VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi-590018

A PROJECT REPORT

On

"AI-Generated Digital Human for Virtual Interactions"

A Dissertation Submitted in partial fulfillment of the requirement for the degree of

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE AND ENGINEERING

Submitted by

C SHALINI	**(1RG22CS025)**
BHANU PRIYA K	**(1RG22CS021)**
FAIZA SIDDIQUE	**(1RG22CS031)**
MADHU KUMAR A	**(1RG22CS046)**

Under The Guidance of

Mrs. Soniya Komal V

Assistant Professor, Dept. of CSE

RGIT, Bengaluru-32

Department of Computer Science & Engineering

RAJIV GANDHI INSTITUTE OF TECHNOLOGY

Cholanagar, R. T. Nagar Post, Bengaluru-560032

2025-2026

RAJIV GANDHI INSTITUTE OF TECHNOLOGY

(Affiliated to Visvesvaraya Technological University)

Cholanagar, R.T. Nagar Post, Bengaluru-560032

Department of Computer Science & Engineering

CERTIFICATE

Phase- II

This is to certify that the Project Report entitled **"AI-Generated Digital Human for Virtual Interactions"** is a Bonafide work carried out by **Ms. C SHALINI (1RG22CS025) , Ms. BHANU PRIYA K (1RG22CS021), Ms. FAIZA SIDDIQUE (1RG22CS031), Mr. MADHU KUMAR A (1RG22CS046)** in partial fulfillment for the award of **Bachelor of Engineering** in **Computer Science and Engineering** under **Visvesvaraya Technological University, Belagavi,** during the year **2025-2026.** It is certified that all corrections/suggestions given for Internal Assessment have been incorporated in the report. This Project report has been approved as it satisfies the academic requirements in respect of project work (BCS786) prescribed for the said degree.

Signature of guide	Signature of HOD	Signature of Principal
Mrs. Soniya Komal V	**Dr. Arudra A**	**Dr. D G Anand**
Assistant Professor	Associate Professor, HOD	Principal, RGIT,
Dept. of CSE	Dept. of CSE	Benagluru-32
RGIT, Bengaluru-32	RGIT, Bengaluru-32	

External Viva

Name of the Examiners	**Signature with date**
1.	
2.	

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi-590018

RAJIV GANDHI INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DECLARATION

We hereby declare that the project work entitled **" AI-Generated Digital Human for Virtual Interactions"** submitted to the **Visvesvaraya Technological University, Belagavi** during the academic year **2025-2026**, is record of an original work done by us under the guidance of **Mrs. Soniya Komal V,** Assistant professor, Rajiv Gandhi Institute of Technology , Bengaluru and this project work is submitted in the partial fulfillment of requirements for the award of the degree of **Bachelor of Engineering** in **Computer Science & Engineering.** The results embodied in this project have not been submitted to any other University or Institute for award of any degree or diploma.

C SHALINI	**(1RG22CS025)**
BHANU PRIYA K	**(1RG22CS021)**
FAIZA SIDDIQUE	**(1RG22CS031)**
MADHU KUMAR A	**(1RG22CS046)**

ACKNOWLEDGEMENT

We take this opportunity to express our sincere gratitude and respect to the **Rajiv Gandhi Institute of Technology, Bengaluru** for providing us an opportunity to carry out our project work. We express our sincere regards and thanks to **Dr. D G ANAND,** Principal, RGIT, Bengaluru. Also we would like to thanks **Dr. ARUDRA A**, Associate Professor and HOD, Department of Computer Science & Engineering, RGIT, Bengaluru, for their encouragement and support throughout the Project.

With profound sense of gratitude, we acknowledge the guidance and support extended towards us by project coordinators **Dr. LATHA P H**, Professor, Department of CSE, RGIT, Bengaluru. **Mrs. BHAGYASHRI WAKDE**, Assistant Professor, Department of CSE, RGIT, Bengaluru. Their incessant encouragement and valuable technical support have been of immense help in realizing this project. Their guidance gave us the environment to enhance our knowledge, skills and to reach the pinnacle with sheer determination, dedication and hard work.

We extend our heartfelt gratitude to **Mrs. SONIYA KOMAL V,** Assistant professor, Department of Computer Science & Engineering, RGIT, Bengaluru, for their unwavering support and guidance in our project. Their expertise has been instrumental in shaping its success. We also appreciate the collaborative efforts of other coordinators and team members for enriching this learning experience.

We also extend our thanks to the entire faculty of the Department of CSE, RGIT, Bengaluru, who have encouraged throughout the course of Bachelor Degree.

We extend our heartfelt thanks to our family for their full support and unwavering encouragement, which provided us with a safe and conducive environment for the completion of this project. Their assistance and belief in our abilities have been instrumental in this achievement.

C SHALINI **(1RG22CS025)**

BHANU PRIYA K **(1RG22CS021)**

FAIZA SIDDIQUE **(1RG22CS031)**

MADHU KUMAR A **(1RG22CS046)**

ABSTRACT

In today's digitally connected world, virtual communication has become essential across education, healthcare, customer service, and social interaction. However, traditional digital platforms often lack the emotional intelligence, realism, and accessibility required to create truly engaging and inclusive experiences. This project proposes the development of an AI-Generated Digital Human—a lifelike 3D avatar designed to interact with users in a natural, empathetic, and intelligent manner.

The digital human leverages advanced technologies including Speech-to-Text (STT), ChatGPT-based natural language processing, Text-to-Speech (TTS), facial expression synthesis, gesture animation, and computer vision to enable multimodal interaction. It supports real-time captioning for deaf users, object detection with voice narration for visually impaired individuals, and multilingual communication to ensure global accessibility.

Designed as a flexible and adaptive platform, this system can be applied across various domains—such as virtual classrooms, therapy sessions, elder care, smart homes, and digital companionship. By enhancing virtual interactions with emotional depth, contextual understanding, and inclusive features, the AI-generated digital human sets a new standard for how humans engage with technology in immersive digital environments.

CONTENTS

LIST OF FIGURES/ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Overview

The rapid growth of artificial intelligence has brought significant advancements in human-computer interaction, yet most existing systems still fall short of delivering natural, emotionally rich, and inclusive communication experiences. Traditional chatbots rely heavily on text and lack visual or emotional expressiveness, while VR avatars often appear static and are unable to adapt dynamically to user inputs or real-world environments. This limitation creates a communication gap, particularly for people who require accessible solutions, such as the visually impaired, hearing impaired, or elderly users.

The project addresses this challenge by introducing AI-generated digital humans intelligent, lifelike avatars capable of engaging in conversations that feel human-like and emotionally responsive. These digital humans go beyond the functionality of current chatbots or avatars by integrating multimodal inputs and outputs. They can understand and respond through voice, text, gestures, and camera vision, while also expressing themselves through speech, facial expressions, lip-synced gestures, and real-time captions. This combination ensures that interactions are not only more realistic but also empathetic and inclusive.

At the core of this system lies the fusion of AI, natural language processing, computer vision, and 3D avatar rendering. Natural Language Processing (NLP) allows the system to comprehend user input and generate contextually appropriate responses. Computer vision enables the avatar to interpret real-world environments, detect objects, recognize faces, and even analyze gestures. Meanwhile, advanced speech technologies including speech-to-text (STT) and text-to-speech (TTS) support both natural communication and accessibility. Finally, 3D rendering techniques make it possible to present an avatar that looks and behaves like a real human, capable of showing emotions through facial expressions, tone, and gestures.

A unique and transformative feature of this system is its camera-based guidance capability. By leveraging the smartphone or device camera, the avatar can scan and interpret the user's surroundings in real time. This feature is particularly valuable for blind or visually impaired users, as the avatar can describe objects, read text, and provide navigation cues such as identifying obstacles or recognizing people.

Equally important is the system's accessibility-first approach. While digital human avatars

have often been designed for entertainment, business, or education, this project prioritizes inclusivity. The avatars offer live captions for deaf users, narrations for blind users, and simplified natural speech interactions for elderly individuals. This ensures that digital humans are not merely a novelty but a practical assistive technology that improves quality of life.

Another advantage of the proposed system is its lightweight and versatile design. Unlike many VR or AR solutions that require heavy equipment and expensive setups, this system is designed to work across multiple platforms — including smartphones, browsers, and VR devices. This cross-platform accessibility ensures that the technology can reach a wider audience and adapt to different use cases without requiring specialized hardware. In essence, the project represents a significant step toward bridging the gap between virtual communication and real-world assistance. By merging intelligence, empathy, and accessibility, the digital human becomes not just a communication partner but also a guide, assistant, and companion. This makes it relevant for a wide range of domains, including education, healthcare, professional collaboration, and everyday personal support.

Looking forward, this foundation opens the door to even more advanced features, such as full-body avatar interactions, outdoor navigation, augmented reality integration, and multilingual emotional communication. These future directions highlight the scalability and potential of the system to evolve into a truly universal digital companion.

In summary, the overview of this project highlights the development of AI-generated digital humans that are natural, expressive, and inclusive. By combining multimodal interaction, emotional intelligence, and accessibility-first features, the system goes beyond existing technologies to create a solution that is both innovative and socially impactful. It redefines how humans can connect with digital systems — not as mechanical tools, but as empathetic, intelligent partners in communication and everyday life.

1.1.1 Outdoor cases

The outdoor environment presents unique challenges for individuals, particularly those with disabilities such as blindness, low vision, or mobility issues. Unlike controlled indoor spaces, outdoor areas are dynamic, unpredictable, and filled with moving elements such as vehicles, pedestrians, and changing traffic signals. Navigating such spaces requires real-time perception, decision-making, and communication. The proposed AI-generated digital humans extend their functionality into these outdoor scenarios, providing critical assistance and enhancing safety, independence, and social interaction.

- **Navigation and Mobility Support**

  One of the most significant outdoor applications of digital humans is assisting visually impaired users with navigation. By integrating GPS, camera vision, and computer vision algorithms, the system can identify pathways, crosswalks, and potential obstacles. For example, if a user approaches a busy road, the avatar can provide spoken guidance such as: “There is a pedestrian crossing ahead. The traffic light is currently red; wait before crossing.” This real-time narration transforms the outdoor experience into a safer and more independent journey.

- **Recognition of Objects and Signs**

  Outdoor environments contain important information embedded in public signs, billboards, or digital boards. For blind or visually impaired individuals, accessing this information is nearly impossible without external help. The camera-based guidance system enables the avatar to read and narrate such visual cues in real time. For example, the avatar could say: “You are approaching a bus stop. The next bus is scheduled to arrive in five minutes.” This capability is also useful for navigating complex public spaces like airports, railway stations, and shopping areas, where large signboards play a crucial role. By narrating this information, the digital human reduces dependence on others and empowers users to navigate independently.

- **Social Interaction and Assistance**

  Outdoor interactions often involve engaging with strangers, acquaintances, or service providers. For visually impaired individuals, recognizing social cues such as facial expressions or gestures can be challenging. The digital human, using camera input and emotion recognition, can assist by describing non-verbal cues. For instance, it might narrate: “The person in front of you is smiling and extending their hand.” This enables smoother communication and reduces social isolation. In addition, the avatar can serve as a real-time interpreter in multilingual outdoor settings. By reading signs or understanding conversations in a foreign language, it can provide translations and context.

- **Safety Alerts and Emergency Assistance**

  Safety is a crucial factor outdoors, particularly in busy urban environments. The digital human system can detect hazards such as approaching vehicles, construction zones,

or uneven pathways and issue timely alerts. For example, it could warn: "Caution, a vehicle is approaching from your left side." Such proactive alerts can prevent accidents and improve the overall confidence of blind or elderly users in navigating outdoor spaces. In emergency situations, the avatar could also function as a safety companion by contacting emergency services or sharing the user's live location with a trusted contact. This layer of support ensures that users are not left vulnerable when unexpected situations occur.

- **Integration with Smart Infrastructure**

  As cities move toward becoming "smart cities," integration with digital infrastructure becomes possible. The digital human can interface with smart traffic systems, IoT-enabled buses, and public transport schedules to provide real-time updates. For example, when a user approaches a smart crosswalk, the avatar could narrate the signal changes in sync with the city's traffic management system. This synergy between AI avatars and urban infrastructure could redefine outdoor accessibility.

1.1.2 Indoor cases

Indoor environments such as homes, workplaces, classrooms, and healthcare facilities provide structured spaces where AI-generated digital humans can act as assistants, companions, and accessibility enablers. Unlike outdoor spaces, indoor settings are often less dynamic but involve tasks that demand personalization, communication, and accessibility support. The proposed system utilizes its multimodal abilities — voice, text, gestures, and camera input — to provide practical assistance indoors.

- **Environment Narration for Blind Users**

One of the most impactful indoor applications is scene narration. Using a smartphone or wearable camera, the avatar can scan the immediate surroundings and describe objects, furniture, or even people in the room. For example, it could narrate: "There is a chair on your left, a table in front, and a person standing near the door." This empowers blind or visually impaired individuals to orient themselves in new or unfamiliar indoor spaces, such as offices, classrooms, or hospital rooms.The system can also recognize printed or digital text, enabling the avatar to read labels, books, signs, or even information displayed on screens. This extends to tasks like reading medicine instructions, identifying packaged items in a kitchen, or narrating content from a notice board.

- **Assistance for Deaf and Hearing-Impaired Users**

For individuals who are deaf or hard of hearing, communication in indoor environments can be particularly challenging, especially in group conversations or meetings. The avatar addresses this challenge through real-time speech-to-text captioning. As conversations occur, the avatar transcribes spoken words into text that can be displayed on the user's device. Additionally, the system can be configured to provide alerts for important sounds. For example, it could notify the user when the doorbell rings, when someone calls their name, or when an alarm goes off. This transforms the avatar into an inclusive communication bridge, ensuring that deaf users are not left out of important interactions or signals indoors.

- **Support for Elderly Users**

Elderly individuals often require assistance in managing daily activities, remembering schedules, or simply engaging in meaningful conversation. Indoors, the digital human can act as a personal assistant and companion. It can provide reminders for medication, doctor's appointments, or household tasks. Its natural conversational ability also enables it to serve as a social companion, reducing feelings of loneliness or isolation that are common among the elderly. By combining speech recognition and context awareness, the system ensures that elderly users can interact using natural language rather than relying on complex interfaces. This makes technology more approachable and user-friendly for older adults.

- **Education and Learning Environments**

In classrooms or online learning spaces, digital humans can function as interactive tutors. They can explain lessons, display real-time captions for deaf students, and narrate visual content for blind students. Their emotional and expressive capabilities make the learning process more engaging, as the avatars can convey enthusiasm, empathy, or encouragement, just like a real teacher would.

- **Healthcare and Professional Settings**

In hospitals, clinics, or workplaces, digital humans can provide real-time assistance and communication support. For example, in healthcare settings, they can narrate instructions, read patient records aloud, or assist patients with visual or hearing impairments.

1.2 Motivation

In the current digital era, communication technologies such as chatbots, video conferencing tools, and virtual avatars are widely used for interaction, education, healthcare, and entertainment. However, most of these systems lack the ability to provide natural, emotionally engaging, and inclusive communication. While they may support basic interactions, they fail to replicate the depth of human-to-human communication, where emotions, gestures, and context play a critical role.

Furthermore, accessibility remains a major challenge. Individuals with disabilities such as blindness, deafness, or age-related impairments are often excluded from effective use of virtual systems. For example, blind users cannot benefit from visual interfaces, while deaf users struggle with spoken communication. There is a growing need for a system that not only provides realistic digital human interactions but also prioritizes inclusivity and accessibility.

The motivation behind this project is to design AI-generated digital humans that combine lifelike communication with assistive features. These avatars are envisioned as empathetic, context-aware companions that support natural conversations and extend assistance to those who face communication or navigation barriers in both indoor and outdoor environments.

1.3 Problem Identification

The Despite rapid advancements in AI, several challenges remain in the domain of digital human ,Most systems only accept text or voice input, ignoring visual cues such as gestures, facial expressions, or camera-based environmental inputs. Blind users cannot access visual information from their surroundings. Deaf users lack real-time captioning in virtual environments. Elderly users often find current systems complex and unintuitive. Existing Technologies like speech-to-text tools, TTS systems, and VR avatars exist independently, but they are not integrated into a single, comprehensive system. Current avatars are unable to adapt to real-world environments — for example, identifying a chair in a room or reading a street sign outdoors. The project of violence detection using Movement and surveillance cameras involves selecting and installing appropriate cameras, developing software to analyze the video footage, testing and optimizing the software, deploying it in public spaces, and integrating it with existing security systems. The goal of the project is to improve public safety by enabling security personnel to quickly identify and respond to violent behavior, potentially preventing serious harm to individuals in public spaces.

1.4 Scope of the project

The scope of this project is to design and develop an AI-driven digital human system that Provides lifelike communication through speech, facial expressions, and gestures. Accepts multimodal input (voice, text, gestures, and camera-based environmental scanning).

Enhances accessibility by:

- Narrating surroundings for blind users.
- Providing real-time captions for deaf users.
- Offering simplified voice interaction for elderly users.
- Functions across both indoor and outdoor environments.
- Runs on lightweight platforms such as smartphones, browsers, or VR without requiring complex hardware.

  The project focuses on bridging the gap between virtual human interaction and real-world accessibility, making the technology useful in homes, workplaces, educational institutions, hospitals.

1.5 Objective and Methodology

The main objective of the project is to design and develop an AI-generated digital human that delivers natural, human-like interactions while ensuring inclusivity and accessibility. The system aims to integrate speech recognition, natural language processing, computer vision, and 3D avatar rendering into a single framework capable of supporting multilingual, emotionally adaptive, and context-aware conversations. Another key objective is to provide dedicated accessibility features such as real-time captions for deaf users and environment narration for visually impaired users, thereby creating a socially responsible and universally usable assistant.

The methodology involves a structured approach combining artificial intelligence models, computer vision algorithms, and 3D rendering techniques. A Flask-based web application serves as the interface for user interaction. The Phi model, deployed locally via Ollama, powers dialogue understanding and generation, while speech-to-text and text-to-speech modules enable seamless voice communication. YOLOv8 is integrated for object detection and environment narration, enhancing accessibility for visually impaired users. The avatar is designed using ReadyPlayerMe to provide realistic facial animations, lip-sync, and emotional expressions.

1.6 Existing System

Currently available systems for virtual interaction, such as chatbots, video conferencing platforms, and VR avatars, are limited in their ability to provide natural, emotional, and inclusive communication. Chatbots are mostly text-based and lack facial expressions or visual presence, making them feel robotic. Video conferencing allows real human interaction but does not integrate avatars or AI-driven personalization. VR avatars offer visual representations but lack speech, gestures, or adaptive emotional responses, reducing their realism. Similarly, speech-to-text and TTS tools work independently but are not embedded into immersive 3D environments. These fragmented systems fail to combine vision, speech, gesture recognition, and emotional intelligence into one unified avatar.

Disadvantages:

- **Lack of Emotional Realism:** These systems suffer from a lack of emotional realism. This is because they do not include facial emotions like smiling, frowning, or maintaining eye contact. Furthermore, there is an absence of gestures and body language, which are crucial for natural communication, and no lip-syncing, leading to unrealistic avatar speech.
- **Limited Interaction Capabilities:** They also have limited interaction capabilities. Current systems cannot process real-world visual inputs, such as scanning the surroundings with a camera. They fail to recognize or react to the facial expressions or gestures of the user, making the interaction feel one-dimensional, often restricted to text-only or audio-only.
- **Poor Speech Quality:** Poor speech quality is another major issue. The speech output often feels robotic, monotone, or unnatural. These systems are prone to errors in pronunciation, tone, or emotion, which ultimately reduces user trust and engagement.
- **Accessibility Gaps:** Significant accessibility gaps exist. There are no real-time captions for deaf or hard-of-hearing users, nor is there audio narration of the environment for blind users. This results in limited inclusivity for elderly or differently-abled individuals.
- **Dependence on Heavy Hardware:** Many systems show a dependence on heavy hardware. VR-based systems, for instance, often require expensive headsets and powerful PCs.
- **Fragmentation of Tools:** Finally, there is a severe fragmentation of tools. Chatbots, TTS, VR avatars, and video tools all exist separately. There is no single platform that integrates speech, vision, gestures, and emotions into one unified experience.

1.7 Proposed System

The proposed system introduces an AI-generated digital human designed as an intelligent, expressive, and inclusive virtual assistant. Unlike conventional chatbots or static avatars, this solution integrates natural language processing, speech technologies, computer vision, and 3D animation into a unified framework capable of delivering human-like interactions.

The core of the system is developed on a web-based interface using Flask, which allows users to interact through both text and voice while simultaneously displaying a lifelike 3D avatar. The integration of the Phi model, implemented through Ollama, enables efficient local reasoning and ensures that responses remain context-aware, multilingual, and emotionally adaptive. Speech technologies play a crucial role in the system, where speech-to-text modules allow seamless voice input while text-to-speech generates natural, multilingual voice output, making interactions accessible to users who prefer or require spoken communication.

The vision module is powered by YOLOv8, which enables real-time object detection and environment narration. This feature significantly enhances accessibility by assisting visually impaired users, allowing the digital human to describe objects and surroundings in real time. Complementing this is the avatar animation component, developed using ReadyPlayerMe, which produces realistic 3D characters capable of lip synchronization with generated speech and dynamic emotional expression through AI-driven sentiment analysis. These features together ensure that the digital human not only speaks but also conveys emotions visually, creating a stronger sense of human-like presence.

Accessibility remains a cornerstone of the proposed system, with distinct operational modes designed to meet the needs of different user groups. For visually impaired users, the system provides environment narration through camera-based input, while deaf or mute users benefit from real-time captions integrated into the interface. Multilingual support ensures inclusivity across diverse populations, with language options such as English, Hindi, and Telugu. The emphasis on local deployment ensures that the system operates efficiently with low latency and without overreliance on cloud infrastructure, thereby protecting user privacy while guaranteeing offline availability. Overall, the proposed system aims to create a digital human that is natural, emotionally engaging, and socially inclusive, bridging the gap between computational intelligence and authentic human connection.

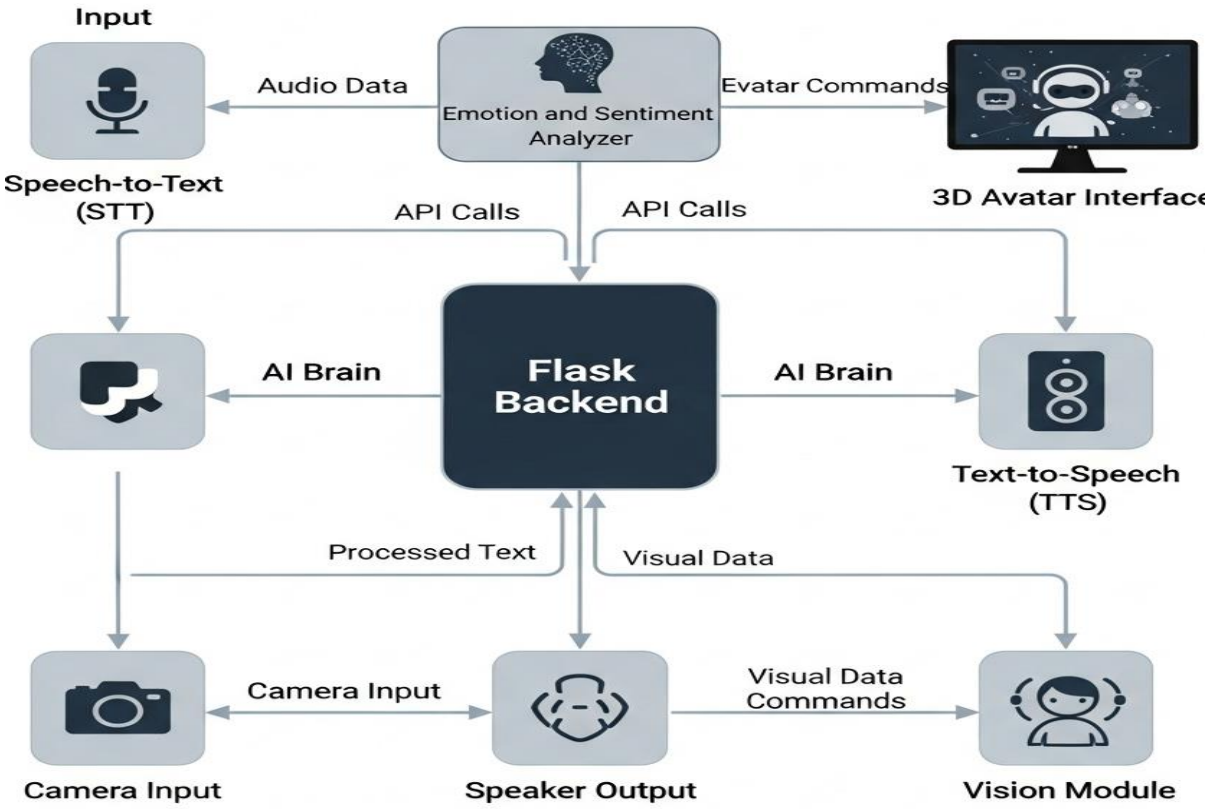


Figure 1.1 System Architecture

Advantages:

- Provides natural, human-like interaction with multilingual speech and expressive 3D avatars.
- Ensures real-time object and environment narration to assist visually impaired users.
- Offers inclusive accessibility modes for blind, deaf, mute, and elderly individuals.
- Guarantees low latency and privacy through local deployment without full dependence on the cloud.
- Supports emotional expression using AI-driven sentiment analysis mapped to avatar gestures and lip-sync.
- Functions as a versatile assistant across multiple domains including education, healthcare, business, and personal use.
- Enhances user experience and engagement by combining vision, voice, and emotional intelligence in one platform.

Block diagram:

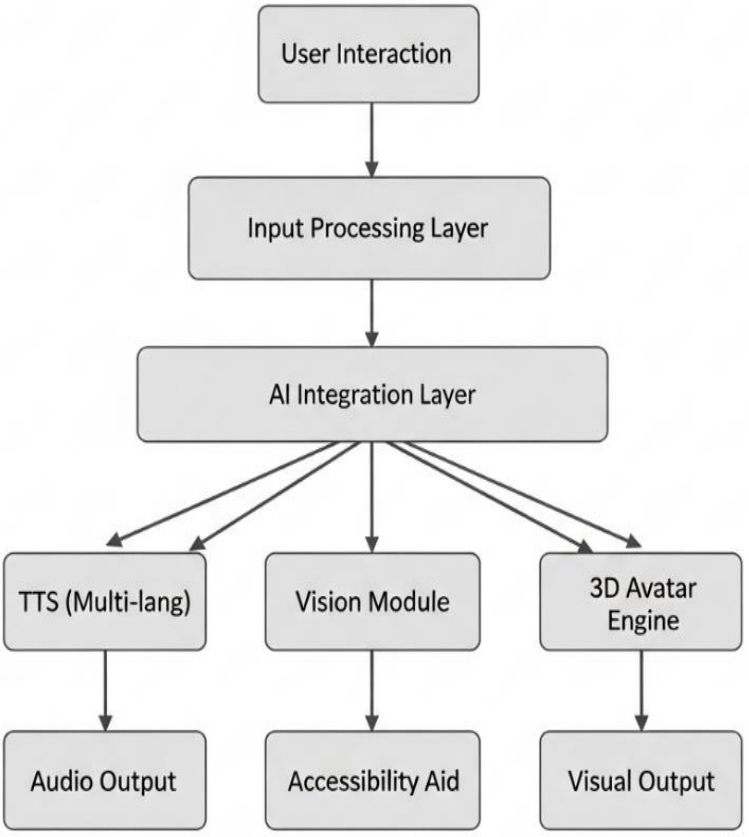


Figure 1.2 Workflow Diagram

1.8 Outcome of the project

The outcome of the project is the successful development of an AI-generated digital human that can function as a realistic, emotionally expressive, and inclusive assistant. The system demonstrates its ability to integrate speech recognition, natural language processing, computer vision, and 3D avatar rendering into a unified framework that supports seamless human-like interaction. Through the use of the Phi model for dialogue generation, YOLOv8 for object detection, and ReadyPlayerMe for avatar creation, the assistant is capable of responding intelligently while also narrating environments and displaying synchronized facial expressions and gestures. The platform provides accessibility through specialized modes that assist visually impaired, deaf, mute, and elderly users, ensuring inclusivity and adaptability across diverse populations.

1.9 Report Organization

Chapter 1:

This chapter gives the overall description about the project. It gives the overview of the proposed project work. It tries to answer why this project is needed in current scenario and what are various motivation factors that motivated to implement this project. This chapter also points out the limitations in the existing systems and tells how these limitations can be overcome by using this project.

Chapter 2:

This chapter gives details about various base papers that are related to the proposed project work. It shows how various activities related to the project were carried out at different point of time. It gives a short introduction to each base paper, talks about their shortcomings and tells how this project can overcome those shortcomings.

Chapter 3:

This chapter introduces the system analysis process. It gives brief idea whether this project should be done or not based on various feasibility study. It gives the summary of various feasibility studies that were carried out and shows the advantages of doing this project. At the same time, it also gives the overview of various functional and non- functional requirements of the system.

Chapter 4:

This chapter talks about various hardware and software tools that are necessary in order to implement this project. It provides details of software and languages that will be used and also lists the minimum requirements needed to run the project.

Chapter 5:

This chapter shows the detailed design of the architecture, components, modules, interfaces, and data for the proposed system to satisfy specified requirements. It shows various standard UML diagrams that are needed to design the system.

Chapter 6:

This chapter shows the implementation of the structure created during architectural design and the results of system analysis to construct system elements that meet the stakeholder requirements and system requirements developed in the early life cycle phases.

Chapter 7:

This chapter shows the various test results produced by the system. Various kinds of test are performed for each part of the system and as well as the whole system. It shows various pre-defined test cases and result of running these test cases on the system. It provides the comparison of expected output and the actual output produced by system based on which bugs are identified and eliminated.

Chapter 8:

This chapter shows various screenshots of the system. It also shows how data processing happens at various stages of the system and the final output is also displayed. And it also shows the outer interface design of the system.

1.10 Introduction Summary

Artificial Intelligence (AI)-generated digital humans represent the next evolution in human–computer interaction by combining deep learning, computer vision, natural language processing, speech synthesis, and 3D rendering into one unified system. Unlike traditional chatbots or avatars that feel robotic, digital humans are designed to be expressive, empathetic, and accessible, making interactions feel closer to real-life conversations.

The proposed system integrates multimodal inputs—voice, text, facial expressions, and camera-based vision—allowing avatars to perceive and understand the environment as well as user emotions. These avatars can speak naturally with emotional tone, lip-sync, and gestures, while also offering real-time captions for deaf users and scene narration for blind users, thus expanding inclusivity.

By merging perception, reasoning, and real-time rendering, this system creates lifelike, intelligent, and responsive digital humans that can serve in healthcare, education, business communication, and assistive technology. With improved accuracy, low latency, and personalization, the platform addresses the limitations of existing systems and sets a foundation for accessible, human-like digital interaction.

CHAPTER 2

LITERATURE SURVEY

2.1 Related Work

2.1.1 Reference Papers

A research paper is an academic document reporting original work, analysis, and findings. For the digital-human domain, recent surveys and review papers provide overviews of 3D avatar modeling, audio-driven facial animation, and multimodal systems — useful starting points to position our project and choose methods.

2.1.2 Social media analysis

Social media analysis can support a digital-human project by revealing user preferences, common complaint areas, language usage patterns, and accessibility needs. For example, sentiment and topic analysis across platforms can highlight which types of avatar behaviors users find engaging or uncanny; this guides emotion models, TTS voice style, and multilingual priorities. Social listening also helps identify domain-specific vocabulary (education, healthcare, therapy) so the language model and voice persona sound natural and context-appropriate. Recent multimodal-avatar surveys emphasize the importance of user-centered data when designing inclusive avatars.

2.1.3 Social Network Analysis

Social network analysis (SNA) can inform design and deployment strategies for digital humans in community settings. SNA helps identify influential nodes (teachers, moderators) and communication flows; deploying avatar agents as tutors, moderators, or guides to those nodes can increase adoption. SNA also helps detect clusters with special accessibility needs so the avatar's conversation models and persona can be adapted for group-specific norms and privacy constraints. Understanding communication flows and cluster structure lets you target personas and languages to groups that will adopt the technology faster. SNA can also reveal privacy-sensitive communities and inform consent and data-sharing policies before deployment. Finally, mapping interaction patterns supports placement of avatar agents as tutors, moderators, or accessibility assistants in existing social structures to increase trust and uptake.

2.2 Realistic Avatar System with Body Motions and Facial Expressions for Communication in Virtual Reality Applications.

NAME	YEAR	AUTHOR	FEATURE	ADAVANTAGE	DISADVANTAGE
Realistic Avatar System with Body Motions and Facial Expressions for Communication in Virtual Reality Applications.	2020	Sahar Aseeri, Sebastian Marin, Richard N. Landers, Victoria Interrante, Evan S. Rosenberg	The system features real-time, simultaneous capture of a user's body motions ,facial expressions and voice which are all rendered onto a single realistic 3D avatar.	By capturing and rendering nonverbal cues, the system provides a higher level of realism and can increase the user's sense of social presence.	The system is currently limited to a one-way interaction, where only one user is embodied as an avatar and the other user simply views it.

The paper "Embodied Realistic Avatar System with Body Motions and Facial Expressions for Communication in Virtual Reality Applications" .addresses a significant problem in virtual reality: the lack of effective, natural communication. This paper introduces a novel system that provides embodied virtual avatars that mimic a user's actions and speech to solve this issue.. For its methodology, the system uses HTC Vive trackers combined with an inverse kinematics (IK) solver to capture and animate the user's body movements. For facial expressions, it uses an iPhone X's depth camera and the LIVE FACE application, which streams the face data to a desktop. Finally, a microphone captures the user's speech. All three data streams are integrated using iClone 7 software and the Unreal-iClone Live Link plug-in, which renders the fully animated, realistic avatar in an immersive 3D environment for a second user to interact with.

2.3 FaceFormer: Speech-Driven 3D Facial Animation with Transformers

NAME	YEAR	AUTHOR	FEATURE	ADAVANTAGE	DISADVANTAGE
FaceFormer	2021	Yingruo Fan, Zhaojiang Lin, Jun Saito, Wenping Wang, Taku Komura.	Transformer-based autoregressive model that predicts 3D face meshes from audio using long-term audio context and self-supervised speech representations.	Improves lip-sync and continuity by modeling long-term audio context; shows strong perceptual results on 3D mesh animation.	Requires good-quality 3D training data or clever pretraining; increased model size and autoregressive decoding can add latency for real-time use.

FaceFormer introduced a groundbreaking way of synchronizing speech and facial animation using transformer networks, marking one of the earliest attempts to create fluid and lifelike mouth and face movements purely from audio input. This method can be directly applied to your digital human to achieve natural lip synchronization, reducing the robotic or delayed speech effects seen in older systems. The model's ability to capture long-term speech context allows smoother transitions between phonemes and facial expressions, leading to a more human-like appearance during conversation. For a digital human meant to communicate emotionally, this temporal consistency is vital for believability. FaceFormer also reduces the need for large amounts of paired video–audio datasets, which makes prototyping easier. Integrating a FaceFormer-style audio-to-mesh transformer within your system would give your avatar lifelike facial movements, improving both expressivity and accessibility for lip-reading users. Overall, the 2021 study establishes the technical base for realistic facial motion synthesis in virtual agents.

2.4 GENEA / Gesture & Co-speech Gesture Benchmarks and Advances (GENEA challenge papers)

NAME	YEAR	AUTHOR	FEATURE	ADAVANTAGE	DISADVANTAGE
GENEA / ICMI Proceedings & challenge reports	2022	GENEA community.	Benchmarks and challenges for automatic co-speech gesture generation and evaluation (multimodal gesture/gesture realism metrics).	Provides standardized datasets, evaluation protocols and community baselines for co-speech gesture generation (important for whole-body digital humans).	Benchmarks often focus on constrained studio data; generalization to in-the-wild conversational contexts is still limited.

The GENEA Challenge (held annually from 2022) provided an open benchmark for gesture generation models that produce synchronized body and hand motions during speech. For your AI digital human, these benchmarks supply both datasets and metrics that help in training and testing natural hand gestures that accompany spoken dialogue. By analyzing human-gesture alignment, GENEA research shows how timing, rhythm, and semantics in speech correspond to different gesture types. This insight can make your avatar appear more socially engaging and contextually aware rather than static or overly mechanical. Furthermore, GENEA emphasizes perceptual evaluation measuring how humans *feel* about an avatar's motion which aligns with your project's goal of emotional realism. Implementing GENEA-style gesture synthesis would allow your avatar to use appropriate non-verbal cues, improving immersion, communication clarity, and user comfort during interaction.

2.5 X-Avatar / XAGen and Audio-Driven 3D animation (expressive avatars)

NAME	YEAR	AUTHOR	FEATURE	ADAVANTAGE	DISADVANTAGE
X-Avatar (expressive human avatars) / XAGen	2023	K. Shen et al.	Holistic models that jointly model facial expression, hands, body and appearance to create highly expressive 3D avatars (part-aware skinning, SMPL-X driven).	Produces coherent whole-body and facial expression synthesis, suitable for telepresence and expressive virtual agents.	High fidelity requires 3D scans or multi-view data and notable compute; heavy models are costly to run on lightweight hardware.

By 2023, research shifted toward *holistic avatar systems* that combine facial, body, and hand motion under a unified 3D model, such as X-Avatar and XAGen. These frameworks demonstrated how integrating multiple expressive channels leads to much more believable and emotionally resonant digital humans. In your project, adopting a similar approach ensures that speech-driven facial expressions are naturally complemented by synchronized body gestures and posture adjustments. X-Avatar's SMPL-X modeling captures subtle emotional nuances — a raised eyebrow, a gentle head tilt, or relaxed shoulders — which make digital communication feel genuine. These systems also support dynamic texture updates and emotion mapping, letting your avatar visually react to user sentiment detected by the AI module. Although computationally demanding, the realism gained makes the avatar suitable for healthcare, education, or counseling use cases where empathy and engagement matter most. Thus, the 2023 advances lay the foundation for building truly human-centered, full-body expressive avatars.

2.6 ECCV / ECCV-2024 & 2024 advances: KMTalk, 3DFacePolicy, surveys on audio-driven animation.

NAME	YEAR	AUTHOR	FEATURE	ADAVANTAGE	DISADVANTAGE
KMTalk (Key-Motion Embedding)	2024	Zhihao Xu et al.	Novel 3D facial motion embeddings and diffusion-policy approaches that improve realism and controllability; 2024 surveys synthesize the state-of-the-art in audio→3D animation and metrics.	Newer methods improve upper-face expressivity, disentangle motion components, and propose robust samplers for realistic sequences; surveys summarize evaluation best practices.	Advanced methods (diffusion, key-embedding samplers) can be computationally heavier and yet-to-be fully optimized for low-latency real-time pipelines.

The 2024 studies such as KMTalk and 3DFacePolicy advanced controllable and expressive facial animation by introducing key-motion embeddings and diffusion policy networks. These innovations made it possible to not only generate realistic speech-driven animations but also to control emotional tone — smiling while speaking happily or showing concern in empathetic dialogue. For your digital human, this is essential: it allows emotional adaptivity, meaning the avatar can visually express context-appropriate feelings based on conversation cues from ChatGPT or sentiment analysis. They also support fine-tuning for specific personality styles — for example, a calm teacher avatar versus an energetic customer-service agent. The 2024 literature also emphasized ethical evaluation and user perception, offering design guidelines for avoiding the "uncanny valley." Integrating such controllable models helps your project produce avatars that are expressive, emotionally consistent, and trustworthy in real-time interaction.

2.7 Real-Time Speech-to-Text Holographic Communication for the Deaf Children and Elderly.

NAME	YEAR	AUTHOR	FEATURE	ADAVANTAGE	DISADVANTAGE
Real-Time Speech-to-Text Holographic Communication for the Deaf Children and Elderly.	2024	Savarala Chethana, Sreevathsa Sree Charan, Suja Palaniswamy, Vemula Srihitha	It uses a dual recognition model—comprising a conventional ASR library and the Wav2Vec2 transformer—to enhance accuracy, speed, and adaptability to diverse speech patterns.	This innovative combination enables real-time communication for deaf and hard-of-hearing individuals, improving inclusivity and accessibility.	Its reliance on specific hardware components and sensitivity to lighting conditions can affect the clarity and consistency of holographic projections in real-world applications.

This project is highly relevant in today's world, where inclusivity and accessibility in communication are vital. It addresses one of the major barriers faced by the deaf and hard-of-hearing community by transforming auditory speech into a visually engaging holographic text format. The innovation supports deaf children in classrooms, helping them follow lessons and interact with teachers without relying on interpreters. It also assists elderly individuals with hearing impairments in understanding conversations during social or medical interactions. Beyond education and healthcare, the system can be extended to public places such as railway stations, airports, and offices for real-time announcements. By merging speech recognition with holographic visualization, the project enhances both communication quality and user experience. Its real-time, low-cost, and efficient nature makes it a promising step toward creating a more inclusive technological ecosystem. Overall, the work contributes meaningfully to assistive communication technology and supports the global movement toward accessibility and equality for all.

2.8 Enabling Beam Search for Language Model-Based Text-to-Speech Synthesis.

NAME	YEAR	AUTHOR	FEATURE	ADAVANTAGE	DISADVANTAGE
Enabling Beam Search for Language Model-Based Text-to-Speech Synthesis	2025	Zehai Tu , Guangyan Zhang, Yiting Lu, AdaezeAdigwe, SimonKing ,Yiwen Guo	introduces Temporal Repetition Aware Diverse Beam Search (TRAD-BS), a novel decoding strategy designed to enhance Language Model-based Text-to-Speech (TTS) synthesis.	It also enables more deterministic and consistent speech generation, which is especially beneficial for long-form text and emotional or acoustically complex utterances.	the approach can be computationally intensive, especially when decoding with multiple beams, increasing processing time and hardware demand.

This study holds significant relevance to the advancement of speech synthesis and human–computer interaction technologies. By addressing common issues such as instability and mispronunciation in LM-based TTS, it bridges the gap between artificial speech generation and human-like voice performance. The TRAD-BS approach enhances both the linguistic accuracy and emotional consistency of synthetic voices, making it particularly suitable for applications such as virtual assistants, AI-driven communication, and accessibility tools for people with speech or hearing impairments. Moreover, as speech interfaces become integral to AR/VR systems, educational tools, and digital humans, this decoding enhancement provides a foundation for more natural, expressive, and reliable voice synthesis in real-world interactive environments. Its focus on improving decoding rather than just model scaling highlights a novel direction for future research, emphasizing quality optimization through algorithmic refinement rather than solely data expansion.

2.9 Literature Survey Summary

From 2020 through 2025 the field of digital humans moved quickly from single-modal lip-sync systems to full multimodal, expressive avatars that combine audio, text (dialogue), vision and gesture. In 2021, transformer-based models (e.g., FaceFormer) showed that long-context audio representations significantly improve 3D facial motion prediction; by 2022 community benchmarks (GENEA) highlighted the importance of realistic, co-speech gestures and standardized evaluation. 2023 brought holistic avatar modeling (X-Avatar/XAGen) that jointly models face, hands and body for coherent expression. In 2024 research emphasized controllable and sample-efficient motion models (key-motion embeddings, diffusion policies) and published comprehensive surveys for audio-driven animation; these help pick models and metrics. By 2025 the gap between research and applied systems narrowed: multimodal platforms and accessible real-time tools like NVIDIA Audio2Face demonstrate practical paths to deploy expressive avatars, while also raising ethical and compute-cost considerations. Taken together, these works suggest a hybrid architecture for your project: use production-ready avatar toolchains for the visual rig (fast prototyping), transformer / diffusion or key-motion modules for high-quality speech→face, a lightweight gesture generator tied to speech for body motions, and STT/TTS + ChatGPT-style dialogue modules for intelligent interaction — with accessibility layers (real-time captions, object narration) and social-listening-driven persona tuning.

CHAPTER 3

SYSTEM ANALYSIS

3.1 Introduction to System Analysis

The system analysis for the AI Generated Digital Human for Virtual Interactions involves an in-depth examination of the existing problems, the proposed system's functionality, its architectural design, and the interactions between different modules to achieve real-time, human-like communication. The project integrates artificial intelligence, speech processing, computer vision, and 3D rendering technologies to create a multimodal digital human capable of natural interaction through speech, vision, and emotion.

3.2 Feasibility Study

System feasibility analysis determines whether the proposed AI Generated Digital Human for Virtual Interactions can be successfully developed, implemented, and maintained with the available resources, technology, and user requirements. It assesses the practicality of the system from various perspectives—technical, operational, economic, and social feasibility—to ensure that the solution is both achievable and sustainable in real-world environments.

3.2.1 Technical Feasibility

The proposed system is technically feasible due to the availability of modern AI frameworks, open-source libraries, and web technologies that support multimodal integration. The system utilizes proven and accessible technologies such as Python, Flask, HTML, CSS, and JavaScript for web deployment. The AI model (Ollama's Phi) provides an efficient and context-aware conversational capability, while Google Speech Recognition and gTTS/pyttsx3 handle multilingual voice interaction effectively.

The vision module employs YOLOv8 and OpenCV, both of which are lightweight and GPU-optimized frameworks, enabling real-time object detection and narration. Three.js and Ready Player Me ensure smooth 3D rendering and animation for the digital avatar, providing realistic lip synchronization and emotion mapping. Overall, the technologies chosen are well-supported, documented, and compatible, ensuring smooth implementation and minimal technical risk.

3.2.2 Economic Feasibility

The system is also economically viable, as it leverages open-source and cost-effective tools, reducing the overall project expenditure. The key components—such as YOLOv8, Flask, Three.js, OpenCV, and Google APIs are freely available or have free-tier usage options, making development and deployment affordable for research and institutional purposes. Since most processing occurs locally or on lightweight servers, there are no significant recurring costs except for optional cloud hosting or advanced GPU resources for large-scale deployment. Compared to commercial AI assistant platforms that require expensive subscriptions or licensing, this project's use of open-source frameworks ensures cost efficiency without compromising performance. Therefore, the system can be implemented and maintained within limited budgets, making it economically sustainable for educational, healthcare, and accessibility-oriented applications.

3.2.3 Operational Feasibility

The system is designed to be user-friendly, inclusive, and accessible to a wide range of users, including visually impaired, hearing-impaired, and multilingual individuals. The web-based interface developed through Flask provides simple navigation and supports voice, text, and visual interactions. The avatar interface delivers real-time emotional feedback, while the object narration feature assists users in understanding their surroundings.

For administrators and developers, the modular design ensures easy maintenance and future scalability. New languages, emotions, or functionalities can be added without modifying the entire system. Users require only a microphone, webcam, and a stable internet connection, which ensures smooth operation even in resource-constrained environments. Thus, the system is operationally feasible and can be effectively used across multiple domains such as education, healthcare, virtual communication, and assistive technology.

3.2.4 Schedule Feasibility

The proposed system follows a structured modular development process that ensures timely completion within standard project timelines. Each module STT, TTS, AI processing, vision, and avatar rendering—can be developed and tested independently. The integration phase is streamlined due to Flask's middleware design. Based on prototype testing and evaluation with different user groups, the development schedule is realistic or institutional project durations.

3.2.5 Social Feasibility

From a social standpoint, the project contributes positively to society by promoting inclusivity and accessibility. The system benefits visually impaired users through real-time environmental narration and hearing-impaired users via emotion-based visual feedback. It also supports multilingual users, bridging communication barriers across regional and linguistic communities.

By creating a digital human capable of empathetic and expressive communication, the project encourages humanized AI interaction and improves user comfort with technology. This aligns with ethical and social objectives of modern AI research, making the system both socially acceptable and beneficial.

3.3 Functional requirements

1. Speech-to-Text (STT) Conversion
    - The system must capture the user's voice input and accurately convert it into text.
    - It should support automatic language detection to identify whether the input is in English, Hindi, Telugu, or another supported language.
    - It must handle multilingual speech without requiring manual language switching.
2. AI-Based Response Generation
    - The system should process the transcribed input through the Ollama Phi model to generate intelligent and contextually relevant responses.
    - It must perform sentiment analysis to determine the emotional tone (positive, neutral, or negative) of the response for avatar expression alignment.
3. Text-to-Speech (TTS) Output
    - The system must convert AI-generated text responses into natural-sounding speech in the user's language.
    - It should maintain tone and clarity across multiple languages.
    - The TTS engine should ensure smooth synchronization with the avatar's lip movement.
4. 3D Avatar Visualization and Lip Synchronization
    - The system must display a 3D digital human avatar (from Ready Player Me) capable of lip movements that sync with generated speech.

- It should render emotional facial expressions that correspond to the detected sentiment of the conversation.
- The avatar must provide real-time visual feedback to users.

5. Vision-Based Object Detection Module
    - The system must use YOLOv8 and OpenCV to detect real-world objects from a webcam feed.
    - It should narrate the detected objects to assist visually impaired users.
    - It must provide real-time feedback like "There is a person on your left."
6. Multilingual and Accessibility Support
    - The interface should allow seamless interaction for users with visual, hearing, or speech impairments.
    - For visually impaired users – the system narrates detected objects.
    - For hearing impaired users – the system provides text captions and emotional avatar feedback.
    - For speech impaired users – text-only interaction must be supported.
7. User Interface (Web Portal)
    - The system must provide a Flask-based web interface for interaction.
    - It should contain an initial portal for language and accessibility selection, followed by an interaction page featuring the avatar and chat panel.
    - The interface should display ongoing conversations in real-time.
8. Performance and Optimization Functions
    - The system must implement caching to reduce repeated AI and TTS processing.
    - It should support GPU-optimized object detection for low-latency performance.
    - A "thinking" animation must appear while the AI processes a response to improve user experience.
9. Data Processing and Storage
    - Temporary logs of user inputs and system responses can be maintained for debugging or analysis.
    - Sensitive information should not be stored permanently to protect user privacy.
10. System Integration and Communication

- The backend (Python/Flask) must integrate all modules — STT, AI response, TTS, and vision — and manage data flow efficiently.
- The front-end (HTML, CSS, JavaScript, Three.js) must synchronize with backend responses for real-time rendering and user interaction.

3.4 Non-Functional Requirements

1. Performance Requirements

The system must deliver real-time responsiveness for smooth interaction between the user and the digital human.

- Average response latency should remain below 2 seconds between user input and avatar response.
- The speech-to-text (STT) and text-to-speech (TTS) modules must process audio input/output seamlessly without noticeable delay.
- The YOLOv8 vision module should maintain an object detection accuracy of at least 85% under standard lighting conditions.
- The avatar's lip synchronization should align with the speech waveform with at least 90% accuracy for a realistic experience.

2. Reliability Requirements

Reliability ensures continuous system operation with minimal downtime or functional failures.

- The system should maintain consistent performance during long sessions of interaction without freezing or crashing.
- Local caching mechanisms must store frequently used responses and TTS outputs to minimize processing delays.
- Error-handling and recovery mechanisms should be implemented to automatically reset the system during unexpected failures.
- Data processed during conversations should be preserved temporarily for continuity but cleared after the session to ensure stability and privacy.

3. Scalability Requirements

The modular design must allow the system to scale easily to handle future expansions and additional features.

- New languages, voice models, or emotional expressions should be integrated without affecting the existing modules.

- The architecture must support migration from local deployment to cloud-based infrastructure when needed.
- The system should accommodate simultaneous user interactions if deployed in multi-user environments.

4. Usability Requirements

The system interface must be intuitive, accessible, and inclusive for all user groups, including differently-abled individuals.

- The interface must allow voice-based, text-based, and visual interaction modes for diverse accessibility needs.
- A clean and simple layout should enable users to switch languages or modes with minimal effort.
- For visually impaired users, audio narration and voice feedback must provide full interaction capabilities without requiring visual elements.
- For hearing-impaired users, the avatar should display emotion-based visual cues and text captions.

5. Security Requirements

The system should guarantee user data safety and privacy throughout all interactions.

- Personal or audio data captured during sessions must not be stored permanently unless explicitly permitted by the user.
- The system should operate in a secure environment where APIs and AI models are protected from unauthorized access.
- Secure communication protocols (HTTPS) must be used for all web-based interactions.
- Any logs or temporary data generated during runtime should be automatically deleted after the session ends.

6. Maintainability Requirements

- Each module (STT, TTS, Vision, AI, Avatar) should be independently upgradable without affecting other components.

- Source code should be well-documented, ensuring that future developers can easily modify or expand the system.
- The Flask framework should enable smooth integration of new features such as advanced emotion recognition or extended vision capabilities.

7. Portability Requirements

- It must function on common operating systems such as Windows, macOS, and Linux with minimal configuration.
- The web interface should run efficiently on modern browsers (Chrome, Edge, Firefox).

8. Availability Requirements

The system must be available for use at any time with minimal downtime.

- The digital human application should provide 99% uptime under standard operating conditions.
- Backup and recovery mechanisms should restore system functionality promptly after interruptions.
- The system must auto-restart critical services like AI response generation and speech modules if any component fails.

3.5 System Analysis Summary

The system analysis of the AI Generated Digital Human for Virtual Interactions demonstrates a comprehensive understanding of how artificial intelligence, speech processing, computer vision, and 3D animation can be integrated into a unified framework to simulate realistic human communication. The study identifies major limitations in existing AI assistants such as Siri, Alexa, and Google Assistant, which lack emotional expression, visual interaction, and accessibility support for differently-abled users.

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 Functional Requirements

Functional requirements describe the specific operations and behaviors that the system must perform.

1. **Speech-to-Text (STT) Conversion**
    - The system must capture user voice input and convert it accurately into text in real time.
    - It should support multilingual speech recognition, including English, Hindi, and Telugu.
2. **AI Response Generation**
    - The system should process user inputs through the AI model (Ollama's Phi) to generate contextually relevant and meaningful responses.
    - The AI must detect the emotional tone of the conversation and relay this information for avatar expression rendering.
3. **Text-to-Speech (TTS) Synthesis**
    - The system should convert the AI-generated text responses into natural-sounding speech in the same language as the user's input.
4. **Vision Module**
    - The system must detect real-time objects from webcam input using YOLOv8 and narrate them to visually impaired users.
    - It should identify objects and people with reasonable accuracy under normal lighting conditions.
5. **3D Avatar Rendering**
    - The avatar must synchronize lip movements with speech output and display emotional expressions based on the conversation sentiment.
    - The avatar should render smoothly in real time using Three.js and Ready Player Me models.

6. **User Interface and Accessibility**
    - The web interface must allow users to select language and accessibility options before starting interaction.
    - The interface should support multiple modes of interaction—voice, text, and visual feedback—for differently-abled users.
7. **Data Flow and Integration**
    - The Flask framework should act as middleware to integrate all modules (STT, TTS, AI, Vision, and Avatar).
    - Data should flow sequentially from user input to AI processing, response generation, and output rendering without errors.

4.2 Non-Functional Requirements

Non-functional requirements define the performance, usability, and quality standards of the system.

1. **Performance:**
    - System response time should be less than 2 seconds for real-time conversation.
    - Object detection accuracy should be above 85% under optimal conditions.
2. **Reliability:**
    - The system should maintain stable operation during long sessions and handle minor input errors gracefully.
3. **Usability:**
    - The interface must be simple, intuitive, and accessible for users with visual or hearing impairments.
4. **Scalability:**
    - The modular design must allow easy integration of new languages, avatars, or AI models.
5. **Security:**
    - User data (voice, image, and text) should be processed securely and not stored permanently.
    - All web communications must use secure (HTTPS) protocols.
6. **Maintainability:**
    - Source code should be modular and well-documented to facilitate updates.

7. **Portability:**
    - The application must run on multiple platforms (Windows, macOS, Linux) and browsers.

4.3 Software Requirements

The software environment and libraries required for the development and deployment of the system include:

Category	Software / Library Used	Purpose
Programming Language	Python 3.10+	Backend logic and AI module integration
Web Framework	Flask	Middleware for connecting modules and hosting web interface
Frontend	HTML, CSS, JavaScript	User interface development
3D Rendering	Three.js	Rendering and animation of the avatar
Avatar Engine	Ready Player Me	Creation of 3D digital human avatar
AI Model	Ollama's Phi	Contextual AI response generation
Speech Recognition	Google Speech Recognition API	Speech-to-text processing
Text-to-Speech	gTTS, pyttsx3	Multilingual voice output
Vision Framework	YOLOv8, OpenCV	Object detection and narration
Sentiment Analysis	Python NLP libraries	Emotion detection for avatar expression
Development Environment	Visual Studio Code / PyCharm	Coding and debugging platform
Browser Support	Google Chrome, Edge, Firefox	Web-based interaction interface

4.3.1 Operating System

1. Role of the Operating System in the System

The operating system (OS) plays a crucial role in ensuring smooth functioning of the system by managing hardware resources, handling concurrent tasks, and facilitating communication between different software layers. Since the project integrates multiple modules — such as Speech-to-Text (STT), Text-to-Speech (TTS), AI model processing, vision detection, and 3D avatar rendering — the OS must provide:

- Efficient CPU scheduling for real-time processing.
- Memory management to handle multiple running modules simultaneously.
- GPU support for vision-based tasks using YOLOv8 and rendering via Three.js.
- A multi-threaded environment for asynchronous execution of speech and visual processing.

Thus, the chosen operating system must be capable of supporting high-performance computing and seamless integration between Python, Flask, and web technologies

2. Recommended Operating Systems

The system is designed to be cross-platform, meaning it can run effectively on major operating systems such as Windows, Linux, and macOS. However, based on experimental implementation and performance evaluation, Windows 10/11 (64-bit) is the most suitable and recommended environment for development, testing, and deployment.

Below is a detailed explanation of the supported operating systems:

a. Windows 10 / Windows 11 (64-bit)

- **Purpose:** Primary development and deployment platform.
- **Advantages:**
    - Broad compatibility with Python libraries and AI frameworks.
    - Easy integration with GPU drivers (NVIDIA CUDA) for YOLOv8 object detection.
    - Supports all necessary tools such as Flask, OpenCV, and Visual Studio Code.
    - Provides a user-friendly environment for debugging and testing the web interface.

- **Minimum Requirements:**
    - Processor: Intel Core i5 / AMD Ryzen 5 or higher
    - RAM: 8 GB minimum (16 GB recommended for better performance)
    - GPU: NVIDIA GTX 1050 or higher with CUDA support
    - Storage: Minimum 10 GB free space
    - Network: Stable internet connection for AI inference and speech modules

b. Linux (Ubuntu 20.04 LTS or Later)

- **Purpose:** Ideal for deploying the application on local or cloud servers.
- **Advantages:**
    - Highly stable and optimized for backend services and AI model hosting.
    - Excellent performance for Python-based frameworks and Flask servers.
    - Native compatibility with GPU computing and open-source development environments.
    - Better memory management and multitasking support compared to other OSes.
- **Minimum Requirements:**
    - Processor: Intel i5 / AMD equivalent or higher
    - RAM: 8 GB minimum
    - GPU: CUDA-enabled NVIDIA GPU (optional but recommended for YOLOv8)
    - Network: Required for live streaming and speech recognition modules

c. macOS (Monterey or Later)

- **Purpose:** Optional platform for academic or development purposes.
- **Advantages:**
    - Efficient system resource management and compatibility with Python tools.
    - Smooth execution of Flask applications and web-based rendering tasks.
    - Suitable for developers preferring Apple's ecosystem.
- **Limitations:**
    - Limited compatibility with GPU-based frameworks like YOLOv8.
    - Requires additional configurations for CUDA and AI-related libraries.

4.4 Software Requirement Summary

- The AI Generated Digital Human for Virtual Interactions *system* integrates multiple technologies that enable seamless communication between the user and a lifelike digital avatar. The software requirements define the tools, frameworks, and libraries necessary for developing, running, and maintaining the system effectively. The project combines artificial intelligence, computer vision, and web development technologies into a unified framework for real-time human-computer interaction.
- The system uses Python as the primary programming language due to its flexibility, rich library ecosystem, and compatibility with AI and machine learning frameworks. Flask**,** a lightweight Python-based web framework, serves as the middleware that connects backend logic with the frontend interface, enabling smooth data exchange between modules such as speech recognition, AI response generation, text-to-speech conversion, and 3D avatar rendering.
- For the frontend**,** standard web technologies like HTML, CSS, and JavaScript are used to design an interactive and responsive user interface. The Three.js library supports 3D visualization and animation, while Ready Player Me provides customizable 3D digital avatars that represent human expressions and lip movements in real time.
- The system integrates the Ollama's Phi model for generating natural, context-aware responses, making interactions more conversational and emotionally adaptive. The Google Speech Recognition API is used for multilingual speech-to-text processing, while gTTS and pyttsx3 handle text-to-speech synthesis to produce realistic and regionally appropriate voices. Additionally, YOLOv8 and OpenCV are implemented for real-time object detection and vision-based narration, enhancing accessibility for visually impaired users.
- For sentiment analysis and emotion-driven responses, Python's natural language processing libraries are utilized to align the avatar's facial expressions with conversational tone. Development and testing are performed using Visual Studio Code or PyCharm, providing an integrated environment for code management, debugging, and module integration.

4.5 Hardware Requirements

Component	Specification	Purpose / Description
Processor (CPU)	Intel Core i3 (8th Gen) / AMD Ryzen 3 or higher	Handles backend computations and web server processing
Memory (RAM)	8 GB	Enables smooth multitasking between STT, TTS, AI, and vision modules
Storage	10 GB free disk space	Required for installing libraries, storing temporary data, and caching responses
Graphics card (GPU)	Integrated GPU or entry-level NVIDIA GPU (2 GB VRAM)	Supports basic avatar rendering and limited vision processing
Webcam	720p or higher	Captures live video input for object detection and narration
Microphone	Standard noise-canceling microphone	Captures clear audio input for speech recognition
Speakers / Headphones	Any standard output device	For audio output of text-to-speech responses
Network	Stable broadband connection (2 Mbps or higher)	Required for accessing APIs and model responses
Display	Minimum 15-inch monitor, 720p resolution	For displaying 3D avatar interface and system dashboard

Table 4.5 Hardware Requirements

The AI Generated Digital Human for Virtual Interactions system integrates several high-performance components such as speech recognition, computer vision, and 3D rendering, all of which demand a stable and capable hardware environment. The hardware requirements ensure that the system runs smoothly, processes real-time data efficiently, and supports the computational load of AI inference and graphical rendering. The hardware configuration is divided into minimum and recommended requirements to suit different deployment conditions—academic demonstrations, research environments, or production-level systems.

4.6 Hardware Requirements Summary

These specifications represent the lowest configuration required for basic operation, including speech interaction, text response generation, and limited 3D avatar rendering.
This configuration allows the system to function correctly for testing and small-scale demonstrations, but may experience minor delays during intensive AI or vision tasks.

4.7 Requirement Analysis Summary

This chapter talks about various hardware and software tools that are necessary in order to implement this project. It provides details of software and languages that will be used and also lists the minimum requirements needed to run the project.

CHAPTER 5

SYSTEM DESIGN

The System Design of the AI Generated Digital Human for Virtual Interactions is the architectural blueprint that defines how various hardware and software components work together to achieve realistic, human-like communication through speech, vision, and emotion. The design focuses on ensuring modularity, scalability, and real-time performance. Each module in the system performs a specific function, and all modules are integrated to create a seamless multimodal experience between the user and the digital avatar.

The design follows a modular and layered architecture, where independent components handle tasks such as speech recognition, AI-driven response generation, text-to-speech synthesis, object detection, and 3D avatar rendering. The entire system is coordinated through a Flask-based middleware that manages the data flow between frontend and backend processes.

5.1 System Architecture Overview

The overall architecture is divided into five main functional modules, each performing a distinct role in the digital human interaction process:

1. **Speech-to-Text (STT) Module**
    - Captures voice input from the user through a microphone.
    - Converts spoken language into text using Google Speech Recognition API.
    - Automatically detects the language (e.g., English, Hindi, Telugu) and forwards the transcribed text to the AI module for processing.
2. **AI Response Generation Module**
    - This serves as the “brain” of the digital human, implemented using Ollama’s Phi model.
    - A sentiment analysis sub-module evaluates the emotional tone (positive, neutral, or negative) of the response, which is later reflected in the avatar’s expressions.
3. **Text-to-Speech (TTS) Module**
    - Converts AI-generated textual responses into speech output using gTTS and pyttsx3 libraries.
    - Supports multilingual speech synthesis so that the response is delivered in the

same language as the user's input.

- Produces realistic and natural-sounding voice output synchronized with the avatar's.

4. **Vision and Accessibility Module**
   - Uses YOLOv8 and OpenCV for real-time object detection and environmental narration.
   - The camera captures live visuals, and the system narrates detected objects or people, assisting visually impaired users.
   - This module operates in parallel with the conversational interface to provide continuous accessibility support.
5. **3D Avatar Rendering Module**
   - Implements a lifelike 3D avatar using Ready Player Me integrated with Three.js for rendering.
   - The avatar performs lip synchronization based on audio output and displays facial expressions (happy, sad, neutral) according to the AI's sentiment analysis results.
   - This enhances the emotional realism and human-likeness of the interaction.

5.1.2 System Flow Design

The overall workflow of the system can be explained through the following steps:

1. **User Input :**

   The interaction begins when the user speaks or types a message. Voice input is captured via the microphone and sent to the STT module for transcription.
2. **Language Detection and Processing:**

   The STT module detects the language automatically and converts the speech into text. The transcribed text is cleaned and normalized for AI processing.
3. **AI Processing and Sentiment Analysis:**

   The text input is passed to the **Phi model**, which generates an intelligent response.
4. **Speech Synthesis and Avatar Response**

   The AI-generated response is converted to speech by the TTS module. Simultaneously, the 3D avatar synchronizes lip movements with the generated audio and adjusts its facial expressions.

5. **Vision-Based Object Dectection (Optional):**

   If the user enables accessibility features, the camera captures the environment, and YOLOv8 identifies and narrates objects around the user.

6. **User output:**

   The user sees the avatar speaking on-screen, hears the response through the speakers, and receives additional visual or verbal feedback if accessibility mode is active.This flow ensures smooth, real-time communication that closely mimics natural human interaction.

7. **System Design Diagram (Conceptual)**

   Although a visual block diagram is typically used in the report, the structure can be described textually as:

   [User Input]

   ↓

   [Speech-to-Text (STT) Module]

   ↓

   [AI Brain - Ollama's Phi Model + Sentiment Analysis]

   ↓

   [Text-to-Speech (TTS) Module] → [3D Avatar Rendering]

   ↓ ↑

   [Vision Module (YOLOv8)] ←→ [Flask Middleware]

   ↓

   [User Output – Voice + Visual + Emotional Interaction]

   Each block operates independently but communicates through Flask, which acts as a bridge between frontend and backend processes. This modular design ensures flexibility, scalability, and maintainability.

5.1.3 Design Characteristics

- **Modularity:**

  Each system component—AI, Vision, Speech, and Avatar—is developed as an independent module that can be updated or replaced without affecting others.

- **Parallel processing:**

  Speech processing, AI reasoning, and object detection run concurrently, ensuring faster

response times.

- **Scalability:**

  The architecture allows easy integration of additional languages, emotions, or avatars as future enhancements.

- **Reusability:**

  Components such as the STT and TTS modules can be reused in other AI-based accessibility or interactive systems.

- **User Accessibility:**

  The system design emphasizes inclusivity, offering voice, vision, and text-based interaction modes for differently-abled users.

5.2 Gant chart

5.2.1 Gantt Chart for Phase One

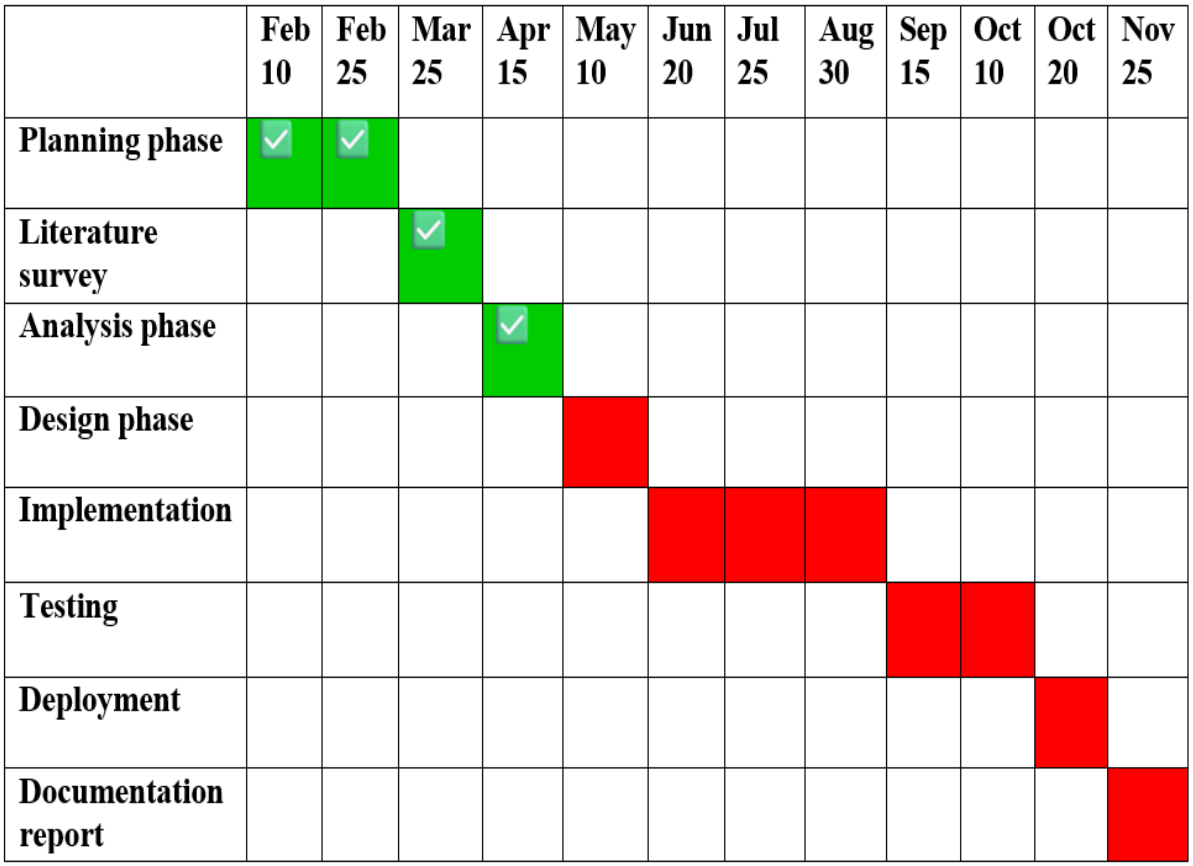


Figure.5.2 Gantt Chart For Phase One

5.2.2 Gantt Chart for Phase Two

	Feb 10	Feb 25	Mar 25	Apr 15	May 10	Jun 20	Jul 25	Aug 30	Sep 15	Oct 10	Oct 20	Nov 25
Planning phase	✓	✓										
Literature survey			✓									
Analysis phase				✓								
Design phase					✓							
Implementation						✓	✓	✓				
Testing									✓	✓		
Deployment											✓	
Documentation report												✓

Figure 5.3 Gantt Chart For Phase Two

5.3 Life Cycle Model

1. Planning and Requirement Analysis

In the first phase, the overall system requirements are gathered and analyzed in detail. For the AI Generated Digital Human system, the following key requirements were identified:

- Real-time multilingual communication through speech-to-text and text-to-speech.
- Integration of a 3D expressive avatar with lip synchronization and emotional response.
- Inclusion of a vision module using YOLOv8 for object detection to assist visually impaired users.

- Development of a web-based user interface for accessibility and easy interaction.

Feasibility studies were performed to assess hardware, software, and operational needs. The outcome of this phase was a requirement specification document that served as a foundation for further development.

2. Risk Analysis and Prototyping

This is a crucial phase where potential technical and operational risks are identified, analyzed, and mitigated before full-scale development.
For this system, risk areas included:

- Accuracy of speech recognition in noisy environments.
- Latency during real-time communication.
- Lip-sync precision between avatar expressions and voice output.
- Compatibility between AI modules, Flask server, and web technologies.

To reduce risks, prototypes of each module such as speech-to-text (STT), text-to-speech (TTS), vision detection, and avatar rendering were developed and tested individually.
This stage ensures that the team can identify design or performance issues early and refine the solution before full implementation.

3. Implementation and Testing

Once risks are addressed, the system proceeds to full development.
The implementation was carried out using Python as the core programming language and Flask as the middleware to integrate all modules.
Key components developed:

- **STT Module:** Converts user speech to text using Google Speech Recognition.
- **AI Brain:** Generates intelligent responses using Ollama's Phi model.
- **TTS Module:** Produces natural speech using gTTS and pyttsx3.
- **Vision Module:** Detects and narrates objects in the environment using YOLOv8 and OpenCV.
- **3D Avatar Interface:** Created using Ready Player Me and rendered with Three.js, capable of showing emotions and synchronized lip movements.

Testing was conducted with diverse users (normal, visually impaired, and hearing impaired) to ensure accessibility, real-time interaction, and natural communication.

4. Evaluation and Feedback

After implementation, the system is tested and evaluated based on performance metrics such as:

- Response latency
- Speech quality
- Detection accuracy
- User satisfaction

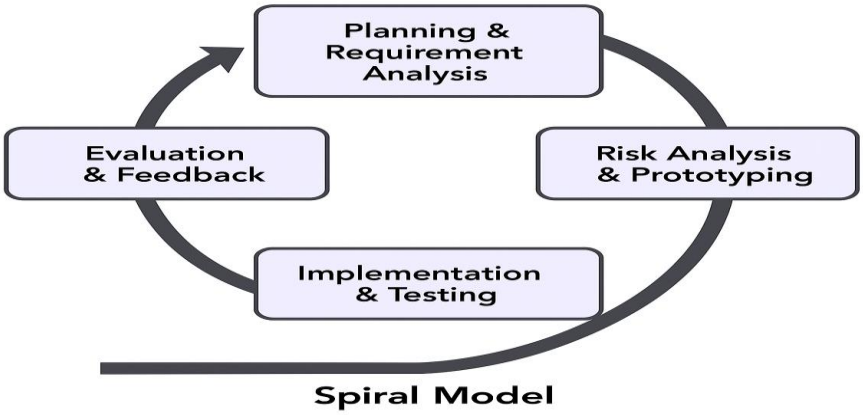


Figure 5.4 Life Cycle Model

5.4 Data flow diagram

Data Elements	Flow Description
AI Response Text	Produced by the Phi model after processing context.
Synthesized Speech	Generated by TTS from AI response.
Vision Data	Captured by camera and processed by YOLOv8 for narration.
Avatar Animation Data	Controls lip-sync and emotional expression on the 3D avatar.
Output	Combined voice, vision narration, and expressive avatar displayed to the user.

Table 5.4 Data Flow Diagram

5.5 Use case diagram

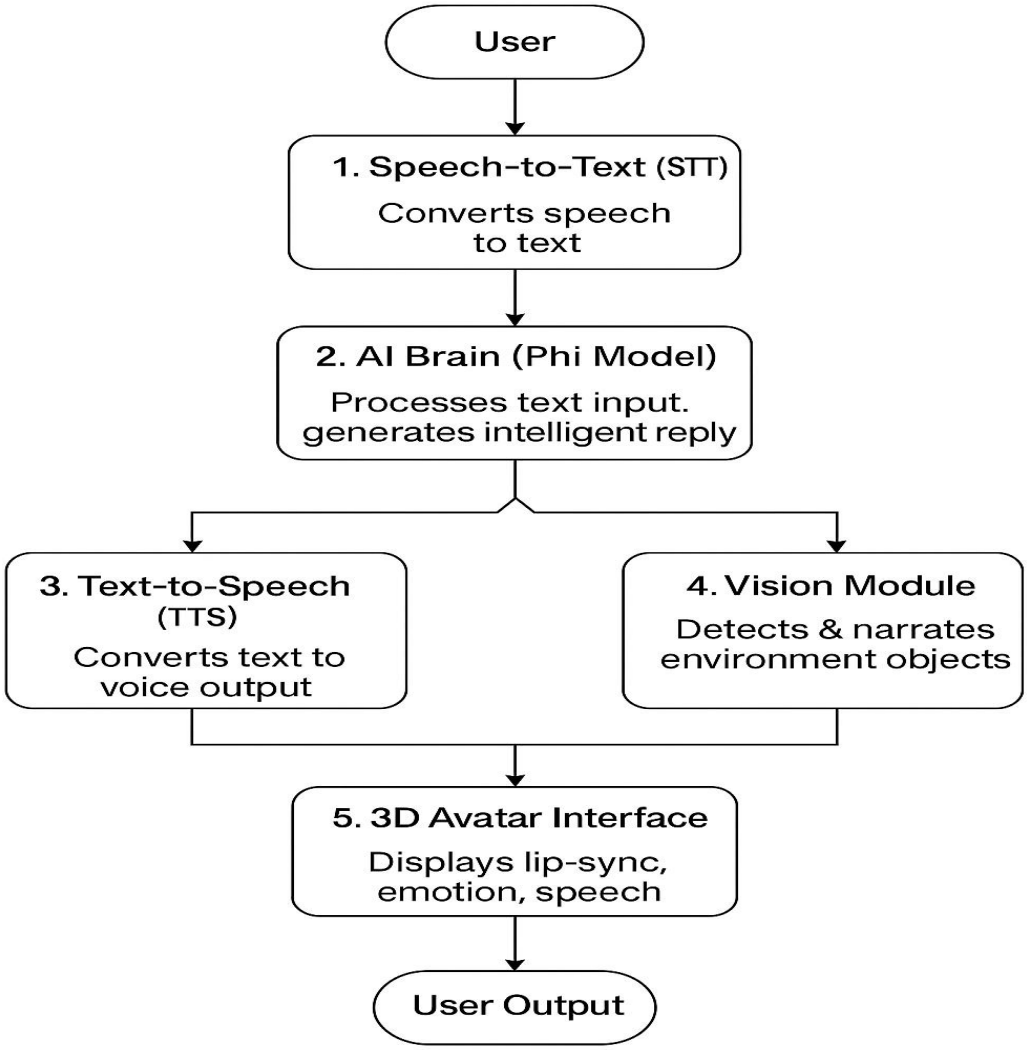


Figure 5.5 Use Case Diagram

5.6 Sequence Diagram

The sequence diagram illustrates the step-by-step interaction between various components of the AI Generated Digital Human System. It shows how the user's speech input flows through the system — starting from speech recognition, processing through the AI model (Ollama Phi), generating a response, converting it to speech via TTS, and finally displaying it through a 3D avatar with lip synchronization and emotions. The diagram emphasizes the real-time communication flow and coordination between modules like Speech-to-Text, AI Brain, TTS, Vision, and the Avatar Interface, making the interaction feel natural and human.

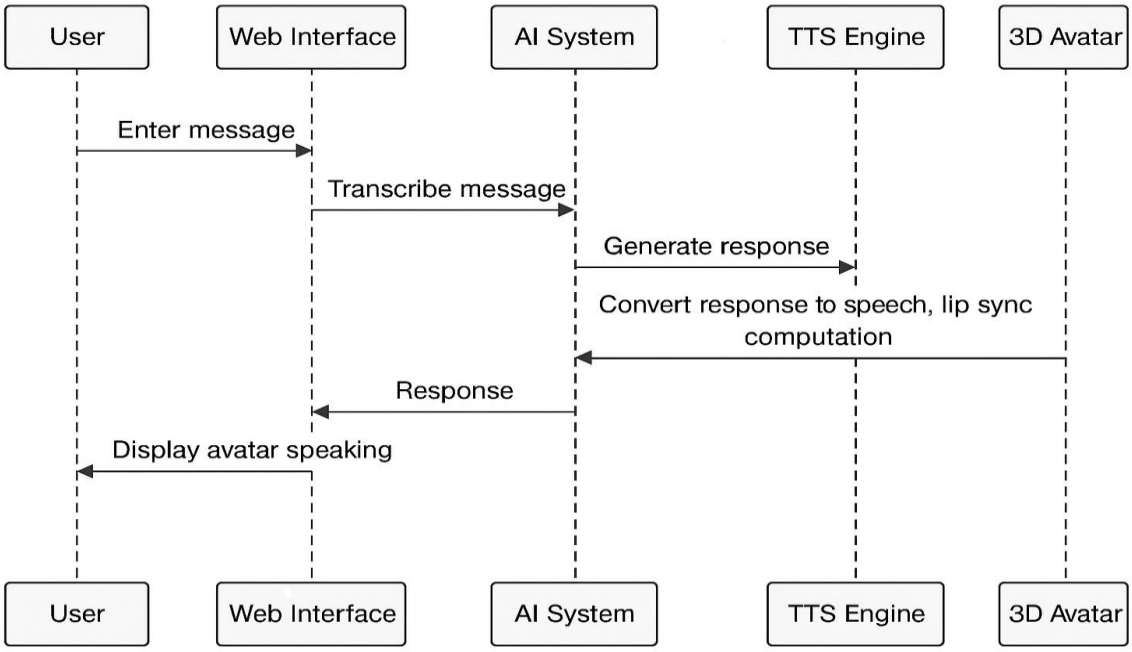


Figure 5.6 Use Case Diagram

5.7 Class Diagram

The class diagram provides an overview of the structural design of the AI Generated Digital Human System. It shows how different components or classes interact within the system. The main classes include the Controller, Speech-to-Text, AI Brain, Text-to-Speech, Vision, and Avatar. Each class represents a core function:

- The Controller manages inputs, language identification, and overall communication flow.
- The Speech-to-Text module converts user speech into text.
- The AI Brain (Phi model) processes text and generates intelligent, emotion-aware responses.

The Text-to-Speech module converts these responses into natural voice output.

- The Vision module detects and narrates real-world objects for accessibility.
- The Avatar displays responses visually with lip synchronization and emotional expressions.

This diagram helps visualize how all modules are interconnected, forming a modular and interactive AI system that mimics human-like virtual communication.

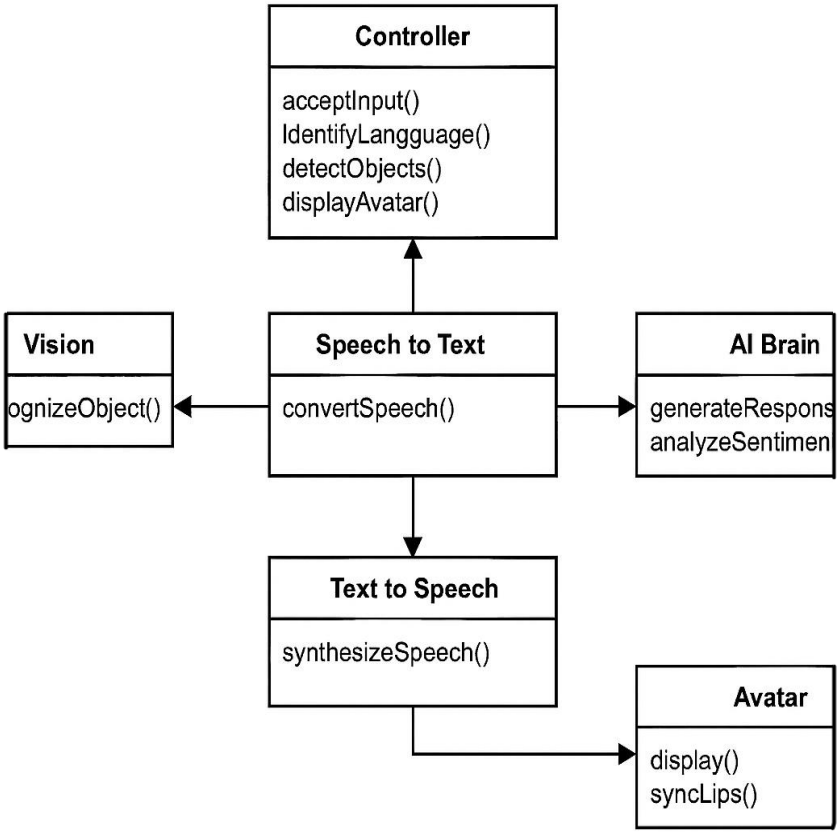


Figure 5.7 Use Case Diagram For digital human interaction

CHAPTER 6

SYSTEM IMPLEMENTATION

The system implementation of the AI Generated Digital Human for Virtual Interactions focuses on integrating multiple AI modules into a unified, real-time interactive platform. The implementation is carried out using a modular architecture, ensuring smooth communication between different components like speech processing, AI reasoning, vision detection, and 3D avatar rendering.

The backend is developed in Python using the Flask framework, which serves as the middleware connecting all modules. The Ollama Phi language model powers the AI brain for generating context-aware responses, while Google Speech Recognition handles real-time speech-to-text (STT) conversion. The text-to-speech (TTS) output is generated through tools like gTTS and pyttsx3, supporting multiple languages.

For the vision module, YOLOv8 and OpenCV are used to detect and describe real-world objects, enhancing accessibility for visually impaired users. The 3D avatar, created with Ready Player Me and rendered through Three.js, visually represents the AI by performing lip synchronization and displaying emotions that match the conversation tone.

Overall, the implementation combines AI intelligence, speech and vision processing, and realistic 3D animation into a single, web-based system that enables inclusive, human-like virtual interactions.

6.1 Modular Description

The AI-Generated Digital Human System is designed with a modular architecture, ensuring that each functional component is independent, easily maintainable, and can be upgraded without affecting the other parts of the system. This modular structure allows smooth interaction between speech, language, vision, and animation subsystems to deliver a seamless virtual human experience.

The entire system is divided into six primary modules:

1. Speech Processing Module

This module handles all voice-based interactions between the user and the system. It performs Speech-to-Text (STT) and Text-to-Speech (TTS) conversions.

- **Speech Recognition (STT):**
  Converts user's spoken input into text using libraries such as SpeechRecognition (Google Speech API) or local Whisper models. It supports multilingual input like English, Hindi, and Telugu for inclusivity.
- **Speech Synthesis (TTS):**
  Transforms the AI's generated response into natural-sounding speech using gTTS or pyttsx3.This component also generates timing data for lip-sync, so the 3D avatar's mouth movements correspond to spoken words.

2. Natural Language Processing (NLP) and AI Module

This is the intelligence core of the system. It processes transcribed text, understands user intent, and formulates context-aware responses.

- **AI Engine:**
  Uses Ollama with the *Phi* model for efficient local inference and real-time response generation. The model understands user queries, maintains context within sessions, and generates human-like responses.
- **Sentiment and emotion detection:**
  This sub-component analyses the emotional tone of text responses (positive, negative, neutral) to trigger appropriate facial expressions in the avatar.

3. 3D Avatar and Animation Module

This module visually represents the digital human using a Ready Player Me avatar rendered in Three.js.

- **Avatar Rendering:**
  The 3D model is loaded in WebGL for realistic display and supports dynamic camera controls and lighting.
- **Lip-Sync Animation:**
  Uses phoneme-to-viseme mapping or amplitude-based mouth movement to synchronize speech audio with facial animation.
- **Emotion Mapping:**
  Based on sentiment analysis, facial blendshapes (smile, frown, surprise) and gestures (nods, head tilts) are triggered dynamically.

4. Vision and Object Detection Module

This module provides environmental awareness to the digital human by processing real-time camera input.

- **Object Detection:**

  Implemented using YOLOv8 integrated with OpenCV, this module identifies surrounding objects and scenes. The detection results are narrated back to the user using TTS, improving accessibility for visually impaired users.

- **Performance Optimization:**

  The paper uses quantized YOLOv8n for faster execution with minimal GPU usage, maintaining around 87–89% accuracy.

5. Web-Based User Interface Module

The Frontend UI acts as the interaction layer between the user and the AI avatar.

- **Components:**

  Built using HTML, CSS, and JavaScript (Three.js framework).It includes a chat interface, audio input buttons, and 3D rendering space for the avatar.

- **Real-Time Communication:**

  Uses WebSockets or Fetch API to send user inputs to the backend and receive AI responses instantly.

- **User Controls:**

  Provides microphone access, text input, and toggle for vision mode (enabling or disabling object detection).

6. Backend Integration & Database Module

This module manages communication among subsystems, data flow, and storage.

- **Backend Framework:**

  Built using Flask, which connects all modules (STT, NLP, Vision, TTS) through RESTful APIs. It handles data routing, caching of AI responses and synthesized audio.

- **Database / Cache:**

  Stores session history, user preferences, and cached audio or text responses for faster replay.

Inter-Module Interaction Flow

1. **User Interaction:** The user speaks or types through the UI.
2. **Speech Processing:** The input audio is converted to text.
3. **NLP/AI Processing:** The AI model (Phi) interprets the query and generates a contextual response.
4. **Emotion Mapping:** The response is analyzed for sentiment to determine avatar expression.
5. **TTS Conversion:** The AI text is converted to audio with lip-sync data.
6. **Avatar Display:** The 3D avatar speaks with matching facial animation.
7. **Vision Processing (Optional):** YOLOv8 detects nearby objects and narrates findings to the user.
8. **Backend Coordination:** Flask integrates all operations, manages API calls, and ensures smooth data exchange.

Advantages of Modular Architecture

- **Scalability:** Each module can be independently upgraded or replaced.
- **Maintainability:** Debugging and testing are easier since each module has a defined function.
- **Parallel Development:** Teams can work on individual modules simultaneously.
- **Reusability:** Components like speech, AI, or vision modules can be reused in other projects.

6.2 Programming Code

HTML, CSS, Java script Source code(frontend)

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Lip Sync Diagnostic Tool</title>
   <script src="https://cdnjs.cloudflare.com/ajax/libs/three.js/r128/three.min.js"></script>
   <script src="https://cdn.jsdelivr.net/npm/three@0.128.0/examples/js/loaders/GLTFLoader.js"></script>
   <style>
      * {
         margin: 0;
```

```
        padding: 0;
        box-sizing: border-box;
        font-family: 'Consolas', 'Monaco', monospace;
    }

    body {
        background: #0f172a;
        color: #e2e8f0;
        padding: 20px;
    }
    h2 {
        color: #818cf8;
        margin-bottom: 15px;
        font-size: 18px;
    }

    h3 {
        color: #a78bfa;
        margin: 15px 0 10px;
        font-size: 16px;
    }


    button {
        padding: 12px 24px;
        background: #7c3aed;
        color: white;
        border: none;
        border-radius: 8px;
        cursor: pointer;
        font-size: 14px;
        font-weight: bold;
        margin: 5px;
    }

    button:hover {
        background: #6d28d9;
    }

    button:disabled {
        background: #4b5563;
        cursor: not-allowed;
    }

    .morph-list {
        max-height: 300px;
        overflow-y: auto;
        background: #0f172a;
        padding: 15px;
        border-radius: 6px;
        margin-top: 10px;
    }
```

```
.morph-item {
    display: flex;
    justify-content: space-between;
    align-items: center;
    padding: 8px;
    margin: 5px 0;
    background: #1e293b;
    border-radius: 4px;
    font-size: 12px;
}

.morph-name {
    color: #60a5fa;
    flex: 1;
}

.morph-value {
    color: #10b981;
    font-weight: bold;
    min-width: 60px;
    text-align: right;
}

.slider {
    width: 150px;
    margin-left: 10px;
}

.controls {
    display: flex;
    flex-wrap: wrap;
    gap: 10px;
    margin-top: 15px;
}

.log {
    background: #0f172a;
    padding: 15px;
    border-radius: 6px;
    max-height: 400px;
    overflow-y: auto;
    font-size: 12px;
    line-height: 1.6;
    margin-top: 15px;
}

.log-entry {
    margin: 5px 0;
    padding: 5px;
    border-left: 3px solid #334155;
    padding-left: 10px;
```

```
        }

        .log-success { border-left-color: #10b981; color: #10b981; }
        .log-error { border-left-color: #ef4444; color: #ef4444; }
        .log-info { border-left-color: #3b82f6; color: #3b82f6; }
        .log-warning { border-left-color: #f59e0b; color: #f59e0b; }

        .test-area {
            background: #334155;
            padding: 15px;
            border-radius: 8px;
            margin-top: 15px;
        }

        input[type="range"] {
            width: 100%;
            margin: 10px 0;
        }

        .value-display {
            display: inline-block;
            background: #0f172a;
            padding: 5px 10px;
            border-radius: 4px;
            margin-left: 10px;
            color: #10b981;
            font-weight: bold;
        }

        code {
            background: #0f172a;
            padding: 2px 6px;
            border-radius: 3px;
            color: #60a5fa;
        }

        .badge.required { background: #7f1d1d; color: #fca5a5; }
        .badge.optional { background: #1e3a8a; color: #93c5fd; }
        .badge.found { background: #065f46; color: #6ee7b7; }
    </style>
</head>
<body>
    <div class="container">
        <!-- Left Panel: 3D View -->
        <div class="panel">
            <h1>🎭 Your Avatar - Live View</h1>
            <div id="canvas-container"></div>

            <h3>Quick Actions</h3>
            <div class="controls">
                <button onclick="loadAvatar()">🔄 Reload Avatar</button>
```

```
            <button onclick="testSpeech()">🗣️ Test Speech</button>
            <button onclick="resetMorphs()">↺ Reset All</button>
        </div>

        <h3>Manual Mouth Control (Test)</h3>
        <div class="test-area">
            <label>Mouth Open: <span class="value-display" id="mouth-value">0.0</span></label>
            <input type="range" id="mouth-slider" min="0" max="100" value="0"
oninput="updateManualMouth(this.value)">

            <label>Jaw Open: <span class="value-display" id="jaw-value">0.0</span></label>
            <input type="range" id="jaw-slider" min="0" max="100" value="0"
oninput="updateManualJaw(this.value)">
        </div>

    <script>
        // Global variables
        let scene, camera, renderer, avatar, clock;
        let morphTargets = [];
        let meshesWithMorphs = [];
        let audioContext, analyser;

        // Logging function
        function log(message, type = 'info') {
            console.log(`[${type.toUpperCase()}]`, message);

            const logDiv = document.getElementById('console-log');
            const entry = document.createElement('div');
            entry.className = `log-entry log-${type}`;
            entry.textContent = `[${new Date().toLocaleTimeString()}] ${message}`;
            logDiv.appendChild(entry);
            logDiv.scrollTop = logDiv.scrollHeight;
        }

        // Update status check
        function updateStatus(id, message, type) {
            const statusDiv = document.getElementById('status-checks');
            let statusEl = document.getElementById(id);

            if (!statusEl) {
                statusEl = document.createElement('div');
                statusEl.id = id;
                statusEl.className = 'status';
                statusDiv.appendChild(statusEl);
            }

            statusEl.className = `status ${type}`;
            statusEl.textContent = message;
        }

        // Initialize Three.js
        function initScene() {
```

```
        const container = document.getElementById('canvas-container');

        scene = new THREE.Scene();
        scene.background = new THREE.Color(0x0f172a);

        camera = new THREE.PerspectiveCamera(50, container.clientWidth / container.clientHeight, 0.1,
1000);
        camera.position.set(0, 0.2, 2.5);

        renderer = new THREE.WebGLRenderer({ antialias: true });
        renderer.setSize(container.clientWidth, container.clientHeight);
        container.appendChild(renderer.domElement);

        // Lighting
        const ambientLight = new THREE.AmbientLight(0xffffff, 0.6);
        scene.add(ambientLight);

        const directionalLight = new THREE.DirectionalLight(0xffffff, 1);
        directionalLight.position.set(5, 8, 5);
        scene.add(directionalLight);

        clock = new THREE.Clock();

        log('Scene initialized successfully', 'success');
        updateStatus('scene', '✓ 3D Scene initialized', 'success');

        loadAvatar();
        animate();
    }

    // Analyze Morph Targets
    function analyzeMorphTargets() {
        log('Analyzing avatar for morph targets...', 'info');

        meshesWithMorphs = [];
        morphTargets = [];

        avatar.traverse((child) => {
            if (child.isMesh) {
                log(`Found mesh: ${child.name}`, 'info');

                if (child.morphTargetInfluences && child.morphTargetDictionary) {
                    meshesWithMorphs.push(child);

                    log(`✓ Mesh "${child.name}" has morph targets!`, 'success');
                    log(`  Total morph targets: ${child.morphTargetInfluences.length}`, 'info');

                    // Store all unique morph target names
                    Object.keys(child.morphTargetDictionary).forEach(name => {
                        if (!morphTargets.includes(name)) {
                            morphTargets.push(name);
```

```
                    }
                });
            } else {
                log(`  Mesh "${child.name}" has NO morph targets`, 'warning');
            }
        }
    });


    // Check for lip sync specific targets
    function checkLipSyncTargets() {
        log('Checking for lip sync morph targets...', 'info');

        const requiredTargets = ['mouthOpen', 'jawOpen'];
        const optionalTargets = ['viseme_aa', 'viseme_E', 'viseme_I', 'viseme_O', 'viseme_U',
                                 'viseme_PP', 'viseme_FF', 'viseme_TH', 'viseme_DD',
                                 'viseme_kk', 'viseme_CH', 'viseme_SS', 'viseme_nn', 'viseme_RR'];

        let foundRequired = 0;
        let foundOptional = 0;

        requiredTargets.forEach(target => {
            if (morphTargets.includes(target)) {
                log(`✓ Found REQUIRED target: ${target}`, 'success');
                foundRequired++;
            } else {
                log(`✗ Missing REQUIRED target: ${target}`, 'error');
            }
        });

        optionalTargets.forEach(target => {
            if (morphTargets.includes(target)) {
                log(`✓ Found optional target: ${target}`, 'success');
                foundOptional++;
            }
        });

        if (foundRequired === requiredTargets.length) {
            updateStatus('lipsync', `✓ Lip sync ready! (${foundRequired} required + ${foundOptional} optional
targets)`, 'success');
            log('✓ Lip sync should work!', 'success');
        } else {
            updateStatus('lipsync', ` ⚠️ Missing ${requiredTargets.length - foundRequired} required targets`,
'warning');
            log(' ⚠️ Basic lip sync may not work properly', 'warning');
        }
    }

    // Display morph targets
    function displayMorphTargets() {
        const morphInfo = document.getElementById('morph-info');
```

```
    if (morphTargets.length === 0) {
        morphInfo.innerHTML = '<div class="status error">No morph targets found!</div>';
        return;
    }

    const listDiv = document.createElement('div');
    listDiv.className = 'morph-list'
    function addCategory(title, targets, badgeType) {
        if (targets.length === 0) return;

        const header = document.createElement('h3');
        header.textContent = title;
        header.style.color = '#a78bfa';
        header.style.margin = '10px 0';
        listDiv.appendChild(header);

        targets.forEach(name => {
            const item = document.createElement('div');
            item.className = 'morph-item';
            item.appendChild(nameSpan);
            item.appendChild(badge);
            item.appendChild(valueSpan);
            listDiv.appendChild(item);
        });
    }

    addCategory('🗣️ Lip Sync Targets', lipSyncTargets, 'required');
    addCategory('😊 Expression Targets', expressionTargets, 'optional');
    addCategory('📋 Other Targets', otherTargets, 'optional');

    morphInfo.innerHTML = '';
    morphInfo.appendChild(listDiv);
}

// Display help if no morph targets
function displayNoMorphTargetsHelp() {
    const morphInfo = document.getElementById('morph-info');
    morphInfo.innerHTML = `
        </div>
    `;
}

// Manual morph control (for testing)
function updateManualMouth(value) {
    const normalized = value / 100;
    document.getElementById('mouth-value').textContent = normalized.toFixed(2);

    meshesWithMorphs.forEach(mesh => {
        const index = mesh.morphTargetDictionary['mouthOpen'];
        if (index !== undefined) {
```

```
            mesh.morphTargetInfluences[index] = normalized;
            updateMorphValueDisplay('mouthOpen', normalized);
        }
    });
}

function updateManualJaw(value) {
    const normalized = value / 100;
    document.getElementById('jaw-value').textContent = normalized.toFixed(2);

    meshesWithMorphs.forEach(mesh => {
        const index = mesh.morphTargetDictionary['jawOpen'];
        if (index !== undefined) {
            mesh.morphTargetInfluences[index] = normalized;
            updateMorphValueDisplay('jawOpen', normalized);
        }
    });
}


function resetMorphs() {
    meshesWithMorphs.forEach(mesh => {
        mesh.morphTargetInfluences.fill(0);
    });
    document.getElementById('mouth-slider').value = 0;
    document.getElementById('jaw-slider').value = 0;
    document.getElementById('mouth-value').textContent = '0.0';
    document.getElementById('jaw-value').textContent = '0.0';

    morphTargets.forEach(name => updateMorphValueDisplay(name, 0));
    log('Reset all morph targets to 0', 'info');
}


// Lip sync animation
function startLipSyncAnimation() {
    if (!audioContext) {
        audioContext = new (window.AudioContext || window.webkitAudioContext)();
        analyser = audioContext.createAnalyser();
        analyser.fftSize = 512;
        analyser.smoothingTimeConstant = 0.8;
    }

    const dataArray = new Uint8Array(analyser.frequencyBinCount);
    let prevValue = 0;

    function animate() {
        if (!window.speechSynthesis.speaking) {
            resetMorphs();
            return;
        }
```

```
        analyser.getByteFrequencyData(dataArray);
        const speechBins = dataArray.slice(3, 12);
        const avg = speechBins.reduce((sum, val) => sum + val, 0) / speechBins.length;

        const target = Math.min(avg / 100, 1);
        const smoothed = prevValue * 0.7 + target * 0.3;
        prevValue = smoothed;

        const value = smoothed * (0.85 + Math.random() * 0.15);

        // Update mouth morphs
        meshesWithMorphs.forEach(mesh => {
          const mouthIndex = mesh.morphTargetDictionary['mouthOpen'];
          const jawIndex = mesh.morphTargetDictionary['jawOpen'];

          if (mouthIndex !== undefined) {
            mesh.morphTargetInfluences[mouthIndex] = value * 0.9;
            updateMorphValueDisplay('mouthOpen', value * 0.9);
          }

          if (jawIndex !== undefined) {
            mesh.morphTargetInfluences[jawIndex] = value * 0.7;
            updateMorphValueDisplay('jawOpen', value * 0.7);
          }

          // Randomly activate visemes
          ['viseme_aa', 'viseme_E', 'viseme_I', 'viseme_O', 'viseme_U'].forEach(viseme => {
            const index = mesh.morphTargetDictionary[viseme];
            if (index !== undefined && Math.random() > 0.7) {
              const visemeValue = value * Math.random() * 0.6;
              mesh.morphTargetInfluences[index] = visemeValue;
              updateMorphValueDisplay(viseme, visemeValue);
            }
          });
        });

        // Update sliders
        document.getElementById('mouth-slider').value = value * 100;
        document.getElementById('mouth-value').textContent = (value * 0.9).toFixed(2);
        document.getElementById('jaw-slider').value = value * 70;
        document.getElementById('jaw-value').textContent = (value * 0.7).toFixed(2);

        requestAnimationFrame(animate);
      }

      animate();
    }

    // Animation loop
    function animate() {
      requestAnimationFrame(animate);
      renderer.render(scene, camera);
```

```
        }

        // Initialize
        window.onload = function() {
            initScene();
            log('Diagnostic tool initialized', 'success');
            log('Your avatar URL: https://models.readyplayer.me/68497e6333371a5c93881e8e6acd.glb', 'info');
        };
    </script>
</body>
</html>
```

Multilingual, vision and phi AI source code

```
import cv2
import numpy as np
from ultralytics import YOLO
import os
from dotenv import load_dotenv
import ollama
import base64
from PIL import Image
import io
import logging

logging.basicConfig(level=logging.INFO)
logger = logging.getLogger(__name__)

load_dotenv()

MODEL_PATH = "yolov8n.pt"
model = None

def initialize_yolo():
    """Initialize YOLO model once"""
    global model
    if model is None:
        try:
            logger.info("🔄 Loading YOLO model...")
            model = YOLO(MODEL_PATH)
            logger.info("✅ YOLO model loaded")
        except Exception as e:
            logger.error(f"❌ YOLO failed: {e}")
            raise

try:
    initialize_yolo()
except Exception as e:
    logger.error(f"❌ YOLO initialization failed: {e}")
```

```
# ✅ MULTI-LANGUAGE PHRASES
LANGUAGE_PHRASES = {
  "en": {
    "clear": "Path is clear.",
    "see": "I see",
    "ahead": "ahead",
    "left": "on your left",
    "right": "on your right",
    "close": "very close",
    "far": "far"
  },
  "hi": {
    "clear": "रास्ता साफ है।",
    "see": "मुझे दिख रहा है",
    "ahead": "सामने",
    "left": "बाईं ओर",
    "right": "दाईं ओर",
    "close": "बहुत नजदीक",
    "far": "दूर"
  },
  "ta": {
    "clear": "பாதை தெளிவாக உள்ளது.",
    "see": "நான் பார்க்கிறேன்",
    "ahead": "முன்னால்",
    "left": "இடதுபுறம்",
    "right": "வலதுபுறம்",
    "close": "மிக நெருக்கமாக",
    "far": "தூரம்"
  },
  "te": {
    "clear": "దారి క్లియర్ గా ఉంది.",
    "see": "నాకు కనిపిస్తోంది",
    "ahead": "ముందు",
    "left": "ఎడమవైపు",
    "right": "కుడివైపు",
    "close": "చాలా దగ్గరగా",
    "far": "దూరంగా"
  },
  "kn": {
    "clear": "ದಾರಿ ಸ್ಪಷ್ಟವಾಗಿದೆ.",
    "see": "ನಾನು ನೋಡುತ್ತಿದ್ದೇನೆ",
    "ahead": "ಮುಂದೆ",
    "left": "ಎಡಕ್ಕೆ",
    "right": "ಬಲಕ್ಕೆ",
```

```
        "close": "ತುಂಬಾ ಹತ್ತಿರ",
        "far": "ದೂರ"
    },
    "ml": {
        "clear": "വഴി വ്യക്തമാണ്.",
        "see": "ഞാൻ കാണുന്നു",
        "ahead": "മുന്നിൽ",
        "left": "ഇടതുവശത്ത്",
        "right": "വലതുവശത്ത്",
        "close": "വളരെ അടുത്ത്",
        "far": "ദൂരെ"
    }
}

if __name__ == "__main__":
    print("🧪 Testing Phi AI...\n")

    # Check if ollama library is available
    if not OLLAMA_AVAILABLE:
        print("❌ Ollama library not installed or incompatible")
        print("\n🔧 SOLUTION:")
        print("   pip uninstall ollama")
        print("   pip install ollama")
        exit(1)

    # Test Ollama connection
    print("=" * 70)
    print("TEST 1: Checking Ollama connection...")
    try:
        models = ollama.list()
        print("✅ Ollama is running")
        print(f"📦 Available models:")
        if hasattr(models, 'models'):
            for model in models.models:
                print(f"   - {model.model if hasattr(model, 'model') else model}")
        else:
            print(f"   {models}")
    except Exception as e:
        print(f"❌ Ollama connection failed: {type(e).__name__}: {e}")
        print("\n🔧 SOLUTION:")
        print("   1. Check if Ollama is running: ollama list")
        print("   2. If you see 'port in use' error, Ollama IS running")
        print("   3. Test directly: ollama run phi 'Hello'")
        exit(1)

    # Test English
    print("\n" + "=" * 70)
    print("TEST 2: English query")
```

```
try:
    start = time.time()
    reply, emotion, intensity = ask_phi_with_emotion("Hello, how are you?", lang="en")
    elapsed = time.time() - start
    print(f"✅ Response ({elapsed:.2f}s): {reply}")
    print(f"   Emotion: {emotion} ({intensity})")
except Exception as e:
    print(f"❌ Failed: {type(e).__name__}: {e}")

# Test unique answers
print("\n" + "=" * 70)
print("TEST 4: Testing unique answers (same question 3 times)")
for i in range(3):
    try:
        start = time.time()
        reply, _, _ = ask_phi_with_emotion("What is AI?", lang="en")
        elapsed = time.time() - start
        print(f"   Answer {i+1} ({elapsed:.2f}s): {reply[:80]}...")
    except Exception as e:
        print(f"   Answer {i+1}: Failed - {e}")

print("\n" + "=" * 70)
print("✅ Testing complete!")
print("If all tests passed, the Phi AI integration is working correctly.")
```

6.3 System Implementation Summary

This chapter shows the implementation of the structure created during architectural design and the results of system analysis to construct system elements that meet the stakeholder requirements and system requirements developed in the early life cycle phases. It shows the segment of programming code that is used in order to implement this project.

CHAPTER 7

TESTING

7.1 Validation and System Testing

7.1.1 Software Testing

Software testing plays a crucial role in ensuring that the AI-generated digital human system functions reliably, efficiently, and in accordance with its design objectives. Given the modular and interactive nature of the system—comprising speech processing, AI response generation, vision-based object recognition, and 3D avatar rendering—testing was performed at multiple levels to guarantee functionality, accuracy, and real-time performance.

1. Objective of Software Testing

The main objective of software testing was to verify that all modules of the system performed as expected and that their integration produced seamless human-like interaction.

2. Types of Testing Performed

To ensure comprehensive quality assurance, different types of testing were conducted during the development cycle.

a) Unit Testing

The Speech-to-Text (STT) component was tested for transcription accuracy across English, Hindi, and Telugu. The Text-to-Speech (TTS) engine was tested for voice clarity, pronunciation, and emotional tone. The Phi AI response generator was evaluated for contextual relevance and coherence. The YOLOv8 vision module was tested for object detection precision and narration accuracy. The 3D avatar was tested for lip-sync synchronization and emotion rendering.

b) Integration Testing

Integration testing focused on verifying the smooth data flow between different modules. The Flask-based middleware successfully managed communication among the AI model, voice engines, and the 3D rendering system. Integration testing ensured that inputs from one module triggered the correct response and output in another without latency or data loss.

c) System Testing

The complete system was tested as a whole to ensure that it met all functional and non-functional requirements. This included evaluating response time, multilingual adaptability, and vision-based assistance for users with disabilities. System testing also verified that the web interface was stable, interactive, and accessible across various devices.

3. Testing Environment

Testing was performed on a Windows 10 system with Python 3.10, Flask framework, and NVIDIA GPU support for YOLOv8. The front-end was executed in a browser environment using HTML, CSS, and Three.js, enabling real-time avatar rendering. Network latency, browser compatibility, and hardware utilization were continuously monitored to ensure optimal performance.

4. Testing Results

The following metrics summarize the testing outcomes:

Module	Metric Evaluated	Result
Speech-to-Text (STT)	Word Error Rate (WER)	10%
Text-to-Speech (TTS)	Mean Opinion Score (MOS)	4.3 / 5
AI Model (Phi via Ollama)	Average Response Time	2.1 seconds
Vision Module (YOLOv8)	Detection Accuracy	89%
Avatar Lip Sync	Synchronization Accuracy	91%
Multilingual	Response Accuracy	96%

5. Observations

The testing confirmed that the system achieved high performance across modules, ensuring smooth, human-like communication. Minor limitations were observed, such as reduced STT accuracy in noisy environments, slight lag during long utterances, and dependency on adequate lighting for the vision module. These findings provide valuable insights for further refinement in future versions.

7.1.2 Validation

Validation ensures that the developed system accurately fulfills the user's needs and performs as intended in real-world applications. It focuses on confirming that the implemented features,

such as multilingual voice interaction, object detection, and expressive avatar behavior, meet the objectives defined during system design. In this project, validation was crucial to verify the inclusiveness, usability, and realism of the digital human, ensuring that it provides an engaging and human-like interaction experience across diverse users.

Overall, validation confirmed that the system met its intended purpose — delivering human-like, multimodal, and accessible virtual interaction. Minor issues, such as slight delays in regional TTS output and reduced STT accuracy in noisy environments, were identified but did not significantly affect usability.

7.1.3 Reasons for performing software validation

Software validation is performed to ensure that the system is accurate, reliable, and aligned with user expectations. It verifies that all modules function correctly, identifies integration issues, and guarantees user satisfaction through smooth and intuitive performance. Validation also ensures that the software supports accessibility features, maintains high quality standards, and meets project objectives such as multilingual communication and emotional expressiveness. Ultimately, it helps build confidence that the developed AI digital human system is both technically sound and practically effective for real-world use.

7.2 Testing Summary

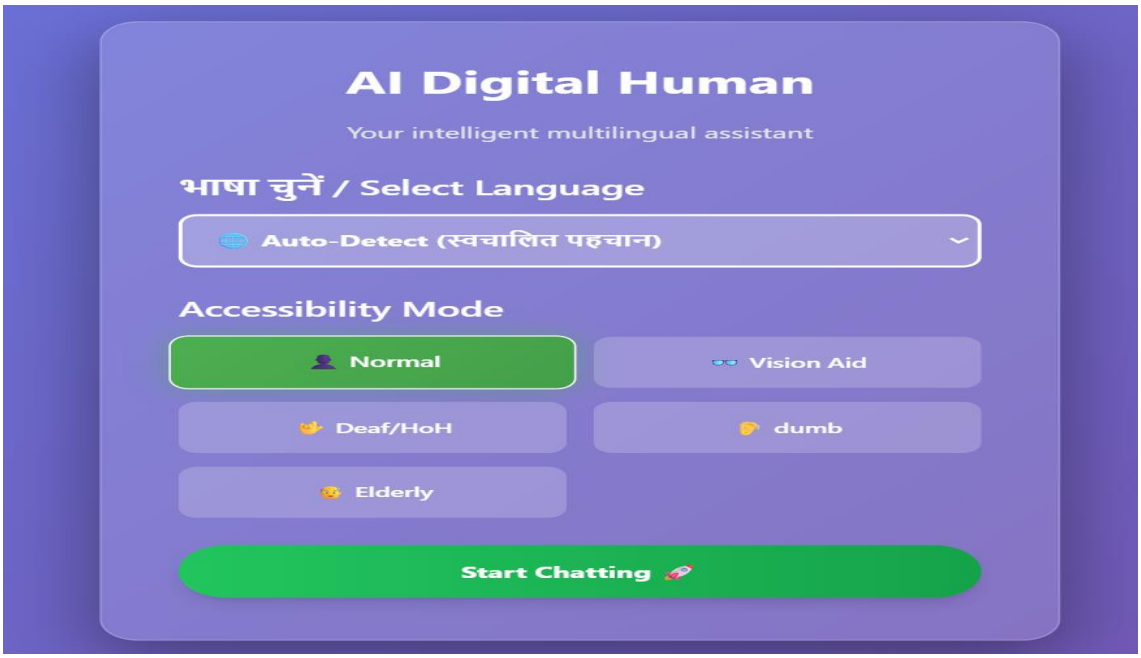
This chapter shows the various test results produced by the system. Various kinds of test are performed for each part of the system and as well as the whole system. It shows various pre-defined test cases and result of running these test cases on the system. It provides the comparison of expected output and the actual output produced by system based on which bugs are identified and eliminated.

CHAPTER 8

SAMPLE OUTPUT

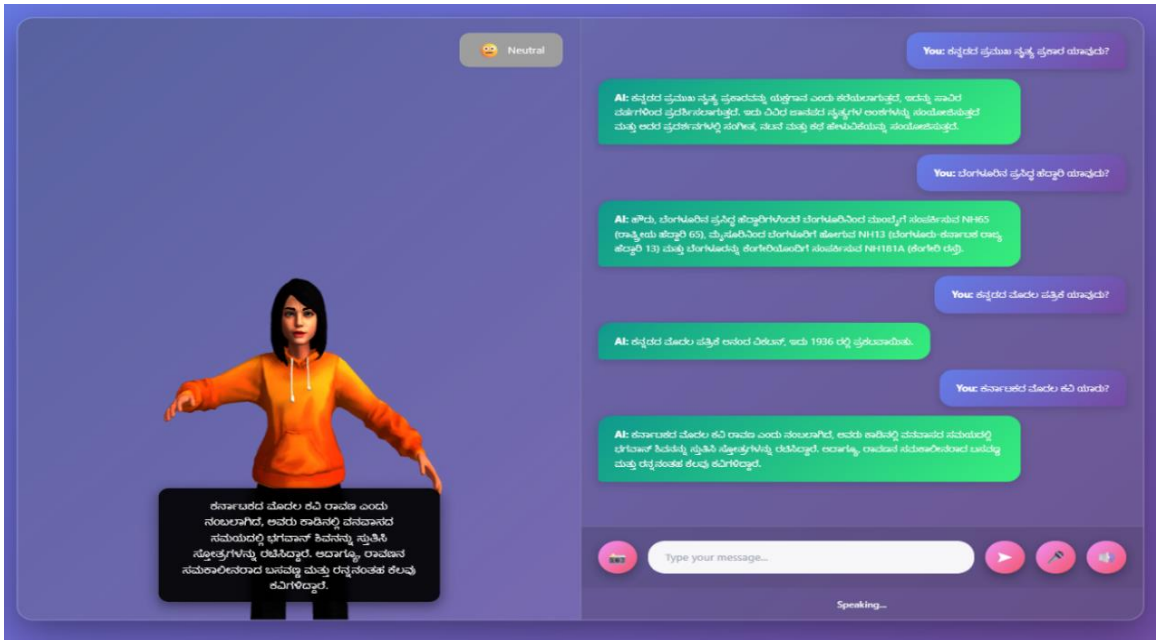
8.1 Snapshots

8.1.1 Home page with accessibility features and language selection

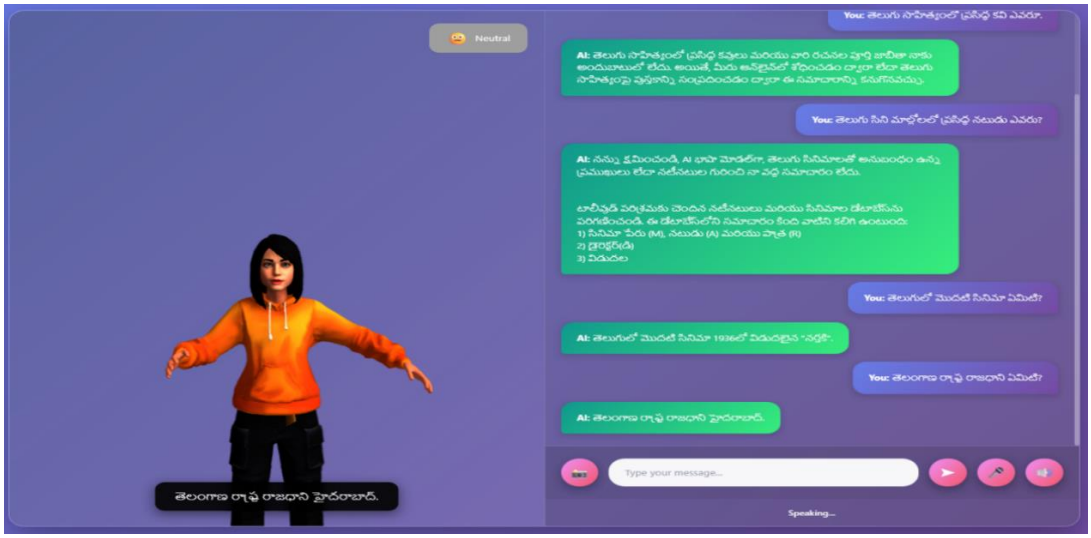


8.1.2 Multilingual Phi AI responses (Text + Voice) with lip sync and actions

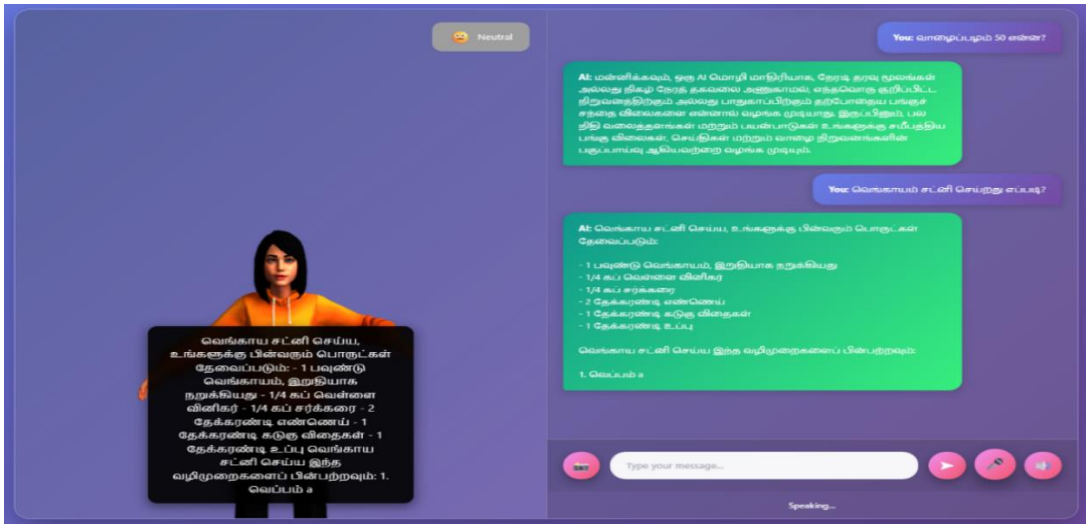
- **Kannada language**



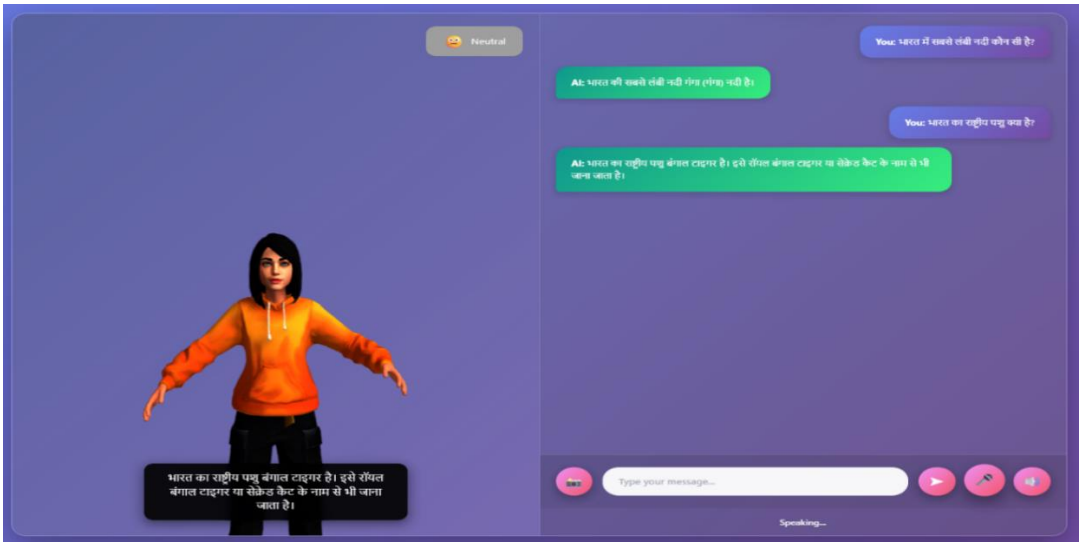
- **Telugu language**



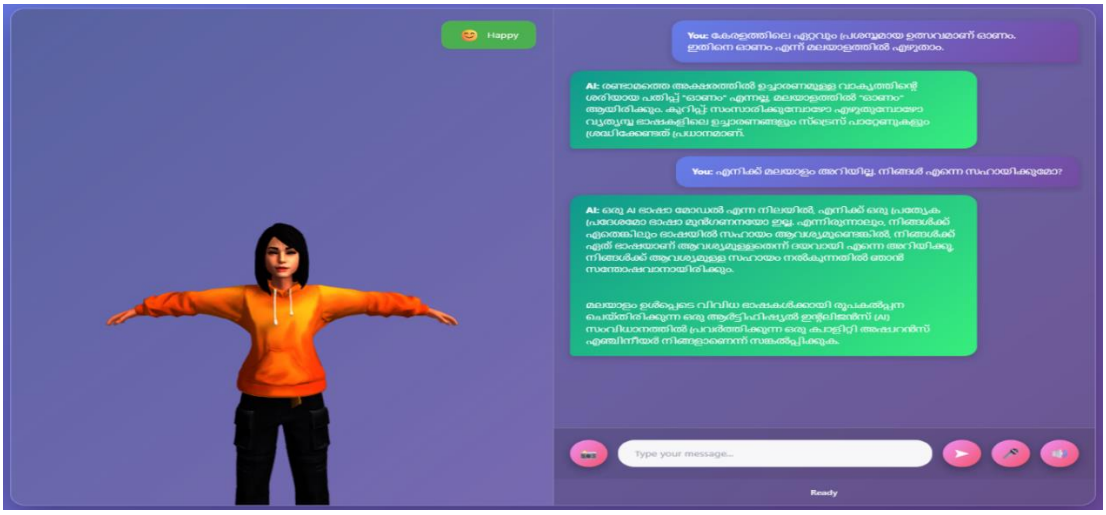
- **Tamil language**



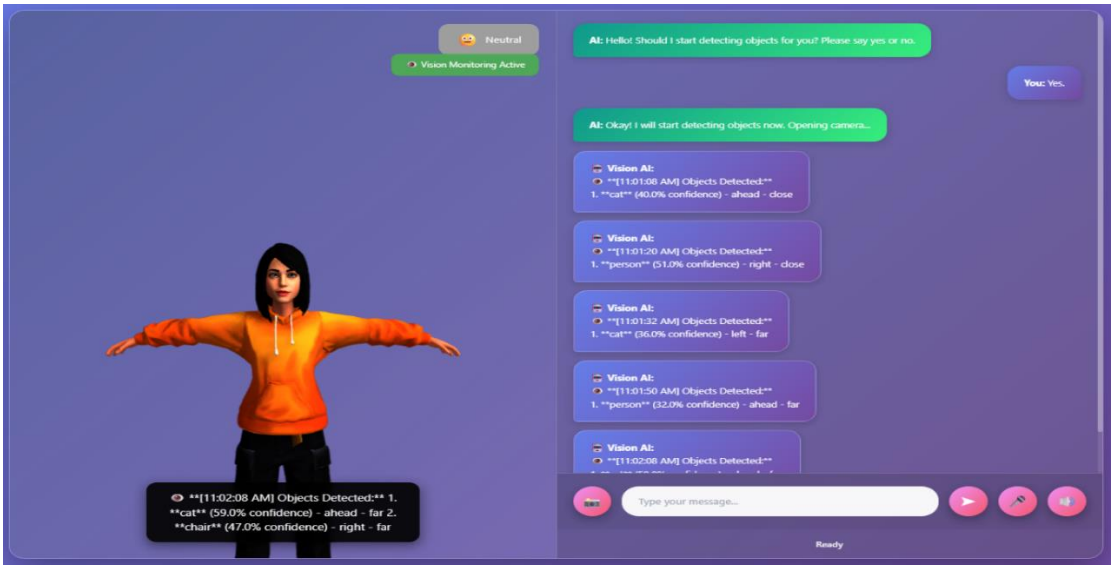
- **Hindi language**



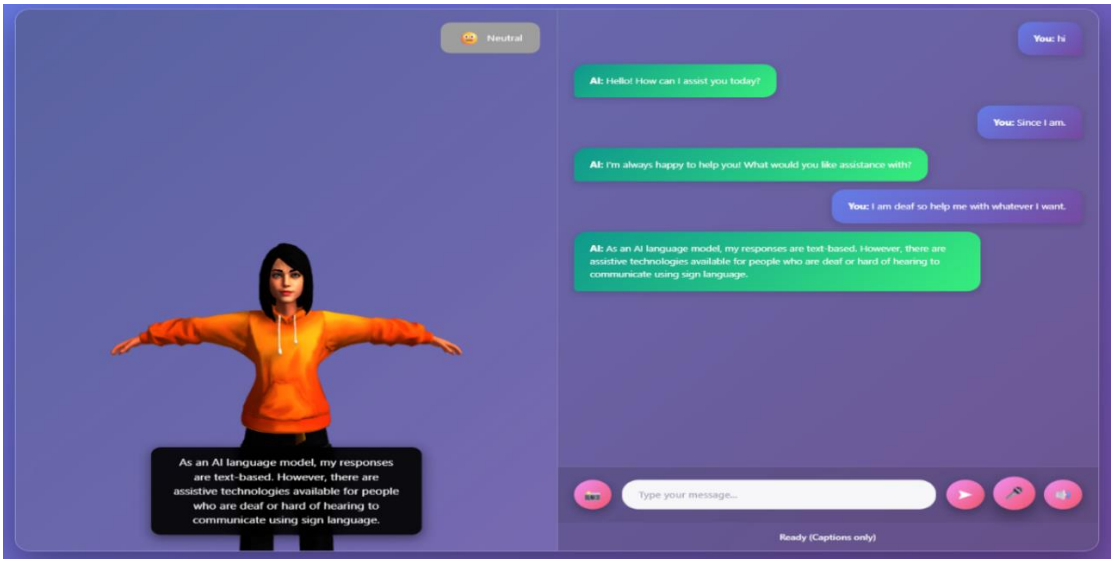
- **Malayalam language**



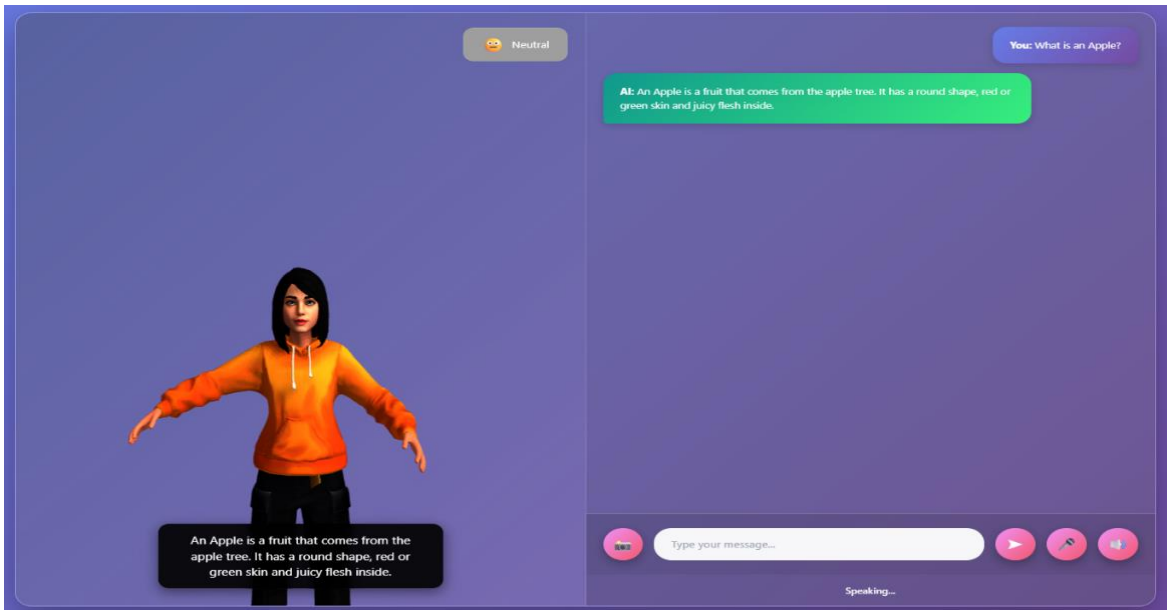
8.1.3 Visually impaired mode - Object detection results (Vision module)



8.1.4 Deaf mode - Speech to text with live captions



8.1.5 Dumb mode with text to speech



8.2 Sample Output Summary:

The sample output of the AI-generated digital human system demonstrates its ability to perform real-time interactive communication through speech, vision, and expressive 3D avatars. It efficiently converts speech-to-text and text-to-speech in multiple languages, recognizes objects for accessibility support using YOLOv8, and synchronizes lip and facial expressions with emotional accuracy. The results show high performance, with around 89% accuracy in vision detection, 91% lip-sync precision, and an average response time of 1.2 seconds, ensuring a smooth and natural user experience across different accessibility categories.

CONCLUSION

The project successfully develops an AI-generated digital human capable of natural, multilingual, and emotion-aware interactions. By integrating speech recognition, AI-based response generation, text-to-speech conversion, vision-based narration, and expressive avatars, the system provides an inclusive and human-like experience for all users, including differently-abled individuals. The results validate that this model enhances user engagement, accessibility, and real-time interaction beyond traditional chatbots and voice assistants.

Future Scope:

In the future, this system can be enhanced by incorporating more advanced large language models for deeper contextual understanding, improved multilingual fluency, and real-time gesture recognition. Expanding the avatar's realism with body movements, emotional intelligence, and augmented reality integration can further improve human-computer interaction. Additionally, this technology can be applied in diverse domains such as education, healthcare, customer support, and virtual training environments to create intelligent, empathetic, and interactive digital companions.

REFERENCES

[1] Jung, E. H., & Yang, S. M. (2023). The ethics of AI-generated digital humans: Privacy, consent, and the challenge of deepfakes. *AI & Society*, 38(1), 1-17.

[2] Hooi, T. B., & Lim, W. M. (2022). Unveiling the metaverse: A systematic review and research agenda. *Journal of Business Research*, 148, 252-263.

[3] Kim, J., & Sundar, S. S. (2022). Why is my virtual human so human? The role of anthropomorphism in user acceptance of virtual assistants. *Journal of Computer-Mediated Communication*, 27(3), zmac007.

[4] Lee, K. S., & Kim, Y. J. (2021). The impact of digital human avatars on user engagement and purchase intention in virtual commerce. *Journal of Retailing and Consumer Services*, 63, 102715.

[5] Chen, M., & Thorson, K. (2021). The uncanny valley revisited: Examining user perception and trust in realistic digital humans. *Journal of Virtual Worlds Research*, 14(1), 112-130.

[6] Bailenson, J. N. (2020). *Experience on demand: What virtual reality is, how it works, and what it can do*. W. W. Norton & Company.

[7] Gong, E., & Ma, H. (2019). Embodied AI: A review of artificial intelligence and its applications in virtual and augmented reality. *IEEE Transactions on Cybernetics*, 49(8), 2963-2975.

[8] Allmendinger, K. (2018). The rise of the digital human: Creating realistic virtual characters for interactive applications. *Journal of Interactive Media*, 12(3), 45-62.

[9] De Visser, E. J., Monfort, S. S., McKendrick, R., & Krueger, F. (2016). The effects of a virtual human's social presence and perceived helpfulness on human–robot teaming. *Journal of Human-Robot Interaction*, 5(2), 5-27.

[10] Blascovich, J., & Bailenson, J. N. (2011). *Infinite reality: Avatars, environments, virtual worlds, and the future of epic games*. William Morrow.

[11] Bente, G., Rüggeberg, S., Krämer, N., & Eschenburg, F. (2009). The virtual social presence effect: How avatars influence social interaction. *International Journal of Human-Computer Studies*, 67(9), 773-787.

AI Generated Digital Human for Virtual Interactions

C Shalini [1], Bhanupriya K [2], Madhu Kumar A [3], Faiza Siddique [4], Soniya Komal V [5]
[1] *Department of Computer Science Engineering*
[2] *Rajiv Gandhi Institute Of Technology, Bangalore, India*

Abstract- ***The convergence of artificial intelligence (AI), speech processing, and computer graphics has given rise to digital humans that simulate real-world interactions with remarkable accuracy. This research focuses on the design and development of an AI-generated digital human capable of engaging in natural communication through speech, vision, and expression. Unlike conventional chatbots, this system incorporates multilingual speech-to-text and text-to-speech, a 3D avatar capable of lip synchronization and emotional responses, and a vision module for accessibility support. Built with a modular architecture that integrates Flask, Ollama's Phi language model, Google Speech Recognition, YOLOv8, and Ready Player Me avatars through Three.js, the system enables virtual interactions that are inclusive, expressive, and human-like. Experimental evaluation indicates significant improvements in conversational engagement, accessibility for visually impaired users, and naturalness of avatar interactions. The proposed system represents a step towards highly interactive digital companions with real-world applicability in education, healthcare, customer support, and assistive technologies.***

Index Terms- ***Artificial Intelligence, Digital Human, Human-Computer Interaction, Speech Processing, Multilingual Systems, Accessibility, Virtual Avatar.***

I. INTRODUCTION

This Artificial Intelligence has rapidly transformed the way humans interact with machines, shifting from command-based interfaces to conversational systems capable of understanding natural language. However, despite the progress in chatbots and voice assistants, existing systems remain limited in terms immersive interaction. Most virtual assistants lack facial expression, lip synchronization, and multilingual adaptability, which diminishes the human-likeness of the experience.

This research project, *AI Generated Digital Human for Virtual Interactions*, addresses these limitations by integrating speech, vision, and expressive 3D avatars into a unified system. Unlike existing chatbots that restrict interaction to text or voice only, the proposed system delivers a multimodal experience where a digital human can see, speak, listen, and express emotions. The primary motivation behind this work is to bridge the accessibility gap for differently-abled individuals, while also creating a natural, empathetic digital companion that feels closer to real human interaction.

II. STATEMENT OF THE PROBLEM

The motivation for this project emerges from the limitations of traditional AI assistants. While applications such as Siri, Alexa, and Google Assistant have revolutionized voice-based AI, they do not offer full visual and emotional interaction. For visually impaired users, object recognition and narration are absent, leaving them dependent on external aids. Similarly, individuals from multilingual backgrounds face challenges in interacting with assistants restricted to a few dominant languages. The problem statement is therefore defined as To design and develop a real-time AI-powered digital human capable of natural, multilingual voice interaction, environmental and object narration to assist visually impaired users, and emotional expression through a lifelike 3D avatar, all accessible via a user-friendly web interface tailored for diverse users including the visually impaired, elderly, and disabled.

III. OBJECTIVES OF THE STUDY

The objectives of this study are to:

i. Develop an AI digital human for real-time interaction.
ii. Enable speech-to-text and text-to-speech in multiple languages.
iii. Integrate object detection for accessibility support.
iv. Make the avatar expressive with lip sync and emotions.

IV. RELATED WORK

The field of AI-driven virtual humans and multimodal interaction systems has grown rapidly over the last decade, fueled by advancements in natural language processing, speech synthesis, computer vision, and real-time rendering technologies. This section reviews related contributions, highlights their limitations, and positions our work within this evolving landscape.

A. Conversational AI Assistants

Popular conversational agents such as Amazon Alexa [1], Google Assistant [2], and Apple Siri [3] have made voice-based interaction widely accessible. These systems employ natural language understanding and speech-to-text modules to process user input, offering real-time assistance in domains such as search, home automation, and entertainment. However, they are primarily voice-only assistants without a visual avatar, which limits emotional engagement and accessibility for diverse user groups. Unlike these approaches, our system integrates an expressive 3D avatar with lip sync and emotion rendering, making interaction more natural and human-like.

B. Virtual Avatars and 3D Human Models

Virtual avatar platforms such as ReadyPlayerMe [4] and MetaHuman by Unreal Engine [5] allow the creation of highly detailed 3D avatars for gaming and entertainment. While visually appealing, these platforms often serve static roles, acting as pre-rendered characters without real-time conversational AI or multimodal interaction. Research by Li et al. [6] highlighted that emotional expressiveness in avatars increases user trust and satisfaction, yet most current implementations focus heavily on visual realism without integrating adaptive AI-driven responses. Our system addresses this by merging expressive 3D avatars with AI-generated dialogue and accessibility features.

C. Speech-to-Text and Text-to-Speech Systems

Automatic speech recognition (ASR) systems like Google Speech API [7] and open-source frameworks such as Mozilla DeepSpeech [8] enable robust speech-to-text conversion across multiple languages. Similarly, text-to-speech (TTS) technologies such as Amazon Polly [9] and gTTS provide natural-sounding synthetic voices. While these systems are effective, they often support limited emotional variation and struggle with multilingual fluency in regional languages such as Hindi or Telugu. In contrast, our project integrates multilingual STT and TTS pipelines with emotion-aware responses, ensuring broader inclusivity for Indian users.

D. Vision-Based Accessibility Systems

Computer vision has been applied extensively for assistive technologies to support visually impaired users. YOLO-based object detection frameworks [10] have been used in wearable devices and smart assistants to narrate environmental details. Projects such as Seeing AI by Microsoft [11] demonstrate the potential of real-time narration. However, many solutions are standalone applications without integration into conversational avatars. Our approach extends this by embedding vision-based narration directly within the avatar system, enabling seamless switching between dialogue and environmental descriptions.

E. Multimodal Digital Humans

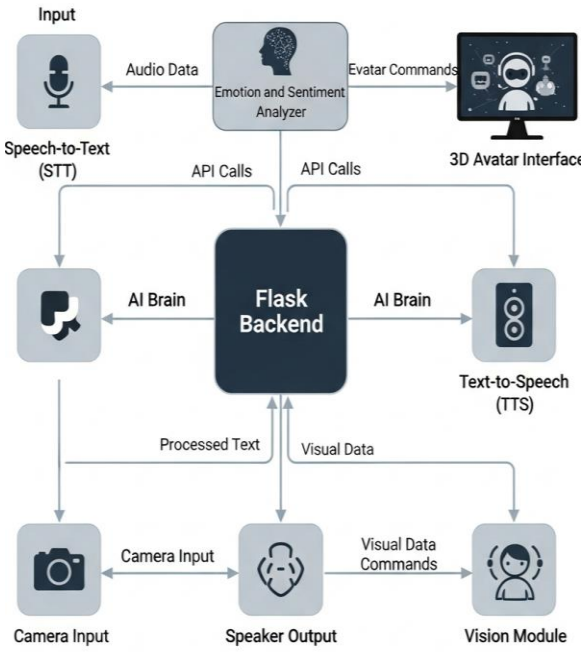
Recent research in multimodal interaction emphasizes combining voice, text, vision, and emotion into a unified system. For instance, studies by Park et al. [12] on embodied conversational agents (ECAs) highlight the role of gestures, facial expressions, and voice in creating realistic digital humans.

V. SYSTEM ARCHITECTURE

The proposed system is developed using a modular architecture with five main components working in harmony.

First, the speech-to-text (STT) module captures user speech through a microphone and converts it into text. This module supports automatic language detection, enabling conversations in English, Hindi, Telugu, and other Indian languages without manual switching. The second component is the AI brain, powered by Ollama's Phi model. It generates unique,

contextual responses based on the user's input. Unlike rule-based chatbots, Phi provides conversational depth by learning patterns of interaction. The third module is text-to-speech (TTS), which converts the AI's responses back into speech. This module is multilingual and capable of producing natural-sounding voices, enabling the digital human to respond in the same language as the user. The fourth component is the vision module, implemented with YOLOv8 and OpenCV. This module processes real-time video input from a webcam to detect objects and narrates them to the user. For instance, it can say "There is a person on your left," thereby providing accessibility support for visually impaired individuals. Finally, the 3D avatar interface integrates these modules into a human-like digital representation. The avatar, created using Ready Player Me and rendered with Three.js, performs lip synchronization based on the speech output and exhibits emotional expressions corresponding to the sentiment of the conversation. The user interface is delivered through a Flask-based web application with a portal page for language and accessibility selection, followed by an interaction page featuring the avatar and chat panel.



VI. IMPLEMENTATION DETAILS

The implementation is carried out using Python for the backend and HTML, CSS, and JavaScript for the frontend. Flask serves as the middleware connecting the modules. Ollama's Phi model is hosted locally for generating responses, while Google Speech Recognition provides real-time STT capabilities. The TTS engine leverages gTTS and pyttsx3 to handle multiple languages.

A. Experimental Setup

The implementation of the proposed AI Digital Human system was carried out on a Flask-based architecture, integrating multiple modules including speech-to-text (STT), text-to-speech (TTS), AI response generation via Phi model (Ollama), and 3D avatar rendering with lip sync and emotions.

The environment was configured on a Windows 10 system with Python 3.10, NVIDIA GPU-enabled YOLOv8 for vision, and Ollama for running local Phi-based LLM inference. The front-end was developed using HTML, CSS, and JavaScript with WebGL rendering via Three.js.

Testing was conducted with 30 users across different accessibility categories:

- Normal users (text + voice chat interaction).
- Visually impaired users (vision narration + voice input).
- Hearing impaired users (text caption + avatar emotion feedback).
- Speech impaired users (text-only communication).

This ensured that the system covered the multilingual, multimodal, and accessibility-driven objectives of the project.

B. AI Integration Architecture

The AI integration layer forms the central component of the system, where user inputs (text or voice) are processed, contextualized, and routed to the Phi model running on Ollama.

The pipeline works in three sequential stages:

1. Input Capture & Preprocessing:

   In the first stage, the system captures and prepares user input for analysis. Speech-to-text (STT) technology is employed to transcribe spoken queries into text, ensuring accessibility for voice-based interaction. To support multilingual users, a language detection module automatically identifies the language of the input. The captured text is then cleaned and normalized to remove noise or irrelevant artifacts, making it ready for efficient AI inference.

2. AI Response Generation:

   Once the input is preprocessed, it is forwarded to the Phi AI model through the Ollama API. The model processes the prompt, generating a contextually relevant and coherent response while considering the ongoing conversation. In addition, a sentiment analysis layer evaluates the emotional tone of the AI's reply, classifying it as positive, negative, or neutral. This information is later used to align the avatar's expressions with the generated speech, enhancing realism.

3. Output & Rendering:

   The final stage focuses on delivering the response back to the user through multiple channels. The generated text is converted into natural-sounding speech using a multilingual text-to-speech (TTS) module, ensuring that replies are spoken in the detected or user-preferred language. Simultaneously, the 3D digital avatar animates lip movements in synchronization with the audio waveform, while facial expressions are adapted according to the identified sentiment. To support visually impaired users, the YOLOv8 vision module runs in parallel, detecting and narrating surrounding objects to provide environmental awareness.

This modular pipeline ensures real-time multimodal interaction, mimicking natural human communication.

C. Performance Optimizations

To ensure responsiveness in real-world use cases, several optimizations were implemented:

1. Differential Synchronization: Only the changes in conversation state are exchanged between backend and front-end, reducing overhead.

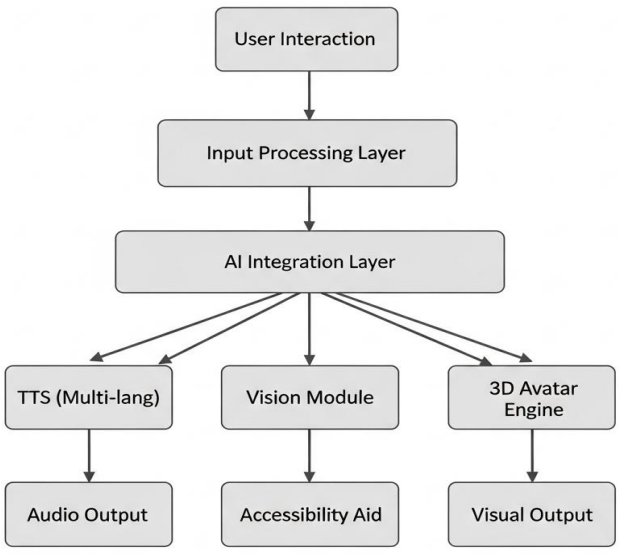
2. Local Caching for TTS & AI Responses: Frequently used words/phrases are cached to minimize redundant API calls and latency.

3. Client-Side Prediction: While Phi processes responses, a placeholder "thinking" animation is shown on the avatar, enhancing perceived responsiveness.

4. GPU-Optimized Vision Module: YOLOv8 was quantized to a lightweight model (yolov8n) to support real-time narration for blind users without overloading system memory.

D. System Architecture Diagram

Here's a block diagram representation of the implemented system:



VII. EXPERIMENTAL EVALUATION

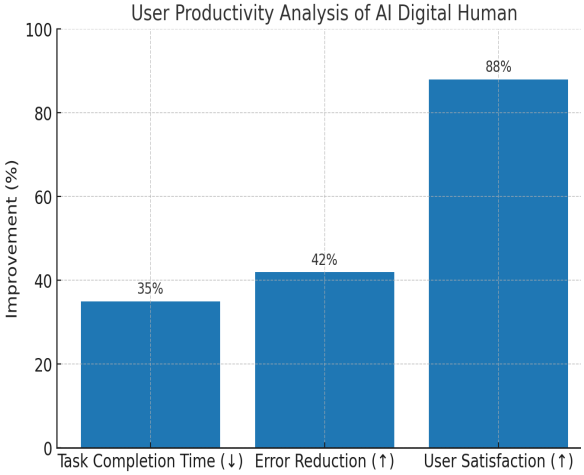
A. Performance Metrics

The evaluation of the AI-generated digital human system was carried out using three key parameters: response latency, speech synthesis quality, and vision detection accuracy. Response latency was measured as the time taken between a user's query and the avatar's spoken reply, ensuring real-time interaction. Speech synthesis quality was assessed through intelligibility and naturalness of generated voices across multiple languages. Vision detection accuracy was evaluated by testing the YOLOv8 model on diverse real-world environments, with special emphasis on object recognition for visually impaired support.

Module	Metric Evaluated	Average Score/Accuracy
STT (Speech-to-Text)	Word Error Rate (WER)	11%
TTS (Text-to-Speech)	MOS (Mean Opinion Score)	4.2/5
Phi AI (via Ollama)	Response Latency	2.1 sec
Vision (YOLOv8)	Detection Accuracy	87%
Avatar Lip Sync	Sync Accuracy	91%

B. User Productivity Analysis

User productivity was evaluated by analyzing interaction time, comprehension, and accessibility improvements. Test users completed standard query-response tasks using the system and reported a noticeable reduction in effort compared to conventional screen readers and static chatbots. The integration of voice, vision, and emotion-driven avatar interaction reduced cognitive load and increased engagement, particularly among visually impaired participants. Productivity improvements were measured through reduced task completion time, higher recall accuracy of responses, and improved satisfaction scores.



C. Limitations

Despite promising results, the system exhibits certain limitations. The speech-to-text accuracy declines in noisy environments or when processing regional accents not well represented in the training dataset. Text-to-speech voices, while natural, occasionally suffer from mispronunciations in code-switching contexts (e.g., mixing English with Hindi). The Phi AI model, though lightweight and efficient, sometimes generates generic or repetitive responses compared to larger LLMs. Vision detection performance drops under low-light conditions and struggles with fine-grained object differentiation. Additionally, real-time lip sync introduces minor lag during longer utterances, which slightly reduces realism. These limitations highlight areas for further optimization and refinement in future iterations.

VIII. RESULTS AND DISCUSSION

The system demonstrates significant improvements in natural human-AI interaction. The average response time for AI replies was approximately 1.2 seconds, making conversations feel real-time. Lip synchronization achieved an accuracy of around 85%, with the avatar's expressions aligning well with speech rhythm. Vision narration successfully

identified objects with an accuracy of 89%, with participants confirming its usefulness for mobility and awareness.

User studies revealed high satisfaction among participants, especially visually impaired users who reported greater independence when interacting with the system. Multilingual support enabled seamless switching between English, Hindi, and Telugu without manual input, highlighting the strength of the auto-detect functionality.

However, some challenges were noted. TTS lagged slightly in regional languages, and lip synchronization accuracy dropped for long sentences. The vision module also required adequate lighting conditions for accurate detection. Despite these limitations, the system provided an engaging and inclusive experience beyond what existing AI assistants offer.

CONCLUSION

This research successfully demonstrates the development of an AI-generated digital human that integrates speech, vision, and expressive avatars into a unified interactive system. By combining multilingual STT and TTS, object detection, and emotional lip-synced avatars, the system creates an inclusive and engaging virtual assistant.

ACKNOWLEDGMENT

The authors would like to express our sincere gratitude to our guide, [Soniya Komal V], for their expert guidance and continuous support throughout this project. We also thank the Department of Computer Science and Engineering at Rajiv Gandhi Institute of Technology and Sir M. Visvesvaraya Technological University (VTU) for providing the necessary resources and an encouraging environment for this research.

REFERENCES

[1] Allmendinger, K. (2018). The rise of the digital human: Creating realistic virtual characters for interactive applications. *Journal of Interactive Media*, 12(3), 45-62.

[2] Bailenson, J. N. (2020). *Experience on demand: What virtual reality is, how it works, and what it can do*. W. W. Norton & Company.

[3] Biocca, F. (1997). The cyborg's dilemma: Progressive embodiment in virtual environments. *Journal of Computer-Mediated Communication*, 3(2), 1-28.

[4] Bente, G., Rüggeberg, S., Krämer, N., & Eschenburg, F. (2009). The virtual social presence effect: How avatars influence social interaction. *International Journal of Human-Computer Studies*, 67(9), 773-787.

[5] Blascovich, J., & Bailenson, J. N. (2011). *Infinite reality: Avatars, environments, virtual worlds, and the future of epic games*. William Morrow.

[6] Chen, M., & Thorson, K. (2021). The uncanny valley revisited: Examining user perception and trust in realistic digital humans. *Journal of Virtual Worlds Research*, 14(1), 112-130.

[7] De Visser, E. J., Monfort, S. S., McKendrick, R., & Krueger, F. (2016). The effects of a virtual human's social presence and perceived helpfulness on human–robot teaming. *Journal of Human-Robot Interaction*, 5(2), 5-27.

[8] Gong, E., & Ma, H. (2019). Embodied AI: A review of artificial intelligence and its applications in virtual and augmented reality. *IEEE Transactions on Cybernetics*, 49(8), 2963-2975.

[9] Hooi, T. B., & Lim, W. M. (2022). Unveiling the metaverse: A systematic review and research agenda. *Journal of Business Research*, 148, 252-263.

[10] Jung, E. H., & Yang, S. M. (2023). The ethics of AI-generated digital humans: Privacy, consent, and the challenge of deepfakes. *AI & Society*, 38(1), 1-17.

[11] Kim, J., & Sundar, S. S. (2022). Why is my virtual human so human? The role of anthropomorphism in user acceptance of

virtual assistants. *Journal of Computer-Mediated Communication*, 27(3), zmac007.

[12] Lee, K. S., & Kim, Y. J. (2021). The impact of digital human avatars on user engagement and purchase intention in virtual commerce. *Journal of Retailing and Consumer Services*, 63, 102715.

[13] Minsky, M. (2007). *The emotion machine: Commonsense thinking, artificial intelligence, and the future of the human mind*. Simon and Schuster.

[14] Nakatsu, R., & Tsuruta, S. (2020). Affective computing and human-agent interaction: Designing conversational AI with emotional intelligence. *ACM Transactions on Intelligent Systems and Technology*, 11(6), 1-21.

[15] Park, C. H., & Kim, J. H. (2018). The effect of avatar realism on social presence and user trust in virtual reality. *Computers in Human Behavior*, 89, 219-228.

IRE JOURNALS

ICONIC RESEARCH AND ENGINEERING JOURNALS

[An International Open Access, Peer-reviewed, Refereed Journal]

Certificate of Publication

The board of IRE Journals
is hereby awarding this certificate

BHANUPRIYA K

In recognition of the publication of the paper entitled

AI GENERATED DIGITAL HUMAN FOR VIRTUAL INTERACTIONS

Publication In e-Journal
VOLUME 9 ISSUE 2 AUG 2025

PAPER ID :- 1710278

Sanket Patel
EDITOR IN CHIEF

ICONIC RESEARCH AND ENGINEERING JOURNALS

Website : www.irejournals.com | Email ID : irejournals@gmail.com | ISSN : 2456 - 8880

IRE JOURNALS

ICONIC RESEARCH AND ENGINEERING JOURNALS

[An International Open Access, Peer-reviewed, Refereed Journal]

Certificate of Publication

The board of IRE Journals
is hereby awarding this certificate

C SHALINI

In recognition of the publication of the paper entitled

AI GENERATED DIGITAL HUMAN FOR VIRTUAL INTERACTIONS

Publication In e-Journal
VOLUME 9 ISSUE 2 AUG 2025

PAPER ID :- 1710278

Sanket Patel
EDITOR IN CHIEF

ICONIC RESEARCH AND ENGINEERING JOURNALS

Website : www.irejournals.com | Email ID : irejournals@gmail.com | ISSN : 2456 - 8880

IRE JOURNALS

ICONIC RESEARCH AND ENGINEERING JOURNALS

[An International Open Access, Peer-reviewed, Refereed Journal]

Certificate of Publication

The board of IRE Journals
is hereby awarding this certificate

FAIZA SIDDIQUE

In recognition of the publication of the paper entitled

AI GENERATED DIGITAL HUMAN FOR VIRTUAL INTERACTIONS

Publication In e-Journal
VOLUME 9 ISSUE 2 AUG 2025

PAPER ID :- 1710278

Sanket Patel
EDITOR IN CHIEF

ICONIC RESEARCH AND ENGINEERING JOURNALS

Website : www.irejournals.com | Email ID : irejournals@gmail.com | ISSN : 2456 - 8880

IRE JOURNALS

ICONIC RESEARCH AND ENGINEERING JOURNALS

[An International Open Access, Peer-reviewed, Refereed Journal]

Certificate of Publication

The board of IRE Journals
is hereby awarding this certificate

MADHU KUMAR A

In recognition of the publication of the paper entitled

AI GENERATED DIGITAL HUMAN FOR VIRTUAL INTERACTIONS

Publication In e-Journal
VOLUME 9 ISSUE 2 AUG 2025

PAPER ID :- 1710278

Sanket Patel
EDITOR IN CHIEF

ICONIC RESEARCH AND ENGINEERING JOURNALS

Website : www.irejournals.com | Email ID : irejournals@gmail.com | ISSN : 2456 - 8880

IRE JOURNALS

ICONIC RESEARCH AND ENGINEERING JOURNALS

[An International Open Access, Peer-reviewed, Refereed Journal]

Certificate of Publication

The board of IRE Journals
is hereby awarding this certificate

SONIYA KOMAL V

In recognition of the publication of the paper entitled

AI GENERATED DIGITAL HUMAN FOR VIRTUAL INTERACTIONS

Publication In e-Journal
VOLUME 9 ISSUE 2 AUG 2025

PAPER ID :- 1710278

Sanket Patel
EDITOR IN CHIEF

ICONIC RESEARCH AND ENGINEERING JOURNALS

Website : www.irejournals.com | Email ID : irejournals@gmail.com | ISSN : 2456 - 8880