To scale the program to a web application, we would need to first refactor the Controller into a RESTful API. Currently, the Controller is listening to keyboard presses; instead, we would need it to listen to incoming requests from the network. Also, instead of it passing the JSON object it receives from the Model to the View for rendering, it simply needs to pass the JSON object back to the web client that initiated the request. In addition, it would not need to keep track of the page the user is trying to view; that will now be the responsibility of the web client. This ensures that if a particular web server was slow or if it was to crash, the request with the page the user is trying to view can be routed to another server.

For the Model, we should have it cache its responses. The responses change basically only every hour, so there is no need to query the sfgov API every single time a request comes in. This not only ensures that we do not make too many requests to sfgov API and get blocked (though we will also use an API key to help prevent this), but caching will also speed up response time to the web client.

For the View, we will basically scrap it and rewrite it into a single page app, perhaps in React. It will be served to the users from some webserver, ideally an edge server on a CDN, and it will hit our Controller to get the data it needs to display to the user.