Project Synopsis: HR Connect – Mood Analytics Dashboard with Chatbot

# 1. Project Overview

The HR Connect application is a modern HR-focused analytics and performance management tool built with Angular 19 (frontend) and FastAPI (backend). It combines performance analytics dashboards, mood tracking visualizations, and an AI chatbot assistant to enhance employee engagement, provide insights, and streamline HR workflows.

# 2. Key Features

## Frontend (Angular 19 + PrimeNG + Angular Material)

- Mood Graphs Module: Daily mood distribution pie chart, Top mood distribution pie chart, Range-based mood analytics with ApexCharts.  
- Performance Management: Employee KPIs, goals, and reviews with reactive forms.  
- Analytics Dashboard: HR analytics trends, attrition, and mood insights.  
- Chatbot Component: Floating chatbot bubble with file upload and HR query resolution.  
- Global UX: Responsive layouts, JWT interceptors, spinners, and error handling.

## Backend (FastAPI + PostgreSQL/SQLite)

- Onboarding & User Management APIs.  
- Mood Analytics API: Provides buckets for chart visualizations.  
- Chatbot API: Integrates with OpenAI/Ollama via /ask endpoint.  
- Performance Data APIs: Employee KPIs and analytics.  
- Vector Database for document embeddings and chatbot context.

# 3. Tech Stack

|  |  |
| --- | --- |
| Layer | Technology |
| Frontend | Angular 19, Angular Material, PrimeNG, ng-apexcharts |
| Backend | FastAPI (Python), SQLAlchemy ORM, Pydantic v2 |
| Database | SQLite |
| AI/Chatbot | OpenAI API or Ollama (local LLM), FAISS/Chroma DB, Azure openaI |
|  |  |

# 4. Architecture

The system follows a client-server model:  
- Angular SPA fetches analytics and chatbot UI.  
- FastAPI backend handles analytics and onboarding.  
- Vector DB stores embeddings.  
- External/Local LLM processes chatbot queries.

# 5. Key Modules & Components

Frontend Components:  
- MoodGraphComponent: Visualizes mood analytics.  
- PerformanceModule: Employee goals, KPIs.  
- AnalyticsDashboard: Combines HR + mood insights.  
- ChatbotComponent: Floating assistant.  
- Shared: Interceptors, error handlers.

Backend Modules:  
- onboarding.py: Onboarding endpoints.  
- mood.py: Mood analytics API.  
- chat.py: Chatbot integration.  
- models.py: ORM models.  
- rag\_service.py: Retrieval augmented generation service.

# 6. Workflow

1. Employee moods submitted daily → stored in DB.  
2. API aggregates moods → Angular visualizes charts.  
3. HR monitors top moods & trends.  
4. Chatbot answers HR queries.  
5. Dashboards show analytics + conversational AI.

# 7. Challenges Solved

- NaN/null safety in charts with fallback values.  
- Legend overlap fixed with responsive configs.  
- Dynamic reactive forms with PrimeNG.  
- Privacy-first chatbot with local LLM.  
- Angular 19 dependency alignment with Node 18.

# 8. Future Enhancements

- Sentiment analysis integration.  
- Role-based dashboards.  
- Export charts (PDF/Excel).  
- Slack/Teams chatbot integration.  
- Predictive HR analytics with ML.

# 9. Conclusion

This project delivers a scalable HR analytics and chatbot platform that empowers HR teams to track employee mood, manage performance, and engage employees through conversational AI. The combination of Angular 19 frontend, FastAPI backend, and AI integration makes it modern, secure, and extensible.