# Milestones, Timeline, and Effort Matrix

Coleman Lyski, Tim McCabe, Jeet Shah

## Milestones + Timeline

| **Tasks & Milestones** | **Estimated Start Date** | **Estimated Completion Date** |
|---|---|---|
| 1) Specify which sources we will be leveraging in order to collect textual data (Coleman)<br>- Exploring the use of Facebook, Twitter, Reddit, and other social media platforms and potentially news sites as well<br>- Going to be researching web scrapers to use as an alternative to mainstream APIs for certain sites | October 6th, 2020 | October 17th, 2020 |
| 2) Build web server using Node to access data from specified APIs (Tim)<br>- Research Node frameworks to use for the project, such as Express<br>- Create functions to asynchronously fetch the most popular social media posts about a topic, including the date it was posted and the popularity | October 17th, 2020 | November 15th, 2020 |
| 3) Research existing sentiment analysis libraries to incorporate (Jeet)<br>- Explore usage of the Python NLTK library as well as other natural language packages<br>- Seek out existing APIs related to sentiment analysis to see how others are processing these problems | October 13th, 2020 | October 24th, 2020 |
| 4) Create MVP Python script to perform sentiment analysis (Jeet)<br>- Building out simple logic in a python script for analyzing manually entered data<br>- This logic will be incorporated into the final API | October 24th, 2020 | November 15th, 2020 |
| 5) Build an API around the Python sentiment analysis engine (Coleman)<br>- Build an API that allows the Python Sentiment | November 15th, 2020 | December 15th, 2020 |

| | | |
|---|---|---|
| Analysis engine to communicate with the Node server<br>- Implement batch processing and summarization of data - allow textual data to be streamed in, aggregate results before sending back to Node | | |
| 6) Integrate Node API with sentiment analysis API (Tim)<br>- Update the Node API to send batch queries to the Python script asynchronously as queries to each social media platform complete<br>- Return the results from the Python API calls | January 11th, 2021 | January 25th, 2021 |
| 7) Create React site with search bar and results page (Tim)<br>- Set up the groundwork for the React site<br>- Implement basic functionality with a display of the query results<br>- Start setup in November, integrate Node API once it is complete in January | November 15th, 2020 | January 31st, 2021 |
| 8) Implement graph on results page (Jeet)<br>- Utilizing ChartJS to display the results from the sentiment analysis logic on the home page<br>- Updating graphs based on additional queries<br>- Graphs can be customized by the user as well | February 1st, 2021 | February 21st, 2021 |
| 9) Use a hosting service to productionalize and host APIs (Jeet)<br>- Research AWS API Gateway or another service such as Docker to deploy the API as a standalone<br>- Incorporate documentation for outside users to understand the API | March 1st, 2021 | March 15th, 2021 |
| 10) Implement a SQL database as a cache for common searches (Tim)<br>- Build a database structure that can hold historical data along with search popularity<br>- Create a daily script to remove expired data and searches that have dropped in popularity | February 1st, 2021 | February 21st, 2021 |
| 11) Extend Node API to retrieve popular searches from the database (Coleman)<br>- Create endpoints that can directly grab popular searches and results from the database/cache | March 1st, 2021 | March 15th, 2021 |

| | | |
|---|---|---|
| - Aggregate cache data to create unique charts/trends/graphs | | |
| 12) Implement popular searches API in the front end (Coleman)<br>- Call the Node endpoints that grab data from the database/cache<br>- Display results in charts/graphs/tables/etc. | March 15th, 2021 | March 31st, 2021 |

## Effort Matrix

| **Tasks & Milestones** | **Coleman** | **Tim** | **Jeet** | **Total** |
|---|---|---|---|---|
| 1) Specify which sources we will be leveraging in order to collect textual data (Coleman) | 5 | 2 | 2 | 9 |
| 2) Build web server using Node to access data from specified APIs (Tim) | 10 | 18 | 4 | 32 |
| 3) Research existing sentiment analysis libraries to incorporate (Jeet) | 2 | 2 | 5 | 9 |
| 4) Create MVP Python script to perform sentiment analysis (Jeet) | 6 | 3 | 16 | 25 |
| 5) Build an API around the Python sentiment analysis engine (Coleman) | 18 | 6 | 10 | 34 |
| 6) Integrate Node API with sentiment analysis API (Tim) | 4 | 10 | 2 | 16 |
| 7) Create React site with search bar and results page (Tim) | 4 | 14 | 4 | 22 |
| 8) Implement graph on results page (Jeet) | 10 | 4 | 10 | 24 |
| 9) Use a hosting service to productionalize and host APIs (Jeet) | 2 | 0 | 10 | 12 |
| 10) Implement a SQL database as a cache for common searches (Tim) | 4 | 20 | 0 | 24 |

| | | | | |
|---|---|---|---|---|
| 11) Extend Node API to retrieve popular searches from the database (Coleman) | 10 | 4 | 6 | 20 |
| 12) Implement popular searches API in the front end (Coleman) | 4 | 2 | 2 | 8 |