

Architecture Design Document

1. Introduction

This document provides the architectural design of a Generative AI-driven recommendation system that pe

1.1 Purpose

The system aims to:

- Collect and analyze customer data for hyper-personalization.
- Use DistilBERT for sentiment analysis of user-interaction content in social media.
- Generate AI-powered personalized recommendations.
- Provide actionable insights for businesses to optimize customer engagement.

2. System Overview

The system consists of:

- Frontend (React.js)
- Backend (Spring Boot)
- AI Engine (DistilBERT & Recommendation System)
- Database & Storage (MongoDB)

3. System Architecture Diagram

(See attached diagram)

4. Component Design

- Frontend (React.js): Handles UI, user authentication, and recommendations.
- Backend (Spring Boot): Manages business logic, API endpoints, and AI integration.
- AI Engine: Uses DistilBERT hugging face transformers for sentiment analysis and collaborative filtering fo
- Database Layer: MongoDB for unstructured data.

5. Data Flow Diagram

1. User logs in to the React.js app.
2. Backend fetches user profile & history from the database.
3. AI Engine processes user data & sentiment analysis.
4. Recommendation System generates personalized suggestions.
5. User receives recommendations on the dashboard.
6. User feedback improves AI model training over time.

6. API Design

Authentication APIs:

- POST /api/auth/login - User login
- POST /api/auth/signup - User registration
- GET /api/auth/profile - Fetch user profile

Recommendation APIs:

- GET /api/recommendations - Get personalized recommendations
- POST /api/feedback - Submit feedback

Sentiment Analysis API:

- POST /api/sentiment - Analyze text sentiment

7. Implementation Plan

Phase 1: Development & Testing

- Implement frontend UI and authentication.
- Develop backend APIs for user management & data storage.
- Train DistilBERT for sentiment analysis.

Phase 2: Integration & Model Deployment

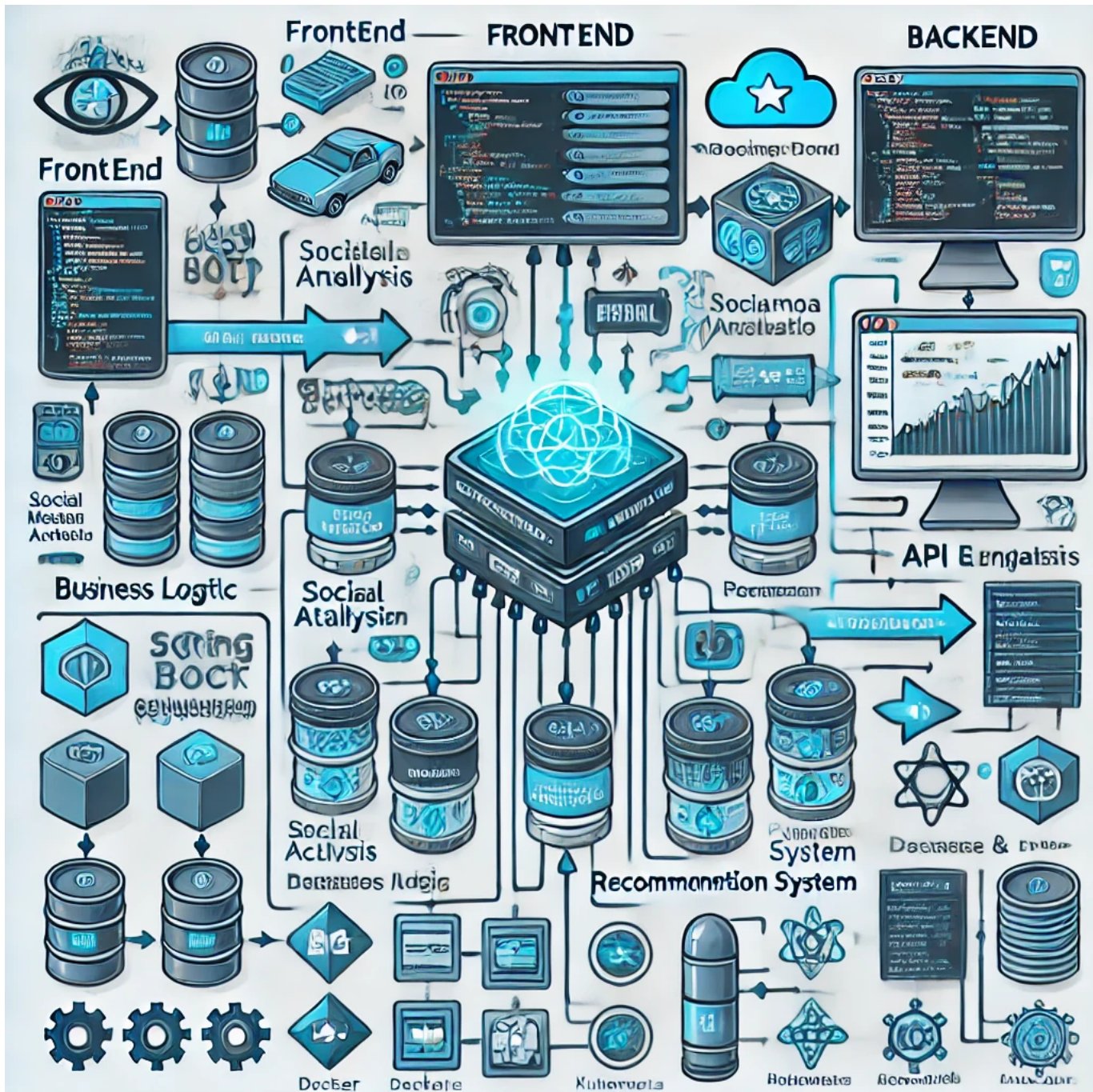
- Integrate AI model with backend.
- Deploy AI service using FastAPI & TensorFlow Serving.
- Optimize recommendation algorithms

8. Conclusion

This system enables real-time AI-powered hyper-personalization using:

- React.js for an interactive frontend.
- Spring Boot for a scalable backend.
- DistilBERT for sentiment analysis
- AI-driven recommendations for personalized content.

System Architecture Diagram



Data Flow Diagram

