

AI Guide: Interactive Lecture Sessions with Virtual Instructors

Software Design and Requirement Specification



Session: 2021 – 2025

Submitted by:

Mahad Mateen 2021-CS-657

Samia Ishfaq 2021-CS-697

Iqra Iqbal 2021-CS-671

Supervised by:

Dr. Qurat-ul-Ain Akram

Department of Computer Science, New Campus
University of Engineering and Technology
Lahore, Pakistan

Contents

List of Figures	ii
List of Tables	iii
1 Requirement Specification	1
1.1 Functional Requirement	1
1.2 Non-Functional Requirement	4
1.3 Use Case Diagram	5
1.4 Sequence Diagrams	6
2 Design specification	10
2.1 Detailed Literature Review	10
2.2 Summaries Of Related Article	15
2.3 Proposed Methodology	22
2.4 Data Collection Techniques	26
2.5 Experimental Design	28
2.6 App Design	37
References	46

List of Figures

1.1	Use Case Diagram	5
1.2	Sequence Diagram for Login Functionality	6
1.3	Sequence Diagram for Sign-up Functionality	7
1.4	Sequence Diagram for Voice Cloning Process	7
1.5	Sequence Diagram for Lip-Synching Process	8
1.6	Sequence Diagram for Video Cloning Process	8
1.7	Sequence Diagram for Lecture Downloading Process	9
2.1	Multi-Speaker TTS Methodology	22
2.2	Lip Synching Methodology	23
2.3	Model Training For Multi-Speaker TTS	31
2.4	Model Training For Lip-Synching Model	36
2.5	Home Page	37
2.6	Login Page	38
2.7	Sign-Up Page	39
2.8	Select Language Page	40
2.9	Upload Insert Page	41
2.10	Upload Text	42
2.11	Select Speaker Page	43
2.12	Download Page	44
2.13	Downloading Progress Page	45

List of Tables

2.1	Summaries	15
-----	---------------------	----

Chapter 1

Requirement Specification

1.1 Functional Requirement

- **BR-Business requirements:**

- **FRBR1** - Our system shall allow users to upload text and convert them into cloned speech using AI-generated speaker voices.
- **FRBR2** - Our system shall provide accurate lip-syncing for the cloned voices to match the speaker's facial movements in the video.
- **FRBR3** - Our system shall enable users to translate English text into Urdu, ensuring proper context and language alignment.
- **FRBR4** - Our system shall offer multiple speaker options for users to choose from a list of speakers when converting textual lecture into video presentations.
- **FRBR5** - Our system shall deliver a mobile application compatible with Android.
- **FRBR6** - Our system shall implement efficient back-end services to handle data processing, user management, and audio/video cloning tasks.
- **FRBR7** - Our system shall allow users to review and edit the final output of the video before downloading or sharing it.

- **AF-Administrative Functions:**

- **FRAF1 - System Performance Monitoring:** The system shall provide administrators with real-time monitoring dashboards displaying

system performance metrics such as server load, API usage, response times, and error rates.

- **FRAF - AI Model Management:** The system shall allow administrators to upload, update, and manage AI models for voice cloning, lip-syncing, and video generation, as well as retrain models with new data.

- **UR-User Requirements:**

- **FRUR1 - User Account Creation:** Users shall be able to create an account with a username, email, and password, or log in using third-party services like Google etc.
- **FRUR2 - Text Upload:** Users shall be able to upload text to the platform for conversion into cloned voices and lip-synced videos.
- **FRUR3 - Language Selection:** Users shall have the option to choose the language of the text input (i.e; English, Urdu) and automatically translate between languages, with accurate context retention.
- **FRUR4 - Speaker Selection:** Users shall be able to select from a list of pre-trained AI-generated voices to convert their text into a video with a cloned speaker voice of their choice.
- **FRUR5 - Video Customization:** Users shall have the ability to customize the generated video by selecting or changing backgrounds and previewing the video before finalizing it.
- **FRUR6 - Real-time Processing:** Users shall be able to view real-time lip-syncing and video rendering of the uploaded text, ensuring integration of voice, lip movement, and video.
- **FRUR7 - Edit and Preview:** Users shall have the ability to review and make final edits to the video output before downloading.
- **FRUR8 - Download and Sharing:** Users shall be able to download the final generated video.
- **FRUR9 - Device Compatibility:** Users shall be able to access the platform via a mobile application compatible with Android or via a web interface.
- **FRUR10 - Account Settings:** Users shall be able to manage their account settings, including updating personal information, changing password.

- **FRUR11 - Notifications:** Users shall receive notifications about the status of their uploads, processing times, and the availability of the final video output, as well as any platform updates or new features.
- **SR-System Requirements:**
 - **FRSR1** - The system shall support text file upload by ensuring words limit maximum of 2000 words and .txt format of input file.
 - **FRSR2** - The system shall use Tacotron 2 or similar deep learning models for voice cloning, ensuring a natural and accurate reproduction of the selected speaker's voice.
 - **FRSR3** - The system shall utilize Wav2Lip or a comparable lip-syncing technology to synchronize cloned audio with the speaker's lip movements in video presentations.
 - **FRSR4** - The system shall integrate APIs for translation services to convert English text to Urdu and ensure accurate context retention during translation.
 - **FRSR5** - The system shall provide a minimum of five pre-trained speaker voices for users to choose from, including male and female options.
 - **FRSR6** - The system shall use deep learning frameworks such as TensorFlow or PyTorch to train and manage AI models for real-time video rendering and background replacement.
 - **FRSR7** - The system shall use React Native or Flutter to develop a mobile application compatible with Android.
 - **FRSR8** - The system shall implement a back-end infrastructure using Python-based frameworks like Django or Flask, ensuring scalable and secure data processing.
 - **FRSR9** - The system shall utilize cloud-based services for data storage, processing, and handling large video rendering tasks.
 - **FRSR10** - The system shall support NLP libraries like NLTK or spaCy to analyze lecture transcripts and ensure accurate speaker cloning based on key information extracted from the text.
 - **FRSR11** - The system shall allow users to preview the processed video output before downloading, providing options for background selection.

- **FRSR12** - The system shall support secure user authentication and authorization protocols to protect user data and prevent unauthorized access.

1.2 Non-Functional Requirement

- Usability: Our system must have user friendly interface design with clear navigation and easily recognizable buttons and controls.
- Reliability / Availability: Our system must have a high level of availability, aiming for minimal downtime to ensure continuous access for users.
- scalability: Our system must be able to handle increasing user load and data volume.
- Performance: Our system must be able to maintain high performance even under high loads like for large dataset processing.
- Security: The system should adhere to implement data protection and privacy regulations to ensure the confidentiality and integrity of user data.

1.3 Use Case Diagram

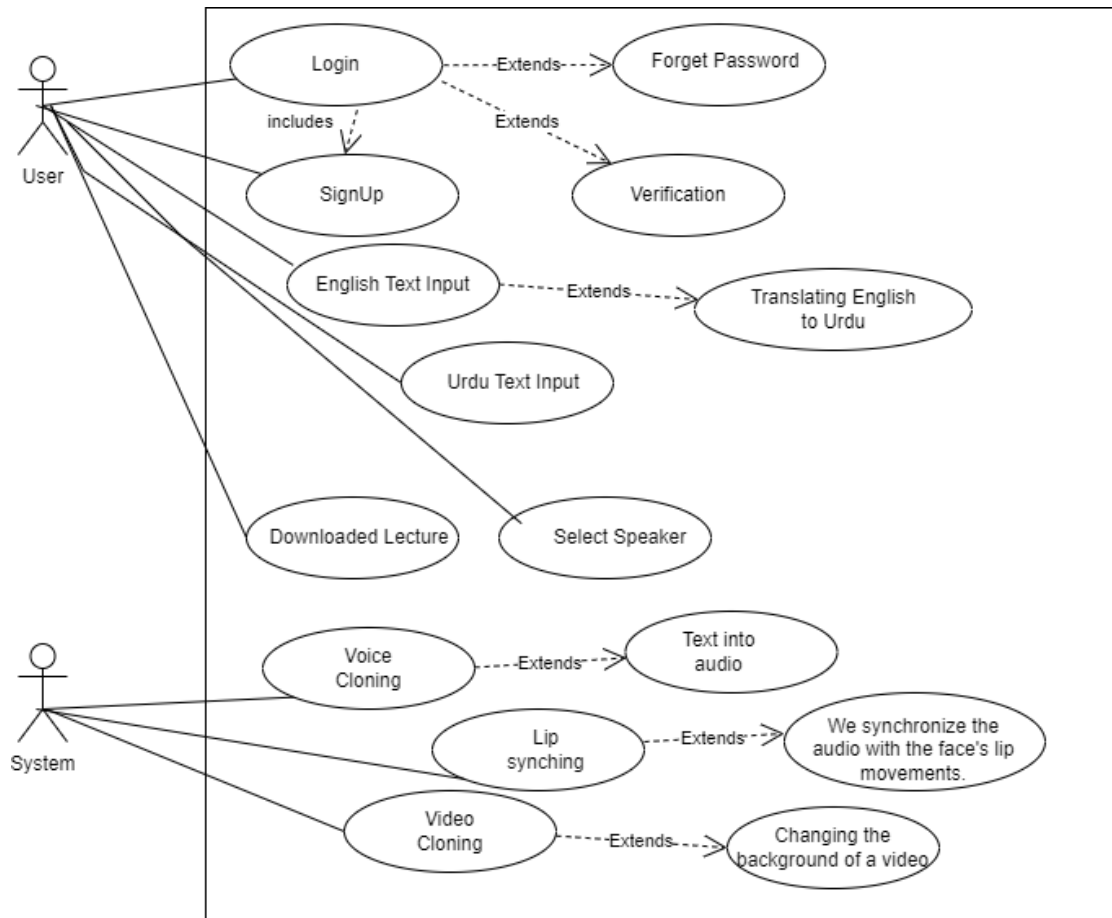


FIGURE 1.1: Use Case Diagram

1.4 Sequence Diagrams

Sequence diagram helps to identify the flow of the system Here are the detailed sequence diagrams of our system;

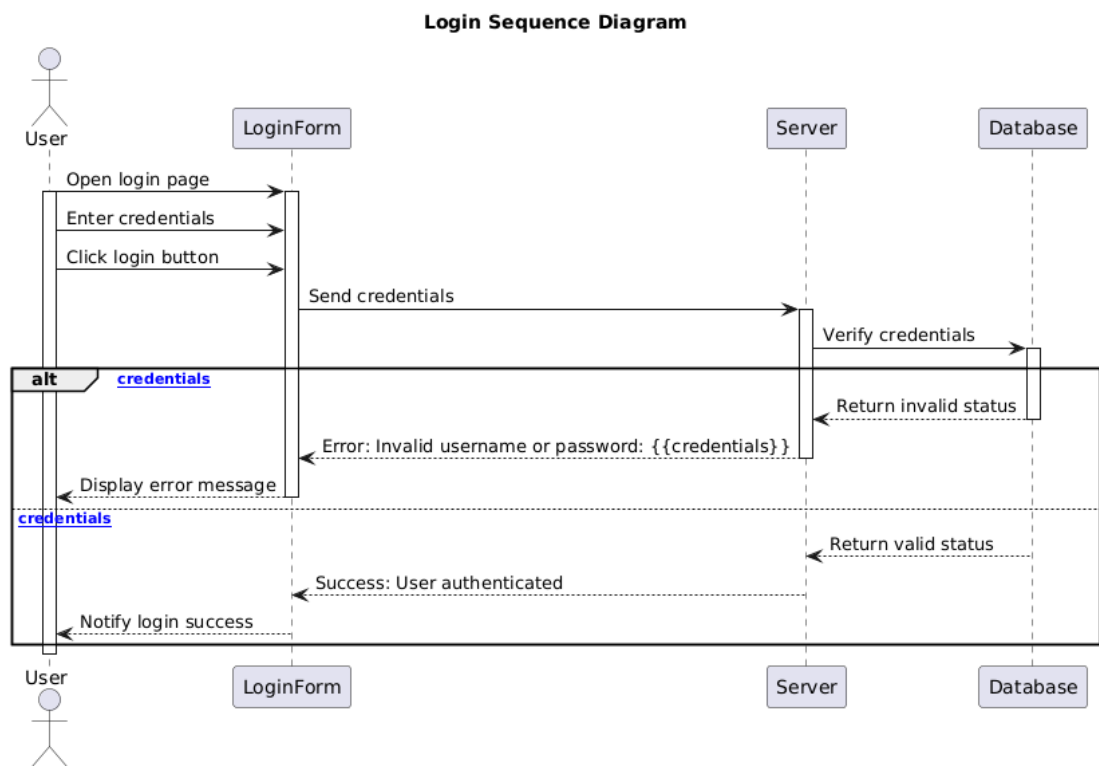


FIGURE 1.2: Sequence Diagram for Login Functionality

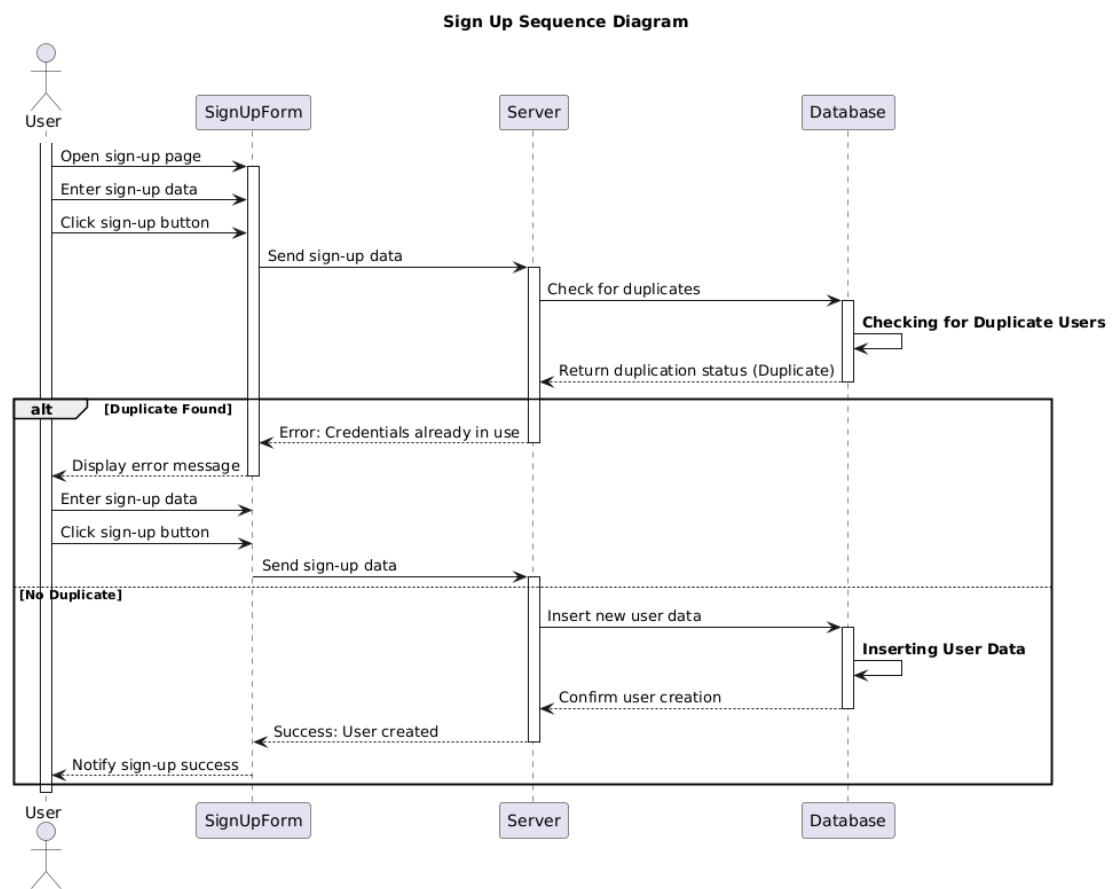


FIGURE 1.3: Sequence Diagram for Sign-up Functionality

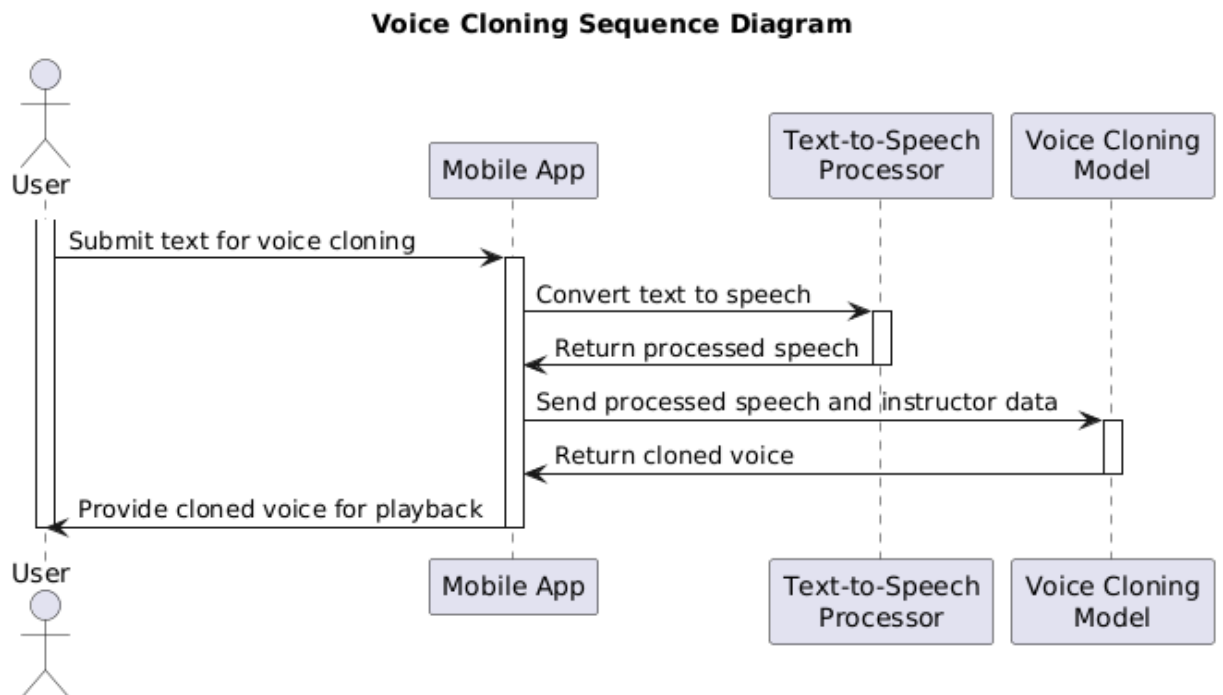


FIGURE 1.4: Sequence Diagram for Voice Cloning Process

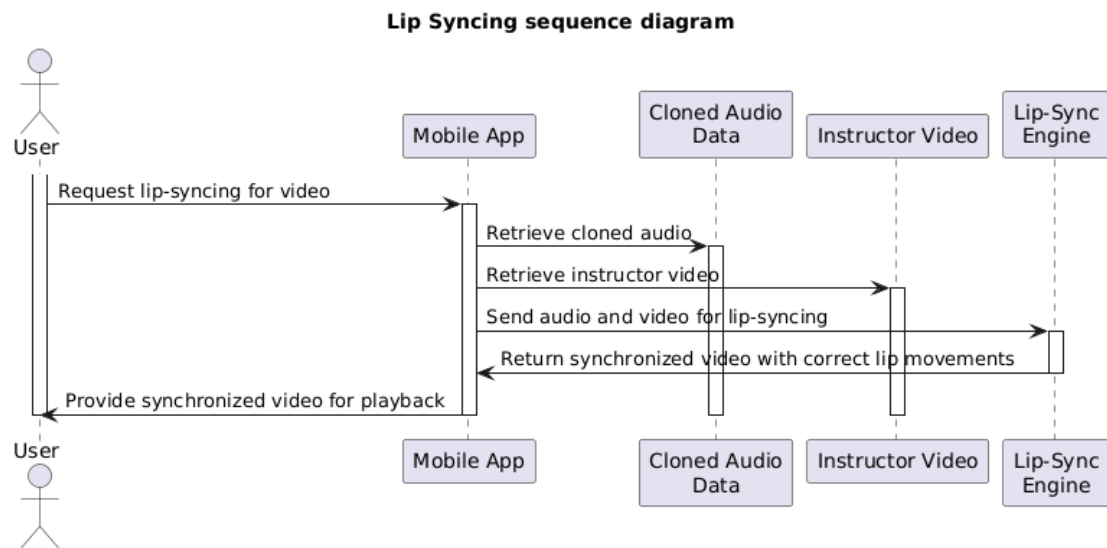


FIGURE 1.5: Sequence Diagram for Lip-Syncing Process

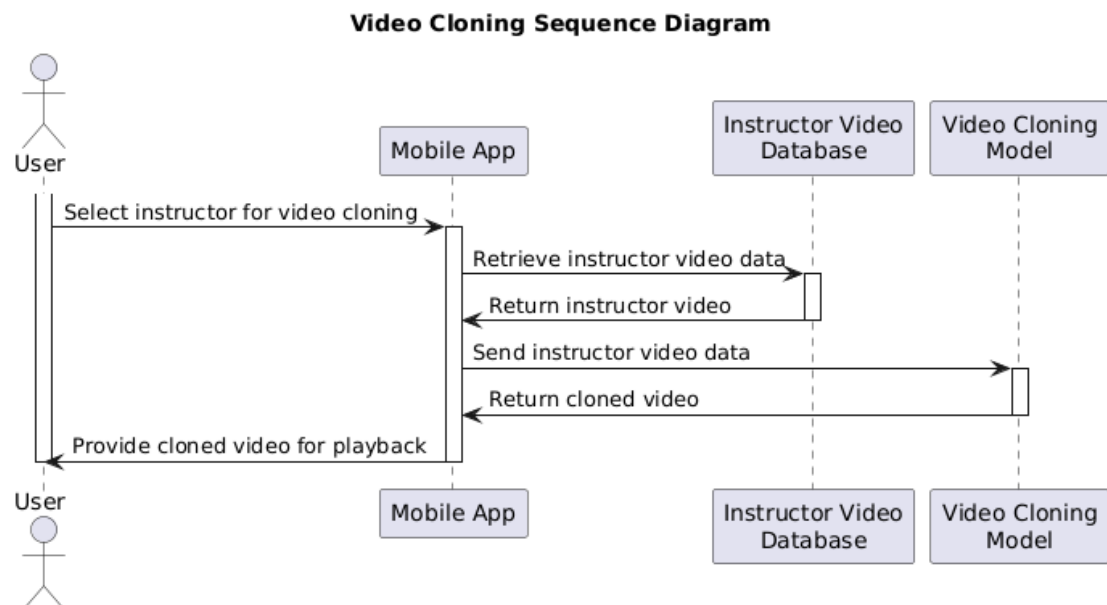


FIGURE 1.6: Sequence Diagram for Video Cloning Process

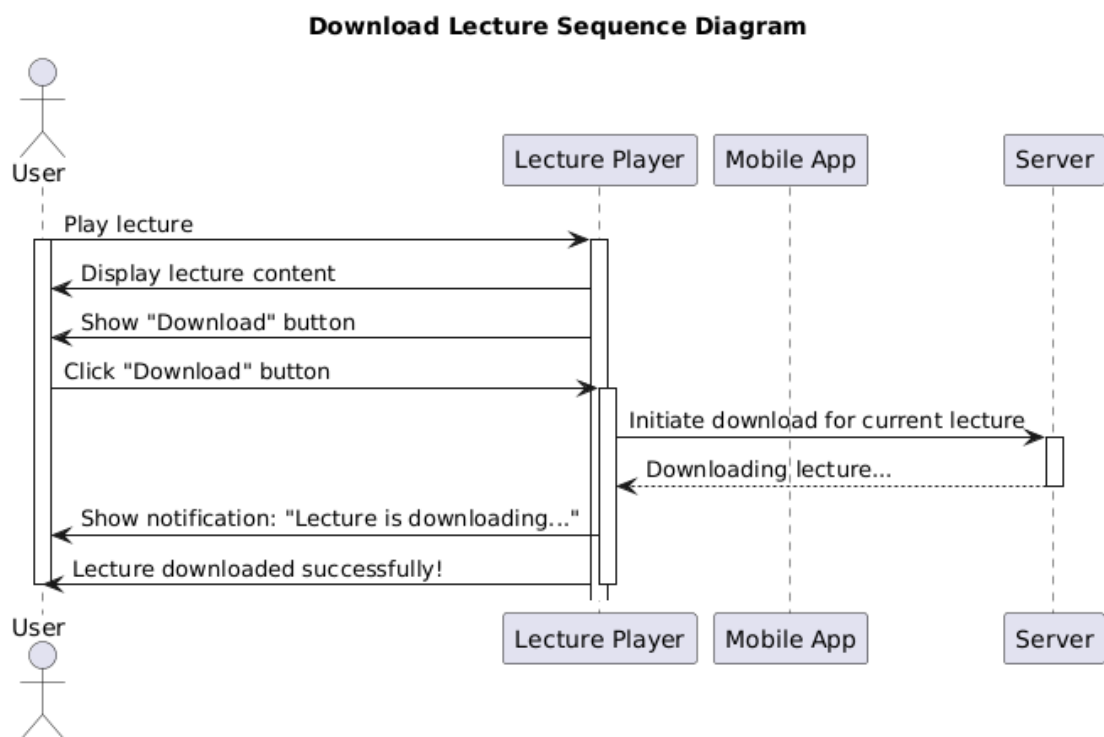


FIGURE 1.7: Sequence Diagram for Lecture Downloading Process

Chapter 2

Design specification

2.1 Detailed Literature Review

”Chen et al[11]” provides an overview of the system’s ability to generate speech audio in the voices of different speakers, including those not seen during training. It highlights the system’s three main components: a speaker encoder network trained on noisy speech from numerous speakers to generate a fixed-dimensional embedding from a few seconds of reference speech. A Tacotron 2-based sequence-to-sequence synthesis network that generates a mel spectrogram from text conditioned on the speaker embedding. And a WaveNet-based vocoder that converts the mel spectrogram into a waveform. The paper emphasizes the importance of training the speaker encoder on a large, diverse dataset to enhance generalization performance. It also demonstrates that randomly sampled speaker embeddings allow for speech synthesis in voices dissimilar to those seen during training, indicating the model’s ability to learn high-quality speaker representations.

”Mendes et al[7]” provides a clear overview of the research focus on enhancing the naturalness of cloned voices and achieving real-time voice synthesis. It highlights the advancements since 2016, when earlier methods required many hours of voice samples to generate a few seconds of speech. With the introduction of deep learning models, this has been reduced to just a few seconds of reference audio. The paper reviews various voice cloning techniques, such as multi-speaker generative models, speaker adaptation, speaker encoding, and vector quantization, drawing attention to their significance in improving the efficiency and quality of voice cloning.

”**Chen et al**[1]” provides a detailed examination of a voice cloning system that can synthesize a person’s voice using only a few audio samples. It explores two primary approaches: speaker adaptation and speaker encoding. Speaker adaptation involves fine-tuning a multi-speaker generative model, while speaker encoding focuses on training a separate model to infer a new speaker embedding, which is then applied to the multi-speaker generative model. The paper highlights that both methods can achieve good performance in terms of naturalness and similarity to the original speaker, even with minimal cloning data. Although speaker adaptation offers slightly better naturalness and similarity, speaker encoding is more efficient in terms of cloning time and memory requirements, making it a favourable option for low-resource environments.

”**Seong et al**[9]” explores the evolution of speech synthesis from simple text-to-speech conversion to the imitation of user voices. In earlier stages, voice generation involved recording commonly used sentences and converting text into speech, but limitations in naturalness and accuracy led to research in phonetics, linguistics, and statistics. With advancements in big data, AI, and parallel processing technologies, it has become possible to replicate human tone and timbre with high precision. The paper highlights potential security risks, as natural voice generation could make it difficult for machines to distinguish between real and synthetic voices, posing threats to personal information security. Additionally, the paper introduces speech synthesis models and voice generation techniques, examining their application in converting and synthesizing Korean into other languages, while also addressing future security concerns.

”**Kadam et al**[5]” explores the transformative potential of voice cloning, which synthesizes speech in a target voice using text and a few audio samples. Recent progress in deep learning has propelled voice cloning, offering new possibilities in human-computer interaction, personalization, and content creation. The paper introduces a personalized voice cloning system that employs deep learning models for text-to-speech (TTS) synthesis and audio generation, enabling users to input text and provide an audio sample to generate speech. By capturing the user’s unique vocal traits, the system produces natural-sounding speech output. The paper highlights various applications, such as actors dubbing films in different languages, individuals who have lost their voices using this technology to communicate, and advertisers generating diverse ad reads. Additionally, it underscores the commercial opportunities, with companies using voice cloning to create voices for chatbots, audiobooks, video games, and more.

”**Goyal et al**[4]” delves into the development of pipelines for generating lip-synced talking faces, aimed at applications like teaching and language translation in videos. Previous approaches have struggled to create realistic videos due to their limited focus on facial expressions and emotions, and their performance has been heavily dependent on the faces in the training dataset. To address these limitations, the paper presents a framework for talking face generation that incorporates categorical emotions, making the resulting videos more lifelike and convincing. By using six key emotions—happiness, sadness, fear, anger, disgust, and neutral—the model demonstrates its ability to adapt to various identities, emotions, and languages. The framework also includes a user-friendly web interface that allows for real-time talking face generation with emotion control. A user study was conducted to evaluate the interface’s usability, design, and functionality.

”**Jamaludin et al**[2]” introduces a method for generating a video of a talking face using two inputs: (i) still images of the target face and (ii) an audio speech segment. The system produces a video where the target face is lip-synced with the provided audio. Notably, this method operates in real time and is effective on faces and audio that were not part of the training data. The approach utilizes an encoder-decoder CNN model, which creates a joint embedding of the face and audio to generate the video frames of the talking face. The model is trained on tens of hours of unlabeled videos. The paper also demonstrates the method’s ability to re-dub videos using speech from a different speaker.

”**Guan et al**[10]” addresses the challenge of balancing lip-sync generation quality with model generalization. Previous methods often rely on long-term data or produce uniform, low-quality movement patterns across subjects. In this work, StyleSync introduces an effective framework for achieving high-quality lip synchronization, even in one-shot and few-shot scenarios. The paper proposes a mask-guided spatial information encoding module that preserves facial details, while modulated convolutions accurately adjust mouth shapes based on the audio input. Additionally, the model supports personalized lip-sync by leveraging style space and refining the generator on a limited number of frames, ensuring that the identity and talking style of the target person are accurately maintained. Extensive experiments validate the method’s ability to deliver high-fidelity results across various scenes. Resources for the project are available at <https://hangz-nju-cuhk.github.io/projects/StyleSync>.

”**Dhanush et al**[3]” explores the critical role of dataset quality in the performance of lip-reading systems, which analyze lip movements to interpret a speaker’s message. Widely applied in various daily life scenarios, the paper emphasizes the

importance of accurate dataset creation. The study uses Scikit-Video to extract frames from source videos, followed by facial detection using Idlib. Lip cropping is achieved by processing facial feature points to isolate lip images. The dataset is further expanded through data augmentation techniques, consisting of 33 voices, each represented by 7,000 lip images per speaker. The paper also proposes a methodology for creating such datasets, emphasizing the preprocessing of videos using the Scikit-Video library.

”**Feng et al**[12]” addresses the challenge of improving the perceptual quality of speech signals. Traditional models often fail to ensure high perceptual quality due to their reliance on L1 or L2 loss functions. This study introduces a novel perceptual loss that leverages lip movement information for speech enhancement. The motivations for this approach are twofold: first, the correlation between spoken language content and lip movements, and second, evidence suggesting that multi-modal models incorporating lip information as auxiliary input outperform audio-only models. To facilitate the extraction of perceptual features and compute the perceptual loss using an audio-only dataset, the paper describes a network that generates lip movements from speech signals, guided by the Wav2Lip model. Experimental results demonstrate that incorporating lip movement-based perceptual loss significantly enhances speech enhancement performance.

”**Hashemi**[13]” presents an assessment of various tools available on the Hugging-Face platform, focusing on two key application categories: video in-painting and voice cloning. The goal of this evaluation is to identify the top three tools in each category based on their effectiveness and usability. Following this selection process, the paper outlines the next steps, which involve installing and configuring the chosen tools on Linux systems to facilitate further experimentation and application.

”**Tariq et al**[6]” addresses the growing concerns surrounding the misuse of deepfake technology, which poses significant security and privacy risks by enabling the impersonation of individuals through video and synthesized audio. With recent advancements allowing AI models to generate a person’s voice using only a few seconds of audio, the threat of impersonation attacks has escalated, highlighting the need for a new generation of deepfake detectors that analyze both audio and video in tandem. The authors emphasize the importance of high-quality data for developing effective detectors, noting that existing datasets are often limited to either deepfake videos or audios and may exhibit racial biases. To address these challenges, the paper introduces FakeAVCeleb, a comprehensive Audio-Video

Deepfake dataset that includes both deepfake videos and corresponding synthesized lip-synced audio. This dataset is created using popular deepfake generation methods and features real YouTube videos of celebrities from four different ethnic backgrounds, aiming to mitigate racial bias and support the development of robust multimodal deepfake detectors. The authors conducted experiments using state-of-the-art detection techniques to evaluate the dataset, highlighting its challenges and demonstrating its utility in advancing deepfake detection research.

”**Masood et al**[8]” addresses the alarming trend of deepfake media, fueled by easy access to audio-visual content on social media, modern tools like TensorFlow and Keras, and the availability of open-source models and affordable computing resources. The proliferation of Generative Adversarial Networks (GANs) has facilitated the creation of deepfakes that can deceive audiences, spreading disinformation, revenge porn, financial fraud, and undermining government functions. While existing surveys have primarily concentrated on detecting deepfake images and videos, this paper offers a thorough review and analysis of tools and machine learning (ML) approaches for both deepfake generation and the methodologies used for detection in audio and video. The authors discuss various manipulation techniques, current public datasets, and key evaluation standards for assessing deepfake detection performance, alongside the results achieved. Moreover, the paper highlights open challenges in the field and outlines future directions for researchers, emphasizing important considerations for enhancing both deepfake generation and detection methodologies. This work aims to deepen readers’ understanding of how deepfakes are produced and identified, as well as their limitations and potential avenues for future research.

The summary of these research papers is presented in the Table [2.1](#)

2.2 Summaries Of Related Article

TABLE 2.1: Summaries

Related Articles	Article Topic	Models	Dataset	Results
Chen et al[11]	Voice Cloning	Speaker Encoder Network Sequence-to-Sequence Synthesis Network (Tacotron 2) Auto-Regressive Wave Net-based Vocoder	VCTK Corpus: 109 speakers, 44 hours. LibriSpeech Dataset: 1,172 speakers, 436 hours.	Synthesizer: 56.77 Speaker Encoder: 38.54 (Speaker Verification Equal Error Rate)
Mendes et al[7]	Voice Cloning	Multi-speaker Generative Model	Similarity Test: 10 speakers (randomly selected from seen and unseen speakers).	Both speaker adaption and speaker encoding achieve an MOS similar to the baseline
Chen et al[1]	Voice Cloning	Multi-Speaker Generative Model a speaker verification model	LibriSpeech Dataset: 2,484 speakers, 820 hours, 16 kHz, used for training multi-speaker model and speaker encoder VCTK Dataset: 108 speakers, 48 kHz (downsampled to 16 kHz), used for voice cloning. 84 speakers for training, 8 for validation, 16 for cloning.	Mean Opinion Score (MOS) evaluations for naturalness with 95 percent confidence interval

Related Articles	Article Topic	Models	Dataset	Results
Seong et al[9]	Voice Cloning	Speech synthesis models: DeepVoice, Tacotron Speech generation models: WaveNet, Parallel WaveNet, WaveGlow, MelGAN	MagnaTagATune: 25,000 music clips, 29 seconds each. YouTube Piano Dataset: Piano performances from YouTube	MelGAN is over 10x faster than WaveGlow on GPU, with model size reduced from 300MB to 10MB, maintaining high-fidelity, real-time performance.
Kadam et al[5]	Voice Cloning	Speaker Encoder Model Tacotron 2	LibriTTS: 247 speakers, 54 hours, 24 kHz, 8.7 GB (train-clean-100 split). LibriSpeech: 1,000 hours, 16 kHz.	Results for this research are not given in numbers. Infact we have results in form of graph.

Related Articles	Article Topic	Models	Dataset	Results
Goyal et al[4]	Lip-Synching	-Wave2lip -Sequential Concatenation (SEQ) -Perceptual Loss and Data Augmentation (PL+DA) -Pre-training with PL+DA (PRE)	CREMA-D Dataset: Clips: 7,442 audio-visual clips. Actors: 91 (48 male, 43 female), aged 20-74. Sentences: 12 sentences spoken by actors. Emotions: Labeled with 6 emotions (Anger, Disgust, Fear, Happy, Neutral, Sad) at 4 levels (Low, Medium, High, Unspecified).	Emotion-wise Accuracy: Wave2Lip Model: 75.02 Simple Concatenation (END) Model: 21.48 Sequential Concatenation (SEQ) Model: 71.51 Perceptual Loss and Data Augmentation (PL+DA) Model: 83.20 Pre-training with PL+DA (PRE) Model: 78.14

Related Articles	Article Topic	Models	Dataset	Results
Jamaludin et al[2]	Lip-Synching	Speech2Vid	<p>VoxCeleb Dataset:</p> <p>Content: Short clips of human speech extracted from YouTube interview videos.</p> <p>Utterances: Over 1 million utterances.</p> <p>Speakers: More than 7,000 speakers.</p> <p>Duration: Over 2,000 hours; each segment is at least 3 seconds long.</p> <p>Gender Distribution: 61 percent male, 39 percent female. LRW Dataset</p> <p>Words: 500 different words</p> <p>Utterances: Up to 1,000 utterances per word.</p> <p>Speakers: Hundreds of contributors.</p> <p>Video Length: 29 frames (1.16 seconds) per video.</p>	They qualitatively find that we strike the best balance between image naturalness and movement naturalness by only blending the lower half of the face, from just below the eyes.

Related Articles	Article Topic	Models	Dataset	Results
Guan et al[10]	Lip-Synching	Wav2lip Wav2Lip-H	<p>VoxCeleb Dataset:</p> <p>Content: Short clips of human speech extracted from YouTube interview videos.</p> <p>Utterances: Over 1 million utterances.</p> <p>Speakers: More than 7,000 speakers.</p> <p>Duration: Over 2,000 hours; each segment is at least 3 seconds long.</p> <p>Gender Distribution: 61 percent male, 39 percent female. LRW Dataset</p> <p>Words: 500 different words</p> <p>Utterances: Up to 1,000 utterances per word.</p> <p>Speakers: Hundreds of contributors.</p> <p>Video Length: 29 frames (1.16 seconds) per video.</p>	<p>Wav2Lip: • LRW: SSIM 0.79, PSNR 30.54, LMD 1.28, Sync conf 7.39, DID 0.90 • VoxCeleb2: SSIM 0.80, PSNR 30.53, LMD 1.92, Sync conf 8.90, DID 0.90</p> <p>Wav2Lip-H: • LRW: SSIM 0.80, PSNR 31.38, LMD 1.20, Sync conf 7.19, DID 0.88 • VoxCeleb2: SSIM 0.81, PSNR 30.53, LMD 1.87, Sync conf 8.35, DID 0.90</p>

Related Articles	Article Topic	Models	Dataset	Results
Dhanush et al[3]	Lip-Synching	Idlib: is a cross-platform library created in C++ that includes a variety of machine learning techniques.	Grid Dataset: Content: English lip-language dataset. Participants: 34 individuals (video for participant 21 is missing). Phrases Recorded: Each participant recorded 1,000 phrases. Video Length: Approximately 3 seconds per video. Frame Rate: 25 frames per second.	The authors present the design concept for creating a dataset to support this project.
Feng et al[12]	Speech Enhancement	Wav2lip Model (Guiding Model) Lips-ync Model	VCTK-DEAND Dataset: Training Sentences: 11,572 sentences Testing Sentences: 824 sentences Speakers: Training: 28 speakers Testing: 2 speakers	Pre-trained Wav2Lip: • LSE-D: 6.30 • LSE-C: 8.56 Lip-sync Model: • LSE-D: 6.21 • LSE-C: 8.42

Related Articles	Article Topic	Models	Dataset	Results
Hashemi[13]	Video Cloning	DETR with ResNet-50 SegFormer B0 Segment Anything Model (SAM)	<p>COCO 2017:</p> <p>Content: 118,000 annotated images across 80 object categories (common objects, animals, everyday scenes).</p> <p>Annotation: Object labels, segmentation masks, bounding boxes.</p> <p>Usage: Training, testing, and benchmarking video cloning algorithms.</p> <p>ADE20K:</p> <p>Content: 20,000 images with diverse indoor and outdoor scenes, featuring complex layouts and multiple objects.</p> <p>Annotation: Pixel-level labels for detailed semantic segmentation masks.</p> <p>Usage: Training and evaluating models for semantic segmentation and scene understanding. .</p>	SAM achieves competitive mIoU scores 8.1+ 0.07 mask quality rating IOU=area of intersection/area of union

Related Articles	Article Topic	Models	Dataset	Results
Tariq et al[6]	Video Cloning	FSGAN Wav2lip	FakeAVCeleb Dataset: Source: Generated from the VoxCeleb2 dataset. VoxCeleb2 Content: Total Videos: 1,092,009 in the development set; 36,237 in the test set. Speakers: 6,112 celebrities. Video Duration: Average of 7.8 seconds.	The AUC scores of these SOTA models on our FakeAVCeleb are 72.5, 61.7, and 60.9.

2.3 Proposed Methodology

The methodology for developing the AI Guide platform follows a structured approach, aligned with the provided diagram. The process is divided into key phases that ensure the project addresses the research problem efficiently by using both *Text-to-Speech (TTS)* and *Wav2Lip* for audio-video synchronization.

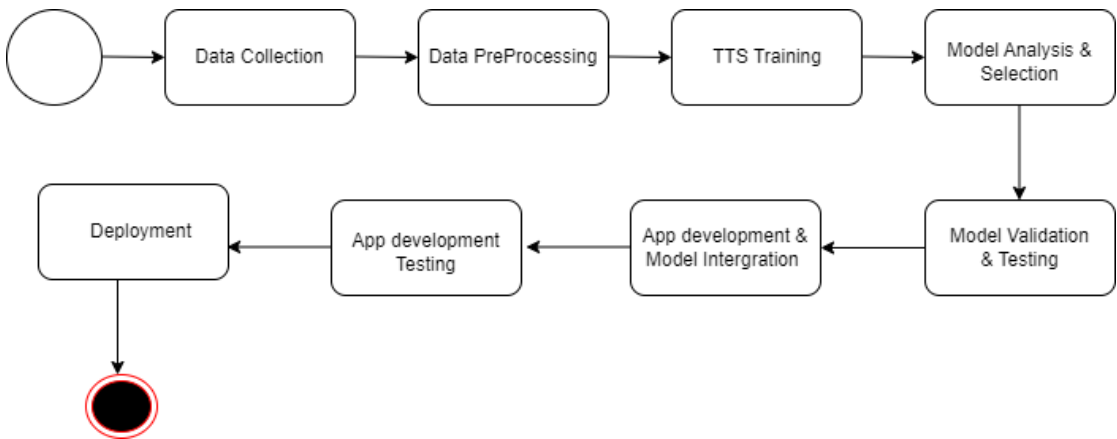


FIGURE 2.1: Multi-Speaker TTS Methodology

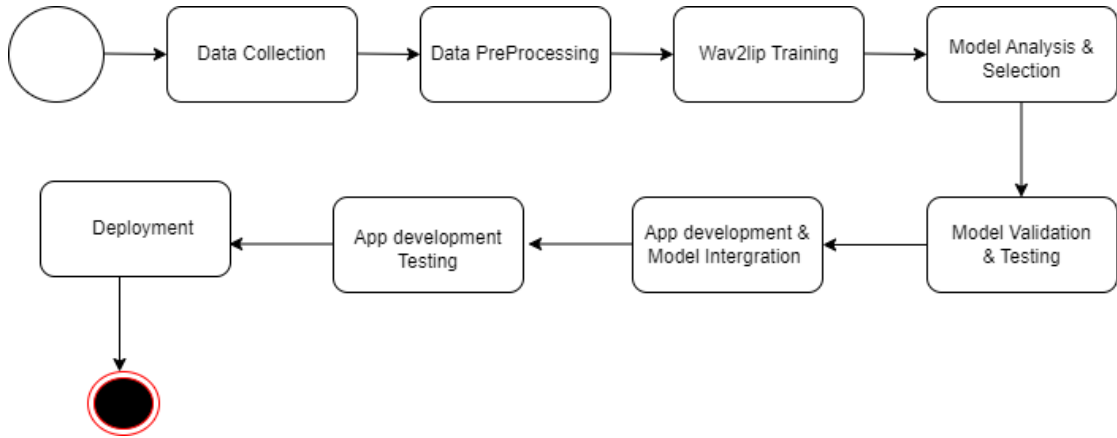


FIGURE 2.2: Lip Synching Methodology

- **Data Collection:**

- **Objective:** Gather relevant data for developing the AI Guide platform.
- **Action:**
 - * Collect lecture audio, video, text materials, and data on computer science education in Pakistan, focusing on language barriers and traditional teaching challenges.
 - * Data Collection Techniques are mentioned in Section 2.4

- **Data Preprocessing:**

- **Objective:** Clean and prepare the collected data for effective use in AI models.
- **Action:**
 - * **Audio Preprocessing:** Normalize audio levels and ensure consistent sampling rates (e.g., 16 kHz). Convert audio files to WAV format. Split long audio files into smaller segments based on sentences for better TTS performance.
 - * **Text Preprocessing:** Clean and standardize text transcriptions, ensuring proper tokenization and handling of punctuation. Use phoneme or grapheme conversion for better pronunciation modeling.
 - * **Video Processing:** Extract frames from video lectures to create a dataset for training the Wav2Lip model. Ensure that audio tracks are synchronized with corresponding video frames.
 - * **Techniques and Tools:** Audio processing: *librosa* for audio manipulation, *pydub* for audio format conversion. Text processing:

nlTK or *spaCy* for tokenization and cleaning. Video processing: *ffmpeg* for frame extraction.

- **TTS Model Selection and Training:**

- **Objective:** Train a multi-speaker TTS model for speech generation.
- **Action:**
 - * Evaluate and select from models like *Tacotron 2*, *Tortoise*, or *XTTS* based on capabilities and research.
 - * Train the selected model using prepared Urdu audio and text data.
 - * Utilize techniques like teacher forcing and attention mechanisms to enhance training efficiency. **Techniques and Tools:** Deep Learning Frameworks: *TensorFlow* or *PyTorch* for model training. Pre-trained Weights: Use pre-trained weights for *Tacotron 2* or other models to expedite training.
 - * What we have discussed so far is shown in Figure [2.1](#)

- **Wav2Lip Model Training:**

- **Objective:** Train the *Wav2Lip* model for lip-syncing.
- **Action:**
 - * Prepare the dataset by segmenting videos into frames and ensuring audio is synchronized with the corresponding frame.
 - * Train the *Wav2Lip* model on the generated audio and segmented video frames to achieve accurate lip movement synchronization.
 - * **Techniques and Tools:** Wav2Lip model training. Dataset Creation as given is Section [2.4](#)
 - * What we have discussed so far is shown in Figure [2.2](#)

- **Model Analysis & Selection:**

- **Objective:** Evaluate and select the best-performing models for speech synthesis and lip-syncing.
- **Action:**
 - * Analyze the TTS and *Wav2Lip* models based on accuracy, naturalness of generated speech, and synchronization quality between audio and video.

- * **Techniques and Tools:** Evaluation Metrics: Use metrics such as *Mean Opinion Score (MOS)* for TTS quality and visual inspection for lip-sync accuracy.
- **Model Validation & Testing:**
 - **Objective:** Test the AI models to ensure accuracy and reliability in real-world scenarios.
 - **Action:**
 - * Conduct functional testing on both the TTS and *Wav2Lip* models, ensuring that the generated speech is clear and lip-syncing is accurate for various types of educational content.
 - *
- **App Development & Model Integration:**
 - **Objective:** Build the AI Guide platform and integrate both the trained TTS and *Wav2Lip* models.
 - **Action:**
 - * Develop the front-end of the mobile application using *React Native* or *Flutter*.
 - * Implement UI/UX design that allows users to upload lecture audio and select a speaker for cloning.
 - * Develop the back-end using *Python (Django/Flask)* for managing audio and video processing.
 - * Implement NLP techniques for extracting key information from text and integrate the trained TTS model to convert text into speech.
 - * Use the *Wav2Lip* model to sync the generated speech with the video of a virtual teacher, creating a seamless audio-visual experience.
 - * **Techniques and Tools:** Frontend Development: *React Native* or *Flutter* for cross-platform mobile app development. Backend Development: *Django* or *Flask* for server-side logic and audio/video management. NLP Libraries: *spaCy* or *NLTK* for information extraction.
- **Deployment:**
 - **Objective:** Deploy the AI Guide platform and ensure its accessibility for users.

- **Action:**

- * Deploy the platform to cloud servers, monitor its performance, and optimize based on user feedback and performance metrics, focusing on smooth text-to-speech generation and real-time lip-syncing features.
- * **Techniques and Tools:** Cloud Services: *AWS*, *Google Cloud*, *Azure* or any other platform for hosting and deployment. Monitoring Tools: Use monitoring tools (e.g., *Prometheus*, *Grafana*) for performance tracking and optimization.

This methodology involves both TTS and Wav2Lip models to generate clear speech and sync it with video, discussed in Figure 2.1 and Figure 2.2

2.4 Data Collection Techniques

The data collection approach will focus on sourcing audio and video lectures from instructors available on YouTube in Urdu. The process will involve identifying five speakers and collecting their audio recordings for training a multi-speaker TTS model. To ensure the clarity and relevance of the data, quality control measures will be implemented. All gathered data will be systematically organized for ease of access and documentation, adhering to ethical considerations to maintain compliance and integrity throughout the data gathering phase.

- **Audio data collected from YouTube for training Multi-speaker TTS:**

- **Source:** YouTube lectures in Urdu.
- **No. of Speakers:** 5.
- **Total No. of Hours per Speaker:** 4-5 hours.
- **Total No. of Sentences per Speaker:** Approximately 400 sentences.
- **Length of Each Sentence:** 2-12 seconds.
- **Audio Frequency:** 16kHz.
- **Loudness Normalization:** -16 LUFS (Loudness Units relative to Full Scale).

- **Video data collected from YouTube for training Lip-Syncing Model:**

-
- **Source:** YouTube lectures in Urdu.
 - **No. of Speakers:** 5.
 - **Total No. of Hours per Speaker:** 5-6 hours.
 - **Video Duration:** 2 seconds with corresponding audio.
 - **Frames per Second:** 25 frames.
 - **Audio Frequency:** 16kHz.
 - **No. of Sentences per Speaker:** Approximately 600 to 1,680 sentences.

2.5 Experimental Design

The Multi-Speaker TTS generate audios through the following steps:

- **Data Preparation**

- **Dataset Creation:**

- * Prepare an Urdu multi-speaker dataset comprising audio recordings and corresponding transcriptions for each speaker.
 - * Ensure diverse representation of speakers in terms of gender, accent, and dialect.

- **Data Format:**

- * Use a structured format where each speaker has a dedicated folder containing audio files and a text file with transcriptions.

- **Tools:**

- * Python for data manipulation and organization.
 - * Libraries such as pandas and numpy for handling datasets.

- **Data Preprocessing**

- **Audio Processing:**

- * Normalize audio levels and ensure consistent sampling rates (e.g., 16 kHz) across all audio files.
 - * Convert audio files to a suitable format (e.g., WAV) for training.

- **Text Processing:**

- * Clean and standardize text transcriptions, ensuring proper tokenization.
 - * Use phoneme or grapheme conversion for better pronunciation modeling.

- **Tools:**

- * librosa for audio processing.
 - * nltk or spaCy for text processing.

- **TTS Model Selection and Training**

- **Model Choice:**

- * Evaluate and select from Tacotron 2, Tortoise, or XTTS based on initial research and capabilities.

- **Training the Model:**
 - * Train the selected model using the prepared audio and text data.
 - * Utilize techniques such as teacher forcing and attention mechanisms to enhance training efficiency.
- **Tools:**
 - * TensorFlow or PyTorch for model training.
 - * Pre-trained weights for Tacotron 2 or other models to speed up training.
- **Model Testing**
 - **Testing Process:**
 - * Test the trained model on unseen text inputs to evaluate its performance.
 - * Use techniques such as:
 - **Objective Evaluation:** Calculate metrics like Mel cepstral distortion (MCD) to quantify audio quality.
 - **Subjective Evaluation:** Conduct listening tests with a group of evaluators to assess speech naturalness and intelligibility.
 - **Tools:**
 - * Custom scripts for generating predictions and calculating metrics.
- **Accuracy Assessment**
 - **Accuracy Check:**
 - * Analyze the generated speech for accuracy in terms of pronunciation, intonation, and overall naturalness.
 - **Issue Identification:**
 - * If discrepancies are found, identify issues related to model training, data quality, or feature extraction.
 - **Tools:**
 - * Visual tools such as spectrograms for detailed audio analysis.
- **Model Fine-Tuning**
 - **Fine-Tuning Process:**
 - * Based on the analysis, fine-tune the model by adjusting hyperparameters or retraining with additional data.

- * Implement techniques like data augmentation to improve model robustness.
- **Tools:**
 - * Use frameworks like Ray Tune for hyperparameter tuning.
- **Benchmarking**
 - **Performance Comparison:**
 - * Compare the three models (Tacotron 2, Tortoise, XTTS) on various performance metrics.
 - **Performance Measures:**
 - * Use measures such as:
 - Naturalness (MOS - Mean Opinion Score)
 - Intelligibility (Word Error Rate)
 - Speed (Real-Time Factor)
 - **Analysis:**
 - * Analyze the results to identify the model that yields the best performance.
- **Finalization**
 - **Final Model Selection:**
 - * Based on the benchmarking results, select the model that offers the best balance of performance metrics.
 - **Deployment:**
 - * Prepare the final model for deployment in applications, ensuring compatibility with user interfaces or APIs.
 - * This experimental design is explain in [Figure 2.3](#)

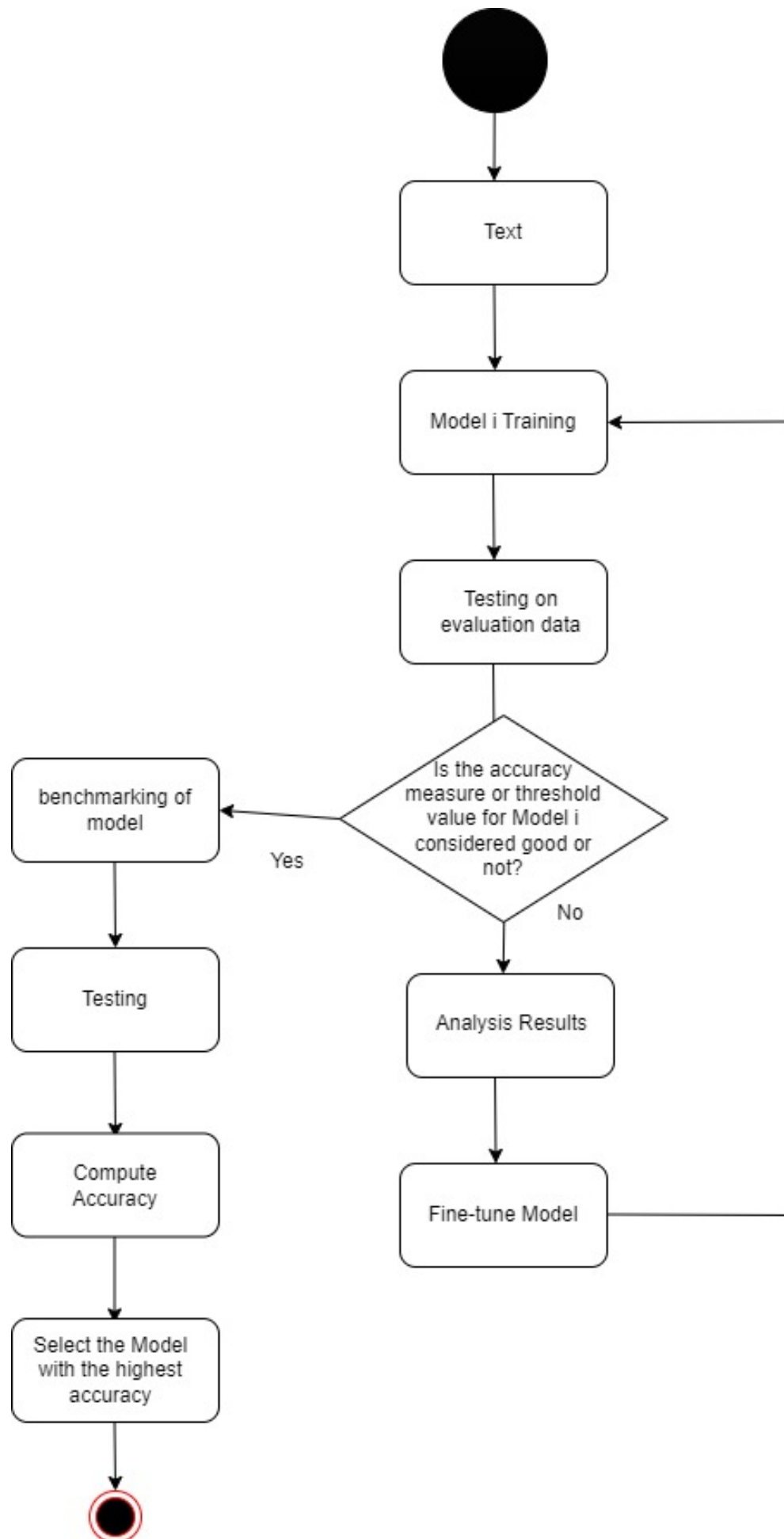


FIGURE 2.3: Model Training For Multi-Speaker TTS

The Wav2Lip system synchronizes lip movements with audio through the following steps:

- **Data Preparation**

- **Data Format:**

- * Structure the dataset with videos and their corresponding audio files. Extract frames from videos to use as input for lip-sync training.

- **Tools:**

- * Python for data manipulation and organization.
 - * Libraries like `ffmpeg` for frame extraction and video handling, `moviepy` for video editing.

- **Data Preprocessing**

- **Audio Processing:**

- * Normalize audio levels and ensure consistent sampling rates (e.g., 16 kHz) for all audio files.
 - * Convert audio files to a suitable format (e.g., WAV) and ensure proper synchronization with video frames.

- **Video Processing:**

- * Extract frames from video data at a specific rate (e.g., 25 frames per second) to create a dataset for training.
 - * Align each frame with the corresponding audio segment to maintain synchronization.

- **Tools:**

- * `librosa` for audio normalization and processing.
 - * `ffmpeg` for frame extraction from video files.

- **Model Selection and Training**

- **Model Choice:**

- * Select the Wav2Lip model, which specializes in syncing lip movements with audio.

- **Training the Model:**

- * Train the Wav2Lip model using the paired video frames and corresponding audio clips to learn accurate lip movements.
 - * Use techniques like attention mechanisms to improve the lip-sync accuracy.
- **Tools:**
 - * PyTorch for model training.
 - * Pre-trained weights for Wav2Lip to expedite the training process and build on prior learning.
- **Model Testing**
 - **Testing Process:**
 - * Test the trained Wav2Lip model using unseen video and audio inputs to evaluate its lip-sync accuracy and performance.
 - **Evaluation Techniques:**
 - * **Objective Evaluation:** Use metrics like Lip Sync Error (LSE) to measure how well lip movements align with the audio.
 - * **Subjective Evaluation:** Conduct visual assessments with evaluators to ensure the lip movements appear natural and synchronized with the speech.
 - **Tools:**
 - * Custom scripts for generating predictions and calculating metrics.
 - * Visual inspection tools to check the quality of lip-sync in generated videos.
- **Accuracy Assessment**
 - **Accuracy Check:**
 - * Analyze the model's ability to accurately sync lips with the generated speech in various scenarios, including different accents and genders.
 - **Issue Identification:**
 - * If issues arise, such as desynchronized lips or incorrect lip shapes, identify potential problems in model training, data quality, or synchronization mechanisms.
 - **Tools:**

- * Use visualization tools like spectrograms and lip-sync graphs to assess timing and audio-visual alignment.

- **Model Fine-Tuning**

- **Fine-Tuning Process:**

- * Based on the analysis, fine-tune the model by adjusting hyperparameters, adding more diverse data, or using advanced augmentation techniques to enhance performance.

- **Techniques:**

- * Use data augmentation techniques, such as adding noise to audio or applying video transformation, to make the model more robust.

- **Tools:**

- * Libraries like `imgaug` for video frame augmentation.
 - * Ray Tune for hyperparameter optimization.

- **Benchmarking**

- **Performance Comparison:**

- * Compare the Wav2Lip model with other lip-sync techniques to assess overall accuracy and efficiency.

- **Performance Measures:**

- * **Lip Sync Error (LSE):** Measures synchronization accuracy.
 - * **Naturalness (MOS):** Evaluates how natural the synchronized video appears.
 - * **Real-Time Performance:** Measures how quickly the model can generate results.

- **Analysis:**

- * Analyze the performance of Wav2Lip to ensure it meets the desired accuracy and speed for real-time applications.

- **Finalization**

- **Final Model Selection:**

- * Select the final Wav2Lip model based on the benchmarking results, ensuring it offers the best lip-sync accuracy, speed, and naturalness.

- **Deployment:**

- * Prepare the Wav2Lip model for integration into applications, ensuring compatibility with user interfaces and real-time video generation pipelines.
 - * This experimental design is explain in [Figure 2.4](#)

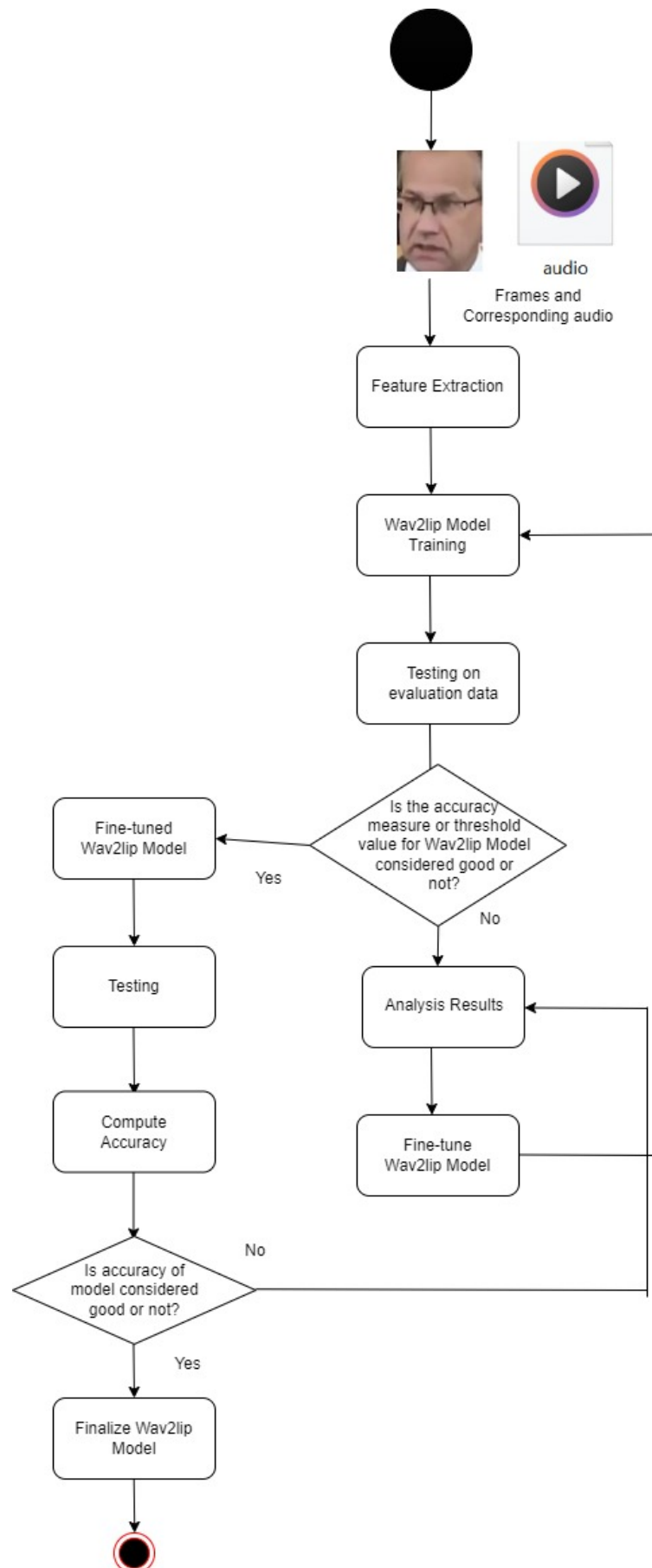


FIGURE 2.4: Model Training For Lip-Synching Model

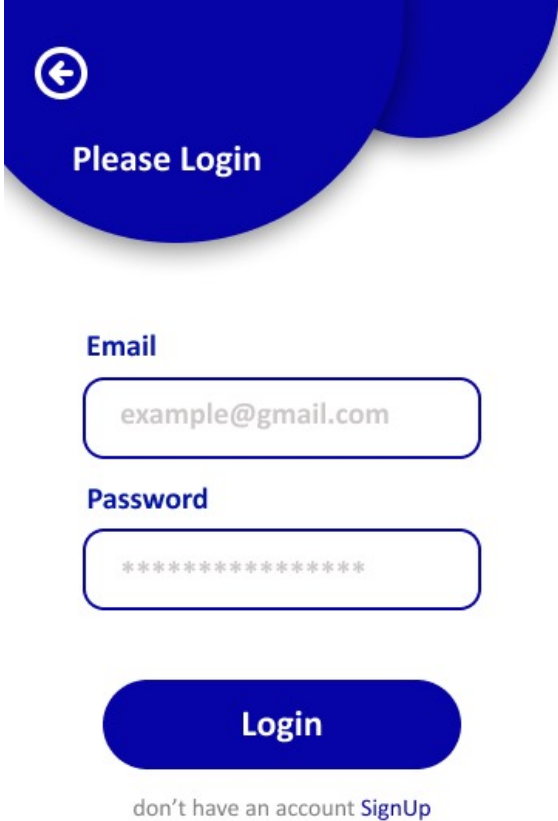
2.6 App Design

We have designed our application in Figma. Our application will consist of these pages. Here are the images of our application design below;



FIGURE 2.5: Home Page

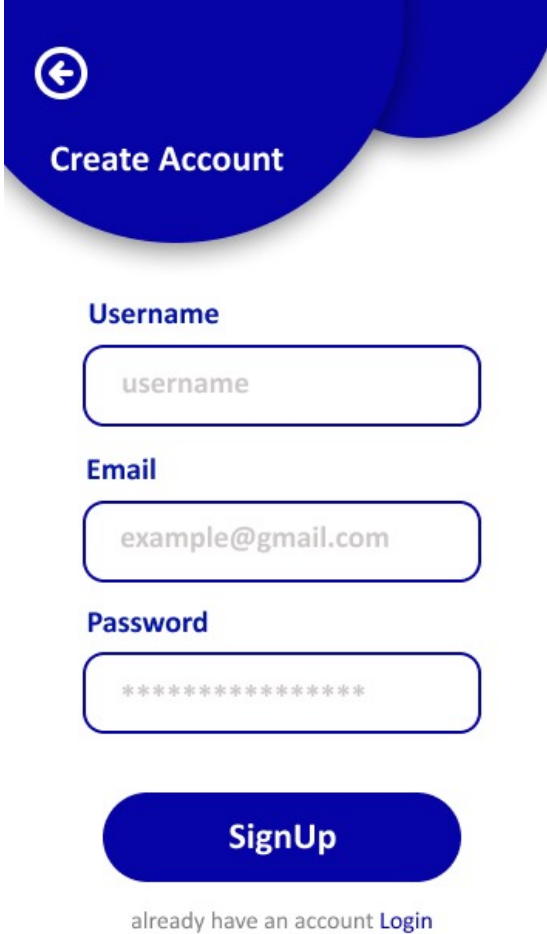
This is the home page of the AI Guide application. It welcomes users and invites them to input their text-based lectures. Users can then choose a teacher to bring those lectures to life through engaging audio and video presentations



The login page features a blue header with a white back arrow icon and the text "Please Login". Below the header, there are two input fields: "Email" with the placeholder "example@gmail.com" and "Password" with a masked password "*****". A blue "Login" button is positioned below the password field. At the bottom, there is a link that reads "don't have an account [SignUp](#)".

FIGURE 2.6: Login Page

This is the login page of the AI Guide App. It requires users to enter their email address and password to access their account. If users do not have an account, they can create one by clicking on the "SignUp". This page allows existing users to log in to their account and continue their learning journey within the AI Guide App.



The image shows a mobile app sign-up screen. At the top, there is a blue header with a white back arrow icon and the text "Create Account". Below the header, there are three input fields: "Username" with the placeholder "username", "Email" with the placeholder "example@gmail.com", and "Password" with a masked placeholder "*****". Below these fields is a large blue "SignUp" button. At the bottom, there is a link that says "already have an account Login".

FIGURE 2.7: Sign-Up Page

This is the sign-up page of the AI Guide App. It allows new users to create an account by providing their username, email address, and password. After filling in the required information, users can click on the "SignUp" button to complete the registration process. If users already have an account, they can log in directly by clicking on the "Login".



FIGURE 2.8: Select Language Page

This is the language selection page of the AI Guide application. It allows users to choose the language of their input text. The available options are English and Urdu. This page is crucial for ensuring that the application can process and understand the user's input text accurately. By selecting the appropriate language, users can effectively utilize the AI Guide's features and receive tailored assistance. Urdu.

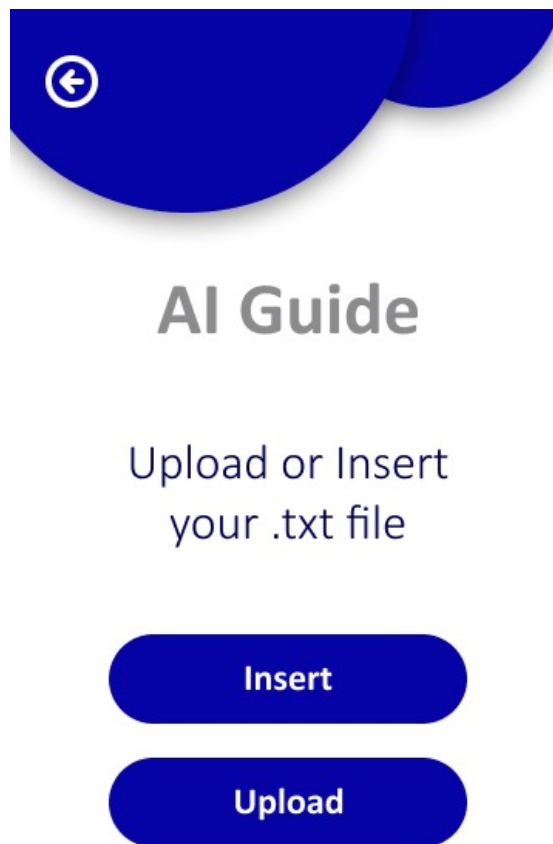


FIGURE 2.9: Upload Insert Page

This is the upload/insert page of the AI Guide application. It allows users to input their text-based lectures by either inserting the text directly into the application or uploading a .txt file. The "Insert" button is for manually typing or pasting the text, while the "Upload" button is for selecting a pre-saved .txt file from the user's device.

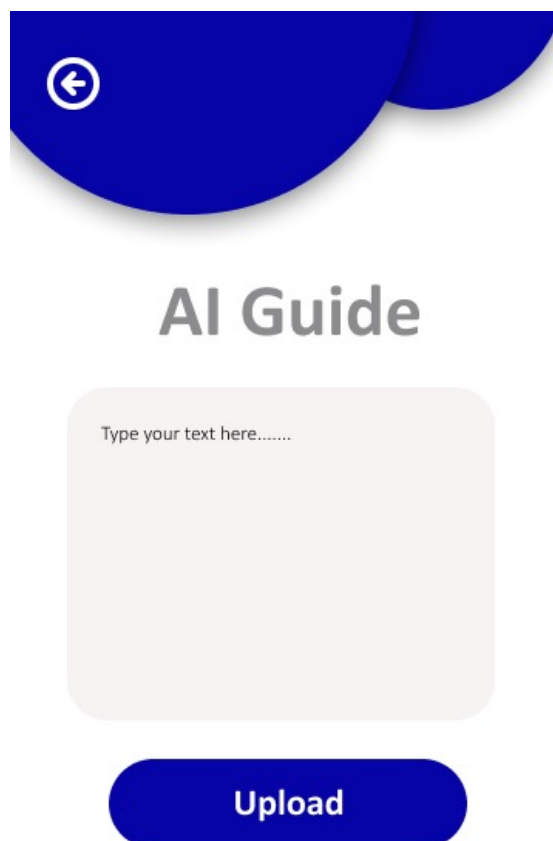


FIGURE 2.10: Upload Text

This is the text input page of the AI Guide application. It provides a text box where users can directly type or paste their lecture content. Alternatively, users can still choose to upload a .txt file by clicking the "Upload" button.

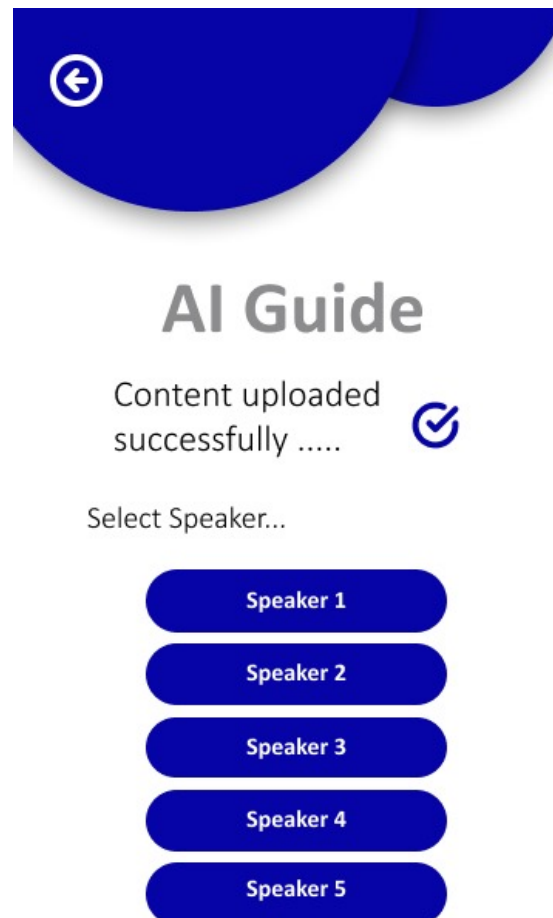


FIGURE 2.11: Select Speaker Page

This is the speaker selection page of the AI Guide application. It appears after the user has successfully uploaded their content. The page displays a confirmation message indicating that the content upload was successful and then prompts the user to select a speaker for their presentation. The user is presented with a list of five pre-defined speakers. By selecting a speaker, the user chooses the voice and style that will be used to narrate their content in the generated audio and video presentations.

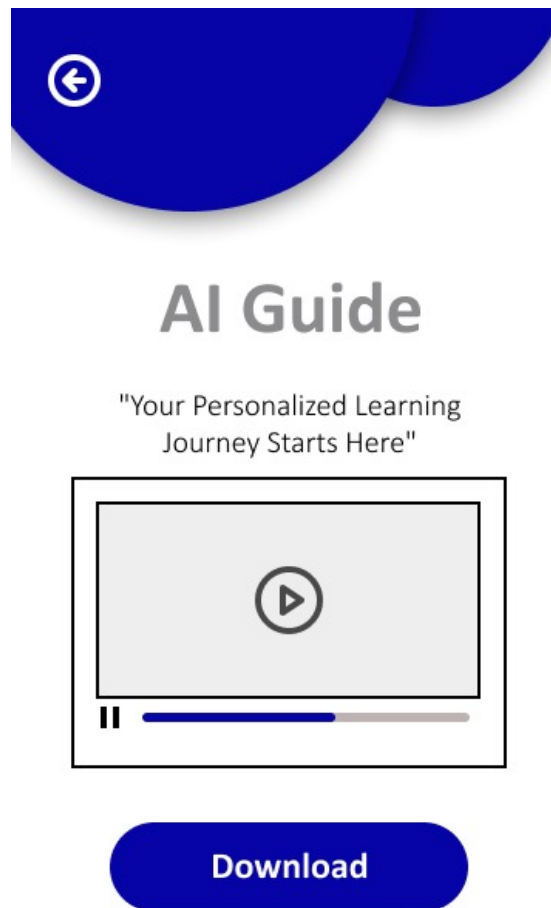


FIGURE 2.12: Download Page

This is the download page of the AI Guide application. A video preview with a play button is included, allowing users to preview the generated presentation before downloading it. The "Download" button enables users to save the final presentation to their device for offline access or further sharing.

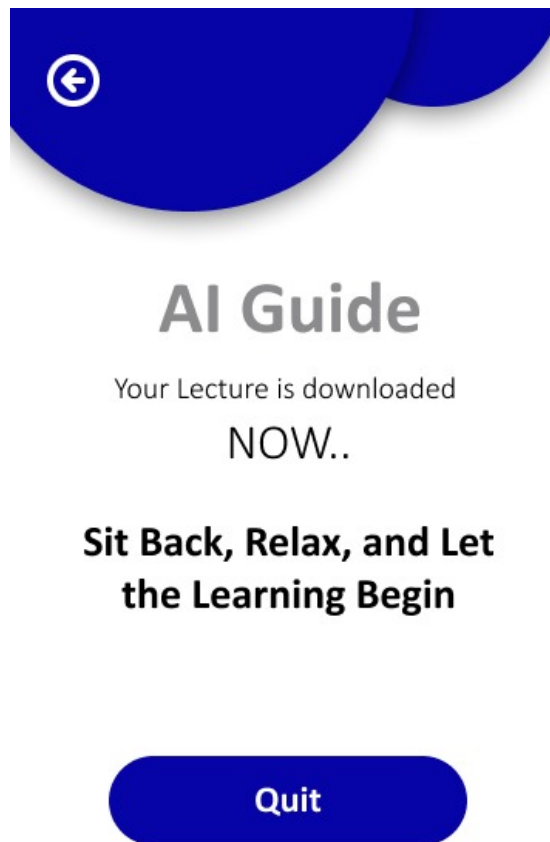


FIGURE 2.13: Downloading Progress Page

This is the downloading progress page of the AI Guide application. It displays a message informing the user that their lecture has been successfully downloaded. A "Quit" button is provided to allow the user to exit the application. This page indicates the successful completion of the download process.

References

- [1] Chen et al. *Neural Voice Cloning with a Few Samples*. last accessed on September 20, 2024.
- [2] Chung et al. *Real-Time Talking Face Generation from Still Images and Audio*. 18 July 2017, last accessed on September 21, 2024.
- [3] Dhanush et al. *LIP-TO-SPEECH SYNTHESIS USING MACHINE LEARNING*. last accessed on September 21, 2024.
- [4] Goyal¹ et al. *Emotionally Enhanced Talking Face Generation*. ¹ IIT, Roorkee, India ² IIIT, Delhi, India ³ Carnegie Mellon University, USA ⁴ National Institute of Informatics, Japan ⁵ A*STAR, Singapore, last accessed on September 20, 2024.
- [5] Kadam et al. *ReVoice: A Neural Network based Voice Cloning System*. 2024 IEEE 9th International Conference for Convergence in Technology (I2CT) Pune, India. Apr 5-7, 2024, last accessed on September 20, 2024.
- [6] Khalid et al. *FakeAVCeleb: A Novel Audio-Video Multimodal Deepfake Dataset*. Department of Computer Science and Engineering, Sungkyunkwan University, South Korea ² Department of Applied Data Science, Sungkyunkwan University, South Korea ³ Department of Artificial Intelligence, Sungkyunkwan University, South Korea hasam.khalid, shahroz, kimminha, swoo@g.skku.edu, last accessed on September 22, 2024.
- [7] Naik et al. *Deep Learning Models for Natural Voice Cloning Methods*. last accessed on September 20, 2024.
- [8] Nawaz et al. *Deepfakes generation and detection: state-of-the-art, open challenges, countermeasures, and way forward*. Published in 2022, last accessed on September 22, 2024.
- [9] Seong et al. *Multilingual Speech Synthesis for Voice Cloning*. last accessed on September 20, 2024.

-
- [10] Wang et al. *StyleSync: High-Fidelity Generalized and Personalized Lip Sync in Style-based Generator*. last accessed on September 21, 2024.
 - [11] Zhang et al. *Transfer Learning from Speaker Verification to Multispeaker Text-To-Speech Synthesis*. last accessed on September 20, 2024.
 - [12] Zhang¹ et al. *Speech Enhancement with Lip Perceptual Loss*. ¹ Beijing University of Posts and Telecommunications, Beijing, China ² Zuoyebang Education Technology (Beijing) Co., LTD, Beijing, China fxfeng@bupt.edu.cn, last accessed on September 21, 2024.
 - [13] Amirreza Hashemi. *Evaluation of HuggingFace Tools for Video In Painting and Voice Cloning*. Amirreza Hashemi November 24, 2023, last accessed on September 22, 2024.