Our Connect Sense: A Web-App to Enhance Accessibility and Learning for Physically Challenged Students

Harsh Rajpal - 20BCI0271
Payal Maheshwari - 20BCE2759
Ayush Kanaujiya- 20BCE2748
Bikash Chauhan- 20BCE2769

J component - Review 3

CSE4015 - Human Computer Interaction

Supervisor: Prof. Navamani T M.

School of Computer Science and Engineering (SCOPE)
Vellore Institute of Technology, Vellore

April, 2023

Table of Contents

I. ABSTRACT

Our Connect Sense is a web application designed to facilitate the learning process for physically challenged learners.

The project offers a variety of features, such as playing games, completing tests, and listening to text, while navigating the web application using a search bar or voice commands. This initiative aims to provide a simple and accessible method for physically disabled students to interact with and learn from educational content. These pupils can overcome physical obstacles and enhance their learning experience with the aid of Our Connect Sense.

Keywords: Our Connect Sense, web-app, physically challenged students, learning, accessibility, games, tests, voice commands.

II. INTRODUCTION

The internet has revolutionized the way we learn and share information, making it an essential tool for education and knowledge exchange. However, people with disabilities often encounter significant barriers when attempting to access online resources. For individuals who are blind or visually impaired, screen readers can be prohibitively expensive and challenging to use. Meanwhile, deaf or hard of hearing individuals may not be able to access online videos or lectures without captions or transcripts. To address these challenges, we have developed "Our ConnectSense," a web application designed to enhance accessibility and learning for students with physical disabilities, including those who are deaf.

"Our ConnectSense" is a comprehensive web app that provides a variety of features designed to make online learning more accessible and user-friendly for students with physical disabilities. The app includes games, quizzes, and text that can be read aloud, all of which can be navigated using a search bar or vocal commands. One of the app's most significant features is its ability to convert speech to text. This feature enables impaired users to read along with lectures, videos, and other audio content, ensuring that they do not miss out on any important information.

The app is designed to be easy to use and navigate for students with physical disabilities. Its interface is user-friendly and intuitive, with clear and concise instructions that make it simple to access and use all of its features. The app's voice-command navigability makes it even easier for users to interact with educational content and access all of its features.

Accessibility is a critical component of the app's design. Our ConnectSense was developed with the goal of improving accessibility for physically challenged students, including those who are deaf or visually impaired. The app's developers understand that not all students have access to the same resources, and that cost can be a significant barrier for many. As such, the app was designed to be affordable and accessible to all students, regardless of their financial situation.

The app's object recognition feature is particularly useful for visually impaired students. It enables them to identify objects by capturing an image of the object with their phone's camera. The app then uses a speech synthesizer to transform the identified object into an audio format, providing visually impaired students with an enhanced learning experience.

In addition to providing accessibility features, the app's developers have also focused on creating an engaging and interactive learning experience for students. The app's games and quizzes are designed to be fun and educational, providing students with an enjoyable way to learn and improve their knowledge.

In conclusion, "Our ConnectSense" is a web application designed to improve accessibility and learning for physically challenged students, including those who are deaf or visually impaired. Its features are designed to be affordable and easy to use, making it accessible to all students, regardless of their financial situation. By making online learning more accessible and user-friendly, we hope to empower physically challenged students and help them reach their full potential.

III. LITERATURE SURVEY

Research in the area of object detection has been going on since last decade. C. M. Asad et al [1], designed a model for hands-free human computer interaction and head movements to control mouse motions. It focuses on the categorization of a disabled person's head motions for hands-free computer operation. V. Kalist, A. A. F. Joe and A. Veeramuthu [2], proposed a strategy for classroom communication that transforms the wordless student's sign language into a text message for the deaf to understand and an audio message for the blind to follow. The model supports Bluetooth-enabled voice assistance for device control. L. Ciabattoni, F. Ferracuti, G. Foresi and A. Monteriù [3], formulated a unique method for creating a user interface for commercial Smart Home Systems (SHS) that takes into account the demands of users who are blind or deaf. Using a mobile application, the interface may convert alerts and visual information into audio signals and vice versa. B. F. Smaradottir, S. G. Martinez and J. A. Håland [4], has worked on assistive technology and found that voice guidance and haptic feedback were effective in helping visually impaired users' complete tasks on touchscreens. The methodology involves recruiting visually impaired individuals to participate in the study and using a combination of surveys, observations, and task performance measures to evaluate the usability and effectiveness of different assistive technologies. J. A. P. De Jesus, K. A. V. Gatpolintan.

C. L. Q. Manga, M. R. L. Trono and E. R. Yabut [5], presented a framework for developing mobile and web applications to assist visually impaired individuals. The framework consists of three main components: Image recognition and object detection, Text recognition, and Navigation and guidance. They present results of user testing with visually impaired individuals. The users found the framework to be helpful in improving their independence and mobility, as well as facilitating tasks such as reading and identifying objects. S. Noel [6] proposed an approach that focuses on creating email applications that can translate speech to text and text to speech for users who are blind or visually challenged. The best email format for users who are blind or visually impaired is one that uses speech or audio. The voice commands used by the application are straightforward and user-friendly. The user can check the dictated content in the email once it has been created using text to speech. S. Dobrisek, J. Gros, F. Mihelic and N. Pavesic [7] proposed a database interface in which text data are grouped in text corpora at the information source center. For blind and visually challenged users, the system's spoken language interface is crucial, notably the text-to-speech component. The system's voice control makes it easier to operate with it since it eliminates the need for mechanical interfaces like a mouse or keyboard. The system is made up of three primary modules: the text to voice module, spoken command recognition module, and the conversation module.

K. S. Sri, C. Mounika and K. Yamini [8] suggested an audiobook with a Python-made web user interface. There are a total of 4 modules, including image to text and voice, PDF to speech, text into speech, and text into voice. Reading a book, listening to a recorded text, spelling a word, and penning a sentence are among the assignments.
The text-to-speech module will read the entered text aloud, assisting those who are deaf in accurately spelling the word. The voice to text module aids in turning spoken words into written ones. The deaf can benefit from it and readers can get better at reading. The Picture to Speech module will convert both the text format and the image's text into voice. The full PDF file will be converted into speech using PDF to Speech.

IV. Proposed System

Overview

The system for the blind focuses on object recognition, with object identification as its primary function. A speech synthesiser is then used to convert the identified objects into an audible format. The system is capable of capturing conversations and converting them to text so that deaf individuals can comprehend them.

The final app integrates these two functions on a single platform. The programme is also voice-command navigable, making it simpler for physically challenged students to interact with educational content and navigate the app. By making online learning more accessible and user-friendly, we aim to empower physically challenged students and assist them in reaching their full potential.

A. Flow control:

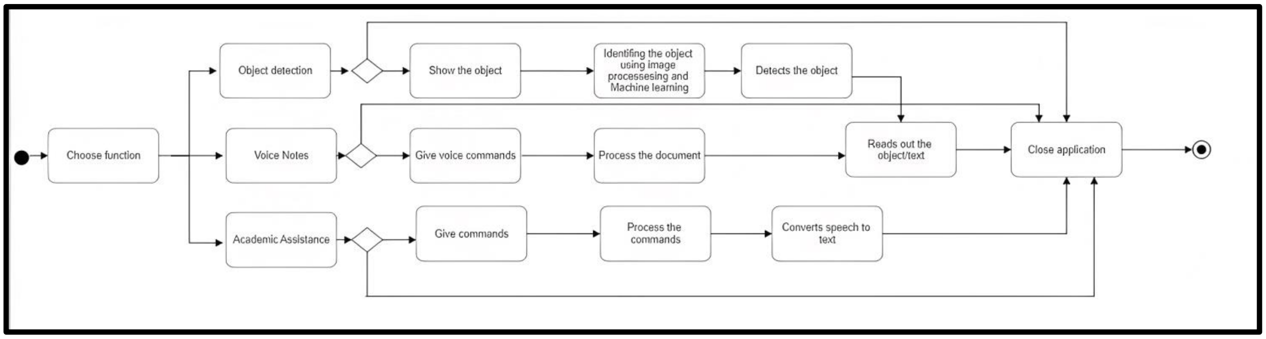


Fig. 1 Flow Control of overall application

The above flow diagram depicts the flow of the system and what are the possible pathways it can go. After sign-in, the user can select one of three features and provide the necessary inputs. The modules will process the input and perform the desired tasks like object detection, text to speech, voice notes, etc.

B. Architecture Diagram

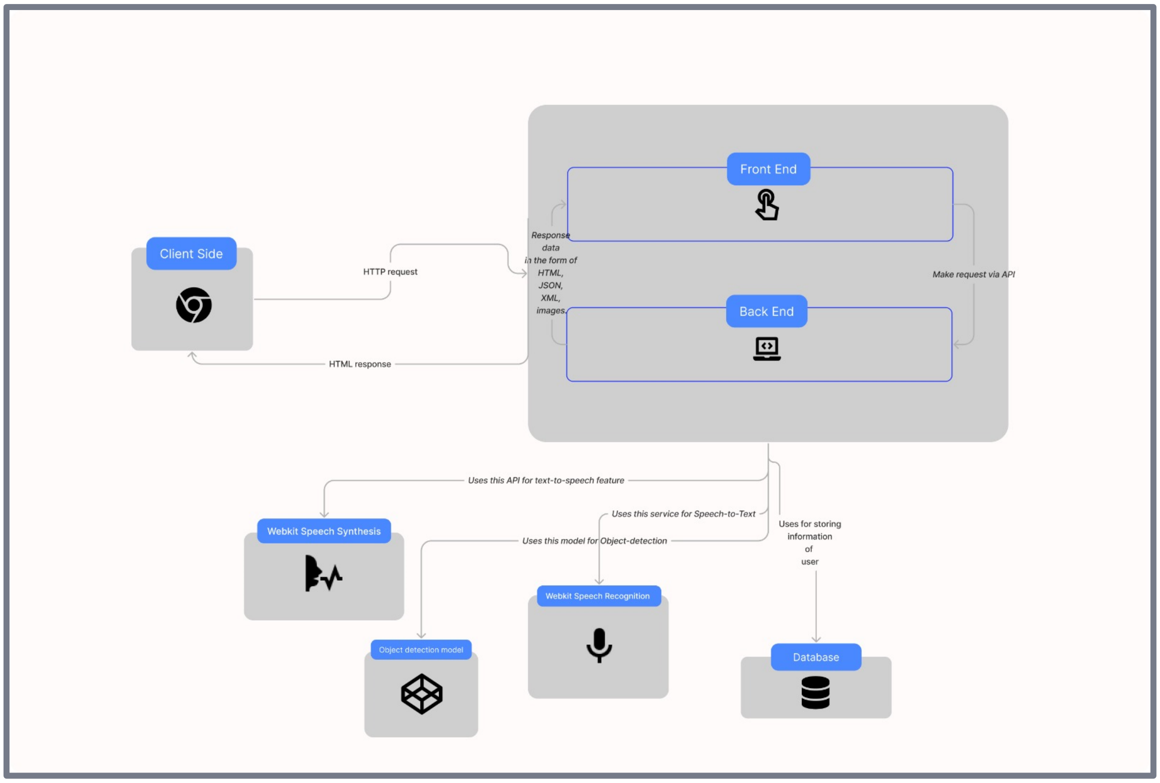

Client Side
HTTP request
HTML response
Response data in the form of HTML, JSON, XML, images.
Front End
Back End
Make request via API
Uses this API for text-to-speech feature
Uses this service for Speech-to-Text
Uses this model for Object-detection
Uses for storing information of user
Webkit Speech Synthesis
Object detection model
Webkit Speech Recognition
Database

Fig 2: Architecture diagram

The web application is based on MERN stack. We are using the react library to develop the front end and tailwind css to design the interface. The data of the user is being stored in the MongoDB database. Different API's like tensorflow-yolo model, speech synthesis and web speech are used to achieve the desired features. The overall flow of the request and response has been depicted in the architecture diagram.

C. Activity Diagram

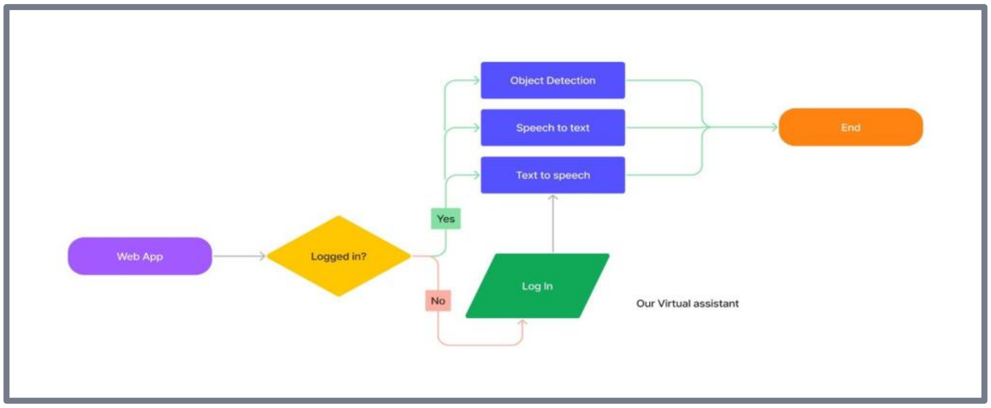


Fig 3: Activity diagram

D. Use Case Diagram

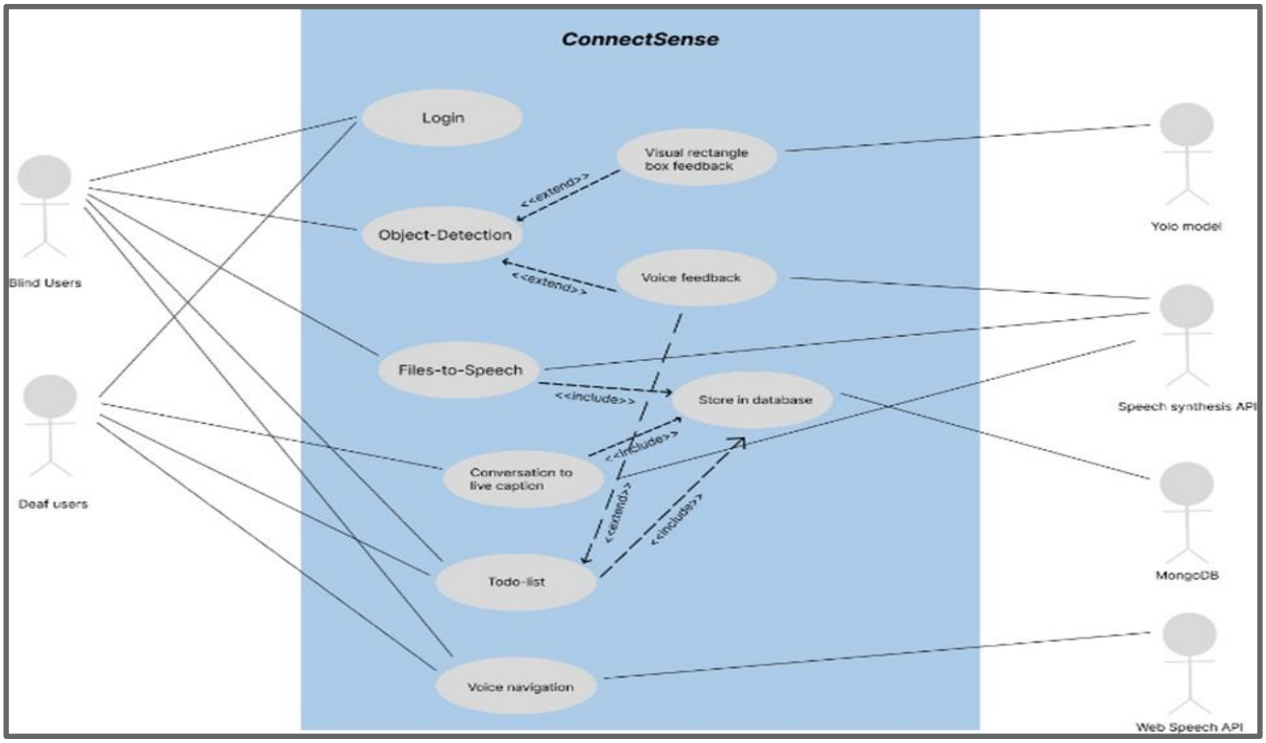


Fig 4: Use Case Diagram

The target audience of this project are blind and deaf people. They are the actors shown in the above diagram who are interacting with the system. Different technology and service providers also are the actors that interact with the system. Object detection extends the feature of visual and voice feedback. Whereas todo-list ind file to speech feature include the capability to store the data of the user in the database.

E. Class Diagram

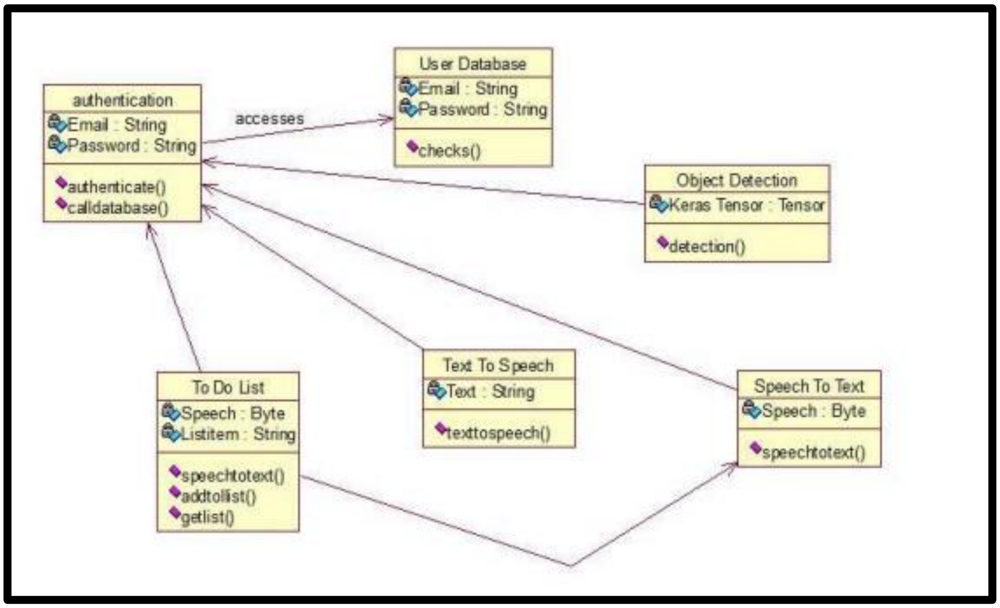


Fig 5: Class Diagram

F. Object Diagram:

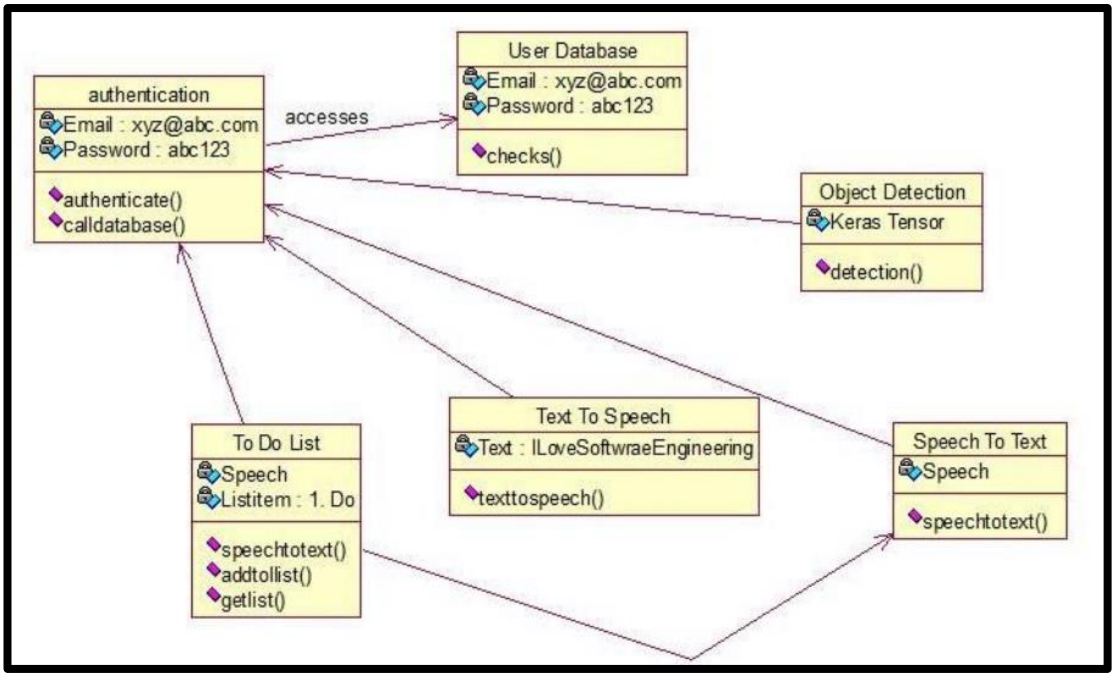


Fig 6: Object Diagram

Motivation:

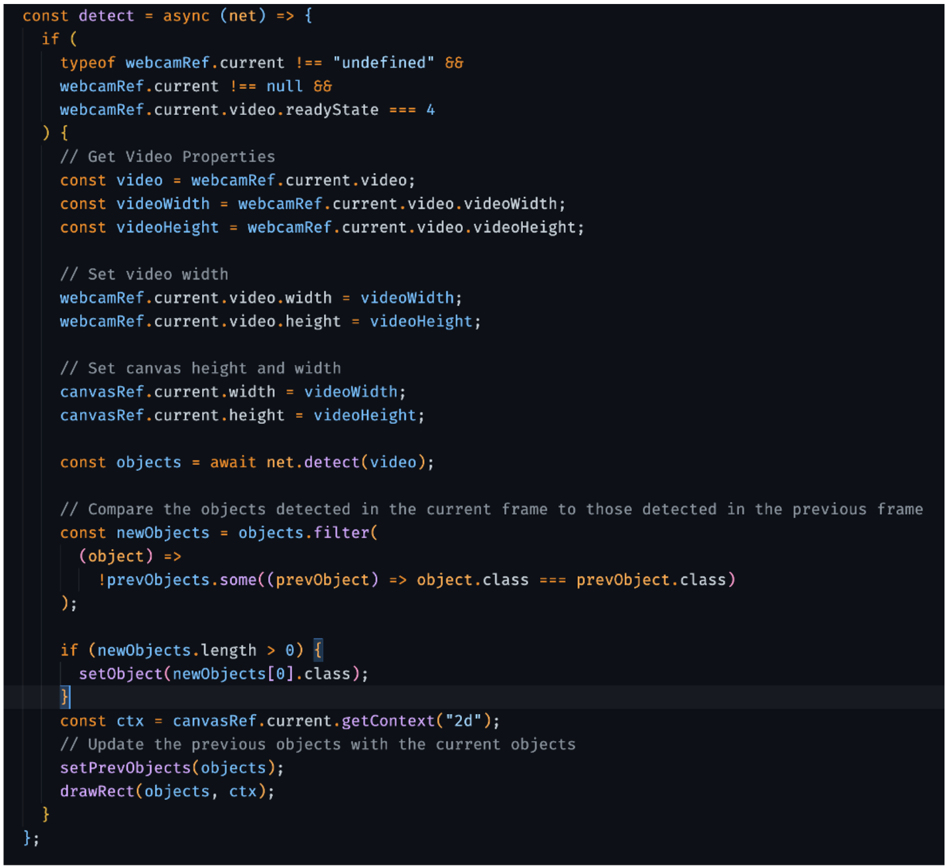
In the developing countries, there is still a lack of facility to provide basic needs to the differently abled children. There is no concept of special education for these children. We want to provide these children a tool that is freely available and easy to access that can help them in performing some of their basic day to day activities. They don’t need to get some expensive technology or advanced skill to manage and run the application. With our connectSense, the children can achieve fundamental right of education and an assistance that will always be there to assist them from detecting the object to reading/ listening books and having a live caption feature for visually impaired children as well.

The main feature that separates us from other similar products are the features for both blind and deaf children. It’s not just for only one group of people. Anyone can use it and it is inexpensive and easy to use with less resource requirement. The system can later be integrated into any kind of device with camera and microphone and be used in different scenarios. There are alot of future scope for this project which are discussed later.

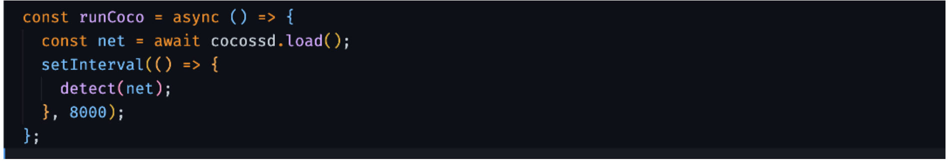
V. IMPLEMENTATION & GOAL ACHIEVED

ConnectSense, a web-based app is designed to empower visually disabled and hearing-impaired users. It includes voice navigation, text-to-speech, speech-to-text, voice-based to-do lists, and object detection features to meet the needs of these users. Voice navigation allows users to navigate the app using voice commands, text-to-speech converts text into spoken words, speech-to-text allows users to input information using their voice, and the voice-based to-do list allows users to manage their tasks using voice commands. Object detection helps visually impaired users identify objects in their environment. The app is accessible, intuitive, and designed to empower users with disabilities.

1. **Object-detection:**



```
const detect = async (net) => {
  if (
    typeof webcamRef.current !== "undefined" &&
    webcamRef.current !== null &&
    webcamRef.current.video.readyState === 4
  ) {
    // Get Video Properties
    const video = webcamRef.current.video;
    const videoWidth = webcamRef.current.video.videoWidth;
    const videoHeight = webcamRef.current.video.videoHeight;

    // Set video width
    webcamRef.current.video.width = videoWidth;
    webcamRef.current.video.height = videoHeight;

    // Set canvas height and width
    canvasRef.current.width = videoWidth;
    canvasRef.current.height = videoHeight;

    const objects = await net.detect(video);

    // Compare the objects detected in the current frame to those detected in the previous frame
    const newObjects = objects.filter(
      (object) =>
        !prevObjects.some((prevObject) => object.class === prevObject.class)
    );

    if (newObjects.length > 0) {
      setObject(newObjects[0].class);
    }
    const ctx = canvasRef.current.getContext("2d");
    // Update the previous objects with the current objects
    setPrevObjects(objects);
    drawRect(objects, ctx);
  }
};
```

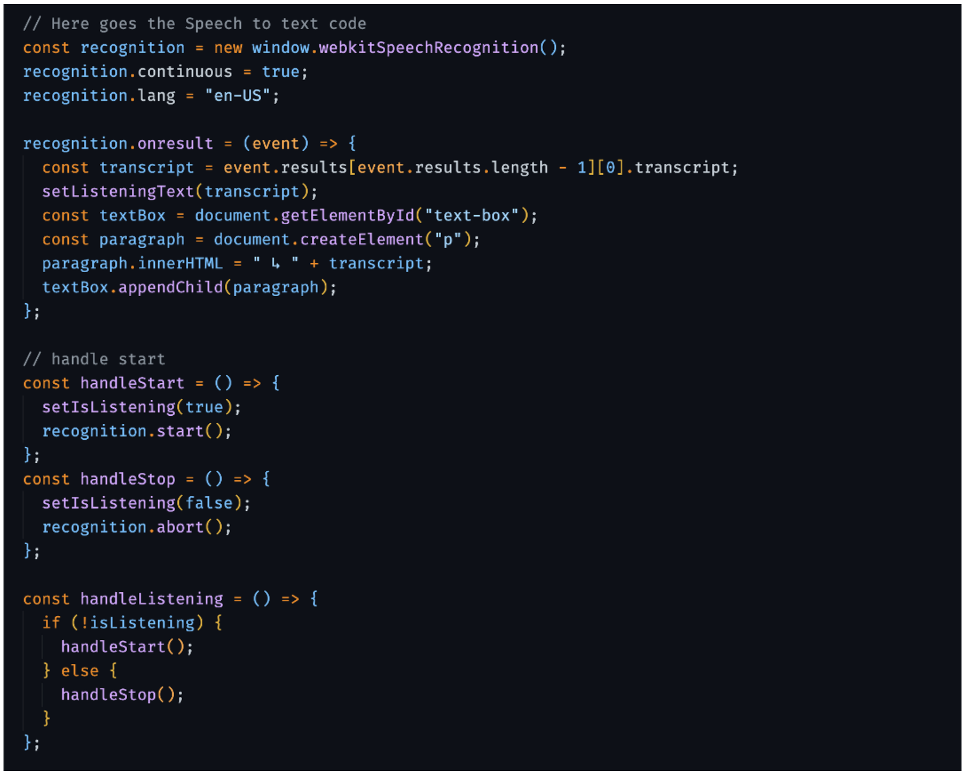


```
const runCoco = async () => {
  const net = await cocossd.load();
  setInterval(() => {
    detect(net);
  }, 8000);
};
```

Description:

The code uses TensorFlow's COCO-SSD model to perform real-time object detection from a webcam feed. The **runCoco()** function loads the model, and the **detect()** function detects the objects from the video frames captured by the webcam. The **drawRect()** function draws a rectangle around the detected object and speaks out the object's name using Text-to-Speech. The object state variable updates to the name of the detected object. The **prevObjects** state variable is used to compare the objects detected in the current frame to those detected in the previous frame. The component uses the Webcam component from the **react-webcam** package to capture the video feed and displays the object's name in the UI.

2. Speech-to-Text:

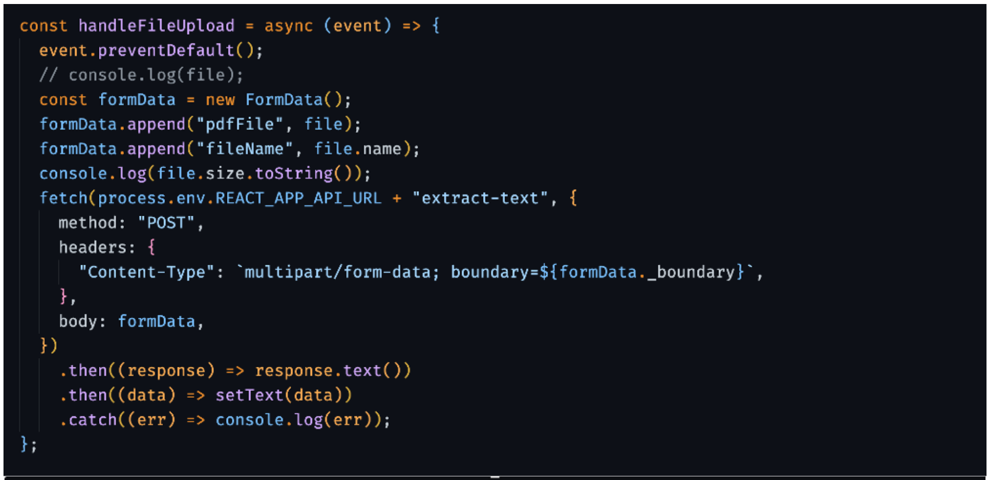


```
// Here goes the Speech to text code
const recognition = new window.webkitSpeechRecognition();
recognition.continuous = true;
recognition.lang = "en-US";

recognition.onresult = (event) => {
  const transcript = event.results[event.results.length - 1][0].transcript;
  setListeningText(transcript);
  const textBox = document.getElementById("text-box");
  const paragraph = document.createElement("p");
  paragraph.innerHTML = " ↳ " + transcript;
  textBox.appendChild(paragraph);
};

// handle start
const handleStart = () => {
  setIsListening(true);
  recognition.start();
};
const handleStop = () => {
  setIsListening(false);
  recognition.abort();
};

const handleListening = () => {
  if (!isListening) {
    handleStart();
  } else {
    handleStop();
  }
};
```

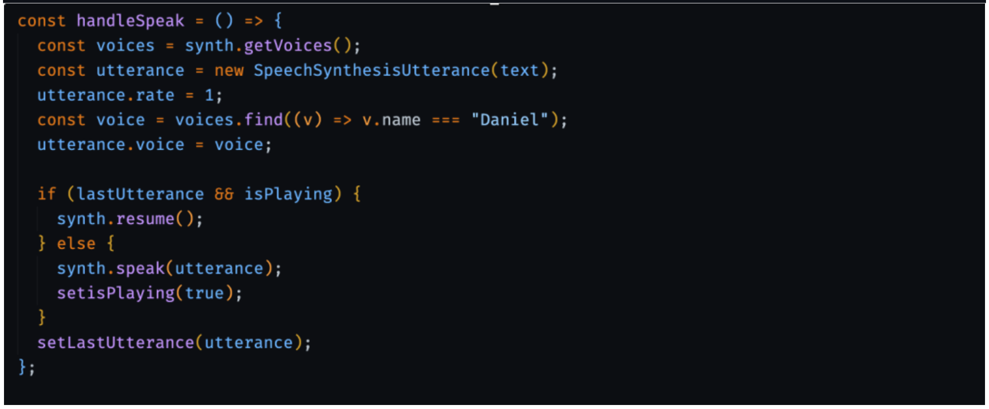
Description:

The component enables speech recognition in the user's web browser using the "webkitSpeechRecognition" object, which continuously listens for spoken English language. It captures spoken text using the "onresult" event handler and displays it in a text box using "setListeningText" and "appendChild" methods. The component has three event handlers to start, stop, and toggle speech recognition, and renders a UI with a microphone button, a text box, and a reset button. The microphone button changes its appearance based on the current "isListening" state. The UI is wrapped within the "FeatureTemplate" component.

3. Text-to-Speech:



```
const handleFileUpload = async (event) => {
  event.preventDefault();
  // console.log(file);
  const formData = new FormData();
  formData.append("pdfFile", file);
  formData.append("fileName", file.name);
  console.log(file.size.toString());
  fetch(process.env.REACT_APP_API_URL + "extract-text", {
    method: "POST",
    headers: {
      "Content-Type": `multipart/form-data; boundary=${formData._boundary}`,
    },
    body: formData,
  })
    .then((response) => response.text())
    .then((data) => setText(data))
    .catch((err) => console.log(err));
};
```

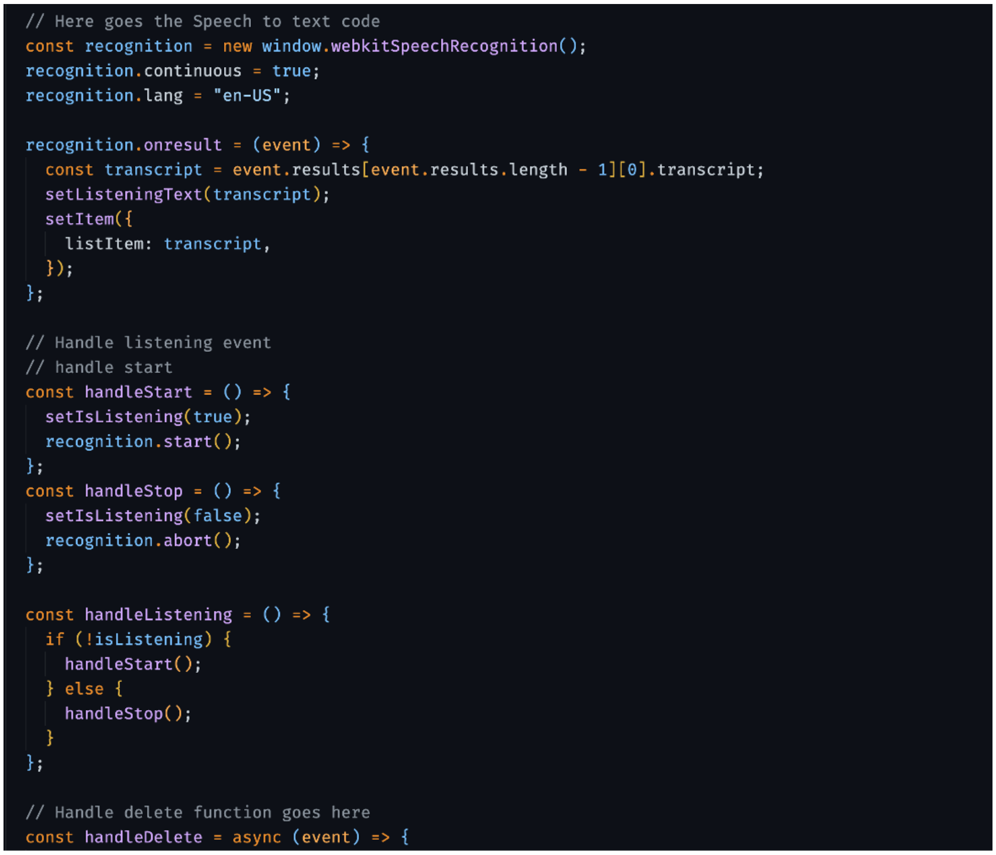


```
const handleSpeak = () => {
  const voices = synth.getVoices();
  const utterance = new SpeechSynthesisUtterance(text);
  utterance.rate = 1;
  const voice = voices.find((v) => v.name === "Daniel");
  utterance.voice = voice;

  if (lastUtterance && isPlaying) {
    synth.resume();
  } else {
    synth.speak(utterance);
    setisPlaying(true);
  }
  setLastUtterance(utterance);
};
```

Description:

This is a React functional component that enables text-to-speech functionality. It allows the user to upload a PDF file, extract the text from it, and use the Web Speech API to have the text read aloud. The component uses the useState and useEffect hooks to manage the state of the component, including variables such as file, lastUtterance, text, isPlaying, and synth. There are several functions defined in the component, such as initSynth, handleSpeak, handleStop, handlePlay, handleFileChange, handleFileUpload, and handleReset, which allow the user to control the speech synthesis object and upload files. The component renders a form for file upload, displays the extracted text, and three buttons to play, pause, and stop the utterance.

4. To-do list (based on speech input)

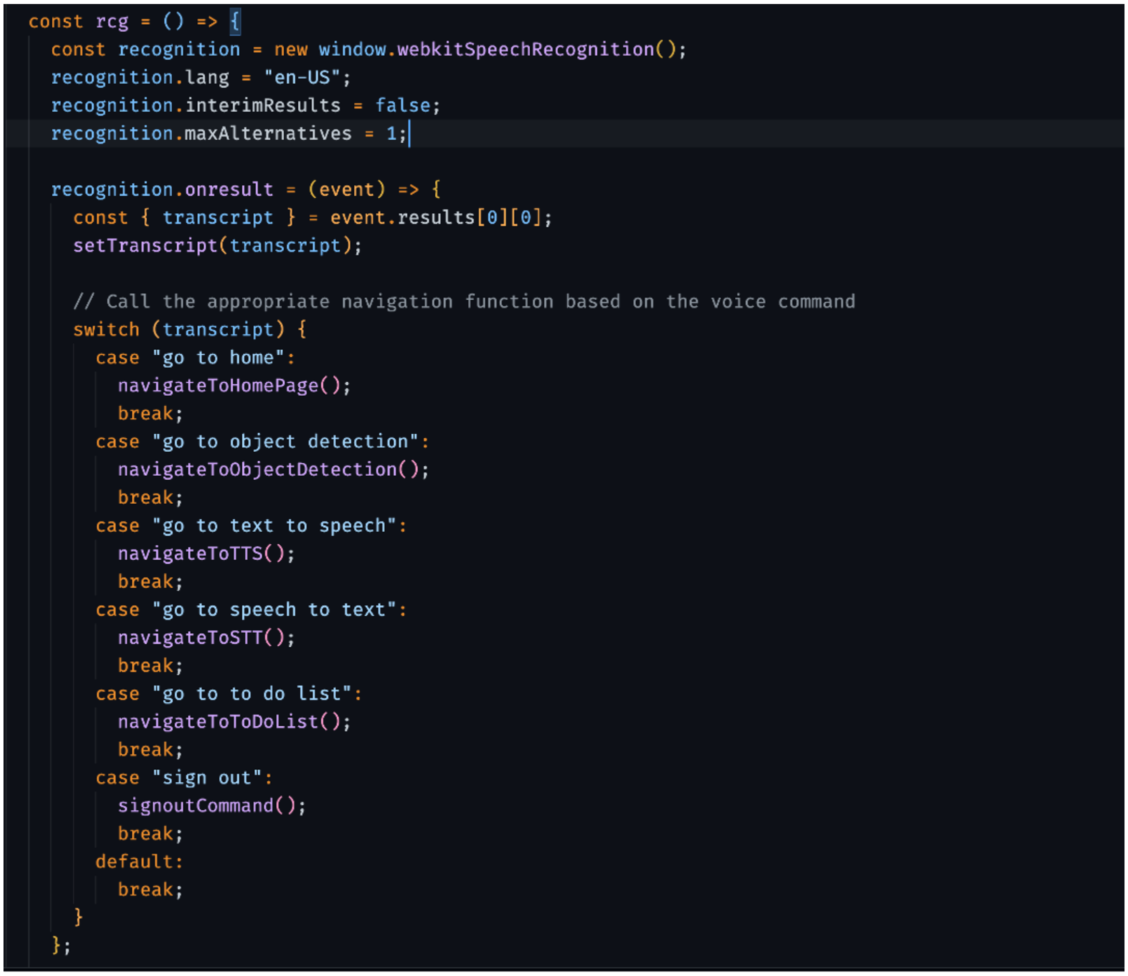


```
// Here goes the Speech to text code
const recognition = new window.webkitSpeechRecognition();
recognition.continuous = true;
recognition.lang = "en-US";

recognition.onresult = (event) => {
  const transcript = event.results[event.results.length - 1][0].transcript;
  setListeningText(transcript);
  setItem({
    listItem: transcript,
  });
};

// Handle listening event
// handle start
const handleStart = () => {
  setIsListening(true);
  recognition.start();
};
const handleStop = () => {
  setIsListening(false);
  recognition.abort();
};

const handleListening = () => {
  if (!isListening) {
    handleStart();
  } else {
    handleStop();
  }
};

// Handle delete function goes here
const handleDelete = async (event) => {
```

Description:

The given React component is for a to-do list app with functionalities like adding and deleting items from the list and a speech-to-text feature. The component has a state that manages the list items fetched from the server using the populateItems function. The **handleAddToList()** function is called to add items to the list, and the **handleDelete()** function is called to delete items. The component also has a text-to-speech feature to listen to the items on the list, and the **handleSpeak()** and **handleStopSpeaking()** functions control this feature. The speech-to-text feature is activated by the handleListening function.

5. Voice based navigation between pages:

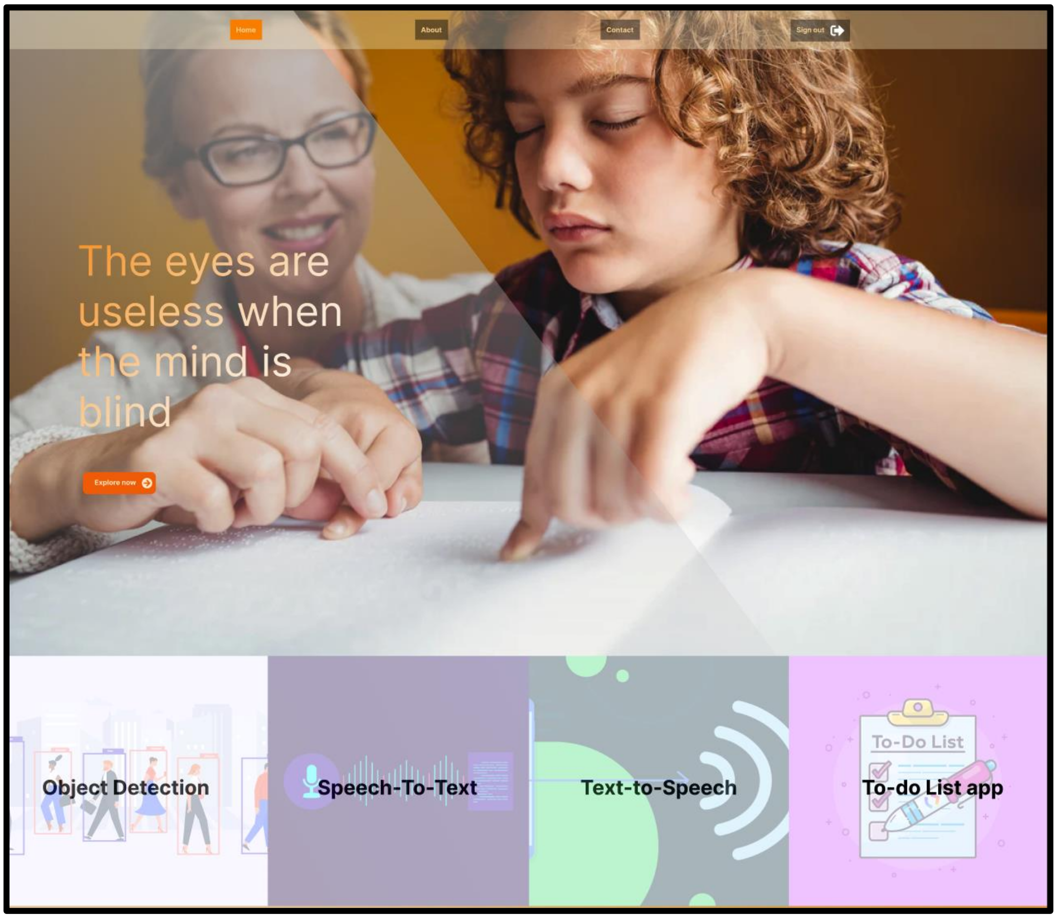


```
const rcg = () => {
  const recognition = new window.webkitSpeechRecognition();
  recognition.lang = "en-US";
  recognition.interimResults = false;
  recognition.maxAlternatives = 1;

  recognition.onresult = (event) => {
    const { transcript } = event.results[0][0];
    setTranscript(transcript);

    // Call the appropriate navigation function based on the voice command
    switch (transcript) {
      case "go to home":
        navigateToHomePage();
        break;
      case "go to object detection":
        navigateToObjectDetection();
        break;
      case "go to text to speech":
        navigateToTTS();
        break;
      case "go to speech to text":
        navigateToSTT();
        break;
      case "go to to do list":
        navigateToToDoList();
        break;
      case "sign out":
        signoutCommand();
        break;
      default:
        break;
    }
  };
```

Description:

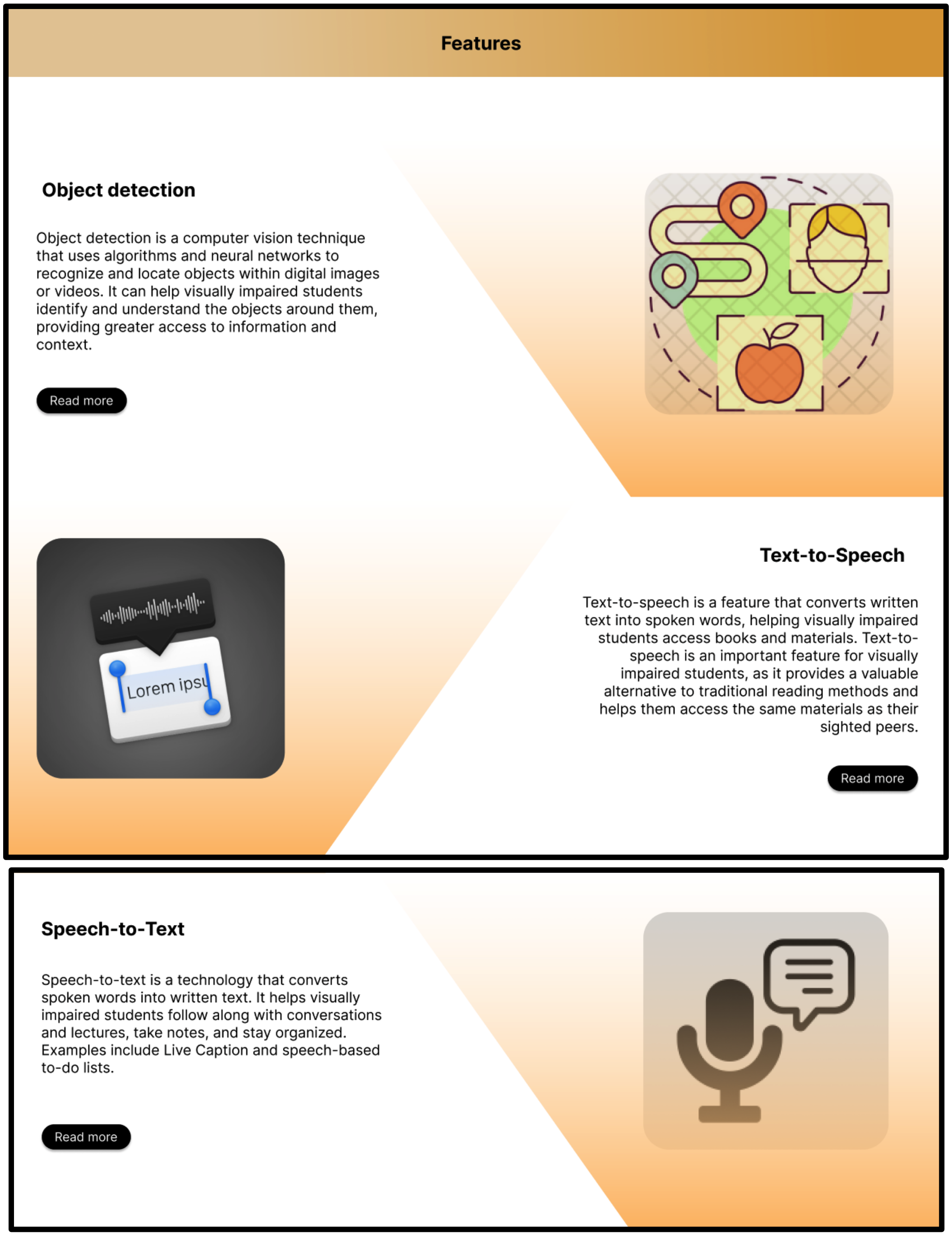
The code defines a React functional component called VoiceNavigation which allows users to navigate a web application using voice commands. The component creates an instance of speech recognition which listens for pre-defined voice commands and navigates to specific pages using React Router's useNavigate() hook. The code also includes a signout function and uses useEffect() to add an event listener for the "Space" key to start and stop listening for voice commands. This makes the application more accessible to users who have difficulty using a mouse or keyboard.

VI. RESULT AND DISCUSSION



The eyes are useless when the mind is blind

Fig 7: Homepage



Features

Object detection

Object detection is a computer vision technique that uses algorithms and neural networks to recognize and locate objects within digital images or videos. It can help visually impaired students identify and understand the objects around them, providing greater access to information and context.

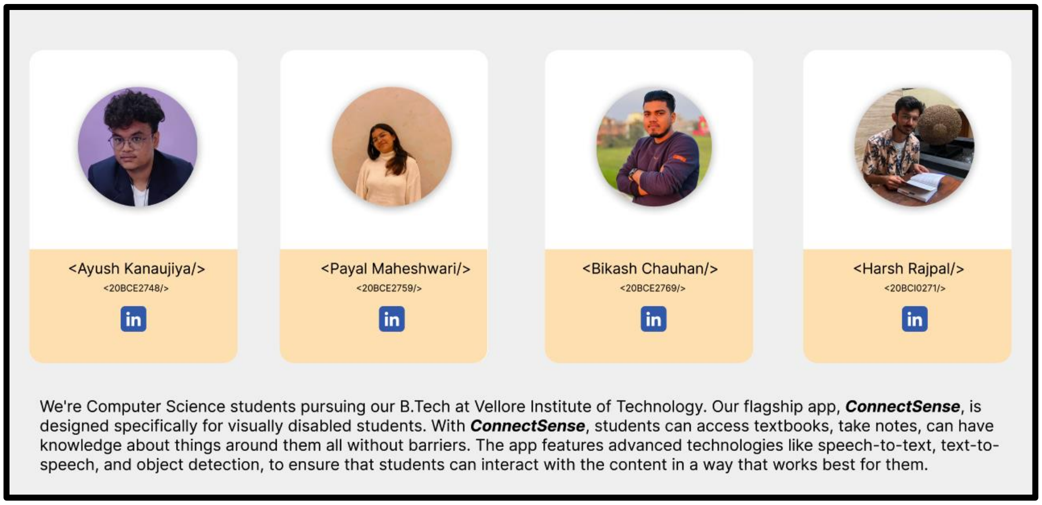
Text-to-Speech

Text-to-speech is a feature that converts written text into spoken words, helping visually impaired students access books and materials. Text-to-speech is an important feature for visually impaired students, as it provides a valuable alternative to traditional reading methods and helps them access the same materials as their sighted peers.

Speech-to-Text

Speech-to-text is a technology that converts spoken words into written text. It helps visually impaired students follow along with conversations and lectures, take notes, and stay organized. Examples include Live Caption and speech-based to-do lists.

Fig 8: Available Features



<Ayush Kanaujiya/>
<20BCE2748/>

<Payal Maheshwari/>
<20BCE2759/>

<Bikash Chauhan/>
<20BCE2769/>

<Harsh Rajpal/>
<20BCI0271/>

We're Computer Science students pursuing our B.Tech at Vellore Institute of Technology. Our flagship app, ***ConnectSense***, is designed specifically for visually disabled students. With ***ConnectSense***, students can access textbooks, take notes, can have knowledge about things around them all without barriers. The app features advanced technologies like speech-to-text, text-to-speech, and object detection, to ensure that students can interact with the content in a way that works best for them.

Fig 9: The developers

Our ConnectSense web app can be easily used by voice commands, The user can create an account and all data recorded by the application is stored in a local database. Our application will help a large population of physically disabled problems because of its simplicity to use. Its use is not just limited to disabled persons but can be used by anyone for day-to-day activities. Our application itself guides the user about the features available on the app and hence makes it easier for first time users to use the application without having the need to go through the user manual. The user manual available in our app guides the user with available features in the app.

We have developed an application that is beneficial for both visually and audibly impaired people increasing the use case in more areas. It's designed in a way that even a novice user can enjoy the benefits of our product. The speech navigation makes it very simple for them to access the application. The aim of this product is to help students in their daily academic activities. We strive to keep making the life of differently abled people better and efficient by continuously adding and updating more features.

The possibility are limitless and we are sure this will be a huge help in achieving quality of living for the people

Performance Analysis

Heuristic	Result
1.Visibility of system status	Our website keeps user informed about what is going through timely feedback
2. Match between system and the real world.	Language used is similar to what is used by users in day to day life.
3. User control and freedom	Users have full control on our website.
4. Consistency and standards	Concepts, words refers to same thing
5. Error prevention	System is designed to prevent serious usability errors
6.Recognition rather than recall	The user doesn't need to recall information from one part of dialogue to another.
7. Flexibility and efficiency of use	Site caters to all levels of users from novice to experts.
8. Aesthetic and minimalist design	Site dialogues doesn't contain irrelevant information
9. Help users recognize, diagnose, and recover from errors	Error messages are expressed in plain language and provide quick and easy instructions for recovery.
10. Help and documentation	The website has a help facility with a voice feedback to explain all the available commands and documentation to support the user's needs.

Test Cases

Test Case ID	Test Case Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
TC001	Verify the display of the main page	1. Open the Connect Sense application	NA	The main page of the application should be displayed with all the required components	User is redirected to home page	Pass
TC002	Verify the login functionality	1. Open the Connect Sense application. 2. Enter valid login credentials. 3. Click on the Login button.	Valid login credentials	User should be logged in and redirected to the dashboard page.	User will be logged in	Pass
TC003	Verify user cannot log in with incorrect credentials	1. Enter invalid username and password in login form 2. Click "Log In" button	Username: fakeuser Password: fakepassword	Error message is displayed	Error message is displayed	Fail
TC004	Verify the logout functionality	1. Open the Connect Sense application. 2. Click on the Logout button.	NA	User should be logged out and redirected to the login page.	User will be logged out	Pass
TC005	Verify the upload of an image	1. Open the Connect Sense application. 2. Click on the Object Detection button. 3. Select an image file from the local system. 4. Click on the Submit button.	Image with objects	The image will be detected and the answer of the object will be displayed	User will receive a voice taking the name of the object	Pass

TC006	Verify the voice navigation feature	1. Open the Connect Sense application. 2. Click on the Voice Navigation button. 3. Give voice commands for navigation.	Valid voice commands	The application should navigate to the corresponding page based on the voice commands.	The application to respond to user's voice commands	Pass
TC007	Verify the display of user profile details	1. Open the Connect Sense application. 2. Click on the Profile button.	NA	User profile details should be displayed.	Profile details will be displayed	Pass
TC008	Verify the display of the help documentation	1. Open the Connect Sense application. 2. Click on the Help button.	NA	The help documentation should be displayed.	Help documentation with commands will be displayed	Pass

The above scenario involves the development and testing of a web-based application called ConnectSense, which utilizes machine learning algorithms and computer vision techniques to assist visually impaired individuals in navigating their environment. The application allows users to capture real-time video using their device's camera and then uses object detection and recognition algorithms to identify and describe the objects in the user's surroundings. Additionally, the application provides voice navigation to guide the user in moving around their environment. The test cases table outlines various scenarios that the application must handle, such as detecting obstacles and providing voice instructions for navigation. The test cases are designed to ensure that the application meets the specified requirements and functions correctly in all scenarios.

VII. CONCLUSION AND FUTURE SCOPE

Our ConnectSense software offers a lot of room for growth in terms of accessibility for those with physical limitations. For further learning possibilities, the app's capabilities might be increased to incorporate extra exercises, tests, and audio material. By including features that help those with cognitive impairments or learning difficulties, the app may also reach students who are not physically disabled.

The programme may be developed into an online learning platform that offers certification programmes or degrees to students with physical disabilities. The software can help close the accessibility gap in education for physically challenged students, giving them access to the same possibilities as their peers, as its reach and functionalities grow. Our ConnectSense programme has the potential to significantly contribute to making the Internet a more welcoming and accessible place for all users with further development and improvement.

Future development can be done by adding the support for native languages which will help the user to select the language of his choice. This will enhance the number of people getting benefited by our application.

VIII. REFERENCES

1. C. M. Asad et al., "Removing Disabilities: Controlling Personal Computer Through Head Movements and Voice Command," 2018 IEEE 12th International Conference on Application of Information and Communication Technologies (AICT), Almaty, Kazakhstan, 2018, pp. 1-4, doi: 10.1109/ICAICT.2018.8747123.
2. V. Kalist, A. A. F. Joe and A. Veeramuthu, "An Aid for Speechless, Visually and Hearing Impaired Students in Classroom," 2020 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2020, pp. 104-108, doi: 10.1109/ICCES48766.2020.9137961.
3. L. Ciabattoni, F. Ferracuti, G. Foresi and A. Monteriù, "Hear to see - See to hear: a Smart Home System User Interface for visually or hearing-impaired people," 2018 IEEE 8th International Conference on Consumer Electronics - Berlin (ICCE-Berlin), Berlin, Germany, 2018, pp. 1-2, doi: 10.1109/ICCE-Berlin.2018.8576163.
4. B. F. Smaradottir, S. G. Martinez and J. A. Håland, "Evaluation of touchscreen assistive technology for visually disabled users," 2017 IEEE Symposium on Computers and Communications (ISCC), Heraklion, Greece, 2017, pp. 248-253, doi: 10.1109/ISCC.2017.8024537.
5. J. A. P. De Jesus, K. A. V. Gatpolintan, C. L. Q. Manga, M. R. L. Trono and E. R. Yabut, "PerSEEption: Mobile and Web Application Framework for Visually Impaired Individuals," 2021 1st International Conference in Information and Computing Research (iCORE), Manila, Philippines, 2021, pp. 205-210, doi: 10.1109/iCORE54267.2021.00055.
6. S. Noel, "Human computer interaction(HCI) based Smart Voice Email (Vmail) Application - Assistant for Visually Impaired Users (VIU)," 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 2020, pp. 895-900, doi: 10.1109/ICSSIT48917.2020.9214139.
7. S. Dobrisek, J. Gros, F. Mihelic and N. Pavesic, "HOMER: a voice-driven text-to-speech system for the blind," ISIE '99. Proceedings of the IEEE International Symposium on Industrial Electronics (Cat. No.99TH8465), Bled, Slovenia, 1999, pp. 205-208 vol.1, doi: 10.1109/ISIE.1999.801785
8. K. S. Sri, C. Mounika and K. Yamini, "Audiobooks that converts Text, Image, PDF-Audio & Speech-Text : for physically challenged & improving fluency," 2022 International Conference on Inventive Computation Technologies (ICICT), Nepal, 2022, pp. 83-88, doi: 10.1109/ICICT54344.2022.9850872