Job Scam Alert System - JobGuard

Capstone Project: Week 1 – Ideation and Documentation Stage

Importance of Business Research and Team Dynamics

Business Research, Ideation, Product Vision

Introduction:

The best products don't start with code — they start with a real user problem.

For many freshers and recent graduates, that problem is job scams. Fake job postings, phishing emails, and fraudulent offer letters are common, and most existing platforms like Naukri, LinkedIn, and Indeed do little to warn users before they apply. As a result, job seekers often lose money or personal data.

Our system solves this problem by detecting scam patterns using AI, verifying company names using a central MongoDB blacklist, and issuing real-time alerts before users submit job applications.

This platform is more than just a job portal — it's a security shield for freshers in their first job search journey.

Business Research:

Chosen Domain:HRTech

Target Audience:

- Final-year college students
- Fresh graduates and job seekers (0–2 years experience)
- Parents of students
- College placement officers

Competitors:

Platform	Weakness
Naukri	No scam alerts or AI validation
LinkedIn	Scam jobs can appear; relies on reporting
WhatsApp	Job groups full of unmoderated scam links
Telegram	Bots used to spread fake job postings
Internshala	Limited to internships; no scam validation

Strengths :

- Solves a real and rising problem
- Uses AI for smart detection (job descriptions)
- Provides real-time alerts via email
- Tailored to freshers with a clean, simple UI

Weaknesses :

- Requires initial company data to train AI
- AI may have false positives/negatives
- Trust-building takes time

Opportunities :

- API as a service for other job portals
- Integration with college placement cells
- Resume scanner or offer-letter fraud detection tool

Threats :

- Competitors may implement similar scam detection
- AI misclassification risks user experience
- Building and maintaining a verified company list

Project Ideation:

Identified Problem:

Freshers are getting scammed by fake companies due to:

- No real-time company verification
- No system to detect suspicious job descriptions
- No alerts or blocking system before applying

Idea:

A MERN stack web platform that allows job seekers to:

- Check company validity using a MongoDB list
- Use an AI API to analyze job descriptions for fraud signals
- Prevent submission of suspicious applications
- Report scam jobs
- Get alerts via email

Admins can:

- Review reported companies/jobs
- Blacklist/verify companies
- View platform stats and scam trends

Product Vision:

"To protect freshers from job scams using AI-based analysis, company verification, and smart alerts — creating a secure job search platform that blocks threats before they happen."

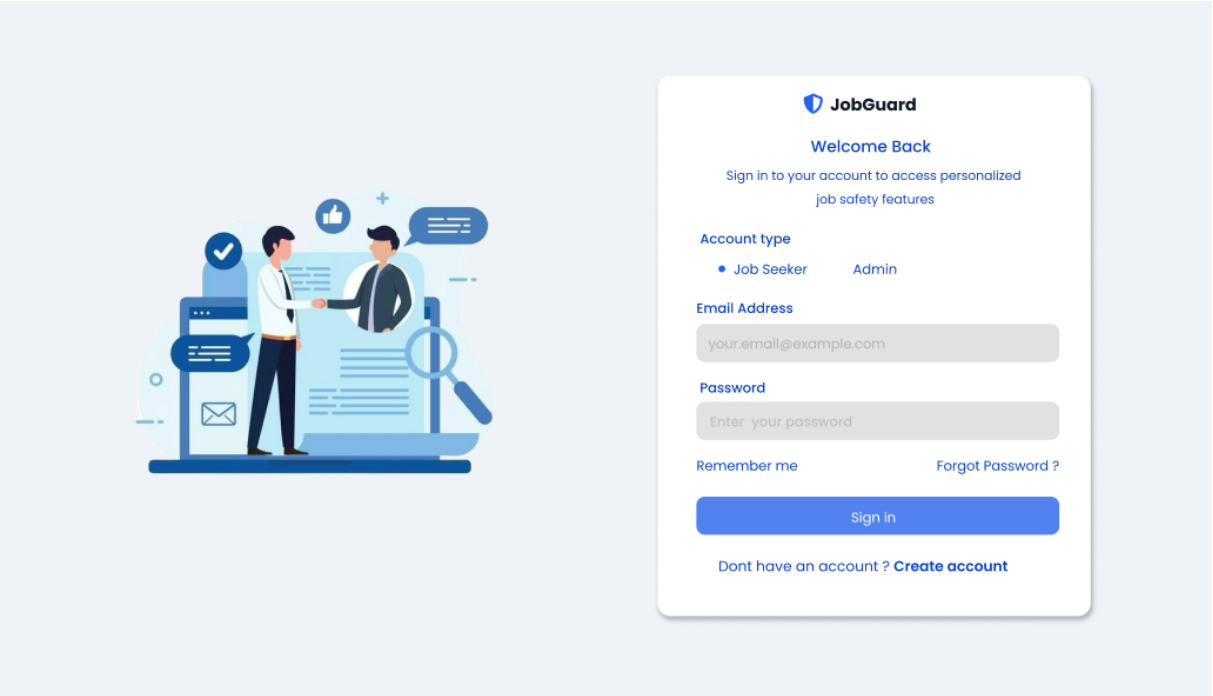
Requirement Gathering, Use Case Analysis:

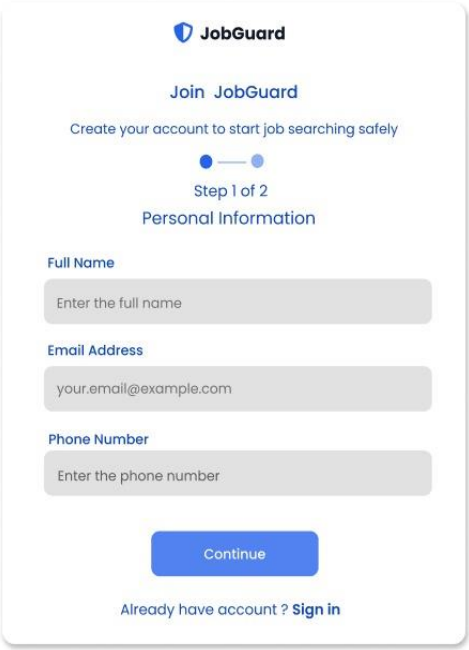
Modules to be implemented:

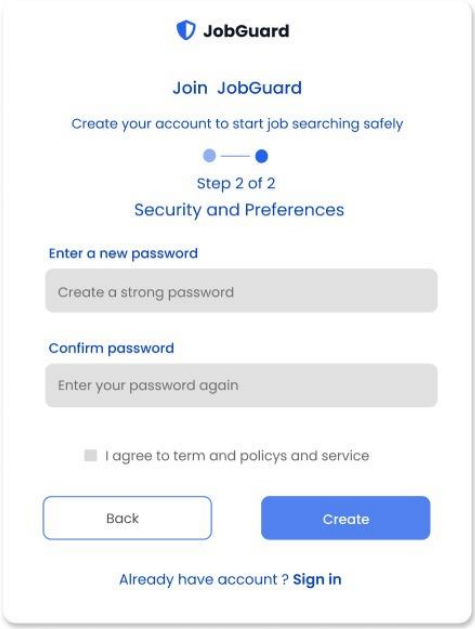
1. User Management
2. Admin Dashboard
3. Company Validation Checker
4. Job Submission Form (Smart Form)
5. Scam Report System
6. Job Listings with Risk Indicators
7. Alerts & Notifications
8. MongoDB Company Database Integration
9. AI-Based Scam Detection Module

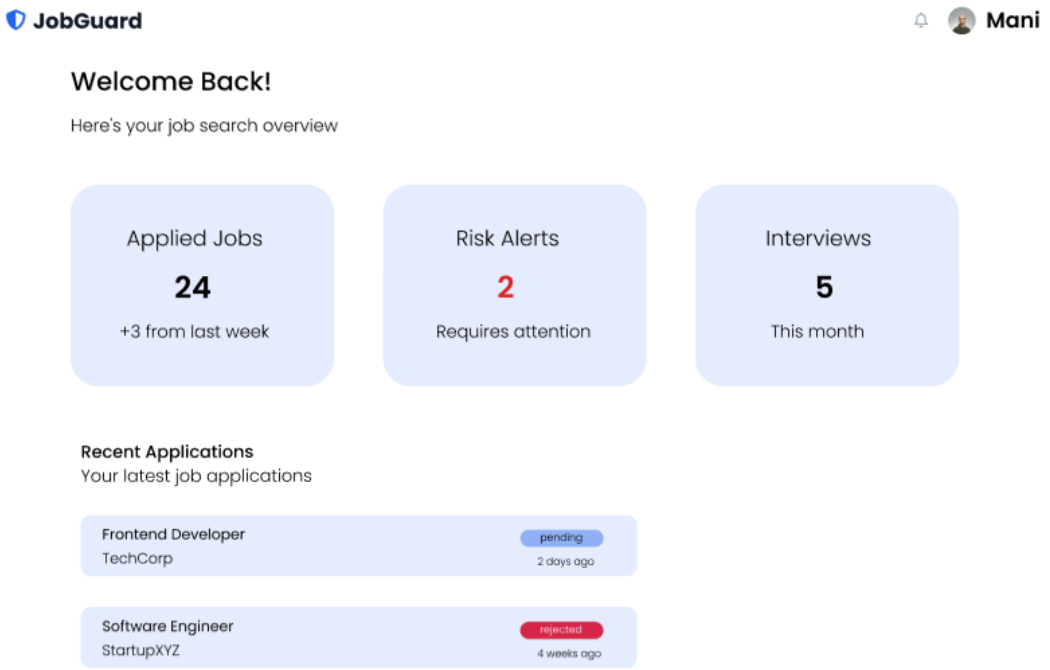
Use Case analysis,User stories and goals:

1. User Management (Loginpage)









Goals:

- Enable secure sign up, login, logout, and password reset functionality.
- Support account type selection (Admin / Job Seeker).
- Store hashed passwords and role-based access in MongoDB.
- Allow account verification (e.g., email OTP).
- Ensure responsive design and accessibility.

User stories:

- As a user, I can register with my personal details and set a secure password.
- As a user, I can log in with my email and password to access personalized features.
- As a user, I can select whether I'm an Admin or a Job Seeker during login.
- As a user, I can reset my password via an email OTP flow.
- As an Admin, I can log in and access administrative privileges and data panels.

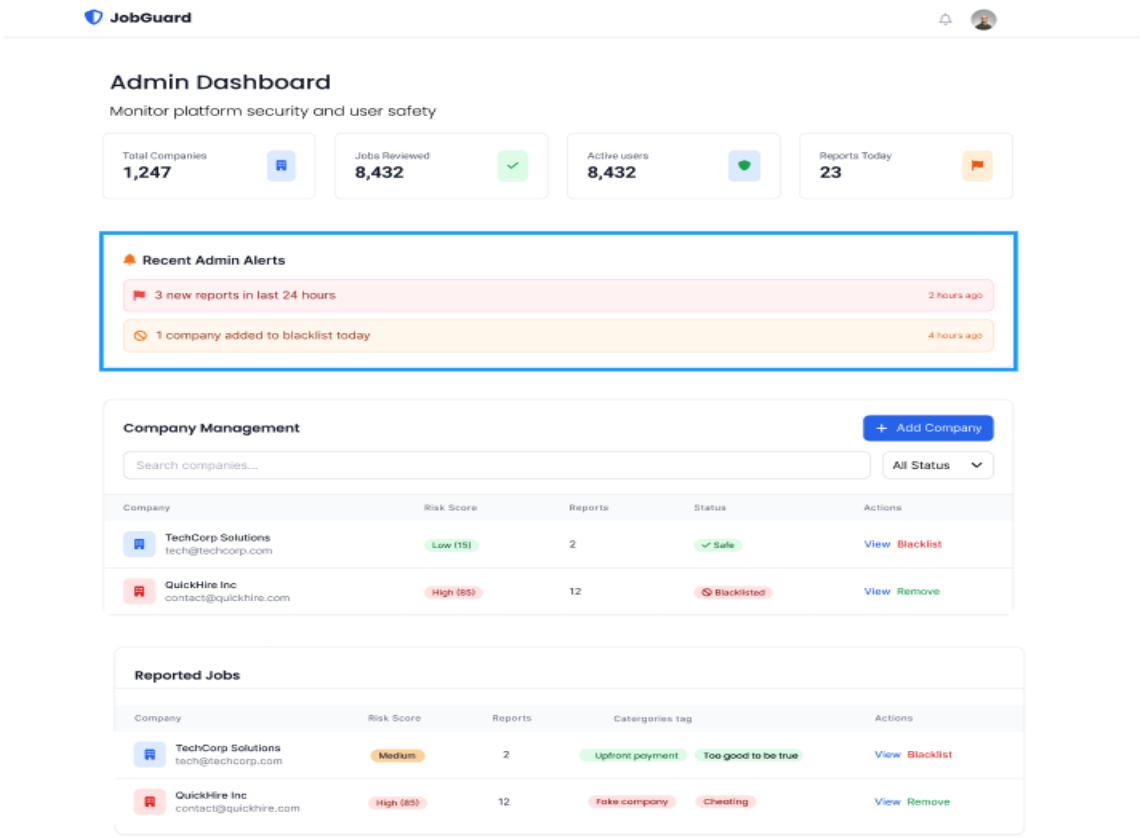
2.Admin Dashboard

Goals:

- Allow users to fill out a job application form with company details.
- Auto-check company name for blacklist or scam warning.
- Prevent submission if the company is marked as Suspicious.
- Validate inputs (email, contact, job title, etc.)

User Stories:

- As a user, I can fill a job application form and submit details for risk assessment.
- As a user, I get a warning if I enter a blacklisted company name.
- As a user, I cannot submit a form if the job listing is suspected to be fake.



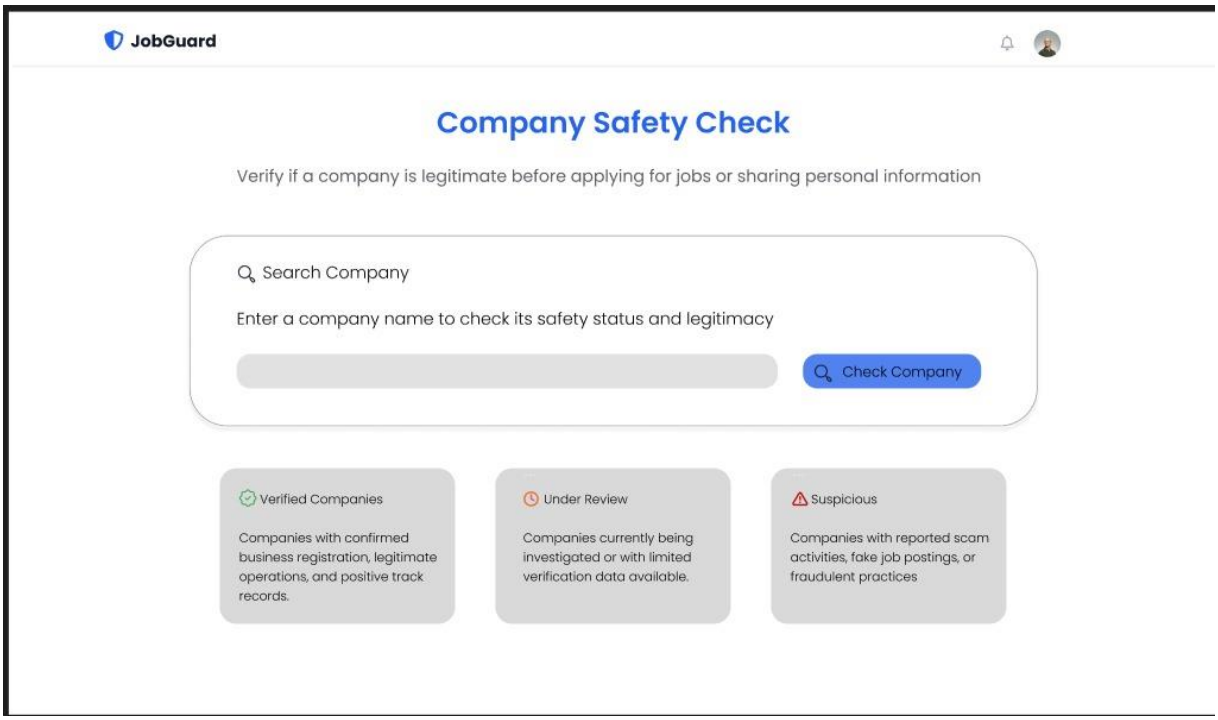
3.Company Validation Checker

Goals:

- Allow users to enter and validate company names.
- Integrate MongoDB and optional AI API to determine legitimacy.
- Classify company into Verified, Under Review, or Suspicious.
- Display risk scores with explanatory labels.

User Stories:

- As a user, I can search a company by name and instantly see its legitimacy status.
- As a user, I can avoid applying to suspicious companies flagged by AI or the admin.
- As a user, I can trust Verified companies based on registration and safety data.


JobGuard
Company Safety Check
Verify if a company is legitimate before applying for jobs or sharing personal information
Search Company
Enter a company name to check its safety status and legitimacy
Check Company
Verified Companies
Companies with confirmed business registration, legitimate operations, and positive track records.
Under Review
Companies currently being investigated or with limited verification data available.
Suspicious
Companies with reported scam activities, fake job postings, or fraudulent practices

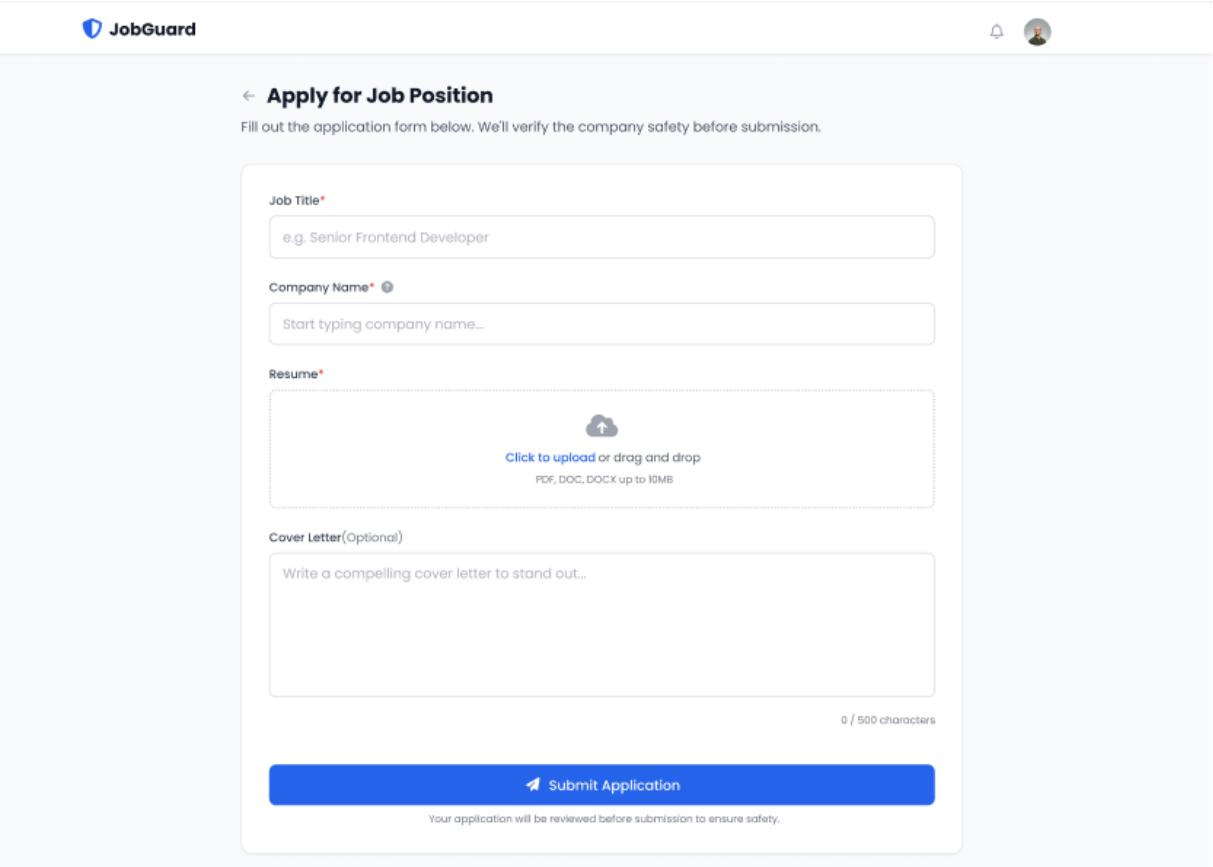
4. Job Submission Form (Smart Form)

Goals:

- Allow users to fill out a job application form with company details.
- Auto-check company name for blacklist or scam warning.
- Prevent submission if the company is marked as Suspicious.
- Validate inputs (email, contact, job title, etc.)

User Stories:

- As a user, I can fill a job application form and submit details for risk assessment.
- As a user, I get a warning if I enter a blacklisted company name.
- As a user, I cannot submit a form if the job listing is suspected to be fake.



5. Scam Report System

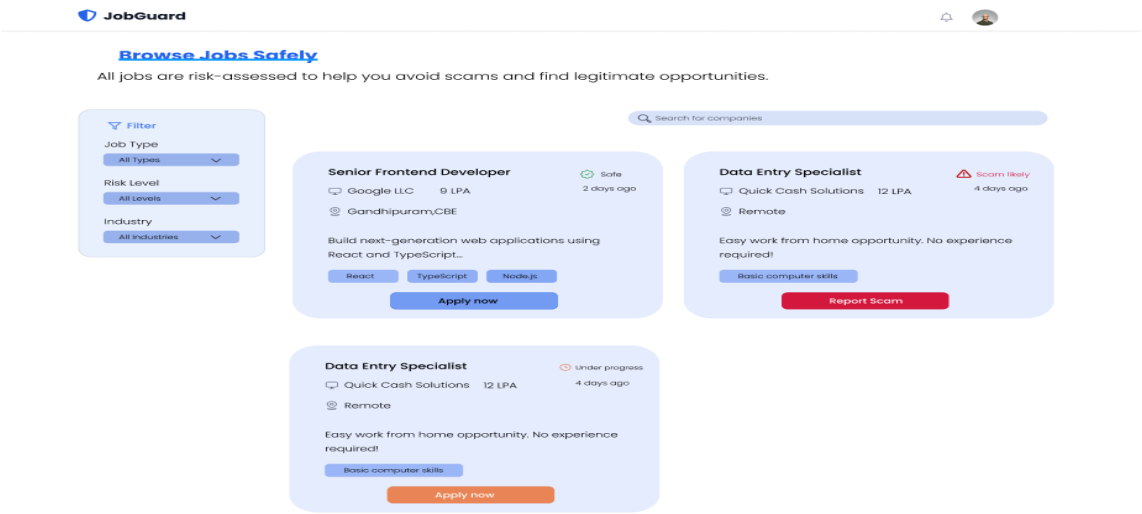
Goals:

- Allow users to report suspicious job offers.
- Include form fields like company name, job description, evidence (screenshots).
- Store reports for admin review.
- Alert other users if similar scams are reported repeatedly.

User Stories:

- As a user, I can report a scam by filling a short form.
- As a user, I can upload evidence of scam emails or messages.
- As an admin, I can view, verify, and blacklist companies based on reports.
- As a user, I am protected from companies that are frequently reported.

6. Job Listings with Risk Indicators



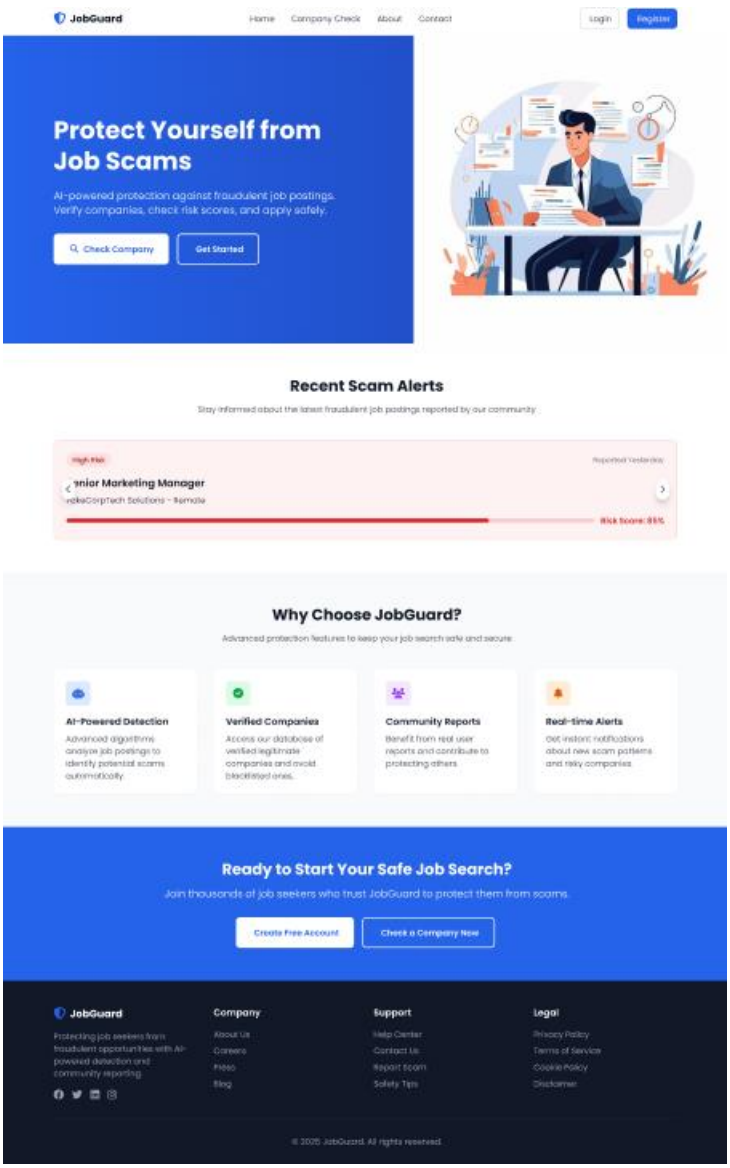
Goals:

- Show posted jobs to users with color-coded risk scores.
- Allow filtering (e.g., All, Verified, Suspicious).
- Show company name, job title, and associated risk score.

User Stories:

- As a job seeker, I can browse job listings and see whether each is safe or risky.
- As a user, I can understand why a job is flagged as suspicious.
- As a user, I can avoid applying to high-risk listings based on the score.

7. Alerts & Notifications



Goals:

- Send email alerts when companies are flagged or reported.
- Display in-app notifications for login, scam alerts, and AI predictions.
- Remind users to update credentials if there's suspicious login activity.

User stories:

- As a user, I receive an alert if a company I applied to gets flagged.
- As a user, I am notified in real-time if a job scam pattern is detected.
- As an user,I am informed via Email if there is a login from an unknown Location.

8. MongoDB Company Database Integration

Goals:

- Store company names, risk levels, validation status in MongoDB.
- Enable search, add, update, and blacklist functions.
- Link company documents with job postings and reports.

User stories:

- As an admin, I can add or edit company records in the MongoDB backend.
- As a system, I retrieve company status (e.g., blacklisted or verified) from MongoDB when users search.
- As a user, I receive updated data powered by MongoDB when checking companies.

9. AI-Based Scam Detection

Goals:

Goals:

- Train a machine learning model (or use OpenAI API) to classify job postings/companies.
- Assign risk scores based on suspicious keywords, past reports, etc.
- Display the AI output visually

User Stories:

- As a user, I can see the risk score for each job/company based on AI analysis.
- As an admin, I can trust the AI model to flag potential scams for manual review.

- As a system, I continuously learn from user reports to improve detection accuracy.

Functional Requirements :

1. User Management

- Users can register as either Job Seeker or Admin.
- Registration includes basic personal details and secure password creation.
- Users can log in, log out, and reset forgotten passwords.
- User roles (Job Seeker, Admin) are enforced with role-based dashboard access.
- Login and registration forms validate email and password strength.
- Optional: OTP or email verification for account activation.

2. Admin Dashboard

- Admin can view all registered users and reported job scams.
- Admin can verify reported scam entries and manage the blacklist database.
- Admin can add, update, or remove company entries from MongoDB.
- Admin can view dashboard analytics (total users, flagged companies, active reports).

3. Company Validation Checker

- Job Seekers can search for a company by name.
- The system cross-references MongoDB and AI-based classifiers to determine risk level.
- Company risk levels include Verified, Under Review, and Suspicious.
- Admins can manually edit company validation status.

4. Job Submission Form (Smart Form)

- Job Seekers can submit job details for AI-based fraud detection.
- Company name is auto-validated on input to check for risk level.
- If company is marked “Suspicious,” form submission is blocked with a warning.

- Input fields include: company name, role offered, description, contact, source.

5. Scam Report System

- Job Seekers can report a suspicious job offer via a dedicated form.
- Form includes fields for job details, evidence (optional file/image), and comments.
- Submissions are stored in MongoDB and visible to Admin for review.
- Repeated reports on the same company raise automated alerts to Admin.

6. Job Listings with Risk Indicators

- A listing page shows submitted job offers by other users.
- Each job entry displays a Risk Score: Green (Safe), Yellow (Under Review), Red (Scam Likely).
- Listings are filterable based on risk level or company name.
- Tooltips explain why a job might be considered suspicious (e.g., no registration, mismatch in info, etc.).

7. Alerts & Notifications

- Users receive in-app alerts for recent scam activity and risky companies.
- Email alerts are sent if a company a user interacted with becomes flagged.
- Notifications appear on dashboard (e.g., “3 new scams reported this week.”)

8. MongoDB Company Database Integration

- Companies are stored with name, status, source link, createdAt, updatedAt.
- Admin can add new companies manually or based on user reports.
- Company data is fetched in real-time for validation in forms and search.

9. AI-Based Scam Detection Module

- AI model (OpenAI or custom classifier) analyzes job descriptions and company metadata.
- Predicts scam likelihood based on red-flag patterns (e.g., urgent hiring, no interviews, Gmail contact, etc.)

- AI output is shown as a percentage risk score (e.g., 85% Likely Scam).
- Model learns continuously from new reports and patterns.

Non-Functional Requirements :

Security

- Passwords are encrypted using bcrypt hashing.
- JWT tokens are used for secure session-based authentication.
- Role-based access (admin vs user) ensures security of sensitive operations.
- All forms validate user inputs to prevent injection, XSS, or malicious uploads.
- Company records are editable only by Admin with authentication checks.

Usability

- Mobile-friendly and responsive layout using Tailwind CSS / Bootstrap.
- Forms show instant feedback on validation (e.g., “Invalid email address”).
- Intuitive flow with minimum steps to complete any major task (e.g., report scam in 2 steps).
- Consistent layout across pages improves navigation and user trust.

Performance

- Search, filter, and company validation actions respond in under 1 second.
- Homepage and dashboard load within 2 seconds under normal load.
- Efficient pagination and indexing for MongoDB job listings and company data.

Scalability

- MERN Stack (MongoDB, Express, React, Node.js) allows modular expansion.
- New roles (e.g., moderator, partner) can be added easily.
- Support for adding AI APIs or microservices independently.
- Cloud-deployable via services like Render, Vercel, or Heroku.

Reliability & Availability

- Uptime goal of 99.9% ensured with cloud deployment monitoring.
- Retry logic added for failed email notifications or file uploads.

Auto-reconnect on MongoDB disconnect via Mongoose options

Non-Functional Requirements :

Security

- Passwords are encrypted using bcrypt hashing.
- JWT tokens are used for secure session-based authentication.
- Role-based access (admin vs user) ensures security of sensitive operations.
- All forms validate user inputs to prevent injection, XSS, or malicious uploads.
- Company records are editable only by Admin with authentication checks.

Usability

- Mobile-friendly and responsive layout using Tailwind CSS / Bootstrap.
- Forms show instant feedback on validation (e.g., “Invalid email address”).
- Intuitive flow with minimum steps to complete any major task (e.g., report scam in 2 steps).
- Consistent layout across pages improves navigation and user trust.

Performance

- Search, filter, and company validation actions respond in under 1 second.
- Homepage and dashboard load within 2 seconds under normal load.
- Efficient pagination and indexing for MongoDB job listings and company data.

Scalability

- MERN Stack (MongoDB, Express, React, Node.js) allows modular expansion.
- New roles (e.g., moderator, partner) can be added easily.
- Support for adding AI APIs or microservices independently.
- Cloud-deployable via services like Render, Vercel, or Heroku.

Reliability & Availability

- Uptime goal of 99.9% ensured with cloud deployment monitoring.
- Retry logic added for failed email notifications or file uploads.
- Auto-reconnect on MongoDB disconnect via Mongoose options.

Maintainability

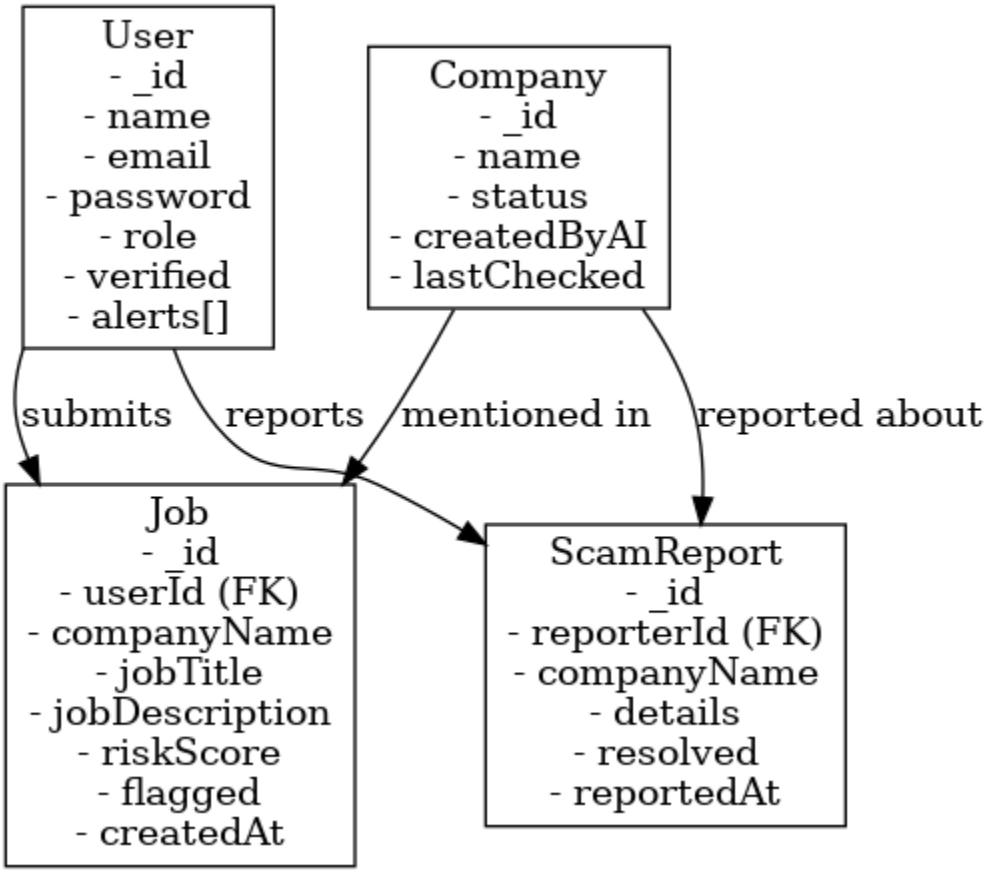
- Clean, well-documented codebase using modular structure (controllers, routes, services).
- Comments provided on backend API logic and frontend form validations.
- Admin dashboard allows manual corrections (e.g., remove false report, reset password).

MoSCoW Feature List:

Feature	Description	MoSCoW Tag
User Authentication	Register/Login system with role-based access (user/admin)	**Must Have**
Company Verification Checker	Search and verify if a company/job is flagged as scam	**Must Have**
Scam Report Submission	Users can submit detailed reports about suspicious jobs/companies	**Must Have**
Admin Dashboard	Admin can view, verify, and manage reported scam entries	**Must Have**
View Scam Reports List	Users can see verified scam companies in a list	**Should Have**
Machine Learning Prediction	Predict if a job post is a scam based on keywords/phrases	**Could Have**
Email Alert System	Notify users if their report has been reviewed or if new scams are posted	**Should Have**
User Profile Page	View/edit user details, check report status	**Could Have**
Live Chat Support	Users can chat with support staff for reporting help	**Won't Have (now)**
Mobile-Friendly Design	Responsive design for smartphones and tablets	**Should Have**

Feature	Description	MoSCoW Tag
Report Analytics Page (Admin)	Admin can view number of reports, common scam words, etc.	**Could Have**
Fake Job Pattern Database	A backend list of flagged keywords/phrases used in scam jobs	**Should Have**
Multilingual Support	Website supports regional languages	**Won't Have (now)**

ER Diagram:



API List with Request Format:

1. /api/auth

- POST /register → Create new user
- POST /login → Authenticate user

2. /api/jobs

- POST /submit → Submit job form
- GET /all → Get all jobs
- GET /myjobs → Jobs posted by user

3. /api/ai

- POST /analyze → Analyze job text using OpenAI

4. /api/company

- POST /check → Check if company blacklisted
- POST /add → Add new suspicious company (Admin)

5. /api/scam

- POST /report → User reports a scam job
- GET /reports → Admin view reports

6. /api/admin

- GET /dashboard → Get metrics
- POST /blacklist → Add company to blacklist

OVERVIEW:

- Stack: MERN (MongoDB, Express.js, React.js, Node.js)
- AI Integration: OpenAI API (for scam detection)
- Email Notifications: NodeMailer
- DB: MongoDB (Mongoose ODM)

BACKEND STRUCTURE:

```
/project-root
    |
    ├── controllers/
    |   ├── authController.js
    |   ├── jobController.js
    |   ├── aiController.js
    |   ├── adminController.js
    |
    ├── models/
    |   ├── User.js
    |   ├── Company.js
    |   ├── Job.js
    |   ├── ScamReport.js
    |
    ├── routes/
    |   ├── authRoutes.js
    |   ├── jobRoutes.js
    |   ├── aiRoutes.js
    |   ├── adminRoutes.js
    |
    ├── utils/
    |   ├── sendMail.js
    |
    ├── .env
    ├── server.js
```

SCHEMA DESIGNS (MongoDB via Mongoose)

1. User.js

```
const mongoose = require('mongoose');

const userSchema = new mongoose.Schema({
    name: String,
    email: { type: String, unique: true },
    password: String,
    role: { type: String, enum: ['admin', 'seeker'], default: 'seeker' },
    isVerified: { type: Boolean, default: false }
    }, { timestamps: true });

module.exports = mongoose.model('User', userSchema);
```

2. Company.js

```
const companySchema = new mongoose.Schema({
    name: String,
    isBlacklisted: { type: Boolean, default: false },
    reason: String
    }, { timestamps: true });
```

3. Job.js

```
const jobSchema = new mongoose.Schema({
    title: String,
    company: String,
    description: String,
    submittedBy: { type: mongoose.Schema.Types.ObjectId, ref: 'User' },
    riskScore: String, // "Safe", "Suspicious", "Likely Scam"
    aiFeedback: String
    }, { timestamps: true });
```

4. ScamReport.js

```
const reportSchema = new mongoose.Schema({
    reportedBy: { type: mongoose.Schema.Types.ObjectId, ref: 'User' },
    jobTitle: String,
    company: String,
    description: String,
    status: { type: String, default: 'Pending' }
    }, { timestamps: true });
```

API ROUTES

2. /api/auth

- POST /register → Create new user
- POST /login → Authenticate user

3. /api/jobs

- POST /submit → Submit job form
- GET /all → Get all jobs
- GET /myjobs → Jobs posted by user

4. /api/ai

- POST /analyze → Analyze job text using OpenAI

5. /api/company

- POST /check → Check if company blacklisted

- POST /add → Add new suspicious company (Admin)

6. /api/scam

- POST /report → User reports a scam job
- GET /reports → Admin view reports

7. /api/admin

- GET /dashboard → Get metrics
- POST /blacklist → Add company to blacklist

AI USAGE (OpenAI Integration):

- Input: Company name + job description
- Output: GPT response like:
  "This job is likely a scam. It asks for money and has no verifiable company info."

Used to:

- Prevent job submission
- Trigger email alert
- Add to DB for later reference