Green Pheasants recommendation system guidelines

*Nir Levy, August 2023*

### **Purpose**

This document provides guidelines to developers, for integrating the Python code that I wrote in the existing [Green Pheasants](https://www.greenpheasants.com/#/home) PWA.

### **Definitions**

Let us begin by clarifying two terms in this document:

User – a person who has a user account and they are logged in to the PWA when they are online.

Visitor – a person who does not have a user account, or who is not logged in to the PWA when they are online.

### **Background**

Green Pheasants aims to provide personal poetry recommendations (see more details about its current functionality [here](https://docs.google.com/document/d/1PVyeR1_FvWXz3Hu3wzE5Hbank6owYDx9/edit?usp=sharing&ouid=102027381488253026282&rtpof=true&sd=true)). Recommendations are currently random, but I have written Python code that provides personal recommendations. In case of visitors, they are based on the actions of previous users. In case of users, they are based both on the actions of previous users in general and on the actions of the specific user who is requesting the recommendation.

**Setting up the recommendation system**

The recommendation system consists of five scripts. The following [table](https://docs.google.com/spreadsheets/d/1k1XA6LFC9VX-iLFXF0TKCtwzBH4AJb96/edit#gid=1638514490) specifies which scripts need to run in each case. Let us elaborate upon the information in the table. The model needs to be trained periodically. There are two training scripts, one for user recommendations and one for visitor recommendations. Run each of these scripts every six hours.

The other scripts provide recommendations in three possible cases, one script for each case:

1. A visitor is online and requests a recommended poem.
2. A user is online and requests a recommended poem.
3. The PWA needs to send recommended poems to all users who chose to receive them by email and/or by push notification.

Please let me know if you have any questions.