

RAPPORT DE PROJET WEB :

Table des matières

Rappel du besoin et bilan de validation de ceux-ci :	1
Contexte :	1
Besoin :	2
Permissions et fonctionnalités disponibles pour un étudiant (visiteur) :	2
Permissions et fonctionnalités disponibles pour un étudiant (connecté) :	2
Permissions et fonctionnalités disponibles pour un étudiant (membre du BDE) :	2
Permissions et fonctionnalités disponibles pour un employé du CESI :	3
Description technique du projet :	3
Choix technique :	3
Justification des choix techniques :	3
Node.js & Express :	3
Laravel :	3
Bootstrap :	3
Description des différents fichiers du projet :	4
Node.js et Express : L'API :	4
Laravel :	4
Dossier « public » :	4
Dossier « views » :	4
Dossier « route » :	5
Dossier « Controller » :	5
Bilan de groupe :	5
Le groupe :	5
Individuellement :	5
Nathan Poret (chef de projet) :	5
Valentin Pain :	6
Théo Langlois :	6
Ophélien Fouques :	6

Rappel du besoin et bilan de validation de ceux-ci :

Contexte :

Notre BDE souhaite un site internet qui permettrait de gérer la promotion des manifestations et une boutique en ligne. Les objectifs sont de faciliter l'organisation et la communication de manifestations au sein de l'école, et de proposer aux membres des

goodies à l'effigie du BDE. Chaque BDE possède son propre site Internet. Cependant les données de celui-ci seront stockées dans une base de données nationale.

Besoin :

Légende : Fait / Incomplet / Non réussi

Différent rôle en fonction des personnes pourront être proposé, le rôle d'étudiant (dit aussi visiteur), le rôle d'étudiant connecté, le rôle d'étudiant membre du BDE et enfin le rôle personnel du CESI.

Permissions et fonctionnalités disponibles pour un étudiant (visiteur) :

- M'inscrire sur le site en fournissant mon nom, prénom, localisation (centre CESI actuel) et adresse mail ainsi qu'un mot de passe contenant au moins une majuscule et un chiffre.
- Après avoir réalisé l'inscription, me connecter.
- Consulter la totalité du site (liste des activités, commentaires, photos...) et accéder à la boutique (sans pour autant pouvoir procéder à l'achat).
- Peuvent rechercher un produit en particulier dans une barre de recherche.
- Voir les 3 articles les plus commandés de la boutique (une fois dans la boutique).

Permissions et fonctionnalités disponibles pour un étudiant (connecté) :

//\ Ajouter toutes les permissions d'un étudiant visiteurs en plus.

- M'inscrire à l'une des activités proposées par les membres du BDE.
- Accéder à la liste des activités proposées par les membres du BDE et m'inscrire à l'une ou l'autre d'entre elles
- Ajouter des photos sur les événements passés pour lesquels j'étais inscrit.
- Commenter et liker les photos d'un événement passé.
- Passer commande par l'intermédiaire d'un panier. Si la commande n'est pas finalisée par l'étudiant alors le panier est sauvegardé et l'étudiant retrouve sa liste de produits à sa prochaine connexion.

Permissions et fonctionnalités disponibles pour un étudiant (membre du BDE) :

//\ Ajouter toutes les permissions d'un étudiant visiteurs et connecté en plus.

- Poster une manifestation avec une description, une image et une date, dans une partie « événement du mois ». Cette manifestation peut être ponctuelle ou récurrente, payante ou gratuite.
- Accéder, pour chaque manifestation proposée, à la liste des inscrits et la télécharger au format PDF ou CSV.
- Administrer les photos et commentaires de la partie « événements passés ».
- Peuvent ajouter, supprimer des produits avec nom, description et prix, en les classant par catégories.
- Reçoivent un mail à chaque achat dans la boutique (mail envoyé automatiquement)

lors d'un achat d'un produit dans la boutique par un étudiant connecté).

Permissions et fonctionnalités disponibles pour un employé du CESI :

/!\ Ajouter toutes les permissions d'un étudiant visiteurs en plus.

- Notifier l'ensemble des membres du BDE que certaines photos, commentaires, manifestations peuvent nuire à l'image de l'école. Dans ce cas les éléments ne peuvent être rendus publics.
- Naviguer sur l'ensemble du site avec les mêmes droits que les étudiants.
- Télécharger via un bouton l'ensemble des photos postées par les étudiants et les membres du BDE.

Description technique du projet :

Choix technique :

Utilisation de **Node.js** avec **Express** pour une **API REST** qui permet la liaison entre la base de données du site en **MySQL** et le site. Utilisation de **Laravel** pour la structure du site. Utilisation de **bootstrap** et de simple **CSS** pour la forme. Utilisation de **JavaScript** pur et **JQuery** ainsi d'**AJAX** pour plus de fonctionnalité (affichage et administration).

Les événements seront affichés en fonction du campus de l'utilisateur s'il est connecté sinon c'est tous les événements qui seront affichés. De plus, chaque utilisateur pourra chercher à l'aide de la barre de recherche : un événement, peu importe si celle-ci se trouve sur le campus de l'utilisateur. Cela permet de mieux lier et souder les étudiants CESI au national les poussant à aller voir ce qu'il se passe dans les autres campus. Donc les événements seront stockés dans la BDD nationale.

Justification des choix techniques :

Node.js & Express :

L'utilisation de Node.js pour une API REST était déjà imposée dans le cahier des charges du projet, l'utilisation d'Express permettait de simplifier l'utilisation et l'échange entre le site et la base de données.

Laravel :

L'utilisation de Laravel était un challenge pour le groupe. Bien que non imposé nous voulions aller plus loin avec Laravel et ainsi l'intégrer à notre projet. Laravel permet aussi de gérer la sécurité du site presque automatiquement avec des tokens et des routes. La structure de Laravel permet une gestion complète du site.

Bootstrap :

L'utilisation de Bootstrap a été surtout utilisée pour la responsive du site. Cependant du CSS pur a aussi été utilisé pour personnaliser le site.

JavaScript, JQuery & AJAX :

L'utilisation de JQuery et de JavaScript purs dépendait surtout de comment les membres du groupe jugeaient leur compétence avec ou sans la librairie JQuery. Quant à AJAX son utilisation a été imposée dans le cahier des charges.

Description des différents fichiers du projet :

Node.js et Express : L'API

L'API situé dans un fichier à part du site, nommé dans notre groupe « *api_bdecesi* », celui-ci contient tous les modules Node ainsi de deux fichiers de configuration en JSON, « *package.json* » qui contient le nom de projet, sa version, la fichier choisit pour le lancement de l'api, l'auteur et les dépendances. Le fichier « *package-lock.json* » contient toutes les descriptions (version, lien, intégrité) des différentes dépendances de chaque module Node. Le fichier qui va être le plus modifié par chacun des membres du groupe sera « *server.js* » qui contiendra la connexion à la base de données ainsi que toutes les requêtes faites depuis les fichiers « *.blade.php* » du site relié grâce à une adresse commune : <http://localhost:3000/api/>... Et ceux-ci sont reçus à l'aide de méthode **.get**, **.post** ou même **.delete** dans celui-ci.

Laravel :

Celui-ci gère le site, en effet le dossier principal « *bdecesi.fr/site* », cependant nous allons surtout utiliser les dossiers « *public* », « *views* » (qui se trouve dans le dossier « *ressources* »), « *routes* » et « *controller* » (situé dans « *app/http* »).

Dossier « public » :

Le dossier « *public* » contient toutes les feuilles de styles en fonction des pages, le JavaScript et toutes les images du site.

Dossier « views » :

Celui-ci contient trois dossiers important : « *emails* », « *content* » et « *layouts* »

Dossier « emails »

Contiens uniquement le mail à envoyer.

Dossier « content »

Contient toutes les pages du site et celles-ci sont triés en fonction des rôles, un dossier « *admin* » qui contient toutes les pages dédiées aux membres du BDE et l'administration de la boutique. Le dossier « *cesi-staff* » qui contient les pages fournissant les fonctionnalités demandées telles que notifier les membres du BDE et télécharger toutes les images du site. Enfin le dossier « *classic_user* » contient tout le reste, c'est-à-dire la majorité des pages comme la connexion, l'inscription, la boutique, les activités, la page d'accueil, le profil, les mentions légales...

Contiens toutes les pages dites « **récurives** » c'est-à-dire que nous allons utiliser sur toutes les pages. Il s'agit bien évidemment du « *header* » et du « *footer* », bien que géré un peu différemment. En effet, ceux-ci sont contenu dans le fichier « *master* » qui sera, lui, affiché dans toutes les pages. La technique utilisée pour remplir ce fichier « *master* » avec le contenu de toutes les autres pages fonctionne de la manière suivante : on appelle le fichier « *master* » avec `@extends(' ../layouts.master')` ensuite on définit le début du contenu avec : `@section('content')` après celui-ci on aura tout le contenu de la page (par exemple : « *login* ») et enfin on ferme la section avec `@endsection('content')`. Enfin le fichier « *callApi* » contient une fonction qui permet d'appeler l'API, cela permet de ne pas réécrire à chaque fois la fonction.

Dossier « route » :

Dans le dossier route il faut surtout s'intéresser au fichier « *web* » celui-ci contient toutes les « *routes* » que doivent prendre les données avant d'être affiché, cela passe notamment par des contrôleurs qui gèreront aussi les données.

Dossier « Controller » :

Dans notre cas nous avons utilisé uniquement les fichiers « *AdminController* », « *StaffCesi* » et « *UsersControllers* », ceux-ci gèrent respectivement les différentes vues dans les dossiers « *admin* », « *cesi-staff* » et « *classic_users* ». Les contrôleurs gères les sessions et les cookies. Et envoie dynamiquement au « *master* » quelle feuille de style et quel fichier JavaScript à utiliser en fonction de la page.

Bilan de groupe :

Le groupe :

Une très bonne ambiance de groupe, bien que parfois désorganisé (surtout au début) le résultat est là avec l'accomplissement des besoins avec succès.

Individuellement :

Chaque fait sera listé. Le **Front-End** regroupe toute la structure en HTML, le CSS et le JavaScript. Le **Back-End** regroupe la connexion à l'API, le PHP, les requêtes SQL, dans l'API, les contrôleurs et les routes.

Nathan Poret (chef de projet)

- Contribution à la réalisation du MCD
- Réalisation du **Front-End** de la page d'inscription, de connexion, des résultats de recherches, du footer.
- Contribution au **Front-End** de la page Boutique et Panier.
- Réalisation des cookies & sessions
- Réalisation du **Back-end** du Panier, des résultats de recherche, de la déconnexion

Valentin Pain

- Contribution à la réalisation du MCD
- Création de l'API Node.js + Express
- Création des « layouts » Laravel
- Réalisation du **Front-End** du header
- Contribution à la réalisation du **Front-End** de la page d'Accueil
- Réalisation du **Back-End** de l'inscription (comprends la vérification des champs), de la connexion, de l'envoi de mail et de l'inscription aux activités, du profil.

Théo Langlois

- Contribution à la réalisation de la maquette graphique
- Contribution à la réalisation du **Front-End** de la page d'Accueil
- Réalisation du **Front-End & Back-End** de la boutique, des activités (compris dans les activités : les likes et les commentaires)
- Ecriture des mentions légales, conditions de ventes et politique de confidentialité

Ophélien Fouques

- Contribution à la réalisation de la maquette graphique
- Contribution au **Front-End** du panier
- Réalisation du **Front-End** du profil
- Réalisation de tout le **Front-End & Back-End** de l'administration