

Examen de rattrapage - Application Web de gestion de Notes

Le but est d'écrire en Node.js une application Web permettant à l'utilisateur de gérer ses notes personnelles.

Ce travail doit être effectué de manière individuelle.

L'étudiant devra rendre l'URL du dépôt git contenant le code source de cette application, hébergé sur le GitLab de l'école.

Critères d'évaluation

Le travail sera évalué selon les critères suivants:

- Fonctionnel: Les fonctionnalités ont été implémentées conformément au cahier des charges fourni ci-dessous.
- Lisibilité: Le dépôt git respecte la structure de fichiers de `express-generator`. Il contient un fichier `README.md` qui explique comment cloner et faire fonctionner l'application depuis une autre machine que la sienne. Les fichiers JavaScript doivent respecter les conventions de codage de Express: chaînes de caractères entre apostrophes, indentation à 2 espaces, usage de point-virgules pour ponctuer chaque instruction.

Cahier des charges

Dans cette section, nous décrivons les fonctionnalités attendues dans l'application Web à produire.

Spécifications fonctionnelles

L'application Web à fournir doit permettre à l'utilisateur de:

- visualiser toutes ses notes, dans l'ordre anti-chronologique, avec leur date de création;
- saisir une nouvelle note.

Les notes sont des chaînes de caractères à stocker et afficher en tant que texte brut (c.a.d. non HTML) mais doivent pouvoir contenir des sauts de ligne, ainsi que n'importe quel caractère Unicode. (accents, emoji...)

Chaque page Web rendue par l'application doit être facilement compréhensible et utilisable à la souris et au clavier par les utilisateurs, quel que soit leur navigateur. Les fonctionnalités doivent être assurées même si l'exécution de code JavaScript a été désactivée dans le navigateur de l'utilisateur.

Spécifications techniques

Environnement d'exécution

Le serveur Web doit pouvoir s'exécuter dans n'importe quel environnement de type Linux, doté de Node.js version 10 et de MongoDB version 4.

Les variables d'environnement suivantes seront fournies:

- `PORT`: numéro de port sur lequel le serveur Web devra être à l'écoute de requêtes HTTP entrantes;
- `MONGODB_URL`: l'URL permettant à l'application de se connecter au serveur MongoDB.

Interfaces

Le serveur Web de l'application implémentera du server-side rendering. C'est à dire que chaque route décrite ci-dessous devra rendre du contenu HTML valide (W3C) et prêt à être affiché dans le navigateur.

Pour des raisons de sécurité et d'esthétique, l'usage de paramètres `GET` est interdit. Les routes doivent être capables d'interpréter les paramètres au format `application/x-www-form-urlencoded`, conformément aux possibilités des formulaires HTML standards.

Les routes à implémenter sont décrites ci-dessous.

Route `GET /`

L'index de l'application Web est une page HTML qui contient:

- Un favicon;
- Le nom de l'application, affiché en haut à gauche;
- Un formulaire permettant de saisir une nouvelle note, via la route `PUT /note` ;
- Toutes les notes saisies précédemment par l'utilisateur, accompagnées de leur date de création.

Route `POST /note`

Cette route permet à l'utilisateur d'ajouter une note à sa liste de notes.

Une fois la note enregistrée en base de données, l'utilisateur est redirigé vers la page d'index, où la note doit être désormais visible en première position.

En cas d'erreur, une description de l'erreur est affichée, et un hyperlien permet à l'utilisateur de retourner sur la page d'index pour essayer à nouveau.

Paramètres attendus:

- `content` : contenu de la note saisie par l'utilisateur.