Plan for implementing GCBVs in the Final Project:

Estimation of Effort:
The estimation of effort for refactoring the existing function-based view, which is currently responsible for displaying city details, to a Generic Class-Based View (GCBV) such as DetailView will be moderate difficulty. This refactoring process involves comprehending the existing code of the view and making the necessary adjustments to align it with the GCBV structure.

Refactoring all views:
There will be views related to city details, job opportunities, user reviews, and mood representation. The number of views can vary based on the project requirements.For example, there might be a view to display a list of job opportunities specific to a city. Refactoring this view to a GCBV, such as ListView, involves analyzing the existing view code and connecting it to the appropriate classes. I will allocate a significant portion of the project timeline for refactoring the views and ensuring their correct functionality through testing.

Views not suitable for GCBVs:
In some cases, certain views may not lend themselves well to GCBVs due to complex logic that are not easily expressed with the GCBV attributes. For example, let's say there is a view that calculates a city's livability score based on various factors like cost of living, job opportunities, and user reviews. This view involves complex data from different models. Refactoring this view to a GCBV might not be feasible.

Elimination of code lines:
Refactoring to GCBVs can lead to a reduction in the overall lines of code in the project. Let's say there are multiple views responsible for displaying city details, job opportunities, and user reviews. Currently, each view might have its own logic for handling common tasks such as retrieving data, rendering templates, and managing pagination. By refactoring these views to their respective GCBVs (e.g., DetailView, ListView), the shared behaviors can be handled by the GCBVs, reducing the lines of code dedicated to these tasks. I would say a minimum of 20 lines of code would be eliminated, but I know much more could be.