🚀 AI-Powered Applicant Tracking System (ATS) – Project Report

1. Project Name

AI ATS Tool – Resume and Job Description Matcher

2. Project Description

This AI-based Applicant Tracking System (ATS) project helps automate the resume screening process by matching resumes against job descriptions and providing a match score. It replicates a simplified version of real-world ATS platforms like Greenhouse or Taleo, but customized for educational, prototype, or small business use cases.

- Sign up and log in
- Upload resumes and job descriptions
- View real-time match scores
- Store and retrieve historical match data (SQLite)

3. My Role

I served as the sole developer and architect of this project. My responsibilities included:

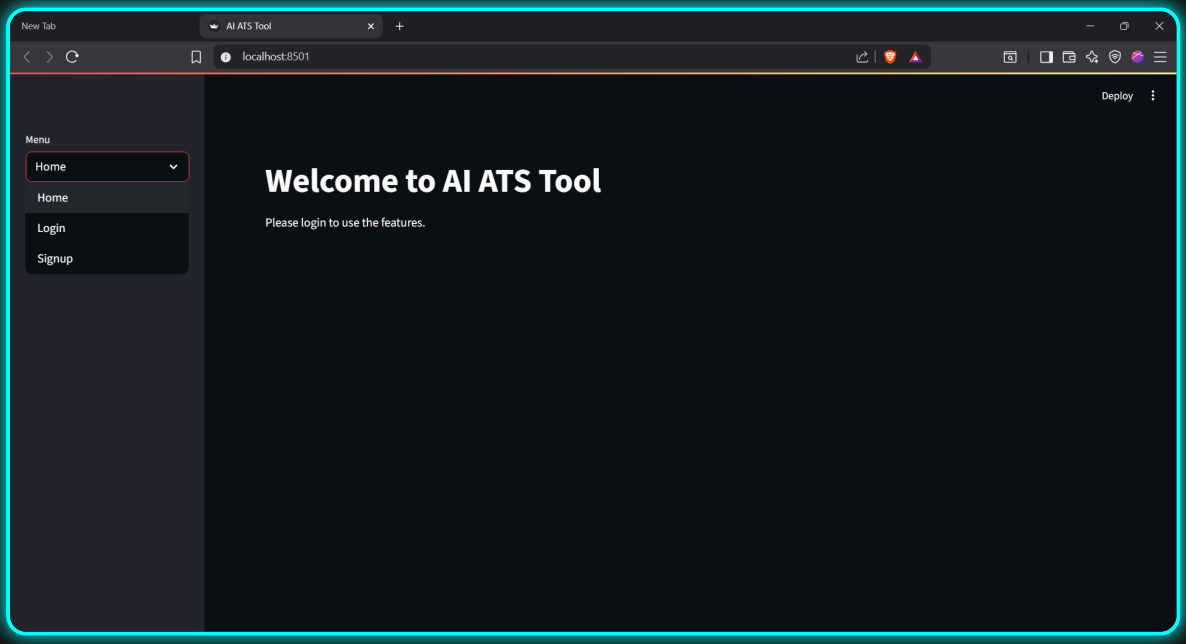
- Designing the UI & user flow using Streamlit
- Implementing login/signup authentication
- Developing resume and JD input functionality
- Creating NLP-based matching logic
- Integrating SQLite database
- Ensuring smooth UI-backend communication

4. Tools & Technologies Used

- **Languages:** Python
- **Frontend:** Streamlit
- **Backend:** Python, SQLite3
- **Libraries:** difflib, PyPDF2, sqlite3, os, datetime
- **Structure:**
  - app.py
  - auth.py
  - database.py
  - matcher.py
  - parser.py
  - email_handler.py
  - utils.py

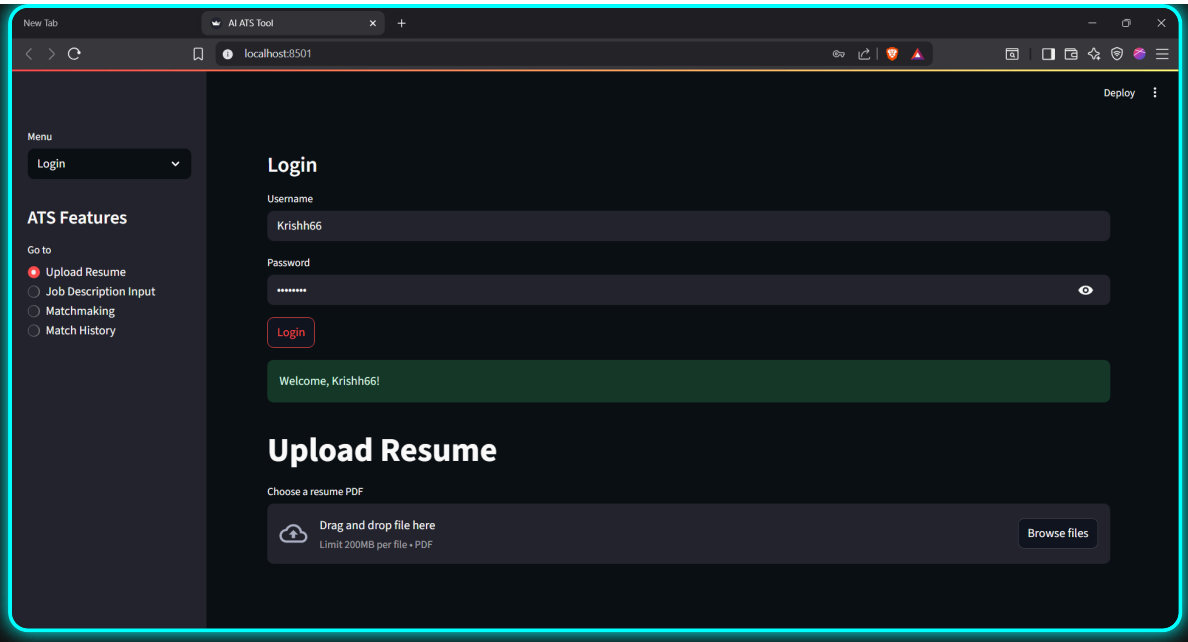
5. Key Features Implemented

🔒 User Authentication (Login/Signup)



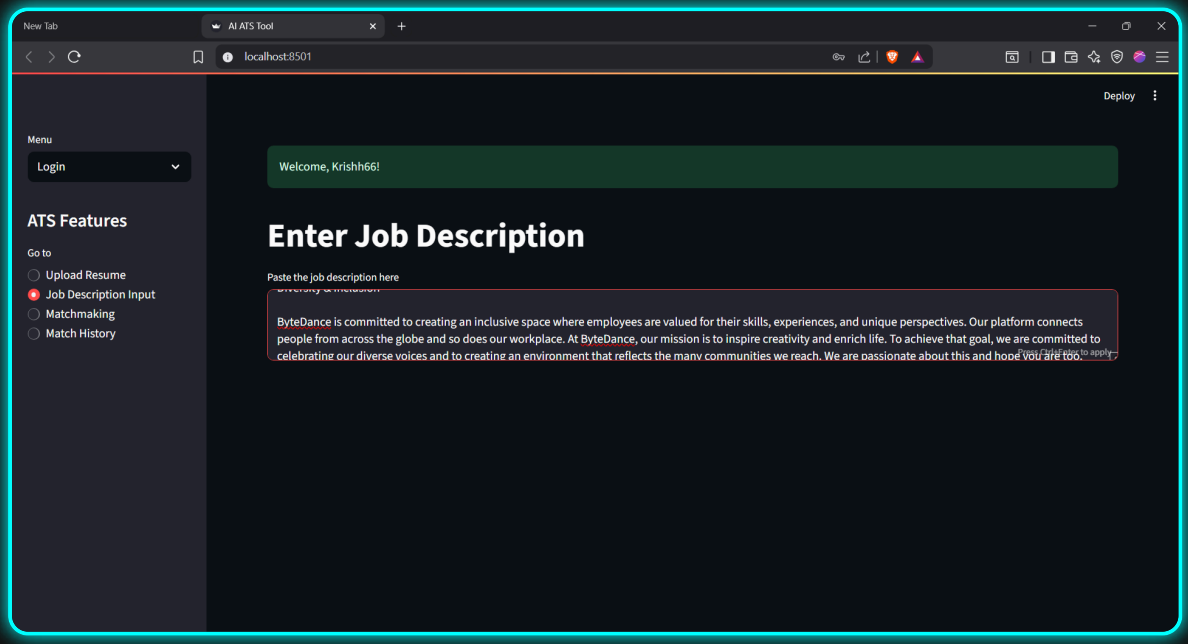
Secure user registration and login flow using Streamlit session state and SQLite.

📄 Resume Upload



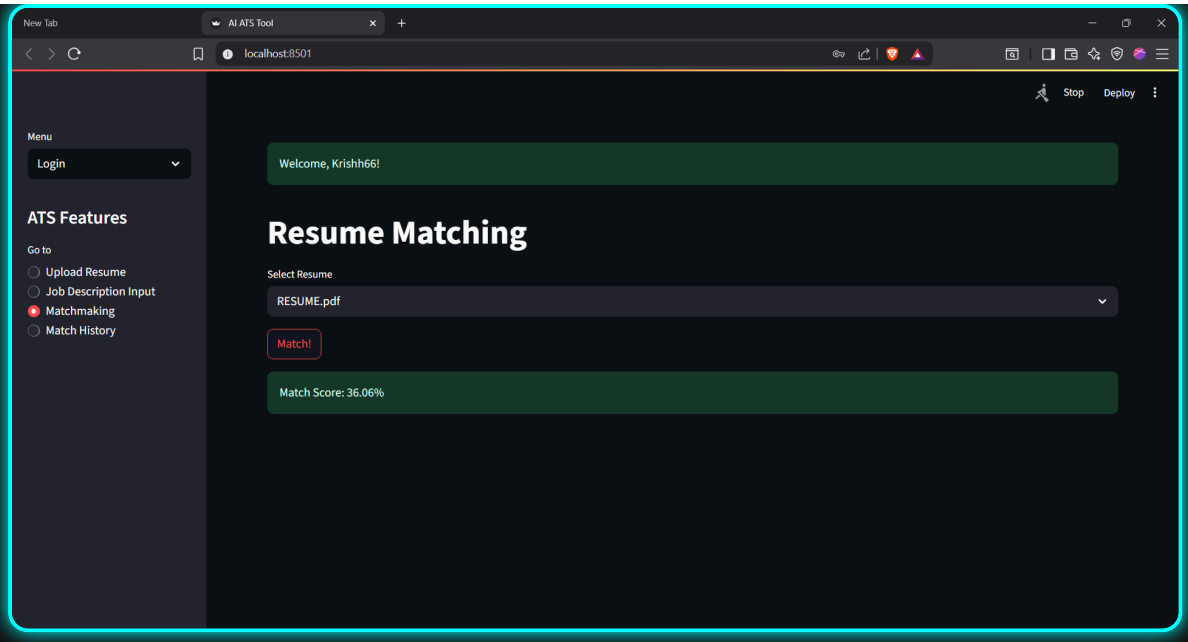
Users can upload their resume as a PDF for parsing and processing.

📝 Job Description Input



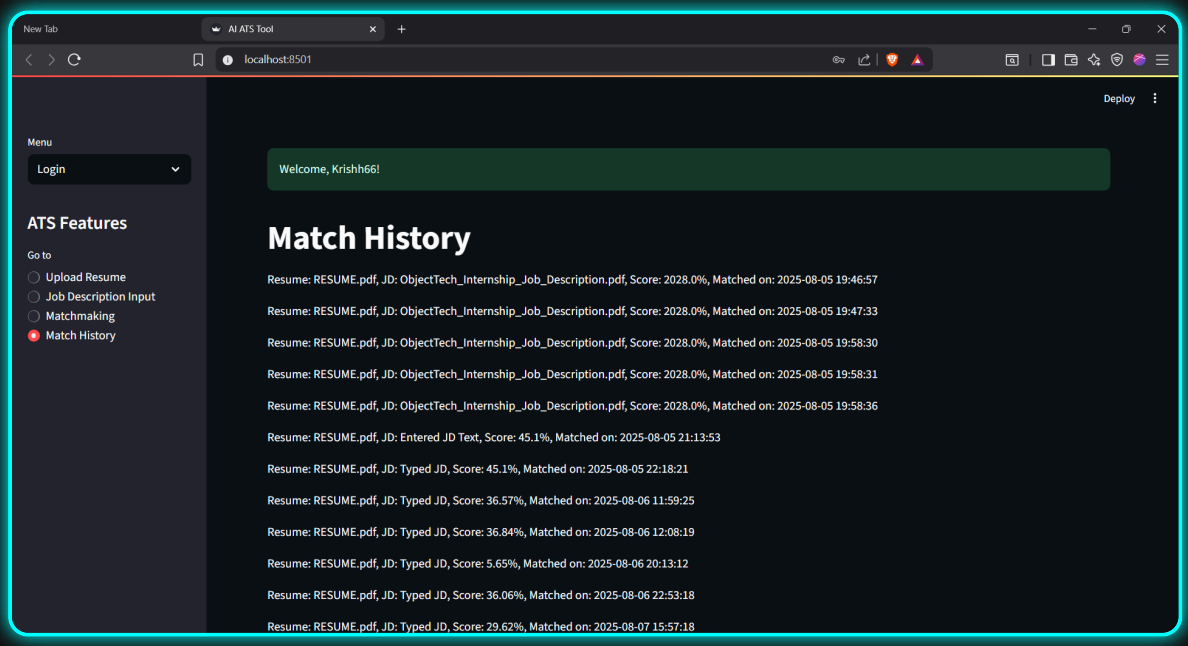
Users input the job description directly in a text field for real-time matching.

📊 Real-Time Matching Score



The system compares resume content with the job description using SequenceMatcher and displays a percentage match score instantly.

📓 Match History Logging



All matches are logged into the database with timestamp and username, viewable by the user.

6. Challenges Faced & Solutions

- **PDF reading delays:** Solved by accepting job description text input directly.
- **Match logging failures:** Fixed parameters and ensured robust DB logic.
- **Missing table errors:** Ensured tables are created at startup.

- **Modular confusion:** Reorganized files for better readability and maintainability.

7. Conclusion

This ATS project proves that an individual can build a fully functional, AI-powered application using simple yet powerful tools. It's scalable, extendable, and can be enhanced with advanced NLP techniques, dashboards, or cloud integrations.