Work Breakdown Structure (WBS)

1. Project Initialization

1.1 Define Project Scope 1.2 Identify Project Stakeholders 1.3 Create Project Plan 1.4 Set Up Version Control (e.g., Git)

2. Requirements Gathering and Analysis

2.1 Identify Functional Requirements 2.2 Identify Non-Functional Requirements 2.3 Create Software Requirements Specification (SRS) 2.4 Review and Approve SRS

3. System Design

3.1 Design System Architecture 3.2 Design Database Schema 3.3 Create UI/UX Wireframes 3.4 Review and Approve Design

4. Frontend Development

4.1 Set Up React Application 4.2 Create Sentiment Form Component 4.3 Create Sentiment List Component 4.4 Integrate API Calls 4.5 Implement Styling (CSS)

5. Backend Development

5.1 Set Up Express Server 5.2 Create API Endpoints 5.2.1 POST /analyze (Analyze Sentiment) 5.2.2 GET / (Fetch Sentiments) 5.2.3 DELETE /

(Delete Sentiment) 5.3 Integrate Python Script for Sentiment Analysis 5.4 Implement Error Handling

6. Database Setup

6.1 Install and Configure MongoDB 6.2 Create Sentiment Model 6.3 Connect Backend to MongoDB

7. Machine Learning Integration

7.1 Create Python Script for Sentiment Analysis 7.2 Train Sentiment Analysis Model (if applicable) 7.3 Integrate Python Script with Backend

8. Testing

8.1 Unit Testing 8.2 Integration Testing 8.3 End-to-End Testing 8.4 Bug Fixing and Optimization

9. Deployment

9.1 Set Up Hosting for Backend (e.g., Heroku, AWS) 9.2 Set Up Hosting for Frontend (e.g., Vercel, Netlify) 9.3 Deploy MongoDB Database (e.g., MongoDB Atlas) 9.4 Perform Deployment Testing

10. Documentation

10.1 Create User Guide 10.2 Create Developer Guide 10.3 Document API Endpoints 10.4 Document Codebase

11. Project Review and Closure

11.1 Conduct Final Review 11.2 Obtain Stakeholder Sign-Off 11.3 Archive Project Documents 11.4 Conduct Post-Implementation Review

Task Descriptions

1. Project Initialization

- Define project objectives, scope, and deliverables.
- Identify key stakeholders and their roles.
- Create a detailed project plan outlining timelines and milestones.
- Set up version control using Git to manage code changes.

2. Requirements Gathering and Analysis

- Gather functional requirements (e.g., sentiment analysis, data display).
- Identify non-functional requirements (e.g., performance, security).
- Document requirements in a Software Requirements Specification (SRS).
- Review and get approval from stakeholders.

3. System Design

- Design the overall system architecture, including frontend, backend, and database components.
- Design the database schema for storing sentiment data.
- Create wireframes for the user interface to visualize the frontend design.
- Review and approve the design with stakeholders.

4. Frontend Development

- Set up the React application using create-react-app or a similar tool.
- Create components for the sentiment form and list.
- Integrate API calls to fetch and submit data.
- Style the components using CSS or a CSS framework.

5. Backend Development

- Set up an Express server for handling API requests.
- Create API endpoints for analyzing sentiment, fetching sentiments, and deleting sentiments.
- Integrate the Python script for sentiment analysis.
- Implement error handling to manage potential issues.

6. Database Setup

- Install and configure MongoDB locally or use a cloud service like MongoDB Atlas.
- Create a Mongoose model for the sentiment data.
- Connect the backend server to the MongoDB database.

7. Machine Learning Integration

- Develop a Python script for sentiment analysis using a library like TextBlob or an ML model.

- If applicable, train the sentiment analysis model using a dataset.
- Integrate the Python script with the backend to process sentiment analysis requests.

8. Testing

- Conduct unit testing for individual components and functions.
- Perform integration testing to ensure different parts of the system work together.
- Conduct end-to-end testing to verify the entire application flow.
- Fix any identified bugs and optimize the code.

9. Deployment

- Set up hosting for the backend on platforms like Heroku or AWS.
- Set up hosting for the frontend on platforms like Vercel or Netlify.
- Deploy the MongoDB database using a service like MongoDB Atlas.
- Perform testing on the deployed application to ensure it works as expected.

10. Documentation

- Create a user guide explaining how to use the application.
- Create a developer guide with setup instructions and code explanations.
- Document all API endpoints with details on their usage.
- Document the codebase, including important functions and classes.

11. Project Review and Closure

- Conduct a final review of the project with stakeholders.
- Obtain sign-off from stakeholders indicating project acceptance.
- Archive all project documents for future reference.
- Conduct a post-implementation review to identify lessons learned.