**Automated Talent Screening Service (ATSS)**

An all-in-one **talent evaluation** web application built to streamline immigration screening processes for prospective clients. This tool enables an **immigration firm** to identify and classify skilled talent at a glance, improving decision-making for case acceptance and service offerings.

**Table of Contents**

1. [Overview](#bookmark=id.gjdgxs)
2. [Features](#bookmark=id.30j0zll)
3. [Installation & Setup](#bookmark=id.1fob9te)
4. [Usage Guide](#bookmark=id.3znysh7)
5. [Architecture & Code Structure](#bookmark=id.2et92p0)
6. [Deployment](#bookmark=id.tyjcwt)
7. [Contributing](#bookmark=id.3dy6vkm)
8. [License](#bookmark=id.1t3h5sf)

**Overview**

The **Automated Talent Screening Service (ATSS)** was designed specifically for **immigration firms** to **evaluate incoming candidate profiles** - often potential immigrants who seek professional assistance. By cataloging and filtering important data points such as **role**, **experience**, **skill rating**, **location**, and **education**, the platform provides an **at-a-glance classification** of each candidate’s skill level. This classification is helpful in determining whether to accept or decline immigration cases based on prospective career strength, projected outcomes, or specialized skill sets.

**Key Objectives**

* **Efficient Screening**: Quickly see who’s “High Skilled,” “Average Skilled,” or “Needs Improvement.”
* **Holistic Overview**: Access a detailed analysis of each candidate, including experience, skill rating, location, and more.
* **Data-Driven Decision**: Help immigration firms decide which clients to onboard for more reliable case success.

**Features**

1. **Candidate List Page**
   * Search and filter candidates by name, role, location, and a skill classification (“High Skilled,” “Average Skilled,” “Needs Improvement”).
   * A quick overview of **experience**, **skill rating**, and **education** for each candidate.
2. **Candidate Analysis Page**
   * **One-click transition** from the list page to a detailed analysis view.
   * Presents an in-depth breakdown of the candidate’s qualifications, including a classification label for fast decisions.
3. **Add New Candidate Modal**
   * A user-friendly form to add new candidate information (role, experience, education, etc.) in a single step.
   * Instantly updates the local data without reloading the entire application.
4. **Summary Dashboard (Optional)**
   * An optional analytics dashboard (using react-chartjs-2) for deeper insights.
   * Display charts for classification distribution, average skill ratings, location breakdowns, and more.
5. **Responsive & Animated UI**
   * **Tailwind CSS** for streamlined styling.
   * **Framer Motion** for smooth transitions and animations.
   * **shadcn/ui** for modern UI components, such as cards, buttons, and dialogs.

**Installation & Setup**

1. **Clone the Repository**

bash

CopyEdit

git clone https://github.com/your-org/atss.git

cd atss

1. **Install Dependencies**  
   Ensure you have **Node.js** and **npm** (or **yarn**) installed, then run:

bash

CopyEdit

npm install

or

bash

CopyEdit

yarn

This command installs all required libraries, such as react-chartjs-2, framer-motion, lucide-react, and shadcn/ui.

1. **Configure Tailwind**  
   Make sure you have a tailwind.config.js and proper PostCSS setup if it’s not already included. For Next.js projects, see the official Next.js + Tailwind docs.
2. **Run the Development Server**

bash

CopyEdit

npm run dev

or

bash

CopyEdit

yarn dev

Visit http://localhost:3000 to view the ATSS application in your browser.

**Usage Guide**

Below is a typical **workflow** for an immigration officer or a staff member:

1. **Access the Candidate List**
   * A high-level overview of all candidates: roles, experience, skill ratings.
   * Use the **search bar** to locate a candidate by name or location.
   * Filter by “High Skilled,” “Average Skilled,” or “Needs Improvement.”
2. **Add a New Candidate**
   * Click the **“Add New Candidate”** floating button.
   * Fill in the candidate’s name, role, experience, skill rating, location, and education in the pop-up modal.
   * Click “Add Candidate” to store them locally for review.
3. **Perform a Detailed Analysis**
   * Click on a candidate’s card to view the **Candidate Analysis** page.
   * This provides deeper insights (total experience, skill classification, educational background).
   * If the classification or skill rating meets your firm’s thresholds, consider accepting the case.
4. **Optional: Access the Summary Dashboard**
   * If you’ve integrated the optional analytics page, open the “Analysis Dashboard” to see bar charts, doughnut charts, and line graphs.
   * Evaluate **overall distribution** of skill classifications and average experience across all candidates.

**Architecture & Code Structure**

This application’s code is organized into **modular React components**, ensuring a clean separation of concerns:

* **AutomatedTalentScreen.tsx**: The main container orchestrating the “list” vs. “analysis” pages.
* **CandidatesList.tsx**: Displays a grid of candidate cards. Handles searching and filtering.
* **CandidateAnalysis.tsx**: Focused view of an individual candidate’s data.
* **AddCandidateModal.tsx**: A form in a dialog for creating new candidate entries.
* **data.ts** and **classifyCandidate** function: Provide data and classification logic.
* **Optional**: **AnalysisDashboard.tsx** for advanced analytics using charts (Bar, Doughnut, Line).

The application leverages:

* **Next.js or Create React App** (whichever you prefer),
* **Tailwind CSS** for styling,
* **shadcn/ui** for UI components (Cards, Buttons, Dialog, etc.),
* **framer-motion** for animations,
* **lucide-react** for icons,
* **react-chartjs-2** / Chart.js for optional data visualization.

**Deployment**

1. **Build for Production**

bash

CopyEdit

npm run build

or

bash

CopyEdit

yarn build

This command compiles your project into an optimized production bundle.

1. **Deploy to a Hosting Platform**
   * **Vercel**: Recommended for Next.js apps.
   * **Netlify**: Also works well with React-based projects.
   * **AWS S3 / CloudFront or Azure**: For static hosting after building.
2. **Environment Variables** (Optional)  
   If you plan to connect to an external backend or store advanced configuration, place these in a .env file and pass them to your build environment.

**Contributing**

We welcome contributions to improve the ATSS application—whether adding new classification criteria, integrating a real backend database, or refining the UI. Please open an **Issue** or create a **Pull Request** with detailed information.

When contributing, ensure code follows our **ESLint** & **Prettier** guidelines (if applicable).