

RFQ MANAGEMENT (team-anurag6569201)

Architecture Design:

- Frontend : React.js , bootstrap
- Backend : Django, DRF
- Database : MySQL
- AI : openai (gpt-4o), Langchain, ChromaDB, RAG
- DevOps : Azure (backend), firebase(frontend), database (railway),Github,Github Actions

Development Workflow:

1. Building APIs using Django DRF for:
 - RFQ Management: Creation, updating, and sharing.
 - Bidding System: Submitting, tracking, and managing bids.
 - Chat System: E2EE (AES-256 or HMAC).
2. Frontend Development
 - Designing responsive, user-friendly UI for:
 - RFQ creation and sharing
 - Bidding interface
 - Chat functionality
 - Dashboards for RFQs, bids, and analytics
3. AI/ML Integration
 - Implementation of AI summarization for RFQs and bids.
 - Implementation of vendors match according to requirements.
 - Building a vendor insights model with an AI-driven "Bid Score."
 - Enable Chat with RFQ Documents using RAG and LangChain.
 - Building rfq through BOQs
4. DevOps & Deployment
 - Based on micro-service architecture of Django using docker, azure, railway and firebase.
 - Host:
 - Frontend on Firebase.
 - Backend on Azure.

Project aligns with the 2 themes

1. Theme 2: AI Integration
2. Theme 3: Privacy and Data Security

Features:

- | | |
|----------------|----------------------------|
| 1. Chatting | 6. summarisation |
| 2. Bidding | 7. ai score of bid |
| 3. Rating | 8. bid summarisation |
| 4. RFQ sharing | 9. chat with rfq and files |
| 5. dashboards | 10. vendors matchmaking |