**SOFTWARE REQUIREMENTS SPECIFICATION (SRS)**

**Express AI**

**Emotionally Intelligent TTS Extension**

A blue and black logo

AI-generated content may be incorrect.Presented by

**Team Apex**

**Bachelor of Computer Application**

Under the guidance of an assistant professor

**Dr. Rashmi Shekhar**

**Amity Institute of Information Technology**

**Amity University**

**Patna**

**2024-27**

**Our Team Members-**

Kumar Sahil

Bittu Kumar Ravi Ronak

**Table of Contents**

[**01. Introduction** 3](#_Toc198043986)

[**02. Overall Description** 4](#_Toc198043987)

[**03. Specific Requirements** 5](#_Toc198043988)

[**04. External Interface Requirements** 6](#_Toc198043989)

[**05. System Features** 7](#_Toc198043990)

[**06. Non-Functional Requirements** 8](#_Toc198043991)

[**07. Future Enhancements** 9](#_Toc198043992)

[**08. Budget Estimation** 10](#_Toc198043993)

[**09. Appendices** 11](#_Toc198043994)

# **01. Introduction**

**1.1 Purpose**

The purpose of this document is to define and outline the detailed functional and non-functional software requirements for the Express AI Chrome extension. This extension is designed to enhance the digital reading experience of visually impaired individuals by transforming web content into speech and, in future versions, embedding emotional tone using AI.

**1.2 Scope**

* Extract text from web pages.
* Convert extracted text into speech.
* Future versions: analyze emotional tone and modulate voice accordingly.
* Assist visually impaired users by offering dynamic and emotionally intelligent auditory feedback.
* Delivered as a Chrome browser extension with scope for expansion to Firefox and Edge.

**1.3 Intended Audience**

* Visually impaired users
* Educators and researchers in AI and accessibility
* Developers and contributors
* Chrome extension reviewers

**1.4 Definitions, Acronyms, and Abbreviations**

* **TTS** – Text to Speech
* **NLP** – Natural Language Processing
* **UI** – User Interface
* **API** – Application Programming Interface
* **DOM** – Document Object Model

# **02. Overall Description**

**2.1 Product Perspective**

Express AI is a standalone browser extension. It depends on the browser’s DOM for content extraction and integrates TTS APIs like Google Cloud TTS. Future integration includes AI-based sentiment analysis.

**2.2 Product Functions**

* **Text Extraction**: Pull content from web pages.
* **Text-to-Speech**: Read out content using speech synthesis.
* **Voice Modulation (Future)**: Adjust pitch, tone, and tempo based on emotional context.
* **Emotion Analysis (Future)**: Identify emotions in the text using AI/ML.
* **UI Interface**: Control speech modes, voices, and settings.

**2.3 User Characteristics**

* Basic computer knowledge
* Visually impaired individuals needing auditory assistance
* Users familiar with screen readers or accessibility tools

**2.4 Operating Environment**

* OS: Windows, macOS, Linux (via browser)
* Browsers: Chrome (initially), Firefox/Edge (future)
* TTS API: Google Cloud TTS (currently), custom APIs (future)

**2.5 Design and Implementation Constraints**

* Limited by Chrome Extension APIs
* Free-tier APIs (budget constraint)
* Privacy compliance (text data must not be stored)

**2.6 Assumptions and Dependencies**

* Internet required for TTS and future AI model calls
* Requires permissions for DOM access
* Users will allow access to web page content

# **03. Specific Requirements**

**3.1 Functional Requirements**

**FR1: Web Page Text Extraction**

* The system shall extract readable text from the body of any web page.
* It shall ignore script, style, and hidden elements.

**FR2: Text-to-Speech Conversion**

* The system shall convert extracted text into speech using TTS APIs.
* Users can choose from multiple voices (male/female).
* Speech speed and pitch should be customizable.

**FR3: Emotion Analysis (Future)**

* The system shall analyze emotional tone using NLP.
* It must classify emotions such as happy, sad, angry, excited, etc.

**FR4: Voice Modulation (Future)**

* The system shall modulate speech tone to reflect detected emotions.
* Modulation includes changes in volume, pitch, speed, and pauses.

**FR5: UI and Accessibility Settings**

* Users shall be able to:
  + Toggle reading modes (Screen Reader / Emotion Reader)
  + Switch hyperlink reading modes
  + Choose voice type (male/female)
  + Start/Stop reading

**FR6: Sentence-Level Highlighting**

* As each word is spoken, it must be visually highlighted in the text.

# **04. External Interface Requirements**

**4.1 User Interface**

* A popup-based UI with:
  + Toggle buttons
  + Radio buttons for reading and voice modes
  + Blue color theme
  + Semi-transparent logo background
* Accessibility considerations:
  + Keyboard navigation
  + Screen reader support

**4.2 Software Interfaces**

* **TTS API Interface**:
  + Google Cloud TTS (REST API)
  + Optional: Amazon Polly, IBM Watson (future)
* **Sentiment Analysis API (Future)**:
  + Hugging Face / custom-trained models (via REST)

**4.3 Hardware Interfaces**

* None specific; standard computing hardware with audio output.

# **05. System Features**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Description** | **Status** |
| **Basic TTS** | Read aloud webpage text in selected voice | Implemented |
| **Hyperlink Text Reading Modes** | Switch between link text, URL, or both | In Development |
| **Voice Selection** | Choose Male or Female voice | In Development |
| **Screen Reader Mode** | Read in neutral voice | Implemented |
| **Emotional Tone Mode** | Read with emotion | In Development |
| **Emotion Detection** | Detect emotion from sentence | In Development |
| **Voice Modulation** | Modulate tone based on emotion | Planned |
| **Highlight Spoken Word** | Highlight each spoken word | Implemented |
| **Cross-browser Support** | Available on Chrome, planned for Firefox | Planned |

# **06. Non-Functional Requirements**

**6.1 Performance Requirements**

* Real-time speech with minimal lag.
* Must process and start reading within 2 seconds for average-length paragraphs.

**6.2 Security Requirements**

* Text data must not be stored or shared.
* No cookies or tracking without user consent.

**6.3 Portability**

* Should be portable across modern Chromium-based browsers.
* Code modularity for API changes.

**6.4 Maintainability**

* Modular JS architecture with proper documentation.
* Clear version control via GitHub.

**6.5 Usability**

* Designed for users with disabilities.
* Large fonts, clear layout, and keyboard navigation supported.

# **07. Future Enhancements**

**Features Will be added in Future**

* Offline TTS Mode
* Custom Emotion Profiles
* Browser History Integration
* PDF/Web Reader Mode
* Mobile Browser Extension (Chrome Mobile)

# **08. Budget Estimation**

|  |  |
| --- | --- |
| **Component** | **Estimated Cost (INR)** |
| **AI API & Firebase Hosting** | ₹15,000 |
| **Chrome Web Store Listing** | ₹540 (one-time) |
| **Domain Name** | ₹700 |
| **Mail Server** | ₹2,000 |
| **GitHub (Free)** | ₹0 |
| **Total (1st Year)** | **₹18,240** |

# **09. Appendices**

**A. References**

* ChatGPT and Generative AI tools
* Google Cloud TTS Docs
* MDN Web Docs
* YouTube Tutorials on Chrome Extensions
* GitHub – Open-Source Repository

**B. Team Information**

|  |  |  |
| --- | --- | --- |
| **Name** | **Course** | **Enrollment No.** |
| Kumar Sahil | BCA | A45304824162 |
| Bittu Kumar | BCA | A45304824153 |
| Ravi Ronak | BCA | A45304824154 |