ALVA'S INSTITUTE OF ENGINEERING AND TECHNOLOGY

A Unit of Alva's Education Foundation (R)

(Affiliated to Visvesvaraya Technological University, Belagavi.

Approved by AICTE, New Delhi & & Recognized by Government of Karnataka)

Shobhavana Campus, Mijar, Moodbidri- 574 225, Mangalore, D.K., Karnataka State.

YEARLY REPORT

ON

PROJECTS AND LEARNING

Report submitted in fulfillment for the Completion of

Projects

for

EDWIN'S LAB

By

KIRANKUMAR S **4AL23IS024**

ACADEMIC YEAR 2024-25

Submitted to

Dr. Aslam B Nandyal

Associate Professor | Learning and Development Lead | Coordinator- Edwin's Lab

Alva's Institute of Engineering and Technology (AIET)

Shobhavana Campus, Mijar, Moodbidri- 574 225, Mangalore, D.K., Karnataka State.

Artificial Intelligence (AI) is the science of building machines capable of performing tasks that normally require human intelligence, such as decision-making, recognition, and problem-solving.

Machine Learning (ML) is a subset of AI where machines learn patterns from data without explicit programming.

Deep Learning (DL) is a subset of ML that uses multi-layered neural networks to automatically extract complex features and make predictions.

Real-life examples:

- AI: Chatbots, self-driving cars
- ML: Spam email detection, stock price prediction
- DL: Face recognition, language translation

Machine Learning (ML)

Machine Learning is the process of teaching computers to learn patterns from data.

Types of ML:

- Supervised Learning: Models trained with labeled data (input → output)
- Unsupervised Learning: Models trained with unlabeled data to find patterns or clusters
- Reinforcement Learning: Agents learn by trial-and-error using rewards/punishments

---

Supervised Learning Algorithms

Linear Regression

- Predicts continuous values
- Equation: $y = \beta0 + \beta1x + \varepsilon$
- Cost function (MSE): $J(\theta) = (1/2m)\ \Sigma\ (h\theta(xi) - yi)^2$
- Applications: Predicting student scores, sales forecasting

Project(liner regression)

The project applies **linear regression** to predict house prices based on features such as area, number of bedrooms, location, and other property attributes. The dataset is preprocessed (handling missing values, encoding categorical data, and scaling). The model learns the

relationship between input features and price by fitting a best-fit line. Performance is evaluated using metrics like **R^2 score, Mean Squared Error (MSE), and Root Mean Squared Error (RMSE)**. Results show that linear regression provides a simple and interpretable baseline for price prediction, though more complex models (e.g., Random Forest or XGBoost) can improve accuracy.

Logistic Regression

Purpose: Used for binary and multiclass classification tasks.

Hypothesis function:

- **Uses Sigmoid function → σ(z) = 1 / (1 + e^(-z)), which maps values into (0,1).**

Decision boundary:

- **If P(y=1|x) ≥ 0.5 → class 1, else class 0.**

Cost function (Log Loss):

- **J(θ) = -(1/m) Σ [yi log(hθ(xi)) + (1-yi) log(1-hθ(xi))]**

Gradient Descent:

- **Used to optimize parameters θ to minimize cost.**

Regularization (L1/L2):

- **Prevents overfitting by penalizing large weights.**

Types:

- **Binary Logistic Regression → two classes (e.g., spam vs. not spam).**
- **Multinomial Logistic Regression (Softmax Regression) → more than two classes.**

Evaluation Metrics:

- **Accuracy, Precision, Recall, F1-score, ROC-AUC.**

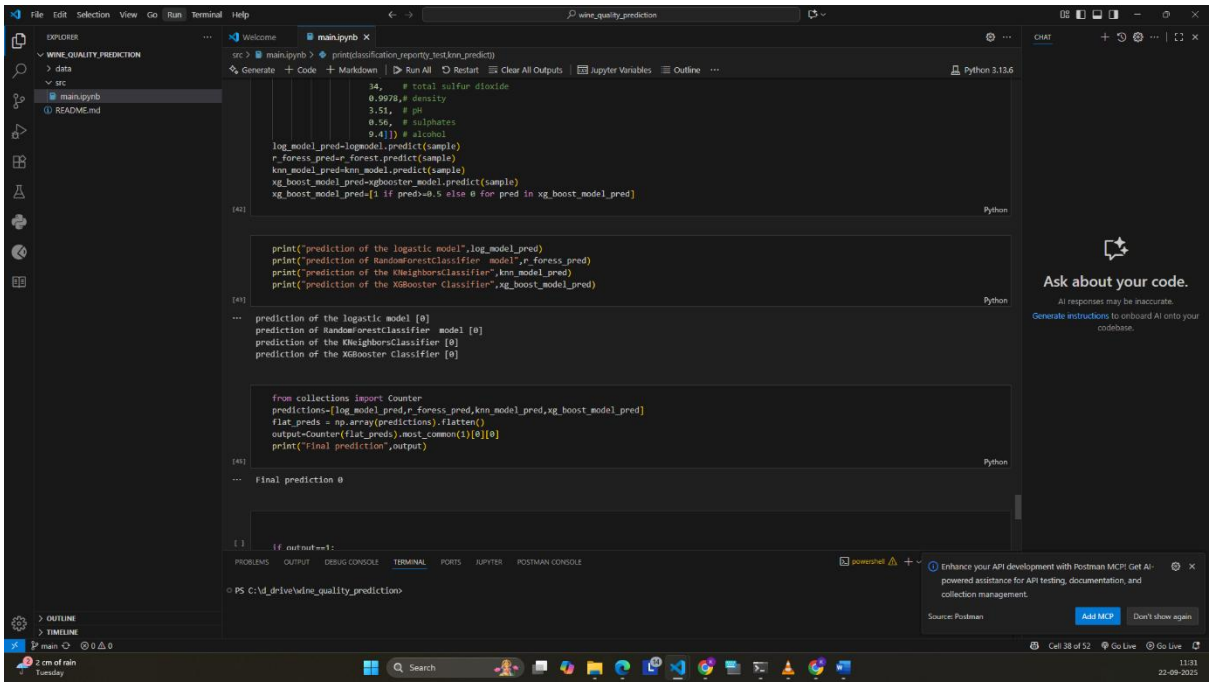
Applications:

- **Email spam detection, medical diagnosis, credit scoring, fraud detection, sentiment analysis.**

Project

Wine quality prediction

The project uses **logistic regression** to predict wine quality (good or bad) based on chemical features. It shows which factors (like alcohol, acidity, and sulphates) most influence quality, achieving around **70–80% accuracy** as a baseline model.



Decision Tree

- Tree-like structure for classification/regression
- Gini Impurity: Gini(p) = 1 - Σ pi^2
- Entropy: H(S) = - Σ pi log2(pi)
- Applications: Customer segmentation, loan approval
- **[Insert diagram/screenshot here]**

K-Nearest Neighbors (KNN)

- Classifies based on nearest neighbors
- Euclidean distance: d(p,q) = sqrt(Σ (pi - qi)^2)

- Applications: Movie recommendations, image classification
- **[Insert diagram/screenshot here]**

Unsupervised Learning Algorithms

Clustering (K-Means)

- Groups similar data points
- Applications: Market segmentation, customer behavior analysis
- **[Insert diagram/screenshot here]**

Dimensionality Reduction (PCA)

- Reduces dataset features while retaining important information
- Applications: Image compression, feature visualization
- **[Insert diagram/screenshot here]**

Reinforcement Learning

- Agents learn to act in an environment to maximize cumulative reward
- Examples: Self-driving cars, game-playing AI
- **[Insert diagram/screenshot here]**

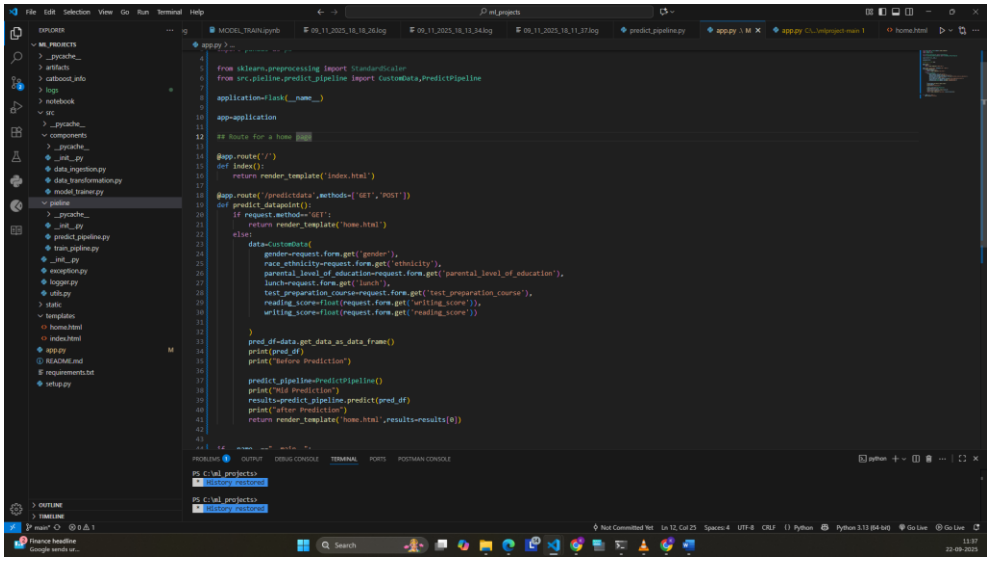
Student Score Prediction Project (End-to-End ML Project)

- Overview: Predict student Maths score using ML regression models based on features like gender, parental education, writing score, and reading score
- Workflow: Data preprocessing → Model training → Flask web app deployment
- Tech Stack: Python, Pandas, Sklearn, Flask, HTML/CSS
- **[Insert UI screenshot here]**
- **[Insert VS Code project structure screenshot here]**
- **[Insert results screenshot here]**

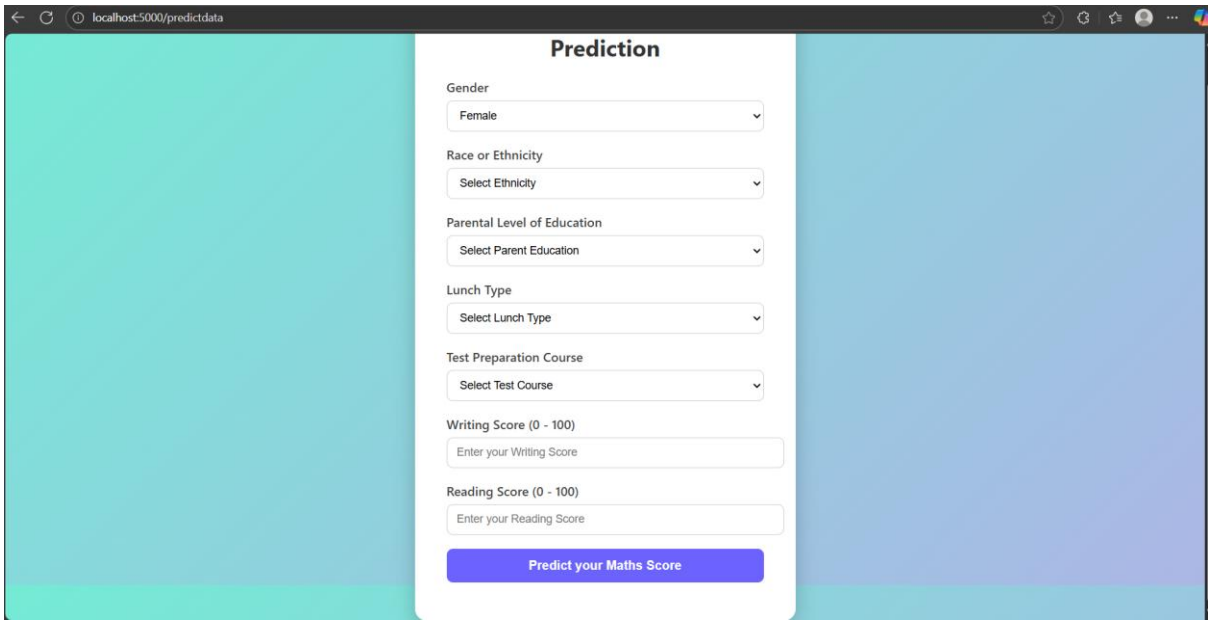
Summary

This project is an **end-to-end ML pipeline** with **data preprocessing, model training, and prediction**, deployed via a **Flask web app**. It's modular, with separate components for ingestion, transformation, training, and pipelines, plus logging, exception handling, and a simple HTML interface for user input and results.

Sample



```
from sklearn.preprocessing import StandardScaler
from src.pipeline.predict_pipeline import CustomData,PredictPipeline

application=flask(__name__)

app=application

## Route for a home page

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/predictdata',methods=['GET','POST'])
def predict_datapoint():
    if request.method=='GET':
        return render_template('home.html')
    else:
        data=CustomData(
            gender=request.form.get('gender'),
            race_ethnicity=request.form.get('ethnicity'),
            parental_level_of_education=request.form.get('parental_level_of_education'),
            lunch=request.form.get('lunch'),
            test_preparation_course=request.form.get('test_preparation_course'),
            reading_score=float(request.form.get('writing_score')),
            writing_score=float(request.form.get('reading_score'))

        )
        pred_df=data.get_data_as_data_frame()
        print(pred_df)
        print("Before Prediction")

        predict_pipeline=PredictPipeline()
        print("Mid Prediction")
        results=predict_pipeline.predict(pred_df)
        print("after Prediction")
        return render_template('home.html',results=results[0])
```



Deep Learning (DL)

Deep Learning uses multi-layered neural networks to learn complex patterns automatically.

Artificial Neural Networks (ANN)

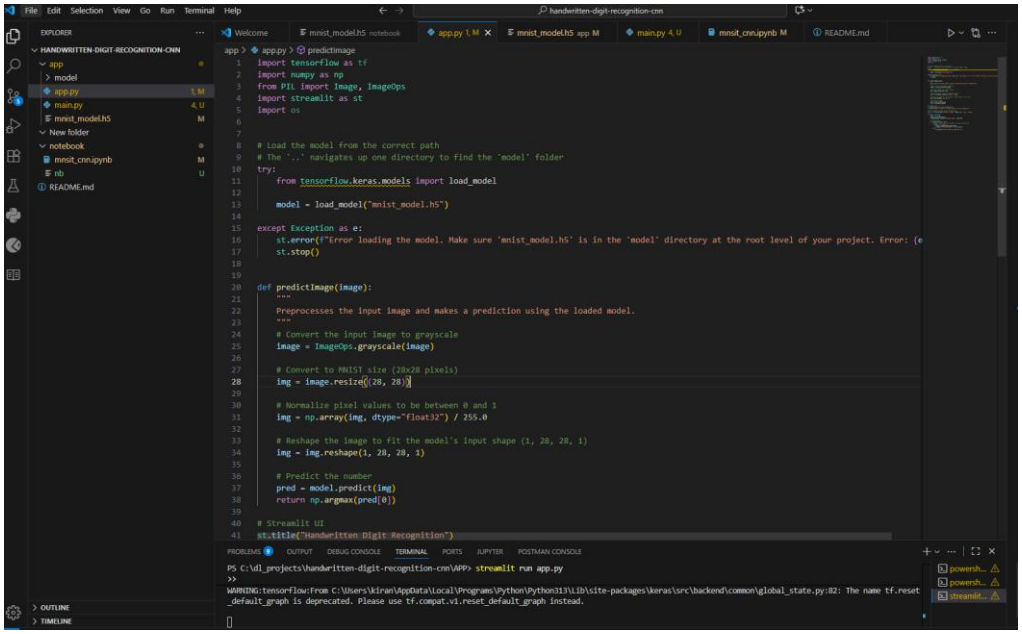
- Structure: Input → Hidden Layers → Output
- Forward pass: a(l) = f(W(l)a(l-1) + b(l))
- Backpropagation updates weights to minimize error
- Applications: Classification, regression
- **[Insert diagram/screenshot here]**

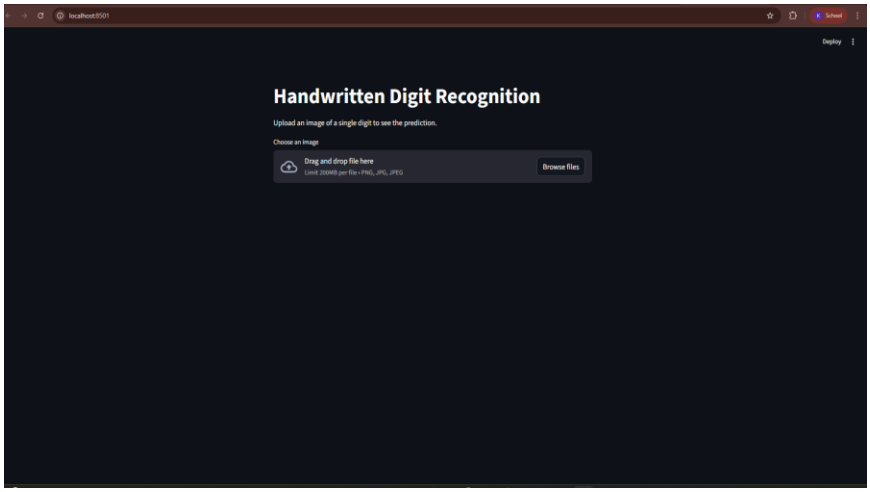
Projects

handwritten-digit-recognition-cnn

The project uses a **Convolutional Neural Network (CNN)** to recognize handwritten digits from the **MNIST dataset (0–9)**. The data is preprocessed (normalization, reshaping, and one-hot encoding), then passed through convolutional, pooling, and fully connected layers to learn visual patterns. The model is trained to classify digits with high accuracy (often above **98%**). Finally, it can predict digits from user-drawn images or test samples, showcasing how deep learning can be applied to **image classification tasks**.

Samples





Convolutional Neural Networks (CNN)

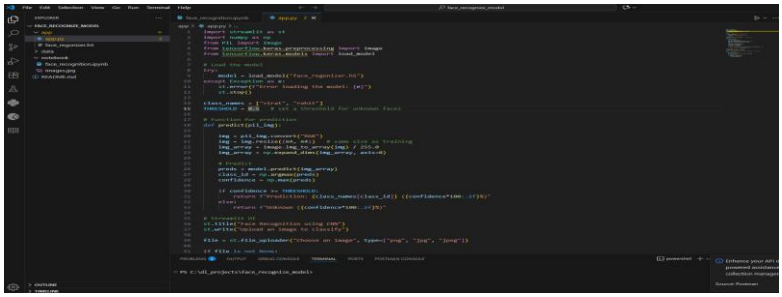
- For image data
- Convolution operation: $(f * g)(t) = \int f(\tau)\, g(t-\tau)\, d\tau$
- Pooling layers: Max pooling, average pooling
- Applications: Face recognition, object detection
- **[Insert diagram/screenshot here]**

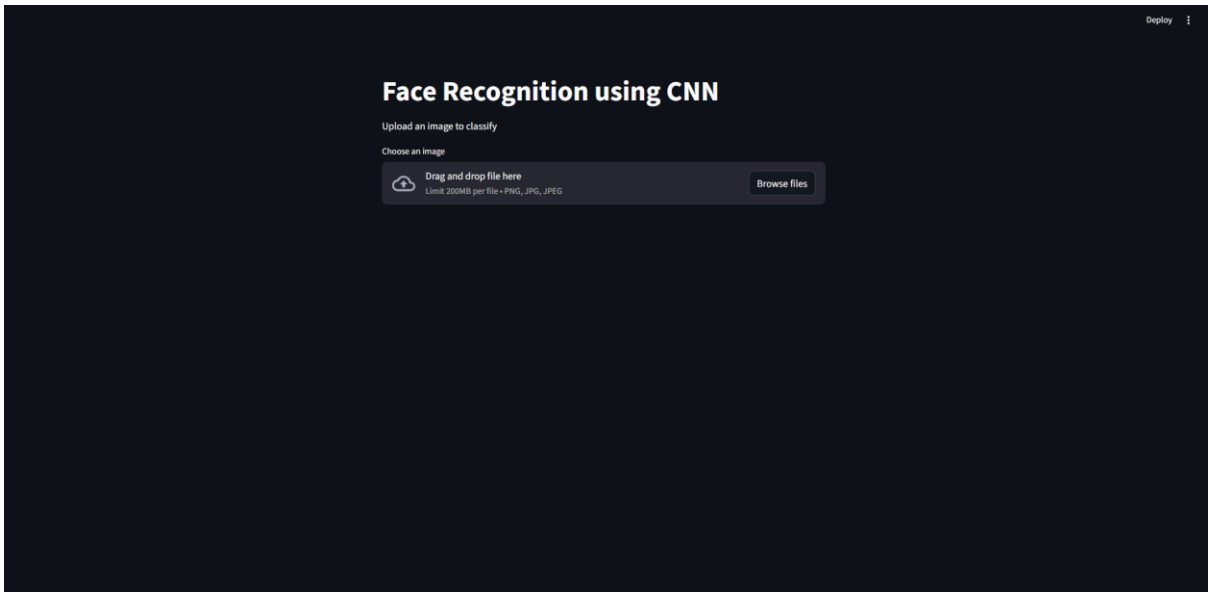
Project

CNN Face Classifie

The project uses a Convolutional Neural Network (CNN) to classify faces into categories such as identity, gender, or emotion. The dataset of face images is preprocessed (resizing, normalization, augmentation) and fed into CNN layers that automatically learn facial features. The trained model achieves high accuracy in distinguishing classes and can be deployed for real-time face classification.

samples





Face Recognition Automated Attendance Management System

Abstract:
This AI-powered Automated Attendance Management System was developed for educational institutions to streamline attendance recording and campus monitoring. The system uses facial recognition technology to automatically mark attendance, prevent proxy entries, and provide secure, tamper-proof attendance logs. It improves efficiency, accuracy, and transparency in academic management.

Objectives:

- **attendance marking using facial recognition.**
- **Reduce manual workload for teachers and administration.**
- **Ensure proxy-proof and accurate attendance records.**
- **Provide instant, secure access to attendance reports for faculty and administration.**
- **Integrate user-friendly dashboards for students, faculty, and HODs.**

System Design and Methodology:

- **Frontend: Developed with Next.js and Tailwind CSS for responsive and intuitive UI.**
- **Backend: Built using Go with PostgreSQL database to store attendance, user, and metadata securely.**
- **Face Recognition: Uses RetinaFace for detection and ArcFace for feature extraction and recognition.**
- **Workflow:**
    1. **Classroom video capture at start/end of classes.**
    2. **Face detection and alignment.**
    3. **Anti-spoofing checks to prevent fake attendance.**
    4. **Embedding generation and matching against the student database.**
    5. **Attendance marking with timestamp upon successful match.**
- **Data Access: Faculty can instantly fetch processed attendance data via dashboards.**

Key Features:

- **Fully automated attendance system.**

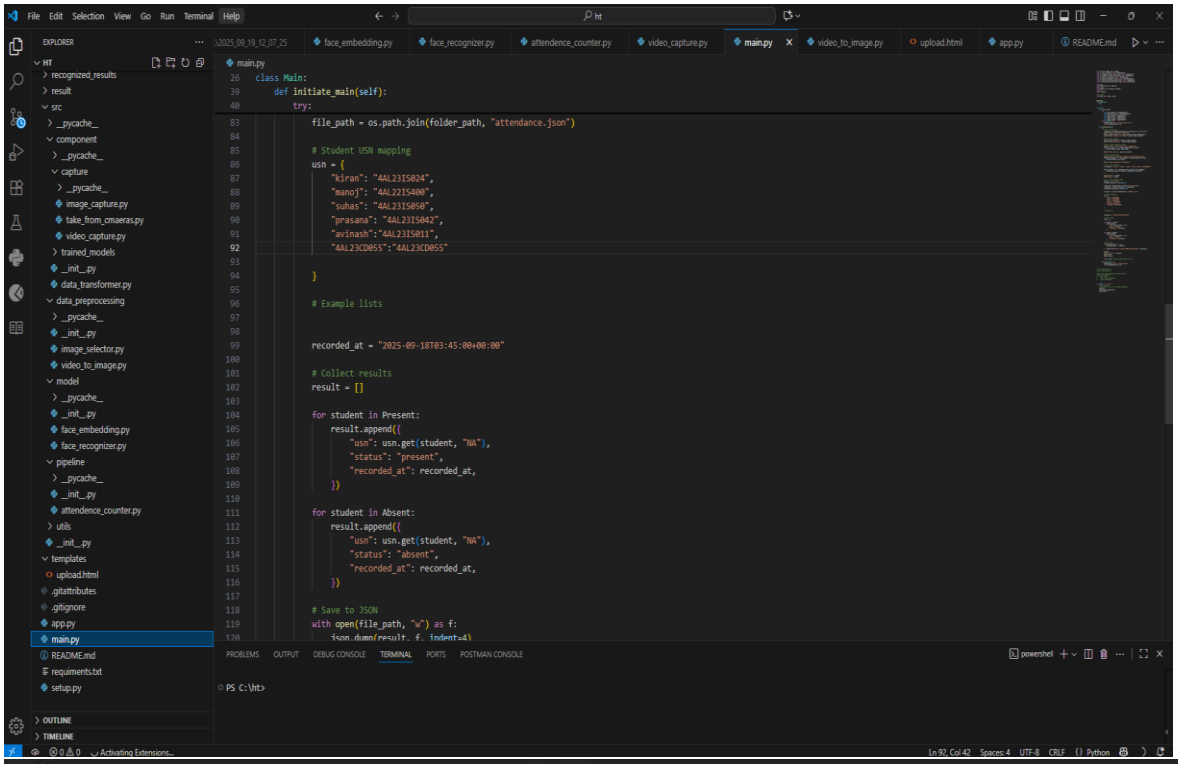
- **Proxy-proof with anti-spoofing measures.**
- **Real-time student attendance reporting.**
- **Student dashboards for attendance summary and detailed history.**
- **Faculty and HOD dashboards with empty-class alerts and inter-branch data sharing.**
- **Integrated payment management for college fees.**
- **Botpress chatbot integration for user support.**
- **Responsive design for web and mobile.**

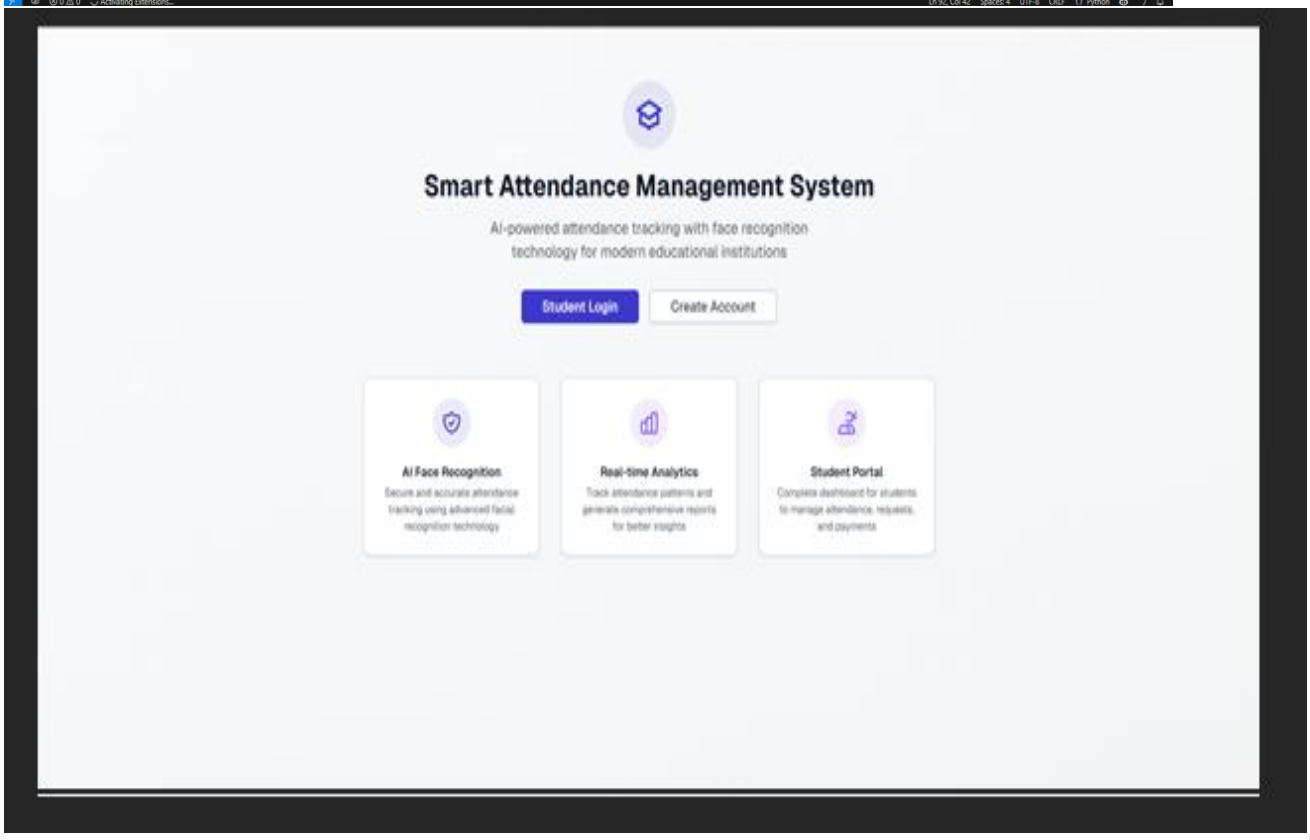
Learning Outcomes:

- **Hands-on experience with AI and ML models for facial recognition.**
- **Knowledge in full-stack development with Next.js, TypeScript, Go, and PostgreSQL.**
- **Understanding of real-time data processing, video feeds, and anti-spoofing algorithms.**
- **Improved skills in designing user-friendly dashboards and workflow automation.**

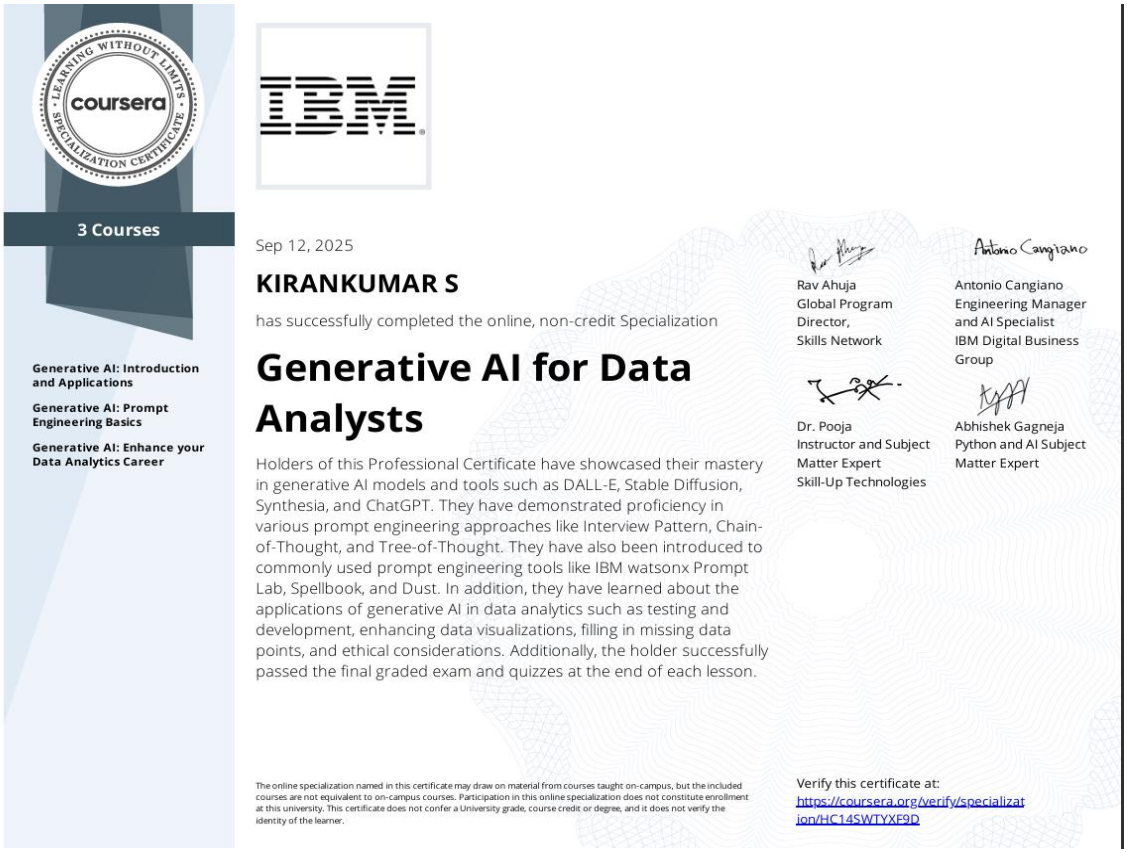
Future Scope:

- **Enhance AI models for faster recognition and higher accuracy.**
- **Expand certificate verification system to detect fake hospital certificates.**
- **Implement advanced HOD dashboards for multi-branch event management.**
- **Integrate more comprehensive student analytics and reporting tools.**
-


File Edit Selection View Go Run Terminal Help
EXPLORER
HT
recognized_results
result
src
pycache
component
capture
image_capture.py
take_from_cmaeras.py
video_capture.py
trained_models
__init__.py
data_transformer.py
data_preprocessing
image_selector.py
video_to_image.py
model
face_embedding.py
face_recognizer.py
pipeline
attendence_counter.py
utils
templates
upload.html
.gitattributes
.gitignore
app.py
main.py
README.md
requirements.txt
setup.py
OUTLINE
TIMELINE
face_embedding.py
face_recognizer.py
attendence_counter.py
video_capture.py
main.py
video_to_image.py
upload.html
app.py
README.md
class Main:
def initiate_main(self):
try:
file_path = os.path.join(folder_path, "attendance.json")
# Student USN mapping
usn = {
"kiran": "4AL23IS024",
"manoj": "4AL22IS400",
"suhas": "4AL23IS050",
"prasana": "4AL23IS042",
"avinash":"4AL23IS011",
"4AL23CD055":"4AL23CD055"
}
# Example lists
recorded_at = "2025-09-18T03:45:00+00:00"
# Collect results
result = []
for student in Present:
result.append({
"usn": usn.get(student, "NA"),
"status": "present",
"recorded_at": recorded_at,
})
for student in Absent:
result.append({
"usn": usn.get(student, "NA"),
"status": "absent",
"recorded_at": recorded_at,
})
Save to JSON
with open(file_path, "w") as f:
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE
powershell
PS C:\ht>
Activating Extensions...
Ln 92, Col 42 Spaces: 4 UTF-8 CRLF Python


Smart Attendance Management System
AI-powered attendance tracking with face recognition technology for modern educational institutions
Student Login
Create Account
AI Face Recognition
Secure and accurate attendance tracking using advanced facial recognition technology
Real-time Analytics
Track attendance patterns and generate comprehensive reports for better insights
Student Portal
Complete dashboard for students to manage attendance, requests, and payments

CERTIFICATS



3 Courses

Generative AI: Introduction and Applications

Generative AI: Prompt Engineering Basics

Generative AI: Enhance your Data Analytics Career

Sep 12, 2025

KIRANKUMAR S

has successfully completed the online, non-credit Specialization

Generative AI for Data Analysts

Holders of this Professional Certificate have showcased their mastery in generative AI models and tools such as DALL-E, Stable Diffusion, Synthesia, and ChatGPT. They have demonstrated proficiency in various prompt engineering approaches like Interview Pattern, Chain-of-Thought, and Tree-of-Thought. They have also been introduced to commonly used prompt engineering tools like IBM watsonx Prompt Lab, Spellbook, and Dust. In addition, they have learned about the applications of generative AI in data analytics such as testing and development, enhancing data visualizations, filling in missing data points, and ethical considerations. Additionally, the holder successfully passed the final graded exam and quizzes at the end of each lesson.

Rav Ahuja
Global Program Director,
Skills Network

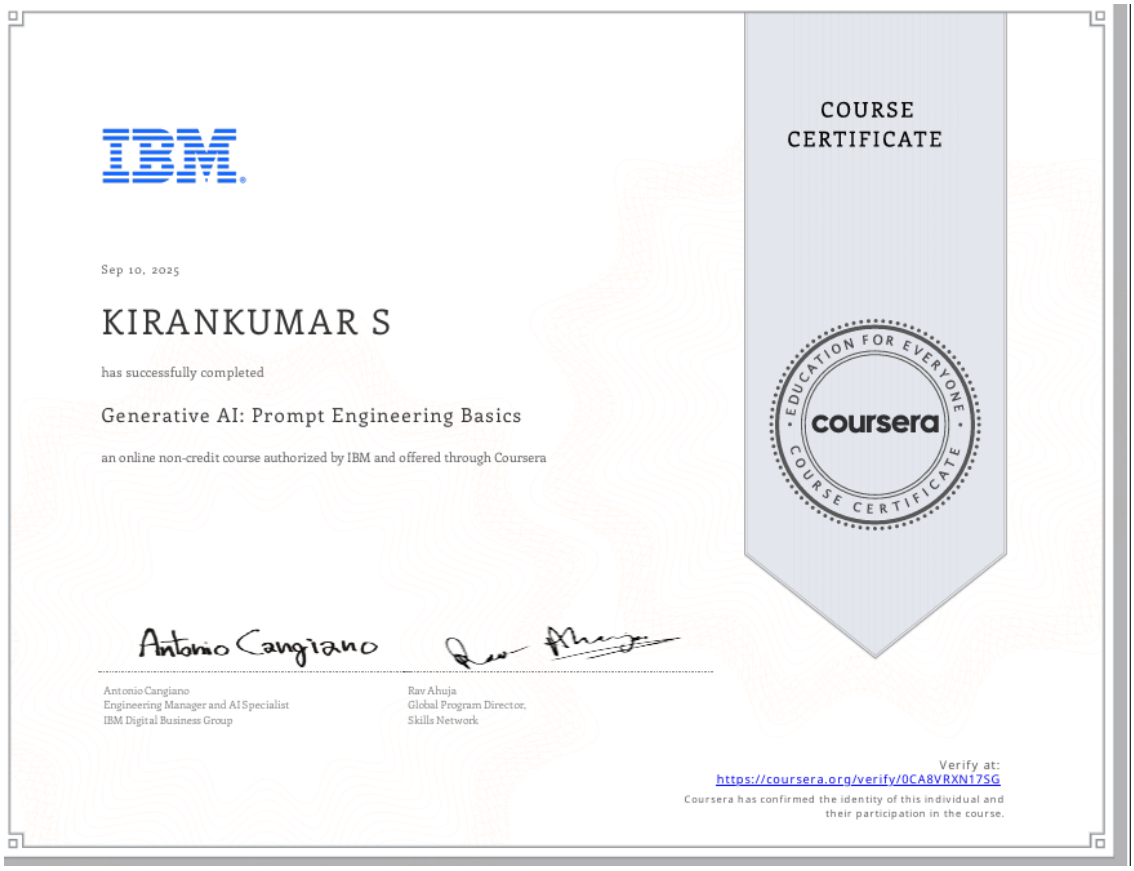
Antonio Cangiano
Engineering Manager and AI Specialist
IBM Digital Business Group

Dr. Pooja
Instructor and Subject Matter Expert
Skill-Up Technologies

Abhishek Gagneja
Python and AI Subject Matter Expert

The online specialization named in this certificate may draw on material from courses taught on-campus, but the included courses are not equivalent to on-campus courses. Participation in this online specialization does not constitute enrollment at this university. This certificate does not confer a University grade, course credit or degree, and it does not verify the identity of the learner.

Verify this certificate at:
https://coursera.org/verify/specialization/HC14SWTYXF9D



COURSE CERTIFICATE

Sep 10, 2025

KIRANKUMAR S

has successfully completed

Generative AI: Prompt Engineering Basics

an online non-credit course authorized by IBM and offered through Coursera

Antonio Cangiano
Engineering Manager and AI Specialist
IBM Digital Business Group

Rav Ahuja
Global Program Director,
Skills Network

Verify at:
https://coursera.org/verify/0CA8VRXN17SG
Coursera has confirmed the identity of this individual and their participation in the course.