

---

## 1 INFORMATIONS GENERALES

Elève:	Nom: <b>LEVEQUE</b>	Prénom: <b>LORIS</b>
Lieu de travail:	<b>ETML</b>	
Période de réalisation :	03.06.2019 – 05.07.2019	
Horaire de travail :	8h-16h35	
Nombre d'heures :	60h	

---

## 2 PROCÉDURE

- L'élève réalise un travail personnel sur la base d'un cahier des charges selon le niveau des priorités.
- Le cahier des charges est approuvé par un enseignant. Il est en outre présenté, commenté et discuté avec l'élève.
- L'élève est entièrement responsable de la sécurité de ses données.
- A la fin du délai imparti pour la réalisation du projet, l'élève devra produire une présentation d'une dizaine de diapositives.

---

## 3 TITRE

Application web de mémos

---

## 4 SUJET

Créer une application multi-OS permettant d'afficher et d'ajouter des cartes mémos avec une description, des tags et des commentaires.

Evolution : Partager l'application web via un serveur (raspberry) sur tout un sous réseau.

---

## 5 MATÉRIEL ET LOGICIEL À DISPOSITION

1 PC ETML  
1 Environnement Windows 7  
Visual Studio Code  
Nodejs  
Git  
Excel  
Trello  
Connexion internet

---

## 6 PRÉREQUIS

Aucun

---

## 7 DESCRIPTIF DU PROJET

Créer une application web multi-OS responsive.  
Utilisant la technologie nodejs.  
Avec un serveur local (distant si le temps le permet)

La partie nodejs :

- Fichiers initial nodejs sera nommé app.js
- Utilisation de Framework tel que Express ou Mongo dB sont autorisées
- Les dépendances seront données dans deux fichiers générés : package.json et package-lock.json
- Coder de façon smart

Avec une database nommée « demomotbd ».

Une carte doit posséder :

- Un titre
- Un contenu
- Une description
- Un/Plusieurs tag(s)
- Un/Plusieurs commentaire(s)
- Un chiffre métrique (priorité)
  - Plus le métrique donné sera faible, plus la carte sera importante
  - Métrique maximum : 1023 ( $2^{10}$  possibilités) → zéro inclus

Des interfaces :

- Vues dans un dossier « views »
- Trois vues :
  - Affichage de toutes les cartes
  - Ajout d'une carte
  - Modification/Affichage d'une seule carte

Un rapport complet

- Descriptif du projet
- Description de la technologie nodejs
- Description de la planification (fait-pas fait)
- Database
- Planification
- Code
- Conclusion

*Evolution :*

Mettre en place un système d'utilisateur (hors-admin)

- Il faut être loggé pour ajouter une carte
- Pas besoin d'être loggé pour afficher toutes les cartes
- Les utilisateurs seront authentifiés par :
  - Un nom
  - Un prénom
  - Une adresse mail
  - Un pseudo
  - Un mot de passe (8 char min)
  - Une image de profil
  - Une description de profil/actualité de profil

Partager l'application web via un raspberry sur tout un sous-réseau

- Partager avec l'adresse IP du raspberry et un port choisi arbitrairement

- Le raspberry doit avoir une adresse et un masque fixe

---

## 8 POINTS TECHNIQUES ÉVALUÉS

- Application web multi-OS
- Technologie nodejs valide
- Affichage de toutes les cartes
- Ajout d'une carte
- Tri avec un tag
- Ajouter/Supprimer un tag sur une carte
- Commenter une carte
- Planification/Rapport/Présentation

*Respect des normes de codage ETML*

---

## 9 VALIDATION

	Lu et approuvé le :	Signature :
Elève :		
Enseignant :		