Explaination of how Post View works:

The Discussion Group system consists of:

* Database Discussion Group table.
* JS component to search for users to add into a discussion group
* A pluggable backend system

The system is using SQLAlchemy to access and manage data in the SQLite database. SQLAlchemy is adapted into pythonic domain language.

In the SQLALchemy python code file, Session is established to connect with the database and represents for all objects you’ve loaded or associated with it during the lifespan. The Session includes add() method, commit() method, delete() method. add() is use to place instances in the session. commit() is use to write changes to the database. delete() is use to place an instance into the Session’s list of objects to be marked as deleted.

Controllers consists of the python code of the CreateGroup. Those forms are using Flask as a microframework, which is using jinja as a template engine.

The js component, createGroup.js, is used to search for users to add to a group. This is dependent on the DiscussionGroup.py model which is the database for the Discussion Group.

In the Flask code, the url\_for() function will build a Url for a specific function. The render\_template() function will render a HTML file which base on the jinja and html code. The get\_flashed\_message() passes a message o the next request, which generally is a template.

The templates folder consists of main.html, GroupView.html, CreateGroup.html, PostEdit.html and PostView.html which are base on jinja and html code. The CSS of the html files will be using the bootstrap library.

Controllers file, routes.py, templates file are all included in app file.