I. Feasibility Study (Key Aspects)

Technical Feasibility

- **Tools & Frameworks**
    - Data Fetching: Pandas, Python (to process historical election-related tweets downloaded from the internet).
    - NLP: NLTK/ TextBlob for sentiment analysis (cleaning tweets, classifying sentiments).
    - Machine Learning: Scikit-learn/TensorFlow for sentiment classification models.
    - **Web Interface:**
        - Frontend: React, HTML/CSS/JS.
        - Backend: Django/Flask.
        - Database: MongoDB/PostgreSQL.
        - Hosting: AWS/Google Cloud for scalability.
- **System Architecture**
    - Data Collection Layer: Historical tweets dataset.
    - Processing Layer: Sentiment analysis (positive/neutral/negative) and trend analysis.
    - Presentation Layer: Dashboard for visualizing sentiment trends.
- **Scalability**
    - Cloud infrastructure (AWS EC2, auto-scaling) to handle spikes in tweet volume during election cycles.
- **Integration**
    - Pipeline for processing tweets, analysing sentiment, and updating dashboards.

Schedule and Milestones

1. Phase 1: Requirement gathering and system design (election focus).
2. Phase 2: Data collection pipeline setup (using historical election keywords/hashtags).
3. Phase 3: Sentiment analysis model development (parties/figures).
4. Phase 4: Trend analysis and dashboard integration.
5. Phase 5: Web application development.
6. Phase 6: Testing, refinement, deployment.

Key Deliverables

- Jan 7: Planning and design of the project.
- Jan 14: Research.
- Jan 21: Collection of datasets.
- Jan 28: Preprocessing of dataset.
- Feb 8: Sentiment analysis.
- March 4: Dashboard prototype.
- March 18: Integration of analysis onto dashboard.
- March 26: Testing.
- April 8: Deployment.

Quality Feasibility

- **Testing**
    - Unit/Integration Testing: Validate tweet processing and sentiment accuracy.
    - Performance Testing: Handle large volumes of historical tweets.
- **Accuracy**
    - Validate sentiment model against labelled political tweet datasets.
- **Market Feasibility**
    - Demand: High interest from political analysts, campaign teams, and media.
    - Target Users: Journalists, researchers, and policymakers tracking election sentiment**.**

II. Project Requirements

Customer Requirements

- Sentiment analysis on historical tweets about election parties, candidates, and keywords.
- Sentiment classification (positive/neutral/negative).
- Dashboard with interactive visualizations (trends, geographic distribution).

Analysis

- Focus on scalability for election-driven data spikes.
- Secure API key management for accessing datasets.

III. Software Requirements Specification (SRS)

a. Introduction

- **Purpose**
    - Analyse Twitter sentiment toward election parties and high-profile figures, providing insights via a dashboard.
- **Scope**
    - Sentiment Analysis: Classify tweets by political entity.
    - Trend Visualization: Track sentiment shifts during campaigns.
    - User Interface: Accessible dashboard for insights.

b. Overall Description

- **Product Features**
    1. Tweet Collection: Uses historical datasets downloaded from the internet.
    2. Sentiment Analysis: Tracks sentiment of parties and figures.
    3. Interactive Dashboard: Visualizes trends, sentiment distribution, and top topics.
- **User Classes**
    - Political Analysts, Campaign Managers, Journalists.

c. System Features

- **Functional Requirements**
    1. Process historical tweets from datasets.
    2. Classify sentiment for parties/figures using NLP.
    3. Display trends in a dashboard.
- **Non-Functional Requirements**
    - Performance: Handle large-scale tweet datasets.
    - Security: Encrypt data storage.
    - Compatibility: Desktop browsers (Windows/macOS).

d. System Attributes

- Availability: 24/7 cloud hosting with redundancy.
- Scalability: Auto-scaling for large dataset processing.
- Backup/Recovery: Daily backups of tweet data and user settings.

IV. SDLC Model: Incremental Model

Justification

- Modules (data collection, sentiment analysis, dashboard) developed incrementally.
- Flexibility to adapt to changing election dynamics or dataset updates.
- User feedback at each phase (e.g., dashboard design).

Team

1. **Ezekiel Tep - 2201072**
2. **Khadka Bahadur Baniya - 2201107**
3. **Deepak Kumar - 2201061**
4. **Manik Kumar - 2201119**