Programmation Web I : Travail pratique 2

20 points

Partie 1 : Jeu du Hi-Low

Objectif :

Développer une application Web en JavaScript permettant de jouer au jeu du Hi-Low, où l'ordinateur génère un nombre aléatoire entre 1 et 100. Le joueur doit deviner ce nombre. Après chaque essai incorrect, l'application indique si le nombre à deviner est plus élevé ou plus bas que l'essai du joueur. Le nombre d'essais effectués doit être comptabilisé et affiché. Le jeu se termine lorsque le joueur trouve le bon nombre, et un message de victoire s'affiche. Le joueur doit pouvoir recommencer une nouvelle partie après avoir gagné.

Fonctionnalités attendues :

1. Interface utilisateur :
   - L'interface doit être simple, claire et intuitive, avec un champ de saisie permettant au joueur de saisir un nombre pour chaque essai;
   - Un bouton pour soumettre chaque essai;
   - Affichage d'une indication sous la forme de « plus haut que » ou « plus bas que » pour chaque essai infructueux du joueur, en fonction de la comparaison avec le nombre généré par l'ordinateur;
   - Un compteur affichant le nombre d'essais effectués lorsque la partie est terminée;
   - Un message de victoire lorsque le joueur trouve le bon nombre.

2. Logique du Jeu :
   - Génération d'un nombre aléatoire entre 1 et 100 au début du jeu;
   - Comparaison de chaque essai du joueur avec le nombre généré;
   - Affichage d'une indication (« plus haut que » ou « plus bas que ») en fonction de l'essai du joueur;
   - Incrémentation et affichage du compteur d'essais après chaque essai;
   - Affichage d'un message de victoire lorsque le joueur trouve le bon nombre et possibilité de recommencer une nouvelle partie.

3. Statistiques :
   - Le nombre d'essais effectués doit être comptabilisé, mis à jour après chaque essai et affiché à l'utilisateur seulement à la fin de la partie. Ce compteur doit être réinitialisé lorsque le joueur recommence une nouvelle partie.

Grille d'évaluation :

Critère	Points	Description
Fonctionnalités :		
• Interface utilisateur	/5	Interface fonctionnelle, claire et esthétique, comprenant un champ de saisie, un bouton pour soumettre chaque essai, et un affichage des indications.
• Génération du nombre aléatoire	/5	Génération d'un nombre aléatoire entre 1 et 100 uniquement au début du jeu.
• Comparaison et indications	/10	Comparaison correcte des essais, indications « plus haut que » ou « plus bas que » avec le dernier essai affiché.
• Compteur d'essais	/10	Compteur d'essais fonctionnel et affiché correctement à la fin du jeu.
• Message de victoire	/5	Affichage d'un message de victoire lorsque le nombre est trouvé.
• Rejouer	/5	Le jeu doit permettre au joueur de recommencer une nouvelle partie après une victoire, en réinitialisant le compteur et générant un nouveau nombre aléatoire.
Qualité globale du code	/10	Code bien structuré, lisible, avec des noms de variables explicites et une indentation correcte.
Total :	**/50**	

Partie 2 : Formulaire de party de Noël

Objectif :

Développer un formulaire Web en JavaScript permettant de recueillir les préférences des invités pour un party de Noël. Le formulaire doit être interactif et dynamique, s'adaptant aux choix de l'utilisateur au fur et à mesure de la saisie.

Fonctionnalités attendues :

1. Interface Utilisateur :
   a) Un champ obligatoire pour entrer le nom de l'invité.
   b) Un champ radio obligatoire pour indiquer si l'invité sera présent ou non.
      - Si « Non » est sélectionné, le formulaire permet de soumettre immédiatement.
      - Si « Oui » est sélectionné, des options supplémentaires (choix de repas, accompagnateur, activités) doivent être affichées.
   c) Des champs pour choisir son repas (un seul choix parmis « Poulet aux ananas », « Bœuf braisé » et « Tourtière végane »).
   d) Un champ radio pour indiquer si l'invité sera accompagné ou non.
   e) Si « Oui » est sélectionné, des champs doivent être affichés pour saisir le nom de l'accompagnateur et son choix de repas.
   f) Une sélection obligatoire de deux activités parmi : Traîneaux à chien, Noël disco, Karaoké, Glissades, Laser tag, Casino.
   g) Un bouton pour soumettre le formulaire.

2. Logique du formulaire :
    - Par défaut, seules les sections a), b) et g) sont visibles lors du chargement du formulaire.
    - Si l'utilisateur sélectionne « Non » pour la question b) sur la présence, les sections c), d), e), f) et les champs associés ne doivent pas être visibles. Le formulaire peut être soumis immédiatement.
    - Si l'utilisateur sélectionne « Oui » à la question b) sur la présence, les sections c), d) et f) doivent devenir visibles, permettant à l'utilisateur de faire son choix de repas, d'indiquer s'il sera accompagné, et de sélectionner les activités.
    - Si l'utilisateur sélectionne « Oui » pour l'accompagnement (section d), les champs concernant l'accompagnateur (section e) doivent apparaître pour que l'utilisateur puisse saisir ces informations.
    - Lors de la soumission du formulaire :
        - Tous les champs texte visibles (nom, repas, activités, etc.) doivent être remplis. Si un champ est vide, un message d'erreur spécifique doit être affiché.
        - Si l'utilisateur ne sélectionne pas exactement deux activités parmi celles proposées, un message d'erreur doit apparaître.
        - Si des erreurs sont détectées, le formulaire ne doit pas être envoyé.
        - Lorsque les erreurs sont corrigées, les messages d'erreur doivent disparaître lors de la prochaine soumission du formulaire.

Points à respecter :

- Le formulaire doit être envoyé par POST à l'adresse suivante :

  https://progweb1-validation-formulaire-e4hae7gedrdvbbaq.canadacentral-01.azurewebsites.net/

- Chaque balise `<label>` doit être correctement associée à son élément de formulaire correspondant, afin de garantir une bonne accessibilité et une expérience utilisateur optimale.
- Les champs du formulaire doivent avoir respectivement les noms suivants :
    - `nomEmploye`
    - `presenceEmploye` (valeurs : `"oui"` ou `"non"`);
    - `choixRepasEmploye` (valeurs : `"pouletAnanas"`, `"boeufBraise"` et `"tourtiere"`);
    - `seraAccompagne` (valeurs : `"oui"` ou `"non"`);
    - `nomInvite`;
    - `choixRepasInvite` (valeurs : `"pouletAnanas"`, `"boeufBraise"` et `"tourtiere"`);
    - `"traineau"`, `"noelDisco"`, `"karaoke"`, `"glissades"`, `"laserTag"` et `"casino"` (valeurs : utilisez les même chaînes).

Grille d'évaluation :

Critère	Points	Description
Fonctionnalités :		
• Interface utilisateur	/5	Interface claire, intuitive et fonctionnelle, comprenant les champs de saisie initiaux, des boutons fonctionnels, et une mise en page responsive (adaptée à différents écrans).
• Présence et choix de repas	/7	Gestion dynamique et correcte des choix de présence (oui/non) et du choix de repas. Le formulaire doit s'adapter aux réponses de l'utilisateur, en affichant ou masquant les sections appropriées.
• Accompagnement	/7	Affichage conditionnel et gestion correcte des champs d'accompagnement (nom de l'accompagnateur, choix de repas) lorsque l'utilisateur indique qu'il sera accompagné.
• Choix des activités	/8	Sélection correcte des activités : l'utilisateur doit choisir exactement deux activités parmi les options proposées. Une validation correcte doit être effectuée (message d'erreur si plus ou moins de deux activités sont sélectionnées).
• Validation des champs	/8	Validation correcte de tous les champs visibles avant soumission. Des messages d'erreur doivent apparaître si des champs obligatoires sont vides ou mal remplis. Les erreurs doivent disparaître lorsque les champs sont corrigés.
• Soumission du formulaire	/5	Soumission du formulaire via POST à l'URL fournie.
Qualité globale du code :	/10	Code bien structuré, lisible, avec des noms de variables explicites et une indentation correcte.
Total :	**/50**	

Informations générales

Qualité du français :

- Tous les textes affichés dans l'interface doivent être en français. Les pénalités seront appliquées conformément à la PDEA (0.5% par faute jusqu'à 10 points maximum);
- Les fautes d'orthographe et de syntaxe dans le code seront également pénalisées. Vous pouvez choisir entre le français ou l'anglais pour l'entièreté votre projet.

Contraintes techniques :

- Travail individuel. Trois périodes en classe y seront entièrement consacrée;
- Pas d'utilisation d'intelligence artificielle;
- Un fichier html et un fichier JavaScript par projet.
- Vous devez remettre votre avancement à la fin de chaque cours.

La proportion des points attribuée à la partie « Qualité globale du code » ne pourra pas dépasser la proportion des points atteinte pour les fonctionnalités. Par exemple, si une personne cumule 20 points pour les fonctionnalités (50% de 40 points possibles) elle ne pourra pas avoir plus de 5 points pour la qualité du code (50% de 10 points possibles).

Les propositions des étudiants peuvent être évaluées en entrevue après la remise du travail.

Le non respect des contraintes techniques vous donnera immédiatement la note 0.

Remise : 11 décembre 2024 à minuit

- Faites une archive de votre projet en format.zip;
- Faites votre remise sur Léa dans la section « Travaux » → « TP2 – énoncé et dépôt final ».