



RAPPORT PROJET PROGRAMMATION WEB METEOR

Sujet : Gestion des tâches et du temps pour la gestion des projets

UV 4.8 : Initiation au développement web

Supervisé par
Joël CHAMPEAU

Levan Fredy MBOVING TEGUETIO

Ali SAAD

Mahmoud SOUAYFAN

Henri STOVEN

RESUME

Dans le cadre du cours d'initiation au développement Web, il nous a été demandé de choisir un sujet pour en faire un projet de site/application web. Il sera question pour nous de nous familiariser avec les principes, outils et techniques de développement de sites et d'applications web. Notre objectif est de réaliser un outil web de gestion de projet. Nous utilisons HTML, CSS et bootstrap pour le frontend. Pour le backend, nous utilisons MeteorJS (basé sur NodeJS) et MongoDB pour la base de données. Nous utilisons aussi des packages meteor pour accélérer la mise en place de fonctionnalités. L'application que nous réalisons permettra l'inscription d'utilisateurs, la création & gestion des groupes de projets, l'attribution des tâches aux membres des groupes.

MOTS CLES

Meteor, gestion des projets, tâches, gestion des équipes, bootstrap.

Table des matières

ABSTRACT	Erreur ! Signet non défini.
RESUME	1
INTRODUCTION	3
PRESENTATION DU SUJET	4
ANALYSE	5
Présentation de la base de données	7
Présentation des outils et packages utilisés	9
DESCRIPTION ET FONCTIONNEMENT	13
Description d'une page HTML	14
Description du CSS	Erreur ! Signet non défini.
Fonctionnement	Erreur ! Signet non défini.
CONCLUSION.....	17

INTRODUCTION

La gestion des projets ou management des projets est l'ensemble des activités visant à organiser le bon déroulement d'un projet et à en atteindre les objectifs. Ce rapport présente l'ensemble de notre travail et ce qui a été réalisé. Nous allons donc présenter le sujet, l'analyse du sujet proposé. Nous allons ensuite présenter notre méthode de travail, les outils et les packages utilisés pour réaliser le projet. Par la suite, nous allons présenter le résultat du travail en insistant sur la structure de certaines pages et sur le fonctionnement de l'application. Pour conclure, nous allons émettre quelques pistes d'améliorations, en passant par les difficultés rencontrées.

PRESENTATION DU SUJET

Le thème de notre projet est « **Gestion des tâches et du temps pour la gestion des projets** ». Nous réaliserons une application web qui permettra à ses utilisateurs de faire de la gestion de projets. Plusieurs utilisateurs pourront s'inscrire dans l'application et pourront eux-mêmes créer des projets. Nous rendrons possible l'attribution des tâches d'un projet à des membres (préalablement invités) d'un groupe ou d'une équipe. Une tâche réalisée pourra être marqué comme tel, par le membre ayant réalisé ladite tâche.

Sur la base de ces spécifications, le site contiendra donc les pages suivantes :

- Inscription
- Connexion
- Accueil
- Tâches

ANALYSE

Pour tout progiciel, il est important de passer par une phase d'analyse pour s'assurer qu'à terme, la solution proposée réponde aux spécifications et aux exigences de départ. Cela nous permettra de définir les limites du projet et d'avoir une idée claire de ce qui est à réaliser.

Dans cette partie, nous présenterons donc une analyse technique du sujet et la méthode de travail (gestion d'équipe) que nous avons adopté pour travailler sur le projet. Nous présenterons aussi les outils et packages utilisés.

Acteurs :

- Administrateurs
- Membre
- Visiteur

Administrateurs :

- Créer, modifier (Groupe/Equipe)
- S'inscrire
- Accorder les privilèges (Modifier, créer, ...)
- Ajouter / Modifier les taches
- Assigner des taches a un membre précis ou à un groupe de membres

Membre :

- Créer et gérer ses propres tâches
- Modifier l'état d'une tache
- Commenter une tache
- S'inscrire

Visiteur :

- Voir les équipes publiques

Présentation de la base de données

La base de données est l'entité qui contiendra presque toutes les informations que stockera le site web. La structurer correctement est donc très important car cela facilitera son utilisation plus tard. Une base de données bien structurée rendra aussi l'application optimale.

Nous réaliserons une application de gestion des projets et de ses tâches dans une ou plusieurs équipes. Sur cette base, la base de données sera donc constituée des entités suivantes :

- Utilisateur
- Tâche
- Groupe

Utilisateur

L'entité « Utilisateur » est celle qui représente les futurs utilisateurs de l'application. Pour chaque utilisateur, nous aurons besoin de retenir **des informations personnelles** tels que les **noms et prénoms, âge**, etc. Nous avons aussi décidé que les utilisateurs n'auront pas tous les mêmes droits dans l'application. C'est-à-dire qu'ils n'auront pas tous accès aux mêmes fonctionnalités. Nous aurons donc besoin de retenir un rôle pour chaque utilisateur

Tâche

Les tâches des projets seront représentées dans la base de données par cette entité. Pour chaque tâche, nous retiendrons les **dates de début et de fin de la tâche, un état** (si elle est en cours, achevé ou non), **le/les utilisateurs** qui travaillent sur cette tâche, et **le groupe** dans lequel cette tâche a été créé.

Groupe

Le groupe représentera ici l'équipe d'un projet. Chaque groupe aura un **identifiant**, une **description**, une **date de création**, et **l'auteur** de sa création.

La figure ci-dessous représente la base de données de l'application. Elle contient toutes les entités mentionnées plus haut, ainsi que les relations qui les lient. Elle a été réalisée dans l'outil « MySQL workbench » et ne sert que d'illustration pour représenter la base de

données de l'application qui elle, a été faite directement via les méthodes fournies par Meteor.

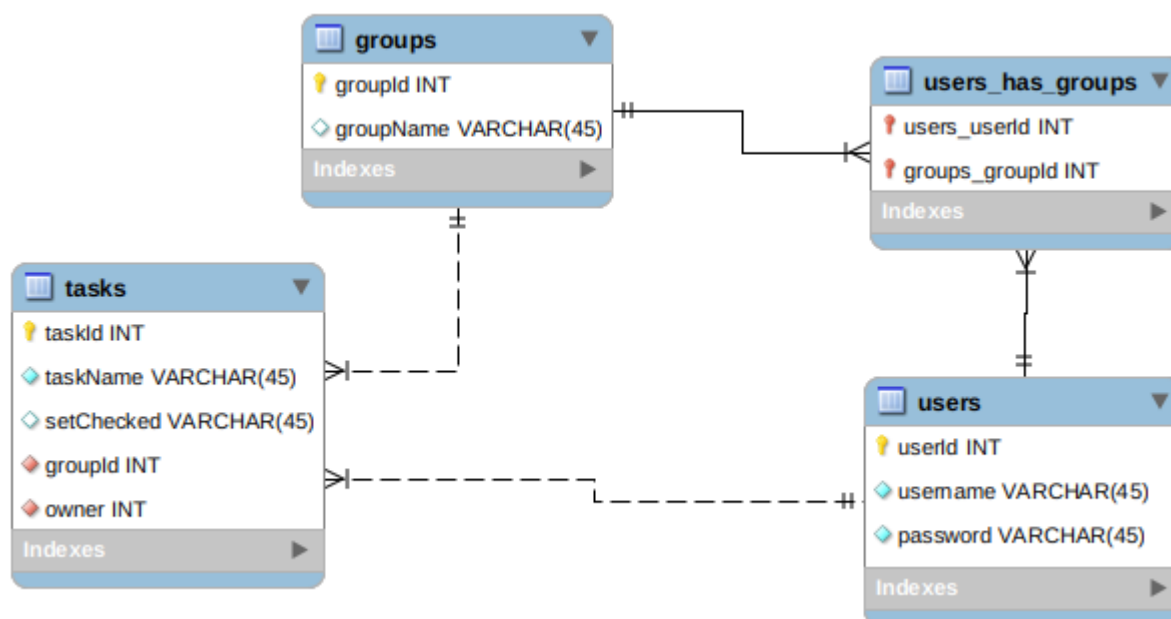


Figure 1: Diagramme d'entités - relations de la base de données

Présentation des outils et packages utilisés

1. iron:router

C'est un package de routage utilisé pour les applications Meteor.

Il est considéré comme aide pour le routage, d'une autre façon, pour mettre en place les chemins.

Tout d'abord, nous avons commencé à installer le package, en tapant cette commande dans le terminal :

Meteor add iron : router

L'objectif donc, c'est de relier des URLs à des templates.

Nous allons alors pointer le router vers un template spécifique contenant un « helper de template » qui est le `{{> yield}}`.

Dans 'layout.html', dans le « body » nous trouvons :

```
<div id= "content">  
  
    {{>yield}}  
  
</div>
```

Ce dernier va nous définir une zone dynamique qui va afficher le template correspondant, automatiquement, à la route courante.

Donc nous prenons par exemple une ligne de code de la page « layout1.html » qui illustre ce que nous avons dit :

```
<li><a href= « {{pathFor route='home'}} »>Home</a></li>
```

Nous allons ensuite, créer un nouveau fichier "routes.js" qui se trouve dans *"/imports/startup/client"* :

```
Router.configure({  
    layoutTemplate : 'layout'  
});  
  
Router.route('/', {name : 'home', });
```

Notant que 'home' le nom du template.

2. accounts-base

Il est considéré comme un système de compte utilisateur de Meteor.

Nous avons l'installé en tapant la commande suivante :

meteor add accounts-base

Ce package implémente les fonctions de base nécessaires aux comptes d'utilisateurs et permet à d'autres packages de se connecter à des services de connexion. Certains de ces services se trouvent dans le paquet suivant:

accounts-password

Nous avons utilisé ces deux packages pour assurer plusieurs fonctions sur le compte d'utilisateur, comme :

Accounts.createUser()

Meteor.loginWithPassword()

Meteor.logout()

Accounts.changePassword()

Accounts.removeEmail(): côté serveur

Accounts.addEmail(): côté serveur

3. simple-schema

Pour pouvoir utiliser ce package, il nécessite d'installer :

```
meteor npm install -save simpl-schema
```

```
meteor npm install -save bcrypt
```

4. aldeed

Nous avons l'installé en tapant la commande suivante :

```
meteor add aldeed:simple-schema
```

Nous avons utilisé ce package pour valider des objets. Puis nous avons ajouté les packages **aldeed:collection2-core** et **aldeed:autoform**. La première c'est pour effectuer une validation automatique pour les opérations : insert et update sur la collection. La deuxième, est reliée aux formes pour rendre flexible leur validation ainsi que leur génération depuis le schéma.

5. Session

“**session**” est utilisé pour sauvegardés les informations lors du transport d'une page à l'autre

meteor add session

Fournit un objet global sur le client que vous pouvez utiliser pour stocker un ensemble arbitraire de paires clé-valeur. Utilisez-le pour stocker des éléments tels que l'élément actuellement sélectionné dans une liste.

6. cfs

Pour donner l'utilisateur une permission de télécharger une image (pour changer l'image de profile)

7. Bootstrap

Comme les autres packages, il fallait installer le package « **bootstrap** » :

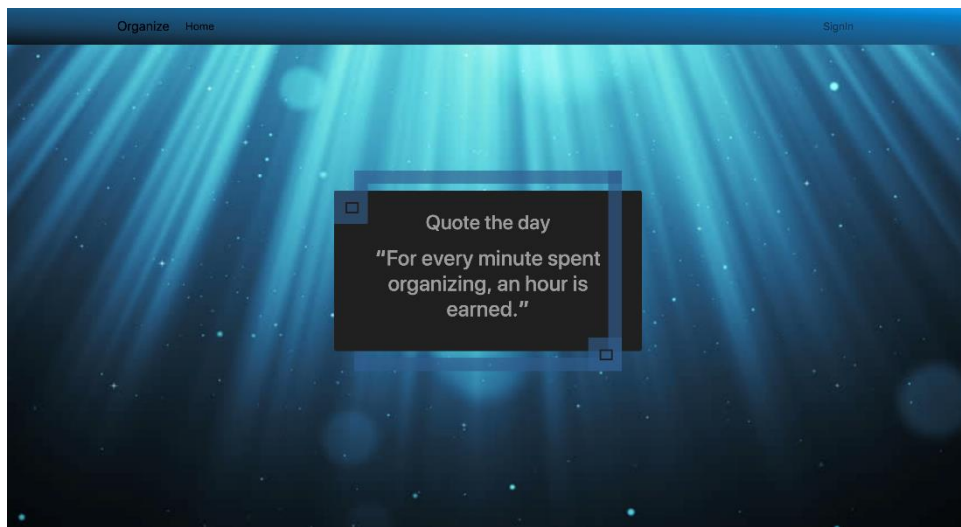
meteor add twbs:bootstrap

L'utilisation de ce package était nécessaire pour faire le contenu le plus important de la page, responsif même que toute la page n'est pas responsive. Donc le but c'est de donner à l'utilisateur l'accès à ce contenu même s'il change la taille de la fenêtre.

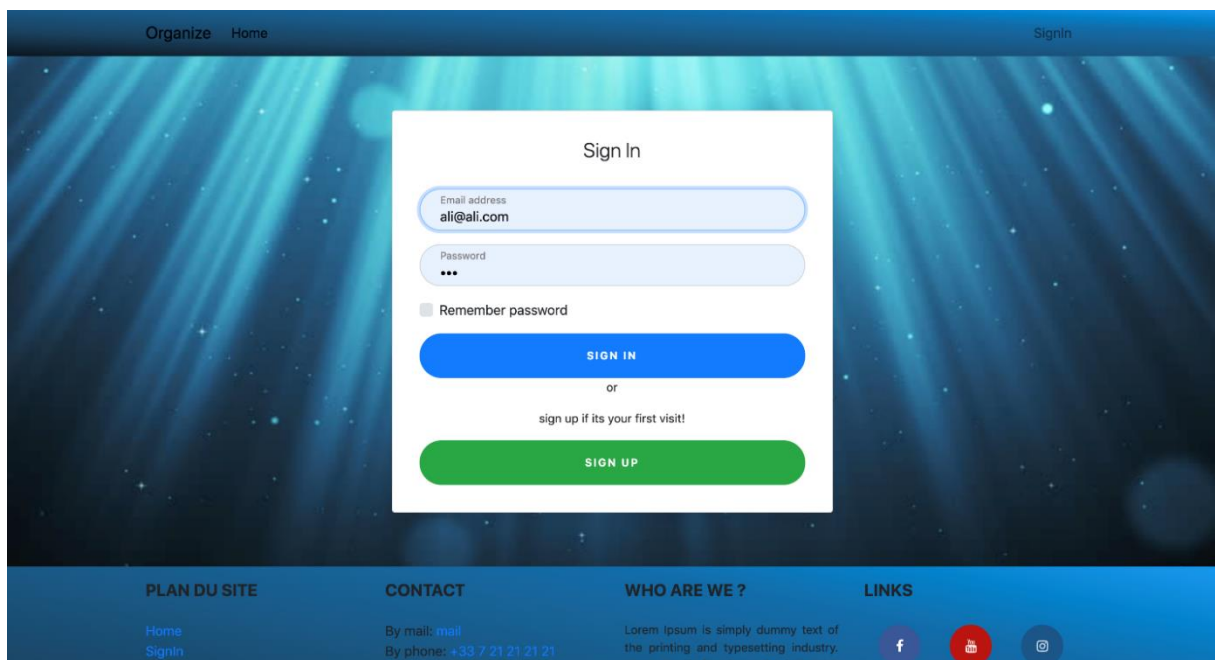
DESCRIPTION ET FONCTIONNEMENT

Cette partie du rapport présentera la structure de l'application. Au travers d'une page HTML, nous représenterons la composition de nos pages et leur agencement. Le rendu visuel sera aussi abordé par la présentation du code CSS utilisé dans les pages de l'application.

Description d'une page HTML, CSS et Fonctionnement



La première page, c'est la page d'ouverture, qui présente un proverbe pour l'utilisateur



L'utilisateur doit s'identifier pour avoir l'accès à son compte, en allant à la page signup pour créer un nouveau compte

Rejoignez notre communauté maintenant!

First Name

Last Name

Date de naissance :

Sexe: ☐ Man ☐ Women

user name

E-mail

password

Confirm your password

REGISTER

PLAN DU SITE: [Home](#), [Signin](#)

CONTACT: By mail: [mail](#), By phone: [+33 7 91 21 21 21](#)

WHO ARE WE ? Lorem Ipsum is simply dummy text of the printing and typesetting industry.

LINKS: [f](#), [y](#), [i](#)

L'utilisateur doit enregistrer les données nécessaires, si il y aura une erreur dans les données une erreur va être affichée, l'erreur est affichée en se servant de la fonction "toast"

Organize Home profile tasks SignOut

PRO User

ali saad

Pseudo: ali

[Mail](#)

subscribed in 1 groups

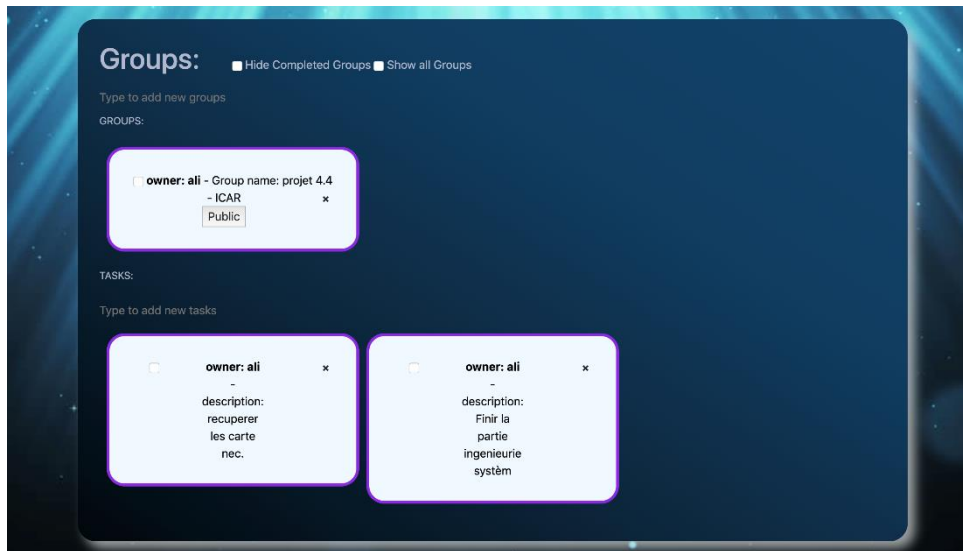
Groups: ☐ Hide Completed Groups ☒ Show all Groups

Type to add new groups

GROUPS:

- ☐ owner: ali - Group name: projet 4.4 - ICAR x

Une carte panel est affichée au début de la page profile, montrant la photo de profile de l'utilisateur, son nom, son pseudo et son mail et le nombre de group qu'il est inscrit



L'utilisateur est capable de créer un groupe, et puis créer des taches associées a cette group

Dans notre projet on a utilisé la librairy Bootstrap (version 4)

CONCLUSION

Ce projet a été pour nous une occasion de nous familiariser avec un domaine passionnant de l'informatique : le développement web. Les outils utilisés nous ont permis d'apprendre à réaliser des progiciels complets malgré le peu de temps que nous avons. MeteorJS est une technologie simple, complet et sécurisé pour développer des applications dynamiques et dignes des exigences d'aujourd'hui. Nous avons pu réaliser des fonctionnalités tels que présenté plus hauts, malgré les difficultés que nous avons rencontrées. Au travers de l'application que nous avons développés, il est désormais possible de

- S'inscrire, se connecter et se déconnecter
- Créer des groupes de projets
- Créer des tâches et les attribuer à des utilisateurs

Pour nous, ce site est l'aboutissement d'un travail intense et passionnant. Nous sommes ravis du résultat obtenu et continuerons à l'améliorer pour le rendre meilleur.

Difficultés rencontrées

Malgré la facilité de la prise en main de MeteorJS, nous avons rencontrés des difficultés durant nos travaux :

- Le temps imparti ne nous permettait pas d'aller jusqu'au bout de nos ambitions pour ce projet, étant donné notre emploi du temps académique chargé ;

Perspectives

Pour rendre l'application plus abouti, il faudra

- Parfaire l'interface pour améliorer l'expérience utilisateur obtenu.
- Ajouter une fonctionnalité de gestion graphique de type gantt chart des tâches.
- Inclure un calendrier des tâches dans l'espace personnel des utilisateurs