

Homework 6: Server-side Scripting

Contents

1. Objectives	2
1.1 Cloud Exercise	2
2. Description	2
2.1. Description of the Home Page	2
2.2. Description of the Search Form	7
2.3. Displaying Results	9
2.4. Saving Previous Inputs	12
3. Hints	13
3.1. How to get Google News API Key	13
3.2 Deploy Python file to the cloud (GAE/AWS/Azure)	13
3.3 List of Category for dropdown	14
3.4 Useful Links	14
4. Files to Submit	15

1. Objectives

- Get experience with the Python programming language.
- Get experience with Google News API.
- Get experience using a JSON parser in Python.
- Get experience using Flask and Django in Python.
- Getting hands-on experience in GCP, AWS or Azure.

1.1 Cloud Exercise

- The backend of this homework must be implemented in the cloud on GCP, AWS or Azure using Python.
- See homework 5 for installation of either one of these platforms. You only have to select one platform to implement your backend.
- **See the hints (section 3) at the bottom, a lot of reference material is given to you.**
- **For Python and Flask kickstart please refer to our updated Lecture 12 slides on class website.**
- **You must refer to grading guidelines, the video, the specs and Piazza. Styling is graded this time and the points breakup is mentioned in guidelines.**
- **The link to the video is: <<<<< video link >>>>>**

2. Description

In this exercise, you are asked to create a webpage that allows you to search for news information using the **Google News API** and the results will be displayed in card format. The *Google News API* is documented here:

<https://newsapi.org/docs>

2.1. Description of the Home Page

A user first opens a page, called **index.html (or any valid web page name)**, where they can read **top headlines** provided by the Google News API along with specific headlines from CNN and Fox News. The top headlines are displayed in a sliding format (Figure 1). The *top headlines* from CNN (Figure 2) and Fox News (Figure 3) are displayed in a card format. A word cloud for frequently used words is displayed as shown in **Figure 1**.



CNN



Figure 1: Initial Home Screen

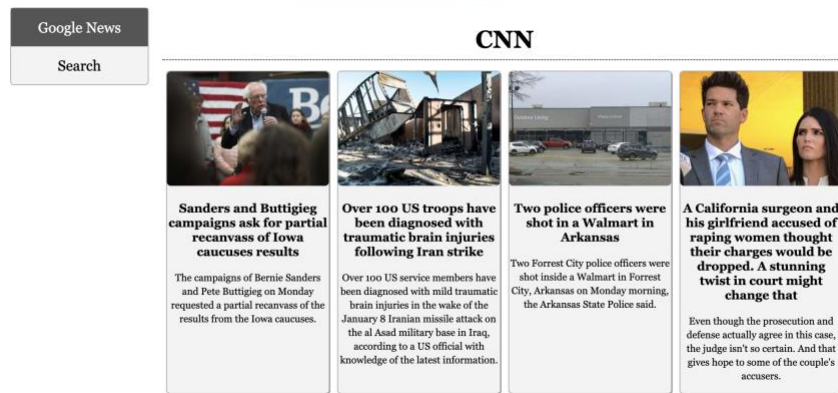


Figure 2: CNN layout on home page

Fox News

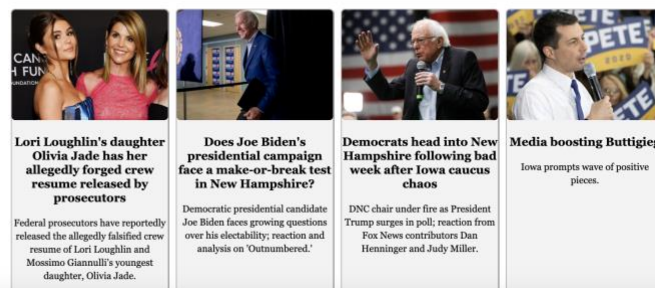


Figure 3: Fox News layout on home page

To access the Google News API using Python, you should use the *Python Client Library* documented here:

<https://newsapi.org/docs/client-libraries/python>

You need to use *Flask* Python framework to make all the API calls. **Using `xmlhttprequest` or any other JavaScript calls for anything other than calling your own backend will lead to a 4-point penalty.**

Define routing endpoints and make your API call from the python backend. The recommended tutorial for *Flask* and more importantly, routing, can be found at the following link:

<https://flask.palletsprojects.com/en/1.1.x/>

To initialize the Google News API client in Python, use this function call:

```
newsapi = NewsApiClient(api_key=[API_KEY])
```

The function call to request the news top headlines can be coded in Python as in the following example:

```
newsapi.get_top_headlines(q='bitcoin', sources='bbc-news,the-verge',  
category='business', language='en', country='us')
```

The URL that will be generated by the Python Client Library looks as follows:

[https://newsapi.org/v2/top-headlines?sources=\[SOURCE_NAME\]](https://newsapi.org/v2/top-headlines?sources=[SOURCE_NAME])

The process to create your API key is explained in section 3.1. When constructing the python request required to obtain the homepage news, you need to provide three parameters:

- The first parameter is your `API_KEY` that you create in section 3.1 for initializing the client
- The second parameter is the news **source** you want the headlines from. For the generic case, there is no source to be used. For CNN and Fox News, the source parameter values should be `'cnn'` and `'fox-news'`
- The maximum number of articles to be returned by the API per page are to be restricted using the `'page_size'` attribute in the client request above and restricting it to 30 articles per page

The **response** of this Python request is a **JSON-formatted object**. **Figure 4** shows a sample response from the request. You need to parse this JSON object and extract some fields as required. The data for the slides and the card layouts is from the corresponding JSON structure as shown in **Figure 5**.

```
{
  "status": "ok",
  "totalResults": 761,
  "articles": [
    {
      "source": {
        "id": "news-com-au",
        "name": "News.com.au"
      },
      "author": "https://www.news.com.au/sport/sports-life/morning-serve-the-sport-stories-you-need-to-see-this-morning/live-coverage/472634c206e58c72b96a111a917a18",
      "title": "Brandon Manning clobbered after using racial slur against Boko Imama",
      "description": "Welcome to the Morning Serve.",
      "url": "https://www.news.com.au/sport/sports-life/morning-serve-the-sport-stories-you-need-to-see-this-morning/live-coverage/472634c206e58c72b96a111a917a18",
      "urlToImage": "https://content.api.news/v3/images/bin/bfb2a402b5fa76162fc20a47f64163d0",
      "publishedAt": "2020-02-11T06:47:17Z",
      "content": "Once again we return to the graceful sport of ice hockey, where throwing hands plays a regular role in proceedings on the rink.\n\nCanadian ice hockey star Brandon Manning learnt the hard way after using a racial slur against Ontario Reign player Boko Imama las_ [+895 chars]"
    },
    {
      "source": {
        "id": "news-com-au",
        "name": "News.com.au"
      },
      "author": "Phoebe Loomes",
      "title": "RUSH HOUR: Sickening moment woman clips young child's nails on flight",
      "description": "Welcome to Rush Hour!",
      "url": "https://www.news.com.au/national/rush-hour/rush-hour-sickening-moment-woman-clips-young-childs-nails-on-flight/live-coverage/211bb3e4a5a37af89462f6583b0e3f7e",
      "urlToImage": "https://content.api.news/v3/images/bin/ac261f7432473f407594e6b0048ac166",
      "publishedAt": "2020-02-11T05:54:25Z",
      "content": "A woman has sickened a fellow passenger after she began clipping her child's nails while on board a flight.\n\nThe passenger snapped a photo of the woman, who had settled in to her seat and was clipping her child's nails. It's a scene you'd perhaps expect to se_ [+1152 chars]"
    }
  ]
}
```

Figure 4: Sample JSON response from Google News API's top-headlines endpoint

```
{
  "status": "ok",
  "totalResults": 768,
  "articles": [
    {
      "source": {
        "id": "news-com-au",
        "name": "News.com.au"
      },
      "author": "https://www.news.com.au/sport/sports-life/morning-serve-the-sport-stories-you-need-to-see-this-morning/live-coverage/472634c206e58c72b96a111a917a18",
      "title": "Brandon Manning clobbered after using racial slur against Boko Imama",
      "description": "Welcome to the Morning Serve.",
      "url": "https://www.news.com.au/sport/sports-life/morning-serve-the-sport-stories-you-need-to-see-this-morning/live-coverage/472634c206e58c72b96a111a917a18",
      "urlToImage": "https://content.api.news/v3/images/bin/bfb2a402b5fa76162fc20a47f64163d0",
      "publishedAt": "2020-02-11T06:47:17Z",
      "content": "Once again we return to the graceful sport of ice hockey, where throwing hands plays a regular role in proceedings on the rink.\n\nCanadian ice hockey star Brandon Manning learnt the hard way after using a racial slur against Ontario Reign player Boko Imama las_ [+895 chars]"
    }
  ]
}
```

Figure 5: Sample JSON response from Google News API for slide and card layout

Slide/Card Data	Data from Google News API response JSON
Image	The value of key "urlToImage"
Title	The value of key "title"
Description	The value of key "description"

Table 1: Mapping JSON data and card display

All news articles are clickable and open the actual news article on a **new tab** using the URL returned in the "url" key of the JSON structure above.

NOTE: The headlines in the slide are limited to 5 headline articles but NOT necessarily the first 5 headline results, but the first 5 that have all keys (author, description, title, url, urlToImage, publishedAt and source with its inner keys) present. The same applies for the card format for CNN and Fox News (4 each).

For the **word cloud**, you need to find the top 30 frequent words from the title of the news article sorted from most frequent to least frequent words. This list of words should **NOT** contain **stopwords**. You can download the **list of stopwords** from the following link:

<https://sites.google.com/site/kevinbouge/stopwords-lists>

Please download the “stopwords_en.txt” file for the homework. To display the list of frequent words, the **d3-cloud** library is to be used. The following links should suffice in helping you display the word cloud:

<https://github.com/jasondavies/d3-cloud/blob/master/examples/browserify.js>

https://www.d3-graph-gallery.com/graph/wordcloud_size.html

The tabs at the left of the home screen – **Google News** and **Search** – have a hover effect as shown in **Figure 6** and in the video.



Figure 6: Hovering on Search while current selection is Google News

Clicking the search button displays a search form as shown in **Figure 7**.

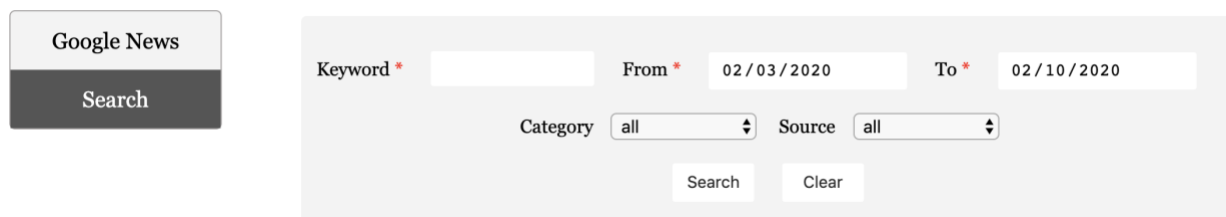


Figure 7: Search Form

2.2. Description of the Search Form

The search form (**Figure 7**) consists of the following fields:

- **Keyword:** A required field that cannot be empty. It is an edit box the allows to enter keywords for searching news about the keyword. The default value is empty.
- **From:** A required field for the date from which the articles are to be fetched. The default value when the form is initially displayed is the date 1 week ago from the current date.
- **To:** A required field for the date to which the articles are to be fetched. The default value is the current date.
- **Category:** A drop down to select among categories that are given to you in section 3.3. Default selection is **all**.
- **Source:** A dynamically changing drop down that changes depending on the selected category. Default selection is **all**.
- **Search button:** To allow search functionality only when a keyword is present. The input field validation is done on clicking the search button and appropriate errors need to be displayed. Missing fields should generate alerts as shown in **Figure 8**. An example of valid input is shown in **Figure 9**.
- **Clear button:** Resets the form to default values and clears search results.

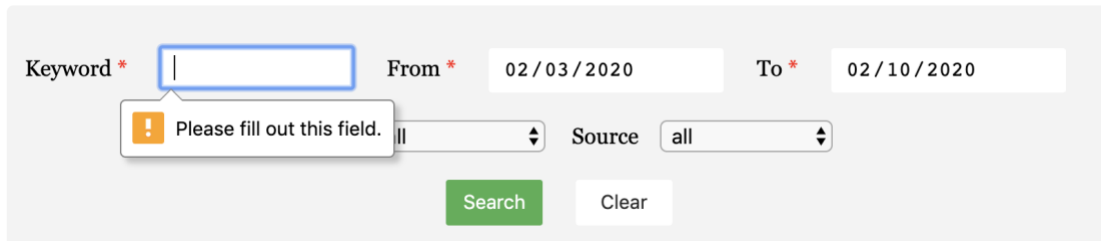
The image shows a search form with the following fields: 'Keyword *' (empty text box), 'From *' (text box with '02/03/2020'), 'To *' (text box with '02/10/2020'), 'Category' (dropdown menu with 'all' selected), and 'Source' (dropdown menu with 'all' selected). A green 'Search' button and a white 'Clear' button are at the bottom. A yellow tooltip with an exclamation mark icon and the text 'Please fill out this field.' points to the empty 'Keyword' field.

Figure 8: Error message for missing keyword

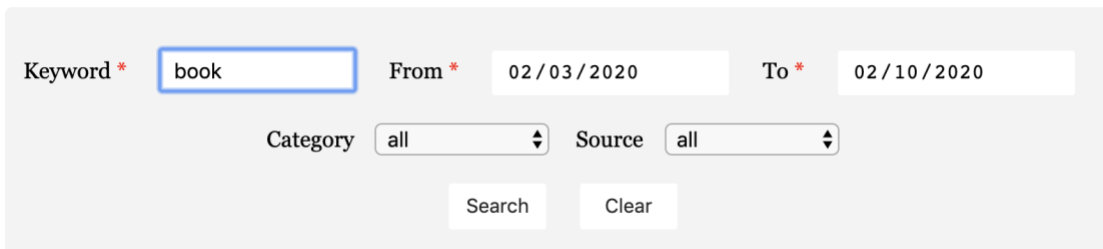
The image shows the same search form as in Figure 8, but with valid input. The 'Keyword *' field now contains the text 'book'. The 'From *' field contains '02/03/2020', the 'To *' field contains '02/10/2020', the 'Category' dropdown is set to 'all', and the 'Source' dropdown is set to 'all'. The 'Search' and 'Clear' buttons are still present at the bottom.

Figure 9: Valid Input example

Another error condition is if the user input time is not in the **mm/dd/yyyy** date format or the **from** date is after the **to** date as like **From** – 02/11/2020 and **To** – 02/07/2020. The error message is shown in **Figure 10**.

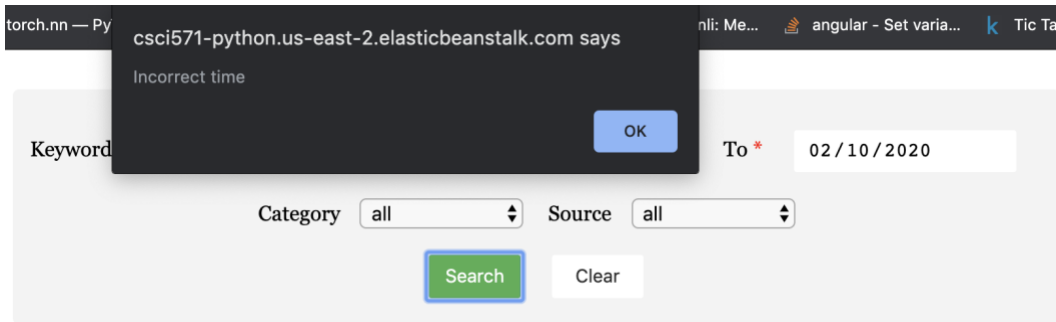


Figure 10: Alert to show incorrect time

NOTE: The dates can be generated using the Date class in JavaScript. While the formatting to display is mm/dd/yyyy, it is up to you to convert it to any format you like for handling it in your code.

Figure 11 and **Figure 12** show the error conditions when the default values of a date are removed completely and are missing some part of the date, respectively.

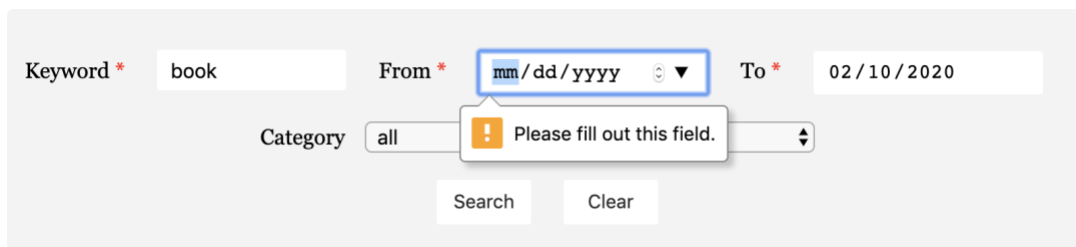


Figure 11: Date is missing completely

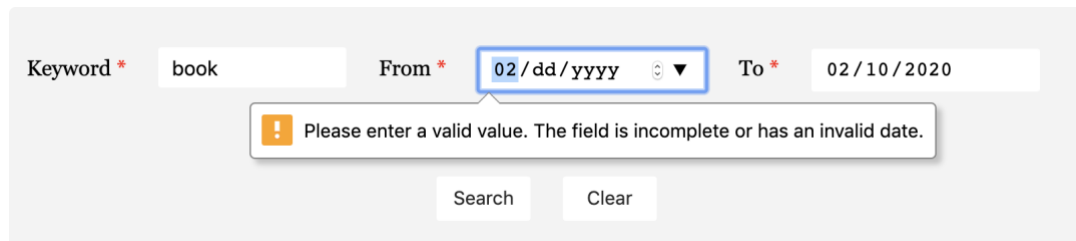


Figure 12: Date is missing the day and year component

The dropdown menu for sources is updated dynamically using the sources endpoint of the Google News API.

```
sources = newsapi.get_sources()
```

Anytime a category is selected in the search form, a call is made to fetch all sources available for that category. The URL that will be generated looks like this:

[https://newsapi.org/v2/sources?category=\[CATEGORY\]](https://newsapi.org/v2/sources?category=[CATEGORY])

The parameters you need to use for the above request are:

- **API_KEY:** The API_KEY generated for the Google News API.
- **category:** The category selected from the dropdown in the search form.
- **language:** The language parameter will always have the value ‘en’.
- **country:** The country parameter will always have the value ‘us’.

2.3. Displaying Results

In this section, we explain how to use the form data to construct a Python request to the Google News API and display the result in your page.

The JavaScript in your file will use the keyword, date from, date to and the source values to construct a URL and send it to your Python endpoint where the request to query the Google News API’s “**everything**” endpoint is made. The URL has the following format

`https://newsapi.org/v2/everything?q=[QUERY]`

This is an example call using the Python client library:

```
newsapi.get_everything(q='bitcoin', sources='bbc-news,the-verge',  
domains='bbc.co.uk,techcrunch.com', from_param='2017-12-01', to='2017-12-12',  
language='en', sort_by='relevancy', page=2)
```

The function call will require the following parameters:

- **API_KEY:** Your API_KEY to initialize the API client.
- **q:** The query keyword retrieved from the search form.
- **from_param:** The date **From** retrieved from the search form as start date.
- **to:** The date **To** which the articles are to be fetched. Retrieved from the search form.
- **sources:** The selected source from the drop down in your search form.
- **language:** The language parameter will **ALWAYS** have the value of ‘en’.
- **page_size:** Restrict to 30 articles per page retrieved.
- **sort_by:** Sort the results by the time they were “*publishedAt*”.

The data returned from the API call is to be displayed as shown in **Figure 14**.

The JSON data returned has the same mapping as shown in **Table 1** on page 4. Only the articles that have author, title, description, url, urlToImage, publishedAt, source and inner key of source

(name) keys present with a non-empty value are to be displayed, if any attribute is missing or empty or blank, do not display it to a user.

If more than 5 articles returned in the results, you should display the first 5 articles along with the **Show More** button underneath. When clicking the button, a maximum of 10 more articles should be displayed and the **Show More** button changes to **Show Less** as shown in **Figure 15**. After displaying the Show More button, at the end of all results (upto 15) there will be a **Show Less** button clicking which will again display just 5 results.

NOTE: The *Google News API* has a restriction on how far back in time it allows the start date for a search to be. As a result of this, you are expected to handle this error case by displaying the **error message returned by the API**, as shown in **Figure 13**.

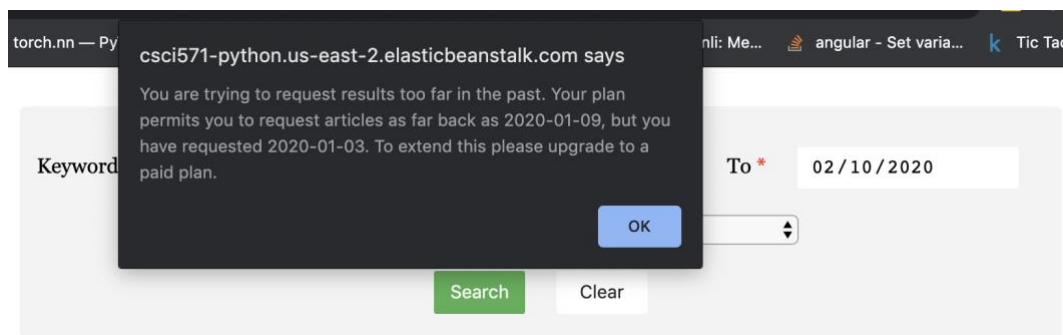


Figure 13: Google News API returns this error

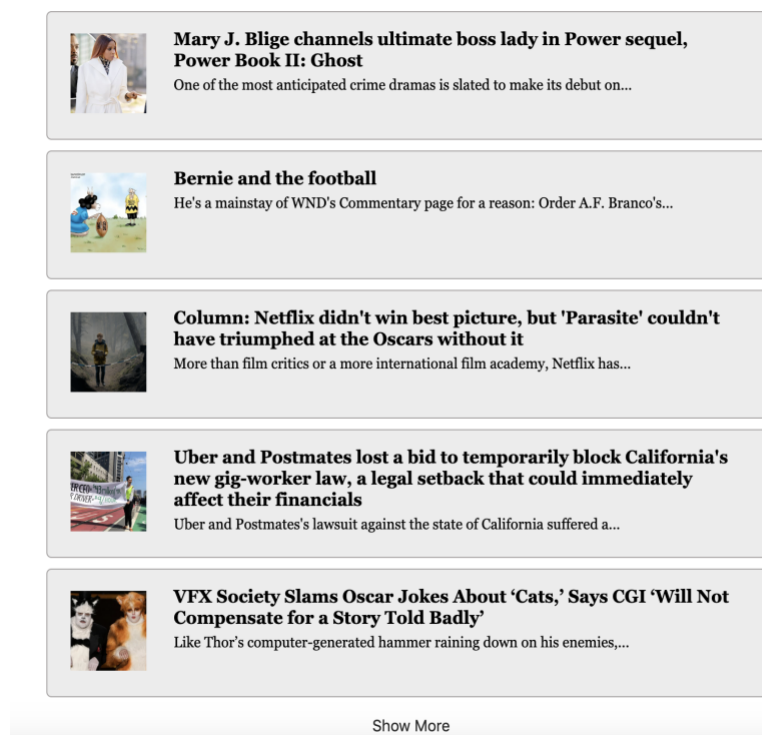


Figure 14: Search Results example

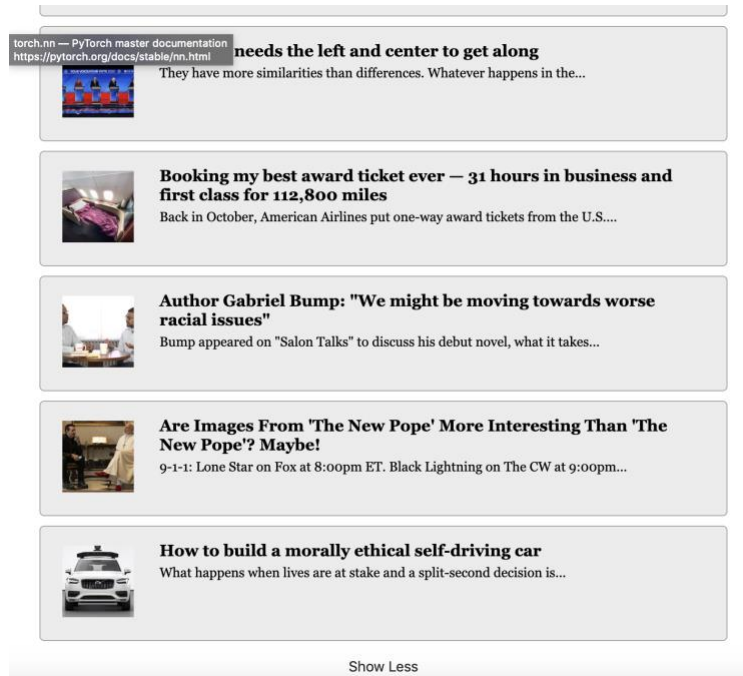


Figure 15: Show More changes to Show Less

The card descriptions need to be truncated with an ellipsis (“triple dots”) without cutting out any word in the middle so as to only span a **SINGLE** line. As can be seen in the reference video and in **Figure 16**, on hovering over the card, the styling changes as shown.



Figure 16: Card styling difference on hover

Each card in the display is clickable and results in additional information being displayed as shown in **Figure 17**.



Figure 17: Expanding the result card

The mapping between the information populated in the result and the JSON object is shown in **Table 2**.

Slide/Card Data	Data from Google News API JSON response
Image	The value of key <i>"urlToImage"</i>
Title	The value of key <i>"title"</i>
Description	The value of key <i>"description"</i>
Author	The value of key <i>"author"</i>
Source	The value of key <i>"name"</i> inside the key <i>"source"</i>
Date	The value of key <i>"publishedAt"</i> converted to the format of mm/dd/yyyy
Link to Original Post	The value of key <i>"url"</i>

Table 2: Mapping between result card and JSON object

The **"See Original Post"** hyperlink opens the original post in a new tab. The **close button** (the "X") at the top right corner of the card returns the card to the collapsed state as shown below.

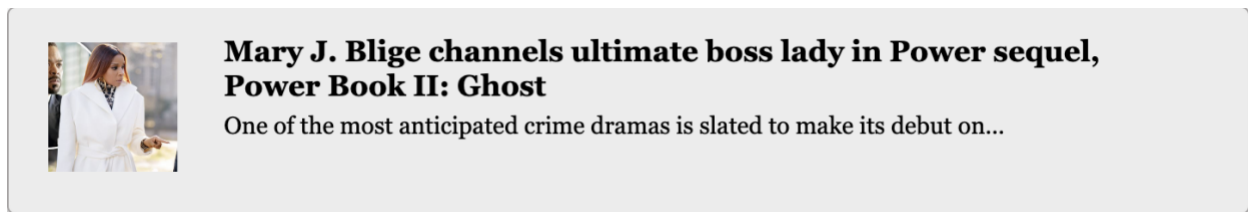


Figure 18: After pressing the **X** button in Figure 16

2.4. Saving Previous Inputs

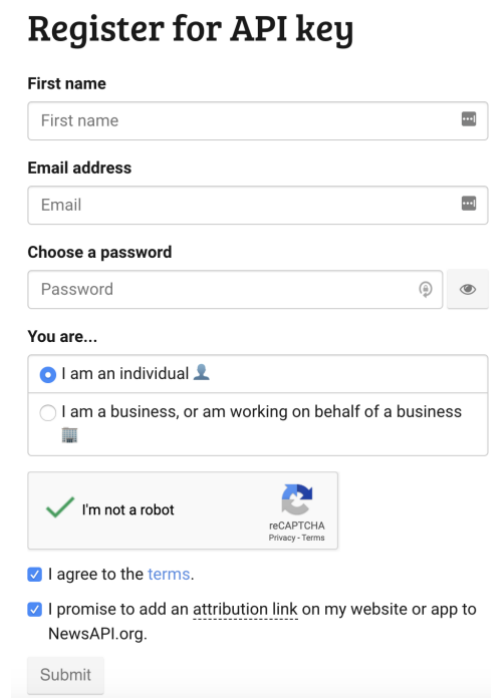
In addition to displaying the results, your page should maintain the provided values to display the current result. For example, if one searches for **"Keyword: Book, From: 02/03/2020, To: 02/10/2020"**, one should see what was provided in the search form and the corresponding results. It follows that you need to retain the whole search box/input fields and buttons even while displaying results/errors. This "saving" happens only when the window is open. You do NOT need a database/Local Storage or any persistence mechanism.

3. Hints

3.1. How to get Google News API Key

Go to <https://newsapi.org/docs>.

Click on “**get API key**”. You should fill out the form as shown in **Figure 19**. The API Key (Google News API key) will be displayed. Copy this key as you will use it in all the Google News web service API calls.



The image shows a web form titled "Register for API key". It contains several input fields and checkboxes. The fields are: "First name" (text input), "Email address" (text input), "Choose a password" (password input with a strength indicator and a toggle for visibility), and "You are..." (radio buttons for "I am an individual" and "I am a business, or am working on behalf of a business"). Below these is a reCAPTCHA section with a green checkmark and the text "I'm not a robot". At the bottom, there are two checked checkboxes: "I agree to the terms." and "I promise to add an attribution link on my website or app to NewsAPI.org." A "Submit" button is at the very bottom.

Figure 19: Google News **Register for API Key** Form

3.2 Deploy Python file to the cloud (GAE/AWS/Azure)

You should use the domain name of the GAE/AWS/Azure service you created in HW #5 to make the request. For example, if your GAE/AWS/Azure server domain is called **example.appspot.com** or **example.elasticbeanstalk.com** or **example.azurewebsites.net**, the following links will be generated:

GAE - <http://example.appspot.com/index.html>

AWS - <http://example.elasticbeanstalk.com/index.html>

Azure - <http://example.azurewebsites.net/index.html>

example in the above URLs will be replaced by your choice of subdomain from the cloud service

3.3 List of Category for dropdown

You should these categories in the respective dropdown in the Search form:

- All
- Business
- Entertainment
- General
- Health
- Science
- Sports
- Technology

3.4 Useful Links

A. For Google News API

- <https://newsapi.org/docs>
- <https://newsapi.org/docs/endpoints/top-headlines>
- <https://newsapi.org/docs/endpoints/everything>
- <https://newsapi.org/docs/endpoints/sources>
- <https://newsapi.org/docs/client-libraries/python>

B. For Word Cloud

- <https://github.com/jasondavies/d3-cloud/blob/master/examples/browserify.js>
- https://www.d3-graph-gallery.com/graph/wordcloud_size.html

C. For Stopwords

- <https://sites.google.com/site/kevinbouge/stopwords-lists>

D. For Flask/Python Backend

- <https://flask.palletsprojects.com/en/1.1.x/>

E Styling hints

- Styling a horizontal rule - https://www.w3schools.com/howto/howto_css_style_hr.asp
- Button styling - https://www.w3schools.com/css/css3_buttons.asp
- Shadow on a card:
https://www.w3schools.com/howto/tryit.asp?filename=tryhow_css_cards2

4. Files to Submit

In your course homework page, you should update the **Homework 6 link** to refer to your new initial web search page for this exercise (for example, **index.html**). Your files must be hosted on GAE, AWS or Azure cloud service. Graders will verify that this link is indeed pointing to one of the cloud services.

Also, submit your source code file to the GitHub Classroom repository so that it can be graded and compared to all other students' source code via the MOSS code comparison tool.

****IMPORTANT**:**

- All discussions and explanations in Piazza related to this homework are part of the homework description and grading guidelines. So please review all Piazza threads, before finishing the assignment. If there is a conflict between Piazza and this description and/or the grading guidelines, **Piazza always rules**.
- You should not use **JQuery** or **Bootstrap** for Homework 6.
- You **should not call the Google News APIs directly from JavaScript**, bypassing the Python proxy. Implementing any one of them in JavaScript instead of Python will result in a **4-point penalty**.