Final Year Project

Automatic Speech Recognition Graphical User Interface (ASR-GUI)

Web App

Project Plan

Student: Tan Liang Meng (U2220261A)

Supervisor: Dr Chng Eng Siong

Project Overview

ASR-GUI is a web-based platform that integrates an Automatic Speech Recognition (ASR) model developed by the NTU Speech & Language Lab. The system currently provides users with the ability to upload audio recordings and receive corresponding text transcripts.

While functional, the present implementation faces challenges such as inefficient audio handling, performance bottlenecks, UI limitations in hotword management, and recurring bugs that affect reliability. These hinder its usability for end users.

This project aims to enhance ASR-GUI by implementing new features, optimizing performance, fixing critical bugs, and improving Dockerization for stability in local environments. A stretch goal is to develop a standalone cross-platform desktop version using ElectronJS.

Project Objectives

The primary objective is to refine ASR-GUI into a stable, efficient, and user-friendly transcription tool. The goals are:

1. **Enhance usability** by extending hotword management and recording functions.

2. **Optimize performance** to reduce memory usage, improve audio playback, and ensure responsive WebSocket communication.

3. **Ensure reliability** through robust error handling and resolution of critical bugs.

4. **Streamline system workflows** by improving Dockerization for consistency and stability.

5. **Explore standalone deployment** via ElectronJS as a stretch goal.

Project Scope

New Feature Implementation

- **Hotword feature UI updates:**
    - Allow users to key in alias directly on the list.
    - Allow adding of multiple hotwords.
- **Recordings:**
    - Support playing of MP3 in addition to WAV format.
    - Enable bulk delete and transcription.

Optimization

- Resolve high memory usage on the tab.
- Fix unresponsive audio playback.
- Improve WebSocket communication for faster updates.
- Refactor Dockerization for smoother workflows.
- Add robust error-handling and failure feedback mechanisms.

Bug Fixes

- Fix issue where creating a new user forces auto logout.
- Ensure deleted jobs are properly cleared from the dashboard.
- Clear deleted recordings from database and Docker volume (preventing old transcripts from reappearing).
- Address issue where bringing down Docker Compose unexpectedly clears transcripts.
- Resolve other outstanding bugs affecting reliability.

Stretch Goal

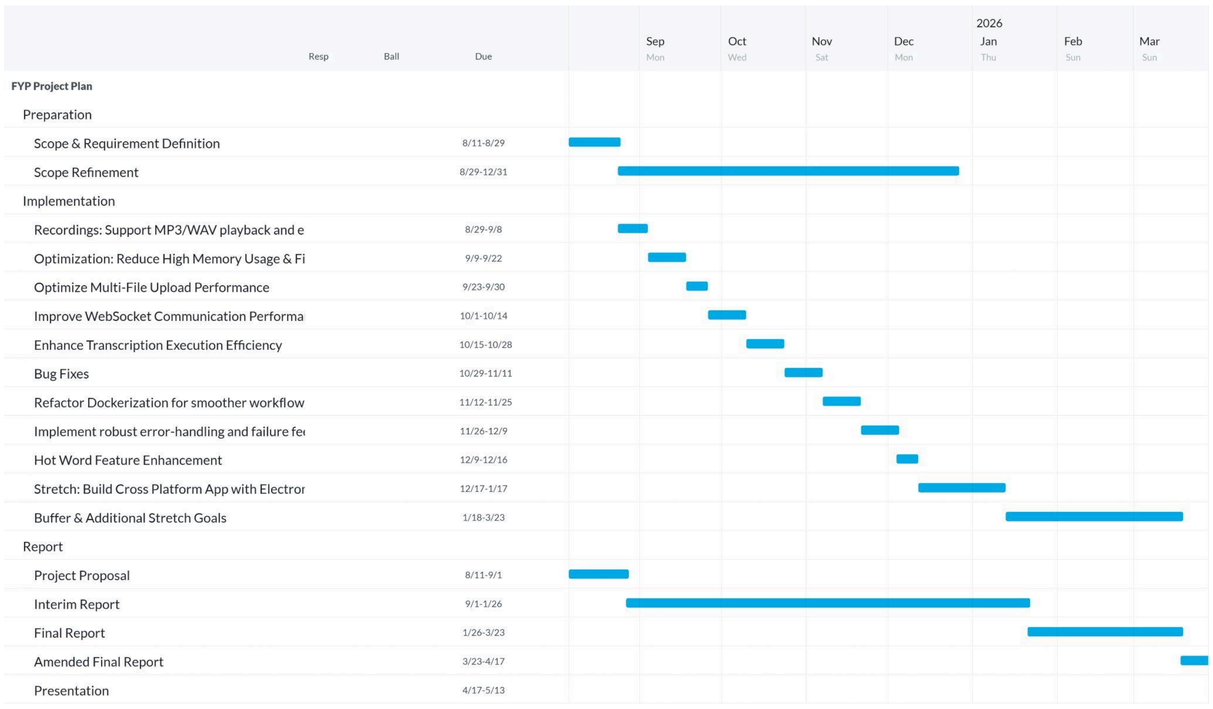
- Build a standalone cross-platform desktop app using **ElectronJS**.

Project Schedule

Index	Task Name	Start Date	End Date
Preparation			
1	Scope & Requirement Definition	**11 Aug 2025**	**29 Aug 2025**
2	Scope Refinement	**29 Aug 2025**	**31 Dec 2025**
Implementation			
3	Recordings: Support MP3/WAV playback and enable bulk delete & transcription	**29 Aug 2025**	**8 Sept 2025**
4	Optimization: Reduce High Memory Usage & Fix Unresponsive Audio Playback	**9 Sept 2025**	**22 Sept 2025**
6	Optimize Multi-File Upload Performance	**23 Sept 2025**	**30 Sept2025**
7	Improve WebSocket communication performance	**1 Oct 2025**	**14 Oct 2025**
8	Enhance Transcription Execution Efficiency	**15 Oct 2025**	**28 Oct 2025**
9	Bug Fixes	**29 Oct 2025**	**11 Nov 2025**
10	Refactor Dockerization for smoother workflows	**12 Nov 2025**	**25 Nov 2025**
11	Implement robust error-handling and failure feedback	**26 Nov 2025**	**9 Dec 2025**
12	Hot Word Feature Enhancement	**10 Dec 2025**	**16 Dec 2025**
13	Stretch: Build Cross Platform App with ElectronJs	**17 Dec 2025**	**17 Jan 2026**
14	Buffer & Additional Stretch Goals	**18 Jan 2026**	**23 Mar 2026**
Report			
15	Project Proposal	**11 Aug 2025**	**1 Sep 2025**
16	Interim Report	**2 Sep 2026**	**26 Jan 2026**
17	Final Report	**27 Jan 2026**	**23 Mar 2026**

18	Amended Final Report	**24 Mar 2026**	**17 Apr 2026**
19	Presentation	**18 Apr 2026**	**13 May 2026**

Gantt Chart



	Resp	Ball	Due
FYP Project Plan			
Preparation			
Scope & Requirement Definition			8/11-8/29
Scope Refinement			8/29-12/31
Implementation			
Recordings: Support MP3/WAV playback and e			8/29-9/8
Optimization: Reduce High Memory Usage & Fi			9/9-9/22
Optimize Multi-File Upload Performance			9/23-9/30
Improve WebSocket Communication Performa			10/1-10/14
Enhance Transcription Execution Efficiency			10/15-10/28
Bug Fixes			10/29-11/11
Refactor Dockerization for smoother workflow			11/12-11/25
Implement robust error-handling and failure fe			11/26-12/9
Hot Word Feature Enhancement			12/9-12/16
Stretch: Build Cross Platform App with Electror			12/17-1/17
Buffer & Additional Stretch Goals			1/18-3/23
Report			
Project Proposal			8/11-9/1
Interim Report			9/1-1/26
Final Report			1/26-3/23
Amended Final Report			3/23-4/17
Presentation			4/17-5/13