DOCUMENTATION

PROBLEM STATEMENT LINK :   
<https://sprinklr-my.sharepoint.com/:w:/p/sagar_savaliya/EVQfVmY0W3FCsZxQpBkE9BABxUmraTYSjLHxrq6A9z-ULA?e=rezMtR>

INTRODUCTION :

Developing a link preview scraper tool that extracts information from a given URL and generates a JSON-formatted preview of the content. The tool supports generating previews for LinkedIn, Facebook, and Twitter, and exposes an API endpoint for accessing the previews.

LOGIC APPLIED FOR EACH CHANNEL :

* TWITTER –

1. **Twitter Cards Check**: When a link is shared, Twitter first looks for Twitter Cards metadata.
2. **Fallback to OG Metadata**: If Twitter Cards are missing, Twitter checks for Open Graph (OG) metadata.
3. **Ensured Preview**: This fallback ensures Twitter can generate a preview card even without Twitter Cards, using OG metadata if available.

* FACEBOOK –

1. Facebook uses Open Graph (OG) tags to generate link previews. It checks for these tags in the webpage's metadata and utilizes their information for creating previews.
2. If OG tags are missing or incomplete, Facebook scrapes the webpage to gather additional information for generating a preview.
3. If an OG image tag exists but the associated image’s dimensions are less than 200x200 pixels, Facebook may disregard it as insufficient for a quality preview, and it may continue to scrape the webpage to find the first image above 200x200 pixels to preview it as the image.

* LINKEDIN –

1. Initially, LinkedIn checks for Open Graph(OG) tags on the webpage. If OG tags are present, LinkedIn uses the information provided.
2. However, if OG tags are missing or incomplete, LinkedIn proceeds to scrape the webpage for images. During this scraping process, LinkedIn looks for the first image with dimensions greater than 120x120 pixels.
3. Unlike Facebook, LinkedIn doesn’t enforce a strict minimum dimension requirement for OG images.

TESTCASE SHEET LINK :   
<https://docs.google.com/spreadsheets/d/1Z6KfiQSIdbg3SJLNWrFgGCjasjapqBVi/edit?pli=1&gid=1790451902#gid=1790451902>  
  
POWERPOINT PRESENTATION LINK :

https://sprinklr-my.sharepoint.com/:p:/p/chinmay\_sharmat/Eac1Nkry5XlJmRoAniVwMkcBP\_EuQiZPtbPb9gg8aGZ0Dw?e=Jd3eX2&wdLOR=cCA0D1461-55B0-D244-8439-43EFEDABF29F

SETUP :

1. Clone the repository from :

https://github.com/ChinmayS114/Link-Preview---Sprinklr.git

2. Import the project into an IDE.

3. We have used Maven in this project, so you need to run the following command in the terminal in the appropriate directory : i.e. run the following command in "springrest" directory.

command : mvn clean install

4. After setting it up, run it in src/main/java/service/ LinkPreviewService.java file.

5. Port number 8080 will be used for the API endpoint. You can send request in the local system through the browser or Postman also. Requests should be sent in the following format:

localhost:8080/api/preview?url=

Specify the required url after the '=' symbol.