Title: Organize articles

Description:

The project aims to develop a Python website that serves as a platform to organize papers they have published, are in the press, and are under review. The user will manually feed the URL/PDF file to the system, and the app/website will extract useful information, i.e., Title, Year, Author, Number of Citation, Name of Journal, and so on.

The website dashboard will provide a clean and efficient overview of all the research articles, enabling them to find and manage the papers they seek. The website will help you to find your article from the internet and organize it based on your preferences. It will help you to keep tabs on the papers that are in the press and papers that are under review. It will help you to know how many citations you have in each article, the year it was published, and the author's name. Providing information beforehand will also prevent users from downloading the same article again.

Target User:

The website aims to target users who are in the research field. They may either be new in the field and want to keep a record of articles they have read or want to read by organizing it. Another set of users can be distinguished in the field and have many publications. They want to keep records of their published, press, and under-reviewed papers.

- Users who want to read papers from certain authors.
- Users who want to organize their papers based on different options.

Problems:

The website's main purpose is to help users organize (their) papers while reading. The problem of going to the same paper again and again while literature review and downloading that paper is a common one. The website helps address the issue by checking whether the paper you want to read has been downloaded before. It will also help users to organize their papers. The problem of mismanagement and misplacement of papers arises for those users. In addition, those users want to know how many papers are under review and in the press for accurate workflow. The website also aims to address those problems.

Task Vignette 1:

Erik is a graduate student currently attending one of the reputed universities. He wants to join the lab of a well-respected professor. He wants to keep tabs on all the papers from the professor and read those papers before he joins the lab.

- Erik navigates the Python website.
- On his dashboard, he creates a new folder.
- He then opens the search toolbar and copies and pastes the URL (DOI).
- The search toolbar will give him all the information (Title, Author, etc.) in that DOI.

- He clicks the save option to save the information on the dashboard.
- He will follow the same steps for each paper (manually typing URL/DOI).
- **After manually typing all the papers, he wants to organize the paper based on the published year.**
  - He will click the sort-by button on the website's left side.
  - He will have options like Year Published, 1st Author, Number of citations, Read/Unread, and others.
  - He then chooses to organize it with the year published.

Technical Details:

While storing the data:

- Create a Login functionality to make it secure and personalized (Maybe).
- Develop a new folder functionality that will allow users to create several folders.
- Develop search functionality that will allow users to search the title from the internet. Search can be done by URL and/or DOI.
- User interface that will display the search article from the internet and display the information gathered from it. (Maybe a new window with different searched articles and click each to see the details).
- Store the information in a structured way effectively.

While filtering the data:

- Develop a search/filter functionality to search/filter the article stored in the folder. Filter can be done from the information gathered from the internet or Read/Unread.
- User interface that will show all the information (which was gathered from the internet while storing). It can be done by popping information when the cursor is on the article or creating a new window when clicked).

Task Vignette 2:

Shan is a distinguished researcher working in a research lab for many years. She has published many papers and assisted many to publish their papers. She has a folder that contains all her research papers. She is busy and wants to organize her papers in a better way.

- Shan navigates the Python website.
- On her dashboard, she clicks the upload files button.
- She selects her paper and clicks upload.
- The website will recognize the Title and DOI from the PDF file and search the paper online.
- A new window will appear with all the information (Title, Author, etc.).
- She can manually edit the information if there is a typo.
- **After manually uploading all the papers, she wants to organize the paper based on the first author.**
    - She will click the sort-by button on the website's left side.

  - She will have options like Year Published, 1st Author, Number of citations, etc.
  - She then chooses to organize it with the year published.

Technical Details:

- Upload functionality to upload the paper.
- Integrate a library to extract text (information) from the PDF file.
- Display information and validate the information to increase accuracy.
- If necessary, manually change the information before saving it.

Task Vignette 3:

Dan has downloaded many research articles to help him with his literature review. Usually, he reads the paper and takes the information he needs. After a couple of days, he wanted more information and searched for articles on the internet. He found a good chunk of information from one article and downloaded it. After some time, he figured he had already downloaded the articles. He wants to keep a record of articles he downloaded and check each to see if he has downloaded it before.

- Dan goes to the website. He then goes to search for the button and enters the DOI.
- A new window with the information appears, and he selects the paper.
- He clicks the save option to save the information on the dashboard.
- A new window with information like 'The paper is already saved; do you want to save it again.'
- Dan realizes that he already has the article and decides not to download it, saving time and storage.

Technical details

Other technical details are the same as before. On new functionality will be:

- Pop the new window to provide the information and give options to users.

Data Processing/Analysis:

The data provided by the user would be processed and stored in a structured format, such as a database or file storage system. The details can be stored in a database along with the path (DOI) or reference to the associated PDF if URL/DOI and PDF file were given to extract information, respectively. The processing step may involve validating and formatting the data, editing the information, and adding additional information.

- Data will be gathered from the internet. URL is the best way to gather information. However, a PDF can also be uploaded that will scan the DOI from it and use that information to gather information.
- OCR tools for Python like Keras-OCR, Tesseract, and Pytesseract can retrieve information from the pdf file.
- Some tools may retrieve information (Author name, Year Published, etc.) from URLs. (Under review)

Technical Flow:

User Interface

The website will provide a user interface asking users to type the URL or upload the PDF files. The user interface will provide different options like the search function, edit function, and others. A web-based interface may be appropriate for this type of project.

PDF upload/URL:

Users can upload the PDF file or type the URL. It has yet to be decided whether a PDF file will also be saved. It can be an option. On the other hand, the link will always be saved that can be used to download the articles.

Pre-processing:

The process is needed only if you upload a PDF file. The uploaded PDF will be pre-proceeded to make extracting text easy for the OCR tool. This process can involve using only the article's first page for text recognition.

OCR Tool

This process is only needed if you upload a PDF file. PDF files are processed through the OCR tool to extract the information. DOI will be recognized by the OCR tool and used to search the file on the internet.

Data extraction

URL (given manually or by OCR Tool) searches the file online. Some libraries or Tools (not sure at this point) will be used to retrieve data from the internet.

Data Validation and editing

The user will validate the extracted data. The edit option will be present if there is a typo. Users will be able to insert new information according to their needs.

Data Structure and Organization:

Users will be able to organize and structure the articles. They can organize articles based on different options. In addition, they can also tag additional information to the articles.