**Weekly Report on OCR of Bank Statements**

**(Web Scraper Development)**

Objective:

Develop a Python-based tool to:

* Download financial-related images using Bing Image Downloader.
* Fetch/download images from the Cloudinary cloud platform using its API.
* Implement file downloading functionality from URLs.
* Upload selected images to Cloudinary for secure storage.

Day 1: Initial Setup and Environment Preparation:

Tasks:

* Set up the development environment and installed required libraries (requests, bing\_image\_downloader, and cloudinary).
* Verified Bing Image Downloader functionality with sample searches.
* Established the project structure and initial class definition.

Outcome:

* Environment setup completed, and tools verified.

Day 2: Implementing Image Fetching Functionality (Bing):

Tasks:

* Developed the fetch\_bing\_images method to download financial-related images based on predefined terms.
* Ensured output directory creation for categorized storage.

Outcome:

* Successfully downloaded images using Bing Image Downloader.

Day 3: Refining Download Settings and Error Handling:

Tasks:

* Enhanced fetch\_bing\_images by setting download limits, enabling adult content filtering, and adding timeout/error handling.

Outcome:

* Improved robustness and reliability for large-scale downloads.

Day 4: Fetching Images from Cloudinary:

Tasks:

* Integrated Cloudinary API to fetch images stored in the cloud platform.
* Developed functionality to download images from Cloudinary to a local directory, maintaining folder structure and naming conventions.

Outcome:

* Successfully fetched and downloaded images from Cloudinary for local use.

Day 5: Implementing File Downloading and Cloud Uploading:

Tasks:

* Created the download\_file method to download files from URLs with error handling.
* Uploaded verified financial images (e.g., bank statements, invoices) to Cloudinary.

Outcome:

* Completed reliable file downloads and Cloudinary uploads.

Day 6: Testing, Documentation, and Cleanup:

Tasks:

* Conducted rigorous testing of all methods to ensure reliability.
* Documented code with detailed comments and function descriptions.
* Created a README.md file with setup and usage instructions.

Outcome

* Delivered a clean and well-documented project.

Day 7: Final Review and Deployment:

Tasks:

* Verified all functionality under different conditions (large datasets, various search terms).
* Reviewed error handling and prepared for deployment, including customization options.

Outcome:

* Finalized a resilient and extensible scraper tool.

Conclusion:

The Financial Image Web Scraper successfully achieved:

* Downloading categorized images using Bing Image Downloader.
* Fetching and downloading images from Cloudinary using its API.
* Reliable file downloads from URLs.
* Secure storage of financial images on Cloudinary.