eMajor group project report

Team: *Kaleidoscope*
Project title: *A5: Digital Journalling App*
Team members: *Hamnah Rassen, Kai Jones, Muhammad (Ahsan) Mahfuz, Neel Suroop Nair, Rayan Popat, Svilen Dilchev, Yohann Pirbay, Zoya Nasir*
Advisor: *Eamonn Postlethwaite*

1. Introduction

The objective of our digital journaling website is to enhance the personal growth and mental well-being of our users. By offering a platform of consistent, reflective writing, the website aims to cultivate a habit to improve productivity and mental health. Our website is designed to make the process of journaling more accessible and engaging, encouraging users to maintain this beneficial habit through ease of use and reminders. By facilitating creative freedom in their journal entries and providing a structure, we aim to support our users in their journey towards self-improvement and emotional awareness.

Our system is a web-based digital journaling application developed in Django and JavaScript for the backend, and HTML and CSS for the frontend. This technology stack ensures a robust and secure platform, offering a responsive user interface on different devices. Deployed on a PythonAnywhere, it allows users to access their journal anytime, given they have access to the internet. Our technology stack makes digital journaling simple and gives the user a rewarding experience.

2. Features

Feature	Implementation	Purpose
The user is able to create an account.	When the user first opens the webpage, the user is able to create an account by clicking the button called Sign Up.	
The user is able to log in.	When the user has created an account, they are able to login to the journal by clicking Log In.	
The user is able to edit their account details	When logged into the journal, the user can click the icon on the top right of the page. It will have a drop down menu option where the user is able to change their profile or change their password.	
The user is able to create a journal entry.	When logged into the journal, the user is at the landing page and should click the journal button to enter their journal, and there is a button called Create Entry where the user is able to create a new journal entry. The user is able to create, delete, bookmark, and search for their entries.	
The user can create a template prompt for their journal entry.	To create a prompt for the journal entry, the user should go back to the landing page and click on the Templates button.They will be redirected to the Templates page where they are able to create their own template by clicking Create a new template. To use the template, the user should go back to the journal and on the right side of the page, the user can choose their own template and create an entry based on that.	

The user can add multimedia files to their entry.	To add multimedia files, the user can add the multimedia through the add image button on the dashboard.	
The user can export their journal entries as a PDF file/ multiple entries as a PDF file.	To download the current entry that the user is on as a PDF, there is a download button where the user can download the specific entry. To download multiple entries, the user needs to go to the landing page, and click on the Archives button. The user can choose which entries they want to download and the entries will save in one PDF.	
The user can track their mood on the entry by using the mood tracker.	The user can use the slider to decide their mood on the entry. The closer it is to the happy emoji, the better the mood. The mood over time will show in the Progess page in a table, and a graph.	The purpose being user can track their moods over a period of time and be able to look at back at graphs showcasing their changes in mood
The user is motivated to try and journal daily.	The user gains streaks when they journal daily. When they have streaked for a certain amount of time, they gain rewards such as being able to change the colour scheme of the website.	
The user is able to generate inspiration for their entry through an AI feature.	The user should navigate to their journal and on the right side of the page, the user is prompted to enter what they have done today to generate inspiration to write when the user is not able to come up with anything.	
The user is able to look at their achievements.	The user should go to to the landing page and click on the Achievements button to go to the achievements page. Here	

	they can see their streaks and the user is able to create reminders for themselves. The mood table is able to tell the user how their mood was over the days that they have journalled through the background colour of the icons in the table. The user is also able to see their stats such as their mood over time and their word count over time.	

3. Testing

For testing, we relied heavily on automated testing. This was primarily done through QUnit for the JavaScript components and Django’s built in testing framework for the backend. QUnit allowed us to test the interactivity and responsiveness of our web application, ensuring that all JavaScript functionalities worked as intended across different browsers and devices. On the backend, Django's testing tools were utilized to test our models, views, and templates, ensuring that the server-side logic and database interactions were error-free and secure. We have 96% code coverage in our testing, and this can be found through running ‘coverage run manage.py test’ and then ‘coverage report’ in our Django application.

4. Machine learning

Overview of Approaches

For the development of the AI feature in our project, we explored several machine learning techniques, focusing on natural language processing (NLP) and specifically prompt engineering within the OpenAI API framework. Our goal was to generate personalised, inspiring questions based on users' journal entries, aiding in overcoming writer's block.

Considered Approaches

- **Baseline Model (Simple Prompting):** Initially, we utilised a straightforward prompt with the OpenAI API, asking it to generate questions based on input text without any personality or context framing.

- **Advanced Prompt Engineering (Personality-Infused Prompting):** We evolved our approach to include a detailed, personality-infused prompt that acted as a function. This method involved crafting prompts that portrayed JournalGPT as a kind and caring AI, designed to elicit deep thought and emotional exploration from the user.

- **Parameter Optimization:** We experimented with various API parameters like temperature, top_p, frequency_penalty, and presence_penalty to refine the output and ensure creativity while maintaining relevance to the user's input.

Evaluation and Comparison

The evaluation of different algorithms was based on the following criteria:

- **Relevance:** How well the questions related to the user's input.
- **Inspirational Quality:** The ability of the questions to inspire deeper thought or new perspectives.
- **Creativity:** The uniqueness and variety of the questions generated.

Approach	Relevance	Inspirational Quality	Creativity
Baseline Model (Simple Prompting)	Medium	Low	Low
Advanced Prompt Engineering	High	High	High
Parameter Optimization	High	Medium	High

Chosen Algorithm

Based on our evaluation, the Advanced Prompt Engineering approach was chosen for incorporation into the system. However, we did notice some interesting increases in creativity especially when lowering the temperature parameter, so we also included parameter optimization. This method provided the best balance of relevance, inspirational quality, and creativity, essential for encouraging journaling users to explore their thoughts and emotions deeply.

Implementation and Replicability

To replicate our results, follow these steps:

1. Environment Setup: Ensure the OpenAI library is installed and configured to use the specified API version and base URL. Set the OPENAI_API_KEY in your environment.
2. Prompt Engineering: Use the detailed prompt template provided, emphasising the personality and function-like behaviour of the AI to guide its responses.
3. API Parameter Tuning: Experiment with the temperature, top_p, frequency_penalty, and presence_penalty parameters to fine-tune the output. We found a temperature of 0.85 and top_p of 1 to yield the best results.

4. Response Cleaning: Implement the response cleaning functions to remove unwanted characters or substrings, ensuring the output is user-friendly.