



**INTRODUCTION À L'INTERNET
INF16107 (MS)**

UNIVERSITÉ DU QUÉBEC À RIMOUSKI

DÉPARTEMENT DE MATHÉMATIQUES, D'INFORMATIQUE ET DE GÉNIE

RAPPORT
Travail Pratique 2

ÉQUIPE :

Mbaye & Sokhna Tahiratou

Mongbo & Raissa Asso Dende

Kraïdy & Adjobi Thierry Henri

PROFESSEUR :

Yacine Yaddaden, Ph. D.

Date : 22 décembre 2024

Table des matières

1	Introduction	3
2	Objectifs	3
3	Implémentation Technique	3
3.1	Langages et Technologies Utilisés	3
3.2	Architecture de l'Application	3
4	Fonctionnalités	4
4.1	Gestion des Catégories	4
4.2	Affichage des Images et Descriptions	4
4.3	Validation des Sélections	4
4.4	Navigation entre les Questions	4
5	Défis rencontrés	4
5.1	Problèmes techniques et solutions	4
5.2	Adaptations et améliorations apportées	5
6	Conclusion	5
6.1	Résumé des réalisations	5
6.2	Perspectives d'avenir	5

1 Introduction

Ce projet a pour objectif la création d'une application de quiz interactive permettant aux utilisateurs de sélectionner une catégorie et de répondre à des questions associées à des images. Le projet vise à offrir une expérience utilisateur engageante et éducative tout en mettant en pratique des concepts fondamentaux en développement web.//

Pour ouvrir le fichier d'abord téléchargez le dossier TrafficSignWebApp dans htdocs, ensuite sur votre navigateur entrez ce lien <http://localhost/TrafficSignWebApp/start.html>

2 Objectifs

- Développer une interface utilisateur intuitive.
- Implémenter une gestion dynamique des catégories, images et descriptions à l'aide de JavaScript.
- Valider les choix de l'utilisateur avant de passer aux étapes suivantes.
- Intégrer des données externes à l'application via des requêtes HTTP.

3 Implémentation Technique

3.1 Langages et Technologies Utilisés

L'application a été développée en utilisant les technologies suivantes :

- HTML : pour la structure du contenu.
- CSS : pour le style et la présentation.
- Visual studio Code : comme environnement de développement.
- JavaScript : Gestion dynamique des données et interactions utilisateur
- AJAX et JQuery : Simplification des appels asynchrones et manipulation du DOM.
- Twitter Bootstrap : Création d'une interface responsive et esthétique.
- Xampp : Plateforme de développement local pour héberger l'application et l'API REST.

3.2 Architecture de l'Application

L'application est structurée autour des composants suivants :

- **Frontend** : Une section pour afficher les catégories.
Une section pour afficher l'image actuelle.
Une section pour les propositions de descriptions.
Un bouton pour naviguer vers la prochaine question.

- **Backend (simulé)** : Données exposées via une API locale (catégories, images, descriptions).

4 Fonctionnalités

4.1 Gestion des Catégories

- Récupération des catégories depuis une API REST via une requête ajax.
- Affichage dynamique des catégories sous forme de boutons radio dans une section dédiée.

4.2 Affichage des Images et Descriptions

- Chargement des images et leurs descriptions depuis l'API.
- Affichage de la description correcte mélangée à des propositions incorrectes pour un choix interactif.

4.3 Validation des Sélections

- Validation obligatoire des choix de catégorie et description avant de permettre à l'utilisateur de passer à l'image suivante.
- Alertes informant l'utilisateur en cas de non-sélection.

4.4 Navigation entre les Questions

- Sélection aléatoire de 10 images parmi les données récupérées.
- Navigation cyclique avec le bouton "Suivant" pour passer d'une image à une autre.

5 Défis rencontrés

5.1 Problèmes techniques et solutions

Au cours du développement, plusieurs problèmes techniques ont surgi :

- **Chargement asynchrone des données** : Les données ne s'affichaient pas correctement car les appels étaient asynchrones. Nous avons utilisé ajax avec des fonctions de rappel pour garantir le traitement des données après leur récupération.
- **Validation utilisateur manquante** : Les utilisateurs pouvaient cliquer sur "Suivant" sans effectuer de sélection. Nous avons ajouté une vérification dans le gestionnaire d'événements pour empêcher la navigation sans sélection préalable.

5.2 Adaptations et améliorations apportées

Pour surmonter les défis rencontrés, plusieurs adaptations ont été effectuées :

- Amélioration des performances grâce à la réduction des appels redondants à l'API.
- Mise en place d'une validation stricte pour éviter les erreurs utilisateur.
- Refonte de l'interface utilisateur avec des styles modernes et responsive via Bootstrap.
- Optimisation de la logique de navigation entre les questions pour une meilleure fluidité.

6 Conclusion

6.1 Résumé des réalisations

Ce rapport a détaillé la création d'une application interactive de quiz éducatif. Le site a été conçu pour offrir une expérience engageante et intuitive en utilisant des technologies modernes telles que jQuery, AJAX, et Bootstrap. À travers une gestion dynamique des catégories, des images et des descriptions, l'application garantit une navigation fluide et interactive. Grâce aux solutions adaptées aux défis rencontrés, notamment l'intégration des données, le projet a été mené à terme avec succès.

6.2 Perspectives d'avenir

À l'avenir, nous envisageons d'intégrer une base de données réelle pour une gestion plus robuste et sécurisée des données, permettant ainsi d'améliorer la fiabilité et la persistance des informations. De plus, la mise en œuvre d'un mode multijoueur offrirait une expérience compétitive et interactive, renforçant l'engagement des utilisateurs. Enfin, l'ajout d'un support multilingue élargirait l'accessibilité de l'application à un public international, consolidant sa portée et son attractivité.

Références

- **Documentation jQuery** : <https://api.jquery.com/>
- **Guide Twitter Bootstrap** : <https://getbootstrap.com/docs/>
- **Tutoriels AJAX** : https://www.w3schools.com/xml/ajax_intro.asp
- **XAMPP** : <https://www.apachefriends.org/>
- **Ressources JavaScript** : <https://developer.mozilla.org/fr/docs/Web/JavaScript>