**Day 1: Planning and Setup**

1. **Prioritize Features:**
   * Choose 3-4 *core features* that are feasible to implement and impactful:
     1. **Trend Forecasting**
     2. **Sentiment Analysis**
     3. **Content Optimization**
     4. **Custom Dashboards** (integrating outputs from the above features)
2. **Divide Roles:**
   * **Member 1**: Backend development and model integration.
   * **Member 2**: Data preprocessing and AI model implementation.
   * **Member 3**: Frontend development and dashboard design.
   * **Member 4**: API research, mock data creation, and documentation.
3. **Set Up the Development Environment**:
   * Install required tools/libraries (Python, Flask, React.js, etc.).
   * Agree on code-sharing platforms (e.g., GitHub).

**Day 2: Backend and Data Collection**

1. **Backend Setup:**
   * Build a simple Flask or FastAPI server.
   * Create endpoints for each feature (e.g., /predict\_trend, /analyze\_sentiment).
2. **Data Collection:**
   * Use public datasets or simulate small datasets:
     + **Trend Forecasting**: Time-series data (likes, comments, shares).
     + **Sentiment Analysis**: Comments or reviews from open datasets like Sentiment140.
3. **Mock Data Generation:**
   * If real data is unavailable, generate plausible mock data using tools like Excel or Python.

**Day 3: AI Model Implementation**

1. **Trend Forecasting:**
   * Use Facebook’s Prophet library for quick results.
   * Train the model on simulated engagement data.
2. **Sentiment Analysis:**
   * Fine-tune a small pre-trained model like distilBERT using HuggingFace.
   * Alternatively, use TextBlob or NLTK for faster (but less sophisticated) sentiment classification.
3. **Content Optimization:**
   * Build simple rule-based recommendations for posting times based on trends in data.
   * Use word frequency analysis for hashtag and caption suggestions.

**Day 4: Frontend and Visualization**

1. **Frontend UI:**
   * Use **Streamlit** for faster prototyping or build basic React.js components.
   * Create a clean, simple interface with sections for:
     + Dashboard (summary of insights).
     + Visuals for trends and sentiment.
2. **Visualization:**
   * Use Plotly or Matplotlib for interactive charts.
   * Create graphs for:
     + Engagement trends over time.
     + Sentiment distribution (pie chart or bar graph).

**Day 5: Integration**

1. **Connect Backend and Frontend:**
   * Test API endpoints with sample data using tools like Postman.
   * Fetch and display data on the frontend.
2. **Custom Dashboard:**
   * Combine outputs from all features into a single view.
   * Add basic filters (e.g., date range).

**Day 6: Testing and Refinement**

1. **Test the Prototype:**
   * Check each feature’s functionality independently.
   * Test the integration for seamless data flow.
2. **Feedback and Iteration:**
   * Ask peers or mentors for feedback.
   * Refine based on usability and visual appeal.

**Day 7: Presentation Preparation**

1. **Create the Presentation:**
   * Explain the problem, solution, and implemented features.
   * Include screenshots or a live demo of the prototype.
2. **Rehearse:**
   * Practice the demo and prepare for possible questions.

**Tips for Success**

* **Simplify the Features**: Focus on getting basic functionality working rather than perfecting details.
* **Collaborate Efficiently**: Use tools like Trello for task tracking and Slack/WhatsApp for communication.
* **Showcase Impact**: In the presentation, emphasize the real-world utility of your prototype.

If you need help with a specific feature, coding assistance, or the presentation, let me know!