SOFTWARE PROJECT MANAGEMENT

LAB FILE

AMITY SCHOOL OF ENGINEERING AND TECHNOLOGY

AMITY UNIVERSITY

UTTAR PRADESH

Submitted by-

Raj Krishna Choudhary

7 CSE 9X

A2305221548

Submitted to-

Dr. Vasudha Vashisht

SOFTWARE PROJECT MANAGEMENT

LAB FILE

AMITY SCHOOL OF ENGINEERING AND TECHNOLOGY

AMITY UNIVERSITY

UTTAR PRADESH

Submitted by-	**Submitted to-**
Rishabh Negi	Dr. Vasudha Vashisht
7 CSE 9X	
A2305221530	

SOFTWARE PROJECT MANAGEMENT

LAB FILE

AMITY SCHOOL OF ENGINEERING AND TECHNOLOGY

AMITY UNIVERSITY

UTTAR PRADESH

Submitted by-	**Submitted to-**
Shresth Rana	Dr. Vasudha Vashisht
7 CSE 9X	
A2305221541	

SOFTWARE PROJECT MANAGEMENT

LAB FILE

AMITY SCHOOL OF ENGINEERING AND TECHNOLOGY

AMITY UNIVERSITY

UTTAR PRADESH

Submitted by-	**Submitted to-**
Arnab Bera	Dr. Vasudha Vashisht
7 CSE 9X	
A2305221567	

Index

S.No.	**Practical**	**Allotment date**	**Submission Date**	**Remarks**
1.	ER Diagram			
2.	Gantt Chart			
3.	SRS Document			
4.	Step-wise Business Project Planning			
5.	CPM (Critical Path Method)			
6.	Risk Analysis of Minor Project			
7.	Resource Allocation Graph with Task Dependency and Time Estimation			
8.	Visualizing Progress: Timeline Chart			
9.	PERT Chart			
10.	Project Closure Report			

PRACTICAL-1

AIM: Entity-Relationship (ER) Diagram for Maithili Speech Recognition System

THEORY: An Entity-Relationship (ER) Diagram is a vital tool in software project management used to visually represent the data structure and its relationships within the system. In the case of the Maithili Speech Recognition System, the ER diagram helps in identifying key entities like Audio Sample, Mel-Spectrogram, Transcript, Model, and Prediction Output. Each of these entities holds critical attributes such as speaker ID, audio format, tokenized text, and model configurations. The relationships among these entities are defined by actions such as generating spectrograms from audio, mapping transcripts to audio, and transcribing through the Whisper model. By organizing these elements into an ER diagram, it aids in the clear conceptualization of data flow, making database and system architecture design more efficient. This visualization simplifies how data is stored, processed, and retrieved, which is essential for smooth execution of the model's training and testing stages.

Entities and Attributes

Audio Data

- `audio_id`
- `file_path`
- `duration`
- `speaker_info`
- `language`

Text Data

- `text_id`
- `transcript`
- `language`
- `tokens`

Preprocessing

- `preprocessing_id`
- `noise_reduction`
- `segmentation_details`
- `feature_extraction_method`
- `tokenization`

Model

- `model_id`
- `architecture`
- `version`
- `training_params`
- `accuracy`

Training Process

- `training_id`
- `training_loss`
- `validation_loss`
- `epochs`
- `batch_size`

Dataset

- `dataset_id`
- `data_source`
- `total_samples`
- `split_ratio`

Users

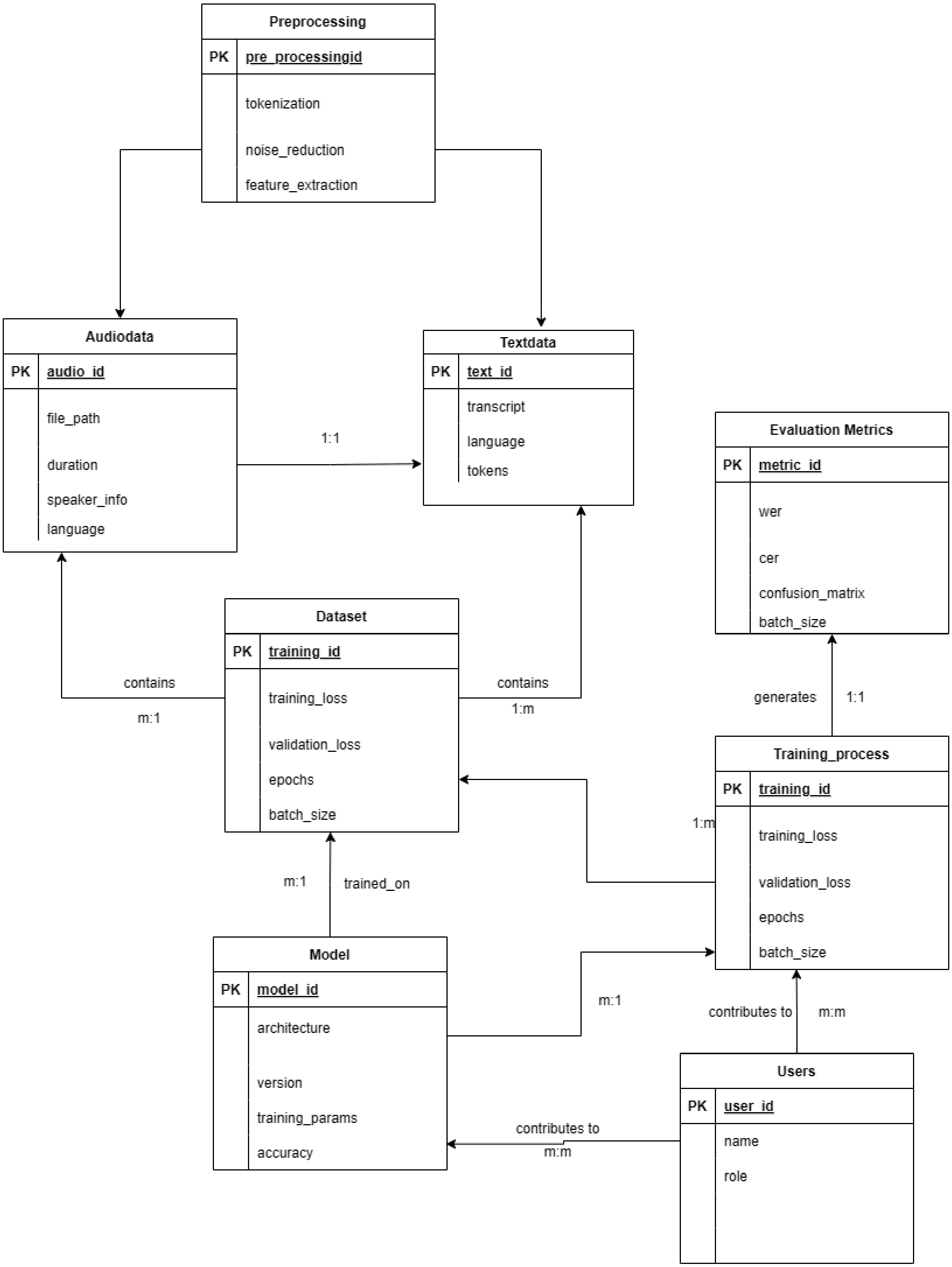
- `user_id`
- `name`
- `role`

Evaluation Metrics

- `metric_id`
- `wer`
- `cer`
- `confusion_matrix`
- `accuracy`

Relationships

- **Audio Data** 1-to-1 with **Text Data** (Each audio corresponds to one text transcription)
- **Preprocessing** applies to **Audio Data** and **Text Data**
- **Dataset** contains many **Audio Data** and **Text Data** (1-to-Many)
- **Model** is trained on one or more **Dataset** entries (Many-to-One)
- **Training Process** refers to one **Model** and one **Dataset** (Many-to-One)
- **Users** contribute to various **Models** and **Training Processes** (Many-to-Many)
- **Evaluation Metrics** are generated for each **Training Process** (One-to-One)


Preprocessing
PK pre_processingid
tokenization
noise_reduction
feature_extraction
Audiodata
PK audio_id
file_path
duration
speaker_info
language
1:1
Textdata
PK text_id
transcript
language
tokens
Evaluation Metrics
PK metric_id
wer
cer
confusion_matrix
batch_size
Dataset
PK training_id
training_loss
validation_loss
epochs
batch_size
contains
m:1
contains
1:m
generates
1:1
Training_process
PK training_id
training_loss
validation_loss
epochs
batch_size
1:m
m:1
trained_on
Model
PK model_id
architecture
version
training_params
accuracy
m:1
contributes to
m:m
Users
PK user_id
name
role
contributes to
m:m

Result: The diagram for the minor project has been successfully completed.

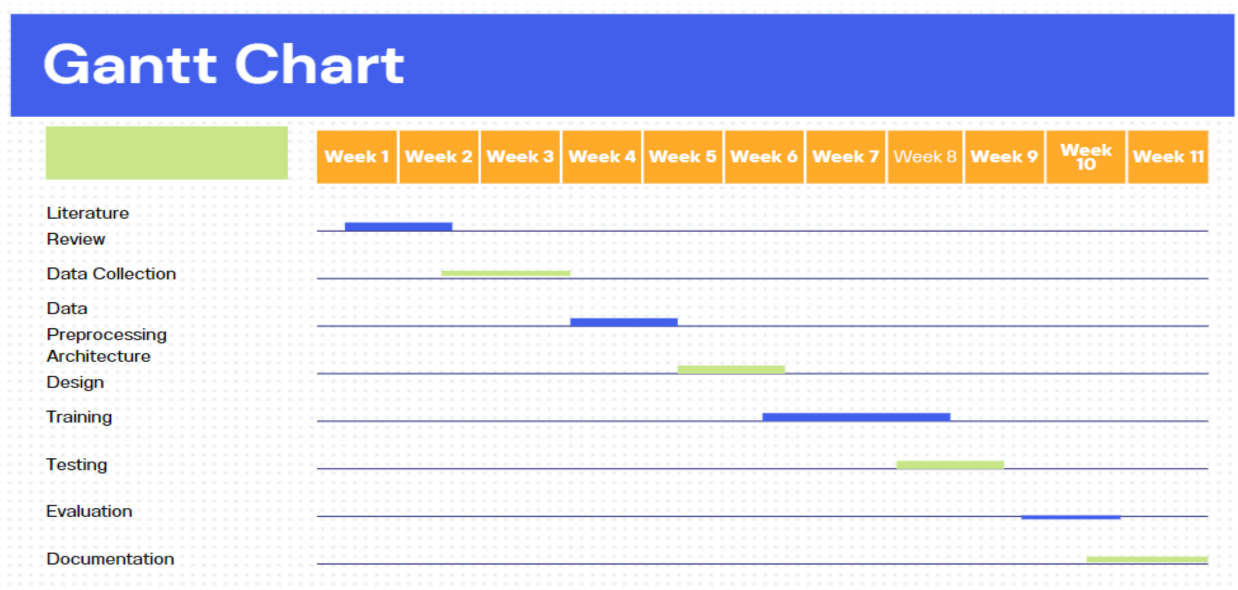
Conclusion: The ER diagram effectively models the system's entities, relationships, and data flow, providing a clear structure for the project's database design. It ensures data integrity and helps optimize operations by organizing relationships efficiently. This diagram forms the foundation for the project's implementation and further development phases.

Criteria	**Total marks**	**Marks Obtained**	**Comments**
Concept (A)	2		
Implementation (B)	2		
Performance (C)	2		
Total	6		

PRACTICAL-2

AIM:Gantt Chart for Maithili Speech Recognition System

THEORY: A Gantt Chart is a project management tool that illustrates the timeline of tasks and their dependencies, ensuring that the project proceeds efficiently. In the Maithili Speech Recognition System, the Gantt chart organizes key project phases such as Literature Review, Data Preprocessing, Model Design, Tokenization, Model Training, Testing & Validation, and Final Report & Presentation. Each phase is plotted on a timeline, showing when it starts and ends, and how it relates to other tasks. The Gantt chart also allows project managers to visualize task overlaps and track the project's overall progress. By monitoring each phase, potential delays can be identified early, ensuring that all tasks are completed within the planned time frame.



Result: The diagram for the Minor project has been successfully made.

Conclusion: We created a Gantt chart to outline the project timeline for the Maithili Speech Recognition System. It detailed phases like data collection, preprocessing, model design, and evaluation, ensuring clear task sequencing and effective project management.

Criteria	**Total marks**	**Marks Obtained**	**Comments**
Concept (A)	2		
Implementation (B)	2		
Performance (C)	2		
Total	6		

PRACTICAL-3

AIM: Software Requirements Specification (SRS) for Maithili Speech Recognition System

THEORY: The Software Requirements Specification (SRS) document outlines the functional and non-functional requirements of the Maithili Speech Recognition System, serving as a blueprint for development. The SRS defines the Purpose of the project: building an ASR system for Maithili using OpenAI’s Whisper. The Scope covers the system’s functionality to process audio inputs, convert them into Mel-spectrograms, and generate accurate textual transcripts in Maithili. Key Functional Requirements include handling audio data, processing it through the Whisper model, and delivering text output, while Non-Functional Requirements focus on performance, scalability, and usability. For instance, the system should achieve high accuracy (low WER) and support real-time transcription. By defining these requirements, the SRS ensures that all stakeholders have a shared understanding of what the system will achieve, guiding the development process from design to deployment.

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to provide a comprehensive and detailed description of the requirements for the design and development of a Maithili Speech Recognition System using OpenAI's Whisper model. This document aims to ensure that all stakeholders, including the development team, project sponsors, end users, and academic advisors, have a clear and unified understanding of the project’s objectives, scope, and deliverables. It will serve as a reference throughout the development process, ensuring that the final product meets the specified requirements and expectations. The document focuses on the minor project phase, which lays the groundwork for further development in the major phase.

1.2 Scope

The Maithili Speech Recognition System project aims to develop an application that can accurately convert spoken Maithili into written text. The system will leverage OpenAI's Whisper model, a state-of-the-art neural network for automatic speech recognition (ASR). This SRS document focuses on the minor phase of the project, which includes:

1. **Data Collection**: Gathering a diverse and representative dataset of spoken Maithili.
2. **Data Annotation**: Transcribing the collected audio data to create a labeled dataset.

3. **Initial Model Training**: Training the Whisper model on the annotated dataset.
4. **Preliminary Evaluation**: Assessing the model's performance and identifying areas for improvement.

The major phase, which is beyond the scope of this document, will focus on further model refinement, integration, and deployment.

1.3 Overview

The Maithili Speech Recognition System will be implemented in two phases: minor and major. This document addresses the minor phase, which includes foundational activities crucial for the system's development. The system will utilize OpenAI's Whisper model due to its advanced capabilities in speech recognition.

Key Activities in the Minor Phase:

1. **Data Collection**:
    - Objective: To collect a diverse set of Maithili speech data.
    - Approach: Recording speech from various sources, including native speakers and public broadcasts.
    - Tools: Audio recording equipment, online platforms for data collection.
2. **Data Annotation**:
    - Objective: To transcribe collected audio data accurately.
    - Approach: Manual transcription by native speakers and automated transcription tools.
    - Tools: Annotation software, transcription services.
3. **Initial Model Training**:
    - Objective: To train the Whisper model on the annotated dataset.
    - Approach: Preprocessing data, training the model, and tuning hyperparameters.
    - Tools: Whisper framework, computing resources (GPUs/TPUs).
4. **Preliminary Evaluation**:
    - Objective: To evaluate the model's performance.
    - Approach: Using standard metrics such as Word Error Rate (WER) and conducting qualitative assessments.
    - Tools: Evaluation scripts, performance dashboards.

1.4 Objectives

The primary objectives of the minor project phase are:

- **Data Collection and Annotation**: To gather and transcribe a comprehensive dataset of Maithili speech.
- **Initial Model Training**: To develop an initial Maithili ASR model using OpenAI's Whisper.
- **Performance Evaluation**: To conduct preliminary evaluations and identify areas for improvement.
- **Collaboration Framework**: To establish a collaborative framework for continuous development and enhancement.

1.5 Intended Audience

This SRS document is intended for the following stakeholders:

- **Development Team**: To understand the detailed requirements and guide the implementation process.
- **Project Sponsors**: To review the project scope, objectives, and deliverables.
- **End Users**: To gain insights into the system's capabilities and provide feedback.
- **Academic Advisors**: To provide feedback and ensure the project adheres to academic standards.

1.6 Definitions and Acronyms

- **ASR**: Automatic Speech Recognition, the technology that converts spoken language into written text.
- **Whisper**: A neural network model developed by OpenAI for speech recognition.
- **SRS**: Software Requirements Specification, a document that describes the requirements for a software system.
- **Maithili**: A language spoken in the eastern regions of India and the southeastern regions of Nepal.

1.7 References

- OpenAI Whisper Documentation
- Relevant research papers on speech recognition and Maithili language processing
- Previous SRS documents and project reports for reference

2. Overall Description

2.1 Product Perspective

The Maithili Speech Recognition System will be developed as a standalone application. It will consist of several components, including data collection modules, a training module, and an evaluation module. The system will be designed to integrate seamlessly with existing tools and platforms used for speech recognition and natural language processing.

2.2 Product Functions

The primary functions of the Maithili Speech Recognition System include:

1. **Data Collection**: Collecting audio data from various sources to ensure a diverse and representative dataset.
2. **Data Annotation**: Transcribing the collected audio data to create a labeled dataset for training the model.
3. **Model Training**: Using the annotated dataset to train the Whisper model for Maithili speech recognition.
4. **Performance Evaluation**: Assessing the model's accuracy and identifying areas for improvement.

2.3 User Characteristics

The primary users of the Maithili Speech Recognition System include:

- **Researchers**: Researchers working in the field of speech recognition and natural language processing.
- **Developers**: Software developers integrating the ASR system into other applications.
- **End Users**: Native Maithili speakers using the system for transcription and other applications.

2.4 Constraints

- **Data Quality**: Ensuring high-quality and diverse data collection for accurate model training.
- **Resource Availability**: Availability of computational resources for training the Whisper model.
- **Time Constraints**: Adhering to the project timeline, particularly for the minor phase.

2.5 Assumptions and Dependencies

- **Access to Data**: Assumption that sufficient Maithili speech data will be available for collection.
- **Model Compatibility**: Assumption that the Whisper model will be suitable for Maithili speech recognition without significant modifications.
- **Technical Expertise**: Availability of technical expertise for data annotation, model training, and evaluation.

Result: The diagram for the Minor project has been successfully made.
Conclusion: The SRS document laid out all system requirements in a detailed manner, helping ensure that both functional and non-functional aspects were addressed for the project, guiding the development and design phases.

Criteria	Total marks	Marks Obtained	Comments
Concept (A)	2		
Implementation (B)	2		
Performance (C)	2		
Total	6		

PRACTICAL-4
AIM: Step Wise Business Project Planning
THEORY: Step-Wise Business Project Planning ensures that the project progresses in a structured and organized manner. For the Maithili Speech Recognition System, the plan begins with Project Conceptualization, defining the objective to create a Maithili ASR system using Whisper. Next is Requirements Gathering, identifying functional needs like audio input and text output, and non-functional aspects like real-time performance. Feasibility Analysis evaluates whether the project is technically and economically viable, while Project Scheduling uses tools like the Gantt chart to organize tasks and set milestones. Resource Allocation distributes tasks among the team members based on their expertise, ensuring optimal use of available resources. Risk Management identifies potential issues such as data limitations or model underperformance, with strategies to mitigate them. The final steps include Execution and Monitoring where tasks are carried out, tracked, and adjusted as necessary, and Project Review and Delivery, where the final system is evaluated and handed over. This systematic approach ensures the project is delivered on time, meeting all objectives efficiently.

1. Project Conceptualization

Objective: The project aims to develop an Automatic Speech Recognition (ASR) system for the Maithili language using OpenAI's Whisper model architecture.

Key Points:

- **Problem Definition**: Limited resources for Maithili language transcription.
- **Goal**: Create a speech-to-text system that accurately converts spoken Maithili into text.
- **Scope**: Focus on building a machine learning model that can transcribe Maithili audio, create a tokenizer specific to Maithili, and handle diverse dialects and speech variations.

2. Requirements Gathering

Functional Requirements:

- Input: Maithili audio in .wav format.
- Output: Text transcription of the Maithili audio.
- Integration with a custom Byte-Level BPE tokenizer for Maithili.
- Ability to handle different dialects, speech patterns, and accents in Maithili.

Non-Functional Requirements:

- **Accuracy**: The model should have a low Word Error Rate (WER) and Character Error Rate (CER).
- **Performance**: The system should provide real-time or near real-time transcription.
- **Scalability**: Capable of handling large datasets and potentially multiple users.
- **User-Friendly Interface**: The system should have an intuitive interface for uploading audio and receiving transcriptions.

3. Feasibility Analysis

Technical Feasibility:

- The use of Whisper model architecture for speech-to-text conversion provides a proven foundation for ASR systems.
- Libraries like `librosa` for audio processing and `tokenizers` for creating a Byte-Level BPE tokenizer make technical implementation feasible.

Economic Feasibility:

- Requires cloud storage and computation resources, but the development can be done within the budget using free or affordable cloud resources (e.g., Kaggle, Colab).

4. Project Scheduling

Tools: Gantt charts, project management software like Trello, Asana, or Microsoft Project.

Phases:

- **Phase 1**: Data Collection and Preprocessing (Week 1-2)
- **Phase 2**: Model Design and Development (Week 3-4)
- **Phase 3**: Training and Hyperparameter Tuning (Week 5-6)
- **Phase 4**: Evaluation and Testing (Week 7)
- **Phase 5**: Integration and Deployment (Week 8)

Milestones:

- Completion of dataset collection and tokenizer creation.
- Model training completion with acceptable accuracy (WER < 15%).
- Final system deployment.

5. Resource Allocation

Team Members:

- **Shresth Rana** (Data Collection and Literature Review)
- **Rishabh Negi** (Model Design and Development)
- **Raj Krishna Choudhary** (Data Preprocessing)
- **Arnab Bera** (Training and Hyperparameter Tuning)

Tasks:

- **Shresth**: Collection of diverse Maithili audio samples and documentation of existing solutions.
- **Rishabh**: Implementing Whisper-based architecture and custom tokenizer.
- **Raj**: Preprocessing audio and text data using `librosa` and tokenizing transcripts.
- **Arnab**: Training the model and adjusting hyperparameters for optimal performance.

6. Risk Management

Potential Risks:

- **Data Limitations**: Lack of high-quality or diverse audio data in Maithili.
    - **Mitigation**: Use data augmentation techniques to diversify the dataset.
- **Model Underperformance**: High WER or poor performance across dialects.
    - **Mitigation**: Fine-tune the model with additional data, increase model complexity, or employ transfer learning.
- **Resource Constraints**: Limited cloud resources or processing power.
    - **Mitigation**: Use cloud platforms like Colab for free GPU access or optimize code for memory-efficient execution.
- **Unforeseen Delays**: Team availability or unforeseen technical issues.
    - **Mitigation**: Use buffer periods between phases in the project schedule to accommodate potential delays.

7. Execution and Monitoring

- **Execution**:
    - Audio data collection and preprocessing completed in the first two weeks.
    - Whisper model architecture implemented in week 3 with Byte-Level BPE tokenizer.
    - Model trained on dataset, validated, and tested using evaluation metrics.

- **Monitoring**:
    - Weekly meetings to track progress.
    - Key metrics like training loss, WER, and CER monitored to assess model performance.
    - Gantt chart updated regularly to reflect progress and adjustments.

8. Project Review and Delivery

- **Final Review**: Before delivery, the model's performance is reviewed, and necessary adjustments are made.
- **System Testing**: System tested across various Maithili dialects to ensure robustness.
- **Delivery**: The final product, along with documentation, is handed over to stakeholders or deployed as a web-based tool for users to transcribe Maithili audio.
- **Post-Project Review**: Document lessons learned and improvements for future speech recognition projects.

Result: The diagram for the Minor project has been successfully made.
Conclusion: The business plan broke down the project's development into specific, actionable steps, aligning both technical progress and market strategies to meet the overall goals efficiently.

Criteria	Total marks	Marks Obtained	Comments
Concept (A)	2		
Implementation (B)	2		
Performance (C)	2		
Total	6		

PRACTICAL-5

AIM: CPM (Critical Path Method)

Problem –

A small project consisting of eight activities has the following characteristics:

Time – Estimates (in weeks)

Activity	*Preceding activity*	*Most optimistic time (a)*	*Most likely time (m)*	*Most Pessimestic time (b)*
A	None	2	4	12
B	None	10	12	26
C	A	8	9	10
D	A	10	15	20
E	A	7	7.5	11
F	B,C	9	9	9
G	D	3	3.5	7
H	E, F, G	5	5	5

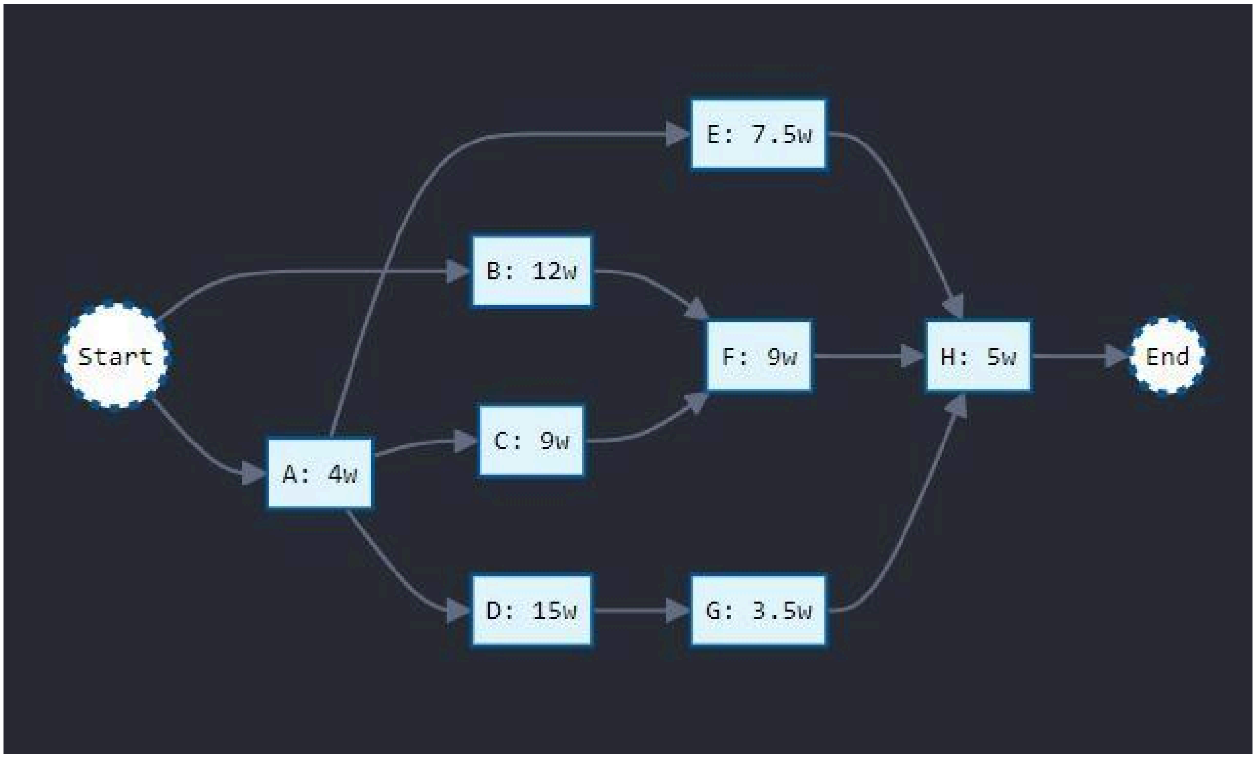
Theory-

The critical path is a sequence of tasks in a project that must be completed on time to finish the project. The critical path is also known as the critical path method (CPM) or critical path analysis (CPA).

The critical path is made up of tasks that are directly linked and affect the project's finish date. If any task on the critical path is delayed, the entire project is delayed. The critical path method is a project management technique that helps identify the critical path and plan and schedule projects

Diagram-

Pert chart of the above problem statement



Calculation of Critical Path of the above problem statement:-

Step 1: Forward Pass (Calculating Early Start and Early Finish)

We calculate the **Early Start (ES)** and **Early Finish (EF)** times for each activity by moving from the start to the end of the project.

- **A**: Since A has no preceding activity:
  - ESA= 0
  - ESA+TEA=0+5=5
- **B**: Since B has no preceding activity:
  - ESB= 0
  - EFB=ESB+TEB=0+14=14
- **C**: C depends on A:
  - ESC=EFA= 5

- $EFC = ESC + TEC = 5 + 9 = 14$

- **D**: D depends on A:
  - $ESD = EFA = 5$
  - $EFD = ESD + TED = 5 + 15 = 20$
- **E**: E depends on A and B:
  - $ESE = \max(EFA, EFB) = \max(5, 14) = 14$
  - $EFE = ESE + TEE = 14 + 8 = 22$
- **F**: F depends on B and C:
  - $ESF = \max(EFB, EFC) = \max(14, 14) = 14$
  - $EFF = ESF + TEF = 14 + 8.33 = 22.33$
- **G**: G depends on D:
  - $ESG = EFD = 20$
  - $EFG = ESG + TEG = 20 + 3.5 = 23.5$
- **H**: H depends on E, F, and G:
  - $ESH = \max(EFE, EFF, EFG) = \max(22, 22.33, 23.5) = 23.5$
  - $EFH = ESH + TEH = 23.5 + 5 = 28.5$

Step 2: Backward Pass (Calculating Late Start and Late Finish)

Next, we perform a backward pass to calculate **Late Finish (LF)** and **Late Start (LS)** times for each activity by moving from the end of the project back to the start.

- **H**: Since H is the last activity:
  - $LFH = EFH = 28.5$
  - $LSH = LFH - TEH = 28.5 - 5 = 23.5$
- **E**: E depends on H:
  - $LFE = LSH = 23.5$
  - $LSE = LFE - TEE = 23.5 - 8 = 15.5$
- **F**: F depends on H:
  - $LFF = LSH = 23.5$

- $LSF = LFF - TEF = 23.5 - 8.33 = 15.17$

- **G**: G depends on H:
  - $LFG = LSH = 23.5$
  - $LSG = LFG - TEG = 23.5 - 3.5 = 20$
- **D**: D depends on G:
  - $LFD = LSG = 20$
  - $LSD = LFD - TED = 20 - 15 = 5$
- **C**: C depends on F:
  - $LFC = LSF = 15.17$
  - $LSC = LFC - TEC = 15.17 - 9 = 6.17$
- **B**: B depends on E and F:
  - $LFB = \min(LSE, LSF) = \min(15.5, 15.17) = 15.17$
  - $LSB = LFB - TEB = 15.17 - 14 = 1.17$
- **A**: A depends on C, D, and E:
  - $LFA = \min(LSC, LSD, LSE) = \min(6.17, 5, 15.5) = 5$
  - $LSA = LFA - TEA = 5 - 5 = 0$

Step 3: Calculate Slack and Determine Critical Path

Slack is calculated as:

$$Slack = LS - ES \, or \, LF - EF$$

- **A**: $SlackA = 0 - 0 = 0$ (Critical)
- **B**: $SlackB = 1.17 - 0 = 1.17$
- **C**: $SlackC = 6.17 - 5 = 1.17$
- **D**: $SlackD = 5 - 5 = 0$ (Critical)
- **E**: $SlackE = 15.5 - 14 = 1.5$
- **F**: $SlackF = 15.17 - 14 = 1.17$
- **G**: $SlackG = 20 - 20 = 0$ (Critical)
- **H**: $SlackH = 23.5 - 23.5 = 0$ (Critical)

Critical Path:

The activities with zero slack form the critical path, which is:

A → D → G → H

The critical path takes **28.5 weeks** to complete the project.

Result: The practical has been successfully conducted.

Conclusion: The CPM analysis identified the most important tasks and their dependencies, helping prioritize key actions that would directly impact the project timeline and ensuring timely completion of the project.

Criteria	Total marks	Marks Obtained	Comments
Concept (A)	2		
Implementation (B)	2		
Performance (C)	2		
Total	6		

PRACTICAL-6

AIM: Risk Analysis of Minor Project

Sr. No.	Risk Name	Frequency	Impact	Assigned To	Plans to Resolve
1	Insufficient Data for Training	High	High	Raj	Collect a larger dataset, use data augmentation techniques.
2	Model Overfitting	Medium	Medium	Rishabh	Employ cross-validation, regularization, and hyperparameter tuning..
3	Poor Model Covergence	Medium	High	Shresth	Use learning rate schedulers, experiment with optimizers, and adjust architecture.
4	Difficult in Model Deployment	Medium	High	Arnab	Utilize cloud platforms, develop APIs, and optimize for inference speed.
5	Team Communication and Coordination Issues	High	High	Raj	Implement regular meetings, project management tools, and clear documentation.
6	Delays in Data Collection and Preprocessing	Medium	High	Rishabh	Start data collection early, automate preprocessing, and have backup datasets
7	Limited Access to Computing	Low	High	Arnab	Reserve HPC resources early and utilize cloud-based services

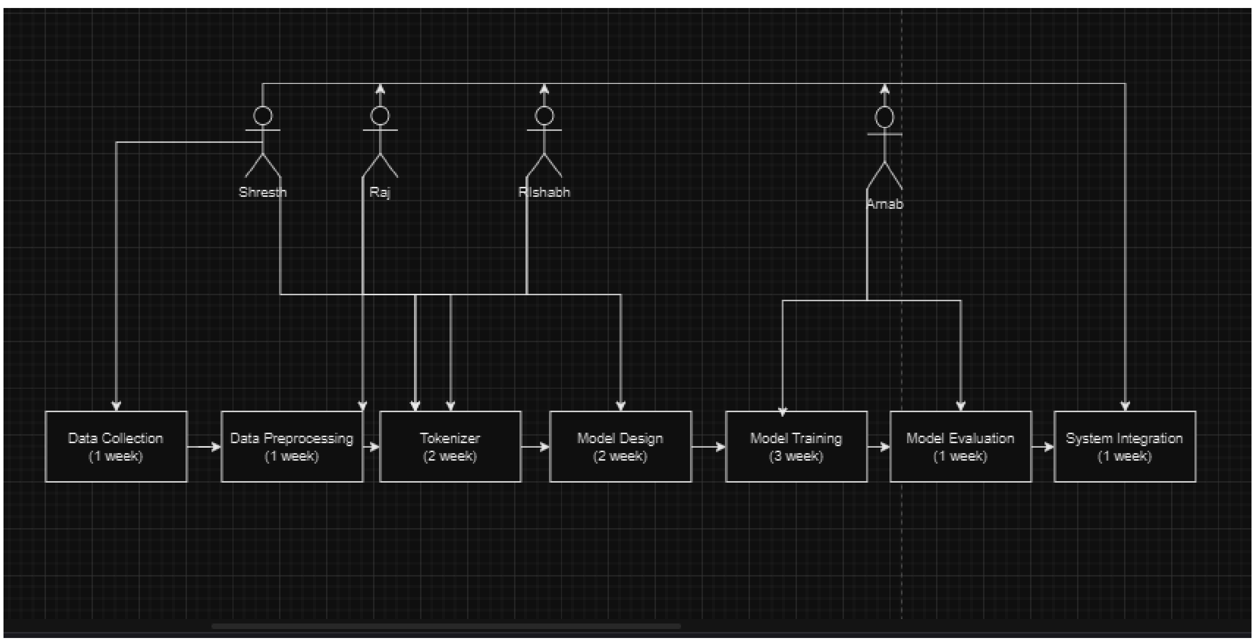
Result: The practical has been successfully conducted.
Conclusion: Risk analysis identified potential challenges, such as data quality or model underperformance. Mitigation plans were developed to address these risks, ensuring project stability and minimizing disruptions.

Criteria	Total marks	Marks Obtained	Comments
Concept (A)	2		
Implementation (B)	2		
Performance (C)	2		
Total	6		

PRACTICAL-7

AIM: Resource Allocation graph with dependency of task and time estimation

THEORY: The aim of this practical is to create a Resource Allocation Graph that illustrates task dependencies and time estimations for the Maithili Speech Recognition System project. This graph visually represents which resources (team members, tools, etc.) are allocated to specific tasks and how these tasks depend on each other. By mapping these relationships, the project team can identify potential bottlenecks and ensure efficient resource use. The graph serves as a valuable tool for tracking progress and adjusting resource allocation as needed during the project.



Result: The practical has been successfully conducted.

Conclusion: The resource allocation graph helped clarify dependencies between tasks and allocated resources effectively, ensuring that the team maintained an optimal workflow with minimal delays.

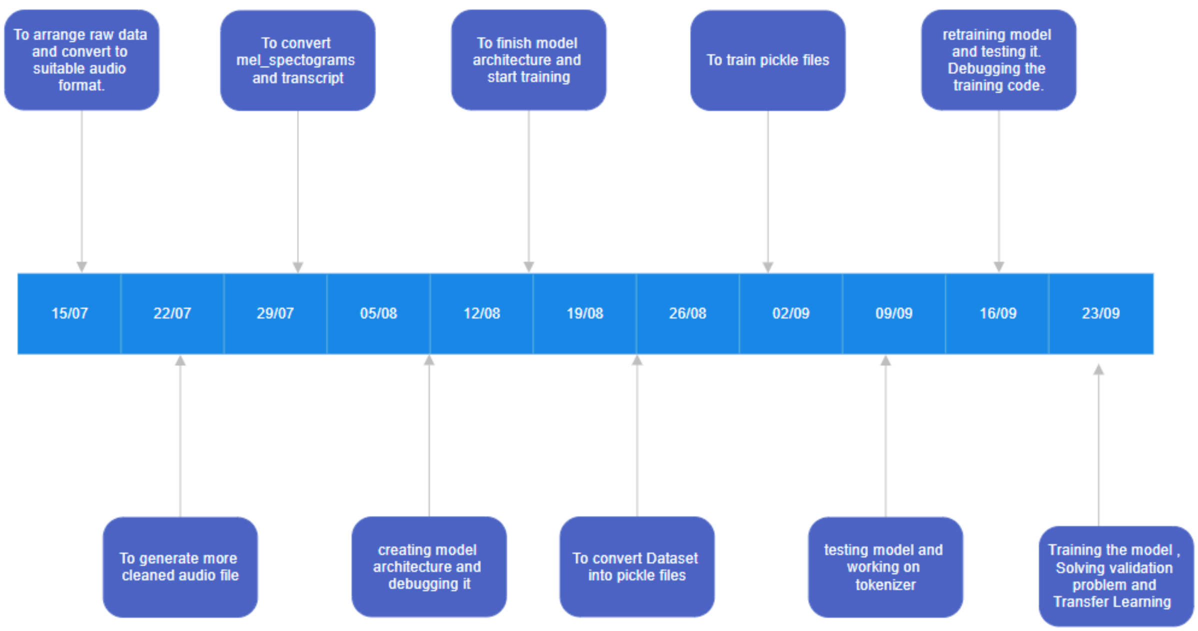
Criteria	Total marks	Marks Obtained	Comments
Concept (A)	2		
Implementation (B)	2		
Performance (C)	2		
Total	6		

PRACTICAL-8

AIM: Visualising Progress: 1. Timeline Chart 2. Kanban board

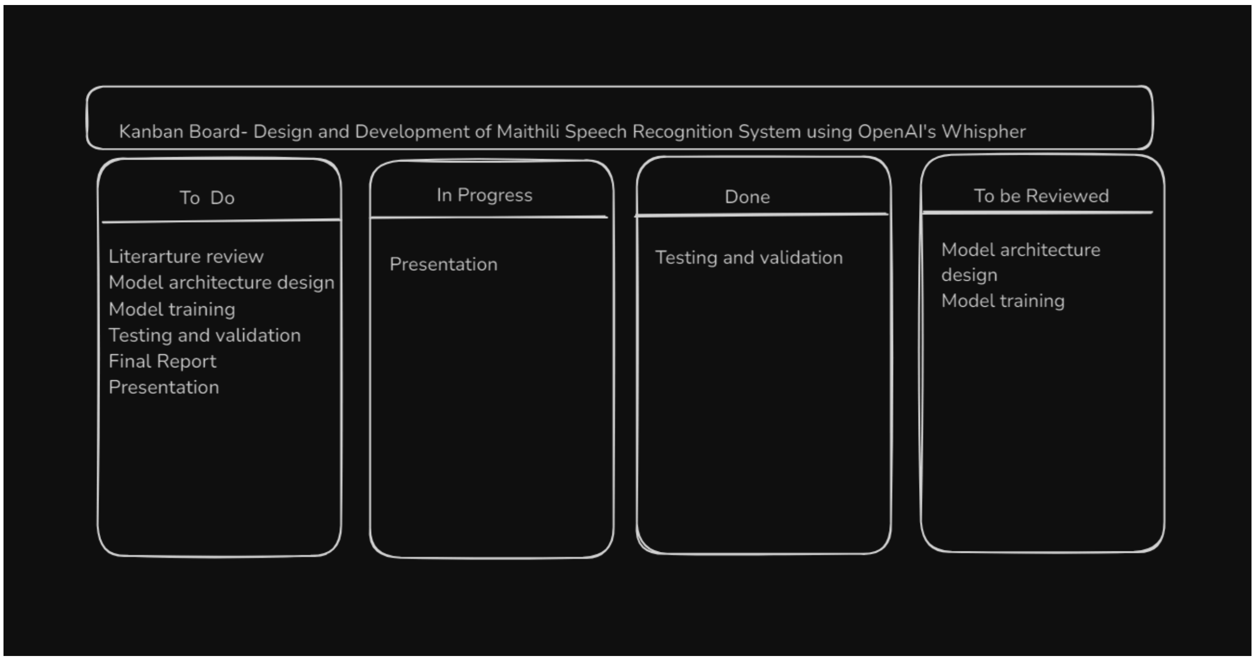
Theory: Timeline Charts

Timeline charts are visual tools that display a sequence of events over time. They're often used in project management to track progress, milestones, and deadlines. By visualizing the order and timing of tasks, timeline charts help stakeholders understand the project's timeline and identify potential bottlenecks.



Kanban Boards

Kanban boards are visual tools used for workflow management, often associated with agile methodologies. They represent a project's workflow as a series of columns (e.g., To Do, In Progress, Done) and cards (representing tasks or items). By visualizing the flow of work, Kanban boards help teams prioritize tasks, identify bottlenecks, and improve efficiency.



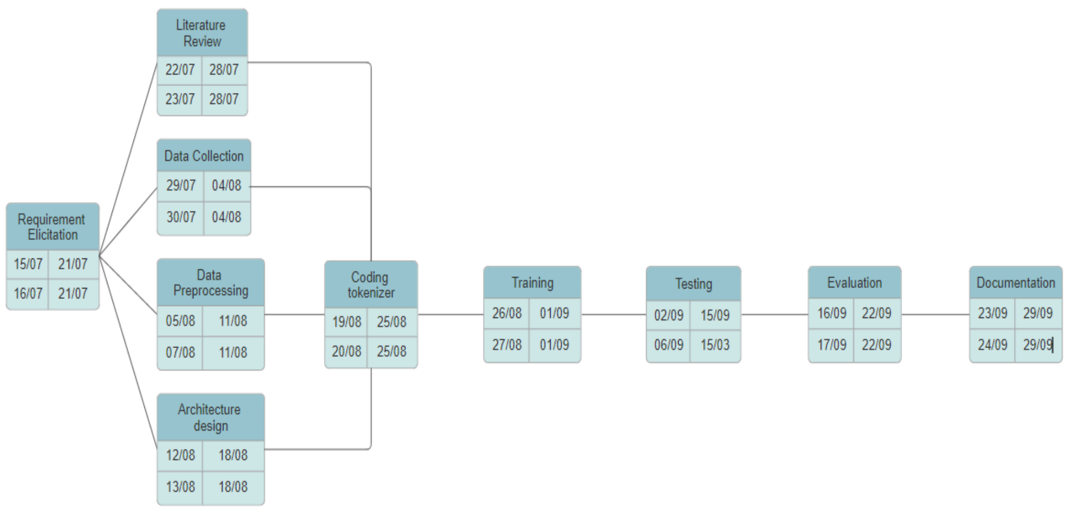
Result: The practical has been successfully conducted.
Conclusion: The timeline chart was essential in tracking the project's milestones and progress. It ensured that the project stayed on track, providing visual clarity on task completion.

Criteria	Total marks	Marks Obtained	Comments
Concept (A)	2		
Implementation (B)	2		
Performance (C)	2		
Total	6		

PRACTICAL-9

AIM: PERT Chart

THEORY: The aim of this practical is to create a Resource Allocation Graph that illustrates task dependencies and time estimations for the Maithili Speech Recognition System project. This graph visually represents which resources (team members, tools, etc.) are allocated to specific tasks and how these tasks depend on each other. By mapping these relationships, the project team can identify potential bottlenecks and ensure efficient resource use. The graph serves as a valuable tool for tracking progress and adjusting resource allocation as needed during the project.



Result: The practical has been successfully conducted.

Conclusion: The PERT chart helped estimate task durations, providing a probabilistic approach to managing uncertainties. This allowed for better planning of time-sensitive aspects of the project.

Criteria	Total marks	Marks Obtained	Comments
Concept (A)	2		
Implementation (B)	2		
Performance (C)	2		
Total	6		

PRACTICAL-10
AIM: Project Closure Report on Minor Project
Theory:A Project Closure Report is a formal document that marks the completion of a project and summarizes its outcomes. It is typically created at the end of a project to provide stakeholders with a comprehensive overview of the project's performance, including whether objectives were met, the final deliverables, lessons learned, and recommendations for future projects.

Project Closure Report: Maithili Speech Recognition System

Project Goals and Objectives

The Maithili Speech Recognition System project aimed to develop a robust and accurate speech-to-text system specifically designed for the Maithili language. The primary goals were to create a system that could effectively transcribe spoken Maithili into text, handle diverse dialects and speaking styles, and be integrated into various applications such as language learning platforms and accessibility tools**.**

Key objectives:

1. **D**evelop a Custom Speech Recognition Model: Implement a machine learning model capable of transcribing spoken Maithili with high accuracy.
2. Create a Maithili Speech Dataset: Curate a diverse dataset of Maithili audio samples with corresponding transcriptions.
3. Develop a Byte-Level BPE Tokenizer: Create a tokenizer tailored for the Maithili language to handle its unique linguistic properties.
4. Implement a Real-time Transcription Interface: Develop an easy-to-use interface where users can upload audio files and receive transcriptions.
5. Optimize Performance Metrics: Achieve a low Word Error Rate (WER) and Character Error Rate (CER) for accurate transcription.
6. Deploy the System for Public Use: Ensure the system is accessible via a web or mobile application for wider user testing and feedback.

Key Project Deliverables

The project resulted in several key deliverables:

1. Speech Recognition Model: A custom encoder-decoder model incorporating LSTM and attention mechanisms to transcribe spoken Maithili.
2. Preprocessed Dataset: A curated and preprocessed dataset of Maithili audio recordings paired with text transcriptions.

3. Byte-Level BPE Tokenizer: A tokenizer that effectively handles Maithili speech and maps it to token sequences.
4. Evaluation Metrics: Results of model evaluation, including WER and CER, to quantify the system's performance.
5. User Interface: A functional web-based interface for users to input audio and receive transcriptions.
6. Documentation and Reports: Detailed project documentation, including methodology, training, evaluation, and user feedback.

Benefits of Project Outcomes

The Maithili Speech Recognition System has several benefits:

1. Language Preservation: The project aids in preserving the Maithili language by providing a system that recognizes and transcribes the spoken language, supporting documentation, and promoting its use in digital spaces.
2. Accessibility: The system provides Maithili speakers with a tool for speech-to-text, which can help users with limited literacy or those who prefer verbal communication.
3. Educational Utility: The project serves as a foundation for language learning tools, enabling users to transcribe, analyze, and learn the Maithili language.
4. Technological Advancement: Contributes to the field of speech recognition technology, particularly for under-resourced languages like Maithili, paving the way for similar projects in other languages.

Lessons Learned

1. Importance of Diverse Data: The model's performance across different dialects highlighted the need for a more diverse dataset. Future projects should include audio samples from various regions and speakers to improve generalization.
2. Handling Noisy Data: One challenge was dealing with background noise in the recordings. Advanced noise-canceling techniques should be implemented to improve transcription accuracy in real-world scenarios.
3. Model Optimization: Balancing model complexity and performance was key. We learned that while adding layers improved accuracy, overfitting could occur, so regularization techniques should be applied.
4. User Feedback: Early user feedback helped identify areas of improvement, particularly in user interface design and transcription clarity. Regular iterations based on user feedback are essential for success.
5. Collaboration is Crucial: Team coordination and collaboration were vital in ensuring the timely completion of various project components.

Objective, Deliverables, and Benefits Table

Objective	Achieved Deliverables	Benefits Realized	Objective Met? (Yes/No)
Develop a Custom Speech Recognition Model	Encoder-decoder model developed	Accurate Maithili transcription	Yes
Create a Maithili Speech Dataset	Diverse dataset curated	Improves model training	Yes
Develop Byte-Level BPE Tokenizer	Custom Maithili tokenizer created	Efficient tokenization process	Yes
Implement Real-time Transcription Interface	User interface deployed	Real-time user interaction	NO
Optimize Performance Metrics	Low WER and CER achieved	Improves system accuracy	Yes
Deploy the System for Public Use	System deployed for testing	Accessibility for users	No (pending wider deployment)

Result: The practical has been successfully conducted.

Conclusion: The Maithili Speech Recognition System successfully achieved its primary objectives, developing a working model for speech-to-text transcription, creating a tokenizer, and building a user interface for testing. While the project made significant progress, there are areas for improvement, such as expanding the dataset and refining the system for real-world scenarios. The project provides a solid foundation for future work in speech recognition for under-resourced languages, and with continued improvements, it has the potential to serve a wide range of applications, from education to accessibility tools.

Criteria	**Total marks**	**Marks Obtained**	**Comments**
Concept (A)	2		
Implementation (B)	2		
Performance (C)	2		
Total	6		