✅ Project Title:

"Skill Gap Analyzer and Employability Enhancer for Engineering Graduates in India"

📌 1. Problem Statement

Despite a surge in engineering graduates, India continues to face high unemployment and underemployment rates among freshers. Studies show that over **83% of engineering graduates are either jobless or not industry-ready**, with major gaps in soft skills, modern tech knowledge, and practical experience. There's also **no standardized mechanism to benchmark job readiness**, track post-graduation employment, or align academia with real-time market needs.

🎯 2. Objectives

1. Analyze skill gaps among final-year engineering students using resume parsing, surveys, and coding profile analysis.
2. Compare these findings with current job role expectations from top tech firms (Google, Microsoft, startups, etc.).
3. Provide **personalized roadmaps** using AI that address the gaps.
4. Recommend **entrepreneurial paths (MAST model)** where relevant.
5. Encourage the use of **certifications, micro-credentials**, and **industry projects** for enhanced employability.

📐 3. Scope

Included	Not Included
Resume parsing, skill extraction, coding profile analysis	Company-specific recruitment pipeline simulation
Gap analysis with top job profiles, AI-based recommendations	Offline workshop implementation
Integration with industry trends (EV, AI, Chip Design, Cybersecurity)	Internships and placements
Suggesting upskilling content & platforms (e.g. Coursera, Udemy, GitHub)	In-depth psychometric testing
Visual dashboard for college/institute-level analytics	Government policy change

---

🧱 4. Methodology (Phase-wise Plan)

Phase	Task	Tools/Methods
1	Literature Survey (7 Papers + TOI/ET Reports + Govt. Policies)	Manual synthesis, Excel, Zotero
2	Resume and Profile Analyzer Development	Python, NLP, BERT, SpaCy
3	Industry Role Matching + Gap Engine	Web scraping (LinkedIn, GitHub), BERT
4	Personalized Roadmap Generator	ML + Rule-based recommender
5	Dashboard for College Analytics (skills, readiness, placement gap)	Flask + React + Plotly/Chart.js
6	Pilot testing with dummy resume dataset (100 students)	Testing scripts
7	Report generation for stakeholders (students, colleges, recruiters)	PDF/CSV export

---

🧠 5. Core Technologies

- **Backend**: Flask (Python)
- **Frontend**: React.js
- **ML/NLP**: BERT, SpaCy, Scikit-learn
- **DB**: Firebase or MongoDB
- **Resume Parsing**: docx2txt, pdfminer.six
- **Scraping Tools**: BeautifulSoup, Selenium
- **APIs**: LinkedIn public APIs, GitHub stats
- **Roadmap & Certification Recommenders**: Rule engine + curated dataset

---

📋 6. Data Sources

- Resume and coding profiles from consenting students
- Publicly available profiles from LinkedIn, GitHub, and job portals
- Course data from Coursera, NPTEL, Udemy
- Skill demand data from NASSCOM, WEF, MeitY, etc.

🔍 7. Literature Support

Based on 7 core papers + 3 mainstream reports:

- Resume parsing using LDA/BERT models is achievable with 77–100% precision.
- Skill gaps exist primarily in soft skills, domain exposure, and critical thinking.
- AI-based roadmaps and PL (Personalized Learning) systems like LSE show effective guidance for learners.
- MAST entrepreneurship (Micro AI/Software Tech) is emerging as a feasible alternate career path.

📈 8. Expected Outcome

- **Skill Gap Report** for each user (technical + non-technical)
- **Top Job Match Recommendations**
- **AI-Generated Learning Roadmap**
- **Visualization Dashboard** for colleges to track readiness at batch/department level
- **MAST/Startup Fit Score** (Entrepreneurial suitability)

📎 9. Deliverables

Deliverable	Format
Resume Analyzer + Skill Extractor Tool	Python Script/API
Job Profile Comparison Engine	ML Classifier
Roadmap Generator	HTML Dashboard + CSV Export
Institution-level Readiness Dashboard	React.js Web App
Final Report + PPT + Research Document	DOCX/PDF + PPT

🏁 10. Future Scope

- Integrate real-time job listings for dynamic skill matching.
- Add voice-based feedback and resume walkthrough for accessibility.
- National rollout via NEAT, SWAYAM, AICTE-backed platforms.
- Use the tool as a placement cell plugin across Tier-2/Tier-3 colleges.

📊 Comparative Table: Analysis of Research on Skill Gaps and Job Readiness

Paper	What is the Research About	Methodology Used	How the Work is Done	Literature Gap	Work Needed	Current Progress (per paper)	Problems Yet to be Solved
P1: Resume Parsing with BERT	Automated resume parsing and candidate ranking using BERT	NLP + BERT model + Heuristics	Resume data collected, classified, ranked using BERT pairwise comparison	Lack of universal format parsers; poor semantic extraction in older systems	Build generic parsers for all resume formats	Achieved 100% on LinkedIn formats, 73% on others	Can't verify soft skills or candidate truthfulness; limited by input format
P2: Resume Evaluation using LDA + SpaCy	Resume scoring based on NER + topic modeling	Named Entity Recognition + Latent Dirichlet Allocation	Extracted entities (education, skills) using SpaCy, rated relevance with LDA	Traditional resume parsing relies too much on keywords; ignores semantics	Create more content-aware scoring models with feedback loops	Achieved 77% skill-based and 82% overall accuracy	System doesn't account for subjective recruiter biases; lacks personalization
P3: ResumeVis (Visual Analytics)	Semantic mining and visualization of career paths from resumes	Text mining + Visual Analytics + Case studies	Visualized 2500+ resumes using career trajectory and relationship graphs	Existing RA methods focus only on structured data; lack career evolution analysis	Integrate visualization tools in HR systems; support flexible resume types	Able to trace career growth, latent social links in resumes	Complex to scale for global datasets; doesn't support multilingual resumes yet
P4: Latent Skill Embedding (Lesson Recommender)	Personalized lesson sequence prediction based on skill gaps	Latent Skill Space + Hidden Markov Model	Used data from Knewton; predicted student success with personalized paths	No domain-agnostic systems for sequence recommendation existed	Extend model for non-binary evaluations; integrate with resume/job platforms	Predicted success well, discriminated between useful vs. failed paths	Needs real-world deployment validation; lacks integration with HR analytics

P5: AI in Personalized Learning (OECD Compass)	Critical analysis of AI-based personalized learning vs. educational goals	Meta-analysis + Framework mapping	Evaluated systems against OECD goals (agency, SRL, AAR cycle)	Tech solutions too performance-focused; miss holistic learning vision	Merge SRL, emotional intelligence & ethics into AI tools	Strong theoretical model with AI-human hybrid PL concept	Gap in cognitive/metacognitive support; AI tools lack general competency alignment
P6: Learning Analytics for Skill Gaps	Uses survey + ML to assess undergrad awareness & aspirations	Survey, Clustering, KNN, Chi-square, U-Test	Surveyed students; classified gaps in skills and university support	No data-backed insights on how student perceptions align with goals	Incorporate skills tracking + roadmap feedback in universities	Found actionable mismatches between skills taught vs. required	Needs stronger integration with curriculum redesign and internship support
P7: Skill Gap in Indian Education	Evaluates mismatch between employer expectations and graduate skills	Likert-scale survey with students and HR professionals	Identified gaps in technical, soft, behavioral skills via ranking	Fragmented approach across institutions; curriculum outdated	Government-academia collaboration; industry-standard frameworks	Identified soft skills and communication as top gaps	Higher ed still theoretical; poor implementation of gov schemes at local levels

🔍 **Key Insights:**

- 🔴 **Common Literature Gap**: Most systems don’t handle **soft skills, behavioral traits**, or **career evolution** over time. Most models are static and domain-limited.
- 🟡 **Work Still Needed**:
    - Integration of resume + learning analytics + job market data
    - Personalized, **evolving roadmaps** based on real-time feedback
    - Hybrid systems that **combine AI with teacher/HR input**
- 🟢 **Current Progress**:
    - Resume parsers using BERT, LDA, SpaCy are improving.
    - Visualization and recommender systems (ResumeVis, LSE) show strong results.
    - Awareness of gaps is growing among academia (NTU study, OECD goals).

📊 Comparative Analysis of Research Papers on Engineering Employability in India

Paper	What is the Research About	Methodology Used	How the Work is Done	Literature Gap	Work Needed	Current Progress (per paper)	Problems Yet to be Solved
P1: Employability & Skill Set of Engineers (World Bank)	Investigates skill shortages among Indian engineering graduates; employer expectations and satisfaction.	Nationwide employer survey (157 firms), factor analysis, stratified sampling.	Collected ratings on 26 skills' importance and satisfaction; categorized into soft vs technical skills.	Few India-focused employer satisfaction studies; lacked large sample; no standard skill assessment.	Curriculum to focus on higher-order skills; better industry-academia alignment; improve soft skills.	Surveyed skills gap data shows action needed; employer feedback documented; World Bank support.	Persistent skill gaps in problem-solving, creativity; lack of consistent feedback loop from employers.
P2: Employers' Perspectives on Employability Skills	Synthesizes employer expectations and satisfaction from graduates; reviews past studies.	Systematic literature review of Indian studies on employability.	Compared multiple secondary data studies, focused on skill categories and employer views.	Limited research on specific industry needs per region; few empirical longitudinal studies.	Standardized skills training; institutional collaboration; data-driven skill enhancement strategies.	Cataloged over a dozen employability frameworks; common emphasis on soft skills.	Mismatch between curriculum and job roles in Industry 4.0; lack of personalization in skilling.
P3: Future of Core Engineering Jobs in India	Trends in traditional engineering roles and new opportunities in green energy, EVs, manufacturing.	Market trend analysis and projections using sector data.	Analyzed core vs non-core job data, sector hiring patterns, government policies.	Lack of awareness among students about emerging core jobs; inadequate focus on applied training.	Promote internships in core sectors, integrate sustainability and AI into core curricula.	Govt missions like EV and Green Hydrogen show hiring intent; awareness still low.	Limited exposure to core industrial jobs; job preference shifting toward IT regardless of discipline.
P4: Let a Million	Proposes micro-	Position paper; data	Highlights supply-	Entrepreneurship not yet	Create MAST-	Outlined practical	Lack of institutional

Entrepreneurs Grow	entrepreneurship as a solution to engineering graduate unemployment.	from MeitY, NASSCOM, and hiring trends.	demand imbalance; proposes MAST (Micro AI/Software Tech) enterprises.	mainstream in engineering education; rarely proposed as primary employment path.	focused curricula; support via incubation, seed funding, mentorship.	models; pilotable at local level with policy support.	readiness to promote entrepreneurship; risk aversion among students.
P5: Mismatched Skills, Missed Opportunities (TOI)	Analyzes job crisis among engineers including top IITs; impact of GenAI and tech demands.	RTI data, industry expert interviews, hiring data analysis.	Presents trends from TeamLease, WEF, etc.; links skill gaps to job market failures.	Lack of real-time curriculum evolution with tech trends (e.g., AI, EV, chip design).	Revise curricula around AI, cloud, cybersecurity, EV, green hydrogen sectors.	Hiring in niche domains is growing (EV, chip design); GenAI seen as opportunity.	Massive gap in job-readiness; most graduates not skilled for emerging roles.
P6: Less than Half Job-Ready (ET Report)	Reports that less than 50% of graduates are job-ready due to outdated education.	Survey and industry skill readiness report.	Reviewed nationwide readiness data and corporate assessments.	Inadequate measurement tools to benchmark job-readiness in colleges.	Adopt skill certification programs; regular curriculum-industry sync.	Awareness rising on low readiness; policy discussions initiated.	Disjointed reform efforts; weak employer-college feedback integration.
P7: 83% of Engineers Jobless (TOI Report)	Investigates alarming joblessness among engineers across India.	News report backed by survey/statistical data.	Presents national employment survey; uses expert commentary.	No national tracking of graduate employment post-degree.	Create employment audit system; link education funding to outcome metrics.	Reports raise public concern; few concrete policy implementations yet.	Employability still not tied to academic success; poor regional opportunity mapping.

🔍 **Key Insights**:

- 🔴 **Common Literature Gap**:
  - Severe mismatch between graduate skill sets and industry expectations persists across all papers.
  - Most frameworks emphasize soft skills but lack mechanisms for continuous feedback or regional customization.
  - Entrepreneurship as a solution is underexplored in engineering education.

- There is limited use of real-time labor market analytics and job-readiness audits.

- 🟡 **Work Still Needed**:
  - 🔧 Institutionalization of employer feedback into curriculum design.
  - 🔧 Mainstream entrepreneurship (especially MAST model) with support mechanisms in engineering colleges.
  - 🔧 National skill benchmarking system and post-graduation employment tracking.
  - 🔧 Shift from theoretical training to practical, interdisciplinary, and AI-enabled learning modules.
  - 🔧 Establishment of industry-standard certifications and micro-credentialing.

- 🟢 **Current Progress**:
  - 📈 National initiatives like the EV Mission and Green Hydrogen Mission have opened up future hiring potential.
  - 🧠 Literature reviews and employer surveys have clarified the importance of soft skills, communication, and problem-solving.
  - 💼 Awareness is rising in academia and policy sectors regarding low employability rates and the role of generative AI.
  - 🚀 Pilot models like MAST entrepreneurship are gaining conceptual support, though not yet scaled.
  - 📊 Industry now vocal about niche hiring trends (cybersecurity, chip design, cloud), guiding curriculum redesign discussions.

📁 Folder Structure – SkillGapAnalyzer/

```
SkillGapAnalyzer/
├── backend/                        # Flask backend API
|     ├── app.py                    # Main Flask app entry point
|     ├── config.py                 # Flask config (DB, secrets)
|     ├── models/                   # DB Models (User, Resume, Reports)
|     |     └── user.py
|     ├── routes/                   # All API endpoints
|     |     ├── resume.py           # Resume upload and parsing
|     |     ├── compare.py          # Skill comparison logic
|     |     └── roadmap.py          # Learning roadmap generation
|     ├── services/                 # Core logic (ML/NLP, comparisons)
|     |     ├── parser_engine.py
|     |     ├── skill_matcher.py
|     |     └── roadmap_generator.py
|     ├── utils/                    # Helper functions (file ops, validators)
|     ├── static/uploads/           # Uploaded resumes (PDF/DOCX)
|     └── templates/                # Optional Flask views (if used)

├── frontend/                       # React Frontend
|     ├── public/
|     └── src/
|          ├── assets/              # Icons, logos, images
|          ├── components/          # React components (Navbar, Cards)
|          ├── pages/               # Views (Upload, Dashboard, Results)
|          ├── services/            # Axios calls to Flask backend
|          ├── utils/               # Reusable functions
|          ├── App.js
|          └── index.js

├── ml_nlp_model/                   # Resume parser & skill classifier
|     ├── dataset/                  # Training data for resumes/skills
|     ├── training_notebooks/       # Jupyter notebooks for experiments
|     ├── fine_tuned_model/         # Saved model (TF or PyTorch)
|     ├── inference.py              # Prediction entry point
|     └── requirements.txt

├── scraper/                        # Web scraping profiles from LinkedIn, GitHub
|     ├── linkedin_scraper.py
|     ├── github_scraper.py
|     └── top_profiles.json         # Sample data of top engineers

├── database/                       # DB schema, migrations
|     ├── schema.sql
|     └── init_db.py

├── docs/                           # Documentation, report, API usage
|     ├── architecture.png
|     ├── ER_diagram.drawio
|     └── README.md

├── tests/                          # Backend unit tests
|     └── test_resume_parser.py

├── .env                            # Environment variables
├── .ignore
├── README.md                       # Project overview
└── requirements.txt                # Python packages (Flask + ML)
```

Job Skill Gap Analyzer for Tier-3 College Students

Problem Solved: Difficulty in cracking top tech jobs due to unknown gaps.

What It Does:

• Takes resume, academic background, projects, and coding profile.

• Compares with profiles of engineers at Google, Microsoft, etc.

• Suggests personalized learning roadmap.

Tech Stack: NLP, Resume Parsing, Web Scraping, Flask, React.

How it is going to compare Engineers at Google, Microsoft and how we are going to prove it the comparison is legit.

To **legitimately compare tier-3 college students' skills with engineers at Google, Microsoft, etc.**, and **prove the validity of such comparisons**, your project will need a **data-backed, explainable, and transparent methodology**. Below is a detailed explanation of *how it works* and *how you can justify its legitimacy*.

✅ How Comparison with Engineers at Google/Microsoft Works

1. Data Collection (via scraper/)

- You **scrape publicly available professional profiles** from **LinkedIn** and **GitHub** of engineers working at top tech companies.
- These profiles include:
    - Resume-style information (education, skills, work experience)
    - Open-source contributions (GitHub)
    - Projects and repositories
    - Coding platform profiles (LeetCode, HackerRank scores)

✅ *Stored in*: scraper/top_profiles.json

---

2. Data Preprocessing & Feature Engineering

In ml_nlp_model/ and services/skill_matcher.py:

- Clean and normalize skill sets from resumes and top profiles using NLP.
- Extract structured skills:
    - Programming languages
    - Frameworks/Tools
    - CS fundamentals
    - Problem-solving ability
    - Real-world project complexity
    - Communication (optional, through project descriptions or blogs)

You then **vectorize these features** for both the student and reference profiles.

✅ Uses models like **TF-IDF**, **BERT embeddings**, or **custom NLP pipelines**.

---

3. Matching Algorithm (Legit Comparison)

In services/skill_matcher.py:

- Implement **cosine similarity** or **semantic similarity** between:
    - Resume skill embeddings (student)
    - Engineer profile embeddings (from Google, Microsoft)
- Compare:
    - **Skill overlap percentage**
    - **Coding activity (e.g., GitHub commits vs. theirs)**
    - **Project depth and impact**
    - **Resume strength (from parser_engine)**

✅ *Output: A score or "Gap Report"*, showing how far the student is from a typical profile at top firms.

✅ **How to Prove the Comparison is Legitimate**

🔬 **1. Use Real, Public, and Verifiable Data**

- Your scraper should fetch profiles using **known handles** of engineers (LinkedIn open data, GitHub public repos).
- Store their anonymized summaries in top_profiles.json as sample evidence.

📊 **2. Benchmark with Multiple Profiles**

- Don't compare with just one engineer—use an **average** or **aggregated features** from 50–100+ profiles per company.
- Create **company-wise skill sets** (e.g., top 10 skills of Microsoft engineers) and match against that.

🤖 **3. Explainable ML/NLP Models**

- Provide insights like:
    - "You lack X skills present in 85% of Google engineers"
    - "Your LeetCode profile has 200 problems, while the average Google engineer has 600+"
- Display side-by-side skill tables for transparency.

🧪 **4. Quantitative Evaluation**

- Run evaluation on **real student resumes**.
- Compare their gap score with interview outcomes or job offers (if available).
- Publish a **case study** (in docs/) showing improvements after following the roadmap.

📈 Output Example (From compare.py)

json

CopyEdit

```json
{
  "matched_company": "Google",
  "overall_similarity": 62.5,
  "missing_skills": ["Distributed Systems", "Kubernetes", "System Design"],
  "coding_gap": "You have solved 150/600 benchmark problems on LeetCode.",
  "project_gap": "Your projects focus on basic CRUD. Google profiles show projects with ML, scale, or research."
}
```

🗺️ Personalized Roadmap (roadmap_generator.py)

- Suggest resources (Coursera, freeCodeCamp, GitHub repo links) based on the missing skills.
- Suggest resume structure tips based on best practices.

✅ Summary: Legitimation Strategy

Component	Purpose	Legitimacy
linkedin_scraper.py, github_scraper.py	Real engineer data	Factual source
NLP-based skill matching	Resume vs. benchmark	Transparent algorithms
Aggregate company profiles	Not just one-to-one	Generalized and fair
Gap report + Roadmap	Personalized advice	Actionable and measurable
Docs with sample profiles and metrics	Evidence	Proves it's not random