Puneeth Jains S

No. 19, 3rd cross, 1st main, 3rd block, Ayyappa Nagara, KR
Puram Bengaluru - 560036
+916364814142 | puneethjains2005@gmail.com
in https://www.linkedin.com/in/puneeth-jain-s-13534132a

Objective

A highly motivated and challenge-driven professional with a strong ability to thrive in dynamic environments. A dedicated team player with proven proficiency in my skills, I am always eager to learn and grow. I continuously seek opportunities to expand my knowledge, driven by passion and a relentless hunger for learning.

Education

- **ITI Central School** 2021
  SSLC
  89.6%
- **Vision PU College** 2023
  PUC
  86%
- **New Horizon College of Engineering** 2025
  BE - AIML
  9.66

Skills

- Languages & Web: Python, JavaScript, C/C++, HTML, CSS, Next.js, Flask, FastAPI
- Data & ML: Scikit-learn, Pandas, NumPy, Matplotlib,Pytorch,tensorflow,SentenceTransformers
- Tools: Docker, Git, Power BI, Jupyter, Postman
- Other: IoT Security (RSA/ECC), REST APIs

Projects

- **StudyBuddy – AI-Based Multi-Agent EdTech Productivity Tool**
  Developed an AI-powered multi-agent study assistant that uses OCR and LLM-based processing to generate instant summaries, flashcards, quizzes, mock tests, and placement preparation content from uploaded notes. Integrated YouTube recommendations and a seamless Next.js frontend with a Python backend to deliver a rich, personalized learning experience.
  Tech Stack:
  - **OCR** – Extracted text from handwritten & printed notes
  - **LangChain + Gemini** – Multi-agent system for summaries, flashcards & quiz generation
  - **Next.js** – Frontend interface for user interaction
  - **Python (FastAPI/Backend)** – Orchestrated agents and processing logic
  - **YouTube API** – Recommended topic-relevant learning videos
  - **Vectorization / NLP Tools** – For optimized text understanding and retrieval

  Role: Designed and implemented the AI multi-agent workflow powering StudyBuddy’s core features.
  Duration: 36 hours

- **Intelligent PDF Document Analyzer**
  Designed a system to extract structured outlines and prioritize key sections from unstructured PDF documents using layout heuristics and semantic similarity scoring.
  Tech Stack:
  - **PyMuPDF** – Parsed PDF layouts to extract spans, font sizes, and position metadata
  - **SentenceTransformers** – Measured semantic similarity between section headings and user-defined “jobto-be-done” contexts
  - **Rule-based Filtering** – Cleaned and ranked extracted headings using layout features (font size, position,casing)
  - **JSON Output Generator** – Produced structured output with heading levels and page references
  - **Docker** – Containerized the pipeline for reproducibility and submission

  Role: Optimized PDF data extraction workflow
  Duration: 1 weeks
  Result: Achieved ~90% accurate outline extraction with strong title detection and reduced noise through hybrid heuristics and span clustering.
- **IoT Device Authentication and Authorization System**
  Built a secure, scalable system to authenticate and authorize IoT devices before they communicate over an MQTT broker. Devices are assigned cryptographic identities and must prove ownership of private keys to participate in data transmission.
  Tech Stack:
  - **Python** – Implemented device logic, key generation, identity verification, and system coordination
  - **RSA & ECC Cryptography (PyCryptodome)** – Used to generate public-private key pairs for each device and verify digital signatures
  - **MQTT** (Mosquitto Broker) – Enabled lightweight, real-time communication between devices in a publishsubscribe model
  - **Docker** – Containerized the broker, registry server, and devices for consistent cross-platform execution

  Role: Implementing the MQTT-based communication pipeline with a Mosquitto broker.
  Duration: 3 months
  Result:Achieved secure, real-time IoT authentication with 100% successful validation of registered devices, strong spoof-protection, and low-overhead ECC/RSA performance.
- **Fake News Detection System**
  Built a machine learning pipeline to classify news articles as fake or real using natural language processing and
  text classification techniques. The system achieved strong performance after evaluation.
  Tech Stack:
  - **Python** – Developed the data pipeline and implemented ML logic
  - **scikit-learn** – Trained models like Logistic Regression, Naive Bayes, and Random Forest
  - **Pandas, NumPy** – Used for data manipulation and preprocessing
  - **TF-IDF Vectorization** – Converted text to meaningful numerical features
  - **Matplotlib & Seaborn** – Visualized confusion matrix and performance metrics

  Role: Created AI model for news verification
  Duration: 1 week
  Result: Achieved 94% accuracy using Logistic Regression.

Interests

- Gaming • Travel • Movies/TV