## PROJECT NAME

Gratitude++ Journal Web Application

## MEMBER NAMES

Joel Scott, Benjamin Seifried

## ABSTRACT

This project proposal outlines the development of a dynamic web application designed to empower users into cultivating a sense of gratitude through personalized journaling. The application will be built using a technology stack including React, TypeScript, Bootstrap and Firebase.

The application will allow a user to express themselves using various content formats including text video and audio files. The daily titles of the journal entries will be utilized in creating an interactive “word-cloud” that will display keywords more prominently depending on their number of inclusions. By blending technology and personal expression, the application aims to promote mental well-being and foster an “attitude of gratitude" and provide users with an innovative and meaningful experience.

## DESCRIPTION

The premise of the project is to create a web-based application that will help to counteract a user’s tendency to “doom-scroll”. In this day and age, users are constantly inundated with the negativity of polarizing politics, near daily global catastrophe, and a media bombardment that focuses on our problems to try and sell us on solutions. The idea of keeping a “gratitude” journal is not new, but still seems to be a relevant respite from a negatively focused world. Creating that respite is what the “Gratitude++ Journal” hopes to help counteract.

This project will be a web-based journal that will allow for users to store their thoughts daily and keep track of all entries they have made. The headings of each of these entries will be compiled and be converted to a viewable word cloud that will show all the things a user has been grateful for in previous entries. Once the word cloud has been created, the user will preferably be able to click on each of the words in the cloud to show them all the entries with that key word in the heading. A feature that will exist for users is to allow them to record their thoughts and feelings in multiple ways without being constricted to just text input.

The site will feature categorization for the users to keep track of their journal entries. There will be a way for users to view the headings of all entries they have created and click on these headings to view their entries. Every day there will be space for a new entry into the journal for users to write into.

The user will be given the option to create and record these entries as text, audio message, images, or short user created videos into their journal. The text will be added by creating a text box on the journal that allows them to record their entry by either typing or voice to text. This will be accomplished using a Google Cloud Services feature. The insertion of images and videos will be done by allowing users to upload these media files and checking and formatting them into the correct file type, i.e., MP3 and MP4. The users will have a personal page setup for themselves that allows them to look back and view all previous entries they have made. This includes a date to know when the entries were created.

The website will have a community page for users to share information between the users. The community page will be a main page when the user logs in to the site where they can view entries that others have and decided to share. This page will be a main page of the website that will keep track of the posts that people want others to see, whether it be an image, video, or text. This community page will not feature any comments from the users on the selected post to limit the need for moderation on the site. There will be a way for users to flag posts they feel are offensive, which will remove the post from being viewed on the community page but will still allow them to view the post on their personal page.

The site will feature a user login screen with authentication to ensure that the data they have will be secured from others. Users will be required to create an account personal to them with a username and password to verify them before entering the site. This will be done by using security features from outside sources to ensure that the information will be secure and not easily breached. The site will have an administrative account setup to control certain aspects of the site and to manage all the users. This account will also be accessed through the login screen and require authentication to access. This account will have control in the event of a community page to manage posts made to the community and determine if they meet the community guidelines or not.

The site will be run and tested using Windows 10/11, so it will be ensured that the site will work on both operating systems. The work for the site will be completed using Visual Studio Code as our Integrated Development Network (IDE) to compile all the necessary information together to build the site. Programming languages that will be used to complete this task will be Hyper Text Markup Language (HTML), Cascading Style Sheets (CSS), JavaScript, Typescript Java. The project will also pull from 3rd party libraries React and Bootstrap to make the process easier. Bootstrap will be used as an addition to CSS to help the front-end development of the site. React will be a library that will be used as a front-end user interface library to design the overall look of the site.

The website will only be run on Google Chrome to ensure that the site is accessible for all users. The restriction of the browser will allow for the development of the site to be created so that all users will have full access to all features of the site. The browser will be used in conjunction with the server hosting network to create a seamless way for users to utilize the full functionality of the website.

The website will feature a server component and a database to store the user information and user authentication. The site will have a server hosted on Google cloud services and will

feature a database from Firebase. Google cloud services will be used along with its features to help and implement the functionality of the site and ensure that the site remains operational for the users. Firebase will be used as the primary database for user data management across the site. Users' personal data, along with their authentication information will be stored within the database and will use its security to ensure the data is safe.

For the version control and aspects relating to such GitHub will be used by the developers to store the information relating to this project. The text files that will be needed, including all code that has been written, will be categorized into GitHub and be viewable by all developers to allow them to work on the project simultaneously and seamlessly. GitHub will also provide storage for all previous and current versions of the application so that the developers may transition back to earlier versions of the project if necessary.

# FEATURE LIST

## FEATURES THAT WILL BE COMPLETED

- Space for daily journal entries.
- Tab to look back at entries from previous days.
- A dynamic word cloud that will be compiled from all the user's entry headings.
- A place for the user to put a personalized heading for each entry.
- Allow the words or simply the word cloud itself to be clickable, showing all journal entries that have that specific keyword associated with them.
- Journal entries will be able to be recorded by a user text, voice to text, images, and small videos.
- A personal page for the user to view all their entries.
- A user login feature with authentication.
- Categorization by keywords.
- An admin login account to manage all accounts registered with the site.
- Will work on Google Chrome web browser.
- A community sharing page for users to post journal entries they would like to share with other users on the site.

## FEATURES WE HOPE TO IMPLEMENT

- A tracker for the data and time so that they may see when the entry was done.
- A calendar to look back through past journal entries and see what days were and weren't written in.
- Community moderation that would allow users to upvote and downvote the post.
- A button that would allow users to flag posts they feel are offensive in some way, allowing moderators to look over the post.
- Will work on Firefox.
- A dark/light mode toggle
- Google account authentication to log into the site.
- The site will be mobile friendly.

## FEATURES THAT WILL NOT BE IMPLEMENTED

- The application runs as an Android/IOS application.
- Push notifications via email prompting the user to add their daily journal entry.
- Email notifications to tell you how many posts you made that week.

# TECHNOLOGY

## PLATFORM

Our project will be a web-based application.

## OPERATING SYSTEM

Our website will run on Windows 10/11.

## IDE

We will use Visual Studio Code as our IDE.

## PROGRAMMING LANGUAGES

HTML, CSS, JavaScript, Java, TypeScript

## 3RD PARTY LIBRARIES AND TOOLS

React, Bootstrap

## SERVER SOFTWARE

We are going to be using Google Cloud servers as our server, and Firebase as our database.

## COMMUNICATION SOFTWARE

We will be using Discord and Google Docs as our communication software.

GitHub Repository will be used for version control.

## SERVER INFORMATION

This project will utilize Firebase for its server needs.

## DATA SOURCES

No external data sources are known currently.

## TEAM MEMBER'S BACKGROUNDS

Joel is a senior at Austin Peay State University pursuing a B.S in computer science with a concentration in computer theory and systems. He has experience with programming languages such as Java, C#, c++, JavaScript, C, and python. He has experience in basic website design and application creation. Joel will be designated as the Back-end focused developer for the sake of the labor division.

Benjamin is a senior at Austin Peay State University pursuing a B.S in computer science with a concentration in software engineering. He has experience with programming languages such as Java, C#, c++, JavaScript, HTML, CSS. He has experience creating, modifying and deploying websites for content based and e-commerce-based assets. Benjamin will be designated as the Front-end focused developer for the sake of the labor division.

## DEPENDENCIES, LIMITATIONS, AND RISKS

**Risks:**

The use of programming languages, libraries and frameworks that are relatively new to both developers.

The use of Firebase for the first time by both developers.

This is the first time that the developers have worked together.

The fact that both developers must find a way to balance additional coursework, employment and family obligations.

The team believes that adequate time management and planning will help to mitigate most of these risks. The developers have agreed to weekly check-ins and to keep an open line of communication throughout the project to voice concerns and bring up any blockers as they occur.

## DEPENDENCIES:

The project will require the web-based application to run on a server and must connect successfully with the database.

The team believes that by doing proper testing of the interconnectivity of these dependencies the risk of failure can be mitigated.

## TIMELINE

**Weekly Timeline**

While both developers will be involved with all aspects of the application development and launch, it seems prudent to divide and delegate development roles according to Front-end and Back-end focused roles.

The following schedule is tentative, and roles may be reallocated according to real world application as the project progresses.

In general:

Benjamin will be tasked with researching and implementing front-end development/React, TypeScript, Bootstrap

Joel will be tasked with researching and implementing back-end/Firebase, Google Cloud,

Both developers will ensure the other understands the holistic view of the development process and will both be tasked with integration of front and back end as well as launch.

| |
|---|
| **Week 1 Project Planning I**<br><br>**Sept. 11-17**<br><br>Both will continue collaboration with project scope, defining objectives and researching the above referenced stack of development tools (Firebase, Google Cloud, JavaScript, TypeScript, React)<br><br>**Benjamin** focuses on learning and researching technology stack with a focus on Front-End design concepts. (React, TypeScript, JavaScript)<br><br>**Joel** focuses on learning and researching technology stack with a focus on Back-End design concepts. (Firebase, Google Cloud) |
| **Week 2 Project Planning II**<br><br>**Sept. 18-24**<br><br>Building Development Environment, Setting Up Version Control (GitHub), UI Design |

| |
|---|
| **Benjamin** continues technology stack research and planning, setting up development environment ensuring both developers are prepared.<br><br>**Joel** continues technology stack research and planning, initializes accounts Firebase, Google Cloud(or another server) |
| **Week 3 Front-end Development (ReactJS) I**<br><br>**Sept. 5-Oct. 1**<br><br>**Benjamin** works on the development of the UI layout using HTML/CSS with ReactJS<br><br>**Joel** assists with Front-End tasks |
| **Week 4 Front-end Development (ReactJS) II**<br><br>**Oct. 2-8**<br><br>**Benjamin** works on the development of the UI layout using HTML/CSS with ReactJS<br><br>**Joel** assists with Front-End tasks. |
| **Week 5 Back-end Development I**<br><br>**Oct. 9-15**<br><br>**Joel** works on setting up Firebase Realtime Database or Firestore.<br><br>**Benjamin** works on journal entry page |
| **Week 6 Back-end Development II**<br><br>**Oct. 16-22**<br><br>**Joel** continues back-end development working on user-authentication.<br><br>**Benjamin** subject categorization and word-cloud implementation |
| **Week 7 Integration and Testing I**<br><br>**Oct. 23-39**<br><br>Both developers work on connection between front and back end<br><br>**Benjamin** continues front-end focus and integration.<br><br>**Joel** continues backend focus and integration |
| **Week 8 Integration and Testing II** |

**Oct 30-Nov. 5**

Both developers work on connection between front and back end

- **Begin Building Poster Requirement**

**Benjamin** continues front-end focus and integration.

**Joel** continues backend focus and integration

**Week 9 Additional Features and Refinement I**

**Nov. 6-12**

Both developers will continue to refine core features and start to prioritize and implement additional features as needed/desired

**Benjamin** continues front-end focus and integration.

**Joel** continues backend focus and integration

**Week 10 Additional Features and Refinement II**

**Nov. 13-19**

Both developers will continue to refine core features and start to prioritize and implement additional features as needed/desired

**Benjamin** continues front-end focus and integration.

**Joel** continues backend focus and integration

**Week 11 Testing and Quality Assurance**

**Nov. 20-26**

Test Test Test Test

- **POSTER DUE NOV. 22nd**

**Benjamin** focus on fixing front end issues and integration

**Joel** focus on fixing back-end issues

**Week 12 Deployment and Staging**

**Nov. 27-Dec. 3**

| |
|---|
| Preparing full deployment<br><br>• **Both developers should be preparing themselves and each other for the technical interview as well** |
| **Week 13 Documentation, Deployment, Launch**<br><br>**Dec. 4-10**<br><br>Updating documentation and should have final launch ready.<br><br>• **TECHNICAL INTERVIEWS DEC. 6th** |
| **Week 14 Finalization, Presentation**<br><br>**Dec. 11-14**<br><br>Finalizing presentation<br><br>• **PRESENTATION, REPORT, FINALIZED CODE DUE DEC. 14th** |

Document Created: September 2023

Creators: Benjamin Seifried  Joel Scott