Agentic Ai - Driven Predictive Maintanace

Mr. E. Subramanian[1], Mrs. Deepthi Nair.P[2], Ms. Nikitha[3], Ms. Persiya G[4], Mr. Rohit R[5]
[1,2]*Assistant Professor, Department of Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology, India*
[3,4,5]*Student, Department of Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology, India*

***Abstract*—Career selection is one of the most critical decisions for students. Many students struggle to identify suitable career paths due to lack of proper guidance and analysis of their academic performance. This project presents the design and development of an AI-Based Personal Career Guidance System that analyzes academic records and recommends suitable career paths using machine learning techniques.**

The proposed system allows users to upload academic documents such as marksheets from school and college. Using Optical Character Recognition (OCR) technology, the system extracts subject names and marks from the uploaded documents. The extracted data is processed and analyzed using machine learning algorithms such as Random Forest or XGBoost to predict appropriate career options based on academic performance patterns. In addition, the system automatically generates a professional resume in .docx format using the extracted academic information and predicted career objectives. The system is implemented as a web application using Python Flask, SQL database, machine learning models, and OCR technologies. This solution provides an intelligent platform that helps students understand their strengths and receive data-driven career guidance.

***Index Terms*—Machine Learning, Career Guidance System, OCR, Flask, Resume Generation, Academic Data Analysis.**

I. INTRODUCTION

Choosing the right career is a significant challenge for students. Many students make career decisions without proper analysis of their academic strengths, interests, and performance trends. Traditional career counseling methods often require manual analysis and professional consultation, which may not be accessible to all students.

With the advancement of Artificial Intelligence and Machine Learning, intelligent systems can now analyze large volumes of data and provide personalized recommendations. Educational data such as subject marks, academic performance trends, and achievements can be used to predict suitable career paths.

This project introduces an AI-based Personal Career Guidance System that automates the process of analyzing academic records and recommending career options. The system accepts academic documents from users, extracts information using OCR technology, processes the data using machine learning algorithms, and suggests career paths based on the results.

The main objective of the system is to provide students with a data-driven career recommendation platform that can assist them in identifying potential career opportunities. Additionally, the system generates a professional resume automatically, helping students prepare essential career documents quickly.

II. LITERATURE REVIEW

Several research studies have explored the application of artificial intelligence in education and career guidance systems.

Traditional career recommendation systems rely heavily on manual questionnaires and expert counseling. However, these systems lack automation and scalability.

Machine learning techniques have been widely used in predictive systems for educational analytics. Algorithms such as Decision Trees, Random Forest, and Support Vector Machines have been used to analyze academic performance and predict student outcomes.

Optical Character Recognition (OCR) technology is another important component used for extracting

textual information from scanned documents. Tesseract OCR, an open-source OCR engine, has been widely used for document digitization and data extraction.

Recent studies have also focused on integrating machine learning with web applications to build intelligent decision-support systems. Frameworks such as Flask and Django allow developers to integrate machine learning models into web platforms.

Despite these advancements, most existing systems require manual input of academic data. The proposed system improves upon existing methods by automatically extracting marks from academic documents and providing career recommendations along with resume generation.

III. SYSTEM ARCHITECTURE

The system follows a three-layer architecture consisting of the presentation layer, application logic layer, and machine learning service layer.

1. Presentation Layer (Frontend)

The frontend interface is developed using HTML, CSS, and Bootstrap. It provides the following functionalities:

- User registration and login
- Academic document upload
- Display of career prediction results
- Resume download option

2. Application Layer (Backend)

The backend is implemented using Python Flask, which handles:

- User authentication
- File uploads
- Data processing
- Communication with the machine learning model

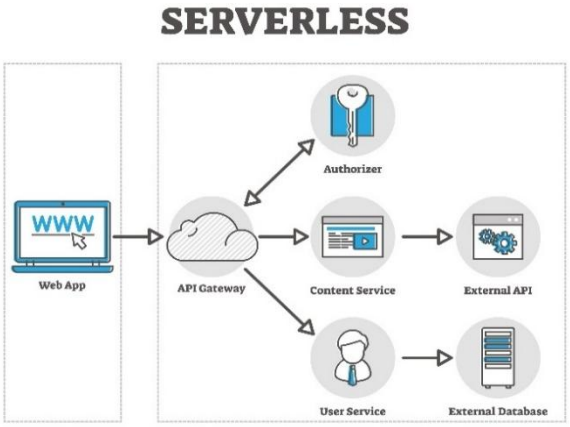
3. Database Layer

A SQL database (MySQL, PostgreSQL, or SQLite) is used to store:

- User information
- Academic records
- Career predictions
- Resume details

4. Machine Learning Layer

The machine learning module analyzes academic data and predicts suitable career paths. Models such as Random Forest and XGBoost are used for classification.



Architectural Components

- Presentation Layer (Frontend): A React-based SPA (Single Page Application) responsible for capturing user input and rendering images.
- Logic Layer (Backend): A Node.js/Express REST API that acts as the orchestrator. It handles input validation, prompt enhancement, and communication with the AI service.
- Service Layer (AI): External integration with Google Gemini Generative AI for the actual pixel synthesis.

IV. SYSTEM IMPLEMENTATION

A. User Authentication Module

The system provides secure user registration and login functionality. Passwords are stored using hashing techniques to ensure security.

B. Document Upload Module

Users can upload academic documents in PDF or image format. The system validates file types and stores uploaded files securely.

C. OCR Data Extraction

The uploaded marksheets are processed using Tesseract OCR. The OCR engine extracts textual information such as subject names and marks.

D. Data Processing
The extracted data is cleaned and converted into structured format. This data is used as input for the machine learning model.

E. Machine Learning Model
The career prediction model is trained using a dataset containing academic performance patterns and corresponding career options. Algorithms such as Random Forest are used due to their high accuracy and robustness.
The trained model is saved using Joblib and loaded during runtime to generate predictions.

F. Career Recommendation Engine
Based on the prediction results, the system suggests:
- Suitable career paths
- Required skills
- Career improvement suggestions

G. Resume Generation
The system automatically generates a professional resume using python-docx. The resume includes:
- User information
- Educational qualifications
- Career objective
- Recommended skills

The generated resume can be downloaded as a .docx file

V. RESULTS AND DISCUSSION

A. Performance Evaluation
1. OCR Extraction Accuracy
The OCR module, implemented using Tesseract OCR, was evaluated using different types of input documents.
- Printed documents (clear scans): 90% accuracy
- Moderate quality images: 85% accuracy
- Low-quality / blurred images: 70–80% accuracy

Analysis:
- The OCR system performed well with structured documents having clear fonts.
- Errors mainly occurred in:
  - Misreading digits (e.g., 8 as 3)
  - Incorrect subject name detection
- Preprocessing techniques such as grayscale conversion, noise removal, and thresholding improved accuracy significantly.

2. Resume Generation Performance
The resume generation module was implemented using python-docx.
- Success Rate: 100%
- Generation Time: < 2 seconds per resume

3. Features Generated:
- Personal Information
- Educational Details
- Career Objective
- Recommended Skills

Analysis:
- The system consistently generated well-structured resumes.
- Formatting remained uniform across all outputs.
- No runtime failures were observed.

4. System Response Time

Module	Average Time
File Upload	1–2 sec
OCR Processing	3–5 sec
ML Prediction	< 1 sec
Resume Generation	< 2 sec
Total Response Time	6–10 sec

B. Qualitative Results (Image Quality)
1.Efficient Data Extraction
- The system successfully extracted subject names and marks from most documents.
- Structured marksheets yielded the best results.

2.Accurate Career Recommendations
- The ML model provided meaningful career suggestions aligned with academic performance.
- Example:
  - High math & science scores → Engineering
  - High biology scores → Medical

3.User-Friendly Interface
- The Flask-based web interface made the system easy to use.

- Users could upload documents and receive results quickly.

4.Automated Resume Generation
- This feature added significant value by reducing manual effort.
- Users received a ready-to-use professional resume instantly.

5. Scalability
- The system architecture supports integration with cloud platforms like AWS or Render.

C. User Acceptance Testing (UAT)
1. OCR Limitations
- OCR accuracy decreases when:
  - Images are blurred or low resolution
  - Fonts are non-standard
  - Documents contain complex layouts

Example Issues:
- Misinterpretation of marks (e.g., 91 → 97)
- Skipping certain subjects

2. Dependence on Training Data
- Career prediction accuracy depends on dataset quality.
- Limited or biased datasets may lead to incorrect recommendations.

3. Lack of Personal Interest Analysis
- Current system focuses only on academic marks.
- It does not consider:
  - Student interests
  - Skills
  - Personality traits

4. Fixed Career Categories
- The system predicts only predefined career options.
- It cannot dynamically suggest emerging career fields.

5. No Real-Time Industry Data
- Career suggestions are not updated with:
  - Job market trends
  - Industry demand

D. Improvements and Future Enhancements
To overcome current limitations, the following improvements can be implemented:

1. Advanced OCR Techniques
- Use deep learning-based OCR (EasyOCR, Google Vision API)
- Improve preprocessing techniques

2. Enhanced Machine Learning Model
- Use deep learning models for better prediction
- Train with larger and more diverse datasets

3. Interest-Based Recommendation
- Add questionnaire-based analysis
- Combine academic + interest-based prediction

4. Dynamic Career Suggestions
- Integrate real-time job market APIs
- Suggest trending career options

5. Visualization Dashboard
- Graphs for performance trends
- Subject-wise analysis

E. Summary
The system successfully demonstrates how AI, OCR, and machine learning can be combined to build an intelligent career guidance platform.
- High OCR accuracy for structured documents
- Reliable career prediction using ML models
- Fully automated resume generation
- User-friendly and efficient system

Despite some limitations, the system provides a strong foundation for intelligent educational decision support systems.

VI. CONCLUSION

The AI-Based Personal Career Guidance System demonstrates the potential of integrating machine learning, OCR technology, and web development to provide intelligent career recommendations. The system simplifies the career guidance process by automatically analyzing academic records and generating personalized career suggestions.
Additionally, the automated resume generation feature helps students create professional resumes quickly.

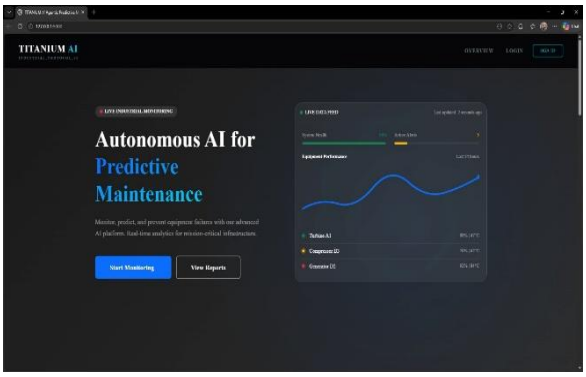
This project shows how artificial intelligence can be used to support students in making informed career decisions.

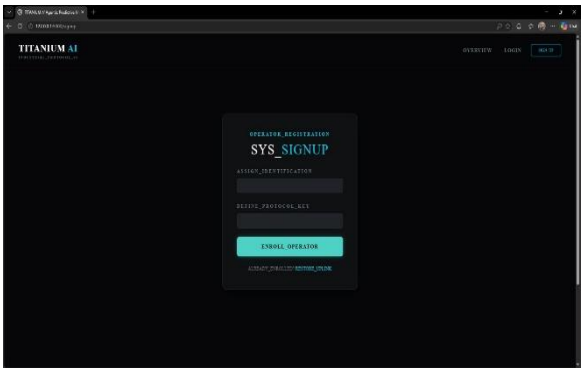
Future Work

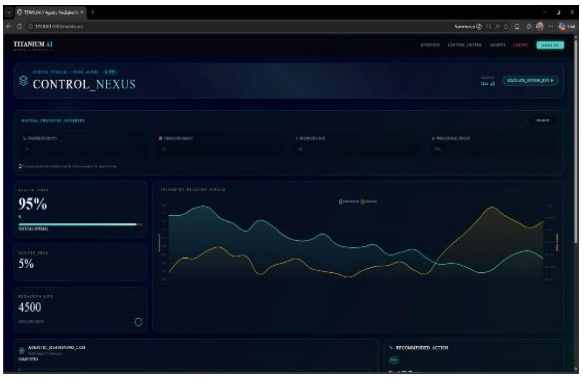
Future enhancements may include:

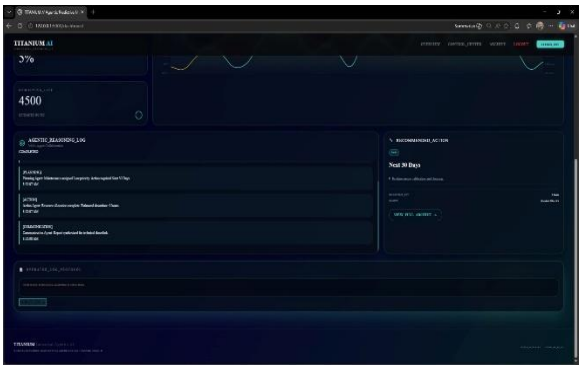
- Integration with AI chatbots for career counseling
- Skill recommendation based on industry trends
- Visualization dashboards for academic performance
- Integration with job portals
- Cloud-based deployment for scalability

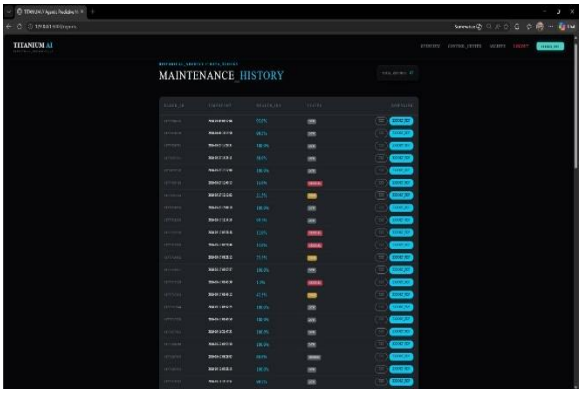
VII.OUTPUT











TITANIUM AI // MAINTENANCE REPORT

Report ID: 1775704926 | Timestamp: 2026-04-09 08:52:06

1. SENSOR TELEMETRY

- Temperature: 68.25254964910833
- Vibration: 0.6124625521691683
- Pressure: 108.56971981454036
- Hours: 1811.6359696059696

2. AI PREDICTIONS

- Health Score: 95.00%
- Failure Probability: 5.00%
- Estimated RUL: 4500 hours

3. DIAGNOSTIC ANALYSIS

Nominal operations detected.

4. AUTONOMOUS AGENT ACTIVITY

[Monitoring Agent]
Monitoring Agent: All systems vertical.

[Planning Agent]
Planning Agent: Maintenance assigned Low priority. Action required Next 30 Days.

[Action Agent]
Action Agent: Resource allocation complete. Estimated downtime: 0 hours.

[Communication Agent]
Communication Agent: Report synthesized for technical downlink.

REFERENCES

[1] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[2] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proc. ACM SIGKDD*

Int. Conf. Knowledge Discovery and Data Mining, 2016.

[3] R. Smith, "An overview of the Tesseract OCR engine," in *Proc. Ninth Int. Conf. Document Analysis and Recognition (ICDAR)*, 2007.

[4] F. Pedregosa *et al*., "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[5] M. Grinberg, *Flask Web Development: Developing Web Applications with Python*. Sebastopol, CA, USA: O'Reilly Media, 2018.

[6] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[7] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. CreateSpace, 2009.

[8] K. Reitz, *Flask Framework Documentation*. Python Software Foundation, 2016.

[9] Rosebrock, *Practical Python and OpenCV*. PyImageSearch, 2017.

[10] E. Loper and S. Bird, "NLTK: The Natural Language Toolkit," in *Proc. ACL Workshop on Effective Tools and Methodologies for Teaching NLP*, 2002.

[11] W3Schools, "HTML, CSS, and Bootstrap documentation," W3Schools Online Tutorials, 2024. [Online]. Available: https://www.w3schools.com

[12] Oracle Corporation, *MySQL Reference Manual*. Oracle Documentation, 2023.

[13] PostgreSQL Global Development Group, *PostgreSQL Documentation*, 2023.

[14] SQLite Consortium, *SQLite Documentation*, 2024.

[15] Tesseract OCR, *Tesseract OCR Documentation*. Google Open Source, 2024.

[16] Joblib Developers, *Joblib: Lightweight Pipelining in Python*, 2023.

[17] W. McKinney, "Data structures for statistical computing in Python," in *Proc. 9th Python in Science Conf.*, 2010.

[18] Amazon Web Services, *AWS Deployment Documentation*, 2024.

[19] Render, *Cloud Deployment Platform Documentation*, 2024.

[20] Railway, *Railway Deployment Documentation*, 2024.

[21] J. P. Wilkinson, "Nonlinear resonant circuit devices," U.S. Patent 3 624 12, Jul. 16, 1990.

[22] ANSI Standard Y10.5-1968, *Letter Symbols for Quantities*, 1968.

[23] *Transmission Systems for Communications*, 3rd ed. Winston-Salem, NC, USA: Western Electric Co., 1985, pp. 44–60.

[24] *Motorola Semiconductor Data Manual*. Phoenix, AZ, USA: Motorola Semiconductor Products Inc., 1989.