



# **EUL - A UN SDG CLASSIFIER AND REPOSITORY**

**Paul Joshua Premacio**

2025



## **Why EUL?**

EUL is just an item in Dota 2, and I came up with that name because I was too lazy to think of other names.

## **What is it about?**

It is a full-stack application that classifies documents into 17 UN Sustainable Development Goals. This application doesn't just tell you what category the document belongs to, it also tells you how much that document belongs to a specific SDG by outputting the percentage. Once classified, the user can choose to store the document along with the tagged SDG and percentage in the system.

## **UN SDG?**

The United Nations Sustainable Development Goals (UN SDGs) are a set of 17 global objectives designed to address the world's most pressing challenges, ranging from poverty, education, and health to climate action and sustainable development. Adopted in 2015 as part of the UN's 2030 Agenda, the SDGs aim to create a more just, inclusive, and environmentally sustainable future for all.

They are highly relevant for guiding research, policymaking, and innovation, and serve as a framework for aligning academic and professional efforts toward real-world impact. (This section is generated by ChatGPT btw, xD)

## **Why SDG?**

Why not?



## Tech Stack

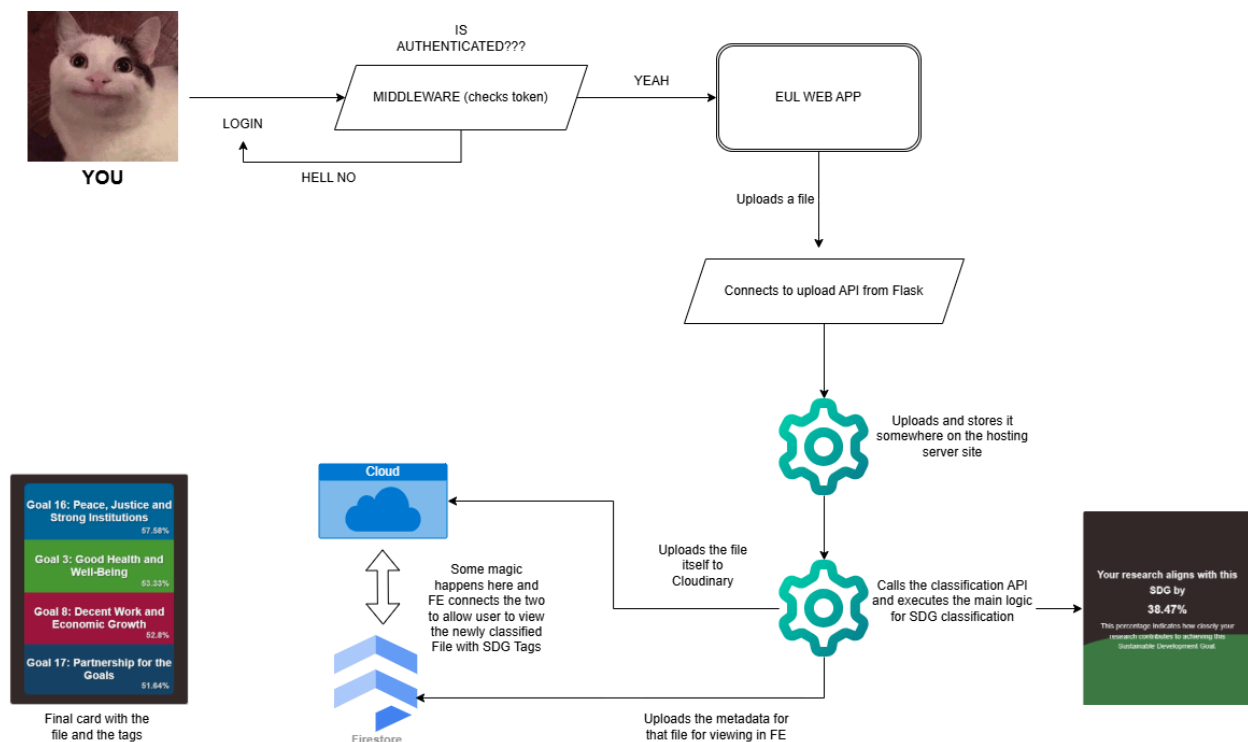
For Frontend, it is using NextJS (React + Typescript), a no-brainer decision since there are a couple of routes, and for SEO purposes.

For the database, Cloud Firestore is used, which avoids unwanted complexity when deploying since it's already in the cloud.

Python Flask is used for the backend algorithm; the algorithm is written in Python and hosted on a Flask server using REST API services.

The uploading and storing of the files are being handled by Cloudinary through its admin API.

## What's the Flow?



## How Does the Classifier Work?

The training for the model is already done before being deployed. The training set comprises 3-5 PDF files worth of content for each SDG. After which, a function is called to process the training set into vectors, and in its final form, is the TFIDF file. So once the user uploads a document, a new item is appended to the TFIDF, which is now the 18th item on the TFIDF. It will then calculate the cosine similarity of those vectors based on the TFIDF and will return the percentages for each goal. Confusing? Yeah, I know. Just download the research paper on the site, and it will explain everything, or schedule an interview with me 😊

