Workflow for Citation Project

|  |  |
| --- | --- |
| Project Name | Citation project |
| Project Client | Melissa (Ole Miss Pharmacy) |
| Project Overview | The project utilizes React for the frontend and Flask along with the Scopus API for the backend. It aims to facilitate the extraction and organization of citations related to pharmacy research based on user-defined parameters such as publication year and citation type (annual or quarterly). |
| Goals  & Objectives | Goals:   * Facilitate Citation Retrieval: Enable users to retrieve citations from Scopus API based on specified criteria. * Automate Citation Management: Automatically generate RIS files containing extracted citation data for easier reference and management. * Containerization: Ensure portability and ease of deployment using Docker containers.   Objectives:   * Develop a user-friendly frontend interface using React for inputting search parameters. * Implement a backend API using Flask to handle user requests and interact with the Scopus API. * Create a structured query mechanism to retrieve pharmacy-related citations. * Generate RIS files containing citation metadata including type, publication year, DOI, and authors. * Containerize the application to manage dependencies and deployment consistency. |
| Constraints  & Assumptions | **Constraints:**   * **Query Limitations:** The query mechanism is tailored primarily for pharmacy-related citations. * **Dependency on Scopus API:** Relies on the availability and functionality of the Scopus API for citation retrieval. * **File System Permissions:** Assumes appropriate permissions to write files into the RISFiles folder.   **Assumptions:**   * **Pharmacy Focus:** Assumes queries will predominantly focus on retrieving citations related to pharmacy research. * **User Input Consistency:** Assumes users will provide valid inputs through the frontend form. * **Docker Environment:** Assumes Docker will be used to manage and containerize the application effectively. |
| Project Scope | The project scope encompasses:   * **Frontend Development:** Designing and implementing a form in React for users to input search parameters (publication year, search type, quarter number). * **Backend Development:** Creating Flask endpoints to receive user input, construct queries, interact with the Scopus API, and generate RIS files. * **Integration with Scopus API:** Utilizing the Scopus API to retrieve citation data based on user queries. * **File Management:** Storing generated RIS files in a designated folder (RISFiles) within the project's root directory. * **Containerization:** Dockerizing the application to ensure seamless deployment and management across different environments. |
| Target Audience | Researcher |
| Tech Stack Used | **Frontend:**   * **React**   + **Description:** React is a JavaScript library for building user interfaces, particularly single-page applications. It allows for efficient rendering and state management, making it suitable for dynamic and interactive web applications.   + **Key Features:** Component-based architecture, virtual DOM for performance optimization, JSX syntax for combining HTML and JavaScript.   **Backend:**   * **Flask**   + **Description:** Flask is a lightweight Python web framework that provides tools, libraries, and patterns to build web applications. It is designed with simplicity and flexibility in mind, allowing developers to choose the components they need.   + **Key Features:** Routing system, HTTP request handling, Jinja2 templating, integration with WSGI servers.   **Libraries and APIs:**   * **configparser**   + **Description:** Python module for working with configuration files. It allows your Flask application to read settings from configuration files, enhancing flexibility and maintainability. * **pybliometrics**   + **Description:** A Python library that provides access to the Scopus API, enabling retrieval of publication and citation data from Elsevier's Scopus database.   + **Key Features:** Interface with Scopus API endpoints, search functionalities, retrieval of abstracts and citation details. * **json**   + **Description:** Standard Python library for parsing and manipulating JSON (JavaScript Object Notation) data. It facilitates handling JSON data exchanged between frontend and backend components. * **datetime**   + **Description:** Standard Python module for manipulating dates and times. Useful for handling date-related operations, such as filtering and formatting publication years in your application. * **flask-cors**   + **Description:** Flask extension for handling Cross-Origin Resource Sharing (CORS), which is necessary when your frontend (running on a different domain or port) needs to make requests to your Flask backend. * **shutil**   + **Description:** Standard Python module for file operations, including copying, moving, and deleting files. In your project, it appears to be used for managing the storage of generated RIS files in the RISFiles folder.   **Project Architecture:**   * **Component Architecture:** Frontend components in React, structured as reusable modules for managing user interactions and data presentation. * **RESTful API:** Backend implemented in Flask, providing RESTful endpoints for handling user requests, interacting with the Scopus API, and managing data. * **Data Flow:** User inputs are captured in the React frontend, sent to the Flask backend via HTTP requests, processed using pybliometrics for Scopus API interactions, and results are formatted and returned to the frontend as JSON responses. * **Deployment:** Containerized using Docker for easy deployment, ensuring consistency across different environments and simplifying dependency management. |
| Implementation Instructions |  |
| Timeline |  |