**Book Lib - Proof of Concept**

**API Used: Google Books**

**Functionality:**

Allows users to search for books based on title and author. App will contact the Google Books API and receive a number of results based on the sent information. App will then compile received data and according to information to be displayed, display results to user. App will also account for input fields, allow continuous searches, and utilize MVC design patterns

**Development Architecture:**

**MainHandler (CONTROLLER)**:

1. Handles instantiation [var p] of FormPage() sub-class
2. Sets values for [var p] FormPage instantiation’s inputs property
3. Verifies if self.request.GET to run code in relation to MVC design pattern with Bookview(), BookModel(), and BookData() classes
4. Code to run if true will be a BookModel() instance and BookView() instance, gathering values of “title” and “author” from form input results, verifying that inputs have been filled out and writing error message accordingly, and updating body content to reflect the received information from the api if conditions are met

**Page(Abstract Class)**

1. Contain properties with set values for the html (head, body, close)
2. Contain a page\_write function to above properties for writing html to page

**FormPage(sub-class to Page)**:

1. Contain properties for form and form inputs
2. Contain inputs setter to receive values of array to create form inputs for page display
3. Contain a page\_write function that overrides the Page Abstract Class’ page\_write function (POLYMORPHISM)

**BookView(VIEW):**

1. Contain property for bdos(“Book Data Objects”) and bdos setter that runs an update()
2. Contain content(“Info received from API”) property
3. Contain update() function that cycles through information provided from the bdos setter and add html strings to the value of content with values of received do(“Data Object”)

**BookModel(MODEL):**

1. Contain property url with set value of basic API address
2. Contain properties of title, author, and jsondoc with title and author setters used
3. Contain callApi function to create a request, opener, result, and json call to API
   1. Contain variable to contain json call to “items” objects within received API information
   2. Contain property array to contain Data Objects (dos)
   3. Creates an instance of BookData within variable – do
   4. Gathers information from API (title, author, book rating, online read link) and sets do(BookData instance) properties according to each
   5. Adds do(BookData instance) to dos array property

**BookData(class):**

1. Contain properties for title, author, rating, and online read link