Project :- Convert HAR file to SWAGGER file

1. Overview

The ".har file to Swagger file" project aims to develop a tool that can convert HTTP Archive files (.har) into Swagger/OpenAPI specification files. The tool will provide an automated and streamlined solution for transforming network traffic data captured in .har format into a Swagger specification, which is widely used for documenting RESTful APIs.

2. Motivation

HTTP Archive files (.har) are commonly used to record and analyze network traffic data. However, when it comes to documenting APIs, Swagger is a popular choice due to its rich features, ecosystem support, and compatibility with various API development tools. Converting .har files to Swagger specifications manually can be time-consuming and error-prone. This project aims to automate this conversion process, making it more efficient and reliable.

3. Key Features

1. **Har File Parsing**: Develop a parser to read .har files and extract relevant information, such as HTTP requests, headers, parameters, and response details.
2. **Swagger Generation:** Implement a logic to transform the parsed data into a Swagger/OpenAPI specification file.
3. **Schema Inference**: Analyze the captured request and response payloads to infer data schemas and generate corresponding Swagger schema definitions.
4. **Path and Operation Generation**: Map the captured requests to Swagger paths and generate corresponding HTTP operations (GET, POST, PUT, etc.) with relevant details.
5. **Parameter Extraction**: Extract path parameters, query parameters, and request body parameters from the captured requests and include them in the Swagger specification.
6. **Response Mapping**: Map the captured response details, including status codes, headers, and response bodies, to appropriate Swagger definitions.
7. **Web Interfaces**: Provide Web interface user interfaces to facilitate easy usage and interaction with the conversion tool.

4. Potential Technologies

The project can be implemented using a combination of the following technologies and tools:

1. Programming Language: Python for parsing and generating files
2. Libraries: Python libraries like "haralyzer" for parsing .har files
3. Swagger/OpenAPI Libraries: swagger_ui for Swagger Doc Editor Generator
4. User Interface: Flask and web-based front-end is html and javascript for the graphical user interface

5. Steps to Build and Run Project without Docker

1. Clone GitHub repository for harSwag.(https://github.com/pratikc10/harSwag)
2. Go to the root directory by executing "cd harSwag"
3. We need to create .venv and install the required Python dependencies mentioned in the ‘requirements.txt’ file

```python
pip install -r requirements.txt
```

4. Now execute the command ‘python app.py’ to run the application
5. Open the browser and go to http://127.0.0.1:5000

6. Steps to Build and Run Project with Docker

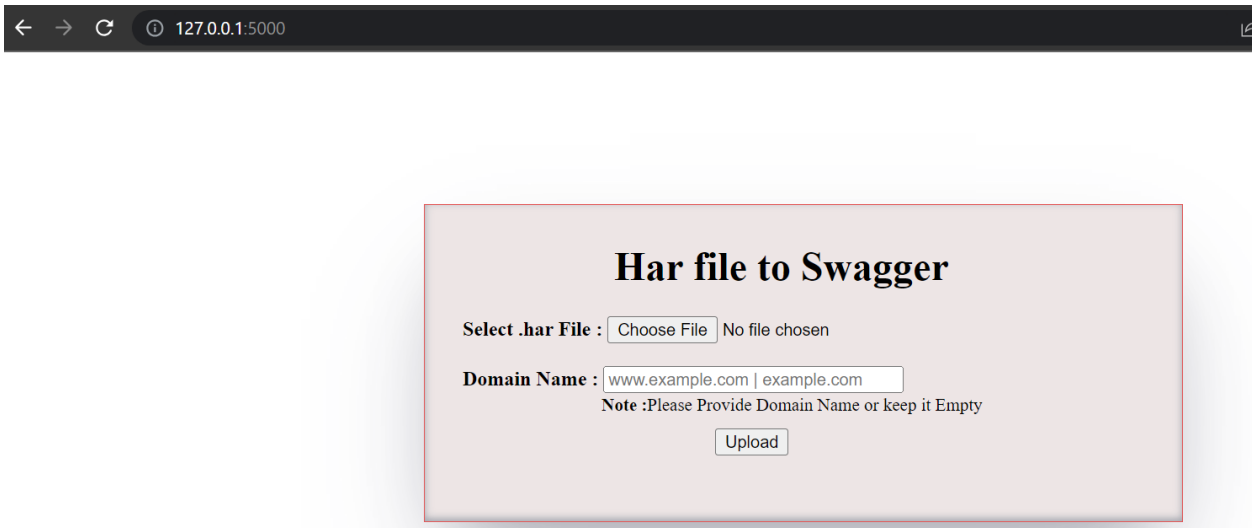
1. Clone GitHub repository for harSwag.(https://github.com/pratikc10/harSwag)
2. Then execute cd harSwag to go into the project root directory
3. Now execute the below command to build the docker image

```
docker build -t harswag:latest .
```

4. Now execute the below command to start the docker container

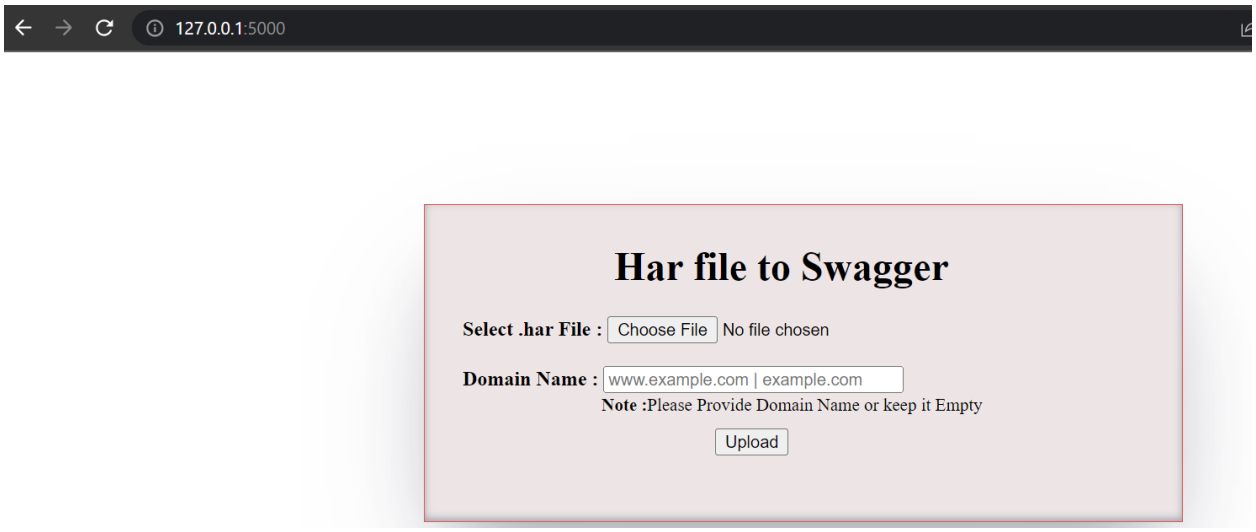
```
Unset
docker run -p 5000:5000 harswag
```

5. Open the browser and go to http://localhost:5000



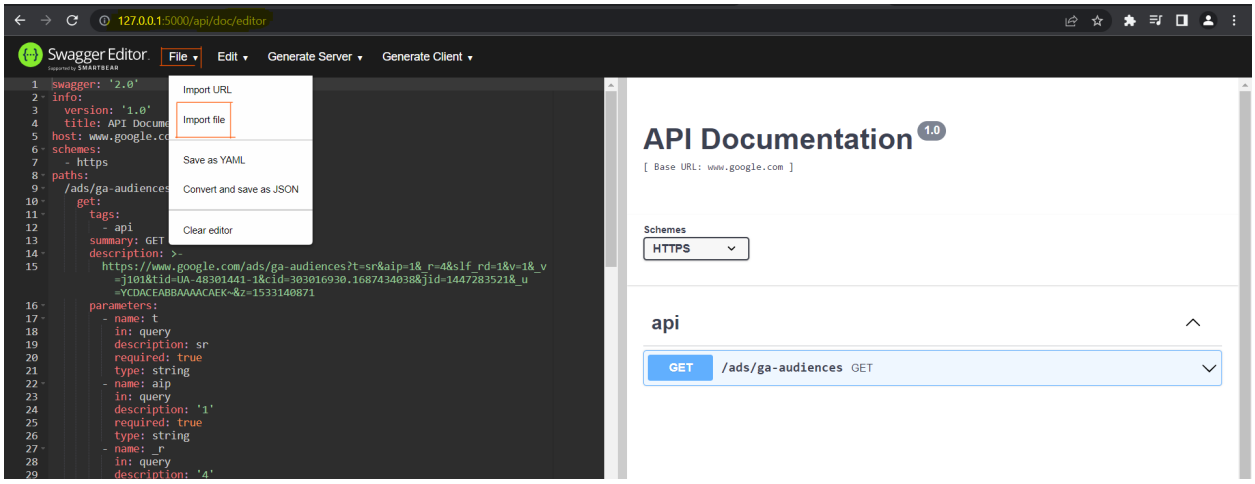
7. Functional Use Case

1. GO to http://127.0.0.1:5000/ URL link we can see the UI interface



2. "Select .har File" From this we can add our .har file
3. In the Domain Name we can put the domain name as per the .har file or we can keep it empty
4. After uploading the .har file It will generate a "har_Swagger.json" json file

5. Go to http://127.0.0.1:5000/api/doc/editor it will redirect the swagger editor, after that from the “File” section we can “Import file” for importing the json file which is downloaded in the previous step



7. On the right-hand side we can get the Swagger API documentation properly.