**Battleground Login/Registration Pages Replica**

Hyunwoo Kim

## INTRODUCTION

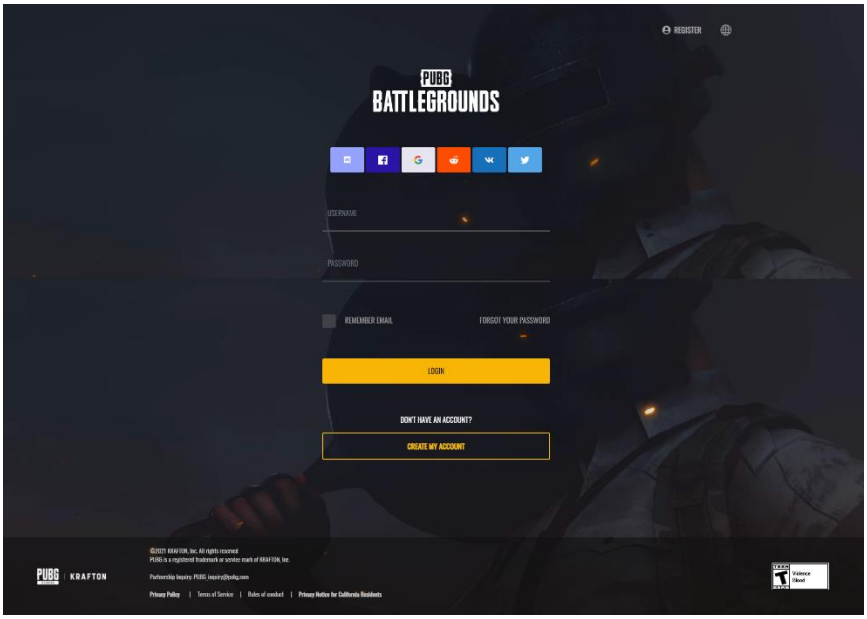
**The idea of the project** is to recreate 'login', 'registration', and 'confirmation' webpages for a standard online game called "battleground" (https://accounts.pubg.com/login). **The primary aim of the project** is to build a web framework that allows a user to access to create and access to his account before his access into the game.

Not only confined to a game website is the identification of a user. Whether it is Ecommerce, SNS, or video-chatting applications, a user is more than often required to create and access to his account to enjoy its services. Reasons behind the identification process may vary depending on kinds of services a website or online platform provides; however, in standard, the process is served to grant users to access to their corresponding "personalized" accounts. **The purpose of my proposed website** is to dabble with front-end and back-end programming to develop my skills as a programmer.

## WEBPAGES CREATED

There are overall 6 webpages created for my replica website for the battleground login/registration pages: index.html, register.html, verification.html, email.html, confirmation.html, and loggedin.html.

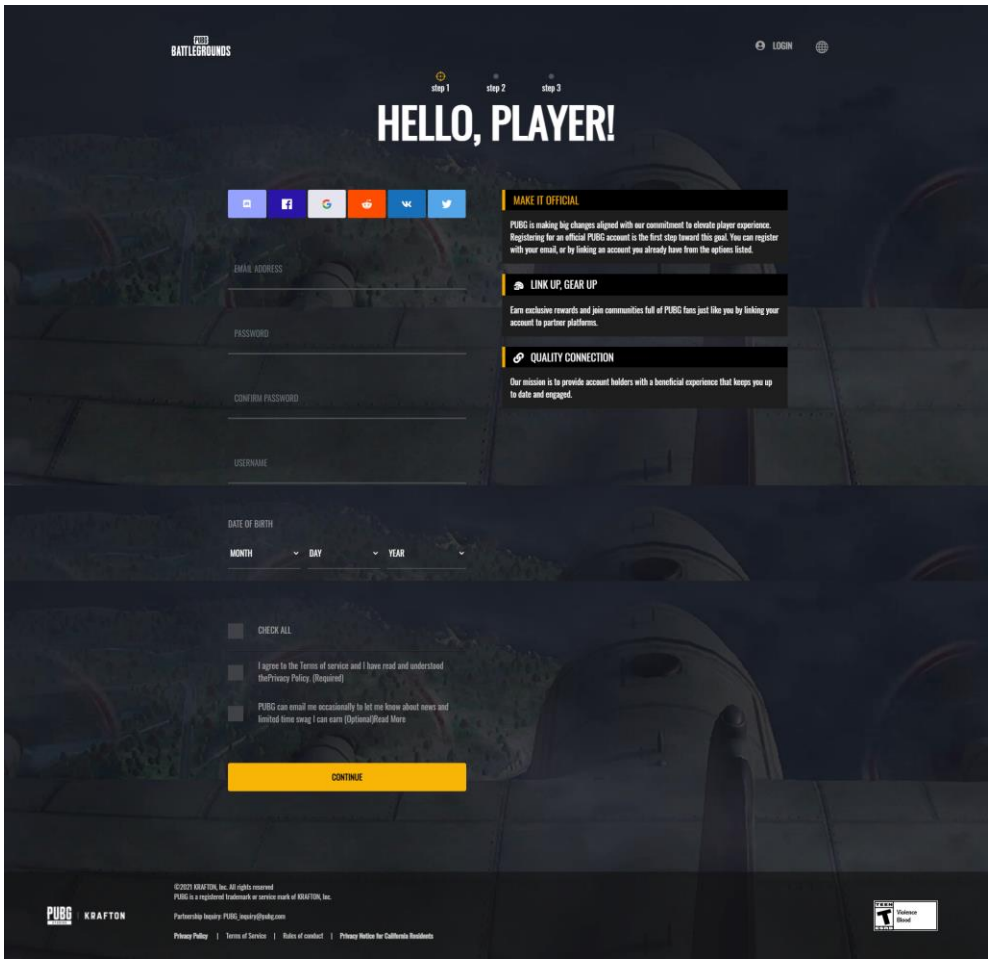
**Index.html (available on http://3.36.88.26)**



This is the homepage of my website where one can login with his account to reach ‘loggedin’ webpage. Inputs of the form are username and password which must match the account information in the database to successfully login. To make your own account, you have to hit the ‘register’ link on the navbar at the top of the page or the ‘Create My Account’ button at the bottom of the page to reach ‘register’ webpage.

All the effects from the original battleground login homepage have been recreated in the document except for language options and ‘forgot my password’ link which has been replaced with an arbitrary link. I opted out such functionalities since they are not very integral to my project of making login/registration page.

**Register.html (available on http://3.36.88.26/registration)**




There are overall four input sections to be filled out to properly continue to the next step of the account making process: text/password input section, date of birth section, and agreement section.

The text/password input section is comprised of four inputs: email address, password, confirm, and username inputs. I have used regular expression codes in JavaScript so that certain conditions should meet to continue through the process. e.g. condition for the correct input of email address is the following: new RegExp(/^[a-zA-Z0-9.!#$%&'*+\/=?^_`{|}~-]+@[a-zA-Z0-

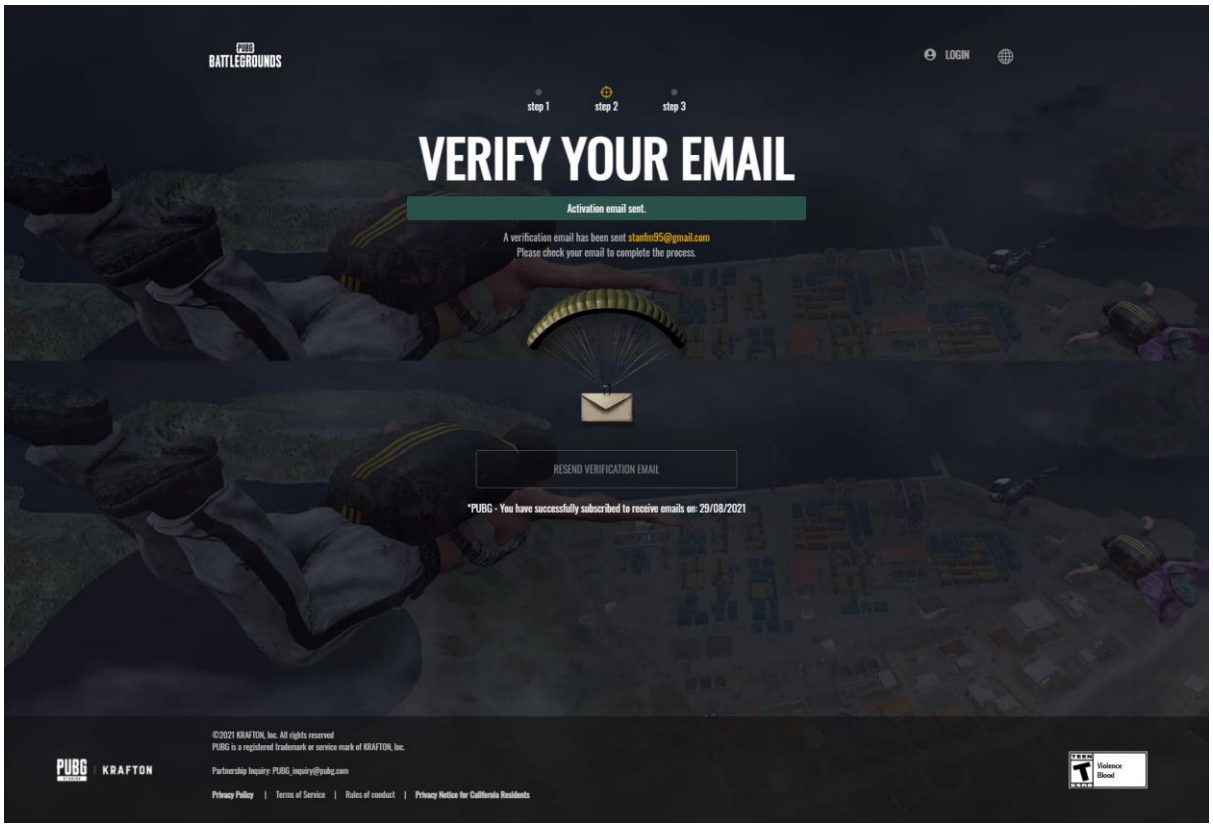
9](?:[a-zA-Z0-9-]{0,61}[a-zA-Z0-9])?(?:\.[a-zA-Z0-9](?:[a-zA-Z0-9-]{0,61}[a-zA-Z0-9])?)*$/);

The date of birth input section is comprised of three ‘select’ parts: Month, Day, and Year. In the original battleground registration, one should be above certain age to continue the process but, in my webpage, as long as you select a month, a day, and a year, one can proceed.

The agreement section is comprised of three checkboxes. I have used JavaScript codes to make sure that if ‘Check All’ checkbox is clicked, other two are clicked and vice versa. Therefore, in either way, if ‘Check All’ checkbox is clicked, one can proceed to the next phase of the account creating process.

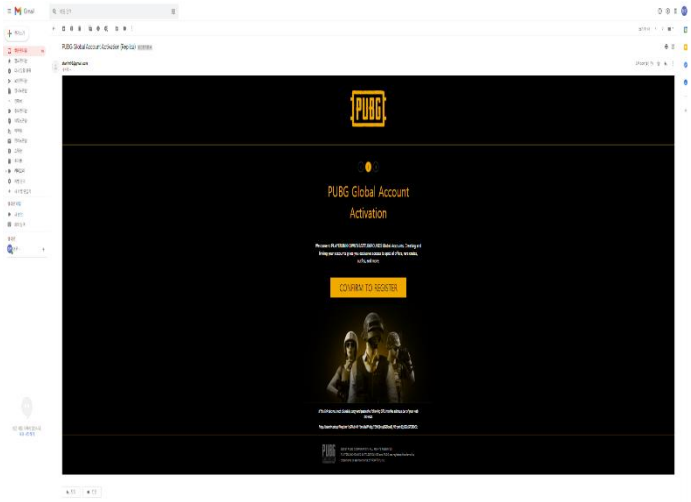
A visitor’s input information are saved in MongoDB. I used MongoDB as the database since I have used it once before. But no doubt, there are other available options such as Posgres SQL which I am prone to use next for my next project.

**verification.html (available on http://3.36.88.26/verification)**




Correct inputs and hitting the continue button in the registration page will lead to verification page and send an email to the input email address. There is a resend button I recreated in this page which resend an email to the input email address. Date shown below the resend button was recreated through datetime package in python.

**email.html (caveat: there is a high probability that it went to your spam email box so please check your spam email box)**



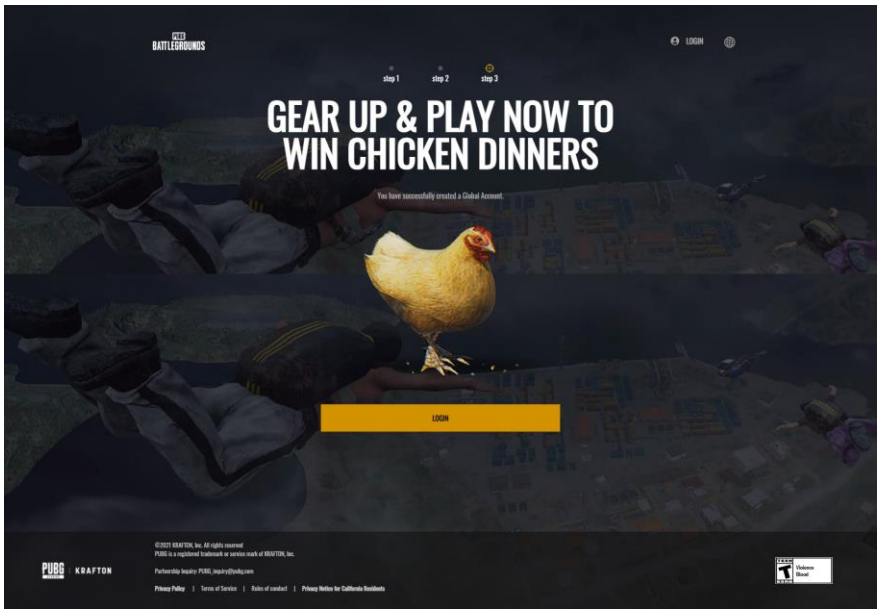

If you reached to verification page, it means that an email is sent you your input email address from 'stanfm95@gmail.com'. If you are not sure about not letting other people know your email address, please use random temporary email service (https://temp-mail.org/).

I used 'smtplib' and 'email.message' package in python for automatic deliver of emails.

I used 'itsdangerous' package in python to serialize confirmation page address with token so that when a visitor reaches to confirmation page, my python application knows which account data to save in the database.

Clicking the 'Confirm to Register' button or using the link at the bottom of the html email will lead a visitor to the confirmation page.

**Confirmation.html (on http://3.36.88.26/confirmation/<token>)**




If a visitor reaches to this page, it means that your account has been successfully saved in the database to login. Click any login buttons or battleground logo at the top of the page to get to

the login page.

**Loggedin.html (on http://3.36.88.26/authentication)**


Correct inputs of the login form will lead a visitor to the authentication page.

## Conclusion

I used only one python file ‘app.py’ for routes, database managing, and initialization mostly with flask package. But I came to realization that it is better to use one python file for each section for better management. Nevertheless, the file runs perfectly, and I am happy about it.

Finally, I took advantage of ec2 of AWS so that my website is running 24/7 on ‘http://3.36.88.26’. Plus, I bought a web domain with a cheap price so that a visitor does not have to type some block of numbers to access to my website but with the address of ‘http://stanfm.shop’

My source code is available on https://github.com/BeatKraQ/BG-Login-Registration

If you want to try something out like this, you can check out the followings:

https://www.youtube.com/watch?v=sndILKXxMsA&list=PLutwis6L8ml-fBrd5p3nhsZ7Y-wduKNPA

https://www.youtube.com/watch?v=2e4STDACVA8