Design Plan: Cyberbullying Detection in Social Media Platform

1. Executive Summary

This document delineates the design specifications and implementation trajectory for an online platform engineered to detect instances of cyberbullying. The primary objective of the design phase was to formulate a comprehensive blueprint to govern the coding phase, which is scheduled to commence in the subsequent week. The platform integrates a Bidirectional Long Short-Term Memory (BiLSTM) model to facilitate toxicity detection, complemented by an intuitive interface designed for real-time analysis and reputation management.

2. System Architecture

2.1 Overview

The system adopts a client-server architectural paradigm wherein the frontend facilitates user interaction, while the backend orchestrates data processing via a machine learning model.

2.2 Data Flow

1. **Input Acquisition:** The user submits textual content through the frontend interface.

2. Data Transmission: The data is transmitted to the Flask backend via designated API endpoints.

3. Algorithmic Processing: The backend performs pre-processing on the text before inputting it into the BiLSTM model.

4. Analytical Evaluation: The model computes toxicity levels and derives a reputation score based on the analysis.

5. Automated Intervention: Should the calculated score meet specific thresholds, the automated blocking mechanism logic is instantaneously triggered.

6. Result Dissemination: The analytical results are transmitted back to the frontend for presentation to the user.

3. Technology Stack

Frontend:

- **HTML5:** Utilized for defining the structural semantics of the web pages.

- **CSS3:** Employed for styling, layout configuration, and ensuring visual aesthetics characterized by clarity and modernity.
- **JavaScript:** Implemented to manage interactive elements, input validation, and asynchronous API communication.

Backend:

- **Python (Flask):** A lightweight web framework selected to manage API requests and model inference efficiently

Machine Learning:

- **Bi-LSTM Model:** A deep learning architecture specifically selected for its efficacy in sequence processing and natural language understanding within the context of toxicity detection.

4. UI/UX Design Strategy

The interface design strategy prioritizes usability, organization, and aesthetic coherence. The objective is to ensure accessibility for general users while effectively communicating complex data, such as toxicity metrics.

4.1 Sitemap & Page Layouts

A streamlined text area designed for the input of messages or comments requiring analysis. Minimalist submission controls to enhance user efficiency.

4.2 Toxicity Analysis Dashboard

A visual decomposition of the analysis results. Distinct indicators (e.g., chromatic badges) to signify whether content is classified as safe, toxic, or severe.

4.3 Reputation Score Visualization

A dashboard view displaying the calculated reputation score, derived from historical data or current input. Graphical representations (e.g., progress bars or gauges) provided for immediate visual feedback.

4.4 Automated Blocking Mechanism

An interface for monitoring the status of automated actions (e.g., "User Blocked" or "Content Flagged"). Configuration options or notifications pertaining to the specific blocking criteria.

5. Implementation Roadmap

Phase 1: Environment configuration (Flask framework and frontend structural setup).

Phase 2: Development of the BiLSTM integration pipeline.

Phase 3: Construction of user interface modules (Input, Results, Score, and Blocking interfaces).

Phase 4: Integration of the frontend with the backend API.

Phase 5: System validation and iterative refinement.