

كلية العلوم
ⵜⴰⵎⴻⵔⴰⵏⵜ ⴰⵎⴰⵏⴰⵢⵜ
FACULTÉ DES SCIENCES



UNIVERSITE IBN ZOHR

FACULTE DES SCIENCES

Département Informatique

Filière Sciences Mathématiques et Informatique

PFE

Présenté par : Aissam EN-NAHEL

Pour l'obtention de la

Licence en Sciences Mathématiques et Informatique

**Développement d'interface graphique UX/UI
pour Dashboard des résultats et la qualité des
QCM**

Soutenu le 16/06/2022

Encadré par : M. Karim AFDEL

Année universitaire 2021-2022

Page laissée intentionnellement vide

Dédicaces

Dédicace à mon encadrant et à mes amis, à qui j'avais raison de faire confiance, et dédicace spéciale à mes parents et à ma famille qui m'ont encouragé et soutenu durant toutes mes études.

Merci de votre soutien.

Remerciements

C'est un plaisir de remercier ceux qui m'ont aidé à accomplir ce travail.

Je tiens à remercier toutes les personnes qui m'ont appuyé dans le développement de ce travail de PFE.

J'exprime ma profonde gratitude à mon encadrant, M. AFDEL, pour son encadrement précieux, ses orientations et ses conseils pour l'accomplissement de mon projet PFE.

J'espère que les membres du jury trouveront mes profonds remerciements pour avoir accepté de juger ce modeste travail.

Résumé

Dans ce rapport, je présente les étapes de la création d'un site Web dynamique portant sur une interface graphique utilisant le framework Angular pour les résultats et la qualité des examens QCM.

Dans le traitement de ce travail, je suis basé sur l'intégration d'un template open source et gratuit.

Je vais créer ce site Web pour afficher les résultats et la qualité des examens QCM en utilisant chart.js, une bibliothèque JavaScript open source, et pour le style, je vais utiliser Semantic UI, un framework CSS.

Les données des examens sont enregistrées sous forme de JSON.

Mots clés : Angular, Chartjs, Dashboard, Interface graphique, JSON, Semantic UI, Site Web.

Abstract

In this report, I present the steps of creating a dynamic website for a graphical interface using the Angular framework for MCQ exam results and quality.

In the course of this work, I rely on the integration of an open source and free template.

I will create this website to display the results and quality of MCQ exams using chart.js, an open source JavaScript library, and for the styling, I will use Semantic UI, a CSS framework.

The exam data is saved in JSON format.

Key words : Angular, Chartjs, Dashboard, Graphic interface, JSON, SemanticUI, Website.

Table des matières

Dédicaces.....	ii
Remerciements.....	iii
Résumé.....	iv
Abstract.....	v
Table des matières.....	vi
Liste des figures.....	vii
Liste des tableaux.....	viii
Liste des abréviations.....	ix
Introduction générale.....	1
Chapitre 1 : Contexte général du projet.....	2
Contexte et objectifs du projet.....	2
Présentation et description du projet.....	2
Architecture de navigation du site Web.....	3
Planification et Conduite du projet.....	4
Chapitre 2 : Analyse et conception.....	6
Spécification des besoins.....	6
Prototype de site web.....	6
Cas d'utilisation.....	8
Diagrammes de séquences.....	9
Modèles de données sous forme de JSON.....	9
Format des information sur les examens :.....	9
Chapitre 3 : Etude technique et environnement.....	11
Architecture adoptée : MVC.....	11
Choix des langages.....	11
Bibliothèque.....	12
Format de données.....	12
Framework utilisé.....	12
Chapitre 4 : Réalisation, Interfaces, Tests.....	13
Environnements matériels.....	13
Logiciels.....	13
Outils.....	13
Interfaces et tests.....	14
Conclusion Générale.....	18
Bibliographie.....	19
Annexes.....	21

Liste des figures

Figure 1: Description de l'algorithme du site Web.....	2
Figure 2: Architecture de navigation du site Web.....	3
Figure 3: Planification et Conduite du projet.....	4
Figure 4: Prototype de la page de connexion.....	6
Figure 5: Prototype de la page de Dashboard.....	6
Figure 6: les éléments de Sidebar.....	7
Figure 7: Les Vues constitutifs du site web.....	7
Figure 8: Diagramme de cas d'utilisation.....	8
Figure 9: Diagramme de séquence.....	9
Figure 10: Interfaces de la page de connexion.....	14
Figure 11: Interfaces de la page Home.....	15
Figure 12: Interfaces de la page d'Examen 2022.....	16
Figure 13: La fonctionnalité d'imprimer les corrections.....	17

Liste des tableaux

Tableau 1: Le contenus de modules de l'architecture MVC.....	11
Tableau 2: Les informations d'environnements matériels.....	13

Liste des abréviations

CSS	Cascading Style Sheets
JS	JavaScript
JSON	JavaScript Object Notation
HTML	HyperText Markup Language
LAMP	Linux Apache MySQL/MariaDB PHP
MCQ	Multiple Choice Question
MVC	Modèle Vue Contrôleur
PFE	Projet de Fin d'Etudes
QCM	Questionnaire à Choix Multiples
UI	User Interface
UX	User eXperience
VSCode	Visual Studio Code
WEB	World Wide Web

Introduction générale

Ce travail a été réalisé dans le cadre de mon projet de fin d'études. L'objectif de ce travail est de développer une interface graphique pour le tableau de bord des résultats et de la qualité des examens QCM, de module de l'architecture d'ordinateur, comme exemple.

Lorsque l'utilisateur se connecte à son compte, il est redirigé vers le tableau de bord. Ce tableau de bord lui permet de visualiser des graphiques représentant l'analyse des résultats relatifs à l'examen de l'année en cours, ainsi que d'autres graphiques représentant des informations sur les étudiants valides et non valides, ainsi que les étudiants absents.

L'utilisateur peut également consulter et imprimer les corrections des examens QCM, en suivant la même procédure pour les examens des années précédentes.

Chapitre 1 : Contexte général du projet

Contexte et objectifs du projet

Ce PFE est un site web dynamique réalisant une interface graphique pour le suivi des résultats et de la qualité des examens QCM. L'objectif du PFE est de concevoir une interface graphique permettant de visualiser les résultats et la qualité des examens QCM.

Je me suis concentré sur les examens des trois dernières années, y compris l'année actuelle, avec bien sûr la possibilité d'ajouter d'autres examens provenant d'autres années.

Présentation et description du projet

Mon projet commence par une page de connexion, où l'utilisateur peut se connecter à son compte avec son adresse e-mail et son mot de passe.

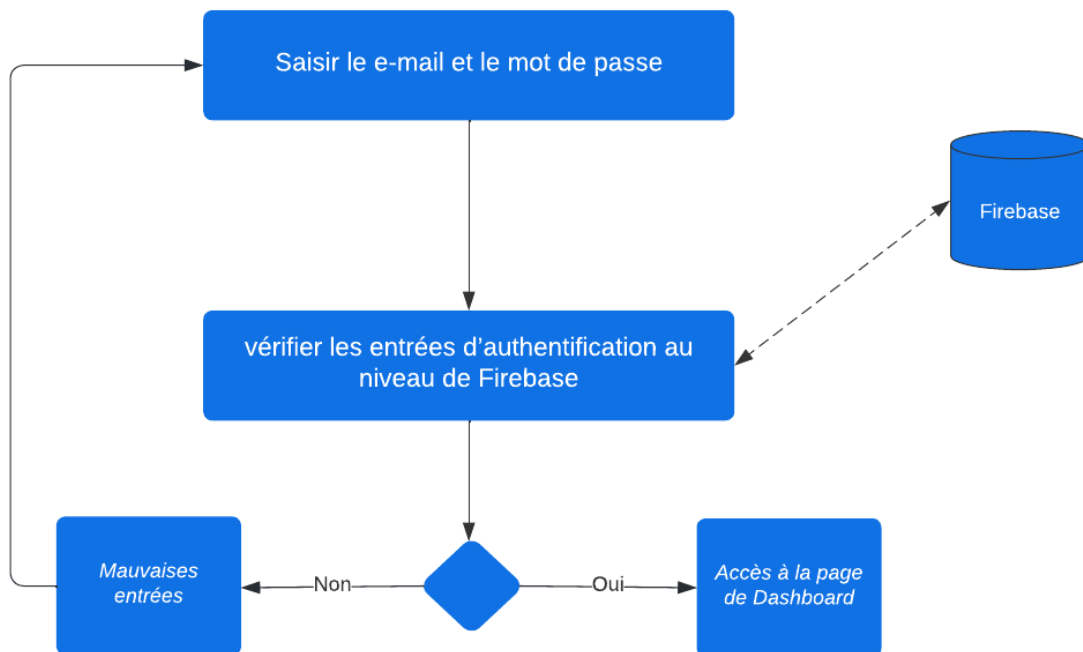


Figure 1: Description de l'algorithme du site Web

Après validation des données entrées, l'utilisateur est redirigé vers la page de tableau de bord. Sur cette page, l'utilisateur peut visualiser des graphiques relatifs aux statistiques des :

- étudiants ayant passé l'examen.
- étudiants n'ayant pas passé l'examen.
- étudiants ayant réussi l'examen.
- étudiants n'ayant pas réussi l'examen.
- pourcentages de questions faciles, moyennes et difficiles.

Et comparer cette statistiques entre la session normal et session de rattrapage.

Et aussi L'utilisateur peut :

- voir les corrections des examens.
- imprimer les corrections des examens.

Architecture de navigation du site Web

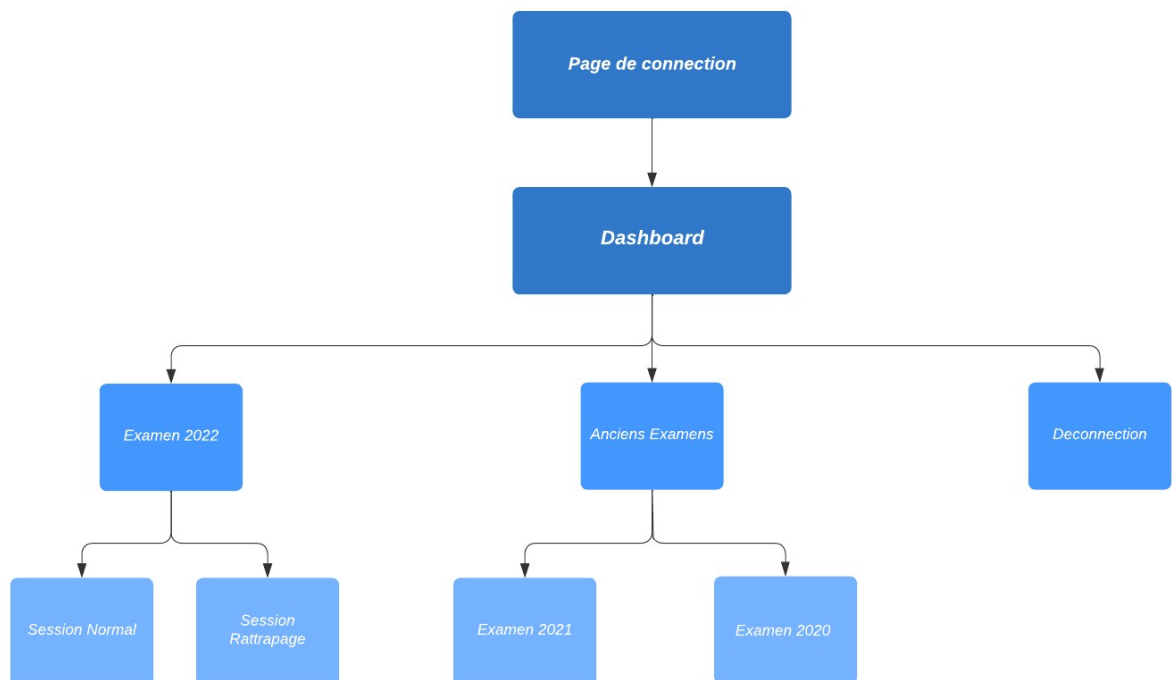


Figure 2: Architecture de navigation du site Web

Planification et Conduite du projet

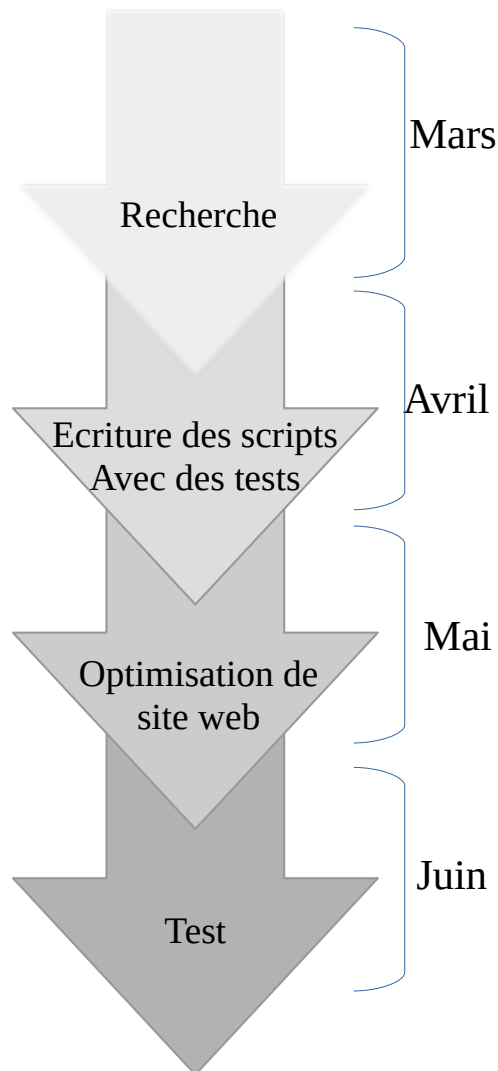


Figure 3: Planification et Conduite du projet

➤ Mars 2022 – Avril 2022

Recherches :

Pour obtenir les informations nécessaires dont j'ai besoin, j'ai recherché le framework AngularJS, la bibliothèque Chart.js, ainsi que le framework CSS Semantic UI.

➤ Avril 2022 – Mai 2022

Ecriture des scripts pour :

- la page de connexion et l'authentification de Firebase.
- les graphiques :
 - Graphique en barres (Bar chart)

- Diagramme circulaire (Pie chart)
- Graphique linéaire (Line chart)
- Graphique en anneau (Doughnut en anneaux)

➤ **Mai 2022 – Juin 2022**

Optimisation de site web : Améliorer l'esthétique du site web.

➤ **Juin 2022 – 16 Juin 2022**

Tests :

Effectuer les vérifications finales et les modifications des défauts.

Chapitre 2 : Analyse et conception

Spécification des besoins

Prototype de site web

1. Page de connexion :

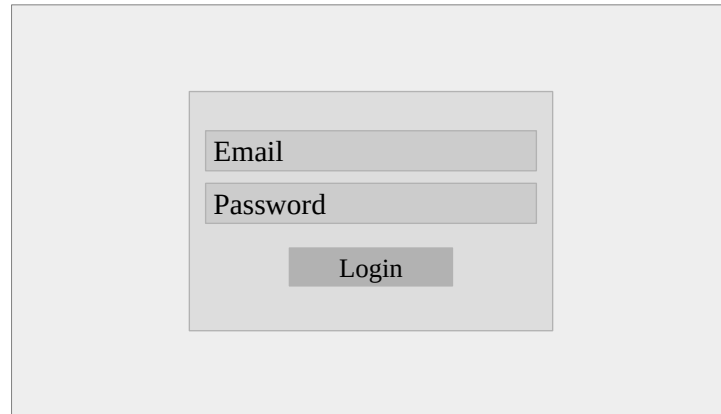


Figure 4: Prototype de la page de connexion

2. Page de Dashboard :

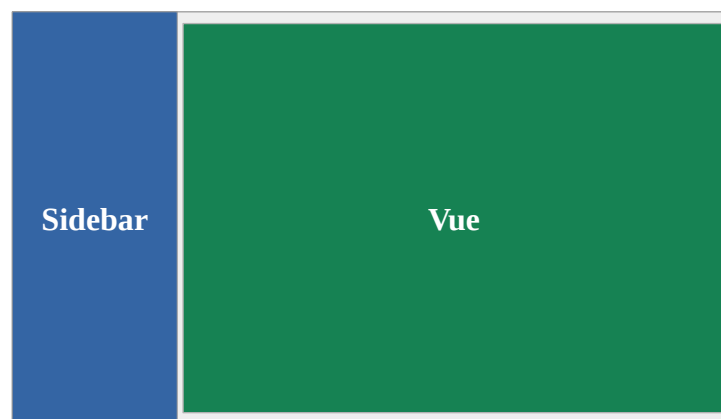


Figure 5: Prototype de la page de Dashboard

La partie Sidebar est composé de blocs menant aux vues spécifiques, comme l'indique le figure ci-après :

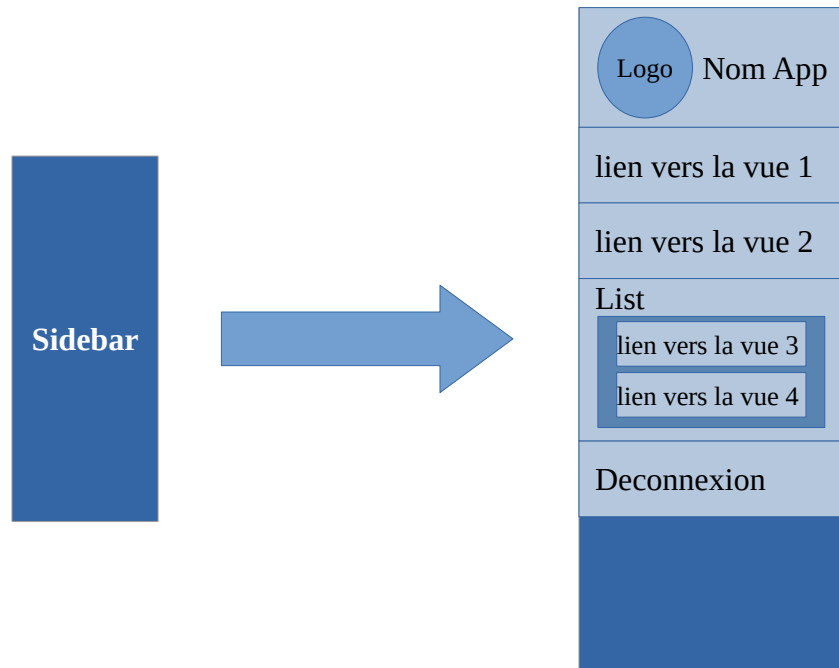


Figure 6: les éléments de Sidebar

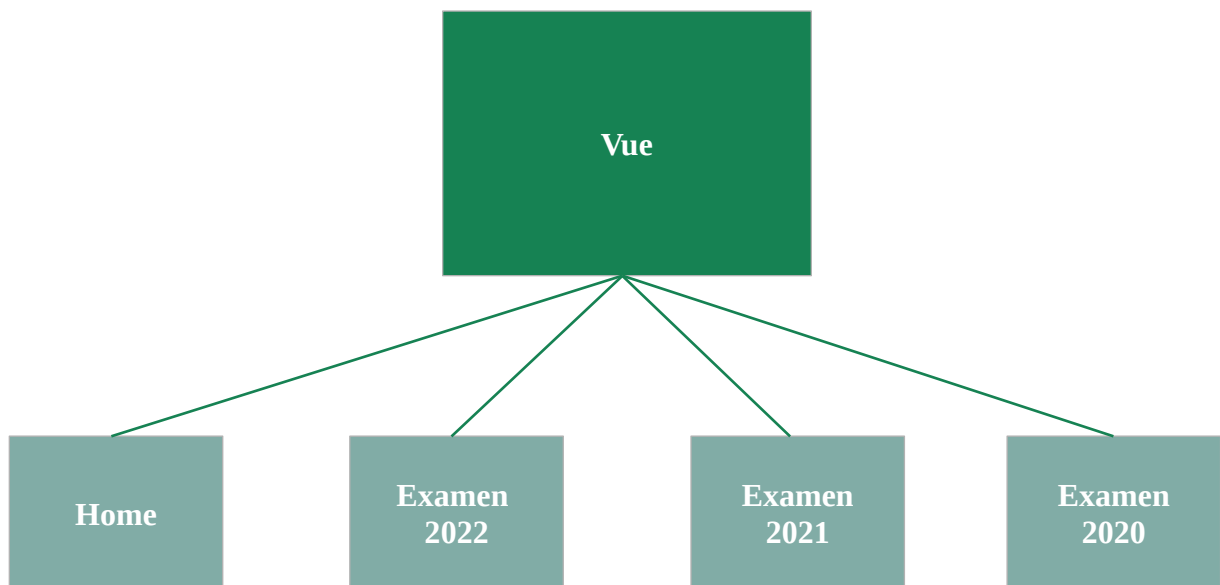


Figure 7: Les Vues constitutifs du site web

La partie Vue varie selon la vue choisi, Je règle cette partie par le routage avec Angular-route.js en utilisant le module ngRoute.

Cas d'utilisation

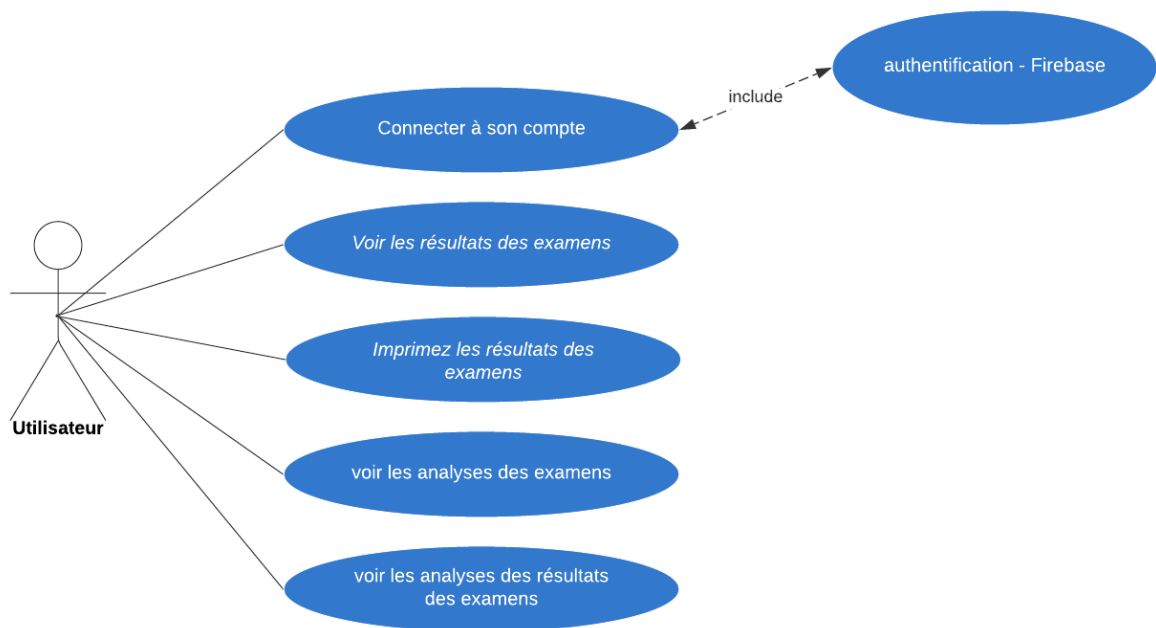


Figure 8: Diagramme de cas d'utilisation

L'utilisateur du site Web peut parcourir les diagrammes des résultats des examens, et des statistiques sur les élèves passent l'examen.

Diagrammes de séquences

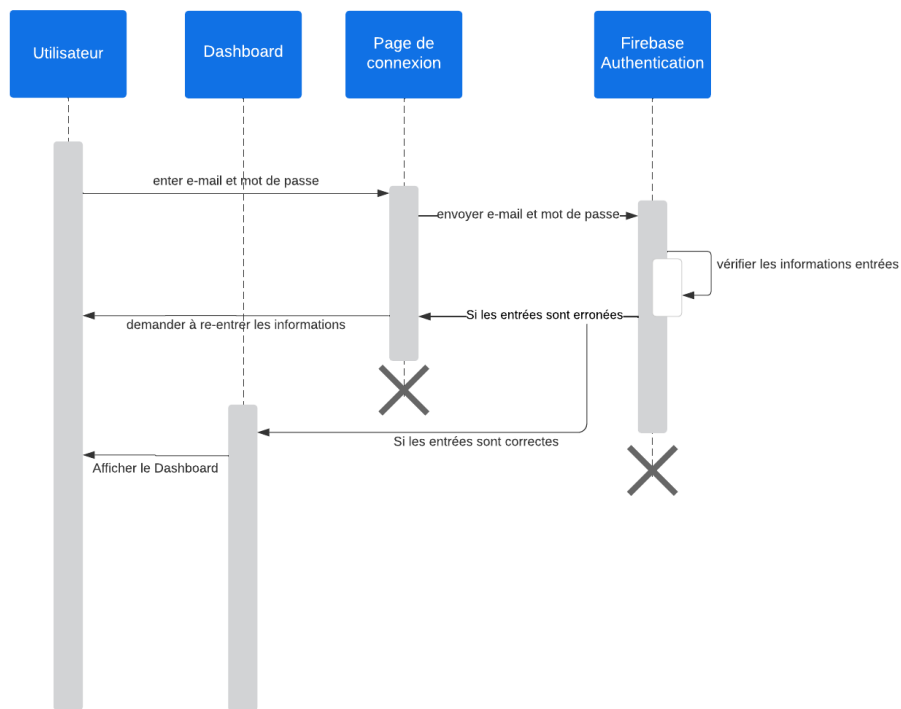


Figure 9: Diagramme de séquence

Modèles de données sous forme de JSON

➤ Format des information sur les examens :

```

"EtudExamen##":{
  "labels": ["Valide","Non Valid","Abs"],
  "data":[50,60,70]
}

"nbr_total_etud_insc" : 342,
"nbr_etud_val"        : 202,
"nbr_etud_noval"      : 130,
"nbr_etud_pass_exam" : 332,
"nbr_etud_abs"        : 10,

"Examen##Data":{
  "labels": ["Session Normal","Session Rattrapage"],
  "difficiledata" : [ 6, 9],
  "moyendata"    : [ 10, 8],
  "faciledata"   : [ 5, 3]
}

```

}

- Structure du format des questions et des réponses, pour que chaque examen contienne 20 questions :

```
"qcmList2019nor":[
  {
    "n":1,
    "question": " L'ordinateur ne peut pas démarrer s'il n'a pas _____?",
    "reponse": "D- Le système d'exploitation",
  },{
    "n":2,
    "question": " Exemple d'opération logique :",
    "reponse": "D- Tout les réponses sont vrais",
  },
  {...}
]
```

Chapitre 3 : Etude technique et environnement

Architecture adoptée : MVC

Modèle-vue-contrôleur est un motif d'architecture logicielle destiné aux interfaces graphiques lancé pour les applications web.

Le motif est composé de trois types de modules ayant trois responsabilités différentes : les modèles, les vues et les contrôleurs.

Tableau 1: Le contenus de modules de l'architecture MVC

MVC	
Modèle	les données à afficher
Vue	la présentation de l'interface graphique
Contrôleur	la logique concernant les actions effectuées par l'utilisateur

Le Modèle représente la structure des données. Leur définition ainsi que les fonctions qui leur sont propres et qu'elles peuvent avoir. Ce module est complètement décollé du code métier ou de l'affichage de telle sorte à ce que la modification de la logique ou de l'interface n'affecte pas la structure des données.

La Vue représente l'interface graphique à livrer au client qui en fait la requête. Avoir le code lié à l'interface isolé de la logique métier ou des données permet de faire des modifications à l'interface graphique sans avoir à se soucier de casser du code métier ou la structure des données.

Les Contrôleurs sont au cœur de la logique métier de votre application. Ils se situent entre les vues et le model. Les requêtes qu'un client va faire depuis l'interface graphique, la Vue, vont être dirigées vers un controller qui sera en charge de manipuler les données dont il a besoin avec la brique Modèle, la traiter suivant le besoin métier, puis ordonner à la Vue de répondre au client avec les bons éléments.

Choix des langages

- HTML

HTML est un type de langage informatique de balisage descriptif, il s'agit notamment d'un format de données utilisé pour le formatage et la représentation des pages web, d'inclure des ressources multimédias dont des formulaires et des images.



- CSS

CSS constitue un langage informatique permettant d'ajouter des contraintes de mise en forme graphique dans des pages web.



- JAVASCRIPT

JavaScript est un langage de programmation qui permet d'implémenter des mécanismes sur les pages web.



Bibliothèque

- Chart.js

Chart.js est une bibliothèque de graphiques JavaScript ,open source gratuite pour la visualisation de données.



Chart.js

- jQuery

La librairie jQuery consiste en un ensemble de blocs de codes JavaScripts préconçus et qui vont être généralement enfermés dans des méthodes. Il suffira donc d'appeler ces méthodes pour exécuter le code qu'elles contiennent.



Format de données

- JSON

JavaScript Object Notation, abrégé JSON, est un format de données textuelles et génériques permettant de représenter de l'information et le donnée.



Framework utilisé

- AngularJS :

AngularJS est un framework Javascript open source et gratuit permet de développer des applications web dynamiques, et placé du côté. Il est créé par Google en 2009.



- Semantic UI :

Semantic UI est un framework CSS offrant un large éventail de styles préconçus afin de créer une interface simple pour démarrer [un nouveau projet] des projets.



Chapitre 4 : Réalisation, Interfaces, Tests

Environnements matériels

Linux parrot 5.10.0-6parrot1-amd64 #1 SMP Debian 5.10.28-6parrot1 (2021-04-12) x86_64
GNU/Linux

Tableau 2: Les informations d'environnements matériels

Processor	Intel(R) Core(TM) i5 CPU M 560 @ 2.67GHz
Architecture	x86_64
RAM	8GB
CPU	4
Graphiques	Mesa DRI Intel® HD Graphics (ILK)

Logiciels

- LAMP :

LAMP est une pile logicielle comprenant le système d'exploitation, un serveur HTTP, un système de gestion de bases de données et un langage de programmation interprété, et qui permet de mettre en place un serveur web.



- VSCode :

Visual Studio Code est un éditeur de code multi-plateforme, open source, gratuit et extensible développé par Microsoft pour Windows, Linux et macOS, supportant plusieurs langages de programmation.



- Firefox :

Firefox est indépendant, c'est un logiciel de Mozilla, une organisation sans but lucratif qui lutte pour vos droits numériques, contrecarre les plans des grandes entreprises et rend Internet accessible à tous et partout.



Outils

- Lucidchart:

Pour créer les diagrammes, j'ai utilisé le site web Lucidchart. Lucidchart est une plateforme de collaboration en ligne, basée sur le cloud, permettant la création de diagrammes et la visualisation de données, et autres schémas conceptuels.



Interfaces et tests

- Page de connexion :

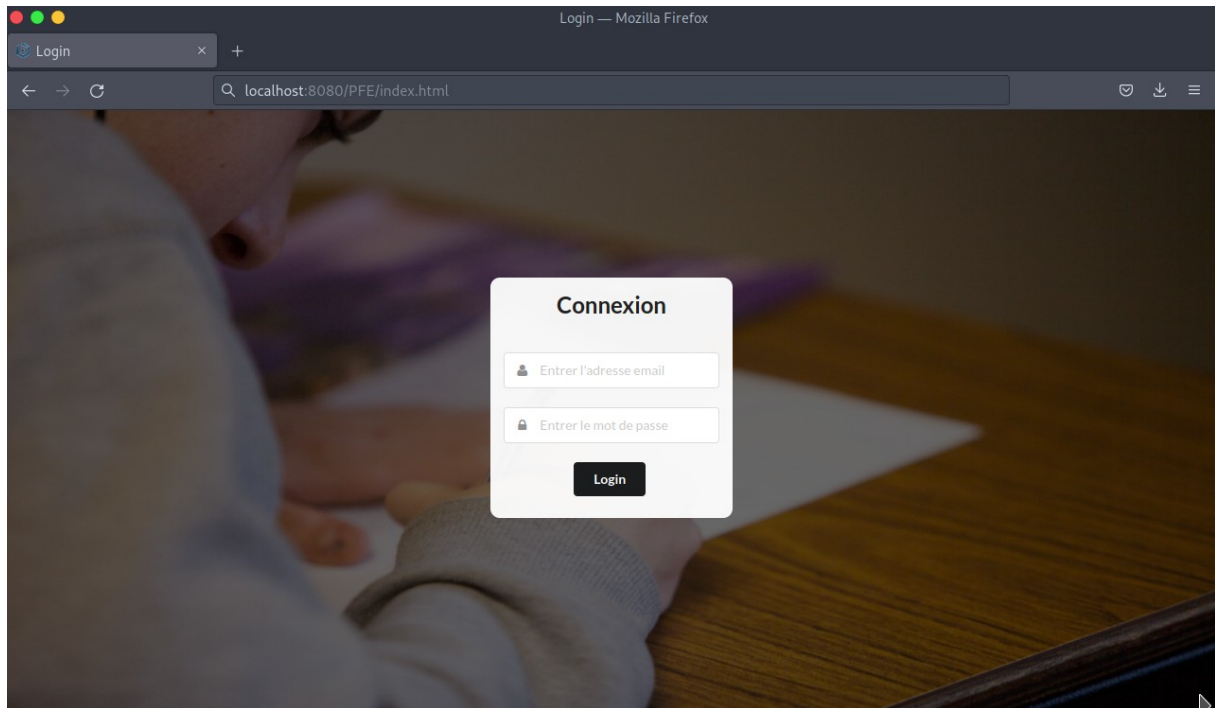


Figure 10: Interfaces de la page de connexion

- Page de dashboard :

➤ Home :

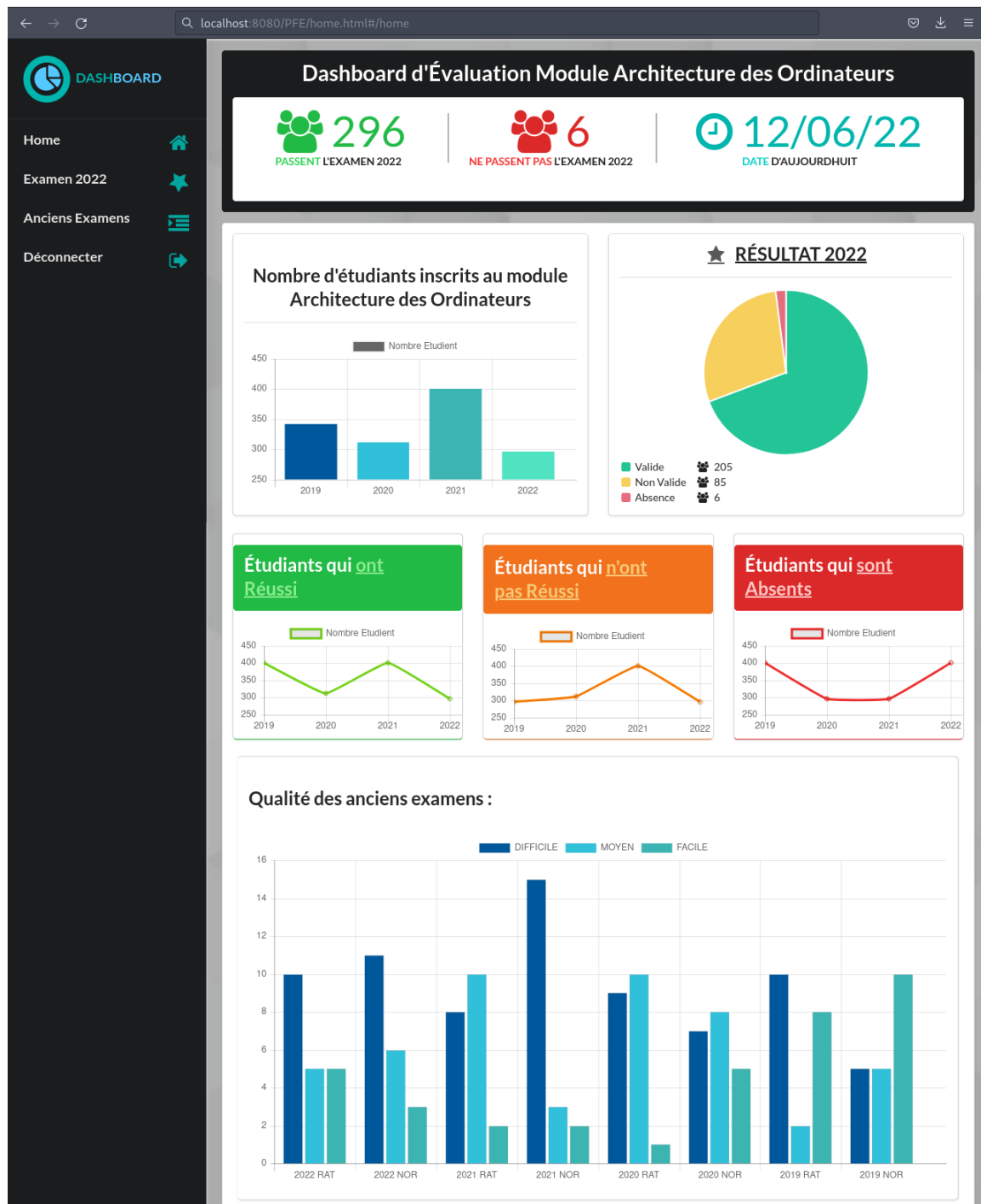


Figure 11: Interfaces de la page Home

➤ Page de l'Examen 2022 :

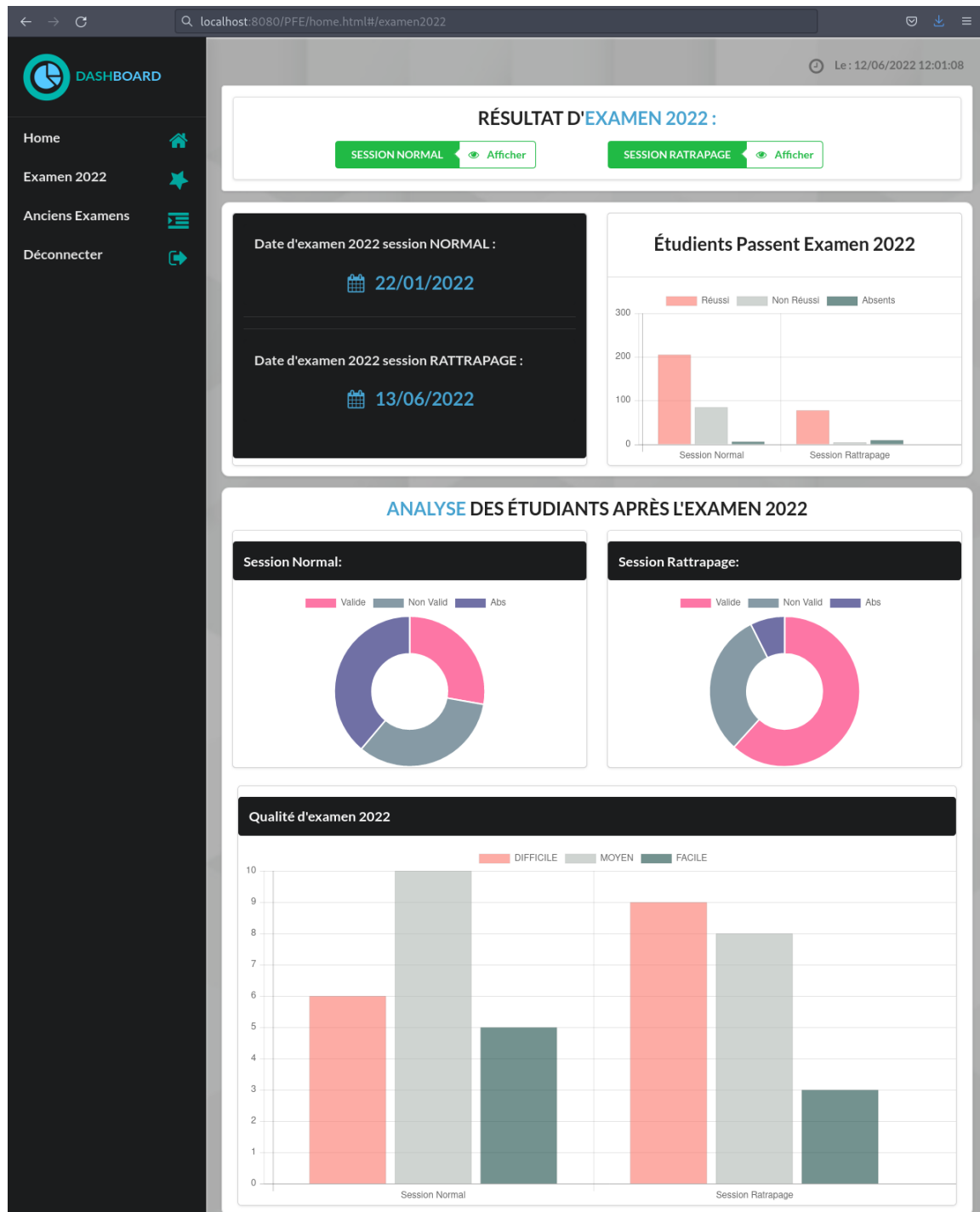


Figure 12: Interfaces de la page d'Examen 2022

La fonctionnalité d'imprimer les corrections :

RÉSULTAT D'EXAMEN 2022 :

SESSION NORMAL 👁 Afficher

SESSION RATRAPAGE

👁 Afficher

Resultat d'examen 2022 | NORMAL:

<p>Question 1: La mémoire qui permet l'opération de lecture et d'écriture simultanées est _____?</p> <p>Result: B- RAM</p>	<p>Question 2: Laquelle des expressions suivantes NE représente PAS le NOR exclusif de x et y?</p> <p>Result: D- $x' \wedge y'$ où \wedge est XOR</p>
<p>Question 3: Qui n'est pas considéré comme un périphérique de l'ordinateur?</p> <p>Result: D- CPU</p>	<p>Question 4: La taille du registre accumulateur peut être de _____?</p> <p>Result: C- 4 octets</p>
<p>Question 5: Le composant informatique le plus rapide est _____?</p> <p>Result: C- registre</p>	<p>Question 6: Le format _____ est généralement utilisé pour stocker des données.</p> <p>Result: A- BCD</p>
<p>Question 7: Un programme source est généralement écrit en _____.</p> <p>Result: C- Langage de haut niveau</p>	<p>Question 8: L'ALU utilise _____ pour stocker les résultats intermédiaires.</p> <p>Result: A- L'accumulateurs</p>
<p>Question 9: Un « BUS » informatique est composée de _____?</p> <p>Result: C- ensemble de lignes parallèles</p>	<p>Question 10: La somme des produits simplifiée de l'expression booléenne $(P + Q' + R')$, $(P + Q' + R)$, $(P + Q + R')$ est _____?</p> <p>Result: B- $(P + Q' . R')$</p>
<p>Question 11: L'ordinateur tire sa force de base de _____?</p> <p>Result: D- Tout les réponses sont vrais</p>	<p>Question 12: Un ensemble de lignes qui connecte plusieurs appareils s'appelle _____?</p> <p>Result: A- Un bus</p>
<p>Question 13: Un système complet de micro-ordinateurs se compose de _____?</p> <p>Result: D- Tout les réponses sont vrais</p>	<p>Question 14: Le Compteur de programme(PC) est également appelé _____?</p> <p>Result: A- pointeur d'instruction</p>
<p>Question 15: Dans un seul octet, combien de bits y aura-t-il?</p> <p>Result: A- 8</p>	<p>Question 16: Le processeur n'effectue pas l'opération _____</p> <p>Result: A- De transfert de données</p>
<p>Question 17: Le temps d'accès à la mémoire est _____ le temps nécessaire pour effectuer une seule opération du CPU.</p> <p>Result: B- plus long que</p>	<p>Question 18: Une condition d'exception dans un système informatique causée par un événement externe du CPU est appelée _____?</p> <p>Result: A- interruption</p>
<p>Question 19: Lorsque le CPU détecte une interruption, il enregistre alors son _____?</p> <p>Result: C- Etat actuel</p>	<p>Question 20: Considérons la fonction booléenne suivante de quatre variables: $f(w, x, y, z) = \sum (1,3,4,6,9,11,12,14)$. La fonction est _____</p> <p>Result: B- indépendant de deux variables</p>

🖨 Imprimer les corrections

Figure 13: La fonctionnalité d'imprimer les corrections

Conclusion Générale

La programmation d'applications web est une tendance adoptée de plus en plus par les entreprises et les sociétés en raison de ses nombreux avantages.

Ce projet a représenté une excellente expérience pour moi, et évidemment, il va me apporter énormément de connaissances techniques. De plus, il s'agit d'une application pratique de tous les aspects de la programmation que j'ai pu rencontrer durant mes études.

Bibliographie

smartrack / SemanticUI-Admin-Template / Public
<https://github.com/smartrack/SemanticUI-Admin-Template>
(Vérifié le 10/06/2022)

La page de documentation d'Angularjs
<https://docs.angularjs.org/guide/introduction>
(Vérifié le 10/06/2022)

La page de documentation de Semantic-UI
<https://semantic-ui.com/introduction/getting-started.html>
(Vérifié le 10/06/2022)

La page de documentation de Chart.js
<https://www.chartjs.org/docs/latest/>
(Vérifié le 10/06/2022)

Authentifiez-vous auprès de Firebase à l'aide de comptes basés sur un mot de passe à l'aide de Javascript
<https://firebase.google.com/docs/auth/web/password-auth>
(Vérifié le 10/06/2022)

Introduction au cours jQuery
<https://www.pierre-giraud.com/jquery-apprendre-cours/introduction/>
(Vérifié le 10/06/2022)

Modèle-vue-contrôleur
<https://fr.wikipedia.org/wiki/Modèle-vue-contrôleur>
(Vérifié le 10/06/2022)

Chart.js
<https://fr.wikipedia.org/wiki/Chart.js>
(Vérifié le 10/06/2022)

Quelques Sites Web :

➤ **Les bibliothèque :**

jQuery
<https://jquery.com/>
Chart.js
<https://www.chartjs.org/>

➤ **Les frameworks :**

Angularjs

<https://angularjs.org/>

Semantic-UI

<https://semantic-ui.com/>

➤ **Les logiciels :**

LAMP

<https://bitnami.com/stack/lamp>

VSCode

<https://code.visualstudio.com/>

Firefox

<https://www.mozilla.org/fr/firefox/browsers/>

➤ **Outils :**

Lucidchart

<https://www.lucidchart.com>

➤ **Format de données :**

Site web de la bibliothèque JSON

<https://www.json.org/json-en.html>

Annexes

Un exemple de code AngularJS :

```
<!doctype html>
<html ng-app="helloApp">
  <head>
  </head>
  <body>
    <h1>Exemple d'AngularJS</h1>
    <div ng-controller="helloCtrl">
      <h3>Hello {{helloCtrl.mot}} !</h3>
    </div>

    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.2.16/angular.min.js">
    </script>
    <script>
      var helloApp = angular.module('helloApp', []);
      helloApp.controller('helloCtrl', ['$scope', function ($scope) {
        $scope.mot = 'world';
      }]);
    </script>
  </body>
</html>
```

ce exemple va affiche : Hello world !

Un exemple de configurer le routeur avec Angular-route:

```
helloApp.config(['$routeProvider',
  function($routeProvider) {
    $routeProvider

    .when('/home', {
      templateUrl: 'vue/home.html',
      controller: 'homeCtrl'
    })

    .when('/examen22t', {
      templateUrl: 'vue/examen-2022.html',
      controller: 'exam22Ctrl'
    });
  }
]);
```