README.md

~~~markdown
# Citizens AI - Intelligent Citizen Engagement Platform

Citizens AI is a platform that helps governments and organizations collect, analyze, and respond to citizen feedback using AI-powered tools such as sentiment analysis, categorization, and ranking.

## Features

- Intelligent sentiment analysis
- Feedback categorization
- Scoring and ranking of feedback
- Easily extensible APIs

## Installation

```bash
git clone https://github.com/yourusername/citizens-ai.git
cd citizens-ai
pip install -r requirements.txt
```

## Usage

```bash
python main.py
```

## License

MIT License
~~~

requirements.txt

```
textblob
flask
```

citizens_ai/sentiment.py

```
from textblob import TextBlob

def analyze_sentiment(text):
    blob = TextBlob(text)
    return blob.sentiment.polarity  # -1 (negative) to 1 (positive)
```

citizens_ai/feedback_handler.py

```
def parse_feedback(data):
    return [entry["text"] for entry in data if "text" in entry]
```

citizens_ai/ai_engine.py

```
from .sentiment import analyze_sentiment

def rank_feedback(feedback_list):
    scored = [(fb, analyze_sentiment(fb)) for fb in feedback_list]
    return sorted(scored, key=lambda x: x[1], reverse=True)
```

data/sample_feedback.json

```json
[
    {"text": "We need better street lighting."},
    {"text": "The new park is amazing, thank you!"},
    {"text": "Trash collection has been late for weeks."}
]
```

main.py

```
import json
from citizens_ai.feedback_handler import parse_feedback
from citizens_ai.ai_engine import rank_feedback

def main():
    with open("data/sample_feedback.json") as f:
        data = json.load(f)

    feedback_list = parse_feedback(data)
    ranked = rank_feedback(feedback_list)

    print("Ranked Feedback:")
    for fb, score in ranked:
        print(f"[{score:.2f}] {fb}")

if __name__ == "__main__":
    main()
```

.gitignore

```
__pycache__/
*.pyc
.env
```