|  |  |  |
| --- | --- | --- |
|  | | Projet web 2019 |
|  | |  |
| 06/11/2019 | | Rapport de projet |
|  | Contexte :  Notre BDE souhaite un site internet à la hauteur d’une école d’ingénieur informatique pour gérer la promotion des évènements à venir ainsi qu’une boutique en ligne pour des goodies ou autres vente en ligne. Au sein de cette communauté chacun aura un rôle (employés, étudiants, membres du BDE) qui lui procurera certaines fonctionnalités.  L’objectif est de faciliter l’organisation et la communications au sein de l’école et de proposer des avantages et des goodies à l’effigie du BDE. | |

Projet web 2019

Rapport de projet

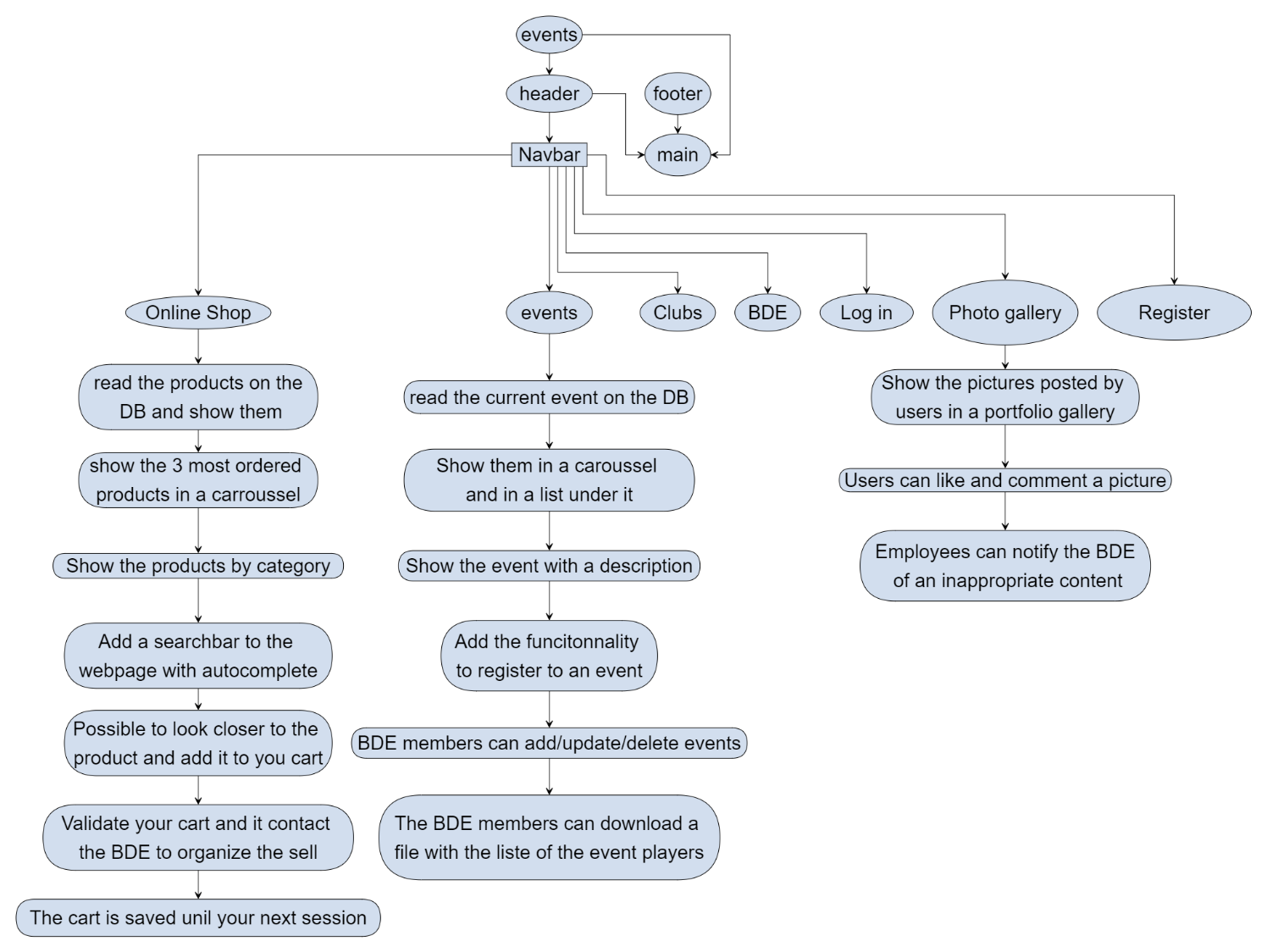
# Rappel des besoins

Suite à une étude générale faite auprès des demandeurs, des besoins ont été définis comme ce qui suit :

* Il y a 3 rôles définis en plus de celui de simple visiteur :
* Etudiant, employé et membre du BDE.
* Ce qui accorde l’accès à différentes fonctionnalisées définies.
* Enfin, le site sera connecté à une base de donnée MySQL.



# Description des différents fichiers composant le site Web



Nous avons ci-contre une description approximative des différentes fonctionnalités ainsi que les pages sur lesquelles elles apparaissent. Le site a été construit en séparant au maximum les fichiers et en se reposant sur des technologies modernes tels que le langage HTML, CSS, PHP ou encore JavaScript. De plus le site est lié avec une base de données MySQL étant dans notre cas locale et les base de données nationale utilisera des procédures pour communiquer avec les autres BDE afin d’organiser d’autres évènements. En effet il aurait été possible d’utiliser la base de données nationale mais cela aurai augmenter le trafic de données sur la même base et donc ainsi le même serveur. Pour dynamiser au maximum le site web nous avons donc opté pour une architecture plus sectaire. En plus de la façade du site en PHP nous avons une version d’API fonctionnant sous Node.js pour l’administration des données de la BDD.

# utilisation de Base de données

Dans ce projet nous avons donc connecté une base de données MySQL stockant les données relatives au site web. Nous avons donc établi un MCD (Modèle Conceptuel de Données) schématisant les relations entre les différentes tables de cette base.( [MCD : Annexe 1](#_User_stories_:) )

Les différentes tables présentes nous permettent donc d’organiser la gestion des informations, des évènements, des actions des utilisateurs (inscription, vérification de mot de passe …) ensuite nous avons dû programmer des différents services.

# Description technique du site

Nous avons donc pour ce site crée différents fichiers incorporant les différentes pages et/ou fonctionnalités. Premièrement nous avons une page d’accueil ou page principale qui va inclure un fichier header contenant certaines conditions lui permettant de d’adapter ses liens de redirections en fonction du statut de connexion de l’utilisateur, le body contenant simplement un carrousel des différents évènements à venir. Pour finir nous avons un footer nous fournissant accès aux mentions légales, aux pages de réseaux sociaux relatifs au BDE.

## Inscription et connexion

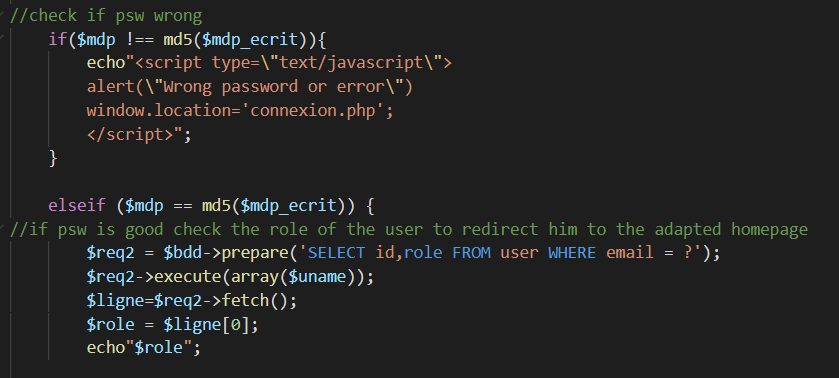
La partie intéressante étant le header qui nous sert donc de barre de navigation au sein du site web.

Au sein du header nous avons des liens de navigation tout simples pour ce qui est de la redirection vers la boutique, les évènements ou autre, mais la partie relative à la connexion ou l’inscription a été dynamisée avec un script PHP afin de donner certains droits aux différents types d’utilisateurs.

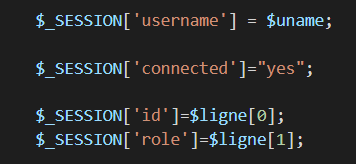


Ce script simple test si la variable de session ‘connected’ existe et a une valeur existante, si c’est le cas alors l’utilisateur est connecté et donc il peut donc accéder à son ancien panier s’il existe et peut donc naviguer sur les différentes pages du site. Sinon, il peut se connecter ou s’inscrire sur le site.

L’inscription insère simplement des informations dans notre base de données via des requêtes préparées de type PDOStatements. Pour la connexion nous avons utilisé la même procédure mais pour lire les données présentes cette fois et les comparer aux données rentrées par l’utilisateur lors de sa connexion. Le mot de passe étant crypté lors de l’enregistrement dans la base de données pour des raisons de sécurité, il faut donc crypté le mot de passe entré pour comparer les 2 chaines de caractères si elles sont identiques alors c’est le bon mot de passe et nous faisons le même test pour le nom d’utilisateur.



Si les deux tests réussissent nous définissons des variables de sessions qui seront actives et utilisées lors de la navigation sur le site pour accorder l’accès ou non à certaines fonctionnalités.



## Evènements :

Enfin nous avons la page des événements du BDE du CESI de Rouen dans cette partie nous avons mis en place un bouton pour ajouter un événement dans la base de données via des requêtes préparées par ailleurs toutes les requêtes que nous avons effectuées sont des requêtes préparées. Après avoir rempli la BDD des événements nous affichons la liste des événements sur une page web ‘events.php’ cette page affiche les événements dans avec un tri (du plus proche au plus lointain par rapport à aujourd’hui). Bien sûr il nous fallait afficher les détails de chaque événement alors nous avons créé une page ‘event.php’ à laquelle on accède via la page précédente. Par la suite les utilisateurs pouvaient s’inscrire aux événements et voir la liste des inscrits en pdf.

## Galerie photo :

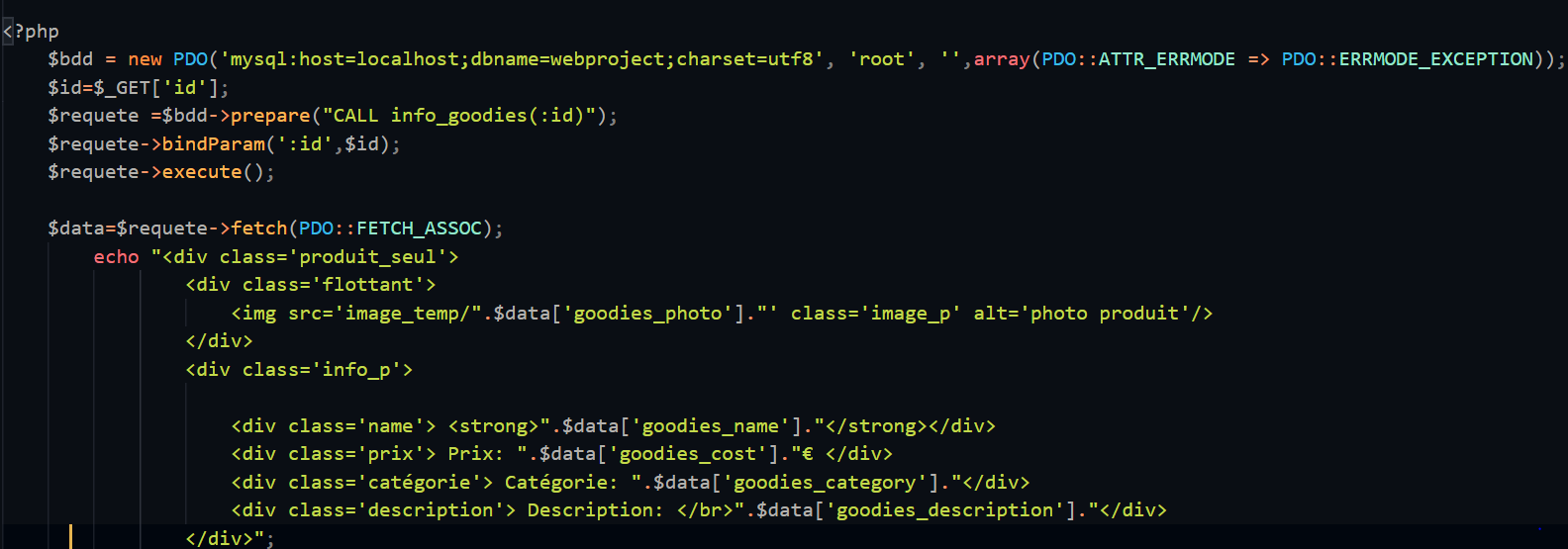
Pour la galerie photo il fallait récupérer les champs de la table ‘picture’ qui sont contenues dans la base de données une fois ces champs récupérés nous affichons les images avec leur nom et leur description.

Nous devions ajouter en dessous de cette description une section like qui comme sur le célèbre réseau social Facebook permettrait de ‘liker’ une photo. Pour cela nous avons dessiné un cœur en CSS puis utilisé une fonction JavaScript qui permet de changer la couleur du cœur dessiné. De plus cette action rajoutera alors un ‘like’ et l’enregistrera dans la base de données permettant ainsi d’afficher les ‘like’ postés par l’utilisateur connecté.

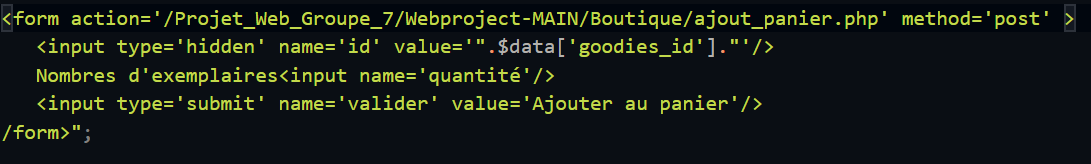
Pour finir la partie galerie nous devions créer un espace commentaire en dessous des photos pour cela nous utilisons une sous- boucle qui permet de récupérer chaque commentaire et l’utilisateur qui la posté.

## Boutique :

Il y a 3 pages pour la boutique qui s’affichent à l’utilisateur : la boutique, la page d’un produit et le panier. C’est 3 pages ne contiennent pas de codes par elles même , ce sont des templates qui permettent d’inclure les informations avec des include et qui chargent le css et les script.  
  
Une particularité pour la page d’un produit, on y accède en cliquant sur un article et ensuit l’identifiant de l’article est passé par l’url pour que le template charge la page, ainsi si on connais l’id d’un article on peut aussi naviguer en modifiant cet valeur manuellement.

Ensuite on a dans le dossier boutique les fichiers qui font le lien avec la bdd et fournissent les informations.  


On commence par la requête préparée avec PDO qui utilise un procédure stockée de la bdd pour aller chercher les infos puis on les récupère avec $data afin de les disposés dans la page.

On utilise aussi plusieurs balise form pour modifier la base de données par des actions qui redirigent ensuite sur la bonne page du site avec la fonction header.  
  
  


On utilise pour l’affichage de la page principale 2 framework , jquery pour la barre de recherche avec autocompletion et bootstrap pour le carrousel.  
  
On utilise aussi un script trouvé sur w3school adapté pour filtrer les catégories de goodies en mettant le nom de la catégorie dans la classe des articles pour ensuite sélectionner lesquelles s’affichent grâce à des boutons. Le tout évoluant dynamiquement en fonction des catégories présentes dans la bdd et des autres filtres.  
  
Le contrôle des prix des articles est géré comme pour la page d’un article en passant les valeurs extrémales dans l’url.

## Administration :

Pour la partie administration nous devions créer des tableaux contenant le contenu des tables de la base donnée. Depuis ces pages nous pouvons ajouter, éditer ou supprimer un champ dans une table de la base de donnée. Pour créer cette partie administration avec node.js nous utilisons l’infrastructure web express. Express nous a permis de créer une api REST.

## Retour d’expérience :

**Théophile Renouf :**

Ce projet a été un projet éprouvant d’un point de vue personnel du fait que je n’apprécie pas spécialement le développement web, j’ai alors dû fournir plus de travail que sur d’autre projet pour pouvoir finir mes tâches dans les échéances imparties. Toutefois se projet m’a permis de réaliser que même si une discipline n’est pas notre fort on peut toujours réussir à réaliser des taches plus ou moins complexes.

**Antoine Cahard :**

Un projet qui a permis la mise en pratique des connaissances apprises récemment dans plusieurs langages et leurs interactions, c’était bien. La mise en œuvre avec des élèves que je connaissais peu c’est bien passé et la mise en place conjointe de nos efforts nous a fait progresser régulièrement.

**Cédric Letalleur :**

**Achille Brossier :**

Un projet un peu effrayant au regard de la montagne de travail au premier abord, au second regard on se rend compte que le charge de travail est grande et parfois très complexe. Il a alors fallu diviser les tâches entres les différentes parties et s’organiser pour travailler, certes dans la semaine durant la journée il était facile de se retrouver toutefois durant le week-end ou même le soir après les cours, difficile de se retrouver les membres du groupe n’étant pas tous motorisés et certains travaillant, il a fallu donc fournir de gros efforts en passant par le sacrifice de nombreuses heures de sommeil et de litres de café. Cependant après un combat acharné nous avons je pense réussi.

# Annexes :

## MCD :

