# JavaScript for Front-End – Exam Preparation II Log Horizon

You have been tasked to create a simple Front-End application, which works like simple **server logs system**. Website statistics are based on **server****logs**. A server log is a simple text file which records activity on the server. There are several types of server log — website owners are especially interested in **info****logs** which record hits and related information.

Our **server** **log** **application** contains **three** **types** **logs**.

## Task

The task is split into **2 sub-tasks**:

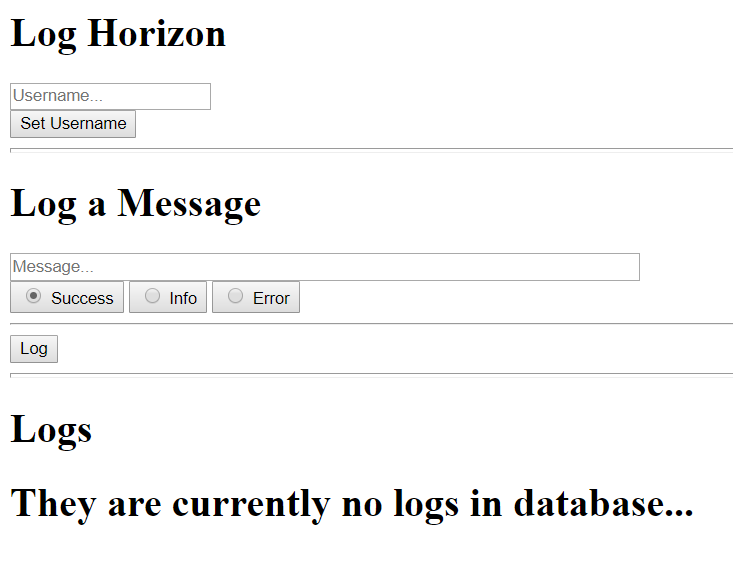
* A simple but fancy **web page design**.
* A simple but dynamic **web page functionality**.

The clients are understanding people, and they understand that you are quite novice as a developer, so they are willing to accept **ONLY** the **web page design** (without **functionality**) or **ONLY** the **web page functionality** (without **design**), if you are **unable** to **implement** one or the other.

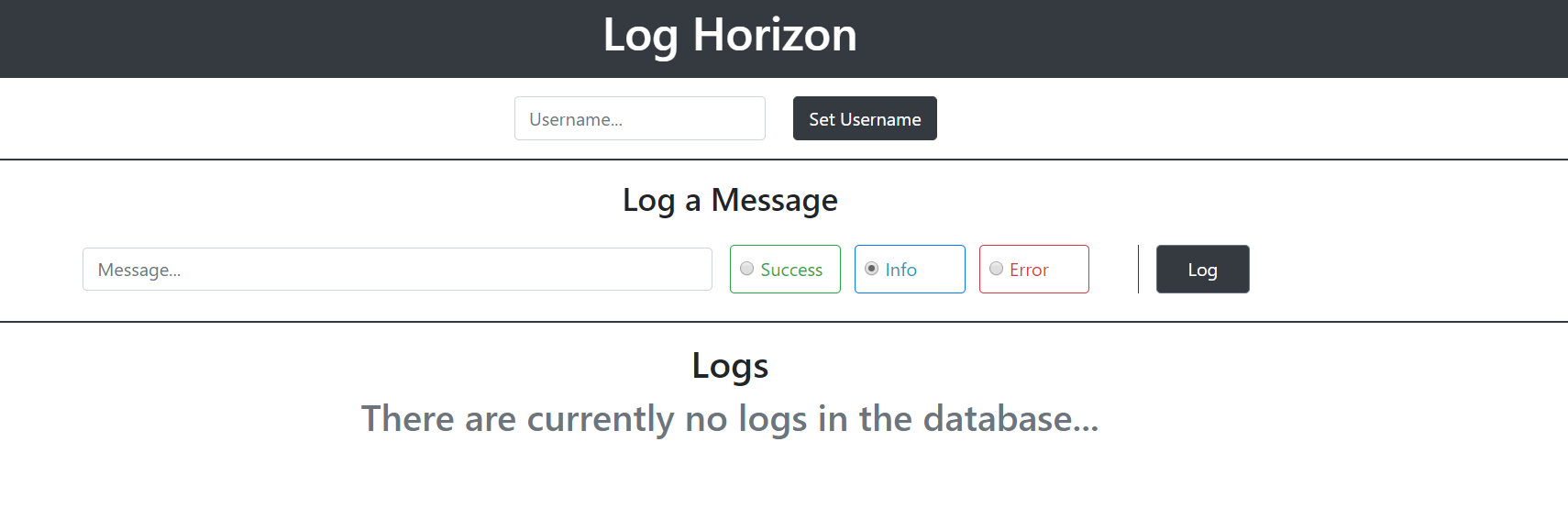
## Overview

### Web Page Design

The web page should look like this **WITHOUT design**:



The web page should look like this **WITH design**:



### Web Page Functionality

The application contains 2 main functionalities – to login and to log messages.

#### Login Functionality

The first functionality is Login.

There is a simple **text input field** on the **top** of the page, with a **button** next to it – [Set Username]. When you **click** the [Set Username] button you should take the **value** of the **input field** as a username and **replace** the **input field** with a **heading** which **contains** that **username**.   
You should also replace the [Set Username] button with a [Logout] button.

When you **click** the [Logout] button you should **delete** the **heading** and **replace** it again with the **input field** from the beginning, and you should also **replace** the **button** again with the [Set Username] button from the beginning.

This is a simple functionality which is a **semi-login / logout toggle**.

#### Log Message Functionality

The second functionality is Log Message.

There is another simple **text input field** in the **middle** of the **page**, with **3 radio buttons** next to it and a **button** – [Log].

When you click the [Log] button, you should **create** a **message**, which is **appended** to the page. The **message** should contain 3 elements:

* The message content - the **value** of the **selected radio button** + the **value** of the **input field**, in the following format:

{selectedRadioButtonValue}: {textInputFieldValue}

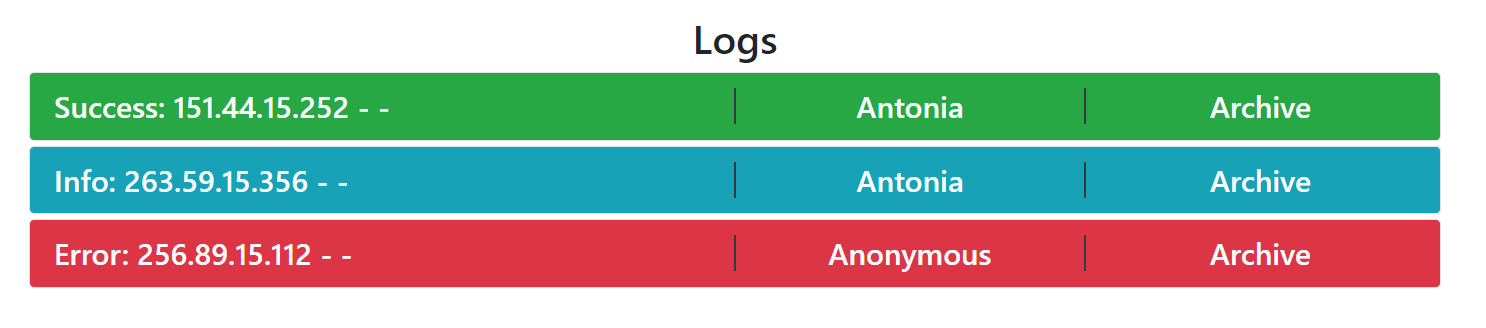
* The logger – the username of the **logged in user**, or “Anonymous” as value, if there is no **logged in user**.
* The [Archive] button – a **button** which would **DELETE** the **message** from the **page** when **clicked**.

As it was specified above there are **3 radio buttons**:

* **Success** (default selected)
* **Info**
* **Error**

Depending on the **selected button** when the **message** is **logged**, the **message** can be one of the **3 types**.

**Each message** has a different **style**, you can see examples below**:**



You will be given a **GIF** representing the functionality of the **web application**.

## Constraints

The clients are not that much into Front-End technologies, however:

* They heard **Bootstrap** is a good web page styling framework, and they have requested that you **use it**.
* They heard JavaScript makes pages quite dynamic, and **jQuery** is a good library, so you must **use it**.

That being said, you have several constraints to follow, during the implementation of the web application:

* You must use **ONLY** **Bootstrap** for the **web page design**.
* You must use **ONLY** native **JavaScript** & **jQuery** for the **web page functionality**.

## Hints

The <hr> (horizontal line) does not quite have a styling with **Bootstrap** which could make it look like the screenshots, so here is a **hint**. Implement **inline style** and set its height to 2px.