Project: TalkTask

Your Personal To-Do List Tracker

Contributors (Team 20):

- Ivan Wong (Software Configuration Management Coordinator)
- Tristan Vosburg (Back-End Developer)
- Nicholas Woodley (Front-End Developer)
- Kai Lindskog-Coffin (Database Developer)
- Raymond Cen (Back-End Developer)
- Bailey Bundlong (UI/UX Designer)
- Jordan L Cowan (Front-End Developer)

Important Resources:

- Git Repository: https://github.com/IvanW5X/CS362-Winter2025-Team20-TalkTask
- WebSpeech API Documentation: https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API/Using_the_Web_Speech_API
- Database (MongoDB): https://www.mongodb.com/
- OSU Web Server
- React + Vite
- Node.js

Communication Channels:

- Microsoft Teams chat that all members have been invited too
- Direct message group chat for simple communication
- Email

Rules of Communication:

- Each team member must ensure they check each mode of communication at least twice a day.
- Each team member must respond to every message, It could be as simple as a thumbs up as long as the message is acknowledged.
- Each team member must ensure the team knows about any issues that prevents them from completing their tasks or from attending meetings and classes.

Abstract:

TalkTask is a web application that utilizes speech recognition to streamline to-do list management. Using speech recognition, we are able to allow users to add, remove, prioritize, and even schedule daily tasks with simple voice commands for multiple languages including, but not limited to English, Spanish, French, and more. Designed with accessibility and flexibility in mind, TalkTask provides a hands-free and user-friendly experience across various devices such as desktops and mobile devices. The platform prioritizes ease-of-access for all users and user privacy, ensuring secure handling of voice input. Stretch goals or future iterations aim to implement a database to allow users to save and store their tasks after going offline, and set reminders for upcoming tasks if given a time requirement.

Goal:

The goal of our application is to improve accessibility to users while allowing users to maintain an optimal lifestyle where they can efficiently plan their day. Our application addresses the issue of accessibility in current to-do list applications. Users today are required to interact with a mouse, keyboard, touchscreen or something similar in order to use such an application. But with our application, by using voice input and voice commands, our application aims to help users perform the essential features of a to-do list application through manual hands-on input and voice input.

Current Practice:

The current practice of today is that users can add and delete tasks, set start or due dates, provide descriptions for the tasks and mark them as completed or incomplete. Tasks can also be categorized into different groups for it to be more easily visualized. Organization of the tasks can depend on due date, priority or status. While these are useful, there are many limits to this current practice. Most of the modern-day to-do list applications require manual input. Requiring users to interact with a mouse, keyboard, touchscreen or something similar. These types of interactions can be inaccessible to many people, especially to those with physical disability in their hands. For example, typing tasks and marking/clicking tasks. It also requires the user to focus on the screen to see what they are doing. Which can be an inconvenience when a hands-free option may be preferred.

Novelty:

The to-do list is a powerful tool that can be used to help organize, categorize, and optimize a given time period. In our current day, there are many different versions of the to-do list on the market today; including some made by major corporations like Microsoft and Apple. Our team is aiming to create a to-do list that can be tailored to a user-specific time period. We aim to create a UI that is accessible to all users, while putting a unique touch. One of our stretch goals is to implement a database that can be utilized in creating user task suggestions, aiding in user productivity and usability. We plan to implement an API that allows users to speak their goals into the page, as well as navigate through menus and actions with their voice.

Our to-do list will be different in the way that it will draw from many common day practices of the present to better aid the person in their own daily lives. The to-do list can be utilized beyond a simple stack of deliverables, and we feel that it can be used to help organize much more than one's day. We aim to implement voice recognition AI to aid

Effects:

As with other to-do list applications, ours will help users plan their time efficiently. However, TalkTask also has greater accessibility for users, due to the speech recognition. People with disabilities will find it easier to use compared to traditional lists, and not all applications have this accessibility option. Non-native speakers will also have their experience improved, as there are multiple language options. Our application focuses on accessibility first, as that is very important for software that a lot of different people might use. Because of this focus on efficiency and accessibility, multitaskers will also enjoy using the software. They can use voice recognition to quickly create and edit lists, compared to the more traditional, slower applications. We want to create an application that is simple to use but that also accompanies different types of users. If successful, our application will be easier to use than previous to-do lists and address previous accessibility concerns.

Technical Approach:

Our web application will be implemented using the React framework, the Vite build tool for efficient builds and improved performance, JavaScript for backend/business logic, and (hopefully) the OSU web server to publish our website onto the internet. The main component of the project will be based on the WebSpeech API, which will be used to get input from the user, so that our program will be able to parse it and handle the commands given. We will split up the user interface into components for a modular design, so that with each sprint, we are able to integrate changes and features seamlessly and effectively. Additionally, if we have the time, we can use tools like MongoDB to implement a database and user accounts to store tasks onto the cloud, so that users will be able to exit and return to the application with their previous session data saved. Collaboration and communication will be done via Git, MS Teams, Zoom, text messages, and in-person meetings to increase productivity and push changes into the codebase. Other tools, such as Figma are still being decided on whether or not we will use it, but if so, we will use it for a reference as to how our web UI will be designed, so that there's no room for ambiguity for how the UI shall be implemented.

Risk:

There are two risks facing our group as we attempt to design, create and implement this application. We have worked together as a group before, so we are aware of the fact that we are prone to procrastination and underestimating the work required to complete certain tasks. To avoid this problem, we plan on setting checkpoints for what we expect to have completed each week. Additionally, each checkpoint will have small tasks associated with it. Each group member will be responsible for completing their small tasks. At each stand-up meeting, the status of each small task will be checked in to see if help is needed, changes need to be made, or if our stretch goals can be attempted because we are ahead of schedule. Our second challenge is implementing a user database so that separate users can use the application, along with state being preserved between uses and devices. We have been warned several times that implementing something like this has many challenges, so we must be careful not to over exert our resources on this stretch goal and to make sure it is implemented accurately if we do choose to attempt it.

Major Goals:

1. Create a working To-Do list where the user can add, delete, and update tasks in a simple & understandable manner.
2. Implement a database to store past list items.
3. Use an AI to recommend tasks based on previous tasks.
4. Make a working UI that not only looks good, but interacts in a way that makes users want to continue using the app.

Stretch Goals:

- Implement reminders for timely tasks to increase the success rate of tasks being completed.
- Optional Idea: Gamify the web application to keep current users engaged and promote continuation of using the app into continuing using our app for productivity.
- Extend AI API functionality to process natural language and allow users to communicate naturally to our speech recognition system and convert their commands into a dedicated function for the to-do list management system.