

CMPSC472_finalproject

Installation

1. Clone the repository to your local machine using the following Git command:

```
git clone git@github.com:darsh8692/CMPSC472_finalproject.git
```

2. Install the npm packages:

```
npm install
```

3. Run the project using following command:

Runs the app in the development mode.

```
npm start
```

Launches the test runner in the interactive watch mode.

```
npm test
```

Builds the app for production to the **build** folder.

```
npm run build
```

Usage instructions

Dashboard Navigation:

- Upons running the app, users will be directed to the dashboard.
- Use the navigation menu to explore different sections: Home, Entries, Personalized Prompts, and New Journal entry.

Create a Journal Entry:

- Click on the "New Entry" or "Create Journal Entry" button.

- For your entry begin typing or use the voice-to-text feature to record your thoughts.

Personalized Prompts:

- Explore the "Personalized Prompts" section for suggestions to enhance your gratitude practice.
- Respond to prompts by typing.

Project Overview

The Gratitude Journal Web Application is a user-friendly digital platform designed to cater specifically to elderly users, providing them with a seamless and accessible tool for expressing gratitude, enhancing emotional well-being, and reducing feelings of loneliness.

Some of the OS concepts used in the project are:

File System Interaction:

The project simulates file system interactions using the localStorage API in the browser. This abstraction allows for data persistence similar to how files are stored in a traditional file system.

System Calls:

The project utilizes underlying system calls made by the browser when interacting with the localStorage API for storing and retrieving data.

Project Goals

- Develop a user-friendly Gratitude Journal web application tailored specifically for elderly users.
- Implement features for creating journal entries, including personalized prompts and voice-to-text entry.
- Prioritize simplicity and accessibility in the user interface to accommodate varying levels of technological familiarity among elderly users.
- Integrate gentle reminders to encourage and support consistent engagement with gratitude practices.
- Provide an option for users to attach images to their entries, fostering a more personalized and meaningful reflection.

Significance of the Project

Enhancing Emotional Well-being:

- Cultivates a positive mindset by facilitating the expression and acknowledgment of daily moments of gratitude.
- Supports mental and emotional well-being by encouraging regular reflection on positive aspects of life.

Tailored Experience for the Elderly:

- Addresses the specific needs and preferences of elderly users through features like larger fonts, simplified navigation, and voice-to-text entry.

- Recognizes the importance of customization in technology for diverse user demographics.

Digital Companion for Loneliness:

- Acts as a digital companion for the elderly, fostering a sense of connection and companionship.
- Provides a platform for self-expression and reflection, reducing feelings of isolation.

Multimodal Input with Voice-to-Text:

- Enhances accessibility by including a voice-to-text feature, allowing users to effortlessly create journal entries without the need for typing.
- Acknowledges and accommodates diverse abilities and preferences among elderly users.

Code Structure

Data Storage Location:

- The data and entries are stored on the client side within the browser's localStorage. This local storage provides a simple way to persistently store and retrieve JSON-formatted data.
- The React component (Entries) utilizes the localStorage feature to maintain and persist the journal entries. The entries are stored as a JSON-formatted string, allowing them to survive page reloads or closures.
- In this implementation, data storage is handled entirely on the client side, making it suitable for scenarios where server-side storage or a database is not required.

Components:

This directory contains individual React components that are used to build the application's user interface.

AddEntryButton.js: Component for the "Add New Entry" button and the modal for creating entries.

Entries.js: Component for managing journal entries, including adding, updating, and deleting entries.

Header.js: Component representing the application header/navigation bar.

PersonalizedPrompts.js: Component for displaying personalized prompts for journal entries.

UserResponsesDisplay.js: Component for displaying user responses to prompts.

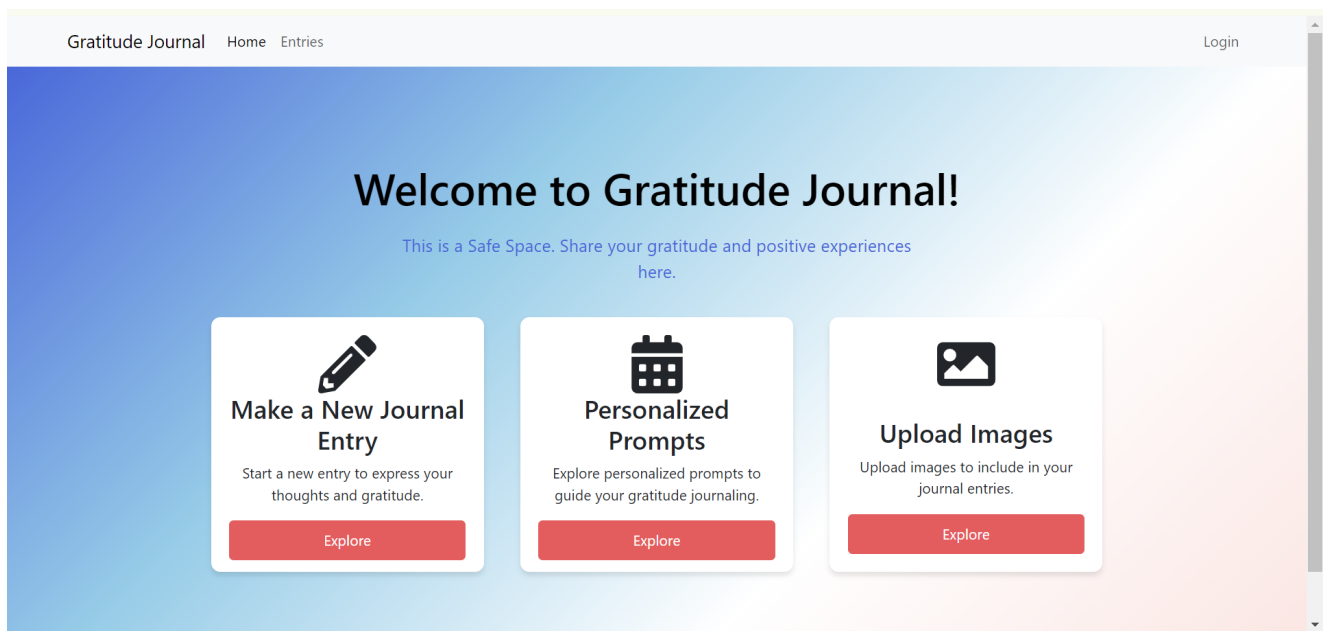
Functionalities and Test Results

Home Page:

It will display the user interface that will allow the user to navigate to different areas in the app. The User Interface is designed to simplify the task categorization for the elderly specifically.

- Path to make a journal entry
 - 'Entries' tab on the top bar will allow to see the entries directly.
- Path to Prompts window
- Path to add images to the entries

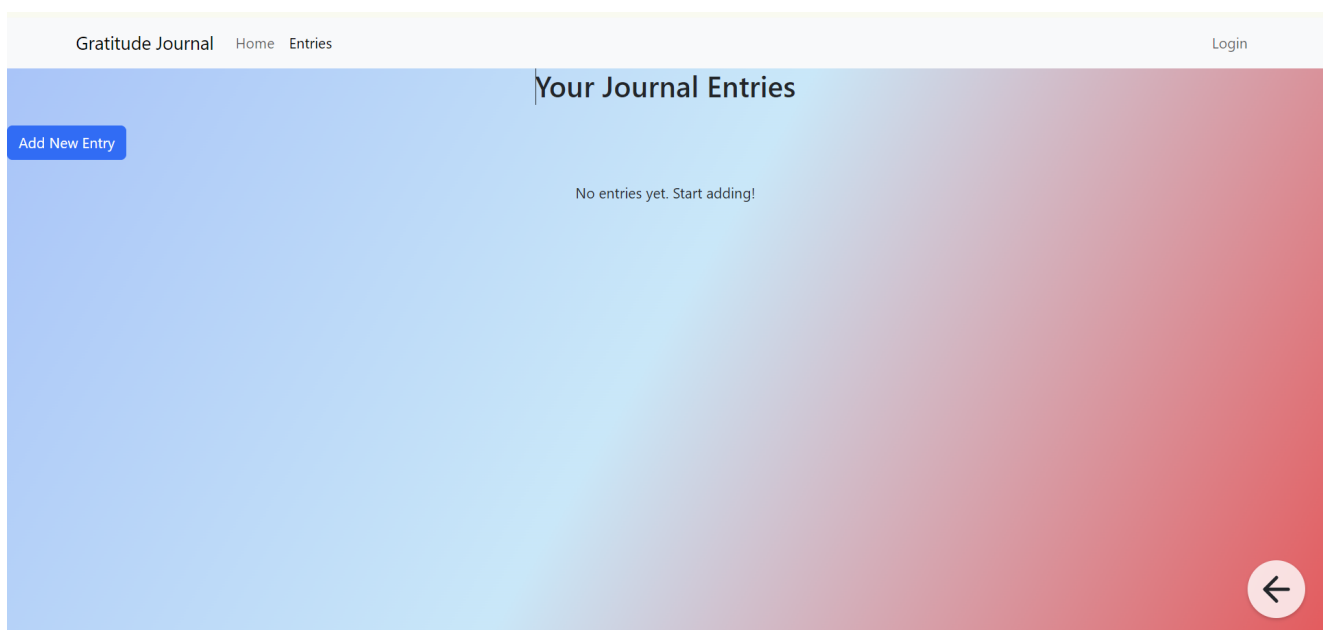
Test Results:

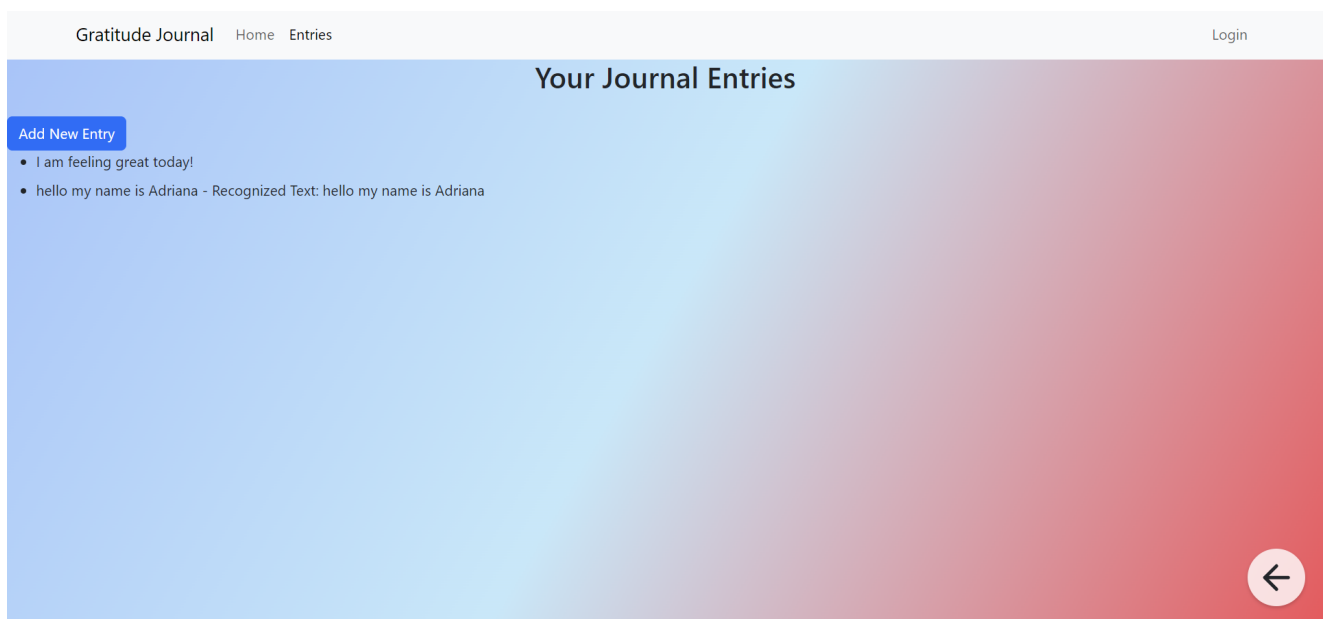
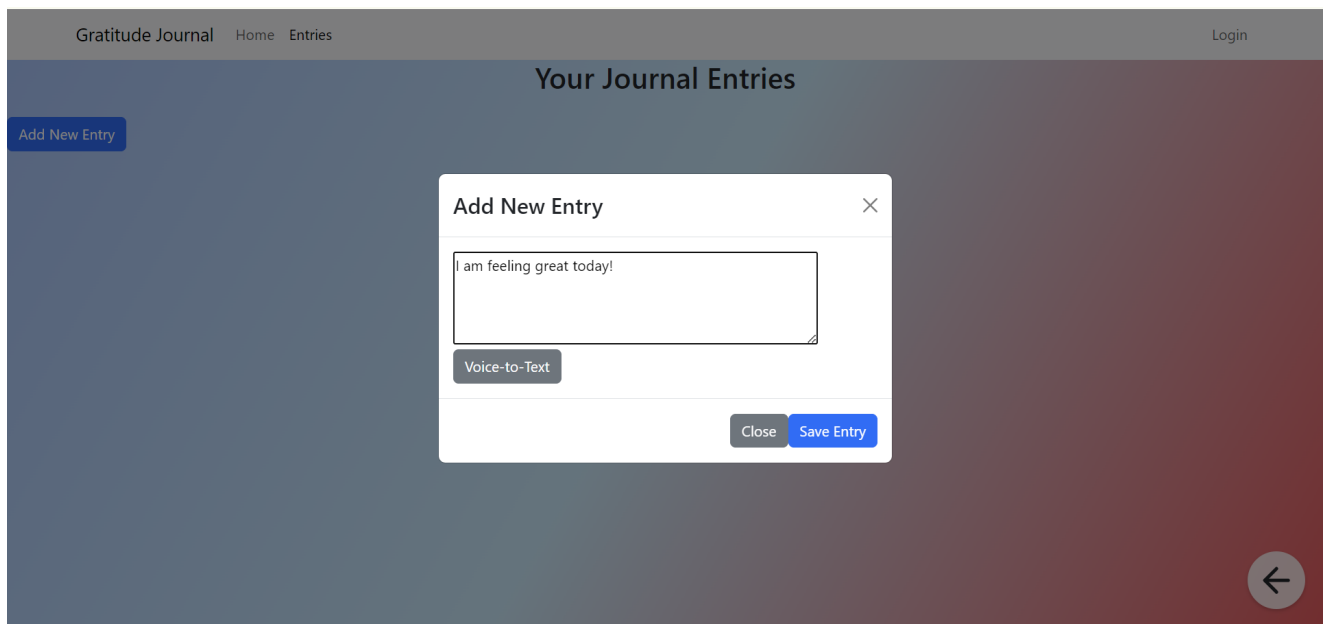


Make a New Journal Entry:

- Take the user to 'Entries' page where:
 - Users can add their feelings, notes, or their thoughts in two primary ways:
 - Entering their input in the text box and saving the entry.
 - Using a voice-to-text operation which will record thier voice and automatically form an entry when the user stops speaking.
 - Display Data: Notes will be displayed accurately. The voice notes will have an additional display feature, 'Recognized text:' which will tell the user what computer interpreted from their voice. Through this, user can also identify the way the note was added.

Test Results:

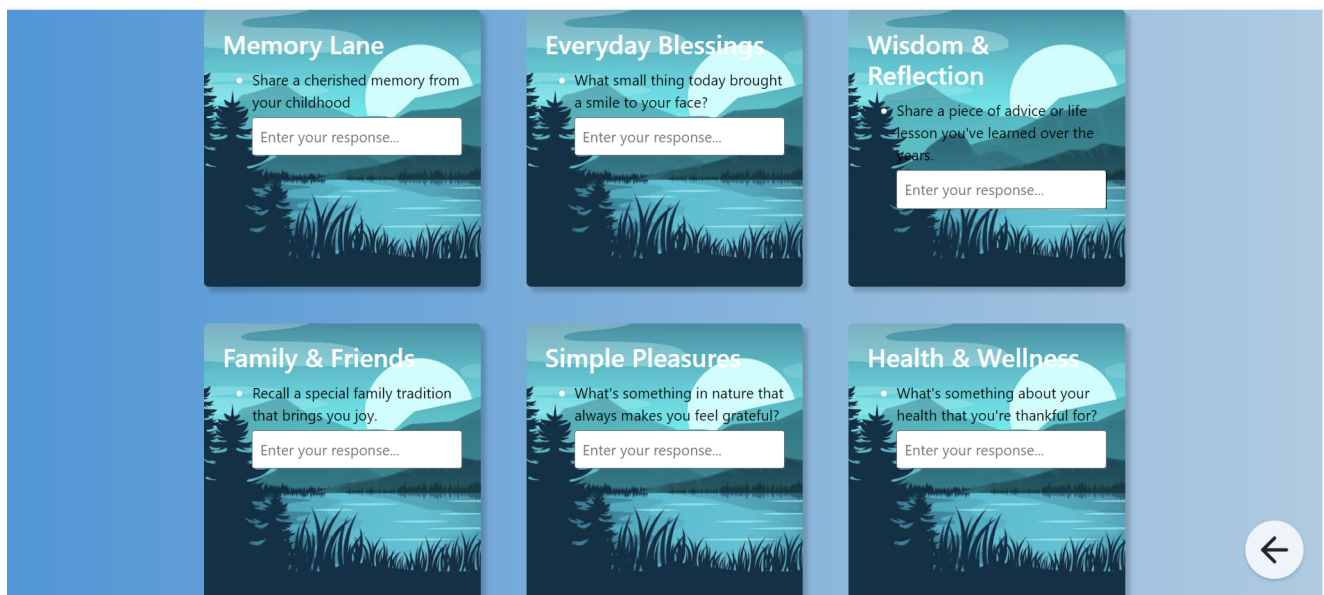




Personalized Prompts:

- **Answer Prompt:** Displays 6 prompt cards with different prompt in each. Users can enter their response to one or all prompts and then submit their response.
- **Display Responses:** Users will be displayed with their responses to each category they answered on the previous screen.

Test Results:



Upload Images:

- **Integrating Image Adding Functionality:** Users will have the opportunity to express themselves through multiple mediums—writing, speaking, and adding images—to convey their feelings and experiences more comprehensively.
 - User will be able to add images (png, jpeg, etc) to thier entries.
 - Delete the image that was added previously.
- **Enhanced Memory Recall:** Incorporating photos or visual elements into their gratitude entries can help users recall specific moments and experiences more vividly, reinforcing the positive emotions associated with those memories.

Test Results:

*THIS AREA OF OUR APP IS UNDER CONSTRUCTION.
WE APPRECIATE YOUR PATIENCE.*

Upload Images

Choose File



Discussion and Conclusion

Issues:

- **Memory Management:** The app's image upload functionality could pose memory management challenges, especially on devices with limited storage capacity. Large image files might strain the system resources, affecting overall app performance. This is also why the image uploading functionality has not been incorporated completely into our app as of now. It is part of one of the features that could be added to the application in future.
- **Concurrency and Synchronization:** Simultaneous access to the app by multiple users or simultaneous operations like uploading images and writing entries might lead to issues related to concurrency and synchronization. Conflicts in accessing and modifying shared resources could occur.
- **File System Limitations:** Handling a large number of image files within the app might lead to issues related to file system limitations, especially in terms of file organization, naming conflicts, or potential fragmentation.

Limitations:

- **Device Compatibility:** Older devices commonly used by the elderly might have limited processing power or outdated operating systems, which could limit the app's functionality or performance.
- **User Interface Adaptability:** The app's interface might not be easily adaptable to different screen sizes or accessibility features present in various devices, creating potential problems with usability for some users.
- **Security Concerns:** Storing personal images within the app could raise security concerns. To deal with this, we would be required to ensure data encryption and implementing secure storage measures to protect users' private information.

Application of Course Learnings:

- **Memory Optimization:** Knowledge of memory management techniques taught in the course helped in optimizing the app's memory usage. It will also come in handy while ensuring efficient handling of image data and preventing memory leaks.
- **Concurrency Control:** Concepts of synchronization and concurrency control can be applied to manage simultaneous access to shared resources within the app, ensuring data integrity and preventing conflicts during image uploads or journal entries.
- **File System Handling:** Understanding file system structures and limitations guided the design of a robust file management system within the app, addressing issues related to file naming, organization, and potential fragmentation.

Conclusion

In conclusion, considerations related to memory management, concurrency, file system handling, and device compatibility are crucial. Applying concepts learned in the course can aid in mitigating these issues, optimizing app performance, and ensuring a smoother user experience for elderly users.