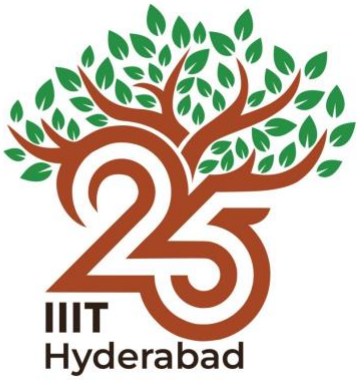

BTech. Project Report

Monsoon 2023

Stuttering Detection Platform

TEAM 3

Akash Vallamsetty

Jay Ghevariya

Rohan Eswara

Tirth Motka

Rohan Nain

Table of Contents

INTRODUCTION

- Speech Systems like ASR, TTS, SSMT … etc. are increasingly being looked into, with the rise of AI tools as well as Speech Processing Methods.

- Various models that involve both Signal Processing and AI knowledge are being developed. Currently, they are in the fledging stages, but there is a lot of scope and expectation for the future.

- Hence, it is important to design applications that can be integrated with the speech systems and allow real-time usage for the user. An example of it is IIITH's very own website asr.iiit.ac.in which provides real time speech to text conversion in multiple languages.

- In this project, we have attempted to develop a web application for stuttering detection speech system, with basic functionality of how it can be used in medical field in a doctor-patient setting.

STUTTERING & DETECTION

- Stuttering is a speech disorder characterized by disruptions in the normal flow/rhythm of speech, manifested as repeated or prolonged sounds, syllables, words, or phrases, & hesitations or pauses in speech.

- It affects people of all ages and involves tension, struggle, or avoidance behaviors during speech, and can vary in the severity & frequency depending on the genetic, neurological, and psychological factors.

- Detection of stuttering can lead to timely intervention and therapy, which can improve speech fluency and reduce negative impacts on an individual's social, emotional & professional life.

- It can also contribute to research on speech and language disorders, leading to a better understanding of the condition and potential treatments.

OBJECTIVES

In the process of enhancing the stuttering detection website, we focused on key developmental areas to elevate the site's functionality, accessibility, and security. The following objectives were set to guide the project to fruition:

LANGUAGE SUPPORT EXPANSION

- Expand the website's language capabilities by integrating English support alongside the existing Kannada version, transforming the site into a bilingual platform.

SIMPLIFY DOCKER BUILDING FOR NON TECH USERS

- Develop and integrate a user-friendly webpage that enables non-technical personnel to effortlessly build and deploy Docker containers for the stuttering detection website, thereby democratizing the setup process and enhancing accessibility to project deployment.

WEBSITE DEPLOYMENT

- Deploy the updated website onto the server provided by the All India Institute of Speech and Hearing (AIISH), ensuring full operational integration with their system.

SECURITY AND DATABASE MANAGEMENT

- Conduct a comprehensive security assessment of the website to identify and address vulnerabilities, and enhance the management of the website's database to ensure robustness and data integrity

CHOOSING THE TECH STACK

- React provides a robust and efficient way to build user interfaces. It has a large community and many libraries and tools available, making it easy to customize and extend your application. React also uses a virtual DOM, which allows for efficient rendering of complex UI components.

- FastAPI is a modern and high-performance web framework that is well-suited for building APIs that serve ML models. Its async capabilities and automatic generation of OpenAPI and Swagger documentation make it a great choice for building scalable and maintainable ML model pipelines.

- Flask is a lightweight web framework that is easy to use and ideal for building small to medium-sized web applications. It is highly customizable and provides a simple and flexible approach to building APIs and web applications.

- MongoDB is a popular NoSQL document database that is designed for high availability, scalability, and performance. Its flexible schema and ability to handle large volumes of unstructured data make it well-suited for storing and managing data in ML applications.

- Docker packages an application and its dependencies into a portable container for easy deployment and management across different environments. The consistent, portable, and isolated environment provided by Docker maintains uniformity in deployment.

- JWTs allow stateless authentication, reducing the server load and enabling efficient and scalable authentication and authorization, especially in distributed systems. Self-contained JWTs can be used across different domains and platforms, including additional information for fine-grained access control and authorization policies.

IMPLEMENTATION:

LANGUAGE SUPPORT EXPANSION

- ### Frontend Development

  A new dropdown menu—has been added to the patient's demographic details form on the doctor's webpage. This menu allows for the selection of the patient's preferred language for conducting the stuttering test. The interface has been updated to incorporate this new field seamlessly, maintaining the webpage's clean and navigable layout.

- ### Backend Development

  - Upon selection, the patient's preferred language is now stored or updated in the browser's local storage. This ensures that the chosen language setting persists throughout the testing process.
  - When questions and passages for the test are being loaded into the database, an additional field specifying the language of each item has been added. This allows the database to house multilingual content effectively.
  - As patients commence their test, the back-end code has been adapted to filter and display only those questions and passages that align with the language stored in the local storage. This dynamic filtering ensures a customized test experience for each patient.

WEBSITE DEPLOYMENT

- ### Pre-Deployment Preparation

  Our existing website was fully containerized using Docker, which ensured an isolated environment for our application and its dependencies, facilitating a straightforward deployment process.

- ### Server Access and Container Build

Upon gaining access to the server provided by AIISH, we proceeded with the construction of our Docker containers on the server. This was a crucial step in transitioning from a local development environment to a live educational environment.

- Network Configuration and Deployment:

  In alignment with the organization's policy of limiting access to its internal network, we configured our deployment to use a private IP address assigned to the server. This measure was necessary to adhere to the network security and access protocols of the organization.

- Port Exposure and Service Activation:

  We exposed the specific port on which our Nginx server was configured to run the website. This port was opened on the server's firewall to allow for internal network traffic, thereby making the website accessible within the organization's private network.

DOCKER BUILDING PAGE FOR NON TECH PEOPLE

- Design of the Interface:
  - We designed a clean and intuitive web interface where users can find clearly labeled buttons for each Docker operation, such as 'Build', 'Start', and 'Stop'. This design allows users to manage Docker containers without the need for command-line interaction.
- Implementation of Docker Operations:
  - Behind each button, we implemented the corresponding Docker commands:
    - **Build**: Tied to the docker-compose build command to construct the Docker images.
    - **Start**: Connected to the docker-compose up command to start the containers.
    - **Stop**: Linked to the docker-compose down command to stop and remove containers, networks, images, and volumes.

SECURITY AND DATABASE

- Utilized reverse engineering and manual testing to identify and rectify unprotected routes within the website, enhancing endpoint security.

- Optimized data storage by saving audio files as BLOBs in MongoDB, facilitating efficient data management.

CHALLENGES AND OTHER WORK:

- SSL Deployment and Expiry:
    - Successfully deployed the project on an internal server with SSL, enhancing security with encrypted connections.
    - Encountered SSL certificate expiry; current deployment is maintained on a local server without SSL.
- Deployment Research:
    - Dedicated significant effort to understand deployment nuances, such as port mapping and forwarding, to ensure smooth application delivery.
    - Utilized ngrok to create secure tunnels to local machines, enabling thorough testing of the deployment in a controlled environment before going live.
- Deployment Challenges:
    - The search for a reliable deployment platform was extensive, considering AWS, Digital Ocean, and GitHub Pages.
    - Deployment on AIISH server presented numerous obstacles, particularly with Docker container builds due to:
        - Conflicts between Docker's embedded DNS and the server's DNS settings.
        - yarn install operations timing out, necessitating a deep dive into network and configuration issues.
        - Port conflict resolution due to pre-existing server deployments.
- Security and Bug Fixes:
    - Addressed critical bugs and security flaws that could potentially expose sensitive routes to unauthorized users.

DEMO:

LINK FOR THE DEMO OF THE PROJECT here

IMPROVISATIONS & FUTURE SCOPE

- Automated Test ID Generation:
    - Transition from manual Test_ID input to an automated system that assigns unique IDs based on the count of tests previously generated in the database, enhancing efficiency and reducing human error.
- Enhanced search functionality
    - Introduce a robust search feature to query patient records by ID or name, which will become crucial as the application scales and the volume of data increases.
- Security Protocol Upgradation:
    - Upgrade the website from HTTP to HTTPS to ensure secure data transmission, coupled with obtaining SSL certificates for the continuous secure operation of the application.
- Ongoing Security Assessments:
    - Establish a routine for identifying and mitigating security vulnerabilities, ensuring the application maintains the highest standards of security as new threats emerge.

ACKNOWLEDGEMENT

We extend our heartfelt thanks to everyone who played a part in bringing the stuttering detection project to life. Special appreciation goes to our project supervisor, Dr. Anil Kumar Vuppala, whose guidance was the keystone of our project's success. His invaluable feedback and steadfast support were the compass that guided us through our journey.

A profound gratitude is also due to our mentors, Vamsi and Koushik, for their pivotal roles. Their expertise and timely advice were instrumental in navigating the complexities we encountered along the way.

This project has been a significant learning experience, and the knowledge we've gained is as much a product of their contributions as our efforts.

CONCLUSION

In conclusion, we have successfully designed and developed a primary web application for the Speech System – Stuttering Detection project. Throughout the project, we faced several challenges as we had limited prior experience with the technologies used. However, with the guidance of our project supervisor, mentors, and team members, we were able to overcome these challenges and complete the project successfully.

THANK YOU