**MVC Models and why we are using one**

We chose to use an MVC for our project. Our project basically lets the user see what it is like to code in Python. This would involve receiving and acting upon inputs given from the user (i.e. Python code) and displaying a visual result such as error messages/output (through the view component). The data required would be enough to successfully compile and run python code (which would be a part of the Model Component). Using an MVC will also create a clean separation of duties. The three-pronged system seems perfect for web-related projects since almost every project needs user input, output displays, and a unit that gathers and tells the data what to do. The MVC has proven itself when it comes to the creation of a web-based application system. Our idea is essentially a web application, so this model is a perfect fit.

A close up of a map

Description generated with high confidence