TalentScout – AI-Powered Candidate Pre-Screening System

1. Problem Statement

Traditional candidate screening is time-consuming, inconsistent, and highly dependent on manual effort from HR teams. Recruiters often spend significant time collecting basic candidate information and evaluating initial technical suitability before moving to interviews.

There is a need for an **automated, structured, and intelligent pre-screening system** that:

- Collects candidate details in a guided manner
- Evaluates technical understanding based on role and skills
- Stores responses for structured review
- Improves efficiency while maintaining fairness and consistency

2. Project Overview

TalentScout is an AI-powered pre-screening web application designed to automate the initial screening phase for technical roles. It uses a conversational chatbot interface to gather candidate details and assess technical competency through progressively challenging questions.

The system dynamically adapts questions based on:

- Candidate's desired role
- Experience level
- Declared tech stack

All responses are summarized and presented in a structured **summary view**, allowing both candidates and reviewers to understand the screening outcome clearly.

3. Key Functionalities Considered & Implemented

3.1 Candidate Information Collection

- Full Name
- Desired Role
- Location
- Experience Level (Fresher / Experienced)
- Tech Stack (comma-separated)

The chatbot guides candidates step-by-step, ensuring no information is skipped.

3.2 AI-Driven Technical Evaluation

- Three-stage technical questioning:
  - **Conceptual Question**
  - **Practical Question**
  - **Real-World Scenario Question**
- Questions are dynamically generated using AI based on:
  - Role
  - Experience
  - Tech stack

Difficulty increases progressively to fairly assess depth of understanding.

3.3 Response Capture & Storage

- Each question and corresponding candidate answer is stored

- Responses are saved in a structured JSON format
- Timestamp added to maintain submission context

This allows clear traceability and evaluation.

3.4 Screening Completion & Summary View

- Once all technical questions are answered:
    - Screening session is completed
    - Responses are saved
    - A confirmation message is shown

A **Summary Page** displays:

- Candidate details
- Tech stack
- All technical questions with answers

This summary can be viewed post-screening for transparency.

3.5 Navigation & User Experience

- Clean homepage with clear call-to-action
- Top navigation enables:
    - Starting a new screening
    - Viewing screening summary
- Clear progress indicators enhance usability

4. Tech Stack Used

Frontend

- **Next.js (App Router)** – Routing & server integration
- **React** – Component-based UI
- **TypeScript** – Type safety and maintainability
- **Tailwind CSS** – Responsive and modern UI styling
- **shadcn/ui** – Consistent UI components

Backend

- **Next.js API Routes**
- **Node.js File System (fs)** – JSON-based data storage

AI Integration

- **Google Gemini API**
- Used for generating contextual technical questions

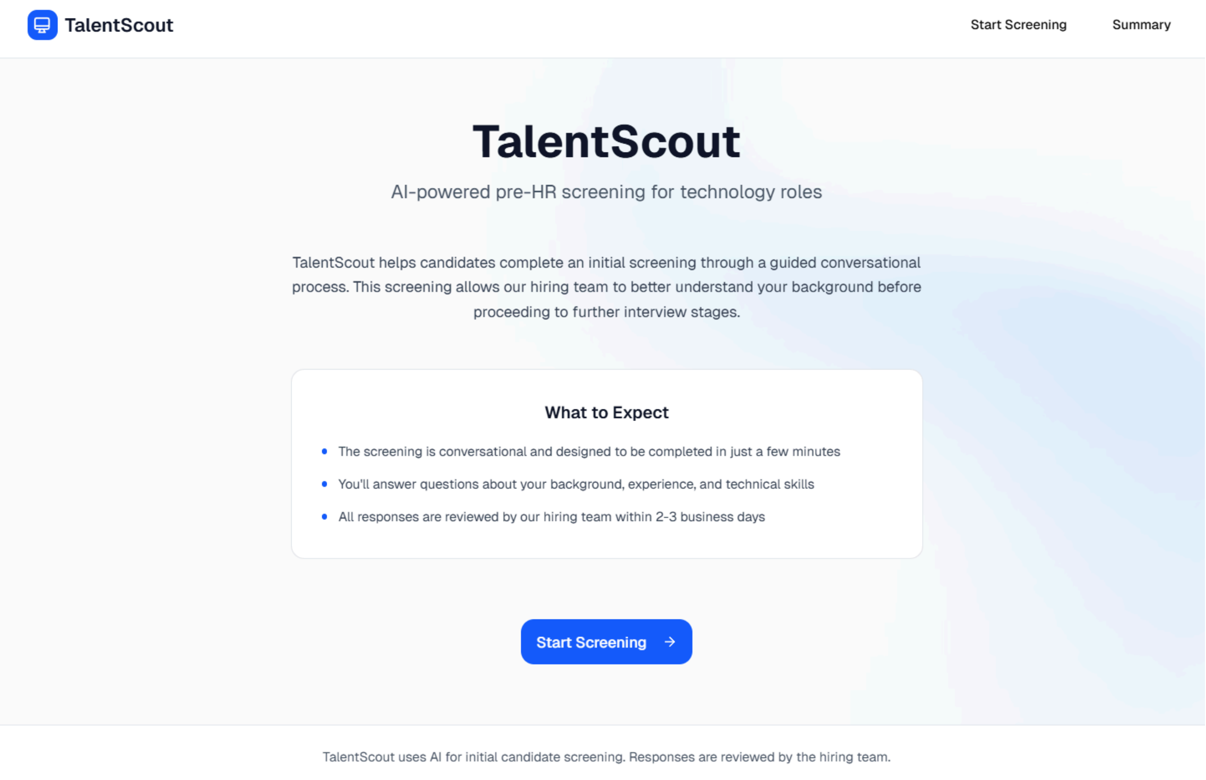
Deployment

- **Vercel**
- Environment variables managed securely via Vercel dashboard

5. User Flow

Step 1: Home Page

- User lands on the TalentScout homepage
- Clicks **“Start Screening”**



TalentScout

AI-powered pre-HR screening for technology roles

TalentScout helps candidates complete an initial screening through a guided conversational process. This screening allows our hiring team to better understand your background before proceeding to further interview stages.

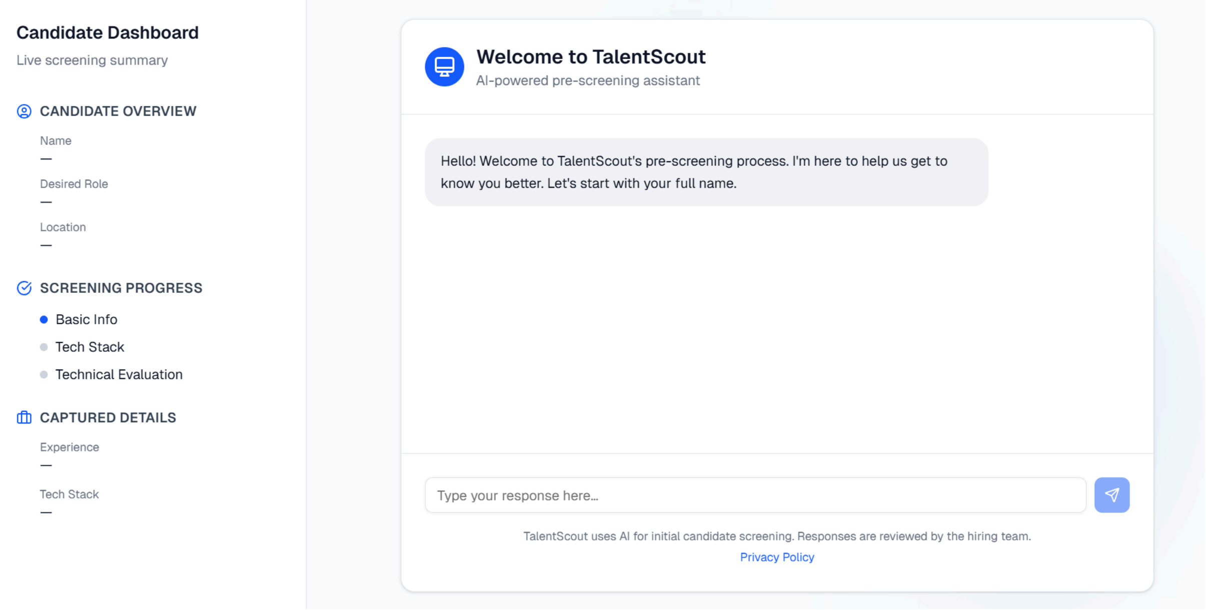
What to Expect

- The screening is conversational and designed to be completed in just a few minutes
- You'll answer questions about your background, experience, and technical skills
- All responses are reviewed by our hiring team within 2-3 business days

TalentScout uses AI for initial candidate screening. Responses are reviewed by the hiring team.

Step 2: Candidate Screening (Chatbot)

- Candidate enters details conversationally
- Chatbot asks role-specific technical questions
- Progress indicator updates dynamically



Candidate Dashboard

Live screening summary

CANDIDATE OVERVIEW

Name

—

Desired Role

—

Location

—

SCREENING PROGRESS

- Basic Info
- Tech Stack
- Technical Evaluation

CAPTURED DETAILS

Experience

—

Tech Stack

—

Welcome to TalentScout

AI-powered pre-screening assistant

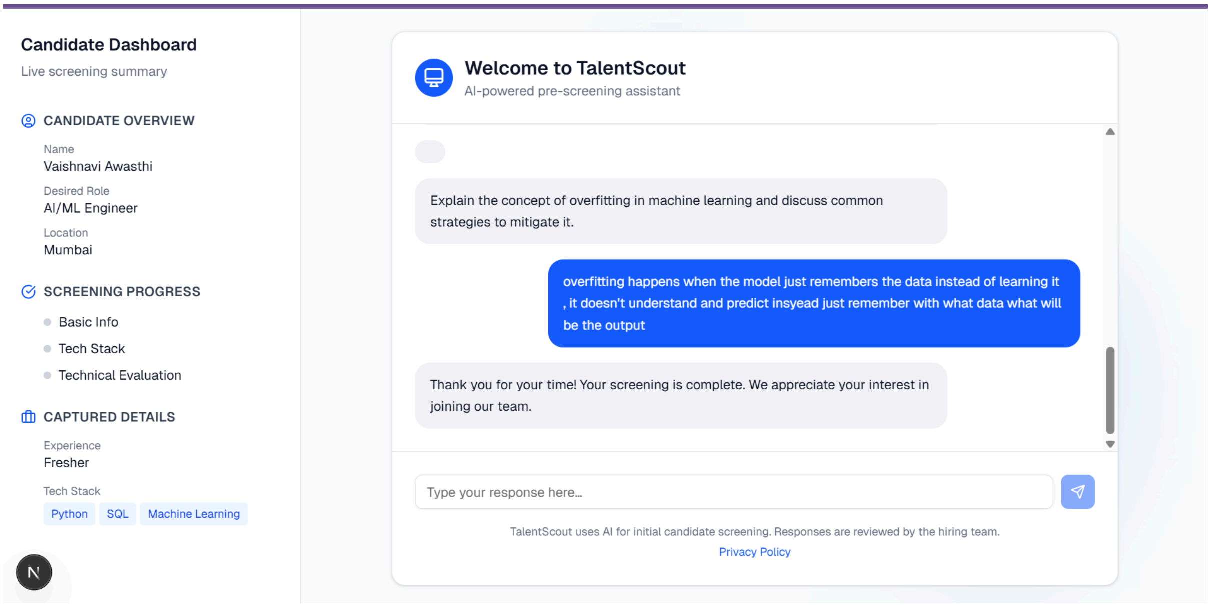
Hello! Welcome to TalentScout's pre-screening process. I'm here to help us get to know you better. Let's start with your full name.

TalentScout uses AI for initial candidate screening. Responses are reviewed by the hiring team.

Privacy Policy

Step 3: Screening Completion

- System saves responses
- Candidate receives confirmation message



Candidate Dashboard

Live screening summary

CANDIDATE OVERVIEW

Name

Vaishnavi Awasthi

Desired Role

AI/ML Engineer

Location

Mumbai

SCREENING PROGRESS

- Basic Info
- Tech Stack
- Technical Evaluation

CAPTURED DETAILS

Experience

Fresher

Tech Stack

Python SQL Machine Learning

Welcome to TalentScout

AI-powered pre-screening assistant

Explain the concept of overfitting in machine learning and discuss common strategies to mitigate it.

overfitting happens when the model just remembers the data instead of learning it , it doesn't understand and predict insyead just remember with what data what will be the output

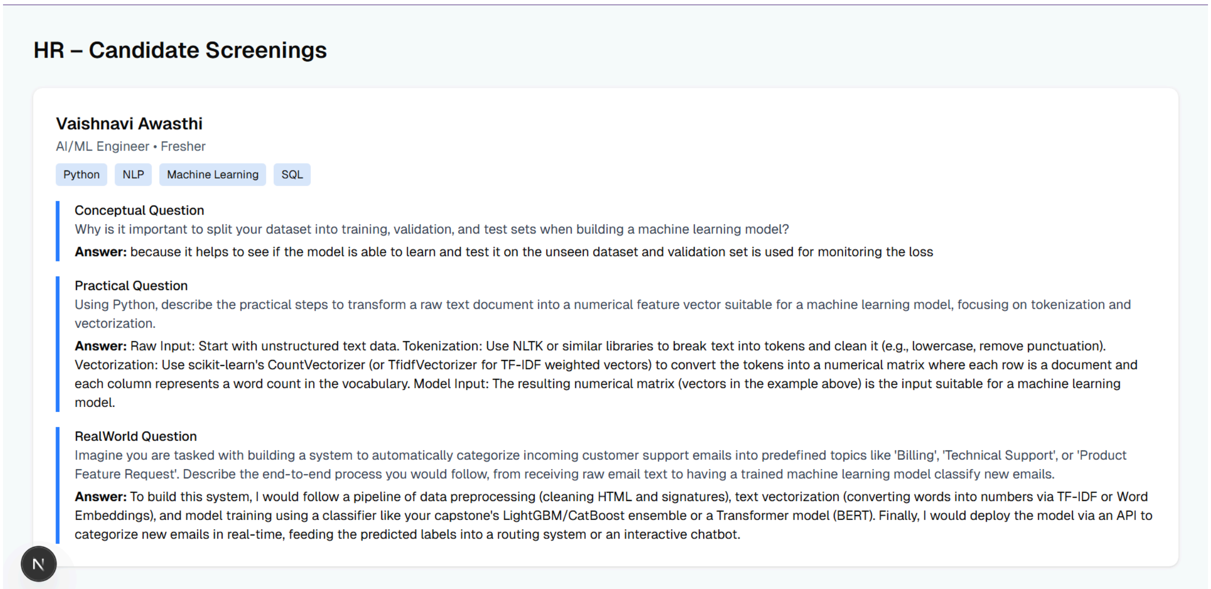
Thank you for your time! Your screening is complete. We appreciate your interest in joining our team.

TalentScout uses AI for initial candidate screening. Responses are reviewed by the hiring team.

Privacy Policy

Step 4: Summary View

- Candidate or reviewer can view:
  - Candidate details
  - Tech stack
  - Questions & answers



HR – Candidate Screenings

Vaishnavi Awasthi

AI/ML Engineer • Fresher

Conceptual Question

Why is it important to split your dataset into training, validation, and test sets when building a machine learning model?

Answer: because it helps to see if the model is able to learn and test it on the unseen dataset and validation set is used for monitoring the loss

Practical Question

Using Python, describe the practical steps to transform a raw text document into a numerical feature vector suitable for a machine learning model, focusing on tokenization and vectorization.

Answer: Raw Input: Start with unstructured text data. Tokenization: Use NLTK or similar libraries to break text into tokens and clean it (e.g., lowercase, remove punctuation). Vectorization: Use scikit-learn's CountVectorizer (or TfidfVectorizer for TF-IDF weighted vectors) to convert the tokens into a numerical matrix where each row is a document and each column represents a word count in the vocabulary. Model Input: The resulting numerical matrix (vectors in the example above) is the input suitable for a machine learning model.

RealWorld Question

Imagine you are tasked with building a system to automatically categorize incoming customer support emails into predefined topics like 'Billing', 'Technical Support', or 'Product Feature Request'. Describe the end-to-end process you would follow, from receiving raw email text to having a trained machine learning model classify new emails.

Answer: To build this system, I would follow a pipeline of data preprocessing (cleaning HTML and signatures), text vectorization (converting words into numbers via TF-IDF or Word Embeddings), and model training using a classifier like your capstone's LightGBM/CatBoost ensemble or a Transformer model (BERT). Finally, I would deploy the model via an API to categorize new emails in real-time, feeding the predicted labels into a routing system or an interactive chatbot.

6. Privacy & Data Handling Considerations

- No authentication implemented (assignment scope)
- Screening data stored locally in JSON format
- System can be configured to:
  - Store only the latest screening
  - Reset data per session
- Designed to avoid unnecessary exposure of candidate data

7. Limitations & Future Enhancements

Current Limitations

- No authentication or role-based access
- JSON-based storage (not scalable for production)
- No scoring or ranking system

Future Improvements

- Authentication for candidates and reviewers
- Database integration (PostgreSQL / MongoDB)
- Automated scoring and shortlisting
- Admin dashboard for HR analytics
- Multi-session support with privacy controls

8. Conclusion

TalentScout successfully demonstrates how AI can streamline and standardize the candidate pre-screening process. By combining conversational UI, dynamic AI-generated questions, and structured response storage, the system reduces manual effort while improving screening quality.

This project showcases:

- End-to-end system design
- AI integration in real workflows
- Clean frontend-backend separation
- Practical problem-solving aligned with real hiring scenarios