

# BigBen Project

# Outlines

I.	<b>Introduction .....</b>	<b>2</b>
II.	<b>User documentation .....</b>	<b>3</b>
	Logging system	
	Main page	
	The game	
	Point system	
III.	<b>Documentation technique .....</b>	<b>4</b>
	Technologies utilisées	
	La partie serveur/client	
	La base de données	
IV.	<b>Conclusion .....</b>	<b>5</b>

# Introduction

---

The English project is a perfect fit between programming and learning some things about the English culture. It's proved that in order to be the more attractive possible, an application which wants to teach something to a user should be made like a game with challenges, otherwise people are easily bored. With this conclusion, we tried to make an app: "Learning without knowing that you learn". It was the first step to build the BigBen Project.

The original idea was to make a kind of "Who Wants to Be a Millionaire?" game: the players have to answer questions and climb a point's scale. This system is mainly adapted in this project but the questions belong to a specific topic in order to learn on the British or American culture. We also wanted to add multiple categories. Someone playing the game can say that the game is a kind of Trivial Pursuit for who wants to be a Millionaire!

Within the following parts, we are going to introduce in details the concept of our game and how to play. Then we will explain how the game is working: the technologies behind the interface.

# User documentation

---

In this section, we will explain to anyone who wants to try BigBen, how to subscribe and log in our web application, and after this formality: how to enjoy the game.

## Logging system

We decided that a player need to have an account for playing. The account helps us to save scores and allow each user to play from any device at any time. So when we connect to the Big Ben website, we have to log into our account or create one in order to play. This implementation asked us an extra time of development but was really important to us.

## Main page

Once connected, we have access to the 4 categories of game: Cinema, London, The British music and Royal family. On the top of the screen, we have a tab to access to the home page, another to view the rank page and a button to see our account details.

## The game

When we start a game, like in “Who wants to be a millionaire”, a question is asked and 4 possible answer are proposed. We have to choose the only good answer to access to the next question and climb the levels. The game end up when the first mistake is made. In case of doubt, we included a bonus: the 50/50 that erase 2 wrong answer around the 4. This bonus can be played only one time per game.

## Point system

The points system is quite simple: for each category, a user can get a number of points depending on the level of questions he succeeded to access. For example, if the player failed to answer of the question 4, he will get an amount of points corresponding to this level. In addition, and in order to increase the challenge between players, there is some random points which are added to the score.

If a player wants more points on a category, he can play again and hope to win more random points but he can also decrease his score! In fact, if he succeeded the 10 levels and try again but this time can't reach all the questions, his score will not be the highest but the last one, in this example he will get less points than before the second game. That's the trick!

# Documentation technique

---

Ce projet n'est pas le premier que nous réalisons ensemble, nous avons donc décidé d'utiliser des technologies que nous connaissions déjà tous les deux afin d'accélérer le temps de développement et d'apporter un rendu appréciable.

## Technologies utilisées

Notre projet étant une application web, son développement nécessite une partie serveur et une partie client. La partie serveur est entièrement écrite en Node.JS en utilisant le Framework Express, la partie cliente est gérée en HTML, CSS (utilisation de Bootstrap et du Material Design) et de Javascript. L'application utilise également une base de données SQL pour retenir les informations des utilisateurs et des parties, accessible par l'intermédiaire de l'interface PHPmyAdmin.

## La partie serveur/client

Le serveur est assez simple à mettre en place en définissant toutes les routes, c'est-à-dire les différentes adresses URL en les liant aux pages de rendu HTML. Il faut néanmoins plus de réflexion quant aux requêtes et données que l'on souhaite passer à chaque page, par exemple sur la page d'accueil : il nous faut chaque catégorie disponible ainsi que ses points associés, dans le cas où le joueur a déjà testé le quizz correspondant.

Mise à part le routage, il faut gérer des sockets, ce sont des fonctions qui sont exécutées dynamiquement lorsqu'un joueur appuie sur un bouton ou réalise une action précise, par exemple appuyer sur « Play » génère une requête en temps réel pour récupérer une première question à poser à l'utilisateur. Ces sockets sont spécifiques à chaque action et ils font le lien entre l'obtention de données et le rendu graphique, il faut donc faire les bonnes requêtes à la base de données selon le type de jeu et d'utilisateur.

## La base de données

La base de données est composée de 6 tables qui servent à sauvegarder de manière organisée les informations qui sont créées ; on a ainsi une table user, question, category, level, match et game. Ainsi lorsqu'un utilisateur s'enregistre sur l'application, la table user est remplie avec les informations qu'il nous fournit c'est-à-dire son pseudo (en vérifiant l'unicité de celui-ci) et son mot de passe. Pour des questions de sécurité, le mot de passe ne peut pas être stocké en clair dans la base, il nous a donc fallu trouver une solution pour le « hasher ».

C'est à partir d'un problème comme celui-là que l'on peut voir l'avantage d'utiliser un langage comme NodeJS. Il propose des modules à inclure qui permettent d'utiliser des fonctions pré-écrites par la communauté, ainsi nous avons utilisé un module « Password-hash » qui fait lui-même les transformations nécessaires avant l'insertion dans la base de données.

Le code est open source et disponible à cette adresse : <https://github.com/Qu1lby/BigBen>

## Conclusion

---

This project is a success since the application is fully running without bugs and can be played by anyone at any time. We have given the application to be tested to potentials players and we've get good appreciations from them. The game is kind of addictive but we should had more questions by category to improve the randomness of the questions asked.

Further ameliorations could be developed, for instance we could build an administrator panel in order to manage the categories and the questions (add, modify and delete). Currently we have to put manually any data into the database without any automating of the process. We could also handle users' information or see all the game which are played in live. Add bonuses, categories or make a 1 versus 1 player game, there is plenty of possibilities.