizontal line. To the left of this line, there is a photograph of a modern building with glass windows and a blue sign that reads 'UNIVERSITÉ NICE CÔTE D'AZUR'. To the right of the line, the heading 'Our feeling' is displayed, followed by a paragraph of text: 'We are a group of students enrolled at the IUT of Nice, pursuing our studies in a technology-related field. We delivered a robust and user-friendly solution that addressed Thales' needs. We are proud of our accomplishment and grateful for the opportunity to work with Thales.' Below this paragraph, the copyright notice 'Â© Adrien Cambier - Jeremy Gardas - Evan Valerino' is visible.

La page d'affichage des enregistrements “Explorer” :



The screenshot shows a dark-themed user interface for a tool named "Thales". At the top center, it says "Explorer". Below that is a table with two columns: "Test Name" and "Execution Date". A single row is visible, containing "VI_DEMO_power_on" and "22-09-06 09:46:53". To the right of this row is a "Open" button. At the bottom of the table is a "Transfert Functions" button.



This screenshot shows a more detailed view of the "Explorer" interface. It includes a header with the test name "VI_DEMO_power_on", the page number "1 of 370", and the total frames "5539". Below this is a "Next" button. The main area is a table with ten columns: ID, frame_date, bench_3, bench_5, frame_size, MAC_dest, MAC_src, field_1, field_2, and field_3. Seven rows of data are listed, each corresponding to a frame entry with specific values for each column.

ID	frame_date	bench_3	bench_5	frame_size	MAC_dest	MAC_src	field_1	field_2	field_3
1	1985-01-13 13:00:34.470864	3	0	15e	00:1f:29:57:94:12 (yeah)	00:10:ec:01:e5:43	800	4500	14c
2	1985-01-13 13:00:34.470864	3	0	4e	00:1f:29:57:94:12 (yeah)	00:10:ec:01:e5:43	800	4500	3c
3	1985-01-13 13:00:34.470874	3	0	62	00:1f:29:57:94:12 (yeah)	00:10:ec:01:e5:43	800	4500	50
4	1985-01-13 13:00:34.470874	3	0	92	00:1f:29:57:94:12 (yeah)	00:10:ec:01:e5:43	800	4500	80
5	1985-01-13 13:00:34.470874	3	0	76	00:1f:29:57:94:12 (yeah)	00:10:ec:01:e5:43	800	4500	64
6	1985-01-13 13:00:34.470874	3	0	7e	00:1f:29:57:94:12 (yeah)	00:10:ec:01:e5:43	800	4500	6c
7	1985-01-13 13:00:34.470874	3	0	24e	00:1f:29:57:94:12 (yeah)	00:10:ec:01:e5:43	800	4500	23c

Thales										
	frame_date	bench_3	bench_5	frame_size	MAC_dest	MAC_src	field_1	field_2	field_3	field_4
5	13:00:34.470874	3	0	76	(yeah)	00:10:8C:01:E5:43	800	4500	64	
6	1985-01-13 13:00:34.470874	3	0	7e	00:1f:29:57:94:f2 (yeah)	00:10:ec:01:e5:43	800	4500	6c	
7	1985-01-13 13:00:34.470874	3	0	24e	00:1f:29:57:94:f2 (yeah)	00:10:ec:01:e5:43	800	4500	23c	
8	1985-01-13 13:00:34.470895	3	0	f6	00:1f:29:57:94:f2 (yeah)	00:10:ec:01:e5:43	800	4500	e4	
9	1985-01-13 13:00:34.470895	3	0	14e	00:1f:29:57:94:f2 (yeah)	00:10:ec:01:e5:43	800	4500	13c	
10	1985-01-13 13:00:34.470906	3	0	d2	00:1f:29:57:94:f2 (yeah)	00:10:ec:01:e5:43	800	4500	c0	
11	1985-01-13 13:00:34.470937	3	0	5e	00:1f:29:57:94:f2 (yeah)	00:10:ec:01:e5:43	800	4500	4c	
12	1985-01-13 13:00:34.470948	3	0	15e	00:1f:29:57:94:f2 (yeah)	00:10:ec:01:e5:43	800	4500	14c	
13	1985-01-13 13:00:34.470979	3	0	62	00:1f:29:57:94:f2 (yeah)	00:10:ec:01:e5:43	800	4500	50	
14	1985-01-13 13:00:34.470979	3	0	24e	00:1f:29:57:94:f2 (yeah)	00:10:ec:01:e5:43	800	4500	23c	
15	1985-01-13 13:00:34.470990	3	0	f6	00:1f:29:57:94:f2 (yeah)	00:10:ec:01:e5:43	800	4500	e4	

Transfert Functions

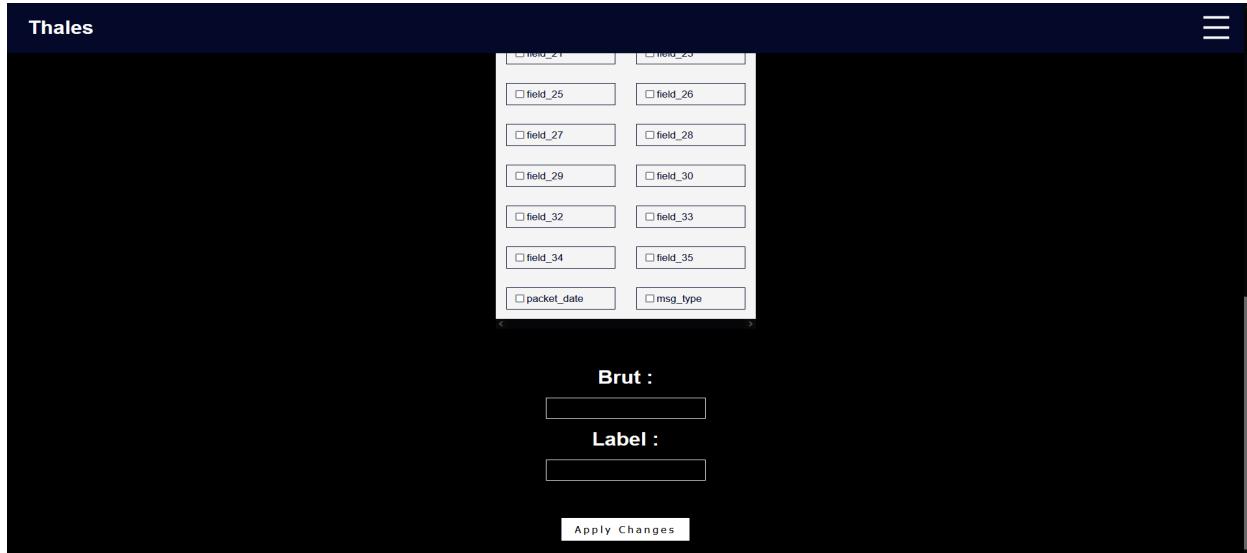
La page destiné à la modification des noms “Transfert Functions” :

Thales

Transfert Functions

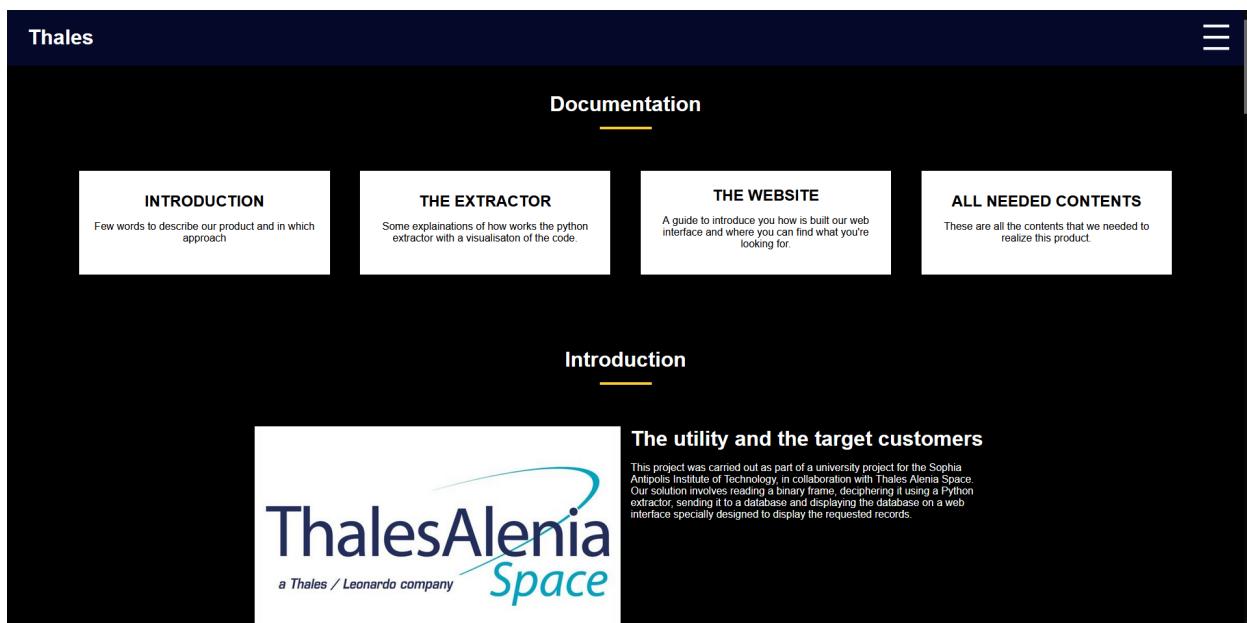
Select the fields to work on

<input type="checkbox"/> frame_date	<input type="checkbox"/> bench_3
<input type="checkbox"/> bench_5	<input type="checkbox"/> frame_size
<input type="checkbox"/> MAC_dest	<input type="checkbox"/> MAC_src
<input type="checkbox"/> field_1	<input type="checkbox"/> field_2
<input type="checkbox"/> field_3	<input type="checkbox"/> field_4
<input type="checkbox"/> field_5	<input type="checkbox"/> field_6
<input type="checkbox"/> IP_src	<input type="checkbox"/> IP_dest
<input type="checkbox"/> field_9	<input type="checkbox"/> field_10
<input type="checkbox"/> field_11	<input type="checkbox"/> field_14



The screenshot shows a dark-themed web application interface. At the top left is the 'Thales' logo. On the right is a three-line menu icon. Below the header is a grid of input fields labeled field_21 through field_35, msg_type, and packet_date. Underneath the grid is a section labeled 'Brut:' with a text input field, followed by a section labeled 'Label:' with another text input field. At the bottom is a 'Apply Changes' button.

La page web de documentation :



The screenshot shows a documentation page with a dark background. At the top left is the 'Thales' logo. In the center is a title 'Documentation'. Below it are four white boxes: 'INTRODUCTION' (Few words to describe our product and in which approach), 'THE EXTRACTOR' (Some explanations of how works the python extractor with a visualisation of the code), 'THE WEBSITE' (A guide to introduce you how is built our web interface and where you can find what you're looking for.), and 'ALL NEEDED CONTENTS' (These are all the contents that we needed to realize this product). At the bottom left is the 'ThalesAlenia Space' logo with the text 'a Thales / Leonardo company'. To the right is a section titled 'The utility and the target customers' with a detailed description of the project's purpose and scope.

Thales

The Extractor

Extractor.py | 12.1 KB

```
import frame
import sql
import argparse
import os
import sys
import termcolor
import bitstring
import macaddress
import datetime
import re
```

This code represents an Ethernet frame extractor from a binary file. It extracts Ethernet frames from the file, performs certain transformations on the extracted data, and adds them to an SQL database.

The code is divided into several parts:

Module Imports: The code begins by importing various necessary modules such as frame (a custom module), sql (an SQL database management module), argparse (a module for command-line argument parsing), os, sys, termcolor (a module for console output coloring), bitstring (a module for bit manipulation), macaddress (a module for MAC address manipulation), and datetime (a module for date and time manipulation).

- Extractor Class: This class represents the extractor itself. It contains several methods to perform the extraction of Ethernet frames from the binary file and add them to the SQL database.
- `__init__(self)` Method: The constructor of the class initializes the instance variables.
- `run()` Method: This method executes the extractor. It calls other methods to parse the command-line arguments, extract frames from the binary file, and add them to the database.
- `parse_cmd_line_args()` Method: This method parses the command-line arguments provided when running the program. It uses the argparse module to parse and validate the arguments.
- `extract_frames_from_file_and_add_them_to_db()` Method: This method extracts frames from the binary file and adds them to the database. It uses the bitstring module to read bits from the binary file and the Frame class (defined elsewhere) to store frame information.

Thales

```
HEADER_SIZE_IN_BYTES = 28
NBR_OF_BYTES_BEFORE_MSG = 90

def __init__(self):
    self.fields = {
```

The Frame class represents an Ethernet frame and contains the specific information about that frame. This class is used in the context of an Ethernet frame extractor from a binary file. Here is a detailed explanation of this class:

Class attributes:

- `HEADER_SIZE_IN_BYTES`: The size of the frame header in bytes (28 bytes).
- `NBR_OF_BYTES_BEFORE_MSG`: The number of bytes before the message in the frame (90 bytes).
- `__init__(self)`: The constructor of the class initializes the instance variables of each Frame instance.

Attributes of each instance:

`fields`: A dictionary representing the different fields of the Ethernet frame with their information. Each field is defined by its name, the number of bits it occupies, its current value, and an optional indication of its usefulness. - `message_type`: The message type associated with the frame. - `packet_date`: The packet date associated with the frame. - `test_name`: The test name associated with the frame. - `test_execution_date`: The test execution date associated with the frame.

The fields dictionary contains different fields with their properties. Each field is identified by a key corresponding to its name. The properties associated with each field are:

- "bits": The number of bits occupied by the field. - "value": The current value of the field. - "useless": An optional indication to indicate if the field is useless in the current context.

This class is used in the context of an Ethernet frame extractor to store the extracted information from each frame. Each Frame instance represents a frame with its specific information, such as source and destination MAC addresses, source and destination IP addresses, frame date, message type, etc.

These pieces of information can be used to perform calculations, transformations, or further analysis on the Ethernet frames extracted from a binary file. The Frame class provides an organized structure to store this information and facilitates access and manipulation of it.

This class can be extended or adapted based on the specific needs of the project or application in which it is used.

```
Sql.py | 5.95 KB
```

Thales

All Needed Contents



Python Programming Language - Simplicity and Power Combined

Python is a dynamic, high-level programming language renowned for its simplicity, readability, and versatility. With its elegant syntax and extensive standard library, Python empowers developers to create robust and efficient solutions for various domains. Whether you're a beginner or an experienced programmer, Python offers a friendly and productive environment for building everything from simple scripts to complex applications.



The Language for Efficient Database Management

SQL (Structured Query Language) is a powerful and standardized language designed for managing and manipulating relational databases. With SQL, developers and data professionals can efficiently store, retrieve, modify, and analyze data stored in databases. It serves as the backbone of modern data-driven applications and plays a crucial role in data management and analysis.

Thales



HTML, CSS, and JavaScript for Web Development

In the world of web development, three essential languages play a crucial role in bringing websites to life. These languages are: HTML, CSS, and JavaScript. Each language serves a unique purpose and works harmoniously to bring websites to life. Let's take a closer look at these languages and understand their significance in web development.



PHP - Empowering Dynamic Web Applications

PHP (Hypertext Preprocessor) is a powerful server-side scripting language widely used for web development. It allows developers to create dynamic and interactive websites by embedding PHP code within HTML. With its extensive features and broad community support, PHP has become one of the most popular languages for web programming.

Copyright © 2023 All right reserved - Adrien Cambier - Jeremy Gardas - Evan Valerino

Conclusion générale

Réponses de nos outils à la problématique :

En conclusion, notre projet de conception d'un extracteur de trame et d'un Sniffer Ethernet s'est déroulé sous la supervision des professionnels de l'entreprise Thalès. Bien que notre équipe initiale de cinq membres se soit réduite à trois au fil du temps, nous avons réussi à atteindre nos objectifs principaux.

Tout d'abord, nous avons développé avec succès un extracteur de trame en utilisant le langage de programmation Python. Ce programme nous a permis d'extraire des données à partir d'un fichier codé en binaire, qui ont ensuite été stockées dans une base de données.

Ensuite, nous avons créé un site web utilisant HTML, CSS et PHP pour visualiser les trames que nous avons extraits. Sur ce site, nous avons mis en place une interface permettant de modifier les noms des enregistrements, de les classer par différents critères tels que le nom et la taille, et de les afficher dans un tableau clair et organisé. De plus, nous avons fourni une documentation détaillée accompagnant le site web, afin d'expliquer les détails techniques de comment notre solution fonctionne.

Dans l'ensemble, notre projet a été couronné de succès grâce à l'expertise et au soutien de l'équipe de professionnels de Thalès. Nous avons pu réaliser un extracteur de trame efficace, stocker les données extraites dans une base de données et créer un site web fonctionnel pour visualiser ces données. Ce projet nous a permis de développer nos compétences en programmation, en conception de bases de données et en développement web.

Nous sommes fiers des résultats obtenus et convaincus que notre travail contribuera positivement aux activités de l'entreprise Thalès. Nous tenons à remercier tous les membres de l'équipe pour leur contribution au projet. Nous espérons que notre solution sera utilisée avec succès et qu'elle ouvrira la voie à de nouvelles opportunités et améliorations dans le domaine des extracteurs de trame et des Sniffers Ethernet.

Retour d'expérience :

Avis des différents participants du projet :

Jeremy Gardas :

Ce projet m'a permis d'apprendre à travailler en groupe et à développer mon esprit d'équipe. J'ai pu mettre mes compétences au service du groupe et profiter de celles des autres.

Adrien Cambier :

Le projet dans sa globalité était intéressant car il a permis de pratiquer différents langages numériques afin de réaliser un projet concret. On a pu voir l'importance du travail en groupe et de la communication car les compétences dans un domaine clé ne suffisent pas pour aboutir de manière pertinente à un projet répondant à la problématique. J'ai pu grandement approfondir mes compétences dans le développement Web ce qui me sera utile plus tard car le web est un domaine qui me passionne grandement.

Evan Valerino :

En effet, au premier regard je pensais que je n'aurais pas eu les capacités pour ce projet, mais malgré mes lacunes, j'ai réussi à m'améliorer et pouvoir apporter ma pierre à l'édifice. Cela m'attriste tout de même de voir deux de mes camarades quitter le navire en si bon chemin. En revanche, malgré ce départ tragique, cela m'a permis d'avoir davantage plus de responsabilités pour le projet. Désormais que le projet est fini, je me pose une question, est ce que cette confiance que j'ai accordé à mes camarades leur était-elle méritée ?
