# Email Duplicate Cleaner Web Application

# Project Documentation

## Project Overview:

## The Email Duplicate Cleaner Web Application is a Flask-based project that allows users to upload a .txt file containing email addresses, extract valid email addresses using regular expressions, filter by selected domain names, and separate them into unique and duplicate lists. The app provides an intuitive interface and demonstrates essential web development concepts with Python and Flask.

## Objectives:

* Build a basic web application using Python and Flask.
* Perform email extraction using regular expressions.
* Allow domain-based email filtering.
* Identify and separate unique and duplicate emails.
* Enable downloading results via text files.
* Showcase flash messaging and input validation.

## Technologies Used:

* Backend: Python, Flask
* Frontend: HTML, CSS, Jinja2
* Styling: Basic CSS
* IDE: Jupyter Notebook
* Deployment: Localhost
* Version Control: GitHub

## Project Structure:

Email\_duplicate\_cleaner/  
├── app.py # Main Flask application  
├── templates/  
│ └── index.html # HTML template for the web interface and CSS style   
├── LICENSE # MIT License  
└── README.md # Project overview and instructions

## How It Works:

* User uploads a .txt file containing a list of emails.
* Flask reads the file, decodes the content, and extracts valid emails using regex.
* If desired, emails can be filtered by selected domains (e.g., gmail.com, yahoo.com).
* Emails are split into **unique** and **duplicate** categories.
* Results are rendered on the same page using an HTML template.
* Users can download unique or duplicate emails as .txt files.

## How to Run Locally:

* Requirements:  
  - Python 3.13.3  
  - Flask (Install with pip install flask)
* Steps:  
  - cd email duplicate cleaner  
  - Email\_Duplicate\_Cleaner\_flask.py  
  - Visit http://127.0.0.1:5000/ in your browser.

## Features:

* Upload and process .txt files containing email addresses.
* Regex-based email extraction.
* Optional filtering based on domain suffix (e.g., @gmail.com).
* Automatic separation into unique and duplicate email lists.
* Download results as plain text files.
* Flash messaging for warnings, info, and errors.
* Auto-launch of the default browser when app starts.

## License:

This project is licensed under the MIT License. See the LICENSE file for details.

## Future Enhancements:

* Integrate live email validation API.
* Support uploading and parsing .csv files.
* Export results to Excel (.xlsx) format.
* Add user login system and session-based history tracking.
* Improve UI with responsive design and JavaScript enhancements.

## Author:

* Name: ARI R
* GitHub: