A purple rectangular sign with white text

Description automatically generated

*A Synopsis on*

***AI-Powered personal tutor:A Scalable,Adaptive,Learning System for Enhanced Student Engagement***

***Submitted for the Intel Unnati Industrial Training Program 2024***

***Team***

**Nayana D.R (1NT22EC102)**

**Tushar DM (1NT22EC178)**

**Shivasai Lahari Mulakala (1NT22EC155)**

Under the Guidance of

**Dr.Ramachandra A.C**

Professor

Dept. of Electronics and Communication Engineering

Blue text on a white background

Description automatically generated

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**YELAHANKA, BENGALURU- 560064**

Contents

[CHAPTER 1 : Abstract 3](#_Toc194784874)

[CHAPTER 2 Introduction 3](#_Toc194784875)

[CHAPTER 3 : Elaboration on Problem Statement 4](#_Toc194784876)

[CHAPTER 4 : EDA and Modelling Process 4](#_Toc194784877)

[CHAPTER 5 : Solution 5](#_Toc194784878)

# : Abstract

This project is a smart AI tutoring system built to make learning more personal and meaningful. Students can take quizzes based on their chosen subjects, and the system analyzes their answers to estimate an IQ score by comparing them with real K12 student data. Based on how well they do, it gives thoughtful recommendations to help them improve and grow. It also keeps track of each user’s progress, so they can see how they’ve evolved over time. With a clean, user-friendly interface and intelligent feedback, this tool makes learning fun, insightful, and tailored just for each student.

# Introduction

**Introduction to the Project: AI-Powered Personal Tutor for K-12 Education**

The rapid advancements in Artificial Intelligence (AI) have transformed the educational landscape, offering innovative solutions to personalize learning and enhance student outcomes. This project proposes the development of an **AI-Powered Personal Tutor** specifically designed for K-12 students. The system aims to predict assessment scores, determine promotion eligibility, recommend starting modules, and dynamically adapt course tracks based on student performance. By leveraging adaptive learning technologies, this solution addresses the diverse needs of students while ensuring scalability and inclusivity.

**K-12 Case**:  
Primarily designed for students aged 5–18 in structured educational settings. The focus is on providing age-appropriate, curriculum-aligned content tailored to diverse learning needs. These systems often serve as supplementary tools to traditional classroom teaching or blended learning environments.Engagement is critical for younger students who require interactive and gamified elements to sustain interest. K-12 systems integrate multimedia content, real-time feedback, and teacher involvement to maintain high retention rates

AI tutors also predict assessment outcomes, recommend starting points for learners, and dynamically adjust course tracks based on ongoing performance. This adaptability ensures that students progress at their own pace while mastering required skills. Additionally, AI tools empower educators by automating routine tasks like grading and offering insights into student progress, enabling teachers to focus on meaningful interactions and active learning strategies.

While challenges like digital access disparities remain, AI's potential to democratize education and provide equitable support for all learners makes it a powerful tool for revolutionizing K-12 education

# : Elaboration on Problem Statement

AI-Powered Personal Tutor: A Scalable, Adaptive Learning System for Enhanced Student

Engagement

In the rapidly evolving educational landscape, personalized learning has become a key driver in improving student outcomes. However, traditional tutoring methods are limited by resources, scalability, and accessibility. The challenge is to develop an intelligent, scalable AI-powered personal tutor system that can provide tailored learning experiences to students in real-time, addressing individual needs and learning styles, while ensuring secure and personalized access through user authentication. Participants are encouraged to leverage Intel’s offerings, including hardware, software, and AI technologies, to build and optimize the solution for maximum performance, scalability, and efficiency.

# : EDA and Modelling Process

EDA:

The Exploratory Data Analysis (EDA) for this project involves a detailed examination of the K12 dataset used for IQ benchmarking. It includes understanding the distribution of IQ scores, analyzing subject-wise performance, and identifying correlations between scores and difficulty levels. Visualization techniques such as histograms, box plots, and scatter plots were used to highlight trends, outliers, and patterns. The EDA helped categorize question difficulty, understand student performance range, and fine-tune the IQ calculation logic. Insights from this analysis were instrumental in building a more personalized and adaptive recommendation system within the AI Tutoring platform, making the learning experience more effective.

Graphs Plotted

1. Histogram of Age Distribution Graph plotted against age and count

2. Scatterplot of Time Used Per Day vs Speed of Learning

3. Bar plot of Gender Distribution

4. Bar plot of Subject Distribution

5. Scatter plot for Time Used Per Day vs Speed of Learning

6. Box Plot for Assessment score by gender

7. Box Plot for Assessment score by subjects

8. Pairwise Relation btw Age, Time Used Per Day (hrs), Speed of Learning (1-10) and Assessment Score.

9. Correlation Heatmap of K-12 Student Profiles

Modelling Process:

1. Data Preparation

The first step involved exploring the dataset to understand the structure and types of variables present. Features such as 'Gender', 'Indian City', 'Indian State', and 'Track' were identified as categorical, while variables like 'Age', 'Time Used Per Day (hrs)', 'Speed of Learning', and 'Historical Assessment Average' were numerical. Categorical variables were encoded using label encoding to make them suitable for model input.

2. Data Splitting and Scaling

The data was divided into training and testing sets to evaluate the model's performance fairly. The numerical features were scaled using a standard scaler to ensure all variables were on a similar scale, which helps certain models perform better by improving convergence and stability.

3. Model Training

A Decision Tree Regressor was selected as the predictive model, which is suitable for regression tasks. This model was trained using the training dataset (features and corresponding target values). The model learns the patterns and relationships between the input features and the target variable through decision-based splits.

4. Model Prediction

After training, the model was used to make predictions on both the training and testing datasets. This step aimed to evaluate how well the model could generalize its learning to unseen data.

5. Evaluation of the Model

Three key performance metrics were used to evaluate the model's accuracy:

Mean Squared Error (MSE): Measures the average squared difference between actual and predicted values.

Mean Absolute Error (MAE): Represents the average absolute differences between the true and predicted outcomes.

R-squared (R²): Indicates the proportion of variance in the target variable explained by the model. These metrics were calculated for both training and test datasets to check for overfitting or underfitting.

6. Saving Predictions

The actual and predicted values for both training and test sets were saved in a structured format as a CSV file. This allows for easy sharing, visualization, and future reference.

Process:

1. User Interaction (Frontend – HTML & CSS): When a student accesses your stunning online application with a dark theme, the adventure starts. They are greeted with a neat form that asks for their name and their choice of topic (such as science or math). A POST request is used to send this input to the server.
2. Flask Gets Data (Backend): This is where your Flask application takes control. The user's name and selected topic are taken from the form by the server. It uses a session to save this data so that we can access it from anywhere on the website.
3. Dynamic Question Loading: The application loads a few pertinent questions from dataset.csv based on the subject that was chosen. Five questions are chosen at random to ensure that the test is always new and different. A fresh test page is then presented after these queries have been returned to the browser.
4. User Submits Answers: The user submits the form following their completion of the multiple-choice questions. Once more, Flask comes into play; it counts the number of correct answers, compares each to the proper one, and determines the final score.
5. 5. IQ Calculation: Here's where intellect comes into play. The program estimates the user's IQ based on performance (accuracy and score). K12.csv, a real educational dataset that includes actual IQ data, is cited in order to accomplish this. The program assigns the user to an approximate IQ band based on accuracy.
6. Recommendation Generation: This web app provides a customized recommendation based on the user's performance. Harder questions can be suggested if their score was exceptionally high. If not, it motivates them to concentrate on the fundamentals or practice more. This advice gives the outcome a more human touch.
7. Saving to Database (SQLite3): A local database (database.db) contains all of this information, including the user's name, subject, score, IQ score, and recommendation. This implies that their development is preserved and can be examined at a later time.
8. Viewing Previous Attempts: Flask retrieves all prior test attempts from the SQLite database when users access the history page, displaying them in a tidy table with scores, IQs, and subjects to assist students in monitoring their progress.

# : Solution

### ****Solution Offered by the Project****

What pupils learn and how their comprehension is evaluated frequently diverge in today's educational system. The majority of tests are static, one-size-fits-all methods that don't take into account a person's unique learning preferences, IQ, or developmental trajectory. By offering a dynamic and customized assessment experience driven by data-driven intelligence, your AI tutoring system fills this gap. Fundamentally, the system lets students choose a subject (like science or math) and enter their information. A list of questions chosen at random from a question bank (dataset.csv) is then displayed. The system automatically assesses the student's responses once they are submitted, determines their score, and—most crucially—compares the outcomes with a benchmark dataset (k12.csv) that contains standardized IQ scores.

Based on the student's performance, the algorithm creates personalized recommendations and uses this comparison to estimate the student's IQ. In addition to a single test, the system uses SQLite to track performance history, providing pupils with a glimpse into their development over time. In order to provide insightful feedback, it also takes previous scores into account. The addition of a personalized suggestion system motivates the learner to make progress, whether that means attempting more difficult content or concentrating on the basics.

**The Solution’s Impact on the World**

The world is moving toward personalized education an era where students are not confined by uniform teaching methods. Your project fits perfectly into this landscape. By integrating automated IQ assessment and adaptive recommendation, your solution is a small but powerful step toward democratizing intelligent education. It empowers students by helping them understand where they stand cognitively, and what path they can take for self-improvement, all without needing a human evaluator. It can especially be impactful in rural or under-resourced areas where access to qualified educators or counselors is limited. Your system offers automated, data-backed feedback something often available only in premium coaching centers. Moreover, educators can use this as a diagnostic tool to identify weak areas in students and adjust their teaching strategies accordingly.