Senior Project Weekly Status Report

Student Name: Enmanuel De los Santos Cruz **Email: del20047@byui.edu**

Phone or Zoom Contact:

Project Title: Kairo

Team Members (if applicable): Myself

Project Description: The primary purpose of Kairo is to solve two fundamental problems with traditional journaling: high friction and low insight. Most people find the act of typing out daily thoughts tedious, leading to inconsistency. Furthermore, a standard journal is merely a static log of text, making it difficult to discover meaningful patterns, track emotional trends, or reflect on past experiences in an organized way. Kairo aims to transform journaling from a chore into a seamless, insightful conversation with oneself.

Technologies Used: I am still deciding but so far, I will probably use React.JS in the frontend and python in the backend.

GITHub URL: https://github.com/EnmaSantos/kairo

(Add MountainDad as collaborator)

Week 1: 9/20/25

Overall Status (on-schedule, behind, ahead):

Number of hours worked this week: 3

Total number of hours worked on the project thus far: 3

Number of total hours anticipated at completion: 126

Accomplishments: Just planning so far

Challenges: The scope of the project, I still do not know how long is going to take me.

Plans / Goals for next week: Decide for sure the project scope, and make a plan on how the 9 hours are going to be spent each week.

SPED Talk Insight (Briefly describe an insight or something interesting you learned from the SPED talks this week):

Other comments for the instructor:

Week 2: 9/27/25

Overall Status (on-schedule, behind, ahead):

Number of hours worked this week: 6

Total number of hours worked on the project thus far: 9

Number of total hours anticipated at completion: 126

Accomplishments: This week, my main accomplishment was fully defining and solidifying the project concept as "Kairo." I created a comprehensive project proposal that outlines the entire semester's work, including defining the project's purpose and significance, identifying the new computer science concepts I will need to learn (RAG, Vector DBs, WebSockets), creating a detailed, multi-phase project schedule with clear milestones, and identifying all project dependencies and potential risks.

Challenges: The primary challenge I faced was narrowing down the project idea into a concept that is both technically impressive and achievable within the semester. I had to pivot from a few initial ideas that were either too risky in a professional context or too similar to existing products. This process required significant critical thinking, but I believe it resulted in a much stronger and more unique final project plan.

Plans / Goals for next week: My main goal for the upcoming week is to begin Phase 1: Foundation & Backend. After submitting the project proposal and awaiting instructor feedback, I plan to set up my local development environment and initialize the project repositories on GitHub. I will then create the basic file structure for the FastAPI backend server and begin designing the initial PostgreSQL database schema for the users and journal entries.

SPED Talk Insight (Briefly describe an insight or something interesting you learned from the SPED talks this week): The talk highlights that accessibility should be adopted as a mindset for designing inclusive, valuable, and future-proof software, going beyond its treatment as a mere feature. While often mandated by laws like the ADA, accessibility also addresses the needs of users lacking resources, such as those without a laptop or reliable internet access.

Other comments for the instructor:

Week 3: 10/4/25

Overall Status (on-schedule, behind, ahead): Behind

Number of hours worked this week: 7

Total number of hours worked on the project thus far: 16

Number of total hours anticipated at completion: 126

Accomplishments: This week was focused on validating the project's technical feasibility on my local hardware. I successfully ran benchmark tests on smaller AI models, which confirmed that my MacBook Air is capable of handling the planned workload by using its GPU for acceleration (MPS). While troubleshooting larger models, I gained experience in diagnosing and attempting to solve complex, low-level library conflicts specific to my platform.

Challenges: The entire week was consumed by a significant and persistent technical roadblock. I encountered a low-level system error (libc++abi: terminating due to uncaught exception... mutex lock failed) when trying to run the audio processing benchmark script. This error has so far prevented the project from moving forward, as it blocks a core component. Standard fixes, such as setting environment variables, did not resolve the issue, suggesting a deeper problem within the Python virtual environment itself.

Plans / Goals for next week: The top priority for next week is to resolve the technical blocker. My plan is to execute a "Fresh Start" by completely deleting the current Python virtual environment and rebuilding it from scratch to ensure a clean and stable installation of all dependencies. Once the environment is fixed and the benchmark scripts run

successfully, I will immediately begin the work originally planned for this week: setting up the FastAPI server structure and designing the PostgreSQL schema for Project Kairo.

SPED Talk Insight (Briefly describe an insight or something interesting you learned from the SPED talks this week): The SPED talk on parsing log files demonstrated the importance of making raw, unstructured data understandable. This directly relates to my Kairo project, which applies the same principle by parsing spoken journal entries instead of system logs. It highlighted that the core skill of extracting meaningful patterns from information is a fundamental task across many different areas of software engineering.

Other comments for the instructor:

Week 4: 10/11/25

Overall Status (on-schedule, behind, ahead): Behind

Number of hours worked this week: 4

Total number of hours worked on the project thus far: 20

Number of total hours anticipated at completion: 126

Accomplishments: This week was dedicated entirely to troubleshooting the mutex lock system crash. I researched the problem extensively, which appears to be a known issue with Apple Silicon (M1/M2) chips and the PyTorch/Transformers libraries. I applied several standard fixes, including setting the OBJC_DISABLE_INITIALIZE_FORK_SAFETY=YES and TOKENIZERS_PARALLELISM=false environment variables, but these attempts unfortunately did not solve the root problem.

Challenges: The project is currently at a standstill. The primary challenge is that the Abort trap: 6 crash remains unresolved, and it's a 100% blocker. I cannot proceed with benchmarking the core audio model or begin any application development (Phase 1) until my development environment is stable and capable of running the test scripts.

Plans / Goals for next week: The only goal for next week is to solve this technical blocker. My plan is to escalate the troubleshooting by performing a "Fresh Start": I will completely delete the current Python virtual environment, create a new one from scratch, and move the project folder to a local-only directory on my machine to eliminate any potential conflicts from OneDrive cloud syncing.

SPED Talk Insight (Briefly describe an insight or something interesting you learned from the SPED talks this week): An interesting insight learned from the talk is how the Playwright automated testing framework tackles reliability issues common in User Interface (UI) testing, specifically through a feature called auto-waiting.

Other comments for the instructor:

Week 5: 10/18/25

Overall Status (on-schedule, behind, ahead): Behind

Number of hours worked this week: 8

Total number of hours worked on the project thus far: 28

Number of total hours anticipated at completion: 126

Accomplishments: This was a breakthrough week. I successfully executed the "Fresh Start" plan by moving the project to a local-only folder (away from OneDrive) and building a brand new, clean Python virtual environment. I methodically installed all system dependencies, including FFmpeg via Homebrew, and navigated a complex chain of library errors. The key accomplishment was modifying the benchmark script to bypass torchcodec and use librosa, which completely solved the audio processing roadblock. I have now successfully benchmarked all selected models (bert-base and whisper-base) and confirmed they run with excellent, high-speed performance on my Mac's MPS GPU.

Challenges: The main challenge was identifying the final dependency issue. The torchcodec library's inability to find FFmpeg (even after installation) required a different approach. Pivoting to librosa was a new, unplanned step, but it proved to be the correct solution. This week involved a great deal of "real-world" engineering: debugging environment-specific issues rather than just writing code.

Plans / Goals for next week: With all technical blockers now 100% resolved and the models validated, my goal for next week is to **aggressively start Phase 1: Foundation & Backend**.

- Install FastAPI and Uvicorn.
- Create the main.py file and run the initial "Hello World" server.
- Design the PostgreSQL database schema for the users and journal_entries tables.
- Begin building the API endpoints for user registration and login.

SPED Talk Insight (Briefly describe an insight or something interesting you learned from the SPED talks this week): An interesting insight learned from the Accessibility talk is that accessibility (A11y) should be approached not as a checklist item or an afterthought feature, but as a core design mindset

Other comments for the instructor:

Week 6: 10/25/25

Overall Status (on-schedule, behind, ahead): On schedule

Number of hours worked this week: 9

Total number of hours worked on the project thus far: 37

Number of total hours anticipated at completion: 126

Accomplishments: After successfully resolving all technical blockers last week, I officially began Phase 1: Foundation & Backend. I installed the necessary backend libraries (FastAPI and Uvicorn) into the stable virtual environment. I then created the main.py file for the server and successfully ran the initial "Hello World" API endpoint, confirming the server is operational. I also began the data modeling process by designing the initial database schema for the users and journal_entries tables.

Challenges: The main challenge is no longer technical, but one of time management. Having lost approximately two weeks to the complex environment issues, I need to maintain a consistent and efficient development pace to get back on the original project schedule. My focus has now fully shifted from troubleshooting to pure, focused development.

Plans / Goals for next week: My goal for Week 7 is to build out the complete, core authentication system. This includes:

- Finalizing and creating the **PostgreSQL** tables.
- Establishing the connection between the FastAPI server and the database.

- Implementing the /register endpoint, including password hashing for security.
- Implementing the /login endpoint and returning a basic access token.

SPED Talk Insight (Briefly describe an insight or something interesting you learned from the SPED talks this week): I liked the streamlit presentation and how it makes it so easy to use data science and web development together.

Other comments for the instructor:

Week 7: 11/1/25

Overall Status: On-schedule

Number of hours worked this week: 10

Total number of hours worked on the project thus far: 47

Number of total hours anticipated at completion: 126

Accomplishments:

This week, I successfully created the W07 Technology Prototype as required by the course. I merged the "Hello World" FastAPI server from last week with the successful audio benchmark script. This resulted in a single, working application that loads the Whisper AI model on server startup and provides a live API endpoint (/prototype/run_transcription_test). When accessed, this endpoint correctly loads a sample audio file, transcribes it using the AI model, and returns the transcription result as JSON.

Challenges:

The main challenge was pivoting from my original weekly plan (building the auth system) to focus on creating a demonstrable prototype for the Week 7 assignment. This required refactoring the benchmark scripts into a proper API endpoint and ensuring the model loaded correctly within the server's lifecycle. It was a necessary step to prove the core technology works before I build the full requirements document.

Plans / Goals for next week:

With the prototype complete, the goal for next week is to get back on the main development track (Phase 1). I will focus on building the core authentication system, which includes:

Finalizing and creating the PostgreSQL tables for users and journal_entries.

Connecting the FastAPI server to the database.

Implementing the /register endpoint, including password hashing.

Implementing the /login endpoint and returning a JWT token.

SPED Talk Insight (Briefly describe an insight or something interesting you learned from the SPED talks this week):

Other comments for the instructor:

I'm glad to have the core technology prototype working and submitted. I had to divert my time this week to focus on that deliverable, but now I have a solid foundation and can proceed with building the database and user authentication system as planned.

Week 8: 11/8/25

Overall Status: On-schedule (I am successfully making up for the initial 2-week delay by completing the core backend milestones.)

Number of hours worked this week: 10

Total number of hours worked on the project thus far: 57

Number of total hours anticipated at completion: 126

Accomplishments:

This was a highly productive week. I successfully built the entire database and authentication foundation for the project. I installed all necessary libraries (SQLAlchemy, psycopg2, passlib, python-jose). I created the database.py file to manage the connection and the models.py file to define the User and JournalEntry tables. The FastAPI server is now successfully connected to my PostgreSQL database and automatically created the tables. I have also begun implementing the core authentication logic.

Challenges:

The main challenge was learning and implementing the security best practices for authentication from scratch. This included correctly integrating passlib for password hashing (ensuring no plaintext passwords are ever stored) and understanding the workflow for creating and signing JSON Web Tokens (JWT) for user sessions. While complex, this is a critical part of the application.

Plans / Goals for next week:

My goal for Week 9 is to complete the backend API and begin the frontend. This includes:

Finishing and testing the /register and /login endpoints.

Building the API endpoints for creating, retrieving, and deleting journal entries (the core CRUD operations).

Beginning Phase 2 (Frontend) by setting up the React.js project.

Creating the initial React components for the Login and Registration pages.

SPED Talk Insight: I liked the talk about UX/UI design. For me it was useful since I will start making the frontend of my app, very soon.

Other comments for the instructor:

I'm very pleased with the progress this week. The backend foundation is now solid, which was a major milestone. I'm on track to begin frontend development next week, which should put me in a great position to catch up fully.

Week 9: 11/15/25

Overall Status: On-schedule (This was a very high-productivity week. I completed all my goals from last week and finished the full end-to-end loop of the application.)

Number of hours worked this week: 10

Total number of hours worked on the project thus far: 67

Number of total hours anticipated at completion: 126

Accomplishments:

This week, I successfully completed the entire backend API and built a functional frontend.

Backend: I finished and tested the /register and /login endpoints, and built all core CRUD endpoints: POST /journal-entries (create), GET /journal-entries (read), and DELETE /journal-entries (delete). I also implemented my first stretch goal by integrating the bert-base model for sentiment analysis on all new entries.

Frontend: I set up the React.js project, styled it with a neo-brutalist theme, and installed anime.js and axios.

Integration: I fully connected the frontend to the backend. The app can now: log in, automatically fetch all entries, create a new text entry, and display the returned sentiment from the AI model in real-time.

Challenges:

The main challenge was the frontend-to-backend integration. I had to solve a Cross-Origin Resource Sharing (CORS) error by configuring the FastAPI server. The most difficult bug was with the login form; the backend's OAuth2PasswordRequestForm expected x-www-form-urlencoded data, not JSON, which required a specific URLSearchParams implementation in React.

Plans / Goals for next week:

My primary goal for Week 10 is to implement the main feature of the project: voice-to-text.

Backend: Create a new protected endpoint (e.g., POST /journal-entries/voice) that accepts an audio file upload.

Backend: Wire this new endpoint to the whisper-base model for transcription and the bert-base model for sentiment analysis.

Frontend: Install a library (like react-media-recorder) to capture audio from the user's microphone.

Frontend: Replace the text-based entry form with a "Record" button that sends the audio blob to the new backend endpoint and displays the final transcribed entry.

SPED Talk Insight (Briefly describe an insight or something interesting you learned from the SPED talks this week):

I like the SPED talk about, hosting in AWS, usually for hosting I would use services like render, vercel or Github pages, but with how my project is growing it might not be a bad idea to explore AWS.

Other comments for the instructor:

I'm thrilled with the progress this week. The application now has a full, end-to-end loop working (register, login, create text entry, read entries, see AI sentiment). I am in a great position to tackle the core voice functionality next week, which is the heart of the project.

Week 10: 11/22/25

Overall Status: On-schedule (I have successfully built the frontend-backend bridge and am now deep into developing the core voice features.)

Number of hours worked this week: 10

Total number of hours worked on the project thus far: 77

Number of total hours anticipated at completion: 126

Accomplishments:

This week was focused on refining the frontend application and beginning the core voice integration.

Frontend Polish: I completed the remaining "Must Have" UI features by building a dedicated Registration page (connecting to POST /users) and adding a "Delete" button to journal entries (connecting to DELETE /journal-entries). The text-based CRUD loop is now fully functional and styled with the neo-brutalist theme.

Voice Feature Initiation: I began the implementation of the voice-to-text pipeline. I researched and installed the necessary frontend libraries (like react-media-recorder) to capture browser audio. I also started designing the backend logic to handle multipart/form-data file uploads, which is required to send audio "blobs" from React to FastAPI.

Challenges:

The primary challenge has been moving from sending simple JSON text data to sending binary audio data. Understanding how to correctly package an audio recording as a File or Blob object in JavaScript and receive it correctly in Python without corruption has been a learning curve. Additionally, managing browser microphone permissions adds a new layer of complexity to the user experience.

Plans / Goals for next week:

My goal for Week 11 is to finalize the Voice-to-Text integration and start the advanced AI search features.

Complete Voice: Finish the POST /journal-entries/voice endpoint so it accepts audio, runs Whisper, runs BERT, and saves the result.

Frontend Voice UI: Create the visual "Recording" state in the UI so the user knows when the app is listening.

Search (Stretch Goal): Implement the search bar on the frontend and the filtering logic on the backend (REQ-S2).

SPED Talk Insight (Briefly describe an insight or something interesting you learned from the SPED talks this week):

I really liked the video about Cloud instances, for my app it would be interesting for my project. Since I could have running my project 24/7

Other comments for the instructor:

The project is coming together well. The frontend now looks and feels like a real application. Getting the audio upload pipeline working is the last major hurdle for the core feature set, and I am confident I will have it running by next week.

Week 11: 11/29/25 Overall Status (on-schedule, behind, ahead): Ahead (We completed the main voice feature AND the stretch goals). Number of hours worked this week: 10 Total number of hours worked on the project thus far: 87 Number of total hours anticipated at completion: 126

Accomplishments: This week was a major milestone where all core and advanced features came together.

Voice-to-Text Complete: Finalized the POST /journal-entries/voice endpoint. The app now successfully records audio, uploads it, transcribes it using Whisper, and analyzes sentiment using DistilRoBERTa.

RAG Pipeline (Stretch Goal): Implemented a "Chat with your Journal" feature. Built a vector search system using FAISS and SentenceTransformers that allows users to ask questions and retrieve relevant past entries based on meaning and sentiment.

Search & Filter: Added keyword search and emotion-based filtering (e.g., "Show me all 'Joy' entries").

UX Polish: Redesigned the auth pages with a modern card layout and added anime.js animations for a polished feel.

Challenges:

Audio Format Handling: We faced issues with librosa not recognizing the raw audio blob format from the browser. I had to implement a temporary file handling strategy in the backend to resolve this.

Dependency Conflicts: The anime.js library caused build errors due to version mismatches (v4 vs v3). I had to debug the build process, clear the webpack cache, and downgrade to a stable version to fix the runtime errors.

Plans / Goals for next week: My goal for Week 12 is to finalize the application polish and prepare for the final presentation.

User Profile Enhancements: Finish adding Username, Full Name, and Avatar generation to the registration flow.

End-to-End Testing: Perform a full system test to ensure all features (Voice, Chat, Search) work seamlessly together.

Code Cleanup: Refactor the codebase to remove prototype artifacts.

Presentation Prep: Begin documenting the project for the final submission.

SPED Talk Insight: I liked the Jinja talk, since I am doing this on Python. It could be useful for doing templates.

Other comments for the instructor: I am incredibly excited to report that the project is ahead of schedule. Getting the RAG pipeline working for the "Chat with Journal" feature was a significant achievement that really elevates the project beyond a standard web app. I am now focused on polishing the experience for the final demo.

Week 12: 12/6/25 Overall Status (on-schedule, behind, ahead): Ahead

Number of hours worked this week: 10

Total number of hours worked on the project thus far: 97

Number of total hours anticipated at completion: 126

Accomplishments: This week focused on major feature expansion and user experience refinements.

Notebooks Feature: Implemented a full hierarchical organization system. Users can now create multiple "Notebooks" (e.g., "Dreams", "Work") and organize their journal entries within them. This involved full-stack changes: new database models, API endpoints, and a new Sidebar UI on the frontend.

Transcription Optimization: Switched the AI model to distil-whisper/distil-small.en. This significantly increased transcription speed (approx. 6x faster) while maintaining high accuracy for English speech.

UX Enhancements: Added a real-time "Typewriter" animation effect to the voice recorder, reinforcing the Neo-Brutalist aesthetic.

Bug Fixes: Resolved a critical issue where audio files longer than 30 seconds caused a server crash by correctly handling timestamp tokens in the Whisper model.

Challenges:

Long Audio Handling: We encountered a ValueError with the Hugging Face Transformers library when processing long audio files. It required deep diving into the model's configuration to enable return_timestamps=True for long-form generation.

CSS Specificity: We faced some styling conflicts where the global app styles were overriding the new component styles for the typewriter effect, requiring careful CSS refactoring.

Plans / Goals for next week:

Google Login: Implement OAuth2 authentication to allow users to sign in with their Google accounts.

Final Polish: Conduct a final round of UI cleanup and bug fixing.

Documentation: Prepare the final project documentation and presentation materials.

SPED Talk Insight (Briefly describe an insight or something interesting you learned from the SPED talks this week): I really focused on my presentation on RAGs. Besides of what I already said, I think the biggest applications would be hospitals. Running these models with the patients information, but not with the information coming out of course.

Other comments for the instructor: The addition of the Notebooks feature really completes the "Journaling" aspect of the app, moving it beyond a simple list of entries. The switch to the distilled model has made the app feel much snappier and production-ready. I am confident in the current state of the project as I head into the final weeks.

Week 13: 12/13/25 Overall Status (on-schedule, behind, ahead): Ahead (The project is functionally complete with all stretch goals achieved).

Number of hours worked this week: 10 Total number of hours worked on the project thus far: 107

Number of total hours anticipated at completion: 126

Accomplishments: This was the "Grand Finale" development week. I completed the full UI redesign and implemented all remaining major features.

Frontend Overhaul: Completely refactored the application into a modular architecture with a sleek, dark-themed Neo-Brutalist design.

Advanced Views: Built and enabled four distinct ways to view journal entries: Calendar, Timeline, Photo Gallery, and an interactive Map view with Google Maps integration.

Robust Authentication: Finalized the Google OAuth integration, adding strict validation to prevent duplicate account creation (e.g., blocking "Register" attempts with existing emails).

Rich Media Support: Added full support for image uploads (including automatic HEIC-to-JPEG conversion) and integrated the voice recorder seamlessly into the new UI.

Dynamic Dashboard: Built a "home" screen that tracks user streaks, word counts, and offers daily writing prompts.

State Management & Routing: Managing the navigation state between the new "Library", "Journal", and the four new data views (Map/Calendar/etc.) required careful restructuring of the main App component logic.

Legacy Test Config: I spent time debugging a persistent Jest/Babel configuration issue related to axios and ES Modules to ensure the test suite passed.

Asset Management: Implementing the HEIC image conversion on the client-side was tricky to ensure compatibility across different browsers.

Plans / Goals for next week: My goal for the final week is to wrap up the administrative side of the project.

Record the Final Project Demo Video.

Finalize the GitHub repository (README, cleanup).

Submit the final project deliverables.

SPED Talk Insight (Briefly describe an insight or something interesting you learned from the SPED talks this week): I really liked the talk about Selenium, I am making an API about this game called Elden Ring, and most of the data I get it from scrapping, so this will be interesting to use.

Other comments for the instructor: I have reached the finish line! The application is not only working but looks professional and includes features I didn't originally think I'd have time for (like the Map view and advanced Google Auth flows). I'm ready to present.