**BUSINESS REQUIREMENTS DOCUMENT (BRD)Project Title: AI-Driven Interview and Scoring System**

**Prepared by: Wardah Khan  
Client/Stakeholder: KSBL (Karachi School of Business & Leadership)  
Date: July 2025**

**1. Project Overview**

The project aims to build an AI-powered interview and scoring system tailored for HR and academic admissions. It automates structured interviews using a chatbot and evaluates candidate responses using NLP and LLM-based scoring. The system includes a React dashboard for administrators to view interview analytics and scoring insights.

**2. Objectives**

* Automate candidate interviews using a conversational chatbot
* Evaluate responses using NLP (RoBERTa/BERT)
* Provide transparent and consistent scoring
* Enable HR/academic staff to view candidate responses and scores via a dashboard
* Ensure scalability and professionalism for future client adoption

**3. Key Features**

**Core Functionalities**

|  |  |
| --- | --- |
| Feature | Description |
| User Login | Secure login system for candidates and admins |
| Chatbot Interview | AI chatbot that asks predefined HR/academic questions |
| Voice Input (Optional) | Allow user to speak responses (STT integration - optional) |
| Camera Integration (Optional) | Collect candidate footage during interview |
| LLM Scoring Engine | RoBERTa-based model to evaluate subjective answers |
| Admin Dashboard | React-based UI to monitor interviews and view scoring |
| Data Storage | PostgreSQL database to store candidates, questions, and scores |

**4. Tech Stack**

|  |  |
| --- | --- |
| Component | Technology |
| LLM | Hugging Face (RoBERTa/BERT) |
| Backend | Python + FastAPI + PostgreSQL |
| Frontend | React.js + TailwindCSS + Recharts |
| Hosting | (Optional) Render/Vercel/HuggingFace Spaces |

**5. User Roles**

* Candidate: Takes the interview via chatbot and submits responses
* Admin (HR/Faculty): Views responses, scores, and can filter/sort performance
* Super Admin (Optional): Manages user accounts, questions, and system configs

**6. Assumptions**

* Project will be developed over 8 weekends
* Basic scoring logic and rubric is ready
* Voice and camera integration are optional enhancements, not MVP requirements
* React is newly being learned, so extra time will be allocated to that part

**7. Constraints**

* Weekend-only availability
* Limited real-world response data initially (bootstrapped from surveys)

**8. Risks**

|  |  |
| --- | --- |
| Risk | Mitigation Strategy |
| Low training data | Start with dummy + survey data, use rubrics |
| Technical integration difficulty | Start with CLI MVP, then move to API + UI |
| React learning curve | Allocate dedicated time in Week 5 |
| User privacy | Remove identifiers, keep data usage transparent |

**PROJECT TIMELINE & ACTION PLAN (8 WEEKS – WEEKENDS ONLY)**

**Duration: July 5 – August 31**

**Workdays: Saturdays & Sundays (16 total)**

**Timeline:**

**WEEK 1 (July 5–6)**

* Goal: Project setup + basic CLI chatbot
* Finalize questions for HR & Academic
* Build Python-based CLI chatbot
* Store responses in JSON
* Create separate question sets (HR, Academic)

**WEEK 2 (July 12–13)**

* Goal: Start backend (FastAPI)
* Set up FastAPI project
* Add route to fetch questions based on role
* Add route to save user responses
* Connect CLI chatbot to FastAPI (optional)

**WEEK 3 (July 19–20)**

* Goal: Build scoring engine (RoBERTa)
* Prepare labeled data from survey or rubric
* Implement scoring logic using HuggingFace
* Create score\_response() function
* Return scores from API

**WEEK 4 (July 26–27)**

* Goal: Connect chatbot to scoring engine
* Integrate scoring into chatbot flow
* Test scoring on sample interviews
* Store scores with responses in PostgreSQL

**WEEK 5 (Aug 2–3)**

* Goal: Learn & set up React dashboard
* Follow basic React tutorial
* Set up React project
* Build dashboard skeleton: sidebar + table + card view
* Display dummy candidate data

**WEEK 6 (Aug 9–10)**

* Goal: Build live dashboard
* Connect React to FastAPI (API call)
* Show candidate answers and scores
* Add filters (HR/Academic)
* Add basic chart (bar or radar)

**WEEK 7 (Aug 16–17)**

* Goal: Polish UI + scoring controls
* Use TailwindCSS or ShadCN for styling
* Add score override/edit option (optional)
* Add session logs and timestamps
* Handle errors and validation

**WEEK 8 (Aug 23–24 or 30–31)**

* Goal: Final testing + demo prep
* Test HR & Academic flow end-to-end
* Fix bugs and polish UI
* Record a demo video
* Write user guide / readme
* Prepare for Capstone-II or client pitch

**DELIVERABLES CHECKLIST:**

|  |  |
| --- | --- |
| Deliverable | Status |
| Chatbot (HR & Academic) |  |
| FastAPI backend |  |
| Scoring engine (LLM) |  |
| PostgreSQL integration |  |
| React dashboard |  |
| Clean, professional UI |  |
| Demo-ready project |  |