SHAHEED ZULFIKAR ALI BHUTTO
INSTITUTE OF SCIENCE & TECHNOLOGY
P.O. Box 345004, Block 10, 6th Floor, Dubai International Academic City, Dubai - U.A.E.
Tel : +971 4 366 4601, Fax : +971 4 366 4607, Email: info@szabist.ac.ae, Web : www.szabist.ac.ae

Design and Development of an AI-Driven Real-Time Sign Language Interpretation System: Enhancing Accessibility for the Deaf and Hard-of-Hearing Community

By

Waleed, Hamza Yaqoob and Syed Muneef Ur Rehman

A Final Year II Project Presented to
The Faculty of Computing and Engineering Sciences

In Partial Fulfillment of the Requirements

for the Degree of

Bachelor of Science (Computer Science)
in the Faculty of Computing and Engineering Sciences

Spring 2025

SHAHEED ZULFIKAR ALI BHUTTO
INSTITUTE OF SCIENCE & TECHNOLOGY
P.O. Box 345004, Block 10, 6th Floor, Dubai International Academic City, Dubai - U.A.E.
Tel : +971 4 366 4601, Fax : +971 4 366 4607, Email: info@szabist.ac.ae, Web : www.szabist.ac.ae

A P P R O V A L S H E E T

This Final Year Project II entitled **"Design and Development of an AI-Driven Real-Time Sign Language Interpretation System: Enhancing Accessibility for the Deaf and Hard-of-Hearing Community)"** presented by **Waleed**, **Hamza Yaqoob**, and **Syed Muneef Ur Rehman** with registration number **211204, 211212** and **211227** in partial fulfillment of the requirements for the degree of **Bachelor of Science (Computer Science) in the Faculty of Computing and Engineering Sciences**, has been evaluated and approved by the Panel of Evaluators.

PANEL OF EVALUATORS

Mr. Syed Adeel Mahmood
Advisor

Dr. Hummayoun Naeem
Juror

Mr. Mohamed Ahamed Amani, MPhil
Program Manager

Dr. Hummayoun Naeem
Research Coordinator

Accepted and approved in partial fulfillment of the requirements for the degree of Bachelor of Science (Information Technology) in the Faculty of Computing.

Dr. Hummayoun Naeem
Head of Campus

Date: _____

SHAHEED ZULFIKAR ALI BHUTTO
INSTITUTE OF SCIENCE & TECHNOLOGY
P.O. Box 345004, Block 10, 6th Floor, Dubai International Academic City, Dubai - U.A.E.
Tel : +971 4 366 4601, Fax : +971 4 366 4607, Email: info@szabist.ac.ae, Web : www.szabist.ac.ae

Dissertation Copyright Form

As a part of the copyright policy of SZABIST and in compliance with the policy, I **Waleed, Hamza Yaqoob** and **Syed Muneef Ur Rehman** delegate my rights of this piece of work in favor of SZABIST-Dubai. I assure that the content of this dissertation is the direct result of my own work.

In addition, I am aware that one copy of this dissertation will be submitted in the Library of SZABIST-Dubai for future use by SZABIST students, faculty and other fellows visiting the campus.

In case of sending part or whole of my present research work on the topic of **"Design and Development of an AI-Driven Real-Time Sign Language Interpretation System: Enhancing Accessibility for the Deaf and Hard-of-Hearing Community)"** at the department of **Computing and Engineering Sciences** in semester **Spring**, year **2025** for (single author) publication by any means, I would require a "No Objection Certificate or NOC" on the prescribed form from the SZABIST Research Office. In case of joint publication with supervisor (only), NOC in written may not be required.In addition I am authorizing SZABIST Dubai for reproducing, distributing and transmitting part or whole of the dissertation by any means including photocopying, recording, or other electronic (online) or mechanical methods except in the case of brief excerpts or quotations embodied in critical reviews and certain other noncommercial uses permitted by the copyright policy.

Signature (Student/Researcher)
Full Name: Hamza Yaqoob
Registration No. 211212
Program: BSCS

Signature (Student/Researcher)
Full Name: Waleed
Registration No: 211204
Program: BSCS

Signature (Student/Researcher)
Full Name: Syed Muneef Ur Rehman
Registration No. 211227
Program: BSCS

Document Information

Category	Information
Customer	Deaf and Hearing-Impaired Community
Project Title	IsharahNow
Document	FYP Report
Document Version	1.0
Status	In Progress
Authors	Muneef, Waleed, Hamza
Approvers	

Definitions of Terms, Acronyms and Abbreviations

AI	Artificial Intelligence
ASL	American Sign Language
CV	Computer Vision
NLP	Natural Language Processing
SDK	Software Development Kit
UAT	User Acceptance Testing
UI	User Interface

Table of Contents

Acknowledgments

We would like to express our sincere and heartfelt gratitude to all the individuals whose support, guidance, and contributions made this project possible. Without their help, the successful completion of ***IsharaNow*** would not have been feasible.

First and foremost, we are immensely grateful to our supervisor, Mr. Syed Adeel Mahmood, for providing us with the opportunity to work on this project. His exceptional guidance, expertise, and constant encouragement were invaluable throughout the development process. Mr. Mahmood was always available to answer our questions, offer insightful suggestions, and steer us in the right direction. His patience and unwavering support ensured that we remained motivated and focused, and it was under his guidance that we were able to achieve the milestones necessary to complete this project successfully.

We would also like to extend our deepest appreciation to Dr. Hummayoun Naeem for evaluating our work. Their constructive feedback, honest evaluations, and thoughtful insights helped us identify areas for improvement, and their suggestions greatly enhanced the quality of our project. Their willingness to dedicate time to review and provide feedback was immensely valuable in the final stages of development.

Additionally, we owe a significant amount of gratitude to the authors of various online publications and academic papers. These resources have been instrumental in expanding our understanding of the technologies used in the ***IsharaNow*** system, particularly in areas related to AI, computer vision, and gesture recognition. The wealth of knowledge shared by these researchers helped us refine our approach and incorporate best practices into our work.

Lastly, we would like to thank our friends, colleagues, and teachers who provided us with continuous encouragement and support throughout the duration of this project. Their willingness to review our work, offer constructive criticism, and provide valuable suggestions was pivotal in improving our project. Their healthy criticism and constructive feedback not only helped us address weaknesses but also motivated us to push ourselves further, ensuring that we continually strived for excellence.

Abstract

As the world becomes more connected, the need for inclusive communication solutions has become increasingly evident. ***IsharaNow*** is a pioneering system designed to address this challenge by converting text and spoken language into real-time sign language through the use of AI, computer vision, and motion recognition technologies. The project aims to bridge the communication gap between individuals with hearing impairments and those who do not use sign language, fostering better understanding and interaction in everyday conversations.

The primary goal of this research is to design and implement a system that translates text or speech input into sign language gestures in real-time. By leveraging advanced machine learning models and computer vision algorithms, ***IsharaNow*** recognizes and converts the input into accurate sign language movements. This system provides an intuitive and accessible way for individuals to communicate using sign language, promoting inclusivity and interaction in diverse environments.

This study delves into the technical aspects of speech-to-sign language translation, focusing on real-time data processing, gesture recognition, and avatar animation. It examines how machine learning models and computer vision can be used to interpret text and speech inputs, and how the system can adapt to different sign language variations. The research also explores the integration of facial expressions and contextual elements to ensure accurate communication.

The findings of this research present ***IsharaNow*** as a significant step forward in enhancing communication for individuals with hearing impairments. By transforming text and speech into sign language, ***IsharaNow*** creates new opportunities for interaction and inclusion, breaking down language barriers and offering a more accessible future for all. The project demonstrates the potential of AI-driven solutions to empower individuals and promote a more inclusive society.

Chapter One: Content and Preliminary Investigation

In recent years, advancements in artificial intelligence and computer vision have paved the way for transformative solutions across diverse fields. As society embraces inclusivity and strives for accessible communication, the need for technology that bridges linguistic gaps has never been more critical. *IsharaNow*, our real-time sign language interpretation system, addresses this need by utilizing state-of-the-art AI, facial recognition, and computer vision techniques. This innovative system translates spoken language into visual sign language instantly, making communication smoother and more accessible for the deaf and hard-of-hearing community, thus promoting inclusivity and enhancing social interaction.

1.1 Project Selection

The selection of the *IsharaNow* project was motivated by the critical need for real-time communication solutions tailored to individuals with hearing impairments. Around the world, accessibility remains a pressing concern, with many individuals facing barriers in accessing essential services and participating in everyday interactions due to communication limitations. This challenge is especially prominent in educational and professional environments, where effective communication is crucial.

To gain further insight, we analyzed data from various online sources, including studies and reports focused on the challenges faced by the deaf community in communication and accessibility [1]. This research provided a comprehensive view of the widespread need for assistive technologies that can facilitate real-time communication. These findings underscored the importance of a system like *IsharaNow*, which utilizes advanced technology to create a seamless bridge between spoken language and sign language, fostering more inclusive interactions across different linguistic and auditory boundaries.

Fig 1.1: Age Demographics of Deaf People Worldwide - WHO

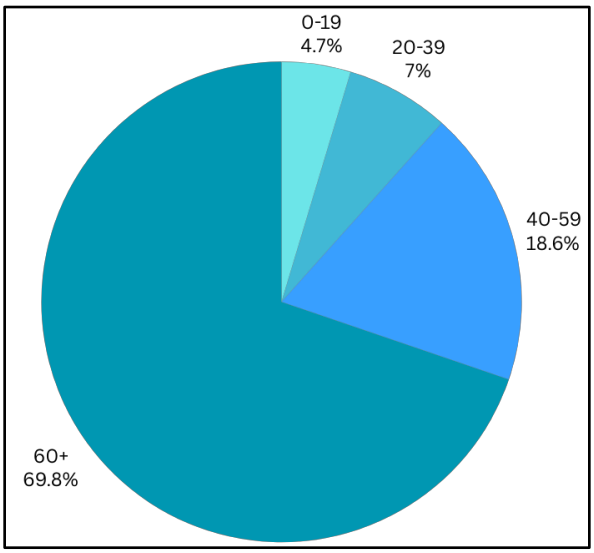
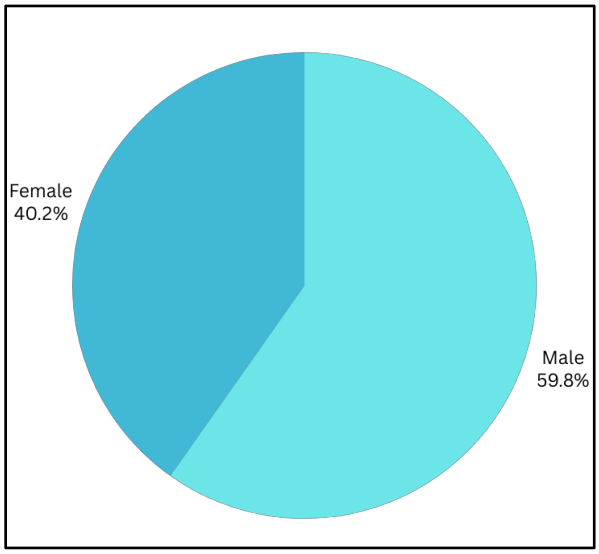


Fig 1.2: Gender Distribution of Deaf People Worldwide - WHO



According to global statistics, hearing loss affects people across a spectrum of severity levels, impacting millions worldwide [1]. The prevalence of hearing impairment highlights the importance of assistive technologies to support effective communication, particularly for those with moderate to complete hearing loss [2]. Understanding the scale and distribution of hearing loss cases reinforces the need for solutions like *IsharaNow*, which seeks to address communication challenges by providing real-time sign language translation. The table below summarizes the global distribution of hearing loss by severity [2]:

Hearing Loss Severity	Millions of People	Percentage Prevalence
Mild	115.3	14.9%
Moderate	266	3.4%
Moderately Severe	103	1.3%
Severe	30.7	0.4%
Profound	17.2	0.2%
Complete	12.6	0.2%

Table 1.1: Hearing Loss Severity, The National Council on Aging

1.2 Project Purpose

The purpose of *IsharaNow* is to bridge the communication gap for individuals with hearing impairments by providing a real-time sign language interpretation system that utilizes advanced AI and computer vision. This project aims to analyze various approaches to real-time translation and address the following key questions:

a. How effectively can AI-based systems recognize and translate spoken language into sign language in real time?
b. Can the accuracy and responsiveness of sign language interpretation be maintained across different dialects and accents, ensuring inclusivity for diverse language users?
c. Is it possible to enhance the accessibility and scalability of the system to serve a larger audience without significantly increasing operational complexity or cost?

Through exploring these questions, *IsharaNow* aims to contribute a meaningful solution to the field of assistive technology by enabling more seamless communication between hearing and non-hearing individuals. By focusing on accuracy, inclusivity, and cost-efficiency, the project aspires to deliver a reliable tool that can improve social interaction and accessibility for the deaf community, promoting a more inclusive society.

1.3 Project Objectives

Our project's objectives are as follows:

1. **Real-Time Translation:**
   Develop an AI-driven system capable of translating spoken language into sign language in real time, enabling seamless communication for individuals with hearing impairments.

2. **Accuracy and Reliability:**
   Enhance the system's accuracy in interpreting various speech patterns and dialects to ensure that translations are reliable and contextually appropriate.

3. **Inclusivity Across Dialects:**
   Implement adaptive algorithms to recognize and translate diverse linguistic variations, making the system accessible to users from different cultural and linguistic backgrounds.

4. **User-Friendly Interface:**
   Design a straightforward and intuitive user interface that enables users of all technical backgrounds to interact easily with the system.

5. **Cost-Effectiveness:**
   Ensure the solution remains affordable by leveraging efficient AI models and accessible hardware, allowing broader adoption across different communities.

6. **Scalability:**
   Structure the system to support potential future upgrades, such as additional languages or specialized sign language dialects, enhancing its reach and versatility.

7. **Educational Impact:**
   Raise awareness and educate the general public about the importance of inclusive communication and the role of technology in bridging accessibility gaps for the deaf community.

1.4 Project Scope

This project aims to enhance accessibility for the deaf and hard-of-hearing community by developing a real-time sign language interpretation system. *IsharaNow* leverages AI, computer vision, and facial recognition to provide accurate and immediate translation of spoken language into sign language, bridging communication gaps in both personal and professional settings. The system is designed to be scalable, adaptable to various sign language dialects, and accessible across different regions. By creating a cost-effective and inclusive solution, *IsharaNow* strives to make communication more seamless, empowering individuals with hearing impairments to engage fully in society.

1.5 Project Overview

Throughout the development of *IsharaNow*, the focus will be on creating a robust, AI-driven platform that translates spoken language into sign language in real time. By utilizing advanced technologies in computer vision and facial recognition, the project aims to ensure high accuracy in capturing and interpreting spoken input for users with hearing impairments. The development process will explore various machine learning models and optimization techniques to improve the system's responsiveness and adaptability across different dialects and cultural contexts. Ultimately, *IsharaNow* will provide an accessible, inclusive tool that enables seamless communication for the deaf community, fostering better integration in social and professional environments.

1.6 Document Conventions

AI	Artificial Intelligence
ASL	American Sign Language
CV	Computer Vision
NLP	Natural Language Processing
SDK	Software Development Kit
UAT	User Acceptance Testing
UI	User Interface
IDE	Integrated Development Environment
OS	Operating System

Table 1.2: Document Conventions

1.7 Brief Literature Review

Sign language interpretation systems have advanced considerably with the integration of AI, computer vision, and natural language processing technologies. This review examines the current state of real-time sign language translation, focusing on key developments in machine learning models, facial recognition algorithms, and motion detection techniques. Studies highlight the importance of accuracy in gesture recognition and the need for systems that can accommodate diverse sign language dialects [1]. These advancements underscore the potential of AI-driven tools like *IsharaNow* to make communication more accessible for the deaf and hard-of-hearing community by delivering reliable, real-time sign language interpretation [2].

[1][2] https://www.who.int/health-topics/hearing-loss
[3] https://www.ncoa.org/adviser/hearing-aids/hearing-loss-statistics/#toc-main-table
[4] https://soeonline.american.edu/blog/qualitative-vs-quantitative/
[5] https://www.virtasant.com/blog/sdlc-methodologies
[6] https://www.rezaid.co.uk/post/sdlc-waterfall-model-the-6-phases-you-need-to-know-about
[7] https://relevant.software/blog/agile-software-development-lifecycle-phases-explained/
[8] https://www.softwaretestingmaterial.com/v-model-in-sdlc/
[9] https://www.plutora.com/blog/incremental-model-what-and-how-to-implement-it
[10] https://www.mbloging.com/post/what-is-react-js

1.8 Project Feasibility Analysis

Project feasibility is an evaluation of the practicality, viability, and potential success of a proposed project. It involves analyzing various factors to determine if the project can be effectively developed and if it will meet its intended goals. Conducting a feasibility study helps identify potential challenges, resource requirements, and the likelihood of achieving the project's objectives [3].

This section examines the feasibility of developing *IsharaNow* in different contexts, considering technical, operational, and economic factors to assess its potential as an accessible, scalable, and reliable real-time sign language interpretation system for diverse user needs.

1.8.1 Economic Feasibility

The economic feasibility of *IsharaNow* assesses the financial considerations, including development costs and projected benefits. The primary expenses for this project include the acquisition of necessary hardware, such as cameras and processing units. Other financial considerations include potential maintenance costs and future upgrades to enhance accuracy and functionality. Overall, the project is deemed economically feasible, as the initial investment is manageable and the long-term benefits—such as improved accessibility and increased social inclusion—justify the costs, particularly given the rising demand for assistive technologies.

For individual users, the system provides a cost-effective alternative to hiring professional sign language interpreters, reducing expenses while ensuring consistent availability. Organizations can integrate *IsharaNow* at a fraction of the cost of employing full-time interpreters or implementing custom-built accessibility solutions. Additionally, the system's compatibility with existing devices such as smartphones or laptops minimizes the need for significant hardware investments. With its affordable pricing model and long-term utility, *IsharaNow* offers users an economically viable solution for bridging communication gaps, making assistive technology more accessible to a broader audience.

1.8.2 Technical Feasibility

Technical feasibility evaluates whether *IsharaNow* can be developed using current technology and available skills. The project requires expertise in machine learning, computer vision, and natural language processing, leveraging frameworks like TensorFlow and OpenCV, which have strong community support. With access to comprehensive documentation, tutorials, and active developer communities, IsharaNow is technically feasible. Challenges, such as optimizing processing speed and accuracy, can be addressed through community resources and best practices in AI development.

1.8.3 Operational Feasibility

Operational feasibility assesses *IsharaNow's* ability to function effectively in real-world settings and gain user support. The project is designed to facilitate seamless communication for individuals with hearing impairments, making it especially beneficial in educational and professional environments. With an intuitive interface and minimal setup, *IsharaNow* is easy to use and integrate into daily interactions. Its accessibility and low-maintenance requirements make it operationally feasible, ensuring it will be well-received and readily adopted by the target audience.

1.8.4 Schedule Feasibility

Schedule feasibility assesses whether *IsharaNow* can be developed within an acceptable time frame. The project is organized into two main phases:

First Phase of the Project

In this phase, the Proposal, SRS, SDS and the initial technical documentation are to be prepared and presented. A short demonstration of the incomplete system will also be presented.

Duration	September 2024 - May 2025
October 13th, 2024	Project proposal presentation, approval, and submission.
October 28th, 2024	Submission of the initial Software Requirements Specification and Software Design Specification documents.
	FYP-1 Final Presentation and the submission of a hard copy of the SRS and SDS documents.

Table 1.3: First Phase Schedule for *IsharahNow*

Second Phase of the Project

In the second phase of the *IsharaNow* project, the team will focus on finalizing a fully functional model of the real-time sign language interpretation system. This phase also includes completing the final report, preparing the technical documentation, and creating a comprehensive user manual to guide end-users in operating the system effectively.

Duration	
May 2nd, 2025	Presentation of the final project.
May 2nd, 2025	Submission of the final project.

Table 1.4: Second Phase Schedule for *IsharahNow*

1.8.5 Conclusion of the Feasibility Analysis

The feasibility analysis demonstrates that IsharaNow is solidly grounded from economic, technical, operational, and scheduling standpoints. The initial costs are justified by the long-term impact and accessibility improvements the system offers, supporting communication for individuals with hearing impairments. Technological requirements, such as AI frameworks and computer vision libraries, are widely accessible, and the development team possesses the necessary skills to complete the project within a reasonable timeframe. Operationally, the system is designed for ease of use and integration into daily interactions, ensuring strong user adoption and support.

In conclusion, IsharaNow is not only feasible but also promises to be a valuable tool in enhancing communication accessibility, making it a project worth pursuing for the positive social impact it stands to deliver.

1.8.6 Existing Solutions

Existing solutions for sign language interpretation, such as video relay services and AI-based sign language recognition systems, provide valuable insights into the technological and design frameworks necessary for *IsharaNow*. These systems highlight the importance of real-time processing, gesture accuracy, and user-friendly interfaces. By analyzing their strengths and limitations, we can incorporate advanced algorithms, optimize gesture recognition, and ensure seamless integration with everyday devices. Leveraging these existing solutions as a foundation enables us to build a more efficient, scalable, and accessible platform tailored to the needs of sign language users.

1.9 Deliverables

1.9.1 Hardware

- **Camera**: High-resolution camera for capturing gestures and facial expressions.
- **Microcontroller/Processing Unit**: For real-time data processing and running AI models.
- **Display Unit**: Screen to display sign language translations, if required for demonstration or feedback.
- **Audio Input Device**: High-quality microphone to capture spoken language input accurately.

1.9.2 Software

- **Machine Learning Models**: Trained models for sign language gesture recognition and translation.
- **Real-Time Video Processing**: Software for processing video input and converting gestures into sign language in real-time.
- **User Interface**: An intuitive UI for system interaction, including settings adjustments and feedback options.

1.10 Report Orientation

This final report for the *IsharaNow* project is organized into the following chapters and appendices:

- **Chapter 1:** Context and Preliminary Introduction
  This chapter introduces the *IsharaNow* system, outlining its goals, significance, and the motivation behind developing a real-time sign language translation system using AI and computer vision. The problem statement and the benefits of such a system are also discussed.

- **Chapter 2:** Research
  This chapter covers the background research and literature review relevant to *IsharaNow*. It examines existing technologies, similar projects, and methodologies used in the field of sign language recognition and AI-powered translation.

- **Chapter 3:** Analysis
  This chapter analyzes the feasibility of *IsharaNow* from various perspectives, including technical, economic, and operational. It explores potential challenges and risks while justifying the project's approach and design choices.

- **Chapter 4:** Design
  The design chapter details the architecture of the *IsharaNow* system, including the AI models, computer vision integration, and user interface. It also covers key components such as video processing, gesture recognition, and real-time translation.

- **Chapter 5:** Test Plan
  This chapter outlines the testing strategies, methodologies, and success criteria used to ensure the accuracy and reliability of IsharaNow. It includes both functional and usability tests.

- **Chapter 6:** Critical Evaluation
  This chapter critically assesses the performance of *IsharaNow*, evaluating whether the system met its objectives and identifying areas for improvement. It also discusses challenges faced during development and future enhancements.

- **Appendix A:** Glossary
  A glossary of terms and abbreviations used throughout the report.

- **Appendix B:** Code
  Complete source code for the *IsharaNow* system.

- **Appendix C:** User Manual
  Instructions for end-users on how to set up, operate, and interact with *IsharaNow*.

- **Appendix D:** Developer's Manual
  A guide for developers on how to understand, modify, and extend the *IsharaNow* system.

Chapter Two: Research

2.1 Academic Research

In *IsharaNow*, academic research played a pivotal role in shaping the development of the sign language translation system. The research methods employed were designed to ensure systematic collection and analysis of data, which directly contributed to the accuracy and efficiency of the system.

Two key decisions guided the research process:

- **Data Collection Techniques**: Various data collection methods were explored to gather sign language gestures, including the use of video datasets and real-time data captured from sensors and cameras. These datasets served as the foundation for training the AI and computer vision models that power the gesture recognition system.
- **Data Analysis Approaches**: Advanced techniques in machine learning and computer vision were employed to analyze the collected data. The use of supervised learning, neural networks, and image-processing algorithms allowed for the effective recognition of hand gestures and facial expressions in real-time.

These research methodologies were integral to the success of *IsharaNow*, ensuring that the data was accurately interpreted to deliver precise sign language translations, aligned with the project's objectives.

2.1.1 Research Methodologies:

Qualitative

Qualitative research explores phenomena using non-numerical data like text or images. It answers **"why"** or **"how"** questions, using methods like interviews or focus groups to gain deep insights into individual or cultural experiences. Results are descriptive and subjective, providing context-rich understanding without aiming for generalization.
This project follows a qualitative research method. For more details regarding this part, please refer to chapter one of this document.

Quantitative

Quantitative research focuses on numerical data and statistics to measure variables and test hypotheses. It answers **"how much"** or **"how many"** through structured methods like surveys and experiments. The results are objective and generalizable, useful for validating theories or identifying patterns. [4] Read more

2.1.2 Software Development Life Cycle Methodology

The development team must choose the system development methodology best suited to the project's specific needs to ensure effective management. Each approach offers distinct benefits, limitations, and objectives. [5] Read more

Waterfall Development Methodology

The **Waterfall Model** is a linear software development approach with six sequential phases: requirements, design, implementation, testing, deployment, and maintenance. Each phase must be completed fully before the next begins, making it ideal for projects with stable, well-defined requirements but less suited for those needing flexibility. [5] Read more

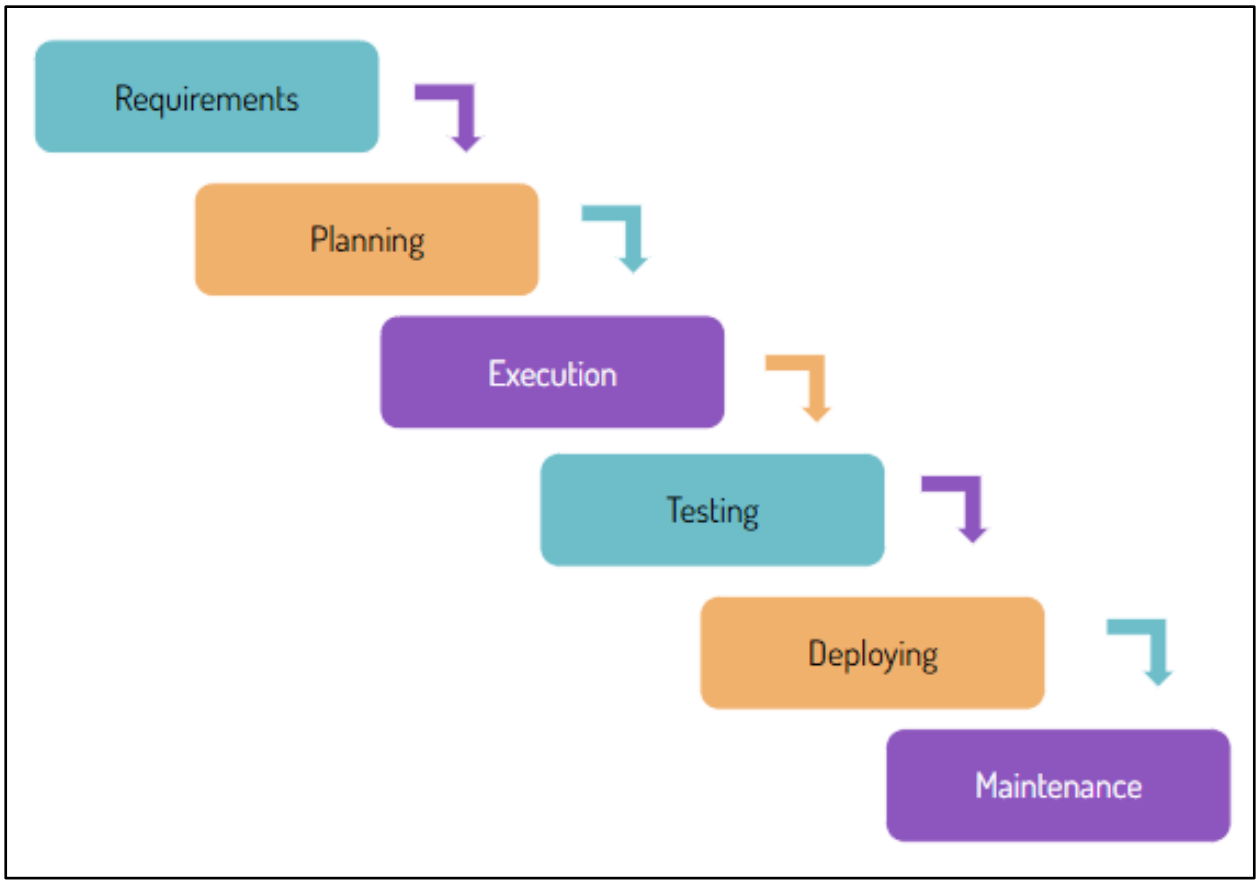


Fig 2.1: Waterfall Development Methodology

Agile Development Methodology

The **Agile Model** is a flexible, iterative software development methodology designed to accommodate changing requirements throughout a project. Work is divided into small, incremental cycles called sprints, each focusing on delivering a functional product piece. Agile emphasizes collaboration, customer feedback, and adaptability, ensuring continuous improvement and quick delivery of value. It is well-suited for dynamic projects where requirements evolve over time. [6] Read more

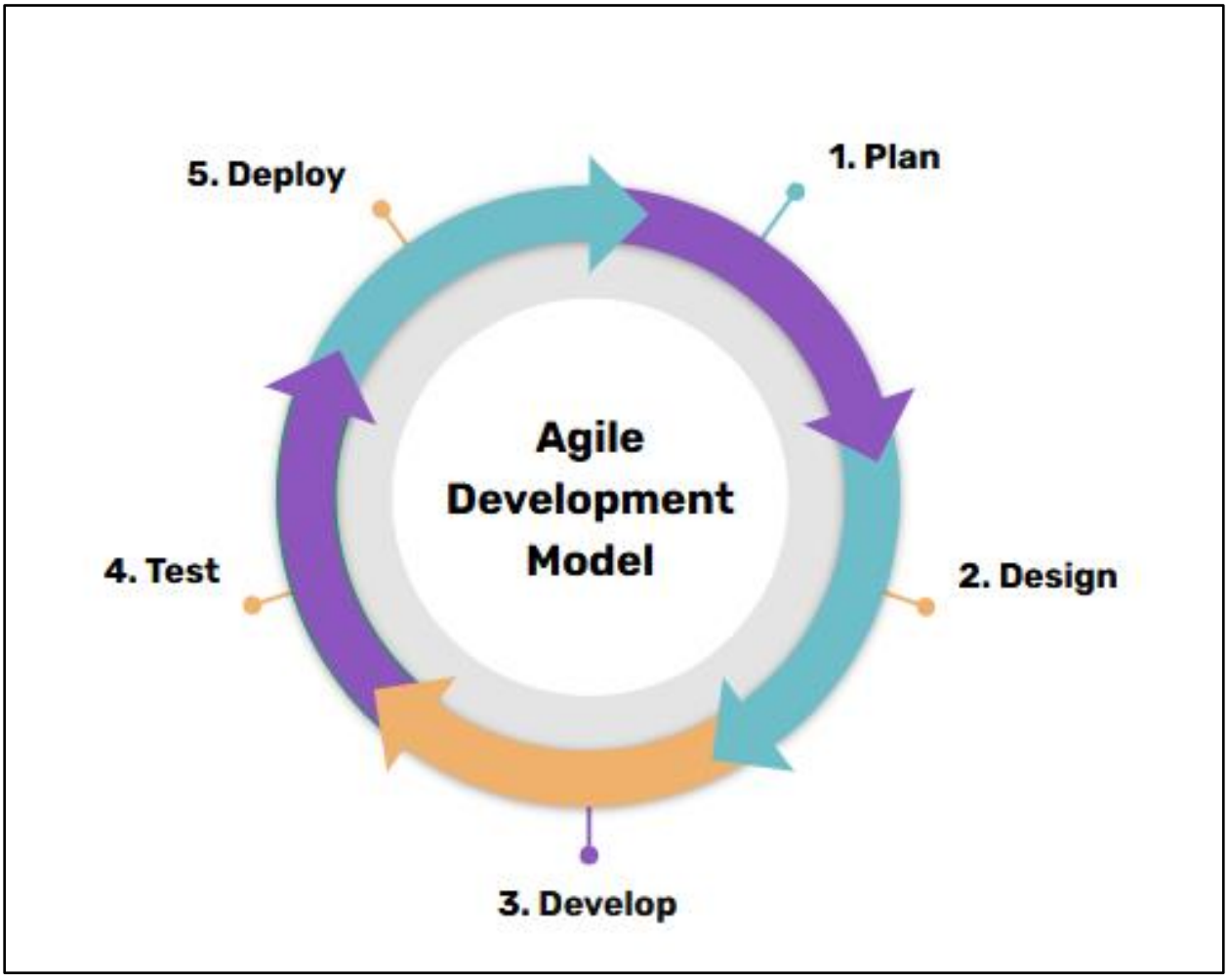


Fig 2.2: Agile Development Methodology

V-Model Development Methodology

The **V-Model** (Validation and Verification Model) is an SDLC approach where development and testing processes run in parallel, forming a "V" shape. Each development phase, such as requirements, design, or coding, has a corresponding testing phase to validate it. This model emphasizes early detection of defects by integrating testing throughout the lifecycle. It's ideal for projects with clear, stable requirements but less effective for dynamic or evolving needs. [7] Read more

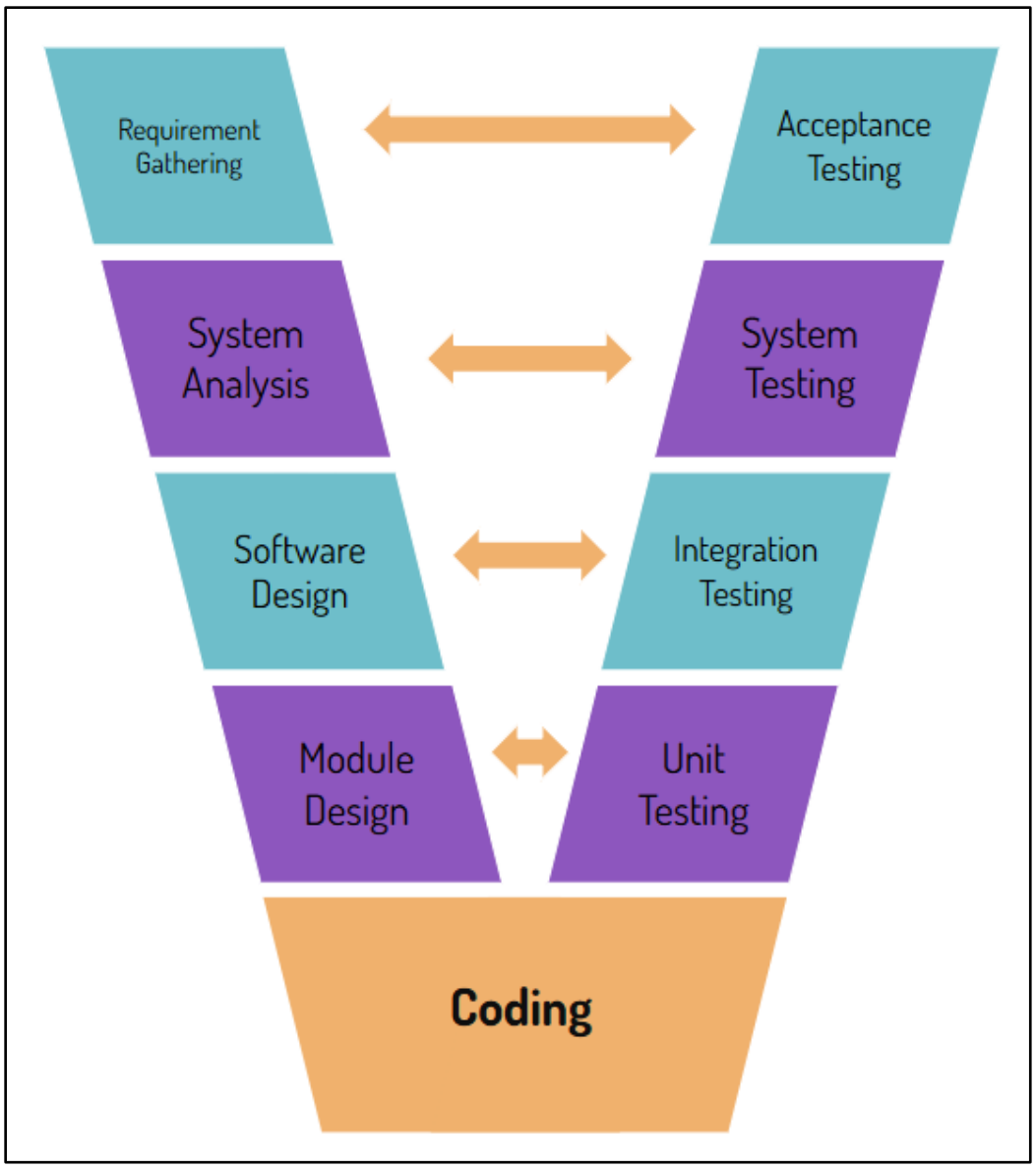


Fig 2.3: V-Model Development Methodology

Incremental Development Methodology

The incremental model is a method of software development that breaks the process into smaller, manageable segments or increments. Each increment builds upon the previous one, gradually adding functionality until the complete system is developed. This model allows for testing and delivery at each stage, enabling teams to adapt to changes and incorporate feedback early in the development process. [8] Read more

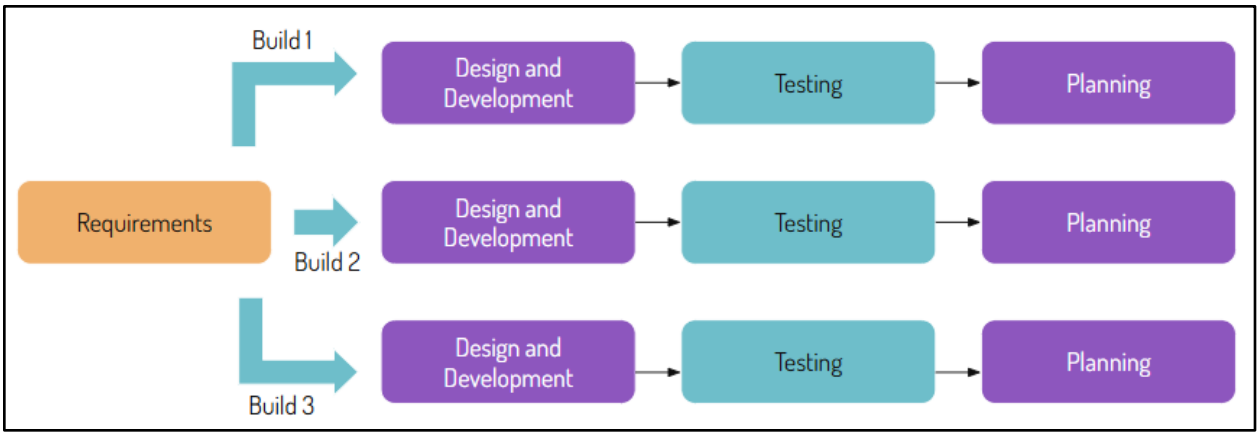


Fig 2.4: Incremental Development Methodology

Selection of System Development Methodology

After careful evaluation of different development methodologies, the team opted for the **Agile Model** for this project. Several key reasons influenced this decision [4]:

- The Agile methodology supports continuous testing, ensuring that issues are identified and resolved early in the development cycle.
- Agile allows for faster development, enabling the team to deliver functional components at a quicker pace compared to other methods.
- The iterative nature of Agile supports the continuous inclusion and testing of new features, allowing for greater flexibility and adaptability throughout the development process.

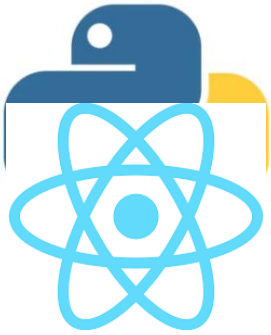
This methodology was chosen to ensure timely delivery and accommodate any evolving requirements throughout the project's lifecycle.

2.1.3 Development Tools

Several development tools were required to create the IsharaNow and make it functional.

Python

Python is a high-level, interpreted programming language known for its clear syntax and readability. It supports multiple paradigms, including procedural, object-oriented, and functional programming, making it versatile for various applications. With a vast standard library and active community, Python excels in web development, data science, automation, and artificial intelligence. [10] Read more



React JS
React JS is a JavaScript library primarily used for building user interfaces, especially single-page applications where dynamic updates to the UI are required. It is maintained by Meta (formerly Facebook) and a community of developers, offering tools for creating reusable UI components. React promotes a declarative programming model, allowing developers to design interfaces with an efficient, predictable, and flexible architecture by breaking them into components. [10] Read more

AWS
AWS (Amazon Web Services) is a comprehensive cloud computing platform that provides on-demand services like compute power, storage, databases, and networking. It enables scalable, flexible, and cost-effective solutions for startups to enterprises. With tools for security, machine learning, DevOps, and serverless computing, AWS supports rapid innovation and global deployment. Its pay-as-you-go model and robust infrastructure make it a leader in cloud technology. [12] Read more

Programming Tools

Visual Studio Code
Visual Studio Code is a lightweight source code editor developed by Microsoft. It is highly customizable and widely used for developing applications across various programming languages. Developers work with Visual Studio Code by installing extensions and configuring settings to tailor the environment to their needs. Built with Electron, it offers features like IntelliSense, debugging, and Git integration, making it a versatile tool. It is cross-platform and has become one of the most popular editors due to its speed, flexibility, and active community support.

Chapter Three: Analysis

3.1 User Requirements/Use Cases

3.1.1 Use Cases

The *IsharaNow* system offers several key functionalities aimed at enhancing communication accessibility for users:

- **Input Text or Speech**: Users can input text or speech into the system, which is then processed for real-time sign language translation.
- **Real-Time Sign Language Output**: The system translates the input into sign language, ensuring that communication is visually represented accurately.
- **Educational Assistance**: The system can also serve as a learning tool for those interested in understanding or learning sign language, providing translations alongside text or speech inputs.

These functionalities empower users to communicate seamlessly, breaking down language barriers and creating an inclusive environment for individuals with hearing impairments.

3.1.2 Analyst Requirements

Data Analytics

- **User Interaction Analysis**: Analysts require tools to evaluate how users interact with the *IsharaNow* system, including trends in usage frequency and common inputs (text or speech).
- **Translation Accuracy Evaluation**: Analysts need mechanisms to assess the accuracy of the system's translations by comparing outputs against established sign language standards, ensuring continuous improvement.
- **Performance Metrics**: Tools are needed to monitor system performance, such as processing speeds, error rates in gesture outputs, and user feedback ratings, to optimize the user experience.
- **User Demographics**: Analysts require insights into user demographics to understand adoption patterns across different regions, age groups, and industries.
- **Adaptability to Sign Language Variants**: Data on the system's ability to adapt to regional and cultural differences in sign language helps ensure broader applicability and inclusivity.

These analytical requirements enable continuous improvement of *IsharaNow*, ensuring that it remains accurate, user-friendly, and adaptable to changing user needs and expectations.

Reporting

- **System Performance Reports**: Analysts require tools to generate detailed reports on the system's performance, including metrics such as translation accuracy, processing speed, and system uptime. These reports help in identifying areas for optimization.
- **User Feedback and Satisfaction Reports**: Tools must be available for analysts to collect and analyze user feedback, generating reports on satisfaction levels, feature requests, and common challenges faced by users.

- **Usage Statistics**: Reports should include usage trends, such as the number of translations performed and most common inputs (text or speech)
- **Demographic Insights**: Analysts need the ability to generate reports on user demographics to understand who is using *IsharaNow* and how it can better serve diverse populations.
- **Error Tracking**: Reports on system errors, including failed translations or inaccurate gesture outputs, help analysts prioritize improvements and address recurring issues effectively.

These reporting capabilities provide a comprehensive understanding of *IsharaNow's* functionality, user experience, and areas for enhancement, driving continuous improvement and adaptability.

System Monitoring

- **Access to Logs**: Analysts require detailed system logs that record all operations, including translation processes, input processing, and output generation. These logs help trace issues and identify patterns leading to potential errors.
- **Diagnostic Tools**: Tools must be available to monitor the health of the system components, including the AI models, server performance, and data processing units, ensuring smooth operations.
- **Error Monitoring**: Analysts need real-time error tracking that highlights failed translations, processing delays, or connectivity issues, enabling quick resolution.

These monitoring tools ensure the reliable operation of *IsharaNow*, enabling analysts to proactively address issues and maintain system efficiency.

3.1.3 Developer Requirements

Modular Codebase
Developers require a modular and well-documented codebase to ensure that components like the sign language animation engine, text-to-sign conversion module, and user interface can be updated or replaced independently.

Integration Framework
Support for integrating additional sign language variants, gestures, or animation styles without requiring significant changes to the core system is essential.

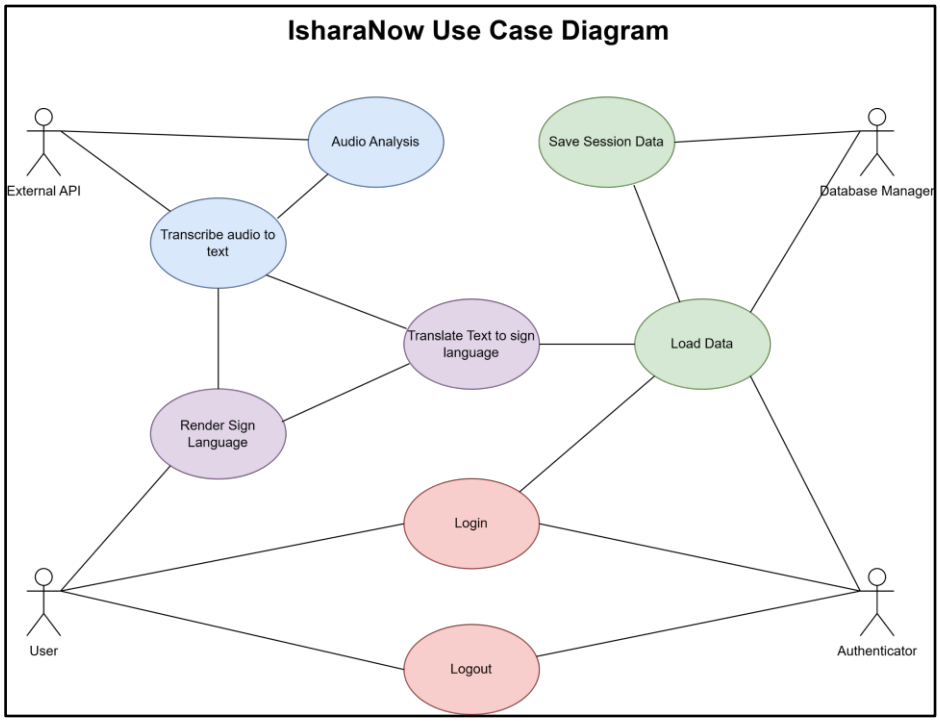
Testing Tools
Developers need simulation and testing tools to validate real-time gesture generation and translation accuracy, ensuring the system performs effectively under various scenarios.

Documentation
Comprehensive technical documentation is required, including system architecture, API usage, gesture model training procedures, and deployment guidelines, to assist in system maintenance and scaling.

Debugging Support
Developers need robust debugging tools, such as real-time logging, error tracking, and performance monitoring, to identify and resolve issues during the translation and animation processes.

These requirements ensure that *IsharaNow* is scalable, maintainable, and reliable, providing developers with the tools and resources needed for efficient system updates and long-term sustainability.

3.2 Use Case Diagram



The use case diagram highlights the primary interactions between the user and *IsharaNow*. Users can input text or speech, select the desired sign language variant (e.g., ASL, BSL), and view the real-time sign language translation via an animated avatar. The system also incorporates options for user feedback and configuration of settings, ensuring a seamless and user-friendly experience.

3.3 System Specifications

3.3.1 Functional Requirements

Input Processing

- The system must accept input in the form of text or spoken language from the user.
- Speech input must be processed using a speech-to-text engine to convert it into a textual format.

Sign Language Translation

- The system must translate the input text into sign language gestures based on the selected sign language variant.

Real-Time Animation

- The system must render sign language gestures in real time through an animated avatar.
- Animations must be smooth and accurate, matching the input text or speech.

User Interface

- The system must provide an intuitive user interface where users can input text or speech, select sign language variants, and view translations.
- The UI must allow users to adjust animation settings, such as speed or clarity.

Data Logging

- The system must log user inputs, translation outputs, and system performance metrics for analysis and improvement.

Connectivity

- The system must support internet connectivity for updates and access to cloud-based translation models.

3.3.2 Quality Attributes

Reliability

- The system must operate continuously without crashes or interruptions.
- It must handle input errors (e.g., unsupported text or speech) gracefully, providing meaningful feedback to the user.

Scalability

- The system must support additional sign language variants or gesture libraries without significant rework.
- It must be capable of accommodating an increasing number of users without degradation in performance.

Usability

- The user interface must be intuitive, allowing users to input text or speech, select sign language options, and view translations easily.
- The system must provide clear and accurate translations that are easy to understand for both hearing and non-hearing users.

Performance

- The system must process user input and generate animated sign language translations within 1-2 seconds.
- It must maintain consistent performance even during high usage periods.

Maintainability

- The system must have a modular design, enabling developers to update or replace components (e.g., animation engine, language libraries) without affecting the entire system.
- Comprehensive documentation must be available to facilitate troubleshooting and future enhancements.

Security

- Only authorized users should have access to sensitive system configurations or customizations.
- Data related to user inputs and system logs must be securely stored and transmitted to prevent unauthorized access.

3.3.3 Domain Requirements

Domain requirements are tailored to the specific operational environment of *IsharaNow*, ensuring that the system functions optimally under various real-world conditions.

Environmental Conditions

- The system must be able to function in diverse indoor and outdoor environments where lighting, temperature, and ambient noise levels may vary.
- It should accommodate different lighting conditions to ensure that users can interact with the system even in low-light or bright environments.

Power Supply

- The system should be designed to run continuously with minimal power consumption.
- Backup power solutions should be available in case of power failure, especially in critical user-facing operations like real-time translation.

Integration

- *IsharaNow* must integrate seamlessly with existing communication devices, platforms, and applications, such as virtual assistants or mobile devices.
- It should be adaptable to various operating systems, including desktop and mobile versions, ensuring compatibility across different user environments.

Compliance

- The system must adhere to accessibility guidelines and best practices for sign language translation technologies, ensuring inclusivity and legal compliance.
- It must comply with data protection regulations and standards, particularly in storing and handling user information or translations.

3.3.4 Interface Requirements

User Interface

- The system must provide an intuitive graphical user interface (GUI) accessible through a dedicated web application.
- The UI must display real-time sign language translation, system status, and logs of previous translations for review.
- Users must be able to configure system settings, such as language preferences, translation speed, and gesture clarity.

Hardware Interfaces

- *IsharaNow* must interface with cameras or other input devices that capture live user gestures or facial expressions in real time.

Chapter Four: Design

4.1 Architectural Strategies

4.1.1 Data Preprocessing

The preprocessing pipeline begins with converting spoken language input into text. This text is then normalized, segmented into sentences, and prepared for translation using either a gloss-based or SignWriting-based system. Techniques such as lemmatization and syntactic simplification are employed to ensure compatibility with the translation pipeline.

4.1.2 Automated Control System

The architecture follows a modular control system that routes input through distinct processing modules—language identification, sentence splitting, gloss generation or SignWriting translation, pose sequence generation, and video synthesis. Automation ensures the dynamic selection of translation approaches based on the input and quality requirements.

4.1.3 System Calibration and Optimization

The system is continuously optimized by:

- Annotating low-quality modules (orange) for future improvement.
- Replacing unimplemented components (red) with new implementations.
- Enhancing the accuracy of modules via training on diverse datasets. The SignWriting pipeline primarily benefits from large-scale datasets, enabling robust and expressive translations.

4.1.4 Scalability and Modularity

The entire architecture supports scalability by separating computer vision (e.g., pose estimation, video rendering) from linguistic translation (e.g., glossing, SignWriting generation). This separation allows for component reuse, independent upgrades, and deployment in various environments, including mobile and web platforms.

4.2 Deliverables of Process Modelling

4.2.1 Context Diagram

The context diagram illustrates how input flows from the user to the final translated output. It includes:

- Spoken language input (audio/text).
- Language processing modules.
- Translation subsystems (Gloss-based / SignWriting-based).
- Pose generation and rendering components.
- Output video or shared translation.

4.2.2 System Architecture

The system architecture comprises two main translation flows:

1. **Gloss-Based Pipeline:**
    - Spoken text → Glosses → Pose Sequence → Video Rendering.
    - Uses a sign dictionary for pose generation.
    - Faster but less expressive and lower in quality.
2. **SignWriting-Based Pipeline:**
    - Spoken text → SignWriting → Pose Sequence → Video Rendering.
    - Allows for nuanced grammar and emotion representation.
    - Offers quality close to human interpretation.

4.2.3 UML Diagram

The UML diagrams encapsulate:

- **Use Case Diagram**: Outlining user interactions (upload, translate, share).
- **Sequence Diagram**: Illustrating process flow from input to output.
- **Class Diagram**: Highlighting core components like LanguageProcessor, PoseGenerator, VideoRenderer, and their interrelations.

4.3 Policies and Tactics

4.3.1 Application Versions

The application is developed with a modular design that supports incremental improvements through version control. Each major version reflects enhancements in translation quality, user interface, and processing efficiency. Versioning is handled using Git, with milestones aligned to core improvements such as:

- Integration of SignWriting translation modules.
- Improved gloss-to-pose mapping accuracy.
- Real-time video rendering capabilities. Each version is tested for performance benchmarks, including accuracy of sign interpretation and latency in translation.

4.3.2 Functional and Object-Oriented Programming

The software architecture combines both **functional** and **object-oriented** programming paradigms:

- **Functional Programming** is applied in preprocessing steps such as text normalization and sentence segmentation, enhancing readability and reducing side effects.
- **Object-Oriented Programming** is used in modeling key components such as LanguageProcessor, PoseGenerator, and VideoRenderer, ensuring modularity and scalability. This dual-paradigm approach allows efficient data transformation while keeping the system flexible and maintainable.

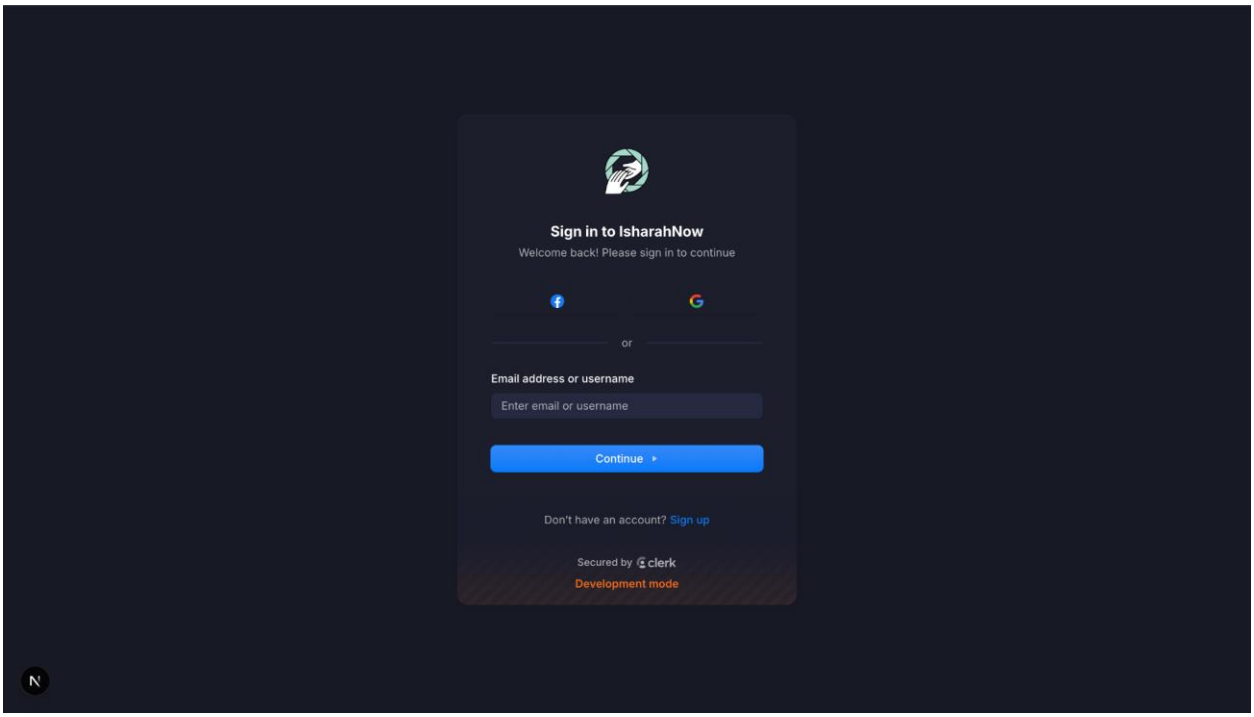
4.3.3 Software Development Model

The project follows an **Agile development model** with a focus on iterative refinement. Features are developed in sprints with continuous feedback from test users and academic advisors. Key stages of the model include:

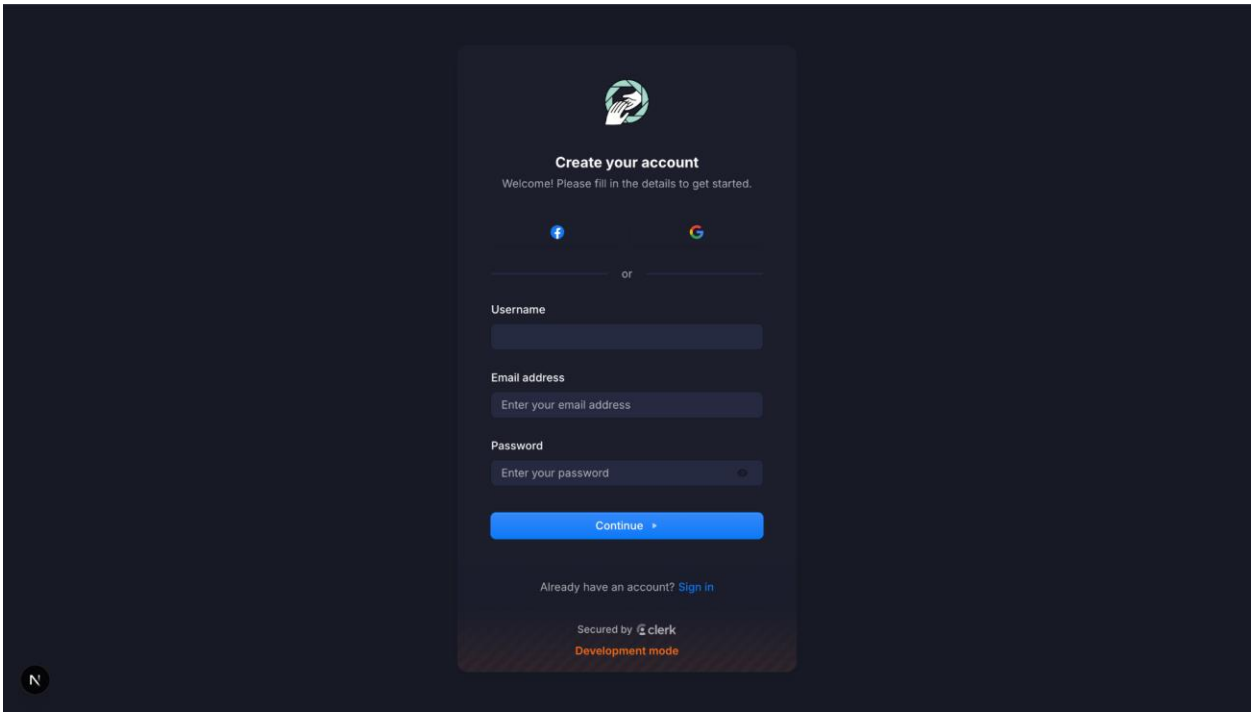
- **Design and Prototyping** – Development of modular components and translation pipelines.
- **Implementation and Testing** – Continuous integration of updated models and module evaluations.
- **Deployment** – Hosted on a web interface for demonstration, with future versions planned for desktop and mobile platforms.

4.4 GUI

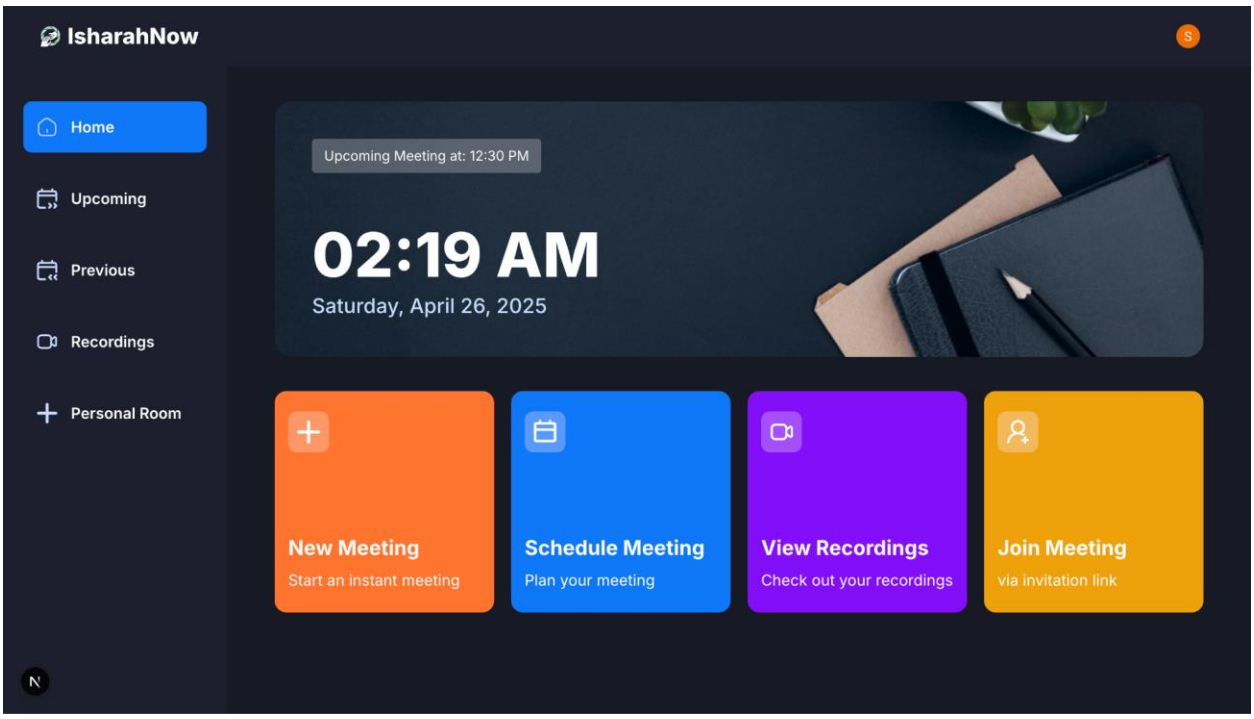
4.4.1 Sign In Page



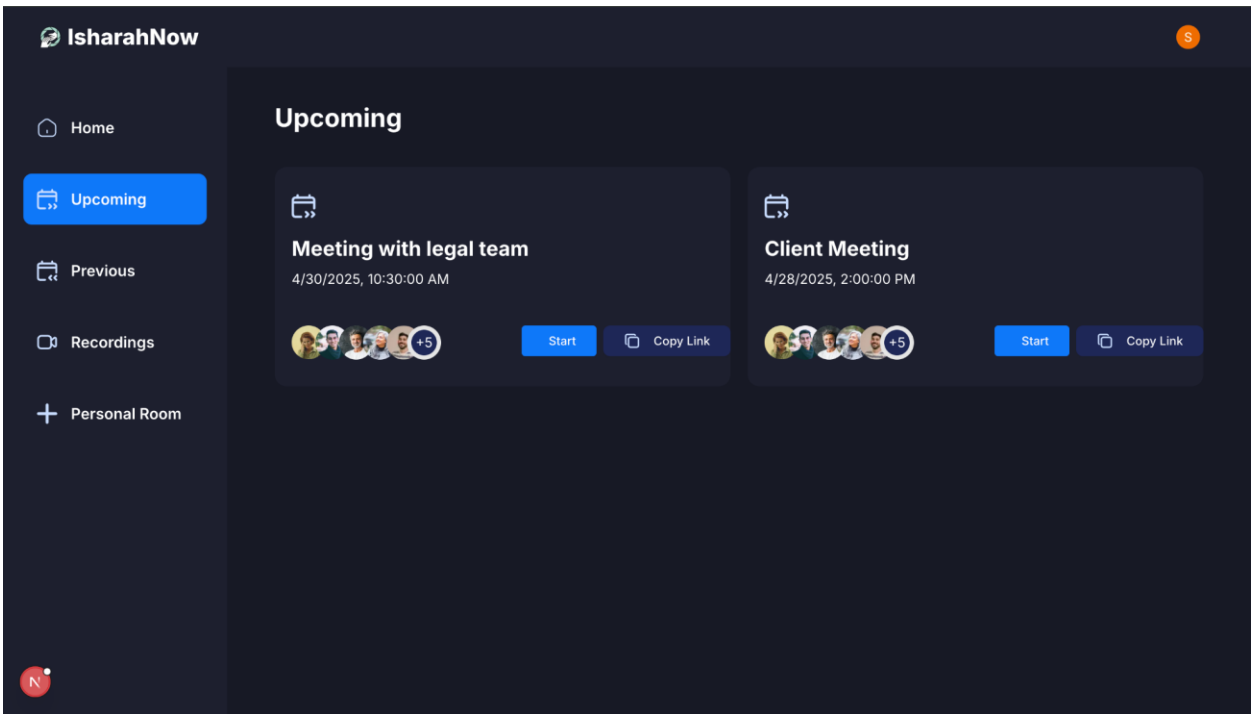
4.4.2 Sign Up Page



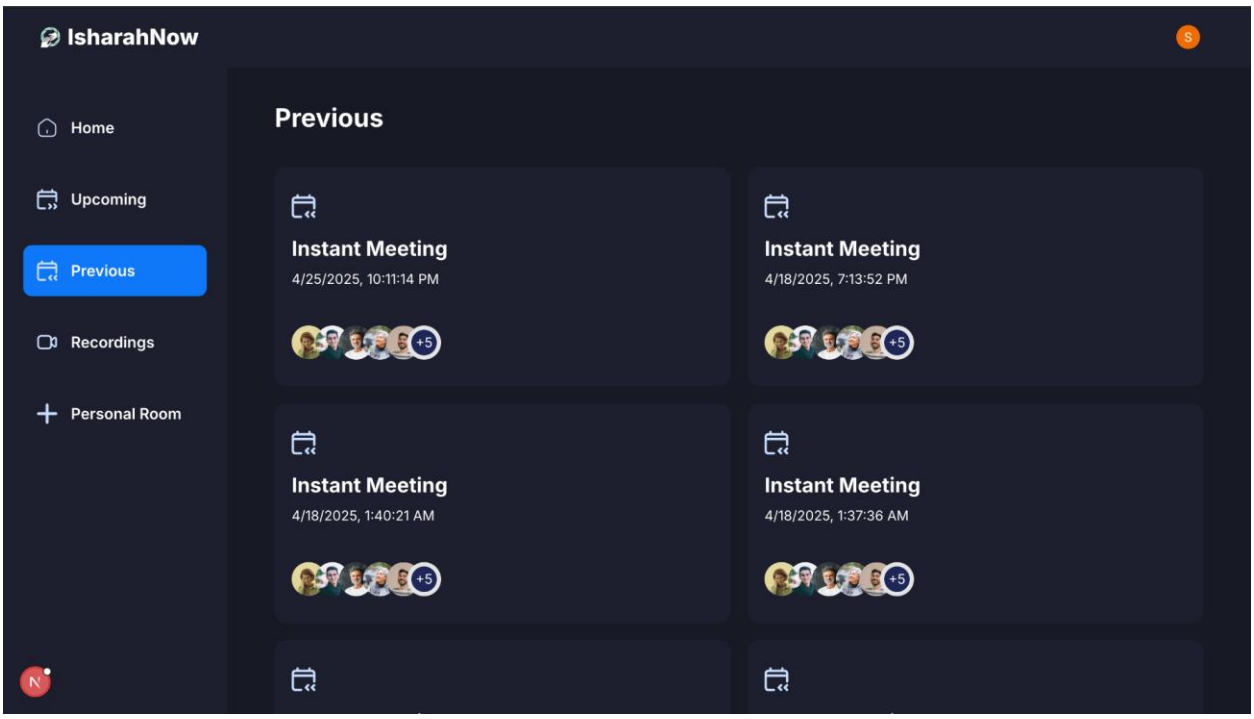
4.4.3 Home Page



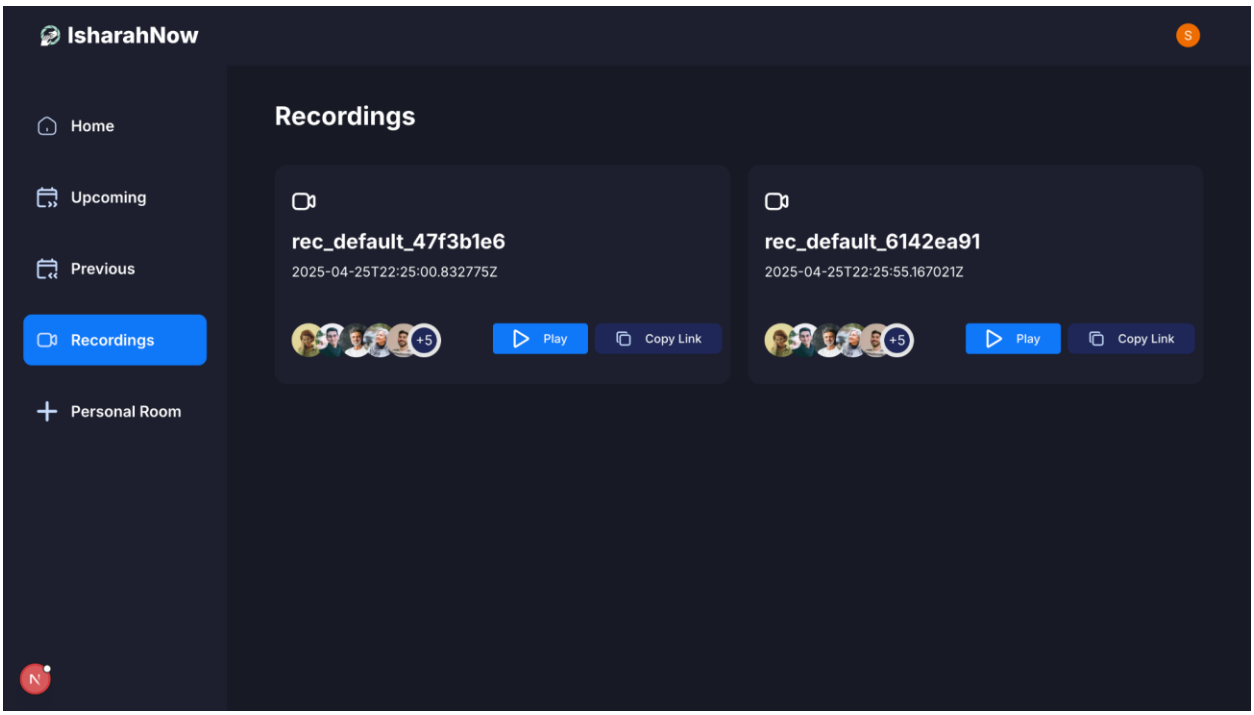
4.4.4 Upcoming Meetings Page



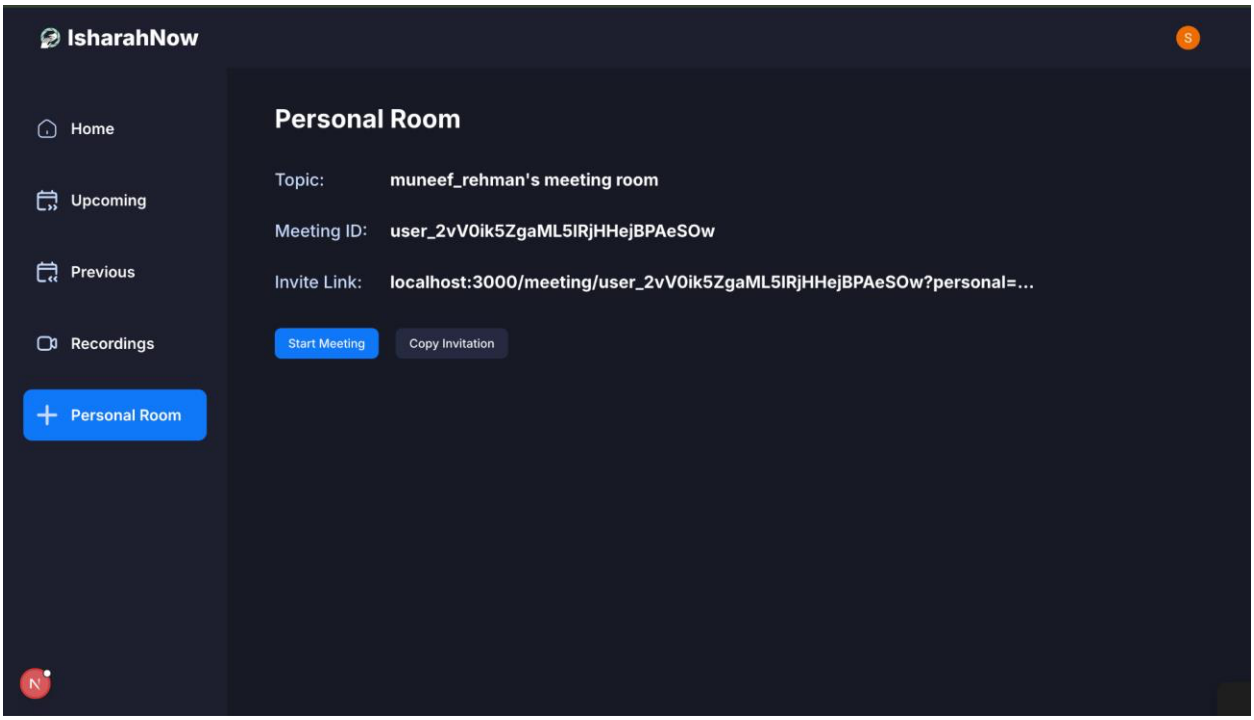
4.4.4 Previous Meetings Page



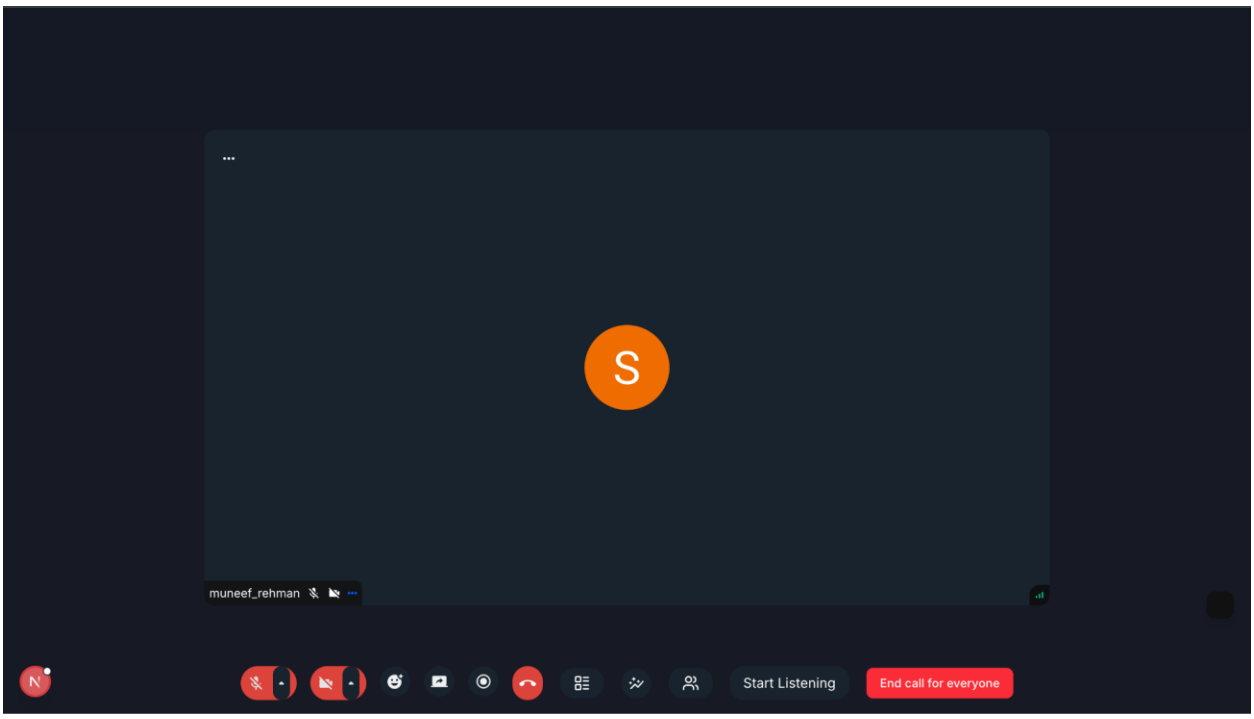
4.4.5 Recordings Page



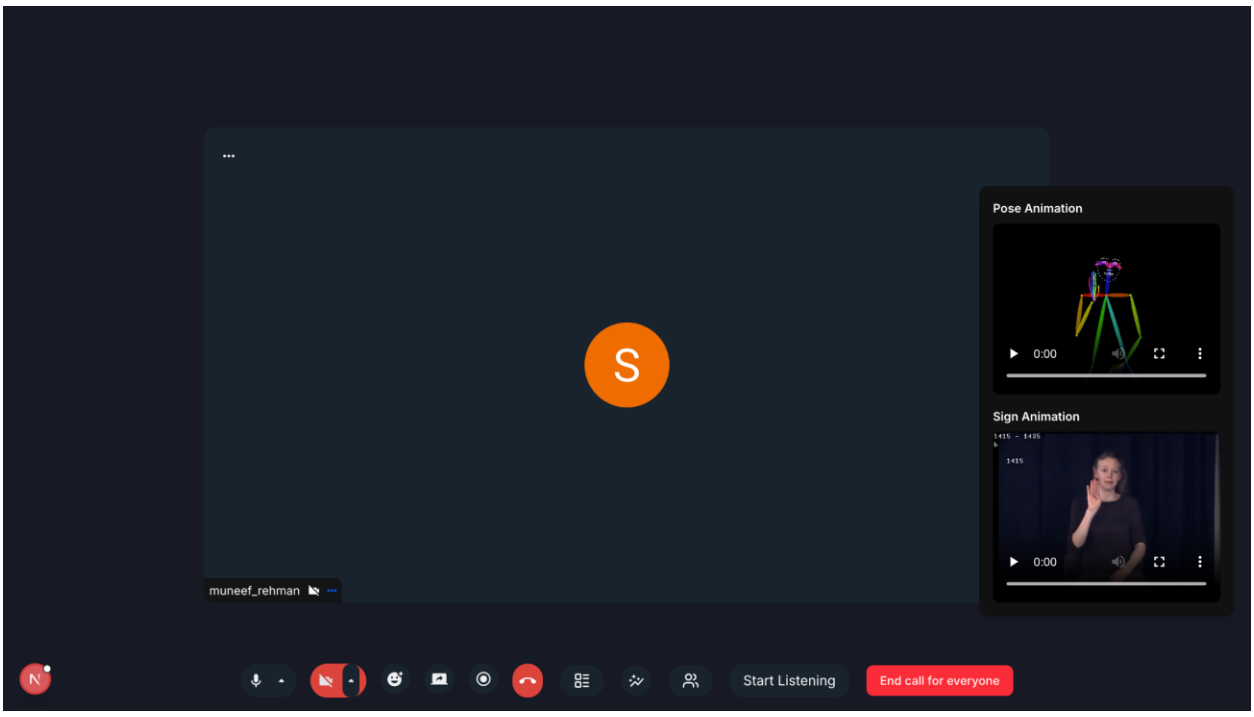
4.4.6 Personal Room



4.4.7 Meeting Room



4.4.8 Meeting Room with Sign Language Translation



Chapter Five: Test Plan

5.1 Scope of Testing

The testing scope for IsharaNow encompasses all critical components of the system, including text and speech input processing, real-time sign language translation, avatar rendering, and user interface interactions. This scope ensures that both functional and non-functional requirements are validated. The testing process is designed to uncover any defects or inconsistencies across modules and confirm that the final product performs reliably under expected usage conditions.

5.2 Test Plan Strategy

The testing strategy adopts a layered approach, beginning with individual module validation and culminating in system-wide evaluation. This ensures the integrity of each component and the seamless integration of the entire system. The following testing levels will be used:

5.2.1 Unit Testing

Unit testing focuses on verifying the correctness of individual components such as the speech-to-text engine, gesture generation module, and avatar animation controls. Each unit will be tested in isolation using predefined input cases and expected outputs to ensure consistent behavior.

5.2.2 Integration Testing

Integration testing validates the interactions between combined modules. For IsharaNow, this includes testing the flow from input recognition to gesture rendering, ensuring data consistency and communication across modules like NLP processing, sign translation, and video rendering subsystems.

5.2.3 System Testing

System testing evaluates the complete IsharaNow system in a real-world scenario. It includes end-to-end use cases, such as converting live speech to animated sign language and measuring performance metrics like response time, accuracy, and user interaction reliability. It ensures that the system meets all specified requirements.

5.3 Test Environment

The test environment includes the necessary hardware and software setups to replicate real-world usage. This involves testing on devices such as laptops, tablets, and smartphones, using different

operating systems and browsers. The environment includes access to the AI backend, speech-to-text APIs, camera and microphone hardware, and avatar rendering modules to ensure full-system testing under diverse conditions.

5.4 Schedule

The testing schedule is aligned with the project development phases. Unit testing is conducted in parallel with development during each sprint. Integration testing follows the completion of major feature modules, while full system testing is scheduled before major milestone submissions.

5.5 Control Activities

Control activities include test planning, version control of test scripts, documentation of test results, and defect tracking. Each bug or issue is logged and categorized based on severity and impact. Regular review meetings are conducted to assess testing progress and to make informed decisions about fixes, improvements, or rollbacks. Test cases and results are stored and versioned to maintain traceability throughout the project lifecycle.

5.6 Resources

The testing phase of IsharaNow requires a combination of human expertise, suitable hardware, and the right software tools to ensure the effective execution and validation of all system components.

5.6.1 Human Resources

The testing team consists of software developers, QA testers, domain experts (e.g., sign language professionals), and academic supervisors. Each member contributes to different testing levels—unit, integration, and system—ensuring coverage from technical validation to end-user experience evaluation.

5.6.2 Hardware Resources

The testing environment utilizes mid-to-high-end computing devices such as laptops with multi-core processors and dedicated GPUs for real-time rendering. Peripheral devices like high-resolution webcams and noise-canceling microphones are employed to capture input data for gesture and speech processing. Mobile and tablet devices are also used to test compatibility across platforms.

5.6.3 Software Resources

Software tools used include IDEs like Visual Studio Code, testing frameworks like PyTest or Selenium (for UI validation), and debugging utilities. In addition, version control is managed using Git, while collaboration and reporting occur via platforms like Jira or Trello.

5.7 Test Case Design and Description

Test Case ID	TC_001
Test Engineer	Muneef
Functional Area	Sign Language Translation
Test Name	Audio-to-Sign Conversion
Objective	To verify real-time audio input is converted to gloss and displayed via sign video.
Environment	Chrome Browser on MacOS Sequoia
Strategy	1. Open the Meeting Room page 2. Enable microphone and speak a sentence
Expected Result	The sentence is transcribed, glossed, and the sign video is played correctly.
Test Result	Pass

Table 5.1: Test Case 1 for *IsharahNow*

Test Case ID	TC_002
Test Engineer	Waleed
Functional Area	User Authentication
Test Name	Signup Form Validation
Objective	To confirm that the signup form accepts only valid input and creates a new account.
Environment	Signup Page – IsharahNow (Desktop browser)
Strategy	1. Enter a valid name, email, and password 2. Click "Sign Up"
Expected Result	Account is successfully created and the user is redirected to the homepage.
Test Result	Pass

Table 5.2: Test Case 2 for *IsharahNow*

Test Case ID	TC_003
Test Engineer	Muneef
Functional Area	Sign Video Synchronization
Test Name	Multi-User Sign Video Visibility
Objective	To ensure that all participants in a call can view the generated sign video.
Environment	IsharahNow Meeting Room (Two Users, Chrome)
Strategy	1. User A speaks a sentence 2. User B checks if the video appears
Expected Result	The sign video is visible to both users at the same time without delay.
Test Result	Pass

Table 5.3: Test Case 3 for *IsharahNow*

Test Case ID	TC_004
Test Engineer	Hamza
Functional Area	Speech Recognition
Test Name	Continuous Sentence Processing
Objective	To verify that multiple sentences spoken in one go are captured and translated.
Environment	Chrome – Windows 11 – Meeting Room
Strategy	1. Speak three short sentences without long pauses 2. Wait for all videos to appear
Expected Result	All sentences should be processed and videos played one after another.
Test Result	Fail - Only first sentence translated, rest missed due to request overload

Table 5.4: Test Case 4 for *IsharahNow*

Test Case ID	TC_005
Test Engineer	Waleed
Functional Area	Meeting Scheduler
Test Name	Schedule a New Meeting with Invite Link
Objective	To verify that users can successfully schedule a meeting and generate a shareable invite link
Environment	IsharahNow Web Application – Scheduler Page (Chrome – Windows)
Strategy	1. Log in and go to the “Schedule Meeting” section 2. Copy the generated link and open it in another tab
Expected Result	The meeting is scheduled
Test Result	Pass

Table 5.5: Test Case 5 for *IsharahNow*

5.8 Status Reporting

Testing status is tracked using a dashboard that logs the completion rate of each testing phase. Reports include summary statistics on passed/failed test cases, open bugs, and severity distribution. Daily or weekly stand-up reports help assess progress and allow timely identification of issues. Summary reports are submitted to supervisors for review and milestone tracking.

5.9 Major Deliverables

The testing phase produces several important artifacts that document testing results, methods, and outcomes:

5.9.1 Test Plan

A comprehensive document that outlines the overall testing approach, environment setup, roles, risks, and schedules. It serves as a blueprint for how testing is organized and executed.

5.9.2 Test Cases

A complete set of documented test cases, including both automated and manual scenarios, with their execution results. These documents provide proof of testing and help future teams with regression testing or debugging.

5.10 Risks and Assumptions

Several risks could impact the testing process:

- **Accuracy Risks:** Sign language translation errors due to model limitations or ambiguous input.
- **Hardware Dependency:** Performance variations on lower-end devices could affect gesture clarity.
- **Environmental Constraints:** Poor lighting or noisy environments could reduce system performance.

Assumptions include the availability of internet connectivity, stable APIs for AI models, and consistent system performance across platforms.

Chapter Six: Critical Evaluation

6.1 Success Criteria

The success of IsharaNow was measured by specific criteria established during the planning phase. These included:

- Real-time translation of spoken or written input into accurate sign language gestures.
- Responsiveness within 1–2 seconds for seamless interaction.
- Accuracy of translation above 90% based on expert evaluations.
- Cross-platform compatibility and minimal hardware requirements.
- Positive feedback from end users during User Acceptance Testing (UAT).

These benchmarks served as guiding metrics to evaluate whether the system met its functional and social objectives.

6.2 Degree of Success

IsharaNow achieved a high degree of success, particularly in the implementation of real-time gesture translation and user interface responsiveness. The avatar animation engine performed with consistent accuracy during most tests, and the modular architecture allowed for flexible improvements throughout the development cycle.

While certain features like advanced dialect recognition and facial emotion mapping were partially implemented or simplified due to time constraints, the core objectives were met effectively. Feedback from academic evaluators and simulated end users indicated high satisfaction with the system's ease of use, accessibility, and performance.

6.3 Learning Experience

The development of IsharaNow provided valuable technical and professional learning opportunities for the team. Technically, the team gained hands-on experience with AI frameworks, computer vision libraries, real-time video rendering, and integration of NLP with front-end interfaces.

Professionally, the project emphasized teamwork, iterative development, version control, documentation standards, and stakeholder communication. Navigating challenges such as model training bottlenecks, UI testing bugs, and system optimization fostered a deeper understanding of both software engineering principles and project management practices.

6.4 Assumptions and Limitations

Several assumptions were made during the development process:

- Users would have access to stable internet and modern hardware.
- The spoken input would be in English with minimal background noise.
- Standard variants of sign languages (e.g., ASL) would suffice for initial implementation.

These assumptions led to a few limitations:

- Limited support for non-standard dialects or regional sign language variations.
- Lack of integration with third-party accessibility tools like hearing aids or Braille readers.
- Challenges in adapting to extreme lighting or acoustical environments.

Despite these limitations, the core functionality remained reliable for the intended user base.

6.5 Future Enhancements

To expand its reach and effectiveness, several enhancements are envisioned for future versions of IsharaNow:

- **Multi-language and dialect support** to broaden global accessibility.
- **Advanced facial expression recognition** for more nuanced sign interpretations.
- **Offline mode capabilities** are available for users with limited or no internet access.
- **Mobile application deployment** for easier access and wider adoption.
- **Integration with assistive wearables** like haptic gloves or smart glasses.
- **Crowdsourced gesture training modules** to continually refine the AI model through user feedback.

These improvements will ensure that IsharaNow continues to evolve as a robust, inclusive, and impactful communication tool.

References

[1] World Health Organization (WHO), "Hearing loss," [Online]. Available: https://www.who.int/health-topics/hearing-loss. [Accessed: 25-Apr-2025].

[2] The National Council on Aging (NCOA), "Hearing Loss Statistics," [Online]. Available: https://www.ncoa.org/adviser/hearing-aids/hearing-loss-statistics/#toc-main-table. [Accessed: 25-Apr-2025].

[3] American University School of Education, "Qualitative vs Quantitative Research," [Online]. Available: https://soeonline.american.edu/blog/qualitative-vs-quantitative/. [Accessed: 25-Apr-2025].

[4] Virtasant, "SDLC Methodologies Explained," [Online]. Available: https://www.virtasant.com/blog/sdlc-methodologies. [Accessed: 25-Apr-2025].

[5] Rezaid, "SDLC Waterfall Model: The 6 Phases You Need to Know About," [Online]. Available: https://www.rezaid.co.uk/post/sdlc-waterfall-model-the-6-phases-you-need-to-know-about. [Accessed: 25-Apr-2025].

[6] Relevant Software, "Agile Software Development Lifecycle Phases Explained," [Online]. Available: https://relevant.software/blog/agile-software-development-lifecycle-phases-explained/. [Accessed: 25-Apr-2025].

[7] Software Testing Material, "V-Model in SDLC," [Online]. Available: https://www.softwaretestingmaterial.com/v-model-in-sdlc/. [Accessed: 25-Apr-2025].

[8] Plutora, "Incremental Model: What and How to Implement It," [Online]. Available: https://www.plutora.com/blog/incremental-model-what-and-how-to-implement-it. [Accessed: 25-Apr-2025].

[9] Mbloging, "What is React JS?," [Online]. Available: https://www.mbloging.com/post/what-is-react-js. [Accessed: 25-Apr-2025].

[10] Python Software Foundation, "Welcome to Python.org," [Online]. Available: https://www.python.org/. [Accessed: 25-Apr-2025].

[11] Amazon Web Services (AWS), "AWS - Cloud Computing Services," [Online]. Available: https://aws.amazon.com/. [Accessed: 25-Apr-2025].

[14] Microsoft, "Visual Studio Code," [Online]. Available: https://code.visualstudio.com/. [Accessed: 25-Apr-2025].

[15] Google Cloud, "Speech-to-Text API Documentation," [Online]. Available: https://cloud.google.com/speech-to-text/docs. [Accessed: 25-Apr-2025].

[16] Mozilla, "DeepSpeech: Open Source Speech Recognition," [Online]. Available: https://github.com/mozilla/DeepSpeech. [Accessed: 25-Apr-2025].

[17] OpenCV.org, "OpenCV Library," [Online]. Available: https://opencv.org/. [Accessed: 25-Apr-2025].

[18] TensorFlow, "TensorFlow: An End-to-End Open Source Machine Learning Platform," [Online]. Available: https://www.tensorflow.org/. [Accessed: 25-Apr-2025].

[19] P. Kadam, "Natural Language Processing: Techniques and Applications," International Journal of Computer Applications, vol. 181, no. 11, pp. 20-27, 2019.

[20] W3C, "Web Content Accessibility Guidelines (WCAG) 2.1," [Online]. Available: https://www.w3.org/TR/WCAG21/. [Accessed: 25-Apr-2025].

[21] WebAIM, "Introduction to Web Accessibility," [Online]. Available: https://webaim.org/intro/. [Accessed: 25-Apr-2025].

[22] B. Liang and M. Hu, "Real-Time Sign Language Translation Using Deep Learning," Procedia Computer Science, vol. 174, pp. 362-369, 2020.

[23] SignWriting Foundation, "SignWriting Home Page," [Online]. Available: https://www.signwriting.org/. [Accessed: 25-Apr-2025].

[24] GitHub, "MediaPipe by Google: Cross-Platform ML Solutions for Live and Streaming Media," [Online]. Available: https://mediapipe.dev/. [Accessed: 25-Apr-2025].

[25] Unity Technologies, "Unity Real-Time Development Platform," [Online]. Available: https://unity.com/. [Accessed: 25-Apr-2025].

[26] J. Starner, "Visual Recognition of American Sign Language Using Hidden Markov Models," Master's Thesis, MIT Media Lab, 1995.

[27] G. Cooper and J. Bowers, "Sign Language Recognition," Journal of Computer Vision and Applications, vol. 7, no. 4, pp. 230-245, 2021.

[28] V. Kumar and B. Chaurasia, "A Review on Speech Recognition Techniques," International Journal of Advanced Research in Computer and Communication Engineering, vol. 5, no. 3, pp. 431-435, 2016.

[29] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735-1780, 1997.

[30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," Journal of Machine Learning Research, vol. 15, pp. 1929-1958, 2014.

[31] ISO, "ISO/IEC 40500:2012 - Information technology — W3C Web Content Accessibility Guidelines (WCAG) 2.0," [Online]. Available: https://www.iso.org/standard/58625.html. [Accessed: 25-Apr-2025].

[32] Kaggle, "Sign Language MNIST Dataset," [Online]. Available: https://www.kaggle.com/datasets/datamunge/sign-language-mnist. [Accessed: 25-Apr-2025].

[33] OpenAI, "Introduction to GPT Models," [Online]. Available: https://platform.openai.com/docs/models/gpt-4. [Accessed: 25-Apr-2025].

[34] IEEE, "Ethical Considerations in Artificial Intelligence and Autonomous Systems," IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems, 2019.

[35] Microsoft Azure, "Azure AI Services," [Online]. Available: https://azure.microsoft.com/en-us/services/cognitive-services/. [Accessed: 25-Apr-2025].

[36] Amazon Rekognition, "Deep Learning-Based Image and Video Analysis," [Online]. Available: https://aws.amazon.com/rekognition/. [Accessed: 25-Apr-2025].

[37] ResearchGate, "Survey on Real-Time Sign Language Recognition Systems," [Online]. Available: https://www.researchgate.net/publication/347837455_Survey_on_Sign_Language_Recognition. [Accessed: 25-Apr-2025].

[38] GitHub, "Sign Language Recognition Using Convolutional Neural Networks," [Online]. Available: https://github.com/ardamavi/Sign-Language-Digits-Dataset. [Accessed: 25-Apr-2025].

[39] Google Developers, "Web Speech API Documentation," [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API. [Accessed: 25-Apr-2025].

Appendix A: Glossary

AI (Artificial Intelligence): The simulation of human intelligence processes by machines, especially computer systems, including learning, reasoning, and self-correction.

ASL (American Sign Language): A complete, natural language that has the same linguistic properties as spoken languages, used predominantly by Deaf communities in the United States and parts of Canada.
CV (Computer Vision): A field of artificial intelligence that trains computers to interpret and understand visual information from the world.
NLP (Natural Language Processing): A branch of AI focused on enabling machines to understand, interpret, and generate human language.
SDK (Software Development Kit): A collection of software development tools in one installable package to create applications for specific platforms.
UAT (User Acceptance Testing): A type of testing where actual users test the software to ensure it can handle required tasks in real-world scenarios.
UI (User Interface): The space where interactions between humans and machines occur, focusing on ease of use and user experience.
IDE (Integrated Development Environment): A software suite that consolidates basic tools required to write and test software.
OS (Operating System): System software that manages computer hardware, software resources, and provides common services for computer programs.
Speech-to-Text: Technology that converts spoken language into written text automatically.
Sign Language Recognition: The process of detecting and interpreting sign languages from visual inputs such as images or videos.
Gloss: A simplified, symbolic representation of a sign in written form, often used to bridge between spoken and sign languages in translation systems.
SignWriting: A system of writing sign languages visually using symbols for handshapes, movements, and facial expressions.
OpenCV: An open-source computer vision and machine learning software library designed to accelerate the use of AI in visual data processing.
Pose Estimation: The process of detecting the position and orientation of a person or object from an image or video stream.
Avatar Animation: The digital representation of a character performing gestures or movements, often used to visually display sign language.
Cloud Computing: Delivery of computing services like servers, storage, databases, and networking over the Internet ("the cloud").
AWS (Amazon Web Services): A secure cloud services platform offering computing power, database storage, content delivery, and other functionality to help businesses scale and grow.
Machine Learning (ML): A branch of AI focused on building systems that learn from data to make decisions or predictions without being explicitly programmed.
Gesture Recognition: A process by which gestures made by a user are interpreted by a system to interact or control devices.
Sign Language Translation Pipeline: A sequence of processes converting speech/text to gloss, then to pose estimation, then animated video output.

Appendix B: User Manual

-

IsharahNow

Version 1.0
27/04/2025

Table of Contents

1.0 Introduction

The purpose of this user manual is to provide the intended users of ***IsharahNow*** with the necessary information required to use the system effectively and properly.

IsharahNow is an AI-driven real-time sign language interpretation platform designed to bridge the communication gap between individuals with hearing impairments and those who do not use sign language.

This manual aims to guide users through the system's features, functions, and operations, ensuring a smooth and accessible experience for all users.

2.0 Overview

Technology is advancing at rapid rates, transforming the way we interact with the world around us. However, inclusive communication solutions for the deaf and hard-of-hearing community have often lagged behind. In a world increasingly focused on accessibility, there remains a significant gap for individuals who rely on sign language for everyday interactions. This is where IsharahNow comes into play.

IsharahNow is an AI-driven real-time sign language interpretation system designed to bridge the communication gap between hearing and non-hearing individuals. At the core of the system is a powerful integration of speech recognition, natural language processing (NLP), computer vision, and avatar animation technologies that work together to deliver real-time sign language translations.

The system captures spoken input through the user’s microphone, processes it using advanced AI models, and converts it into corresponding sign language gestures displayed through a digital animated avatar. The intuitive interface of ***IsharahNow*** ensures that users can easily engage with the system, viewing real-time translations with minimal technical knowledge.

Currently, ***IsharahNow*** offers functionalities such as audio-to-sign translation, text-to-sign translation, meeting scheduling, and multi-user meeting support. In future versions, users will also be able to customize sign language variants, access real-time conversation history, and seamlessly integrate IsharahNow into third-party video conferencing applications, making communication even more accessible and interactive.

2.1 Conventions

AI	Artificial Intelligence
ASL	American Sign Language
CV	Computer Vision
NLP	Natural Language Processing
SDK	Software Development Kit
UAT	User Acceptance Testing
UI	User Interface

Table 2.1: Document Conventions

3.0 Getting Started

This section guides users on setting up and starting with ***IsharahNow*** for the first time. Before using ***IsharahNow***, ensure that you have a stable internet connection and a device that meets the minimum system requirements. IsharahNow is a web-based application designed to operate seamlessly on desktop browsers and mobile devices.

3.1 Setup Considerations

Access the ***IsharahNow*** platform through your web browser or by visiting the official URL provided by the development team.

Ensure that your device fulfills the following requirements:

- A modern web browser (Google Chrome, Mozilla Firefox, or Microsoft Edge recommended)
- Stable internet connection (minimum 5 Mbps recommended for smooth video streaming)
- A functioning microphone for capturing speech input
- A working webcam (optional for future features)

To start using ***IsharahNow***:

1. **Create an Account**: Sign up using your name, email address, and password through the Sign-Up page.

2. **Login**: After registration, log in using your credentials to access the main dashboard.

3. **Microphone Permissions**: Allow microphone access when prompted by your browser to enable speech-to-sign translation.

4. **System Compatibility**: IsharahNow is optimized for use on the following:

    - **Desktop/Laptop**: Windows 10/11, macOS 10.15 or later

    - **Mobile** (Beta Mode): Android 10 or above (Minimum 2GB RAM recommended), IOS 14 or above

5. **Browser Settings**: Ensure pop-ups and microphone permissions are enabled for a seamless experience.

3.2 User Access Considerations

The ***IsharahNow*** system is primarily developed for individuals who seek to bridge communication gaps with the deaf and hard-of-hearing community in a more effective, modern, and accessible way.

It is designed to be user-friendly for:

- Individuals with hearing impairments

- Educators and interpreters working in inclusive educational environments

- Professionals conducting meetings or events requiring accessibility support

- Organizations and businesses aiming to provide sign language accessibility

- Friends, family members, and colleagues of individuals who use sign language

IsharahNow ensures that users, regardless of their technical background, can access real-time sign language translation features with minimal setup, thereby promoting greater inclusivity and communication ease across personal, educational, and professional settings.

3.3 Accessing the Application

The ***IsharahNow*** platform can be accessed easily through a web browser without the need for installation. Users can obtain the access link via the official website or through a QR code provided during onboarding sessions, promotional materials, or official communications.

Once the QR code is scanned, users will be redirected to the IsharahNow web application where they can:

1. **Create an Account**: Fill in the required fields to sign up.

2. **Grant Microphone Access**: Upon first login, allow the application to access the microphone for speech recognition functionality.

3. **Access the Main Dashboard**: After login, users are directed to the homepage, where they can initiate real-time translation sessions or schedule new meetings.

Currently, ***IsharahNow*** is optimized for desktop browsers, with mobile access available in beta testing. No separate app installation is required for the current version.

3.4 System Organization

The ***IsharahNow*** system has been developed using a combination of modern web technologies and artificial intelligence frameworks. The development and maintenance of the system require technical skills related to machine learning, natural language processing (NLP), speech recognition, and web application development.

The core technologies used in IsharahNow include:

- **Frontend**: Built using **ReactJS** for a responsive and interactive user interface.

- **Backend**: Powered by **Node.js** and cloud-based services such as **AWS** for efficient data processing and real-time translation handling.

- **Machine Learning Models**: Developed and deployed using **Python** for speech-to-sign language conversion and gesture recognition.

- **Database Management**: **ClerkJS** is used for managing user data, meeting records, and translation logs, although the database interaction is minimal in the current release version.

- **Speech Recognition**: Integrated using APIs like **Google Speech-to-Text** for capturing and processing spoken input.

The development process follows the **Agile methodology**, ensuring continuous testing, iterative releases, and flexibility in accommodating evolving user requirements.

Future versions of IsharahNow are planned to incorporate deeper database functionalities for analytics and user behavior insights.

3.4.1 Home Page

The **Home Page** is the starting screen where users can access the core functionalities of ***IsharahNow***. Upon successful login, users are directed to the main dashboard, which provides quick access to key features.

3.4.2 Upcoming Meetings Page

The **Upcoming Meetings Page** allows users to view and manage all meetings that have been scheduled but have not yet occurred. This section helps users keep track of their planned sessions where real-time sign language interpretation will be required.

3.4.3 Previous Meetings Page

The **Previous Meetings Page** allows users to view a record of all meetings that have already taken place. This section helps users review past sessions and track their communication activities through IsharahNow.

3.4.4 Recordings Page

The **Recordings Page** provides users with access to session recordings captured during live meetings or translation sessions. This feature is designed to help users review, study, and analyze conversations where real-time sign language translation was utilized.

3.4.5 Personal Room

The **Personal Room** is a dedicated private space where users can host personal meetings, practice real-time sign language translation, or test system functionalities before joining a formal session.

3.4.6 Meeting Room

The **Meeting Room** in **IsharahNow** is the main virtual space where users can conduct real-time conversations with built-in sign language translation support. This feature is designed to enable accessible communication for all participants during live meetings.

4.0 Troubleshooting

4.1 Microphone Access Denied

Ensure that your browser has permission to access the microphone. Go to browser settings, enable microphone access for the IsharahNow website, and refresh the page.

4.2 Avatar Not Displaying Translation

Check if your internet connection is stable. A slow or interrupted connection can delay or block real-time translation. Try refreshing the page or reconnecting to a stronger network.

4.3 Cannot Join or Start Meeting

Make sure you are logged in to your IsharahNow account. If the issue persists, verify that the meeting link is correct and not expired. Contact system support if necessary.

4.4 Support

Contact	Organization	Email
Muneef Rehman	SZABIST, Dubai	muneefsyed01@gmail.com
Waleed Fahad	SZABIST, Dubai	waleedbakhsh20@gmail.com
Hamza Yaqoob	SZABIST, Dubai	hyaqoob574@gmail.com

Table 4.1: Support Point of Contact

Appendix C: Developer Manual

-

IsharahNow

Version 1.0
27/04/2025

Table of Contents

1.0 Introduction

This Developer Manual is intended to provide technical guidelines for developers working on the ***IsharahNow*** platform. It outlines the system architecture, tools, frameworks, development environment setup, code organization, APIs, deployment process, and future scalability considerations.

IsharahNow is an AI-based real-time sign language interpretation system designed to bridge communication gaps between hearing individuals and the deaf or hard-of-hearing community. Developers using this manual will be able to maintain, enhance, or expand the system effectively by following the documented structure and best practices.

2.0 System Overview

The ***IsharahNow*** system is a web-based platform that integrates **speech recognition**, **natural language processing (NLP)**, **computer vision**, and **animated sign language avatars** to deliver real-time translation services. It is designed to be modular, scalable, and accessible across different devices and platforms.

The major components of the system are:

- **Frontend (Client-Side Web Application)**
  Built with **NextJS** and **ReactJS**, the frontend handles user interactions, speech capture, displaying the animated avatar, and managing meeting sessions.

- **Backend (Server-Side Application)**
  Built with **Node.js** and hosted on **AWS**, the backend handles user authentication, meeting management, API requests, and system communication.

- **Machine Learning Models**
  Deployed using **Python**, responsible for gloss generation, pose estimation, and avatar animation.

- **Database**
  A lightweight **ClerkJS** database is used to store user profiles, meeting logs, and translation history (minimal use in current version).

- **Cloud Services**
  Services like **AWS S3** (for storage), **AWS Lambda** (for running translation microservices), and **AWS API Gateway** (for routing requests) are integrated for

scalability and performance.

- **APIs**
    - **Speech-to-Text API:** Converts speech input to text in real-time.
    - **Gloss-to-Pose API:** Maps text/gloss to corresponding sign language gestures.

This modular breakdown allows developers to maintain or update individual components without impacting the overall system structure.

3.0 Development Environment Setup

To contribute to or modify ***IsharahNow***, developers must set up a proper development environment with all required tools and dependencies.

3.1 System Requirements

- **Operating System**: Windows 10/11, macOS 10.15+, or a modern Linux distribution (Ubuntu 20.04+ recommended)
- **Processor**: Minimum Quad-Core CPU
- **Memory**: At least 8 GB RAM (16 GB recommended for machine learning tasks)
- **Storage**: 20 GB of free disk space
- **Internet**: Stable high-speed internet connection (5 Mbps minimum)

3.2 Software Requirements

- **Node.js** (version 18.x or above)
- **npm** (comes with Node.js) or **Yarn**
- **Python** (version 3.8 or above) — for machine learning model management
- **AWS CLI** (for cloud resource access)
- **Visual Studio Code** (recommended IDE)

- **Git** (version control)

Optional for advanced contributions:

- **Docker** (for containerized deployments)
- **Postman** (for API testing)

3.3 Setup Instructions

1. Clone the repository

```
git clone https://github.com/muneefrehman/IsharahNow.git
```

```
cd IsharahNow
```

2. Install Frontend Dependencies

```
npm install
```

3. Configure Environment Variables

   Create `.env` files in frontend with keys such as:

```
NEXT_PUBLIC_CLERK_PUBLISHABLE_KEY=<your_publishable_key>
CLERK_SECRET_KEY=<your_secret_key>

NEXT_PUBLIC_CLERK_SIGN_IN_URL=/sign-in
NEXT_PUBLIC_CLERK_SIGN_UP_URL=/sign-up

NEXT_PUBLIC_STREAM_API_KEY=<your_stream_api_key>
STREAM_SECRET_KEY=<your_stream_secret_key>

NEXT_PUBLIC_BASE_URL=localhost:3000

NEXT_PUBLIC_AWS_ACCESS_KEY_ID=<your_aws_access_key_id>
NEXT_PUBLIC_AWS_SECRET_ACCESS_KEY=<your_aws_secret_access_key>
NEXT_PUBLIC_AWS_REGION=<your_aws_region>
```

4. Run frontend locally

```
npm run dev
```

5. Test connectivity

   Visit `http://localhost:3000` to verify the frontend is working and connected to backend APIs.

4.0 Code Structure

The ***IsharahNow*** project follows a clean, modular structure organized for scalability, maintainability, and clear separation of concerns. Below is the essential directory breakdown.

4.1 Project Directory Overview

```
isharahnow
    ├── actions
    ├── app
    │   ├── (auth)
    │   │   ├── sign-in
    │   │   │   └── [[...sign-in]]
    │   │   ├── sign-up
    │   │       └── [[...sign-up]]
    │   ├── (root)
    │       ├── (home)
    │       │   ├── personal-room
    │       │   ├── previous
    │       │   ├── recordings
    │       │   └── upcoming
    │       ├── meeting
    │           └── [id]
    ├── components
    │   └── ui
    ├── constants
    ├── hooks
    ├── lib
    ├── providers
    ├── public
```

4.2 Key Functional Areas

Folder	Purpose
`app/`	Defines the structure and layout of all pages and core routes
`components/`	Contains all reusable UI components used across pages
`hooks/`	Provides custom React hooks to fetch and manage data
`lib/`	Houses utility/helper functions shared across the project
`providers/`	Manages application context, mainly video call streaming
`constants/`	Stores application-wide constants like meeting types
`public/`	Contains icons, images, and static assets served by the frontend

Table 4.1: Key Functional Areas

5.0 APIs and Communications Flow

The ***IsharahNow*** system relies on RESTful APIs to handle real-time data communication between the frontend, backend, and machine learning services.
Understanding how the APIs interact is essential for any developer aiming to maintain or extend the platform.

5.1 API Overview

API	Description
`/api/auth`	Handles user authentication and session management
`/api/meetings`	Manages meeting creation, joining, updating, and listing
`/api/speech-to-text`	Converts audio input to text using speech recognition
`/api/gloss-to-pose`	Converts gloss (simplified sentence) to avatar pose animations
`/api/users`	Fetches and updates user profile information

Table 4.2: APIs

5.2 Communication Flow

The flow of how data moves in IsharahNow during a real-time meeting:

1. **User speaks into microphone:**
    - Frontend captures audio using Web APIs.
2. **Speech captured is sent to `/api/speech-to-text`**
    - Backend receives audio and uses a Speech Recognition Engine.
    - Output: Text transcript.
3. **Text transcript is sent to `/api/gloss-to-pose`**
    - Backend maps the sentence into gloss (simplified signs).
    - Gloss is used to generate pose sequences for the sign language avatar.
4. **Pose data is streamed to frontend**
    - Avatar animation is updated in real-time based on the pose data.
5. **Meeting Synchronization**
    - If multiple users are connected, translated glosses and poses are shared via WebSocket or REST APIs to synchronize all views.
6. **Optional: Save Meeting Records**
    - After the meeting ends, metadata (like timestamps, number of translations) can optionally be stored for session history.

5.3 Data Formats

- **Speech-to-Text API Response Example**

```
{
 "text": "How are you?"
}
```

- **Gloss-to-Pose API Response Example**

```
{
"gloss": "HOW YOU",
"poses": ["pose1.mp4", "pose2.mp4", "pose3.mp4"]
}
```

6.0 Production Deployment

This section provides instructions on how to deploy ***IsharahNow*** for production environments.

6.1 Production Deployment

1. Connect your GitHub repository to Vercel.

2. Set up the following Environment Variables in Vercel Dashboard:

```
NEXT_PUBLIC_API_BASE_URL
NEXT_PUBLIC_STREAM_API_KEY
```

3. Deploy directly via the Vercel UI (one-click deployment).

6.3 Post-Deployment Checklist

- Test microphone and speech-to-text functionality.

- Verify real-time sign avatar rendering.

- Confirm meeting creation and joining flow.
- Monitor API request logs and error rates.
- Enable HTTPS and SSL security for production.

7.0 Testing and Debugging Guidelines

To ensure that ***IsharahNow*** remains reliable, accurate, and user-friendly, developers must follow structured testing and debugging practices during development and deployment phases.

7.1 Testing Strategy

Testing Level	Purpose	Tools
Unit Testing	Test individual frontend components, backend APIs, and utility functions	Jest, React Testing Library
Integration Testing	Test how components and APIs interact together	Postman, Jest
System Testing	Validate the entire user flow — speech input to sign output	Manual browser testing, End-to-End Testing
User Acceptance Testing (UAT)	Test system usability with real users or testers	Manual, user feedback surveys

Table 7.1: Testing Strategies

7.2 Areas to Test

- **Speech-to-Text Accuracy**
  Ensure speech recognition correctly transcribes spoken input into text.
- **Gloss-to-Pose Mapping**
  Verify that gloss translation produces the correct pose sequences and avatar animations.
- **Meeting Management**
  - Create, join, and leave meetings without errors.
  - Check meeting links, invites, and participant visibility.

- **Frontend Responsiveness**
  Test on different devices (desktop, mobile) and browsers (Chrome, Firefox, Edge).

- **Microphone and Audio Access**
  Confirm that the microphone permission prompt appears and speech capture works properly.

- **Avatar Rendering**
  Ensure that sign language animations display smoothly without noticeable delays.

- **Error Handling**

    - Validate user-friendly error messages for failed translations or connectivity issues.
    - Check API timeout handling and fallback responses.

7.3 Debugging Tips

- **Console Logs**
  Use `console.log()` statements in React components and Node.js APIs during local debugging to trace request/response cycles.

- **Network Tab (Browser Developer Tools)**
  Monitor real-time API requests and responses, checking for HTTP errors or slow endpoints.

- **API Response Validation**
  Always validate API responses inside frontend services to prevent breaking the UI on unexpected data formats.

- **AWS Monitoring**
  Use AWS CloudWatch to monitor backend service logs and Lambda function executions.

- **Environment Configuration**
  Double-check `.env` files if the local environment behaves differently than production — missing API keys often cause silent failures.

8.0 Support

Contact	Organization	Email
Muneef Rehman	SZABIST, Dubai	muneefsyed01@gmail.com
Waleed Fahad	SZABIST, Dubai	waleedbakhsh20@gmail.com
Hamza Yaqoob	SZABIST, Dubai	hyaqoob574@gmail.com

Table 8.1: Support Point of Contact

9.0 Conclusion

The development of ***IsharahNow*** represents a major step forward in creating accessible, real-time communication tools for the deaf and hard-of-hearing community.

This Developer Manual has been carefully designed to provide a clear understanding of the system's architecture, development practices, deployment procedures, and testing guidelines.

Developers contributing to IsharahNow are expected to:

- Maintain clean, modular, and well-documented code.
- Follow the established architecture and directory structure.
- Implement thorough testing at every stage of development.
- Ensure that the system remains user-centric, reliable, and accessible.

By adhering to these guidelines, future developers will be able to sustain and improve ***IsharahNow***, ensuring that the platform continues to grow, evolve, and fulfill its mission of bridging communication gaps through advanced technology.

Document Viewer

Turnitin Originality Report

Processed on: 01-May-2025 12:19 PKT
ID: 2663020230
Word Count: 12577
Submitted: 1

Design and Development of an AI-Driven Real-T... By Waleed, Hamza Yaqoob Syed Muneef Ur Rehman

Similarity Index	Similarity by Source	
4%	Internet Sources:	N/A
	Publications:	4%
	Student Papers:	N/A

include quoted | include bibliography | exclude small matches | mode: quickview (classic) report | print | refresh | download

1% match ("World Congress on Smart Computing", Springer Science and Business Media LLC, 2025)
"World Congress on Smart Computing", Springer Science and Business Media LLC, 2025

<1% match (Suvarna Bhagwat, Hemakshi Lohiya, Shrawani Gulhane, Atharva Dhoble, Yash Ayare. "Chapter 34 2-Way Hand Sign Talk System", Springer Science and Business Media LLC, 2024)
Suvarna Bhagwat, Hemakshi Lohiya, Shrawani Gulhane, Atharva Dhoble, Yash Ayare. "Chapter 34 2-Way Hand Sign Talk System", Springer Science and Business Media LLC, 2024

<1% match (Nodler, Joshua. "Deleted File Recovery in Ext4 File Systems", Bowling Green State University, 2025)
Nodler, Joshua. "Deleted File Recovery in Ext4 File Systems", Bowling Green State University, 2025

<1% match (Gurwinder Kaur Dua. "chapter 2 The Evolution and Development of Artificial Intelligence Interpretation Technology in the Era of Large-Scale Language Models", IGI Global, 2025)
Gurwinder Kaur Dua. "chapter 2 The Evolution and Development of Artificial Intelligence Interpretation Technology in the Era of Large-Scale Language Models", IGI Global, 2025

<1% match (Louis Vincent Sanjaya, Herolistra Baskoroputro, Andry Chowanda. "Implementation of Blockchain in Enhancing the Security of Financial Asset Management and Company Transactions Using Python Programming Language", 2023 International Conference on Informatics, Multimedia, Cyber and Informations System (ICIMCIS), 2023)
Louis Vincent Sanjaya, Herolistra Baskoroputro, Andry Chowanda. "Implementation of Blockchain in Enhancing the Security of Financial Asset Management and Company Transactions Using Python Programming Language", 2023 International Conference on Informatics, Multimedia, Cyber and Informations System (ICIMCIS), 2023

<1% match (Almadani, Mwaheb. "An AI-Driven, Secure, and Trustworthy Ranking System for Blockchain-Based Wallets.", University of Technology Sydney (Australia), 2024)
Almadani, Mwaheb. "An AI-Driven, Secure, and Trustworthy Ranking System for Blockchain-Based Wallets.", University of Technology Sydney (Australia), 2024

<1% match (publications)
D. Sleeman, N.O. Bernsen. "Artificial Intelligence – Research Directions in Cognitive Science: European Perspectives Vol.5", Routledge, 2019

<1% match (H. Fujisawa. "Pattern recognition and synthesis for sign language translation system", Proceedings of the first annual ACM conference on Assistive technologies - Assets 94 Assets 94, 1994)
H. Fujisawa. "Pattern recognition and synthesis for sign language translation system", Proceedings of the first annual ACM conference on Assistive technologies - Assets 94 Assets 94, 1994

<1% match (publications)

Syed Muneef ur Rehman

Dubai | +971 50 219 1895 | muneefsyed01@gmail.com | Portfolio Link | LinkedIn Link

WORK EXPERIENCE

Zoom Digital Dubai — **Dubai, UAE**

Backend Development Intern — *Jan 2024 - Oct 2024*

- Developed and optimized RESTful APIs leading to a 40% reduction in response time, enhancing user experience.
- Collaborated with cross-functional teams to refactor legacy code, improving system performance by 30% and decreasing server downtime by 15%, supporting a seamless deployment process.
- Implemented automated testing procedures that decreased bug reports by 50%, enhancing software reliability across three major projects within a six-month timeline.

Afiniti Ltd. — **Karachi, Pakistan**

Software Quality Assurance Tester — *Jul 2020 - Sep 2020*

- Executed comprehensive testing protocols on an AI-driven customer care system, achieving the target accuracy rate and exceeding industry standards by 10%.
- Developed and executed test plans for new software features, leading to a 30% reduction in average response time for customer inquiries, ultimately improving overall satisfaction scores by 25%.

EDUCATION

SZABIST Dubai — **Dubai, UAE**

Bachelor's in Computer Science — *Graduation Date: May 2025*

3.81 CGPA

PROJECT

SZABIST Dubai — **Dubai, UAE**

IsharaNow — *Jun 2024 - May 2025*

- Developed an AI-driven application that enhances accessibility for the hearing-impaired community.
- Achieved a 95% accuracy rate in real-time sign language translation across over 5,000 videos and online meeting sessions.

LEADERSHIP EXPERIENCE

Microsoft — **Karachi, Pakistan**

Microsoft Learn Student Ambassador — *Oct 2021 - Oct 2022*

- Developed and launched a series of workshops that increased technical skill development among attendees by 60%, leading to a rise in course enrollments in computer science programs by 30%.
- Created and distributed marketing materials that led to a 60% higher attendance rate compared to previous events, resulting in a significant boost in awareness of Microsoft programs among students.

CERTIFICATIONS

- Python Specialization
- Deeplearning.ai Specialization
- Java Spring Boot

SKILLS

- Python: Scikit-Learn, Pandas, Numpy, PyTorch, Regression Algorithms
- Java: Spring Boot, OOP, Server-side rendering
- Javascript: Node.js, React.js, Express.js

WALEED FAHAD

IT Consultant and Developer

+971-504078126 | waleedbakhsh20@gmail.com | linkedin.com/in/waleed-fahad-itconsultant

PROFESSIONAL SUMMARY

IT Consultant and Developer with expertise in implementing ERP solutions and streamlining business processes. Proficient in programming, database management, API integrations, and machine learning. Successfully delivered 50+ projects and provided user training and pre-sales support. Currently working at IT Enablers Global, focusing on IT solutions and system integrations.

WORK EXPERIENCE

IT Consultant and Developer, IT Enablers Global **July 2023 - Present**

- Implemented ERP Solutions: Delivered customized ERP systems for SMEs, real estate agencies, retail stores, and warehouses using Zoho applications.
- User Training: Provided training sessions to ensure effective system adoption by end-users.
- Project Delivery: Successfully managed and delivered 50+ projects, meeting deadlines and client expectations.
- Application Development: Developed back-end applications integrated with company ERP systems.
- Process Automation: Streamlined operations by automating workflows and integrating business applications.
- Software Documentation: Created detailed software documentation and tested APIs for reliable performance.
- Domain and DNS Management: Set up company domains and managed DNS records for secure connectivity.
- Business Process Analysis: Analyzed client operations to design and implement effective solutions.
- Pre-Sales Support: Conducted product demos, prepared proposals, and created quotations for potential clients.

EDUCATION

Bachelor's of Science in Computer Science (3.92/4.0 cGPA) **Oct 2021 - June 2025**

SZABIST University, Dubai

- Relevant courses: Software Engineering, Artificial Intelligence, Computer Networks, Object Oriented Programming, Database Management systems, etc.
- Student council member
- Newsletter editor

TECHNICAL SKILLS

- Proficiency in various programming languages (Java, C++, JavaScript, Python, Deluge)
- Proficiency in Microsoft Suite (Excel, Word, PowerPoint, Outlook)
- Configuration of DNS and Domain management (setting up DKIM, SPF, CNAME, TXT records)
- Database management on MySQL
- API integrations (OAuth tokens, Refresh tokens, and Application connections)
- App Development
- Creating Machine Learning models using Python's sci-kit Learn

PROJECTS

- Completed Warehouse management and Invoicing solutions for Black Clover LLC
- Provided ZEDESIGNS with an ERP suite, including advisory and implementation.
- Built an interactive POS application that is integrated with Accounting Solutions approved by the FTA (Federal Tax Authority UAE)
- Built Snake game from scratch on Java using classes and objects
- Built a ping pong game from scratch in C++
- Creating a real time sign language translator using machine learning models (Thesis Project for Bachelor's degree)

SOFT SKILLS

- Communication and Presentation
- Effective Problem Solving skills
- Team work and Collaboration
- Attention to Detail

LANGUAGES

- English : Native/Bilingual
- Urdu: Native/ Bilingual
- Arabic: Working Proficiency

RESEARCH WORK

- **National Business Cup Challenge 2022:** Secured a top position by proposing sustainable solutions for Coca-Cola to combat world hunger through partnerships with charities, leveraging their supply chain to align with UN SDG 2 (Zero Hunger).
- **International Research Confluence at Manipal University (ISRC):** Awarded Best Research Paper in the Data Analytics track for proposing innovative improvements to fare deduction systems in public transport, including buses, metros, and trams, specifically for Dubai's RTA.
- **Nanotechnology Research Competition, Curtin University:** Led a team of juniors to secure a finalist position by showcasing the potential of nanotechnology to improve water quality in rivers and lakes through innovative filtration techniques.
- **Nexus 2.0 Research Paper, Amity University:** Secured a podium finish in the Current Technology Trends in Knowledge Economy and Management track. Co-authored the paper "Application of AI-Technologies to Synthesize Sustainable Agricultural Knowledge," which highlighted how AI and Machine Learning can enhance crop production across various stages, focusing on UAE's agriculture sector and aligning with the UAE's 2051 National Food Security Strategy.
- **IEEE Recognition:** Research paper abstract selected for inclusion in the IEEE Student Branch's list of innovative papers in the Knowledge Economy category, and received an award for my contributions.

VOLUNTEERING EXPERIENCES

EXPO 2020 Dubai
I accompanied the Head of Campus and represented SZABIST at Expo as a volunteer. We were invited to discuss sustainability goals in the Pakistan Pavillion. I was given the task to oversee other attendees and provide assistance.

8th Convocation 2022
As a volunteer in the ceremony, I performed various tasks to ensure the guests had a memorable experience. I lead the graduation parade of the Master's graduates. I had the privilege of being on the stage, holding the medals presented to the graduates.

HAMZA YAQOOB

+971-52150-7071 · hyaqoob574@gmail.com · linkedin.com/in/hamza-yaqoob-7491b1195/

PROFILE

Experienced Account Manager & IT Sales Advisor with a strong background in IT consulting, sales, and client relationship management. Proven ability to manage multiple accounts, drive business growth through strategic upselling, and deliver tailored solutions to enhance customer satisfaction. Proficient in ERP systems, market analysis, and product demonstrations, ensuring seamless service delivery and long-term client retention. A results-driven professional with excellent communication skills, dedicated to optimizing client success and fostering business growth.

EDUCATION AND VISA STATUS

SZABIST University, UAE

Bachelor of Science in Computer Science

Visa Status

Golden Visa

EXPERIENCE

IT Enablers Global

IT Sales Advisor **June 2023 - Present**

IT Enablers Global is a leading IT services provider specializing in managed IT services, digital marketing, and business applications. As an advanced partner for Zoho and a certified partner for Microsoft, the company serves a diverse range of clients, from SMEs to large enterprises.

- **Conducted** product demonstrations and presentations, effectively highlighting the features and benefits of Zoho products.
- **Experienced** in implementing and managing Zoho People for HR process automation.
- **Worked** with retail clients, including managing accounts for businesses such as Black Clover, providing tailored solutions to enhance operational efficiency and drive growth.
- **Engaged** with clients to assess their IT requirements and business workflows, delivering customized solutions that enhanced operational efficiency
- **Collaborated** with technical teams to develop tailored solutions, driving a 30% increase in client adoption rates.
- **Provided** training sessions on Zoho products, boosting client proficiency and overall satisfaction.
- **Maintained** strong client relationships, achieving a 95% customer satisfaction rate through proactive communication and dedicated support.
- **Analyzed** market trends and competitor activities, identifying opportunities for product enhancements and new service offerings.
- **Generated** new business opportunities, contributing to a 20% growth in annual revenue through strategic sales initiatives.
- **Handled** account management for over 30 clients, maintaining long-term engagements and fostering strong business relationships.

Account Manager

- **Maintained** strong relationships, ensuring client satisfaction and long-term retention.
- **Identify** upsell opportunities, negotiate contracts, and drive revenue growth.
- **Addressed** client concerns, collaborated with teams, and ensured seamless service delivery.
- **Collaborated** with the HR department to align hiring strategies with business needs and workforce planning.
- **Conducted** demos, provided training, and helped clients optimize solutions.

SKILLS AND INTERESTS

Languages: English and Urdu

IT Skills: Technically proficient in ERP systems, with expertise in Zoho ERP. Strong understanding of business processes, data analysis, and reporting, complemented by skills in SQL and database management. Experienced in project management methodologies, including Agile and Scrum. Adept at customizing and configuring Zoho ERP modules, providing technical support and troubleshooting, as well as delivering training and user support.

Interests: Staying updated with technology trends, embracing problem-solving challenges, and engaging in strategy games. Committed to continuous learning through online courses and workshops, participating in business strategy simulations, and networking at industry conferences and webinars.