# SmartInterest AI

**Project Based Learning -IV (AIP105)**

***Submitted by:***

**NISHANT GARG(2310993891)**
**MEHARJOT SINGH(2310993876)**
**JASMEET SINGH(2310993842)**

**BE-CSE (Artificial Intelligence)**

***Guided by:***

**Dr. Rajan**

**CHITKARA UNIVERSITY INSITUTE OF ENGINEERING & TECHNOLOGY**
**CHITKARA UNIVERSITY, RAJPURA**
**05/25**

# ACKNOWLEDGEMENTS

With immense please We, **Jasmeet Singh, Meharjot Singh and Nishant Garg** presenting "**SmartInterest AI**" project report as part of the curriculum of 'BE-CSE (AI)'.

We would like to express my sincere thanks to Dr. Rajan, for his valuable guidance and support in completing my project.

We would also like to express my gratitude towards our dean Dr. SUSHIL NARANG, for giving us this great opportunity to do a project on "**SmartInterest AI**". Without their support and suggestions, this project would not have been completed.

SIGNATURE:

NAME: Jasmeet Singh

ROLL NUMBER:2310993842

SIGNATURE:

NAME: Meharjot Singh

ROLL NUMBER:2310993876

SIGNATURE:

NAME: Nishant Garg

ROLL NUMBER:2310993891

# ABSTRACT

In today's rapidly evolving educational landscape, understanding and nurturing individual student interests is crucial for effective learning and career planning. However, manual evaluation of student potential based on academic records and project involvement is subjective and time-consuming. **SmartInterest AI** addresses this challenge by leveraging machine learning to predict a student's domain of interest using their academic performance and project history.

This project uses a structured dataset of **300,000 student records**, each containing subject-wise marks and four associated project domains. A domain-mapping logic was developed, linking academic subjects (such as Operating System, DSA, Frontend, Backend, Machine Learning, and Data Analytics) to various professional domains (like AI, Web Development, Cybersecurity, and Game Development). For each student, the top three subjects by marks were mapped to their respective domains and cross-compared with their project domains to determine the most aligned field of interest.

Preprocessing steps included normalization of marks, encoding of categorical domains, and removal of students with unknown or non-aligned domain data. After cleaning, multiple machine learning models were trained, with **XGBoost achieving the highest accuracy of 96%**, outperforming alternatives like Random Forest, Logistic Regression, and KNN.

A full-stack web application is in development to make this system accessible. The planned platform will allow students to input their academic data and project history to receive domain predictions, analyze strengths and weaknesses, and obtain personalized learning paths. Technologies like **React.js, Flask, and Firebase** are used to ensure a responsive, scalable deployment.

SmartInterest AI not only aids students in identifying suitable academic and career directions but also empowers educators with insights for more personalized mentorship. Future enhancements include confidence scoring, domain-specific recommendations, and teacher dashboards for cohort-level analytics, aiming to bridge the gap between education and career alignment using intelligent automation.

# Table of Index:

# 1. Introduction

In the current era of personalized learning and data-driven decision-making, it has become increasingly important to understand individual student interests and guide them toward suitable academic and career paths. However, many educational institutions still rely on generic counseling approaches, which often overlook the unique capabilities and interests of each student. This gap between student potential and career direction can lead to disengagement, poor academic outcomes, and misaligned career choices.

**SmartInterest AI** is a machine learning-based system developed to bridge this gap. It analyzes a student's academic performance and prior project experience to predict their most suitable domain of interest—such as Web Development, Artificial Intelligence, Cybersecurity, Game Development, and more. This prediction allows students to receive tailored guidance aligned with their strengths and previous engagements.

The system is built on a large-scale dataset consisting of **300,000 student records**, with each record containing subject marks in six core technical areas and four project domains. By applying a domain-mapping logic to identify overlaps between top-performing subjects and the domains of completed projects, the system determines the student's most likely field of interest. This inferred interest domain is then used to train predictive models.

To ensure accuracy, the dataset underwent extensive preprocessing—scaling, encoding, and cleaning—before being used for training machine learning models. Among several models tested, **XGBoost** demonstrated the highest prediction accuracy of **96%**, making it the model of choice for deployment.

SmartInterest AI goes beyond prediction. A user-friendly web interface is planned, allowing students to input their details and instantly receive feedback on their predicted interest domain, along with subject-wise strengths and personalized learning roadmaps. This system not only supports students in making informed academic choices but also assists educators and institutions in delivering more targeted mentorship and plan

# 2. Problem Formulation:

In traditional academic settings, identifying a student's area of interest often depends on manual assessments, general counseling sessions, or student self-awareness. These methods are typically inconsistent, time-consuming, and subjective. Students may select academic or career paths based on external influences rather than a data-backed understanding of their strengths and inclinations. As a result, many students struggle to stay engaged or excel in their chosen fields.

The core problem lies in the absence of a scalable, intelligent system that can analyze a student's academic and project history to automatically infer their most likely domain of interest. While educational data such as subject marks and project experiences are available, this data is rarely leveraged effectively for personalized guidance.

**SmartInterest AI** addresses this problem by formulating it as a supervised classification task. The goal is to develop a machine learning model that takes as input:

- Marks in six core subjects (Operating System, DSA, Frontend, Backend, Machine Learning, and Data Analytics)
- Domains of four past projects

and outputs the predicted **domain of interest** for each student (e.g., Web Development, AI, Cybersecurity, Game Development, etc.).

To make this possible, a domain-mapping logic is applied to the top three subjects with the highest marks for each student. These subject domains are then compared with the project domains. The overlapping domain is selected as the label (target) for model training. If no overlap is found, the record is marked as *Unknown* and excluded from training to ensure label integrity.

This formulation allows the use of machine learning algorithms like **XGBoost**, which can learn complex patterns from the input data and make highly accurate predictions. The problem formulation also lays the groundwork for building a scalable, web-based recommendation tool that can offer real-time domain predictions for students.

# 3. Proposed Solution / Methodology:

To address the challenge of predicting a student's domain of interest based on their academic and project performance, **SmartInterest AI** employs a structured and data-driven methodology. The overall solution involves transforming raw educational data into meaningful insights using machine learning techniques. The methodology is divided into the following key stages:

### 1. Subject-to-Domain Mapping

Each of the six academic subjects is mapped to related technical domains based on relevance and industry associations:

'Operating System': ['Cybersecurity', 'Robotics', 'Game Development'],

'DSA': ['AI', 'Machine Learning', 'Cybersecurity', 'Game Development'],

'Frontend': ['Web Development', 'Game Development'],

'Backend': ['Web Development', 'Game Development'],

'Machine Learning': ['AI', 'Data Science', 'Machine Learning'],

'Data Analytics': ['Data Science', 'Machine Learning', 'AI']

### 2. Interest Label Generation

- For each student, the top **three subjects** with the highest marks are selected.
- The domains associated with these subjects are extracted.
- The system compares these subject-derived domains with the **domains of four student projects**.
- The **most frequently overlapping domain** between top subjects and project domains is selected as the **predicted interest label**.
- If no overlap is found, the label is set to **Unknown** and that entry is excluded from training.

### 3. Data Preprocessing

- Project domains are numerically encoded for model input.
- Academic marks are normalized to a 0–1 scale.
- The project fields are flattened into four separate domain columns.
- Rows labeled as "Unknown" are removed to maintain label quality.
- Cleaned and labeled data is saved in cleaned_student_data.csv.

## 4. Model Training

Multiple models were trained and evaluated:

- XGBoost (Best Accuracy: **96%**)
- Random Forest
- Logistic Regression
- K-Nearest Neighbors (KNN)

The **XGBoost Classifier** outperformed others and was saved as the final model for deployment.

# 4. System Flow Process:

The SmartInterest AI system follows a structured flow of data collection, preprocessing, domain mapping, model training, and final deployment. Below is the step-by-step breakdown of the system flow:

### Step 1: Input Data Collection

- **Student ID**
- **Subject Marks** in:
  - Operating System
  - DSA
  - Frontend
  - Backend
  - Machine Learning
  - Data Analytics
- **4 Project Domains** (e.g., AI, Web Development)

### Step 2: Subject-to-Domain Mapping

- Identify the **top 3 subjects** with the highest marks for each student.
- Map these subjects to corresponding **interest domains** using the predefined subject-to-domain dictionary.

### Step 3: Label Generation

- Extract the domains associated with the student’s four projects.
- Compare them with the domains derived from top subjects.
- Assign the **most frequent overlapping domain** as the student's **predicted interest label**.
- If no overlap is found, mark as **Unknown** and remove from training dataset.

### Step 4: Data Preprocessing

- Normalize subject marks between 0 and 1.
- Encode project domain values numerically.

- Flatten project columns for model compatibility.
- Save clean, labeled data for model training.

### Step 5: Model Training and Evaluation

- Train multiple models (XGBoost, Random Forest, etc.).
- Evaluate accuracy and select the best model (XGBoost with 93.25% accuracy).
- Save model for deployment.

### Step 6: Deployment via Web Interface

- Students input their marks and project domains.
- The model predicts their domain of interest.
- Output includes:
  - Predicted Domain
  - Strong/Weak Subjects
  - Suggested Learning Path

# 5. Software and Hardware Requirement

### 1. Programming Languages

- **Python 3.10.0** – Core development language for backend APIs and machine learning model implementation.
- **JavaScript (ES6+)** – For frontend development using React.js.
- **TypeScript** – (Optional) for type-safe frontend code.

### 2. Frameworks & Libraries

**Machine Learning & Data Handling**

- **Scikit-learn** – For training and predicting user interests (e.g., XGBoost-based model).
- **XGBoost** – Classifier used for interest prediction.
- **NumPy, Pandas** – For numerical computation and data manipulation.
- **LabelEncoder** – For categorical label transformations.

### 3. Backend Development

- **Flask** – Lightweight Python framework for building REST APIs.
- **Flask-CORS** – For handling Cross-Origin Resource Sharing between frontend and backend.
- **PyMongo / MongoEngine** – To interact with MongoDB database.
- **JSON** – Data exchange format between frontend and backend.

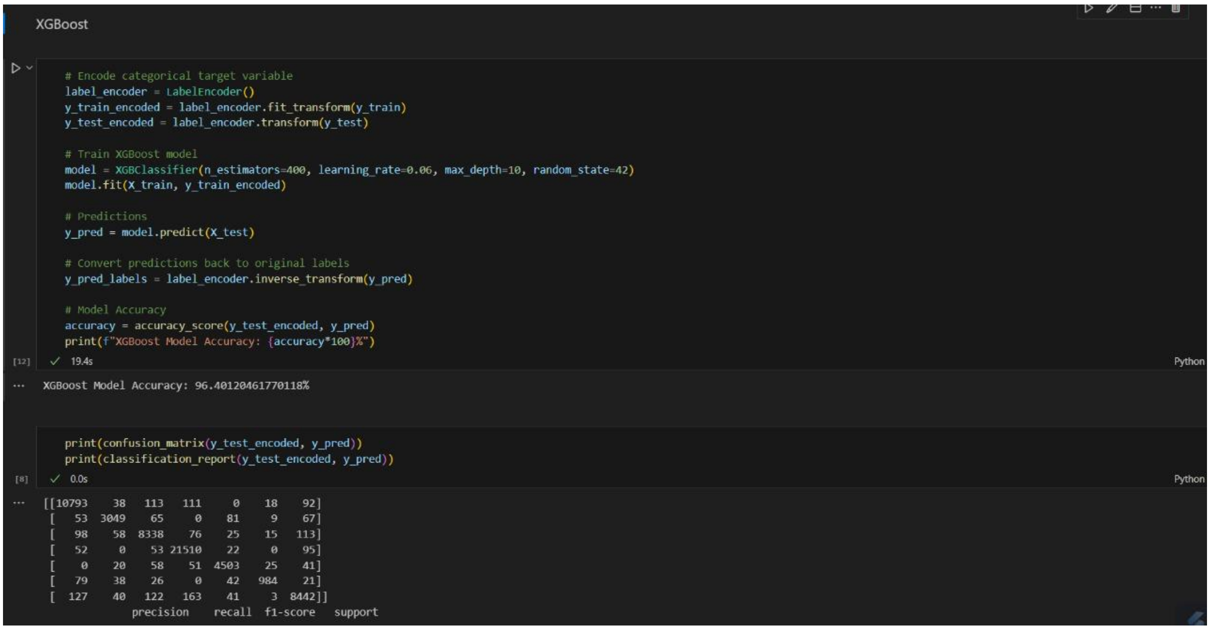
### 4. Frontend Development

- **React.js** – Frontend UI framework for building a responsive dashboard.
- **React Router DOM** – For route management in SPA.
- **TailwindCSS / CSS3** – For styling the application.
- **Axios / Fetch API** – For API communication with Flask backend.

### 5. Database

- **MongoDB Atlas (Cloud)** – To store user data, projects, roadmaps, and quiz questions.
- **Alternatives**: SQLite (for local testing), Firebase Realtime DB (if required for authentication extension).

# 6. Code



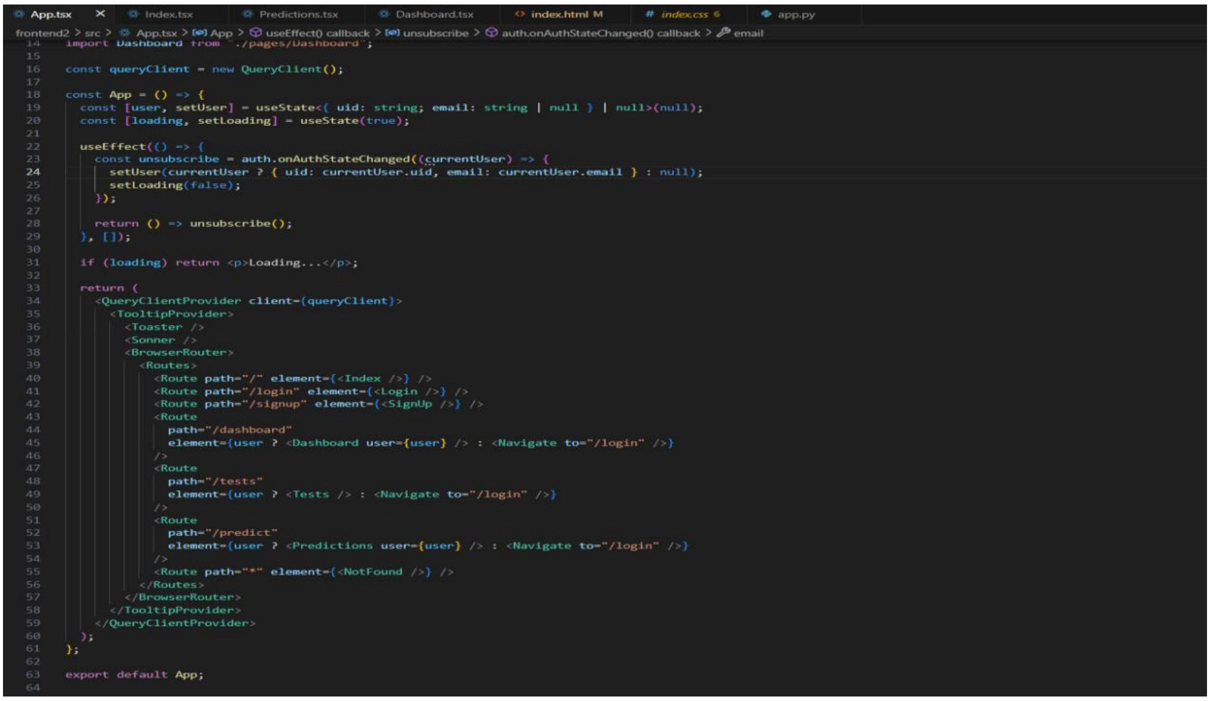
```python
# Encode categorical target variable
label_encoder = LabelEncoder()
y_train_encoded = label_encoder.fit_transform(y_train)
y_test_encoded = label_encoder.transform(y_test)

# Train XGBoost model
model = XGBClassifier(n_estimators=400, learning_rate=0.06, max_depth=10, random_state=42)
model.fit(X_train, y_train_encoded)

# Predictions
y_pred = model.predict(X_test)

# Convert predictions back to original labels
y_pred_labels = label_encoder.inverse_transform(y_pred)

# Model Accuracy
accuracy = accuracy_score(y_test_encoded, y_pred)
print(f"XGBoost Model Accuracy: {accuracy*100}%")
```

```
XGBoost Model Accuracy: 96.40120461770118%
```

```python
print(confusion_matrix(y_test_encoded, y_pred))
print(classification_report(y_test_encoded, y_pred))
```

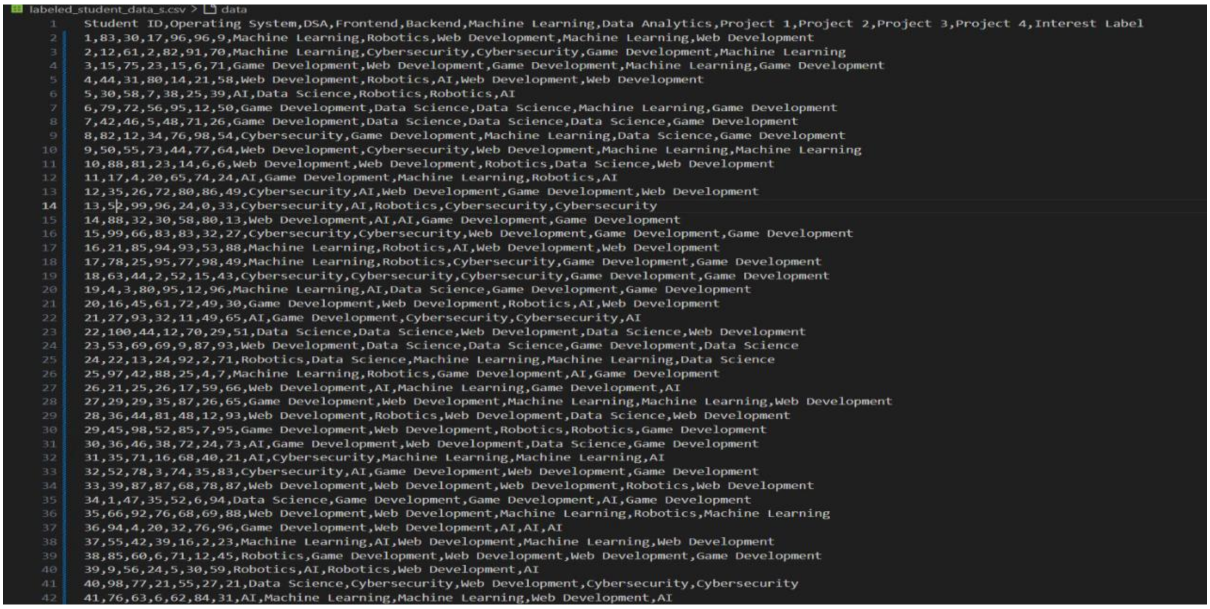
```
[[10793    38   113   111     0    18    92]
 [   53  3049    65     0    81     9    67]
 [   98    58  8338    76    25    15   113]
 [   52     0    53 21510    22     0    95]
 [    0    20    58    51  4503    25    41]
 [   79    38    26     0    42   984    21]
 [  127    40   122   163    41     3  8442]]
              precision    recall  f1-score   support
```

**Fig 1. XGBoost Model Training and Evaluation Results**


App.tsx
Index.tsx
Predictions.tsx
Dashboard.tsx
index.html M
index.css 6
app.py
frontend2 > src > App.tsx > App > useEffect() callback > unsubscribe > auth.onAuthStateChanged() callback > email

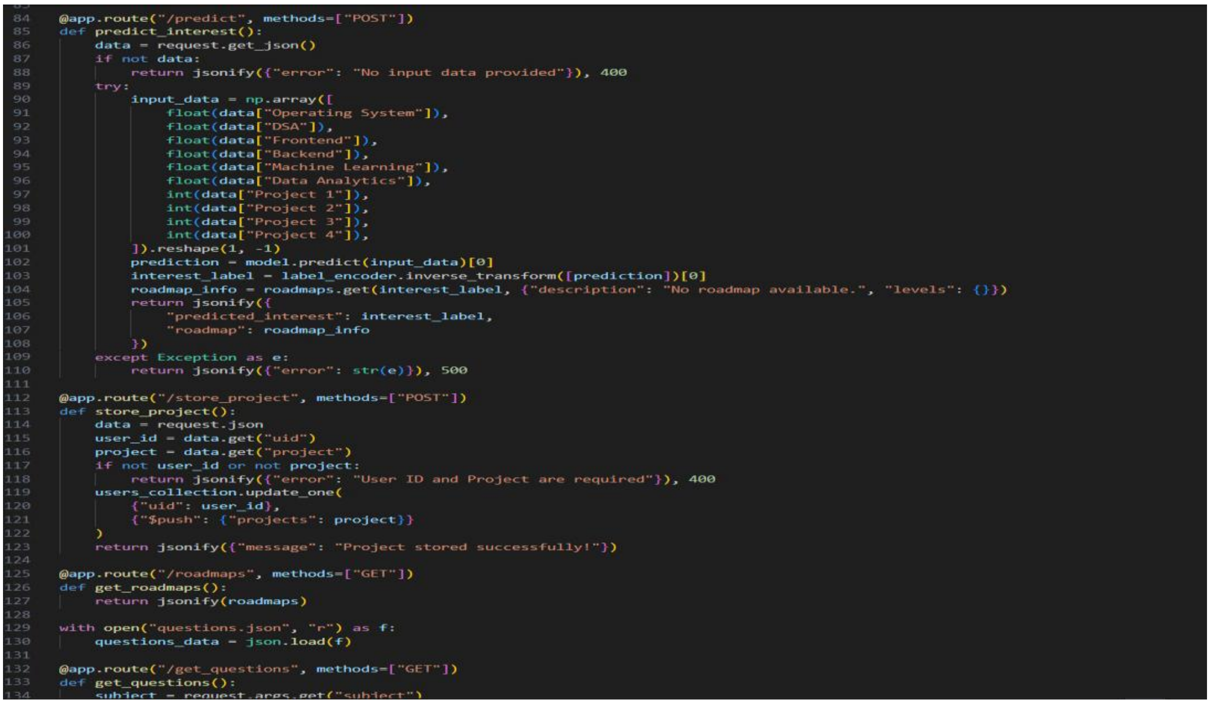
```tsx
import Dashboard from "./pages/Dashboard";

const queryClient = new QueryClient();

const App = () => {
  const [user, setUser] = useState<{ uid: string; email: string | null } | null>(null);
  const [loading, setLoading] = useState(true);

  useEffect(() => {
    const unsubscribe = auth.onAuthStateChanged((currentUser) => {
      setUser(currentUser ? { uid: currentUser.uid, email: currentUser.email } : null);
      setLoading(false);
    });

    return () => unsubscribe();
  }, []);

  if (loading) return <p>Loading...</p>;

  return (
    <QueryClientProvider client={queryClient}>
      <TooltipProvider>
        <Toaster />
        <Sonner />
        <BrowserRouter>
          <Routes>
            <Route path="/" element={<Index />} />
            <Route path="/login" element={<Login />} />
            <Route path="/signup" element={<SignUp />} />
            <Route
              path="/dashboard"
              element={user ? <Dashboard user={user} /> : <Navigate to="/login" />}
            />
            <Route
              path="/tests"
              element={user ? <Tests /> : <Navigate to="/login" />}
            />
            <Route
              path="/predict"
              element={user ? <Predictions user={user} /> : <Navigate to="/login" />}
            />
            <Route path="*" element={<NotFound />} />
          </Routes>
        </BrowserRouter>
      </TooltipProvider>
    </QueryClientProvider>
  );
};

export default App;
```

**Fig 2. App Component Routing and Authentication Handling in React**



```
Student ID,Operating System,DSA,Frontend,Backend,Machine Learning,Data Analytics,Project 1,Project 2,Project 3,Project 4,Interest Label
1,83,30,17,96,96,9,Machine Learning,Robotics,Web Development,Machine Learning,Web Development
2,12,61,2,82,91,70,Machine Learning,Cybersecurity,Cybersecurity,Game Development,Machine Learning
3,15,75,23,15,6,71,Game Development,Web Development,Game Development,Machine Learning,Game Development
4,44,31,80,14,21,58,Web Development,Robotics,AI,Web Development,Web Development
5,30,58,7,38,25,39,AI,Data Science,Robotics,Robotics,AI
6,79,72,56,95,12,50,Game Development,Data Science,Data Science,Machine Learning,Game Development
7,42,46,5,48,71,26,Game Development,Data Science,Data Science,Data Science,Game Development
8,82,12,34,76,98,54,Cybersecurity,Game Development,Machine Learning,Data Science,Game Development
9,50,55,73,44,77,64,Web Development,Cybersecurity,Web Development,Machine Learning,Machine Learning
10,88,81,23,14,6,6,Web Development,Web Development,Robotics,Data Science,Web Development
11,17,4,20,65,74,24,AI,Game Development,Machine Learning,Robotics,AI
12,35,26,72,80,86,49,Cybersecurity,AI,Web Development,Game Development,Web Development
13,52,99,96,24,0,33,Cybersecurity,AI,Robotics,Cybersecurity,Cybersecurity
14,88,32,30,58,80,13,Web Development,AI,AI,Game Development,Game Development
15,99,66,83,83,32,27,Cybersecurity,Cybersecurity,Web Development,Game Development,Game Development
16,21,85,94,93,53,88,Machine Learning,Robotics,AI,Web Development,Web Development
17,78,25,95,77,98,49,Machine Learning,Robotics,Cybersecurity,Game Development,Game Development
18,63,44,2,52,15,43,Cybersecurity,Cybersecurity,Cybersecurity,Game Development,Game Development
19,4,3,80,95,12,96,Machine Learning,AI,Data Science,Game Development,Game Development
20,16,45,61,72,49,30,Game Development,Web Development,Robotics,AI,Web Development
21,27,93,32,11,49,65,AI,Game Development,Cybersecurity,Cybersecurity,AI
22,100,44,12,70,29,51,Data Science,Data Science,Web Development,Data Science,Web Development
23,53,69,69,9,87,93,Web Development,Data Science,Data Science,Game Development,Data Science
24,22,13,24,92,2,71,Robotics,Data Science,Machine Learning,Machine Learning,Data Science
25,97,42,88,25,4,7,Machine Learning,Robotics,Game Development,AI,Game Development
26,21,25,26,17,59,66,Web Development,AI,Machine Learning,Game Development,AI
27,29,29,35,87,26,65,Game Development,Web Development,Machine Learning,Machine Learning,Web Development
28,36,44,81,48,12,93,Web Development,Robotics,Web Development,Data Science,Web Development
29,45,98,52,85,7,95,Game Development,Web Development,Robotics,Robotics,Game Development
30,36,46,38,72,24,73,AI,Game Development,Web Development,Data Science,Game Development
31,35,71,16,68,40,21,AI,Cybersecurity,Machine Learning,Machine Learning,AI
32,52,78,3,74,35,83,Cybersecurity,AI,Game Development,Web Development,Game Development
33,39,87,87,68,78,87,Web Development,Web Development,Web Development,Robotics,Web Development
34,1,47,35,52,6,94,Data Science,Game Development,Game Development,AI,Game Development
35,66,92,76,68,69,88,Web Development,Web Development,Machine Learning,Robotics,Machine Learning
36,94,4,20,32,76,96,Game Development,Web Development,AI,AI,AI
37,55,42,39,16,2,23,Machine Learning,AI,Web Development,Machine Learning,Web Development
38,85,60,6,71,12,45,Robotics,Game Development,Web Development,Web Development,Game Development
39,9,56,24,5,30,59,Robotics,AI,Robotics,Web Development,AI
40,98,77,21,55,27,21,Data Science,Cybersecurity,Web Development,Cybersecurity,Cybersecurity
41,76,63,6,62,84,31,AI,Machine Learning,Machine Learning,Web Development,AI
```

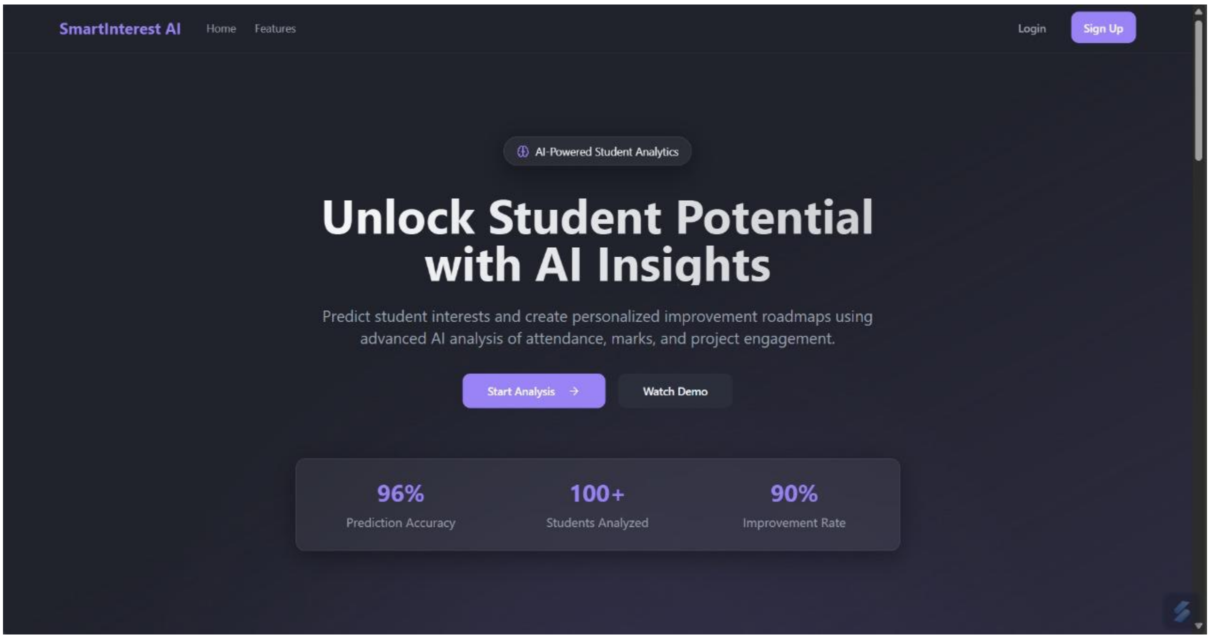
**Fig 3. Labeled Student Skill and Interest Dataset Overview**



```
@app.route("/predict", methods=["POST"])
def predict_interest():
    data = request.get_json()
    if not data:
        return jsonify({"error": "No input data provided"}), 400
    try:
        input_data = np.array([
            float(data["Operating System"]),
            float(data["DSA"]),
            float(data["Frontend"]),
            float(data["Backend"]),
            float(data["Machine Learning"]),
            float(data["Data Analytics"]),
            int(data["Project 1"]),
            int(data["Project 2"]),
            int(data["Project 3"]),
            int(data["Project 4"]),
        ]).reshape(1, -1)
        prediction = model.predict(input_data)[0]
        interest_label = label_encoder.inverse_transform([prediction])[0]
        roadmap_info = roadmaps.get(interest_label, {"description": "No roadmap available.", "levels": {}})
        return jsonify({
            "predicted_interest": interest_label,
            "roadmap": roadmap_info
        })
    except Exception as e:
        return jsonify({"error": str(e)}), 500

@app.route("/store_project", methods=["POST"])
def store_project():
    data = request.json
    user_id = data.get("uid")
    project = data.get("project")
    if not user_id or not project:
        return jsonify({"error": "User ID and Project are required"}), 400
    users_collection.update_one(
        {"uid": user_id},
        {"$push": {"projects": project}}
    )
    return jsonify({"message": "Project stored successfully!"})

@app.route("/roadmaps", methods=["GET"])
def get_roadmaps():
    return jsonify(roadmaps)

with open("questions.json", "r") as f:
    questions_data = json.load(f)

@app.route("/get_questions", methods=["GET"])
def get_questions():
    subject = request.args.get("subject")
```

**Fig 4.** **Flask Backend: Interest Prediction and Roadmap Management API**

# 7. Results



Unlock Student Potential with AI Insights

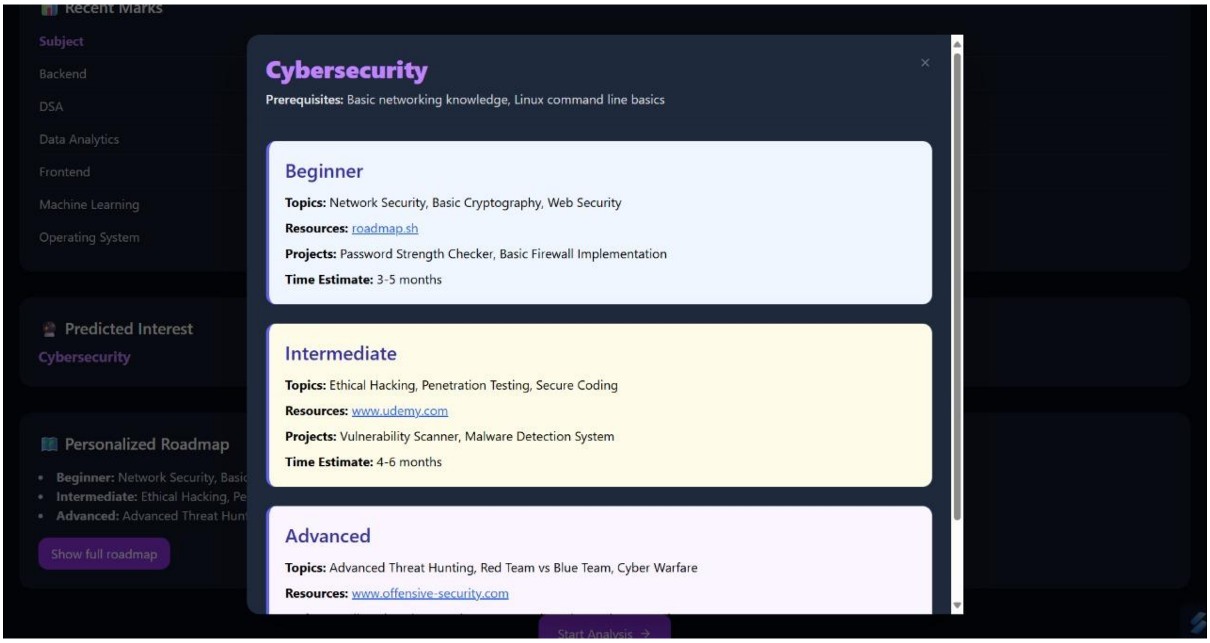
Predict student interests and create personalized improvement roadmaps using advanced AI analysis of attendance, marks, and project engagement.

96% Prediction Accuracy

100+ Students Analyzed

90% Improvement Rate

Figure 5. Homepage of SmartInterest AI Platform



- **Beginner:** Network Security, Basic
- **Intermediate:** Ethical Hacking, Pe
- **Advanced:** Advanced Threat Hunt

Cybersecurity

**Prerequisites:** Basic networking knowledge, Linux command line basics

Beginner

**Topics:** Network Security, Basic Cryptography, Web Security

**Resources:** roadmap.sh

**Projects:** Password Strength Checker, Basic Firewall Implementation

**Time Estimate:** 3-5 months

Intermediate

**Topics:** Ethical Hacking, Penetration Testing, Secure Coding

**Resources:** www.udemy.com

**Projects:** Vulnerability Scanner, Malware Detection System

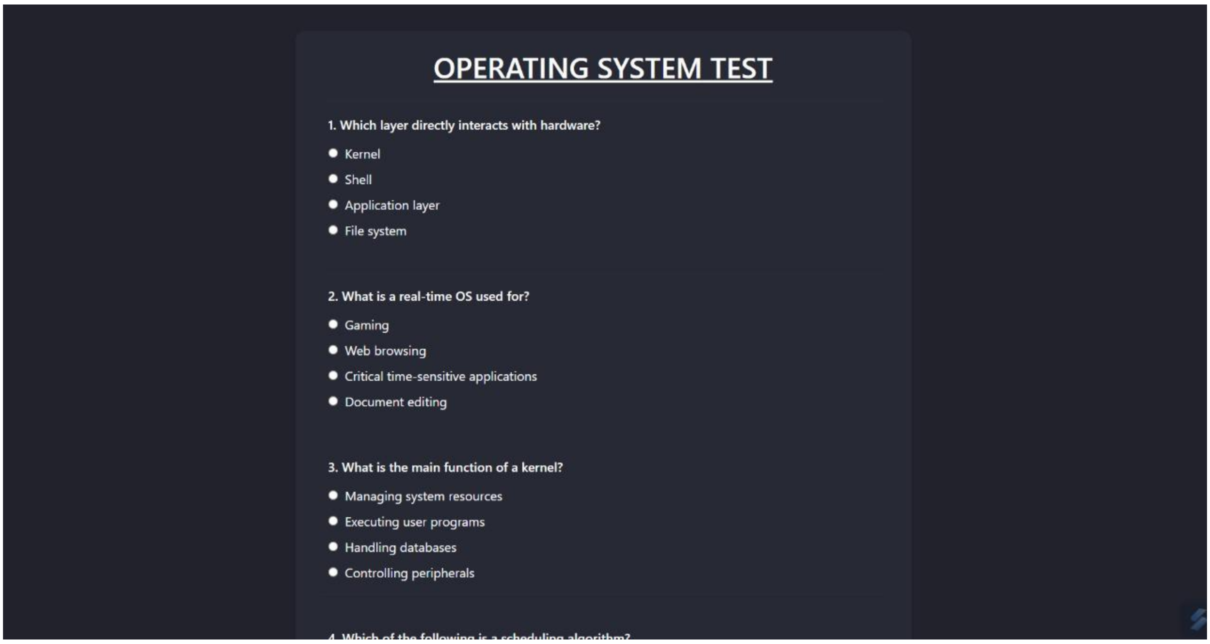
**Time Estimate:** 4-6 months

Advanced

**Topics:** Advanced Threat Hunting, Red Team vs Blue Team, Cyber Warfare

**Resources:** www.offensive-security.com

**Figure 6. Cybersecurity Personalized Roadmap Interface**



## OPERATING SYSTEM TEST

1. Which layer directly interacts with hardware?

- ○ Kernel
- ○ Shell
- ○ Application layer
- ○ File system

2. What is a real-time OS used for?

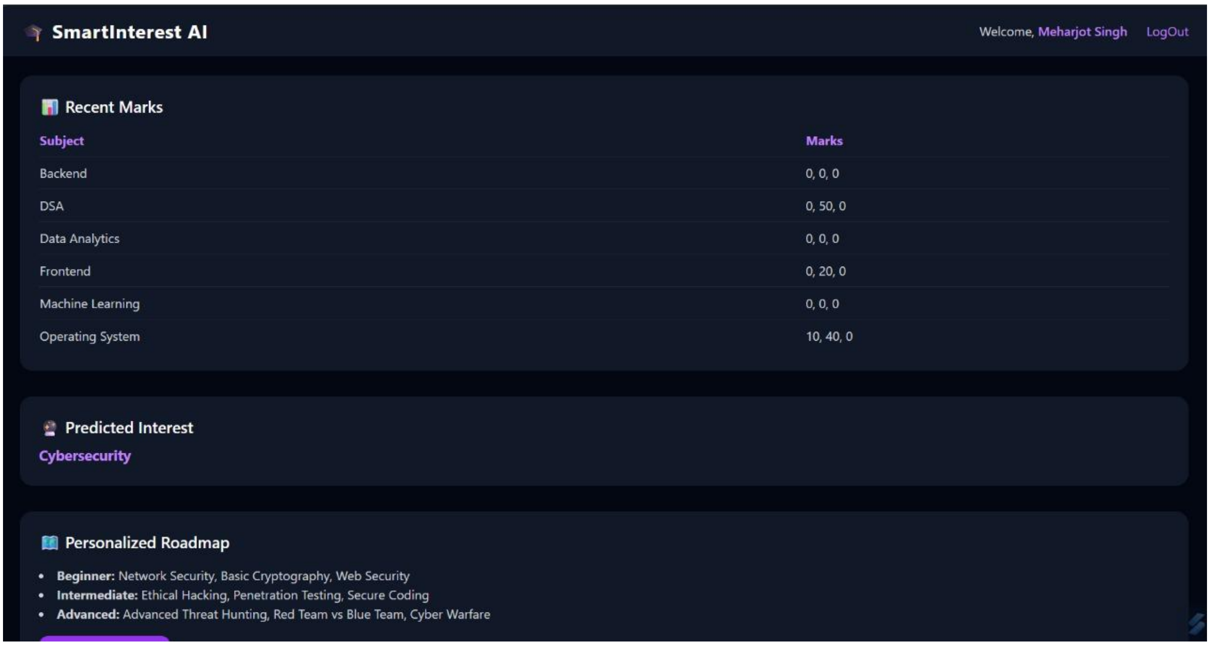
- ○ Gaming
- ○ Web browsing
- ○ Critical time-sensitive applications
- ○ Document editing

3. What is the main function of a kernel?

- ○ Managing system resources
- ○ Executing user programs
- ○ Handling databases
- ○ Controlling peripherals

4. Which of the following is a scheduling algorithm?

**Figure 7. Operating System Test Module**



Recent Marks

| Subject | Marks |
|---|---|
| Backend | 0, 0, 0 |
| DSA | 0, 50, 0 |
| Data Analytics | 0, 0, 0 |
| Frontend | 0, 20, 0 |
| Machine Learning | 0, 0, 0 |
| Operating System | 10, 40, 0 |

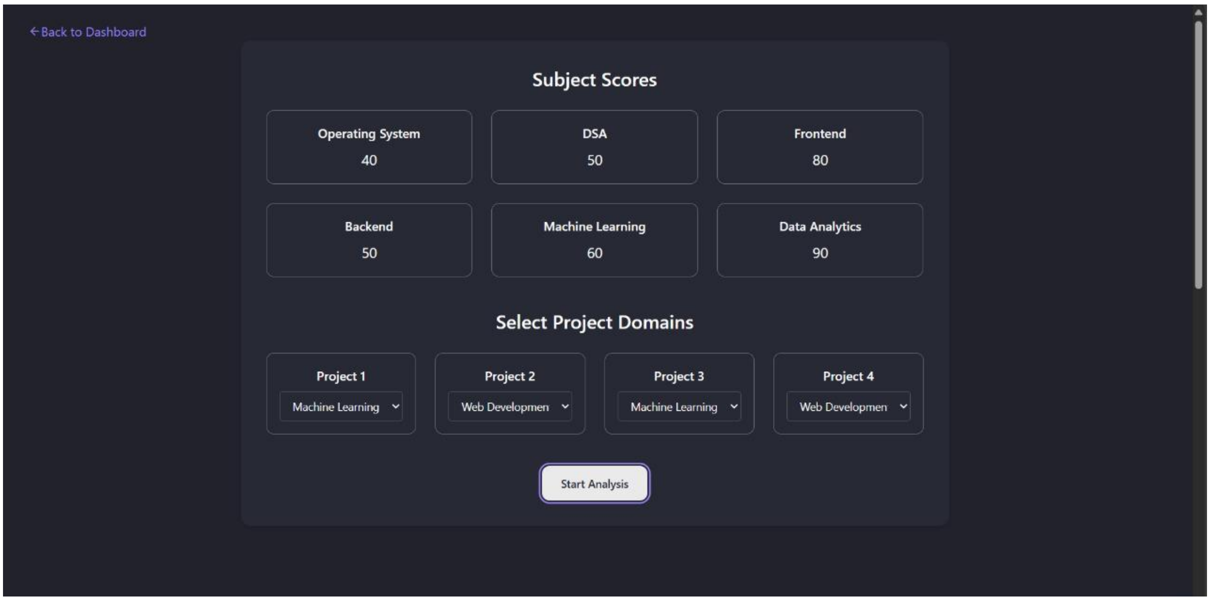
Predicted Interest

Cybersecurity

Personalized Roadmap

- **Beginner:** Network Security, Basic Cryptography, Web Security
- **Intermediate:** Ethical Hacking, Penetration Testing, Secure Coding
- **Advanced:** Advanced Threat Hunting, Red Team vs Blue Team, Cyber Warfare

**Figure 8. Subject Scores and Project Domain Selection**



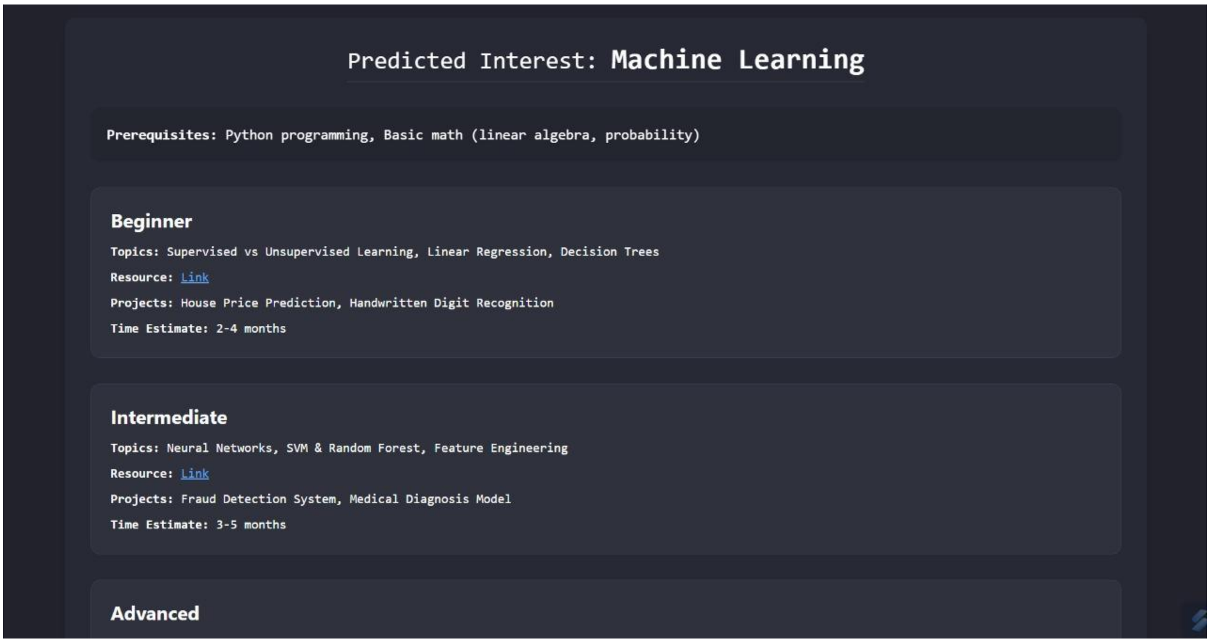
Subject Scores

| Operating System | DSA | Frontend |
|---|---|---|
| 40 | 50 | 80 |
| **Backend** | **Machine Learning** | **Data Analytics** |
| 50 | 60 | 90 |

Select Project Domains

| Project 1 | Project 2 | Project 3 | Project 4 |
|---|---|---|---|
| Machine Learning | Web Developmen | Machine Learning | Web Developmen |

**Figure 9. Dashboard for Academic Performance and Project Preferences**



Predicted Interest: **Machine Learning**

**Prerequisites:** Python programming, Basic math (linear algebra, probability)

**Beginner**

**Topics:** Supervised vs Unsupervised Learning, Linear Regression, Decision Trees

**Resource:** Link

**Projects:** House Price Prediction, Handwritten Digit Recognition

**Time Estimate:** 2-4 months

**Intermediate**

**Topics:** Neural Networks, SVM & Random Forest, Feature Engineering

**Resource:** Link

**Projects:** Fraud Detection System, Medical Diagnosis Model

**Time Estimate:** 3-5 months

**Advanced**

Figure 10. Personalized Machine Learning Roadmap Generated by SmartInterest AI

# 8. Conclusion and Future Work

This project demonstrates a robust and scalable platform that leverages machine learning to predict student interests and recommend tailored roadmaps based on their academic inputs and project history. By integrating a Flask-based backend, MongoDB for data persistence, and a modern React frontend, the system offers an end-to-end solution for personalized student guidance.

The predictive model accurately classifies a student's area of interest (e.g., Web Development, Machine Learning, Cybersecurity, AI, etc.) using key academic indicators, while the roadmap recommendation module supports students with structured project paths aligned to their predicted domain. This intelligent recommendation system aims to bridge the gap between student capability and career direction, fostering more informed, confident, and strategic learning decisions.

While the current implementation delivers meaningful insights and usability, it also opens numerous avenues for improvement and innovation:

- **Improved Prediction Accuracy**: Future versions could integrate deep learning models like neural networks or transformers for better generalization and accuracy in interest prediction.
- **Interactive Roadmaps**: Enable students to track their progress, get milestone recommendations, and dynamically update learning paths based on engagement.
- **AI Assistant Integration**: Embedding a smart chatbot assistant could enhance interactivity and provide real-time answers to student queries.
- **Gamification**: Adding achievement badges, project challenges, and leaderboards can improve user motivation and retention.
- **Feedback Mechanism**: Allowing students to rate their recommended roadmaps will help improve future suggestions via reinforcement learning techniques.
- **Admin and Analytics Dashboard**: A management interface for educators to monitor usage trends, project progress, and common interest areas could be developed.
- **Mobile Application**: A cross-platform mobile app would increase accessibility, enabling users to learn and explore anytime, anywhere.

- **Collaborative Projects**: Supporting group project formation and peer-to-peer mentorship can promote team-based experiential learning.

# 9. References: -

1. Kaggle.com

2. www.geeksforgeeks.org

3. W3school

4. Codebasics.com