

SAE 23

Groupe 6 / Sujet 1 :

Application de Covoiturage

Yousef HAMDI
Hugo COSTON
Paul-Emile NGUYEN-TAN-HON
Nantenaina RANDRIANASOLO

CONTENU

1. Contexte
2. Gestion de projet
3. MCD & Maquette
4. Backend (Technique & Fonctionnement)
5. Hébergement
6. Problèmes rencontrés
7. Compétences acquises
8. Source
9. Site final & conclusion

Application Web à usage interne de l'entreprise : de l'équipage de covoiturage domicile(s)-travail **
difficulté medium to strong **

Solution expérimentale de covoiturage pour les étudiants de l'Université des Portes du Jura à Montbéliard. Cartographie des domiciliations, recensement des capacités de covoiturage et mise en contact de l'offre et de la demande. Saisie des données via JSON/CSV et interface web/mobile.

Fonctionnalités : création d'équipages, gestion des trajets, ajout/suppression de véhicules. Accompagné d'une documentation et d'une formation utilisateur

- Presentation + sommaire
- Gestion de projet
- Jour 1
 - Maquette global du site
 - Shema site
 - Debut dev
- Jour 2
 - Dev 8h - 14h
 - Hors sujet
 - Donc 2 jour perdu
 - Remise en place avec le prof
 - Mise jour maquette + validation
- Jour 3 + nuit
 - refaire la majorité du site à la suite d'un malentendu
 - Nuit dev global du site
 - Concentration sur le backend du site → le faire fonctionner
- Finissement du site le week-end

Notre organisation

Au debut: Yousef, Nantenaina et Hugo : Maquette Site

Paul-Emile : CSV, MCD, MPD

Fin :

Yousef, Nantenaina et Hugo : Backend (php+sql) + css

Paul-Emile: Base de données, schéma global, documentation

Début ..



Connexion ou inscription

Profil

Covoiturage

Doc

Bla Bla Bla

Formulaire de Covoiturage

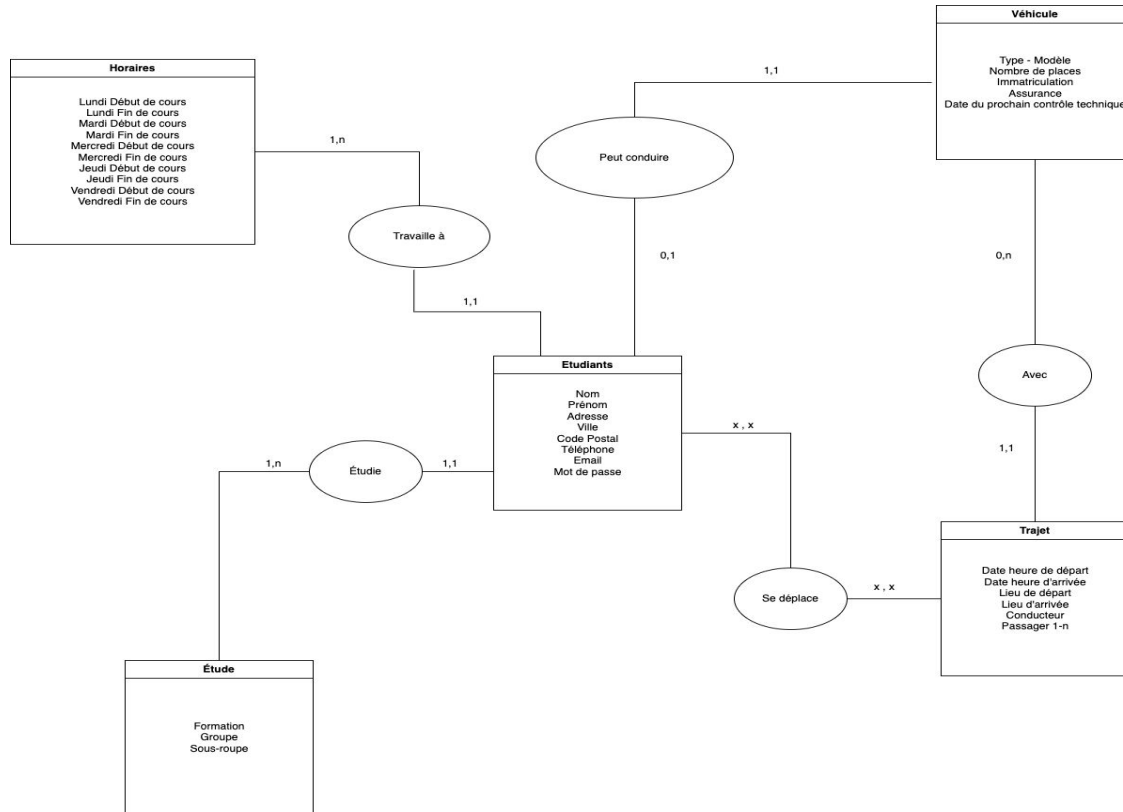
Les champs obligatoires sont suivis par un *.

Informations de contact

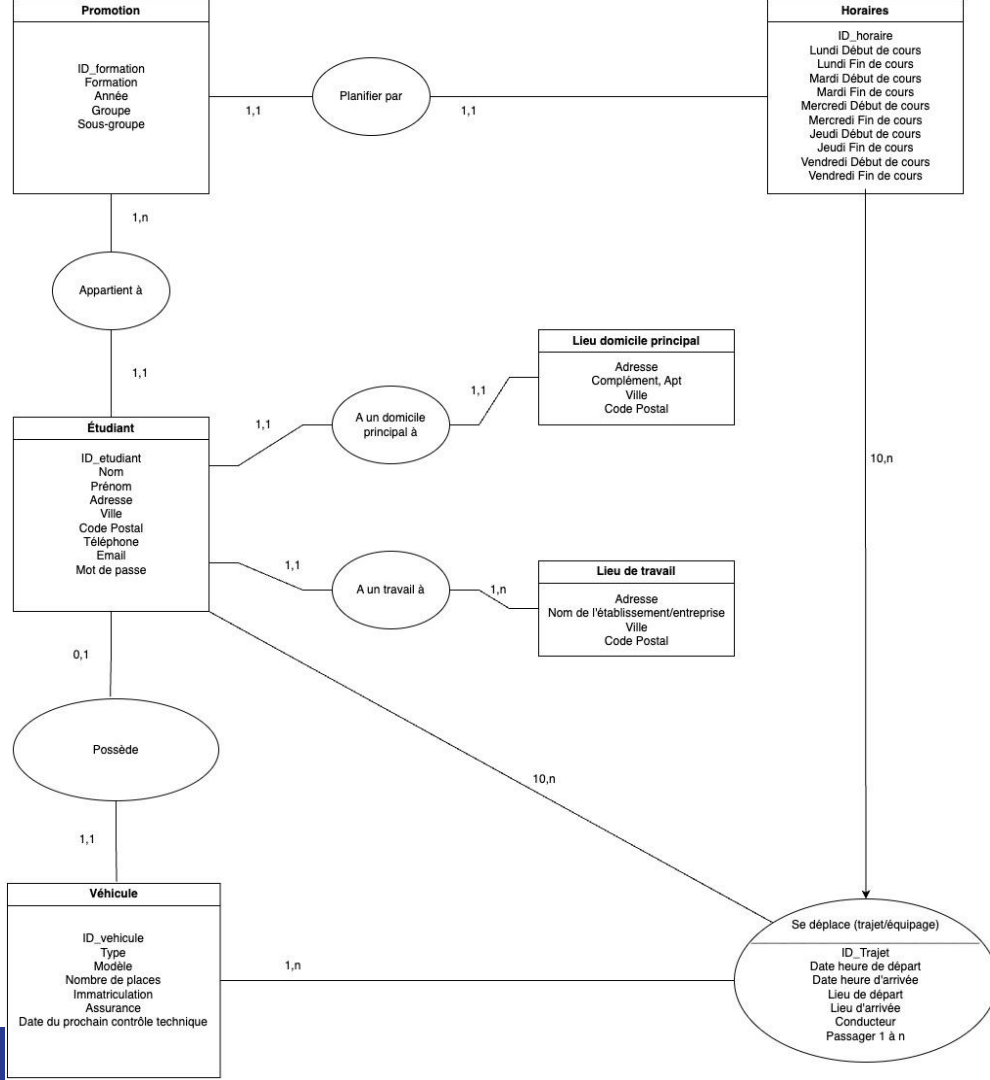
Qualité

- ☐ Passager
- ☐ Conducteur

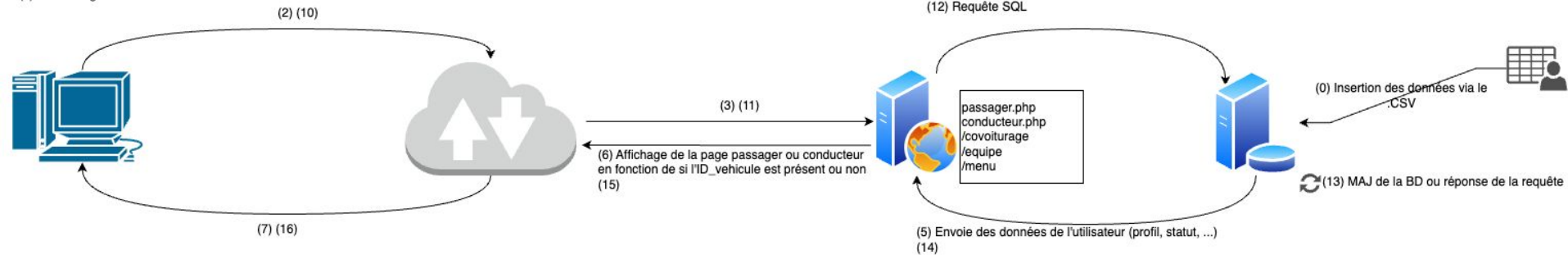
Ancien MCD



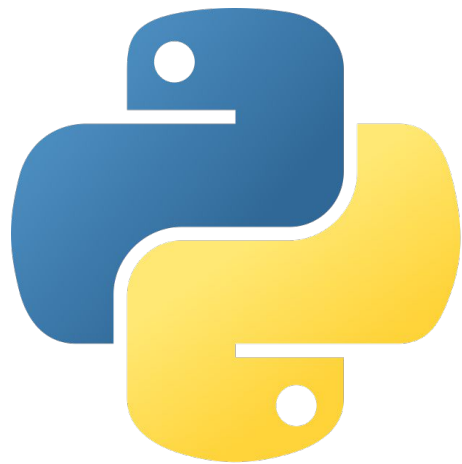
MCD



- (1) Connexion au serveur avec ses identifiants
- (9) (a) Menu
- (b) Équipe
- (c) Covoiturage



CSV



id_etudiant	nom	prenom	adresse	ville	code_postal	telephone	email	mot_de_passe	id_vehicule	type	modele	nb_place
1	NGUYEN-TAN-HON	Paul-Emile	37 Rue Georges Clemenceau	Montbéliard	25200	610641293	paulemile25000@gmail.com	4CGy6i7	NULL	NULL	NULL	NULL
2	COSTON	Hugo	3 rue des deux fontaines	Leberain	90100	616525318	costonhugo@gmail.com	81TeF3S5E	6	Suzuki	Chadine	
3	RANDRIANASOLO	Nantenaina	4 rue des trouduc coston	Montbéliard	47800	615848956	r.nantenaina@outlook.fr	P4s2@mDp!!	NULL	NULL	NULL	NULL
4	HAMDI	Yousef	3 Avenue Charles Bohn	Belfort	90000	612544356	yousefhamdi1804@gmail.com	kjx9B@Ae!	NULL	NULL	NULL	NULL
5	ARAGOU	Souleyman	37 Rue Georges Clemenceau	Montbéliard	25200	668966325	aragou.souleyman@yahoo.com	2i2IP01Lu3	NULL	NULL	NULL	NULL

Script Python

```
#!/usr/bin/env python
```

```
import cgi
import cgitb
import os
import pandas as pd
import pymysql
```

```
cgitb.enable()
```

```
def connect_to_database():
    connection = pymysql.connect(
        host='mysql_server',
        user='penguye2',
        password='mdp',
        database='penguye2'
    )
    return connection
```

```
def import_data_to_tables(connection, csv_file, table_mapping):
    data = pd.read_csv(csv_file)
    cursor = connection.cursor()
    for table, columns in table_mapping.items():
        table_data = data[columns]
        table_columns = ', '.join(table_data.columns)
        query = f"INSERT INTO {table} ({table_columns}) VALUES ({', '.join(['%s']*len(table_data.columns))})"
        data_values = [tuple(row) for row in table_data.values]
        cursor.executemany(query, data_values)
        connection.commit()
    cursor.close()
```

```
def main():
    connection = connect_to_database()
    table_mapping = {
        'etudiant': ['id_etudiant', 'nom', 'prenom', 'adresse', 'ville', 'code_postal', 'telephone', 'email', 'mot_de_passe'],
        'horaire': ['id_horaire', 'id_travail', 'Lu_debut-cours', 'Lu_fin-cours', 'Ma_debut-cours', 'Ma_fin-cours', 'Me_debut-cours', 'Me_fin-cours', 'Je_debut-cours', 'Je_fin-cours', 'Ve_debut-cours'],
        'trajet': ['id_trajet', 'date_heure_depart', 'date_heure_arrivee', 'lieu_depart', 'lieu_arrivee', 'conducteur', 'passager1', 'passager2', 'passager3', 'passager4', 'passager5', 'passager6', 'passager7'],
        'travail': ['id_etudiant', 'id_travail', 'formation', 'promotion', 'groupe', 'sous_groupe'],
        'vehicule': ['id_etudiant', 'id_vehicule', 'tvpe', 'modele', 'nb_places', 'immatriculation', 'assurance', 'date_prochain_controle_technique']
    }
```

```
    }
    csv_file = 'SAe23_-_sae23_1.csv'
    import_data_to_tables(connection, csv_file, table_mapping)
    connection.close()
if __name__ == "__main__":
    main()
```

Backend (SQL)

```
isset($_POST['se_retirer'])) {  
    // Requête pour supprimer l'étudiant connecté de la liste  
    $sql_supprimer_nom_emetteur = "UPDATE etudiant SET nom_emetteur = NULL WHERE nom = '$user_id'";  
    if ($conn->query($sql_supprimer_nom_emetteur) === TRUE) {
```

```
if (empty($changementmdp) || empty($nchangementmdp) || empty($rnchangementmdp)) {  
    $errors[] = "Veuillez remplir tous les champs.";  
} else {  
    $results = mysqli_query($conn, "SELECT mot_de_passe FROM etudiant WHERE nom = '$session'");  
    $row = mysqli_fetch_assoc($results);  
    $rowval = $row['mot_de_passe']; // Correction : Utilisation de la colonne mot_de_passe
```

```
// récupérer les étudiants disponibles pour la proposition avec leur groupe et sous-groupe de la table travail  
$sql = "SELECT etudiant.*, travail.groupe, travail.sous_groupe  
FROM etudiant  
LEFT JOIN travail ON etudiant.id_travail = travail.id_travail  
WHERE etudiant.id_vehicule IS NULL OR etudiant.id_vehicule = ''";  
$result = $conn->query($sql);
```

Backend (SQL)

```
// Récupérer les personnes sélectionnées
if (isset($_POST['personne']) && is_array($_POST['personne'])) {
    // Supprimer les valeurs existantes dans les colonnes nom_emetteur et etudiant_emetteur
    $sql_reset_emetteur = "UPDATE etudiant SET nom_emetteur = NULL, etudiant_emetteur = NULL";
    $conn->query($sql_reset_emetteur);

    // Enregistrer les sélections dans la base de données
    foreach ($_POST['personne'] as $personne_id) {
        // Insérer la proposition avec l'étudiant émetteur
        $sql_insert_proposition = "UPDATE etudiant SET nom_emetteur = '$nom_etudiant' WHERE id_etudiant = '$personne_id'";
        $conn->query($sql_insert_proposition);
    }
}
```

Hebergement site Web



000webhost
POWERED BY HOSTINGER

ngrok

Problèmes rencontrés

Création du formulaire d'inscription → malentendu

Ajout/modification de données dans la base de données

Connexion de MySQLi et PHPMyadmin

Fonctionnement du CSS

-Comment rediriger les users vers leurs page selon leurs type de compte

groupe 6 sujet 1

13h30
groupe 2 13h30

Le groupe

- groupe 7 14h30 ou 1

login: User: Param/Conduite.PMP
Admin.PMP

mon
profil
mdp
compte

Equipe
mon
equipe (2)
Affiche
Session

Cov.
prop.
dispo

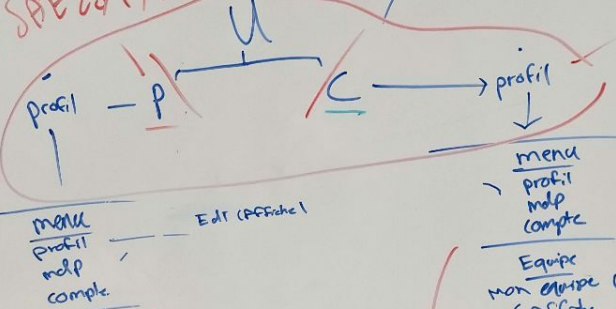
Admin
ins. cov
liste users + c

groupe 7 sujet 8
client

prop: faire un filtre (option)

A SAE 25/1 accuei.pmp

SAE 25/1 accuei.pmp



Edt (Affiche)

Equipe
mon
equipe (2)

Affiche
Session

Cov.
prop.

Creation d'equipe

Rechercher les adresses

Rechercher les adresses

envoyé

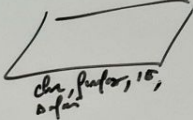
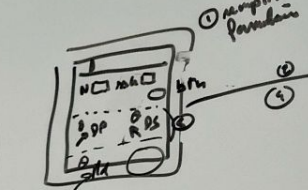
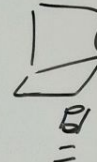


schéma global de



med = Modèle Conceptuel de Données

= 10T
ne s'ajoute pas
dans le MPD
on DPM

Groupe
GB = 2

LK = 2 → 1d

Cote ≤

Affichage: Query → (Tout groupe-c == groupe-p)
id_duchant où Query == TRUE

Travel

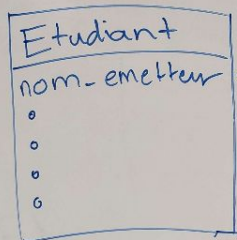
Objectif première:

Conducteur propose aux
Passagers

Version 1

Passager

Les passagers reçoivent
la demande



Conducteur

Default: Il peut créer
une demande aux
passagers

↳ Il sélectionne les
passagers ⊗ nom_1
⊗ nom_2
⊗ nom_3

Problème: → "One-way"

Les Passagers "peuvent pas envoyer une demande
vers les conducteurs en directe (Par une recherche spécifique)
* mais, ils peuvent afficher une liste de TOUTS les conducteurs

Objectif première:

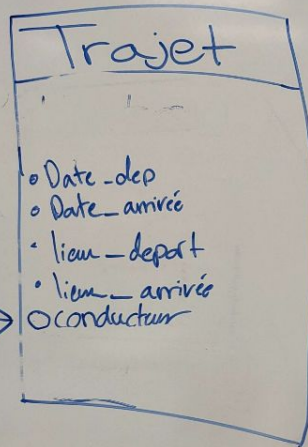
Conducteur propose aux
Passager et Passager
proposé aux conducteurs

Version 2

Conducteur

- Propose un
trajet

- rejoindre un trajet
qui n'a pas de conducteur



Passager

- Rejoindre un
trajet

- Créer un trajet
sans chauffeur



Compétence acquises

Nantenaina : Gestion projet + crise/ SQL théorique/ Backend & Frontend /
Maquette / Hébergement / Github

Paul-Emile : Gestion crise / MCD détaillé / configuration PHPMyAdmin avec
plusieurs tables / SQL Théorique / CSV

Hugo : Gestion crise / SQL théorique / CSV

Yousef : Gestion crise / SQL théorique / Maquette / Backend

Auto-Evaluation

Étape	Non -fait	En cours (date/heure début)	Terminé (date/heure fin)
Fichier CSV rempli (colonnes)		Lundi 8H45	Lundi 14H00
Mise en page du site web (brouillon sur feuille)		Lundi 9H45	Lundi 14H00
Base de données (PHPMyAdmin) rempli		Lundi 11H15	Lundi 14H00
Diagramme MCD		Lundi 13H45	Jeudi 15H00
Création du site web global (HTML, CSS, PHP, JS)		Lundi 13H00	
Remplissage du CSV (avec 5 profils)		Lundi 17H00	Mardi 8H45
Envoie de données via le formulaire d'inscription	X	Lundi 17H00	
Création de la page de connexion		Jeudi 8H00	Jeudi 15H00
Ajout et suppression du véhicule	X	Vendredi 8H00	Vendredi 14H00
Mise en place des trajets		Jeudi 15H00	Vendredi 10H00

Ressource

CHATGPT

PHP

DRAW.IO

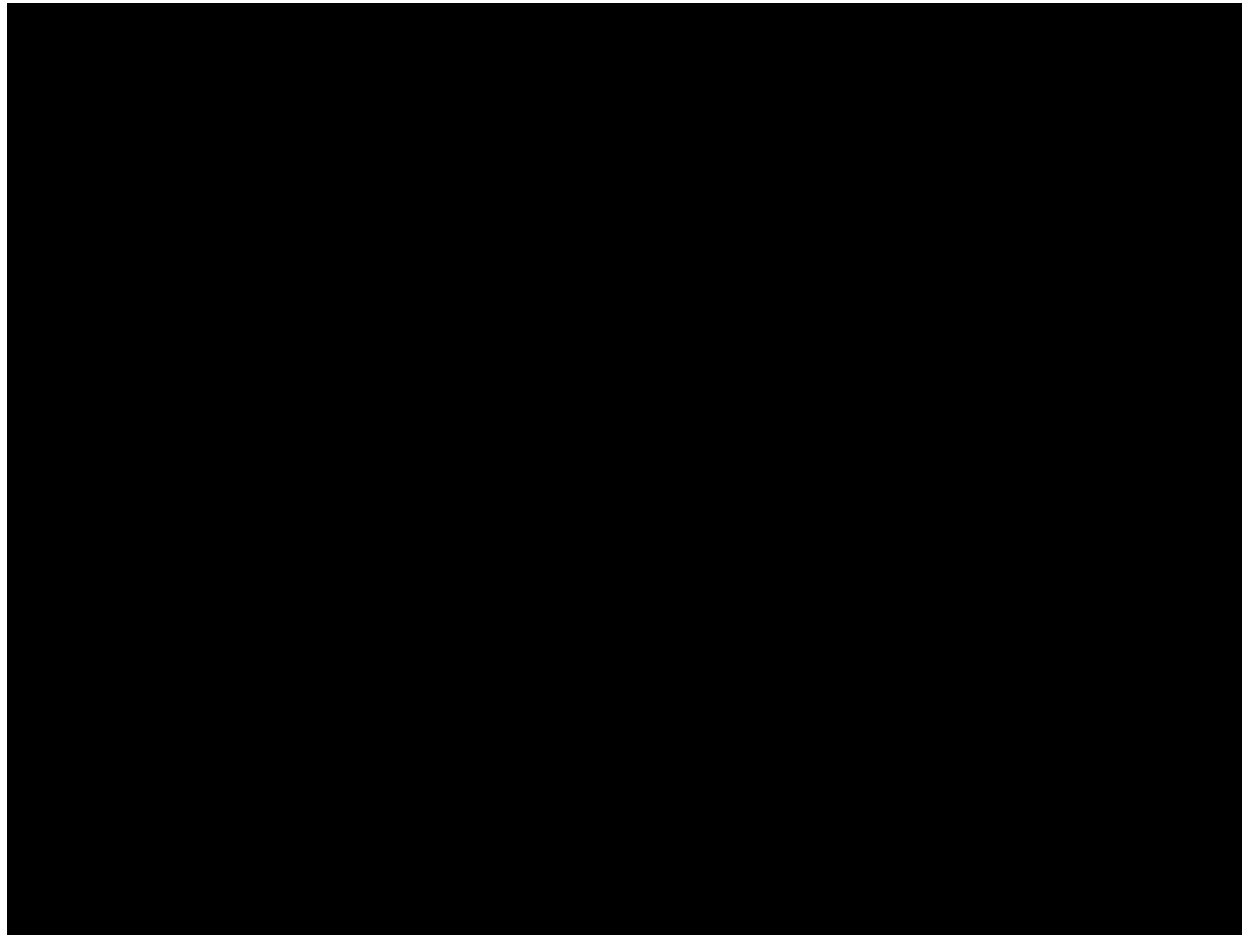
VSCODE

XAMPP

FILEZILLA

OBS

Site final



CONCLUSION

Avez-vous des questions ?

Remerciement à : Philippe CANALDA et Eugen DEDU