SMARTINTEREST AI

**Objective**

SmartInterest AI is a student-centric analytics system that evaluates and predicts student interests based on their academic performance, involvement in projects, and attendance patterns. The system aims to provide personalized insights and actionable recommendations for academic and career growth. It is designed to help students identify their strengths and areas of interest, thereby fostering better engagement and performance in their studies.

### **Core Features**

#### **1. Data Collection and Input System**

* **Inputs Required**:
  + **Attendance Data**:
    - Percentage of attendance in each subject/course.
    - Attendance will have a low weightage in the prediction model.
  + **Marks Data**:
    - Subject-wise marks obtained in exams, assignments, or practicals.
    - Marks will have medium weightage in the model.
  + **Project Data**:
    - Details of up to 4 projects:
      * **Domain** (e.g., Machine Learning, Web Development, Data Science, etc.).
      * **Feature List** (key functionalities of the project).
      * **Technologies Used** (e.g., Python, React.js, TensorFlow).
    - Projects will have the highest weightage in the prediction model.
* **Input System**:
  + A web-based interface with personalized user authentication for students to input their data securely.
  + Dynamic forms to ensure ease of data entry, with options for updating project details and performance over time.

#### **2. Data Analysis and Processing**

* **Attendance Analysis**:
  + Identifies students with consistently low attendance and correlates it with their academic performance and project involvement.
* **Marks Analysis**:
  + Tracks performance trends across subjects.
  + Flags subjects where the student is underperforming or excelling.
* **Project Analysis**:
  + Evaluates the domains of projects to identify patterns in student interest (e.g., a student doing multiple AI projects may have a strong interest in Artificial Intelligence).
  + Analyzes the features and technologies used to assess the depth and scope of the student’s work.

#### **3. Interest Prediction Model**

* **Algorithm**:
  + A custom Machine Learning model trained using supervised learning techniques to analyze:
    - **Attendance** (Low weightage).
    - **Marks** (Medium weightage).
    - **Projects** (High weightage).
  + Domains of projects will be treated as key predictors for interest areas.
* **Outcome**:
  + The system predicts the top 1-3 domains where the student is likely most interested.
  + Provides a confidence score (e.g., 85% interest in AI, 70% in Web Development).

#### **4. Personalized Roadmaps**

* **Insights Provided**:
  + Identifies strong subjects where the student can capitalize on their skills.
  + Recommends a personalized improvement plan for weak subjects.
  + Suggests potential project ideas and skills to learn based on predicted interests.
  + Suggests online courses, certifications, or competitions to further explore the student’s interests.

#### **5. Visualization and Reporting**

* **Data Visualization**:
  + Intuitive dashboards for students to view:
    - Attendance trends.
    - Marks distribution and performance trends.
    - Project summaries and impact on predictions.
  + Graphical representations (bar charts, pie charts, heatmaps) for better understanding.
* **Reports**:
  + Automatically generated personalized reports that summarize the student’s performance, interests, and recommendations.
  + Downloadable in PDF format for easy sharing with teachers, parents, or career counselors.

#### **6. Web Interface**

* **Authentication**:
  + Secure login system with individual accounts for each student.
  + Role-based access (e.g., admin or mentor accounts for batch-level insights).
* **User Experience**:
  + Minimalistic and user-friendly design.
  + Mobile-responsive interface for accessibility on any device.

**Steps**

1. **Data Generation:**

Explanation:

* Student ID: Unique identifier for each student.
* Subject 1, Subject 2, Subject 3: Example subjects (e.g., Mathematics, English, Science).
* Attendance (Subject 1), Attendance (Subject 2), Attendance (Subject 3): Percentage of attendance for each subject.
* Project 1, Project 2, Project 3, Project 4: Names or brief descriptions of the student's projects (leave empty if no project).

**import pandas as pd**

**import numpy as np**

**# Define the number of rows**

**num\_rows = 1000**

**# Define subjects and project domains**

**subjects = ['Operating System', 'DSA', 'Frontend', 'Backend', 'Machine Learning', 'Data Analytics']**

**project\_domains = ['Web Development', 'Data Science', 'Machine Learning', 'AI', 'Robotics', 'Game Development', 'Cybersecurity']**

**# Generate random data**

**data = {**

**'Student ID': np.arange(1, num\_rows + 1),**

**'Operating System': np.random.randint(35, 101, num\_rows),**

**'DSA': np.random.randint(35, 101, num\_rows),**

**'Frontend': np.random.randint(35, 101, num\_rows),**

**'Backend': np.random.randint(35, 101, num\_rows),**

**'Machine Learning': np.random.randint(35, 101, num\_rows),**

**'Data Analytics': np.random.randint(35, 101, num\_rows),**

**'Attendance (Operating System)': np.random.randint(50, 101, num\_rows),**

**'Attendance (DSA)': np.random.randint(50, 101, num\_rows),**

**'Attendance (Frontend)': np.random.randint(50, 101, num\_rows),**

**'Attendance (Backend)': np.random.randint(50, 101, num\_rows),**

**'Attendance (Machine Learning)': np.random.randint(50, 101, num\_rows),**

**'Attendance (Data Analytics)': np.random.randint(50, 101, num\_rows),**

**'Project 1': np.random.choice(project\_domains, num\_rows, replace=True),**

**'Project 2': np.random.choice(project\_domains, num\_rows, replace=True),**

**'Project 3': np.random.choice(project\_domains, num\_rows, replace=True),**

**'Project 4': np.random.choice(project\_domains, num\_rows, replace=True)**

**}**

**# Create a DataFrame**

**df = pd.DataFrame(data)**

**# Save to CSV**

**df.to\_csv('student\_data.csv', index=False)**

**print(f"Dataset generated and saved as 'student\_data.csv'")**

1. **Load and Inspect the Dataset**

import pandas as pd

# Load the dataset

df = pd.read\_csv('student\_data.csv')

# Inspect the dataset

print(df.head()) # View the first 5 rows

print(df.info()) # Summary of columns and data types

print(df.describe()) # Statistics for numerical columns

# Check for missing values

print(df.isnull().sum())

1. **Preprocess the Data**

from sklearn.preprocessing import StandardScaler, OneHotEncoder

# Select numerical features

numerical\_features = df[['Operating System', 'DSA', 'Frontend', 'Backend','Machine Learning', 'Data Analytics','Attendance (Operating System)','Attendance (DSA)','Attendance (Frontend)', 'Attendance (Backend)','Attendance (Machine Learning)', 'Attendance (Data Analytics)']]

# Scale numerical features

scaler = StandardScaler()

scaled\_features = scaler.fit\_transform(numerical\_features)# Encode project domains

project\_columns = ['Project 1', 'Project 2', 'Project 3', 'Project 4']

projects\_encoded = pd.get\_dummies(df[project\_columns].stack()).groupby(level=0).sum()

# Combine scaled numerical features and encoded project data

final\_data = pd.concat([pd.DataFrame(scaled\_features, columns=numerical\_features.columns), projects\_encoded], axis=1)