TAP User Story

Size Estimation	
S	User Stories that are simple to implement and can be completed in 1-2 days.
M	User Stories that are moderate complex and can be completed in 3-5 days.
L	User Stories that are large and require approximately a week to accomplish.

MoSCoW Priority	
Must have	Non-negotiable product needs that are mandatory for the team.
Should have	Important initiatives that are not vital but add significant value.
Could have	Nice to have initiatives that will have a small impact if left out.
Will not have	Initiatives that are not a priority for this specific time-frame.

Epic ID	Epic	User Story ID	As a	I Want To	So That	Size Estimation	MoSCoW Priority	Justification
1	File Upload	1.1	User	Upload audio files	I can transcribe audio files	S	Must have	*Size Estimation:* The development effort is relatively small as it involves implementing standard file upload functionality. *MoSCoW Priority:* This is a fundamental feature for the system.
		1.2	User	Upload video files	I can transcribe video files	M	Could have	*Size Estimation:* Video file upload requires more complex handling compared to audio files such as large file processing, leading to a moderate development effort. *MoSCoW Priority:* This feature adds flexibility but is not critical for the initial release.
		1.3	User	Drag and drop files for upload	I can quickly and easily upload files	S	Could have	*Size Estimation:* It is straightforward to implement with existing web technologies, justifying the small size estimation. *MoSCoW Priority:* Adding drag-and-drop functionality improves usability
		1.4	User	Input my email address to receive the transcription result	I can get the results directly sent to my inbox	S	Must have	*Size Estimation:* The implementation is relatively simple, involving standard email APIs, hence the small size estimation. *MoSCoW Priority:* Email functionality is essential for user convenience, also reflect Client's requirement.
		1.5	User	Select a format for the output file	I can receive the transcription in my preferred format	S	Could have	*Size Estimation:* The small size estimation reflects the minimal additional logic required. *MoSCoW Priority:* Implementing multiple output formats is relatively simple but enhances user experience.
2	Transcription Processing	2.1	User	Handle multiple file transcriptions using a queue by the system	Multiple requests are processed eventually	M	Must have	*Size Estimation:* involves significant development effort but have less technical difficulty, hence the Medium size estimation. *MoSCoW Priority:* Implementing a queue system is critical for managing concurrent transcription requests.
		2.2	User	Use the up-to-date OpenAI Whisper model	I can receive the most accurate transcription result	L	Should have	*Size Estimation:* Integrating the latest model requires a large amount effort due to dependencies and testing, which justifies the Large size estimation. *MoSCoW Priority:* Latest model is critical to perform high transcription accuracy.
		2.3	User	Watch transcribed results in a conversation form	I can easily follow and understand the dialogue structure	M	Must have	*Size Estimation:* Displaying transcribed results in a conversation form involves moderate complexity, including formatting text to resemble a dialogue structure. *MoSCoW Priority:* This feature enhances readability and usability, making it an important to the system.
		2.4	User	Differentiate speakers labeled as "Speaker 1, 2, 3, etc."	I can easily identify who said what in the transcript	L	Must have	*Size Estimation:* Implementing speaker differentiation involves significant complexity, including advanced audio processing techniques to identify and separate individual speakers. This might require integrating machine learning models or algorithms capable of voice recognition. *MoSCoW Priority:* Speaker differentiation adds significant value, which makes the transcription result more clearly.
		2.5	User	Select different transcription language	I can transcribe audio in different languages depending on my region or the language of my research team	M	Could have	*Size Estimation:* Implementing this function require both frontend and backend expansion, as well as API enrichment. Requiring Medium development effort. *MoSCoW Priority:* This is a useful function but not required by the Client.
		2.6	User	Receive transcriptions in my preferred language (e.g. Chinese)	I can receive transcription services in my preferred language, enhancing accessibility and usability for a global audience	M	Could have	*Size Estimation:* Supporting multiple languages, particularly non-Latin scripts, involves moderate effort in handling and processing, justifying the medium size estimation. *MoSCoW Priority:* This is a useful function but not required by the Client.
		2.7	User	See a progress bar on the web page while my file is being processed	I can understand the process's status and estimate how long the processing will take	M	Should have	*Size Estimation:* This involves not only UI development but also backend changes to continuously communicate process status to the frontend. This task requires coordination between the frontend and backend systems to handle real-time data updates, making it moderately complex. *MoSCoW Priority:* Maintained as "Should have" because while this feature is not critical to the core functionality of the system (transcribing audio), it significantly enhances user experience by providing feedback on the processing status.

3	Transcription Result Forwarding	3.1	User	Receive transcription results through email	I can get the results conveniently	M	Must have	*Size Estimation:* Implementation involves standard functionality and available build-in function, hence the medium size estimation. *MoSCoW Priority:* Sending emails with results is essential for user convenience.
		3.2	User	Receive system error reports through email	I am informed if my transcription request fails	S	Should have	*Size Estimation:* This user story is estimated as size S because it involves straightforward logic, standard email functionality, minimal dependencies, and low user interface impact, making it simple and quick to implement. *MoSCoW Priority:* Implementing error reporting via email is straightforward and crucial for user transparency.
		3.3	User	Receive transcription in 'txt' format	I can use it with most text processing tools	M	Must have	*Size Estimation:* This includes handling different character encodings, ensuring proper line breaks, and managing edge cases like special characters or large files. Additionally, integrating this output format into the existing workflow requires testing and validation, justifying a medium size estimation. *MoSCoW Priority:* Supporting the 'txt' format is a basic requirement in this project, which meets the Client's need.
		3.4	User	Receive transcription in 'docx' format	I can use it in a professional document format	S	Should have	*Size Estimation:* Adding support for 'docx' format is a simple enhancement that does not require significant development effort, as the 'txt' format can be transfer to 'docx' format. *MoSCoW Priority:* This user story is prioritized as "Should have" because while it enhances the system's flexibility and aligns with professional standards, it is not critical for the core functionality and can be implemented after essential features are completed.
4	File Management	4.1	User	Save transcription result files in the local database automatically	I can access them later if needed	M	Should have	*Size Estimation:* Storing transcription results automatically involves moderate complexity, including database integration, ensuring data integrity, and handling different transcription scenarios. *MoSCoW Priority:* While important for data management and user access, this feature is not critical for the system's initial operation and can be prioritized after essential functionalities.
		4.2	User	Have my old files automatically cleaned up for 30 days	Storage is efficiently managed, and outdated files are removed	M	Should have	*Size Estimation:* Implementing an automated deletion policy involves moderate complexity, including setting up scheduled tasks or cron jobs, ensuring that files are properly tracked with their respective deletion timelines. *MoSCoW Priority:* Efficient storage management is important for maintaining system performance and preventing unnecessary data accumulation.
		4.3	User	Access an admin portal to see the history record	I can manage and review past transcriptions	L	Should have	*Size Estimation:* It involves significant complexity, including user interface design, secure access controls, and integration with the database, justifying a large size estimation. *MoSCoW Priority:* While valuable for managing and reviewing past transcriptions, it is not vital for the system's immediate operation, allowing it to be a secondary priority.
		4.4	User	Receive a link through email to enter the admin portal	I can securely access the portal	S	Should have	*Size Estimation:* Sending secure email links involves straightforward functionality, using established email protocols and basic authentication mechanisms, resulting in a small development effort. *MoSCoW Priority:* This feature enhances secure access to the admin portal, which is important but not critical for initial system deployment, making it a "Should have" feature.
		4.5	User	Download history transcription result files from the portal	I can access previous transcriptions if needed	S	Should have	*Size Estimation:* AImplementing download functionality for history transcription files is relatively simple, involving basic file handling and retrieval operations, leading to a small size estimation. *MoSCoW Priority:* While important for user convenience and data retrieval, this feature is not essential for the system's core operations, justifying its "Should have" priority.
		4.6	User	See the expiration date of my transcription files in the admin portal	I can know when my files will be automatically deleted	M	Should have	*Size Estimation:* Displaying the expiration date in the admin portal requires moderate complexity, including calculating the deletion date based on upload time and integrating this information into the user interface. *MoSCoW Priority:* While not critical for core functionality, this feature enhances user awareness and control over their stored files, making it a valuable addition to the system.