**JSS MAHAVIDYAPEETHA MYSORE**

**JSS ACADEMY OF TECHNICAL EDUCATION, BENGALURU-560 0056**

**DEPARTMENT OF INFORMATION SCIENCE ENGINEERING**

A logo of a college

Description automatically generated

**Report on**

**AI Based Travel Recommendation System**

For the Course

**Software Engineering & Project Management**

Submitted By

|  |  |
| --- | --- |
| **Name** | **USN** |
| Sheethal Bhushan | 1JS22CI066 |
| Sonika Tekumani | 1JS22CI075 |
| Sowrabh B Shetty | 1JS22CI076 |
| Sujal H R | 1JS22CI078 |

**5th Semester AIML-B**

**Academic Year:2024-25**

|  |  |
| --- | --- |
| **Marks Scored :** |  |
| **Faculty In charge Signature :** |  |

## ABSTRACT

This report provides a comprehensive overview of the methodologies, data flows, and models employed in the development of an AI-based travel recommendation system. Utilizing Agile methodology, the project emphasizes iterative development, collaboration, and flexibility, ensuring that the final product meets user needs and adapts to changing requirements. The system integrates diverse data sources, including user preferences, historical booking data, and real-time travel APIs, to deliver highly personalized travel itineraries.

Machine learning models form the core of the system, enabling accurate recommendations by leveraging techniques such as collaborative filtering, content-based filtering, and hybrid approaches. The project incorporates budget optimization, seasonal trend analysis, and real-time updates, ensuring an enhanced user experience and practical travel planning solutions.

Additionally, the report highlights the importance of a well-structured software architecture, featuring a modular design across data collection, processing, and application layers. Emphasis is placed on ensuring system scalability, responsiveness, and adaptability to user feedback. The document serves as a valuable resource for understanding the integration of software engineering princi

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **Sl.no** | **Figure Description** | **Page. No** |
| **1** | **Introduction** | **1** |
| **2** | **Requirements Specification** | **2** |
| **3** | **Key components** | **3** |
| **4** | **Use Cases** | **4** |
| **5** | **System Architecture** | **5** |
| **6** | **Limitations** | **6** |
| **7** | **System Architecture Overview** | **7** |
| **8** | **Working** | **8** |
| **9** | **Data Flow Diagram** | **9** |
| **10** | **Waterflow Model** | **10** |
| **11** | **Activity Diagram** | **11** |
| **12** | **Conclusion** | **12** |

## AI Accessibility Reader

The **AI Accessibility Reader** is an innovative web-based tool designed to enhance digital accessibility for visually impaired individuals by converting text files into speech. The system is built using React for the frontend, Node.js for the backend, and Coqui TTS, an open-source text-to-speech engine, for text-to-speech conversion. By providing a simple and interactive user interface, the application allows users to upload text files and control audio playback through features like Play, Pause, and Stop.

This project is aligned with **Sustainable Development Goal 10**: Reduced Inequalities, as it strives to bridge the accessibility gap and promote inclusivity. Unlike traditional online text-to-speech services, the AI Accessibility Reader functions offline, ensuring privacy, cost-effectiveness, and usability in areas with limited internet access.

Future improvements may include supporting more file formats, enhancing speech quality, and providing multilingual text-to-speech capabilities. Overall, the AI Accessibility Reader presents a practical and accessible solution for making digital content available to everyone

**SDG Mapping: Reduced Inequalities (SDG 10)**

SDG 10 aims to reduce inequality within and among countries. It focuses on promoting social, economic, and political inclusion for all individuals, regardless of age, sex, disability, race, ethnicity, origin, religion, or economic or other status. This project aligns with Sustainable Development Goal 10: Reduced Inequalities. It aims to bridge the gap in digital accessibility for visually impaired individuals by providing an easy-to-use text-to-speech tool, promoting inclusive access to information and technology.

The AI Accessibility Reader addresses a critical aspect of inequality by making digital content more accessible to visually impaired individuals. Many online resources, documents, and educational materials are designed primarily for visual consumption, which creates a barrier for those who rely on alternative formats like audio. By providing a free, user-friendly text-to-speech tool that can work offline, this project ensures that visually impaired users can independently access and understand written information without relying on costly or proprietary software.

Furthermore, promoting accessibility through this project helps to create a more inclusive digital environment. The tool is developed using open-source technology (Coqui TTS), making it a cost-effective solution that can be adapted and improved by the broader community. This aligns with the objective of promoting social, economic, and technological inclusion for all, as highlighted by SDG 10.

Additionally, the project’s architecture is designed to be scalable and adaptable, meaning it can be expanded to support multilingual text-to-speech conversion, which would further enhance inclusivity for people from various linguistic backgrounds. By providing a simple, effective, and accessible tool, this project contributes towards achieving the broader vision of reduced inequalities in the digital space.

**Role of Automation and Human Control:**

The AI Accessibility Reader is designed with a balanced approach, incorporating automation to simplify processes and human control to maintain user flexibility and accessibility.

**Automation:**Automation in this project focuses on making the text-to-speech conversion process seamless and efficient. It involves:

1. **Text Processing and Conversion:** Once a user uploads a text file, the backend system automatically processes the text using the Coqui TTS engine. This process involves converting textual content into speech without requiring any manual intervention from the user.
2. **Automated Script Execution:** When a text file is uploaded, a Python script utilizing Coqui TTS is triggered automatically. The script processes the text, generates speech, and saves it as an audio file ready for playback.
3. **Playback Handling:** Once the audio file is generated, the user can immediately play the file, demonstrating the efficiency of automation in streamlining the process. Automated file handling ensures smooth communication between the frontend (React) and backend (Node.js) systems.

**Human Control:**

Despite the high degree of automation, the project provides essential human control features to ensure usability and personalization.

1. **User-Driven Input**: Users manually upload text files through an easy-to-use web interface.This ensures that users have complete control over what content they want to convert to speech.
2. **Playback Controls:**After the speech file is generated, users can interact with it through Play, Pause, and Stop buttons.These controls allow users to start, pause, or stop the audio at their convenience, making the tool highly user-friendly.
3. **Customization Opportunities:**The system can be extended to allow users to select different voices, adjust speech speed, or change playback settings, enhancing user control.Users have the freedom to choose when to listen to the audio output and how they want to interact with it.

**Abstract**

The AI Accessibility Reader is a web-based tool designed to provide accessible reading solutions for visually impaired individuals. The application converts textual content to speech using an open-source text-to-speech engine (Coqui TTS). A simple and intuitive interface built with React allows users to upload text files and control playback through interactive buttons. This project supports SDG 10: Reduced Inequalities by promoting accessibility and inclusivity. The proposed system integrates Node.js for backend processing and communication between the frontend and the Python script, ensuring seamless functionality. The application’s effectiveness lies in its simplicity, usability, and potential for further enhancement with multilingual support and compatibility with various file formats.

**Introduction**

Accessibility to digital content remains a significant challenge for visually impaired individuals, especially in a world where most information is disseminated through text-based digital platforms. Although technological advancements have made digital resources more accessible, there is still a considerable gap in providing affordable, user-friendly, and customizable tools that can convert written text into comprehensible speech.

Existing text-to-speech systems are often costly, require internet connectivity, or are integrated with proprietary platforms, limiting their accessibility for economically disadvantaged individuals. Moreover, many tools lack the personalization and control features required for a seamless user experience.

The AI Accessibility Reader aims to address these challenges by providing an open-source, offline-compatible tool that converts text files into speech using the Coqui TTS engine. By enabling users to upload text files and control playback through a straightforward interface, the project ensures ease of use, flexibility, and enhanced accessibility.

This project also emphasizes the importance of human control in automation. While the conversion process is automated for efficiency, users retain control over audio playback through features like Play, Pause, and Stop, promoting a balanced interaction between automation and user preferences.

Additionally, the AI Accessibility Reader aligns with Sustainable Development Goal 10: Reduced Inequalities, as it strives to provide a digital tool that can be freely used by individuals regardless of their financial status or access to high-end proprietary software. It promotes inclusivity by offering a solution that caters specifically to the needs of visually impaired users, helping them access digital information with ease.

The AI Accessibility Reader is a step towards building a more inclusive digital environment where everyone, regardless of their abilities, can have equitable access to information.

**Literature Survey**

**Research Question:** How can an AI-based text-to-speech tool be developed to provide a free, user-friendly, and accessible solution for visually impaired individuals, while promoting inclusivity and aligning with Sustainable Development Goal 10: Reduced Inequalities?

**Relevant Literature:** Various text-to-speech (TTS) systems have been developed to convert text into speech, catering to a wide range of applications. The following are notable TTS systems commonly used:

1. **Google Text-to-Speech:** A widely-used tool integrated into mobile devices and web services.Provides high-quality, natural-sounding speech with multilingual support.However, it is a paid service and requires internet access, limiting its accessibility for economically disadvantaged individuals.
2. **Microsoft Azure Cognitive Services (Speech Service**): Provides advanced neural voices with highly customizable voice generation capabilities. Offers features like speech synthesis, recognition, and translation. Requires internet connectivity and is a paid service, making it inaccessible to users who cannot afford such services.
3. **Amazon Polly:** Converts text into lifelike speech using deep learning models. Supports multiple languages and provides various voice options. Although effective, it remains a paid service dependent on internet connectivity.
4. **Coqui TTS (Used in this Project):** An open-source, free, offline-compatible text-to-speech engine. Supports multilingual models and various neural network architectures. Provides high-quality voice synthesis without requiring internet access, making it ideal for accessibility-focused applications. Chosen for this project due to its flexibility, ease of integration, and zero-cost availability.



While many commercial TTS systems like Google Text-to-Speech, Microsoft Azure, and Amazon Polly provide high-quality voice synthesis, they all require internet access and come with subscription costs. This restricts their accessibility to those who can afford them, which is not ideal for creating an inclusive tool for visually impaired individuals.

On the other hand, Coqui TTS stands out as an accessible alternative due to its offline functionality and open-source nature. It is particularly suitable for creating a tool aimed at promoting digital inclusivity, aligning with the objectives of SDG 10: Reduced Inequalities.

While commercial solutions provide superior voice quality and enhanced features, their dependency on internet connectivity and cost limit their accessibility. Coqui TTS offers a practical alternative that balances quality and accessibility, making it the most suitable choice for the AI Accessibility Reader project.

**Methodology**

**System Architecture:**

The AI Accessibility Reader consists of three main components:

1. **Frontend (User Interface):**
   * + Built with React.
     + Provides a clean interface where users can upload text files and control playback.
     + Uses Axios to communicate with the backend via HTTP requests.
2. **Backend Layer (Server) :**
   * + Built with Node.js and Express.
     + Handles file uploads and triggers the text-to-speech processing by invoking a Python script
     + Manages the communication between the frontend and the text-to-speech engine.
3. **Text-to-Speech Engine (Python):**
   * + Uses the pyttsx3 library for offline text-to-speech conversion.
     + Processes text files and generates audio playback.
     + Allows basic playback control such as Start, Stop, and Pause.

**References**

1. Python Software Foundation. (n.d.). Pyttsx3 Documentation. Retrieved from <https://pypi.org/project/pyttsx3>
2. Node.js Foundation. (n.d.). Node.js Documentation. Retrieved from <https://nodejs.org/en/docs/>
3. Meta Platforms, Inc. (n.d.). React Documentation. Retrieved from <https://react.dev/>
4. United Nations. (n.d.). Sustainable Development Goal 10. Retrieved from <https://sdgs.un.org/goals/goal10>
5. Google Text-to-Speech. (n.d.). Retrieved from <https://cloud.google.com/text-to-speech>
6. Microsoft Azure Speech Service. (n.d.). Retrieved from <https://azure.microsoft.com/en-us/services/cognitive-services/speech-services/>
7. Amazon Polly. (n.d.). Retrieved from <https://aws.amazon.com/polly/>
8. Coqui TTS. (n.d.). Retrieved from <https://github.com/coqui-ai/TTS>