**Using Swagger to Expose an API**

First "flask-swagger-ui" is installed by running the command **pip install flask-swagger-ui** in the Terminal window of VS Code.

The "Activity 11.4.zip" file is unzipped, and the folder is opened in VS Code to run the application.

A screen shot of a computer

Description automatically generated

The running application, displaying the My Books Site website, is seen by pointing the browser to "[http://localhost:5000](http://localhost:5000/)".

A screenshot of a computer

Description automatically generated

The Swagger UI is accessed by pointing the browser to "<http://localhost:5000/swagger>".

A screenshot of a computer

Description automatically generated

The books section is expanded by selecting the "books GET" button in the Swagger UI. Then, the "Try it out" option is chosen, and the "Execute" button is clicked. The cURL request and response section are observed.

A screenshot of a computer

Description automatically generated

The book "The God Delusion" by Richard Dawkins is added by selecting the "POST" method in the Swagger UI. The author name "Richard Dawkins" and book title "The God Delusion" are entered, and the "Try it out" option is selected. The addition of the new book is verified by examining the response.

A screenshot of a computer

Description automatically generated

In conclusion, this project successfully demonstrated the implementation and usage of Flask, Swagger UI, and RESTful API principles to create a web application that manages and displays a collection of books. The use of Swagger provided an interactive interface to interact with the API, allowing for easy testing and validation of endpoints. Through this project, the process of defining, documenting, and consuming APIs was showcased, providing valuable insights into modern web application development and API design practices.