Compte rendu

Haddad selim, Coquerel Leo, Farsi Seddiq

Partie 1 : Gestion des Absences

Cette section du projet vise à créer une solution complète pour la gestion des absences en milieu éducatif. Elle se compose de plusieurs fichiers PHP, HTML et CSS, ainsi qu'un script SQL pour la base de données.

index.php: Gère l'authentification des enseignants (login). Un enseignant se connecte avec un nom d'utilisateur et un mot de passe stockés dans la base de données.

appel.php: Permet aux enseignants de faire l'appel. Il affiche la liste des étudiants et permet de marquer leur présence, absence justifiée (ABJ) ou absence injustifiée (ABI) pour un module et un horaire spécifiques. Les données sont ensuite insérées dans la base de données.

consultation.html: Fournit une interface utilisateur pour consulter les absences. Les utilisateurs peuvent rechercher des absences par nom, prénom ou numéro d'étudiant, ainsi que par période (dates de début et de fin).

consultation.php: Traite la requête de consultation et affiche les absences correspondantes en extrayant les données de la base de données. Il permet également de modifier le statut des absences.

update.php: Un script PHP qui gère la mise à jour du statut des absences directement depuis l'interface de consultation.

style.css: Fournit le style global pour les pages de gestion des absences, assurant une présentation cohérente.

SAE23.sql: Le script de base de données (SQL Dump) qui définit la structure de la base de données SAE23, incluant les tables Absences, Enseignant et Etudiant, avec leurs relations et clés.

L'ensemble de cette partie utilise PHP pour la logique métier et l'interaction avec la base de données MySQL (via mysqli), et HTML/CSS pour l'interface utilisateur.

Partie 2 : Slide Puzzle

Cette partie présente un jeu de "Slide Puzzle" (casse-tête coulissant) classique de 15 pièces.

Puzzle.html: Contient la structure HTML du jeu, y compris le tableau qui représente la grille du puzzle et les boutons individuels.

Puzzle.css: Définit le style visuel du puzzle, des boutons et de l'agencement général, incluant les règles de positionnement des images de fond pour chaque pièce.

image.jpg et puzzle.png: Les images utilisées pour le puzzle et l'icône du titre. Le jeu utilise une seule image découpée dynamiquement en arrière-plan pour chaque pièce du puzzle.

La logique du jeu (mélange des pièces, détection des mouvements valides, mise à jour de l'affichage, comptage des coups) est implémentée en JavaScript directement intégré dans Puzzle.html, manipulant le DOM pour animer les pièces.

Partie 3 : Maître Yoda & Luke (Tri de mots)

Cette section est un défi de programmation axé sur la manipulation de chaînes de caractères et la recherche de motifs.

tri.php: Le cœur de cette partie. Il contient un ensemble de mots et implémente un algorithme pour trouver la plus longue chaîne de mots où les trois dernières lettres d'un mot correspondent aux trois premières lettres du mot suivant. Il s'agit en substance d'une recherche du plus long chemin dans un graphe dirigé où les nœuds sont les mots et les arêtes existent si les suffixes/préfixes correspondent.

tri.css: Applique un style minimaliste pour rendre la page affichant le résultat de l'algorithme plus lisible.

Cette partie démontre des compétences en algorithmique et en programmation PHP pour des tâches de traitement de texte et de graphes.

Partie 4 : TP JavaScript : Gestion d’Événements et Interactions Utilisateur

Cette partie est divisée en deux sous-projets ou exercices pratiques, probablement réalisés par Leo et Selim, tous deux centrés sur la manipulation du DOM et la gestion des événements en JavaScript.

4.1. TP JavaScript (Leo)

TPLeo.html: La page HTML servant de base pour les exercices de Leo.

TP\_Leo.css: Les styles CSS spécifiques à ce TP.

JSLeo.js: Le fichier JavaScript implémentant diverses interactions :

Gestion d'événements avec DOM Level 0 (onclick) et DOM Level 2 (addEventListener), montrant les différences.

Gestion des événements de souris (survol, déplacement, clic, entrée/sortie de zone).

Gestion des événements clavier (saisie de texte, détection de touches).

Interactions avancées comme la prévention du comportement par défaut des formulaires et des liens.

Un détecteur de mouvement qui analyse les tracés de souris pour identifier des lignes horizontales, verticales, ou des gribouillis.

TP\_Leo.ipynb et TPJS\_Leo.pdf: Des documents pédagogiques (Jupyter Notebook et PDF) détaillant les prérequis, les rappels techniques (DOM, événements) et les instructions pour les exercices pratiques.

4.2. TP JavaScript (Selim)

TPSelim.html: La page HTML pour les exercices de Selim.

TP\_Selim.css: Les styles CSS correspondants, incluant l'utilisation de variables CSS pour la gestion des thèmes.

JSSelim.js: Le fichier JavaScript qui implémente :

Un compteur numérique simple.

La validation côté client d'un formulaire (par exemple, longueur minimale du nom).

Une application de liste de tâches (CRUD basique : ajout, suppression, marquage comme terminé).

Un sélecteur de thème clair/sombre avec persistance de l'état via localStorage.

Une fonctionnalité de glisser-déposer pour un élément, en assurant que l'élément reste dans sa zone parente.

TPSelim.ipynb et TPSelim.pdf: Des documents pédagogiques (Jupyter Notebook et PDF) fournissant le cadre du TP, des rappels sur HTML/CSS/JS et le DOM, et les étapes pour chaque exercice.