**Project Synopsis**

**AI-Powered Resume Screener**

# Problem Statement:

Recruitment processes are often overwhelmed by the large volume of resumes received for each job opening. Manually screening resumes is time-consuming, prone to human bias, and often inconsistent in evaluating candidate-job fit. HR professionals spend a significant amount of time shortlisting candidates, which delays hiring decisions and increases organizational costs.

# Proposed Solution:

The AI-Powered Resume Screener is a **backend system built with Spring Boot** that leverages **Natural Language Processing (NLP) and Machine Learning models** to automate resume screening. The system extracts text from uploaded resumes (PDF formats), compares the extracted content against a provided job description, and assigns a **relevance score** to each candidate. A **dashboard built in React** provides recruiters with ranked candidate lists, reducing manual effort and improving decision-making efficiency.

# Objectives:

* To **automate the resume shortlisting process**, saving time and reducing costs.
* To ensure **fair and consistent candidate evaluation** based on job requirements.
* To enable HR professionals to focus on **quality interviews rather than resume scanning**.
* To integrate **AI-driven insights** for smarter hiring decisions.

# Methodology:

* **Backend (Spring Boot):**
  + Develop REST APIs for resume upload, text extraction, scoring, and ranking.
  + Implement authentication and role-based access for HR users.
* **NLP and AI Model:**
  + **Use spaCy / Hugging Face transformers / TensorFlow**for extracting skills, experience, and keywords.
  + Match resumes against job descriptions using **semantic similarity scoring** (e.g., cosine similarity, BERT embeddings).
  + Assign weighted scores to rank candidates.
* **Resume Parsing:**
  + Integrate libraries like **Apache Tika** for text extraction from resumes.
* **Frontend (React):**
  + Dashboard to view ranked candidates with filters (skills, score, experience).
  + Visual charts/analytics for HR to quickly compare candidates.
* **Database (MongoDB)**:
  + Use **MongoDB** to store candidate data, job descriptions, and scoring results.

# Expected Outcome:

* **A fully functional AI-powered recruitment tool**that ranks candidates based on job fit**.**
* **Significant reduction in manual screening time**for HR professionals**.**
* **Consistent and unbiased evaluations** powered by NLP models.
* A scalable foundation that can be extended with advanced features like **chatbot-assisted interviews, video resume analysis, or predictive candidate success models**.