1. Title of the Case Study Voice-to-Text Meeting Minutes Generator

2. Team Details

Member Name	Modules Allocated	Employee Id
	Audio Upload & S3 Integration	
	AWS Transcribe Setup	
	Transcript Post-Processing	
	Meeting Summary Generation	
	Dashboard & Downloadable Reports	

3. Problem Statement Documenting minutes from recorded meetings is a tedious and time-consuming task. This project aims to build a system that automates the generation of accurate meeting transcripts and summaries from uploaded audio recordings. When an audio file is uploaded to an S3 bucket, AWS Transcribe is used to convert speech to text. A Lambda function processes the output, structures it into readable form, and generates downloadable meeting summaries.

Modules and Functionalities:

Module 1: Audio Upload & S3 Integration (Python + AWS S3)

- Develop a frontend or CLI to upload `.mp3` or `.mp4` files
- Store uploaded files in a dedicated S3 bucket
- Validate file format and naming convention
- Store file metadata (name, uploader, timestamp)
- Notify user of upload status

Module 2: AWS Transcribe Setup (Python + AWS Transcribe + Lambda)

- Trigger AWS Transcribe Job on S3 upload event
- Configure Transcribe for supported language and format
- Monitor Transcribe job status using Lambda
- Store transcript output in another S3 bucket
- Handle failed or incomplete jobs

Module 3: Transcript Post-Processing (Python)

- Clean up filler words and transcription noise
- Convert transcript into paragraph or bullet format
- Extract speaker labels and timestamps
- Detect and fix transcription errors (simple rules)
- Store clean transcript in structured format (JSON, TXT)

Module 4: Meeting Summary Generation (Python + NLP)

- Use text summarization (TextRank, spaCy, or GPT API)
- Generate concise summary from full transcript
- Highlight action items and decisions
- Add metadata: date, duration, speakers
- Store summary in S3 (PDF or text)

Module 5: Dashboard & Downloadable Reports (Python + Flask/Streamlit)

- Display recent transcripts and summaries
- Search/filter by meeting date, keyword, or speaker
- Provide download options (TXT, PDF)
- View speaker analytics (talk time, keywords)
- Secure dashboard access with user login (optional)

4. Tools, Technologies & Methodologies Used

- **Programming Language**: Python
- **AWS Services**: S3, Lambda, Transcribe
- **Libraries**: Boto3 SDK, spaCy, NLTK, Flask/Streamlit
- **Others**: JSON, TXT, PDF generation libraries
- **Methodology**: Agile (Kanban or Scrum)

6. Attachments

- Architecture Diagram:
- S3 Upload Bucket → AWS Transcribe → Lambda Processing → Cleaned Transcript/Report Bucket
- Sample Summary Output:

```
{
  "meeting_id": "M2025-001",
  "summary": "Discussed Q2 project deadlines, assigned tasks to team leads, and finalized budget approvals.",
  "action_items": ["Submit design documents by June 25", "Start development by July 1"],
  "duration_minutes": 45,
  "participants": ["Alice", "Bob", "Priya"]
}
```

7. Milestone and Task

Milestone	Task	Deadline
M1: Project Setup	Create S3 buckets, IAM roles, Transcribe configuration	Week 1
M2: Upload & Trigger	Audio upload and S3-Lambda-Transcribe workflow	Week 2

Milestone	Task	Deadline
M3: Transcript Processing	Clean and format transcript, store outputs	Week 3
M4: Summary Generator	Implement NLP summarizer and store summary files	Week 4
M5: Dashboard Reporting	Create dashboard for viewing/downloading transcripts	Week 5