# **YouTube Downloader - Technical Documentation**

**Author**: Tanmoy Bandyopadhyay  
**Version**: 2.0  
**Last Updated**: 18/5/2025

## **1. Overview**

A Python-based tool to download YouTube videos/audio concurrently with:

A Python-based tool to download YouTube videos/audio with:

* Threaded downloads for performance
* Automatic filename shortening
* Voice command support (planned)
* Modular architecture

## **2. Project Structure**

## Youtube\_Downloader/

## ├── .gitignore

## ├── main.py # Entry point

## ├── config.py # Constants

## ├── core/ # Main functionality

## │ ├── \_\_init\_\_.py

## │ ├── downloader.py # Video/audio download logic

## │ └── url\_resolver.py # YouTube search and URL handling

## └── utils/ # Helper modules

## ├── \_\_init\_\_.py

## ├── file\_processor.py # Filename operations

## ├── string\_utils.py # Text processing

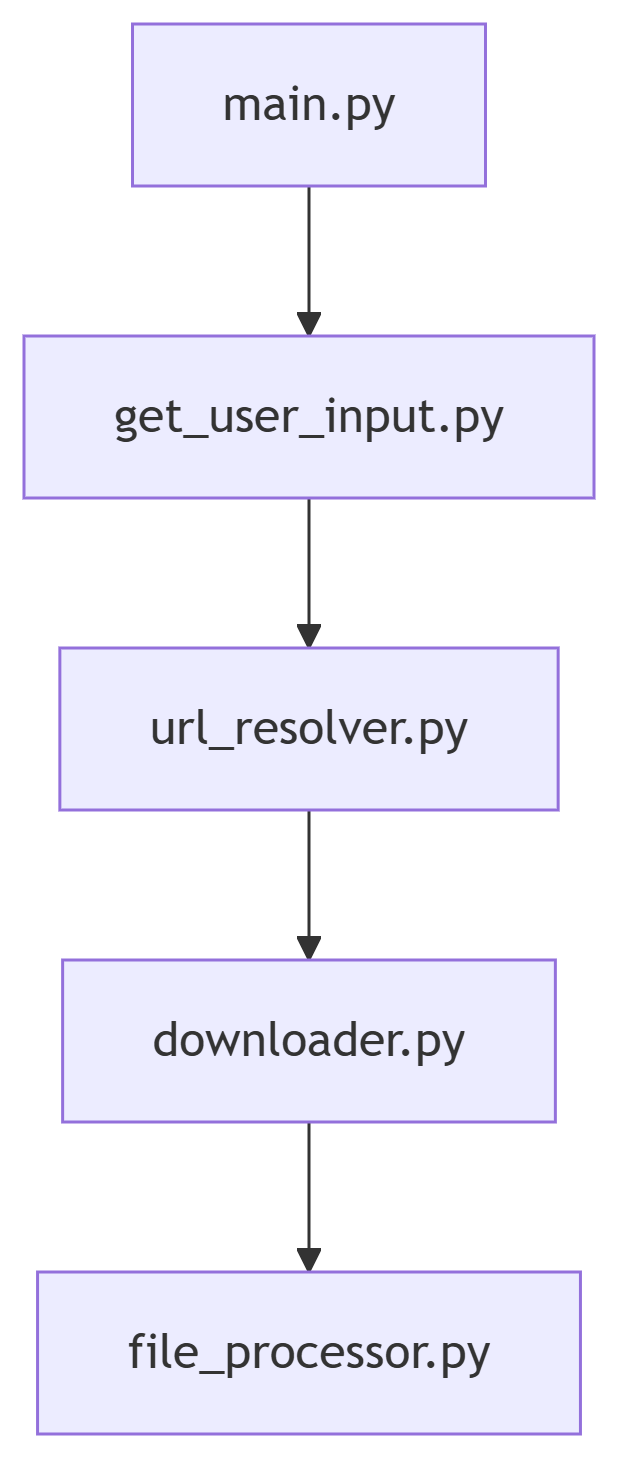
## ├── clean\_create\_download\_dir.py

## └── get\_user\_input.py # Input file parsing

## **3. Package Dependencies**

| **Package** | **Purpose** | **Installation** |
| --- | --- | --- |
| yt-dlp | YouTube content download | pip install yt-dlp |
| youtubesearchpython | URL resolution | pip install youtubesearchpython |
| aria2 | Accelerated downloads | scoop install aria2 (Windows) / brew install aria2 (Mac) |
| FFmpeg | Audio conversion | Download from [ffmpeg.org](https://ffmpeg.org/) |

## **4. Core Function Flow**



## **5. Key Functions**

### **5.1 File Processing**

| **File** | **Function** | **Description** |
| --- | --- | --- |
| **get\_user\_input.py** | **read\_nameof\_songs()** | **Parses input file with v/a prefixes** |
| **clean\_create\_download\_dir.py** | **cleanup\_create()** | **Ensures fresh download directory** |

### **5.2 URL Resolution**

| **File** | **Function** | **Concurrency** | **Description** |
| --- | --- | --- | --- |
| **url\_resolver.py** | **search\_youtube()** | **Threaded** | **Gets YouTube URL for a query** |
| **url\_resolver.py** | **get\_all\_youtube\_urls()** | **ThreadPoolExecutor** | **Batch-resolves URLs** |

### **5.3 Download Handlers**

| **File** | **Function** | **Type** | **Key Features** |
| --- | --- | --- | --- |
| **downloader.py** | **download\_youtube\_audio\_as\_mp3()** | **Audio** | **MP3 conversion via FFmpeg** |
| **downloader.py** | **download\_youtube\_video()** | **Video** | **Adaptive format fallback** |
| **downloader.py** | **download\_all\_ytube\_urls()** | **Threaded** | **Manages concurrent downloads** |

### **5.4 Post-Processing**

| **File** | **Function** | **Threaded** | **Description** |
| --- | --- | --- | --- |
| file\_processor.py | shorten\_downloaded\_file\_name() | Yes | Safe filename shortening |
| string\_utils.py | return\_short\_string() | No | Truncates strings to N words |

## **6. Concurrency Architecture**

# Example from downloader.py

with ThreadPoolExecutor() as executor:

futures = {executor.submit(downloader, url): query for url in url\_list}

for future in as\_completed(futures):

handle\_result(future)

**Threading Used In**:

1. URL resolution (get\_all\_youtube\_urls)
2. Batch downloads (download\_all\_ytube\_urls)
3. Filename shortening (shorten\_all\_downloaded\_file\_names)

## **7. Error Handling**

| **Error Type** | **Mitigation Strategy** | **Location** |
| --- | --- | --- |
| Format Unavailable | Multi-stage fallback | downloader.py |
| Network Timeouts | aria2c with 10 retries | downloader.py |
| File Collisions | Sequential numbering | file\_processor.py |
| Invalid Input | Skip malformed entries | get\_user\_input.py |

## **8. Voice Command Integration (Planned)**