**Project Proposal**

**Title:**

**Sentiment Analysis and Custom Reply Generation for Product Reviews with Role-Based Access Using Flask**

**Group Members:**

* Member 1: (Name)
* Member 2: (Name)
* Member 3: (Name)

**Supervisor:**

[Supervisor's Name]

**Project Overview:**

This project aims to develop a web application using Flask that performs sentiment analysis on product reviews. The system will have two types of users: **Admin** and **Sentiment User**. Admins will have access to the overall dashboard, review statistics, and visualizations. Sentiment Users will be able to input text reviews for analysis and generate custom replies based on the sentiment.

**Project Goals:**

1. **Sentiment Analysis**:
   * Build a Flask-based app that performs sentiment analysis on user-submitted product reviews.
   * Classify sentiment into categories such as Positive, Neutral, or Negative.
2. **Role-Based Access**:
   * Implement a system with two types of users: **Admin** and **Sentiment User**.
     + **Admin** can view overall sentiment statistics, generated graphs, and a dashboard of reviews.
     + **Sentiment User** can submit reviews for sentiment analysis and generate custom replies.
3. **Custom Reply Generation for Reviews**:
   * Generate personalized responses for specific reviews based on sentiment and keywords detected in the review text.

**Proposed Features:**

1. **Sentiment Analysis**:
   * **Description**: The system will analyze user-entered product reviews and classify them as Positive, Neutral, or Negative using Natural Language Processing (NLP) models.
   * **Impact**: Allows businesses or admins to understand customer feedback and improve their services or products based on the sentiments expressed.
2. **Role-Based Access (Admin and Sentiment User)**:
   * **Admin**:
     + Can view the overall dashboard showing:
       - Sentiment trends over time (e.g., line charts).
       - Distribution of sentiments (e.g., pie charts).
       - Word clouds showing common keywords in reviews.
     + Access to the entire review history, including user inputs and custom replies generated.
   * **Sentiment User**:
     + Can submit text reviews for sentiment analysis.
     + After sentiment classification, they can generate and view custom replies based on the review's sentiment and certain keywords.
3. **Custom Reply Generation for Reviews**:
   * **Description**: After analysing a review, the system will generate a custom reply based on the detected sentiment and specific keywords or phrases. For example:
     + A negative review mentioning "price" might generate: "We're sorry to hear you found the price too high. We'll forward this feedback to our team."
     + A positive review mentioning "quality" might generate: "Thank you! We're glad you loved the product quality."
   * **Impact**: Enables businesses to engage with customers by providing relevant responses, simulating an interactive customer service experience.
4. **Data Visualization**:
   * **Description**: Provide the admin with a dashboard that visualizes sentiment data using charts (pie charts for sentiment distribution, line charts for trends) and word clouds to showcase common keywords in reviews.
   * **Impact**: Helps admins and businesses make data-driven decisions based on the insights gathered from customer reviews.

**Tools and Technologies:**

* **Backend**: Flask, Python
* **Database**: SQLite or PostgreSQL (for storing reviews, custom replies, and user data)
* **Machine Learning**: Pre-trained NLP models (e.g., BERT) for sentiment analysis and keyword detection
* **Frontend**: HTML/CSS, Bootstrap, JavaScript (for building interactive user interfaces)
* **Additional Libraries**:
  + Flask-RBAC for managing role-based access.
  + SQLAlchemy for database management.

**Division of Work:**

* **Member 1**: Backend development (Flask API, sentiment analysis, role-based access system).
* **Member 2**: Frontend development (UI for sentiment submission and custom reply generation for Sentiment User).
* **Member 3**: Dashboard and data visualization for Admin, testing, and deployment.

**Expected Outcomes:**

* A Flask-based web application that:
  + Allows Admin users to access a dashboard with sentiment data visualizations.
  + Lets Sentiment Users input product reviews for sentiment analysis and generate custom replies.
  + Provides a clear distinction between user roles and functionalities.
  + Visualizes sentiment data for business analysis.

**Project Timeline:**

* **Phase 1**: Requirement gathering and system design.
* **Phase 2**: Backend development (sentiment analysis, role-based access).
* **Phase 3**: Frontend development (UI for Sentiment User and Admin dashboards).
* **Phase 4**: Integration of features, testing, and deployment.

**Conclusion:**

This project offers a comprehensive solution for analyzing and responding to product reviews with role-based access control. The system provides detailed sentiment insights through a dashboard for Admins while enabling Sentiment Users to interact with reviews and generate relevant replies. This project justifies the effort for a three-member group with its diverse functionality, combining machine learning, web development, and user role management.