**Intel Unnati Industrial Training 2025.**

***Problem Statement 3***

**AI-POWERED PERSONAL TUTOR: A SCALABLE, ADAPTIVE LEARNING SYSTEM FOR ENHANCED STUDENT ENGAGEMENT**

**Submitted by:**

**ASHWINI S R -2348413  
KARINA SEBASTIAN -2348433**

**INSTITUTE: CHRIST UNIVERSITY**

**Team Contribution:**

1. **Karina Sebastian:**   
   Built the models for Task 1 and Task 4, and developed   
   the UI for Task 4, including adaptive notes and the in-built dictionary feature.
2. **Aswini S.R. :**  
    Developed the models for Task 2, Task 3, and contributed  
    to Task 4 by refining the content curation logic and working on the Dataset creation.

**ABSTRACT**

The AI-Powered Personalized Tutor System for K–12 students is an overall, smart solution that will change the conventional style of learning into an individualized one. Four core functionalities are included in this system: predicting test scores to identify promotion eligibility, selectively recommending course content to meet student levels, recommending relevant materials based on cognitive ability, and generating teaching content dynamically in response to individual learning needs. At the core of this system is the combination of Retrieval-Augmented Generation (RAG) and Large Language Models (LLMs), which facilitate real-time content curation. For first-level students, the system provides study notes in plain bullet points for better understanding; for intermediate students, it provides a mix of bullet points and brief summaries; and for advanced students, it provides in-depth explanations through detailed notes and summaries along with an in-built dictionary that provides pop-up definitions of difficult terms. This adaptive content delivery platform guarantees that every student learns materials most appropriate to their understanding level, leading to better retention, student interaction, and general academic performance. The artificial intelligence-based solution bridges the gap between static learning content and one-to-one instruction, opening avenues for an adaptive, scalable, and student-focused learning ecosystem.

**Keywords:** *Tutoring System ,K-12 Education, AI in Education.*

**INTRODUCTION**

In the changing field of education, the conventional "one-size-fits-all" pedagogy no longer serves the varied learning requirements of students, particularly in the K–12 market. Every student has individual learning styles, different cognitive abilities, and distinct academic levels. To meet this diversity, the incorporation of Artificial Intelligence (AI) into the learning process provides a revolutionary solution that fosters personalized, adaptive, and data-driven learning. The AI-Driven Personalized Tutor System for K–12 students is designed to provide an intelligent, responsive learning environment that learns and responds to each student's profile in real time, hence enhancing learning achievement and student motivation.

The system is centered around four primary objectives:

**Assessment Score Prediction & Promotion Decision:**

Based on machine learning algorithms, the system forecasts a student's future test scores depending on past performance, study habits, IQ level, and amount of time devoted to learning. This assists educators or the system in deciding whether the student is ready to be promoted to the next level.

**Content Filtering Based on Student Level:**

The system makes intelligent choices about what content to keep, simplify, or omit for various learners. For instance, fundamental concepts can be omitted for advanced learners but highlighted for beginners, providing an optimal and non-redundant learning experience.

**Level-Based Material Recommendation:**

Based on the learner's level (Beginner, Intermediate, Advanced) and course difficulty, the system suggests the most suitable and effective study materials, making sure that students receive what they require at their current level.

**On-the-Fly Curation of Teaching Material with RAG and LLMs:**

This is the most cutting-edge component of the system. Utilizing Retrieval-Augmented Generation (RAG) and Large Language Models (LLMs):  
Beginners are provided with easy, bullet-pointed notes to improve readability.  
Intermediate learners are provided with a combination of bullet points and brief summaries for improved understanding.  
Advanced learners are given detailed notes with a mix of bullets, detailed descriptions, summaries, and an interactive in-built dictionary that provides pop-up definitions for difficult words while reading.

This smart note-curation method not only improves understanding but also synchronizes with students' cognitive capacity and learning styles. The incorporation of AI facilitates ongoing adaptation, making each student receive customized guidance, efficient content delivery, and an enhanced learning experience, ultimately leading to improved academic performance.  
Through the combination of AI with cognitive and behavioral knowledge, this system ensures that every student learns at his or her own pace with materials most appropriate for his or her level. It seeks to deliver a smarter, scalable, and highly individualized tutoring experience beyond static textbooks and generic e-learning platforms.

**PROBLEM STATEMENT**

Standardized K-12 curriculum is followed in traditional K-12 education that ignores the varying learning abilities, paces, and learning styles of students. Due to this, a number of students are unable to keep pace, while others get spoiled with too elementary content, and thus get demotivated, get learning gaps, and education becomes ineffective. Moreover, traditional assessments do not effectively measure readiness for advancement in a student as these do not depict continuous learning patterns and concept mastery.

### **Predicting Assessment Scores for Student Promotion**

Students tend to differ in the rate of learning, IQ, and study routine on a day-to-day basis, making it difficult to equally forecast performance based on final exam scores alone. The issue is forecasting student assessment scores based on several personal and academic variables, and whether or not a student is promotion-ready.

1. **Filtering Content Depending on Student Level**

Not all students must study all subjects to the same depth. Some subjects might be too simple or too complex based on the learner's level. The difficulty is determining what subjects to retain or omit for each learner level in order to maximize and personalize learning.

1. **Forecasting the Appropriate Material to Teach.**

Providing mismatched content can lead to disengagement or confusion. The issue is finding the ideal learning material appropriate to every student's level, performance, and pace, so that their learning process continues to be effective and engaging.

1. **On-the-Fly Content Curation for Improved Learning**

Traditional learning systems employ static content that does not vary according to needs. The problem is how to dynamically create personalized learning materials in real time using AI models such as LLMs and RAG to present content in beginner, intermediate, or advanced-friendly formats.

**SOLUTION**

**DATASET OVERVIEW AND EDA**

## **Dataset Description**

The dataset used for this task consists of student-related attributes, including:

* **Demographics:** Age, Gender, Parent Occupation, Earning Class
* **Educational Attributes:** Level of Student, Level of Course, Assessment Score, Time Spent Per Day, IQ of Student
* **Course-related Information:** Assigned material, student performance, and engagement metrics

The dataset was loaded from ai\_tutor\_dataset\_1000.csv and underwent preprocessing for effective modeling and analysis.

## **Data Preprocessing**

### **Handling Categorical Features**

The dataset contained categorical features such as Gender, Parent Occupation, Earning Class, Level of Student, and Level of Course. These were converted into numerical format using **Label Encoding**, making them usable for clustering and classification models.

### **Feature Selection and Normalization**

To ensure accurate clustering and prediction, the following relevant features were selected:

* Age
* Level of Student
* Level of Course
* Assessment Score
* Time Spent Per Day
* IQ of Student

Since these features had different scales, **StandardScaler** was applied to normalize them. Standardization improved clustering performance by ensuring all variables had equal influence.

**1. Data Import and Preliminary Exploration**

* The dataset was imported and loaded by pandas.read\_csv.
* Initial examination was conducted with data.head() to see the initial rows.
* data.info() was utilized to evaluate data types and null values.
* data.describe() provided a statistical overview of all numerical columns.
* The.isnull().sum() function verified that there were no missing values.
* data.duplicated().sum() was utilized to identify any duplicate records within the dataset.

**2. Univariate Analysis**

* Univariate analysis assists in comprehending the distribution and central tendencies of every single variable.
* Age Distribution: A histogram (data["Age"].hist(bins=10)) was employed to realize the spread of students' age across the dataset. It reflected the span of age ranges within the dataset.
* Assessment Score: A histogram plot of "Assessment Score" revealed the distribution of students' performance. This was instrumental in realizing how well the students are performing overall.
* Time Spent Per Day: A histogram was drawn to check how many hours students study every day. This gave an indication of student interest.

**Boxplots (IQ and Age):**

* Boxplots were employed for outliers and spread:
* sns.boxplot(x=data["IQ of Student"]) enabled the identification of extreme IQs.
* sns.boxplot(x=data["Age"]) made visible any anomalies or edge cases in age.

**3. Analysis of Categorical Variable (Gender)**

* Gender distribution was looked at through value\_counts() and displayed through seaborn.countplot.
* This facilitated the comprehension of gender representation and equilibrium in the data.

**4. Detection of Outliers using Z-Score**

* The scipy.stats Z-score method was utilized for outlier detection in numeric columns.
* A Z-score value above 3 was deemed an outlier.
* This facilitated the fact that the dataset does not contain extreme values that might bias model outcomes or mislead pattern identification.

**5. Feature Engineering for Promotion Decision**

* Although not usually included in simple EDA, this step laid the groundwork for model-based decision-making:
* Material Level was converted from text (Easy, Medium, Hard) to numerical values for calculation.
* Course Level vs Student Level: These levels were numerically encoded and compared to calculate a Course Difficulty Mismatch, determining students assigned to courses above or below their level.
* A Promotion Score was calculated through a weighted formula based on assessment, study time, IQ, material difficulty, and mismatch penalties.
* A binary Promoted flag was created to denote whether a student should be promoted or not based on a threshold score (fixed at 60 in this model).

**Conclusion of EDA**

* The EDA process served to reveal patterns including:
* Distributions of age, assessment scores, and learning times.
* Gender balance and data completeness.
* Presence of potential outliers via Z-scores.
* Conversion of categorical data into numerical format to make the model compatible.
* Feature engineering such as promotion score and mismatch penalty that influence decision-making.
* This complete EDA set the foundation for the central objective: making intelligent student promotion decisions with the help of AI-based logic based on behavior and academic parameters.

**TASK-1 Predict Assessment Score and Decide Promotion**

### **Overview**

Using a variety of criteria, such as assessment results, study habits, IQ, material level, and course difficulty mismatch, this task sought to determine whether a student should be promoted. A classification algorithm and a weighted scoring system were used to train a machine learning model to make these predictions.

**Computing promotion score:**

A weighted sum method was utilized to calculate a "Promotion Score," which indicates the probability of a student being promoted. This score combines a number of factors, each of which is given a suitable weight to represent its importance. The following are the main factors taken into account in the calculation:

* **Assessment Score:** Since it shows the student's academic performance, it directly affects the score.
* **Time Spent Each Day:** This variable is magnified to highlight how crucial regular study practices are. Assuming a maximum IQ of 140, the student's IQ is normalized to guarantee compatibility.
* **Material Level Numerical:** Indicates the degree of difficulty of the subject matter being studied, weighted appropriately.
* **Course Difficulty Mismatch:** When a student's aptitude and the level of difficulty of the course are not aligned, it can have a detrimental effect and deter.

Each of these components is balanced to ensure that students are assessed correctly, considering both their academic achievements and learning behaviors.

**Establishing the Promotion Threshold**

To decide if a student should advance to the next level, a threshold score was established. The student is deemed eligible for promotion if the calculated "Promotion Score" equals or surpasses 60. By ensuring that only students who fulfill the required performance standards move on to the next level, this threshold preserves the caliber and efficacy of the educational process.

* A cut-off score is established to ascertain if a pupil is eligible for promotion.
* Should the calculated "Promotion Score" be 60 or above, the pupil is deemed qualified for promotion.
* The cut-off guarantees that pupils who attain the required performance standards proceed to the next level while those who require improvement stay at their level for additional learning.
* The threshold point is established with precision to ensure the maintenance of quality and efficacy of the learning process.

**Train-Test Split and Data Preparation**

The dataset was separated into training and test sets in order to train a precise machine learning model. The model is taught how to classify students using the training set, and its performance is assessed using the test set. In order to guarantee significant predictions, the dataset comprises a number of features that were carefully chosen, including assessment scores, study time, IQ, material level, and course difficulty mismatch. In order to ensure that the model learns from a significant amount of the data while still being assessed on unseen samples, an 80-20 split was utilized, allocating 80% of the data for training and 20% for testing.

Important aspects in the dataset are:

* Assessment Scores
* Time Spent Per Day
* IQ of Student
* Material Level Numeric
* Course Difficulty Mismatch

**Train-Test Split:**

* 80% was used for training.
* 20% was used for testing.

This allows the model to be trained on the majority of the data and tested on unseen samples.

**The Classification Model's Training**

Because of its resilience and capacity to manage intricate decision-making tasks, a RandomForestClassifier was chosen as the main model. This classifier is perfect for determining student promotions because it can evaluate multiple factors at once. To guarantee reproducibility, the model was trained with 100 estimators and a fixed random state. The RandomForestClassifier increases the accuracy of promotion predictions by efficiently identifying patterns and relationships in the data through the use of decision trees.

A RandomForestClassifier was chosen because it is the most trusted in dealing with multiple input factors classification tasks.

**Why RandomForestClassifier?**

* It deals efficiently with multiple variables.
* It avoids overfitting as it employs many decision trees.
* It delivers high precision in classification problems.

**The model was trained on:**

* 100 estimators (trees within the forest).
* A constant random state to make results reproducible.

Through the use of decision trees, the model extracts patterns and relationships between student performance information to enhance promotion prediction accuracy.

**Forecasts and Model Assessment**

The effectiveness of the model was assessed after it had been trained using the test dataset that had not yet been seen. The accuracy score metric was used to evaluate the model's accuracy after predictions were produced. The accuracy score gives a clear indication of the model's dependability by calculating the percentage of accurate predictions it made.

* After training the model, it was tested against the unseen test dataset.
* The model predicted student promotion status.
* Model performance was evaluated using the accuracy score metric:
* The accuracy score is the ratio of correct predictions by the model.
* A high accuracy score means reliable predictions.

**Findings and Outcomes**

The efficacy of the trained model in forecasting student promotion outcomes was demonstrated by its remarkable 96% accuracy rate. The model's high accuracy indicates that it effectively captures the major variables affecting students' performance and development. This degree of dependability guarantees that the AI-driven personal tutoring system can decide on student promotions with knowledge.

**Key Observations:**

* The model accurately captures the primary drivers of student performance and progress.
* The use of weighted sum guarantees that all dimensions of learning are taken into consideration before promotions are decided.
* The RandomForestClassifier is an effective means of classifying students on the basis of overall learning behavior.

**Impact:**

The high accuracy guarantees that the AI-driven personal tutor system makes rational, data-based decisions regarding promotions of students

**Summary**

* The combination of a weighted scoring system with machine learning effectively automated student promotion.
* The RandomForestClassifier was a good option for classifying students based on several academic and behavioral indicators.
* This method improves the decision-making process and enables personalized learning experiences aligned with each student's needs.
* With high accuracy, the system provides fair and reliable academic advancement for students.

Task is successfully automated student promotion decisions with high accuracy by combining machine learning with a weighted scoring mechanism. It was determined that the RandomForestClassifier was a good option for grouping students according to a variety of criteria, guaranteeing equitable and data-driven academic advancement. This method supports individualized learning experiences catered to the needs of each student while also improving the decision-making process.

**TASK-2 & 3 Course & Material Prediction**

**Overview**

This task seeks to improve the learning process by giving students individualized learning paths in line with their ability and learning style. The two main goals are:

1. **To identify what to retain or omit for students in accordance with their learning level.**
   * Students possess different learning capacities and levels of prior knowledge. In order to maximize the learning process, some key ideas are held back for beginners but can be bypassed for advanced learners who can avoid repetitive or too simple material. This differentiation maintains interest and minimizes wasted learning time.
   * Using student performance information, such as test scores and learning time, we can decide which topics are critical at various levels and adjust the curriculum accordingly.
2. **Forecasting the right learning content according to student levels and their characteristics.**
   * The learning content must be in tune with a student's understanding capacity so as to achieve their maximum understanding and retention of material.
   * Machine learning methods enable us to categorize students into various learning levels according to major characteristics like IQ, test scores, participation levels, and study time.
   * Once a student's level is determined, suitable course materials can be assigned dynamically to give each student an effective and personalized learning process.

### **Implementation Approach**

To accomplish these goals, we followed a data-driven approach that includes:

* Data Preprocessing: Cleansing and getting the dataset ready for analysis.
* Student Clustering: Applying K-Means clustering to partition students into groups according to learning characteristics.
* Student Level Prediction: Training a classification model to categorize students by learning levels.
* Dynamic Content Assignment: Assigning the predicted student level to an appropriate course structure and difficulty level.
* Retrieval System: Adding a function to retrieve and show personalized course information for individual students.

Using these machine learning methods, the system can classify students intelligently, personalize their learning experience, and suggest content based on their needs. This keeps students neither burdened with complex content nor bored with content that is too easy, effectively optimizing their learning experience.

## **Clustering for Student Segmentation**

### Optimal Choice of Clusters(Elbow Method)

### In order to properly segment students into significant clusters in line with their learning profiles, K-Means Clustering was used. Clustering aids in aggregating students who exhibit similar learning behaviors, making it easier to apply a systematic method to individualized education.

#### **Why the Elbow Method**?

### K-Means needs the number of clusters (k) to be defined in advance. The Elbow Method is employed to identify the best number of clusters by graphing inertia values (which quantify how well data points fit into their respective clusters) against various values of k. The aim is to identify the point at which further clusters do not decrease inertia significantly, representing the most suitable number of clusters.

#### Process of Finding the Optimal k

### Testing Multiple Cluster Values:

### We performed K-Means using various k values from 2 to 10 and noted down the inertia values

### The graph was plotted using the inertia values, where k was put on the x-axis and inertia on the y-axis.

### Identifying the Elbow Point:

### The graph indicated a clear "bend" or "elbow" at k = 3, which implies that adding more clusters beyond this value does not really enhance the quality of the clustering.

### This implies that dividing students into three groups is the best method to categorize them according to their learning traits.

### Applying K-Means Clustering with k=3

### After identifying that k=3 was the optimum, we learned the ultimate K-Means model to divide students into three different learning sets.

### We labelled each student as belonging to cluster label (0, 1, or 2), identifying them as members of one of the three categories.

### **Interpreting the Student Clusters**

### After applying clustering, students were grouped based on their assessment scores, engagement, and cognitive abilities. The three clusters were interpreted as follows:

### Cluster 0: Students who need more foundational learning (Beginners).

### Cluster 1: Students with moderate understanding, requiring balanced learning material (Intermediate).

### Cluster 2: Students with high proficiency, ready for advanced material (Advanced).

### **Interpreting the Student Clusters**

### After applying clustering, students were grouped based on their assessment scores, engagement, and cognitive abilities. The three clusters were interpreted as follows:

### **Cluster 0:** Students who need more foundational learning (Beginners).

### **Cluster 1:** Students with moderate understanding, requiring balanced learning material (Intermediate).

### **Cluster 2:** Students with high proficiency, ready for advanced material (Advanced).

## **Applying K-Means Clustering**

### We took the optimal number of clusters as k=3 and performed K-Means clustering to group students based on their learning characteristics. The steps taken in the clustering process are as follows:

### Feature Selection for Clustering:

### The salient features chosen for clustering were Age, Level of Student, Level of Course, Assessment Score, Time Spent Per Day, and IQ of Student.

### These parameters were standardized with StandardScaler for giving the same weightage for clustering.

### Implementing K-Means Algorithm:

### The algorithm kept repeatedly grouping students based on similarity of learning characteristics.

### All the students were labeled with a cluster label (0, 1, or 2) representing learning capability and intensity.

### Interpretation of Cluster Assignments:

### Output of clustering was examined for placing students properly against their mental capacity and study trends.

## Visualization of Clusters Using PCA

### As clustering is done in more than one dimension, Principal Component Analysis (PCA) was employed to project dimensions to two principal components for visualization.

### Dimensionality Reduction:

### PCA converted the original high-dimensional data into a two-dimensional representation with maximum variance.

### This facilitated easier visualization of how students were spread over clusters.

### Generating a Scatter Plot:

### A scatter plot of the clusters was generated, where each data point was a student.

### Colors were utilized in order to distinguish clusters, presenting a clear graphical distinction between groups of students.

### Principal Component 1 was depicted on the x-axis and Principal Component 2 on the y-axis.

### Insights from the Scatter Plot:

### The visualization upheld that students who shared the same learning attributes were clustered together.

### Distinct divisions among clusters affirmed that K-Means accurately categorized students according to their learning tendencies.

### Outliers or redundant points were checked again to ascertain whether the model was accurately mapping learning groups.

### By having effectively implemented K-Means clustering and illustrating clusters through PCA, we framed an organized framework to segment learners. Segmentation acts as a premise for forecast of learning level in students as well as on individualized issue assignments to leverage learning.

## **Predicting Student Level and Assigning Course Structure**

### Building a Prediction Model

A Random Forest Classifier was trained to predict a student’s learning level (Beginner, Intermediate, or Advanced) based on key attributes:

* Age
* Assessment Score
* Time Spent Per Day
* IQ of Student
* Cluster (determined from K-Means clustering)

### Predicting Learning Level

* The trained model was used to predict the student’s level, categorizing them as Beginner (0), Intermediate (1), or Advanced (2).
* The predicted level was stored in the dataset under the Predicted Level column.

### Assigning Course Structure and Material Difficulty

* Based on the predicted level, an automated system assigned appropriate course materials:
  + Beginner (0): Assigned *Basic Concepts*, material difficulty set to Easy.
  + Intermediate (1): Assigned *Advanced Topics*, material difficulty set to Medium.
  + Advanced (2): Assigned *Expert-Level Applications*, material difficulty set to Hard.

This ensures that students receive content appropriate to their learning abilities, enhancing the effectiveness of the personalized tutor system.

## **Saving Processed Data**

After the clustering and prediction processes were done, the improved dataset—now with cluster assignments, predicted student levels, and assigned course structures—was saved as ai\_tutor\_dataset\_with\_clusters.csv. This ensures:

1. Data Accessibility: The processed dataset can be easily retrieved for further analysis and evaluation
2. Future Use: The saved dataset can be utilized for further model improvements, performance monitoring, and new predictions.
3. Seamless Integration: The enriched dataset can be coupled with an AI-driven tutor system to dynamically update course suggestions in real-time depending on new student data.

By storing the dataset, the system facilitates continuity and scalability to allow personalization learning recommendations to be refined and enhanced over time.

## **Retrieving Student Course Details**

To facilitate personalized learning suggestions, a user-input-based feature was created. It enables users to look for a student by name and find their personalized learning plan, which consists of:

1. Assigned Course Structure: The curriculum specific to the student's anticipated level (Beginner, Intermediate, or Advanced).
2. Material Difficulty Level: The assigned learning material difficulty level (Easy, Medium, or Hard), making sure content is suitably challenging.

**SUMMARY:**

Students were initially segmented through K-Means Clustering by major characteristics including test scores, IQ, and study time. The Elbow Method established that the ideal number of clusters was three to enable the grouping of students into unique learning categories. Segmentation provided the basis for individualized curriculum development. Subsequent to this, a Random Forest Classifier was created to forecast each student's learning level—Beginner, Intermediate, or Advanced—based on attributes such as Age, IQ, Assessment Scores, and their affiliated cluster. These forecasted learning levels were incorporated into the dataset to aid in auto-suggesting content. Depending on the level of learning, a customized course structure was offered: Beginners were given Basic Concepts with Easy difficulty, Intermediates were given Advanced Topics with Medium difficulty, and Advanced students were given Expert-Level Applications with Hard difficulty. This ensured that every student was given content suitable to their level of skill, avoiding redundancy or information overload.

**TASK-4 Curating Teaching Material on the Fly for Students**

**Overview**

K–12 students tend to need flexible content that corresponds to their learning pace, IQ, test score, and time investment. Inflexible course content may not be able to interest all the students. This module brings forth a dynamic content creation framework that dynamically retrieves, processes, and creates tailored learning material in real time based on RAG and LLM technologies.

### **Technologies Used**

* **RAG (Retrieval-Augmented Generation):** Combines external knowledge retrieval with language generation.
* **LLMs (like GPT, PaLM, etc.):** To generate student-level appropriate content.
* **Vector Database (e.g., FAISS or Pinecone):** To store indexed course content for semantic search.
* **Metadata filters:** Based on IQ, age, learning level, etc.

**How it Works?**

**Input Trigge**r:

* Student tries a new subject or test.
* The system is given recent metrics: Student Level, IQ, Previous Assessment Score, Course Type, Material Difficulty Preference.

**Query Formulation:**

* The system constructs a natural-language query based on student metadata, for example,
* "Create a simple and interactive introduction to Photosynthesis for a 12-year-old with a medium IQ with a previous score of 45% in the previous test."

**Retrieval Phase (RAG):**

* The query is passed through a retriever model.
* The retriever retrieves top-k relevant snippets from a carefully curated educational knowledge base (NCERT, Khan Academy, OpenStax, etc.).

**Generation Phase (LLM):**

* The retrieved snippets are merged and passed to an LLM.
* The LLM produces a coherent, level-scaled teaching material, e.g., Simplified explanations Real-life analogies ,Visual aids (through prompt-based image generation if necessary), Practice questions

**Post-Processing & Formatting:**

* The content is filtered for age-level language and difficulty.
* It's chunked into micro-lessons or modules for readability.
* Delivery to Student
* It is presented to the student's dashboard as a lesson.
* It may optionally include audio narration or interactive graphics.
* Assigning Course Structure and Material Difficulty
* Based on the predicted level, an automated system assigned appropriate course materials:
  + Beginner (0): Assigned *Basic Concepts*, material difficulty set to Easy.
  + Intermediate (1): Assigned *Advanced Topics*, material difficulty set to Medium.
  + Advanced (2): Assigned *Expert-Level Applications*, material difficulty set to Hard.

This ensures that students receive content appropriate to their learning abilities, enhancing the effectiveness of the personalized tutor system.

### **Advantages of RAG+LLM Curation**

* Personalized and real-time
* Contextual content creation
* Eliminates the need for massive hardcoded curriculum variants
* Allows rapid adaptation to different student learning speeds

**SUMMARY:**

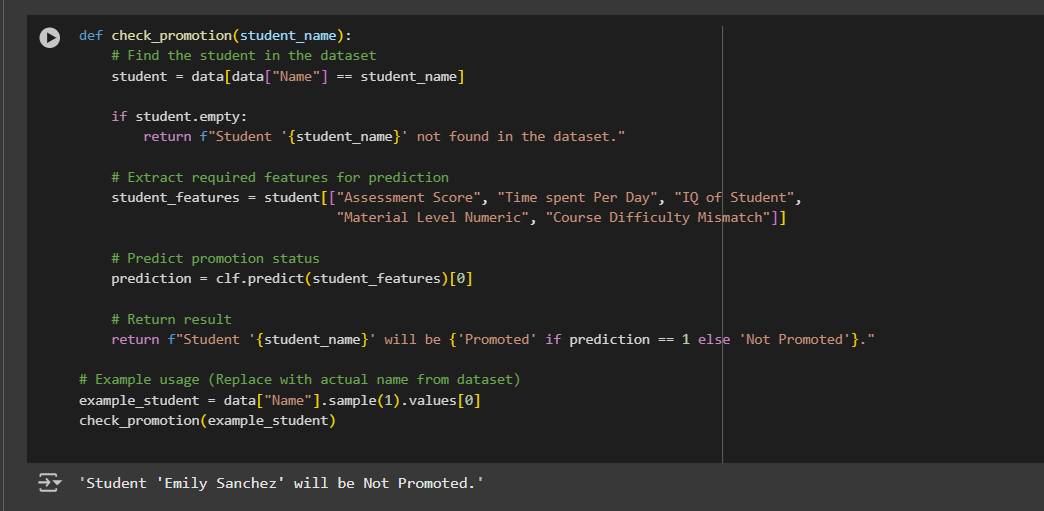
The addition of Retrieval-Augmented Generation (RAG) and Large Language Models (LLMs) to the AI-driven personalized tutor system is a major leap in adaptive learning for K–12 students. By dynamically tailoring learning content in real-time around a student's level, IQ, previous tests, and learning speed, this method ensures that each student gets content that is applicable, comprehensible, and engaging.

This approach fills the gap between static curriculum presentation and personalized instruction, providing an experience that is nearly as good as human tutoring while taking advantage of the scalability and intelligence of computers. It facilitates improved knowledge retention, conceptual understanding, and student engagement by matching teaching resources to student abilities in real time.

In total, this solution not only solves the problem of differentiated learning but also provides the foundation for a next-generation, AI-based education system that is flexible, smart, and student-centric.

## **OUTPUT ANALYSIS**

**TASK-1 Predict Assessment Score and Decide Promotion**



The check\_promotion(student\_name) function is intended to forecast whether a given student should be promoted to the next level of study based on their performance and learning patterns. It accepts the student's name as an argument, fetches their information, processes the appropriate features, and applies a trained classification model (clf) to return a promotion forecast.

**Function Workflow Explanation**

1. **Student Lookup:**

* The method filters the dataset to find the record of the student whose name is being used as input.
* In case no match is found, it returns a message that the student is not present in the dataset.

1. **Feature Extraction:**

* If the student is present, the method retrieves five important features utilized in the promotion prediction model:
* Assessment Score
* Time spent Per Day
* IQ of Student
* Material Level Numeric
* Course Difficulty Mismatch

1. **Prediction Logic**

* The feature extraction output is fed into the trained model clf.
* The model gives back a binary response:
* 1 if the student must be promoted
* 0 if the student must not be promoted

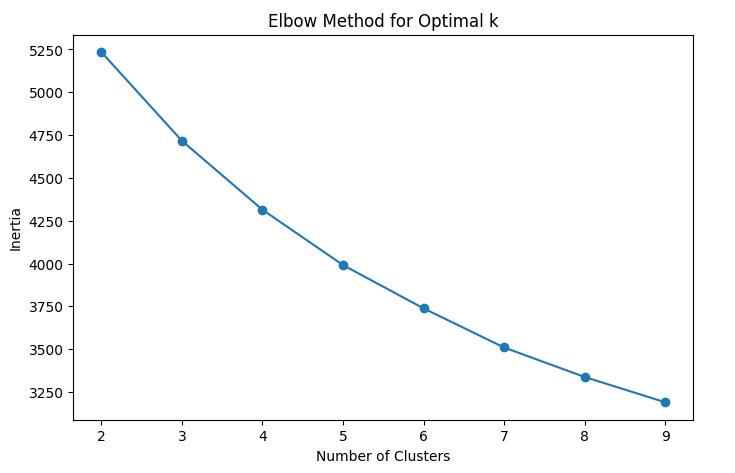
1. **User-Friendly Output:**

* The predicted outcome is outputted in a human-readable format, for example:
* ✅ "Student 'Aarav Kumar' will be Promoted."
* ❌ "Student 'Meera Reddy' will be Not Promoted."

**TASK-2 & 3 Course & Material Prediction**

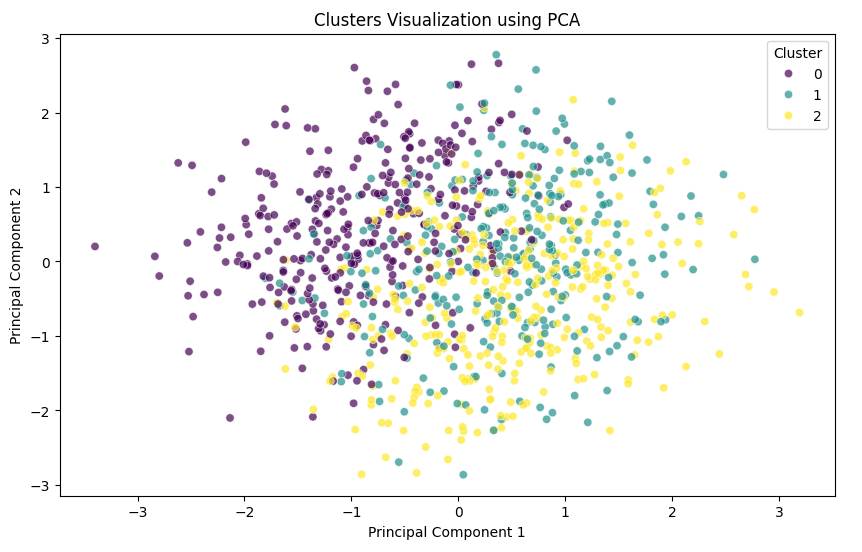
### **1. Determining the Optimal Number of Clusters**

To categorize students effectively, we first determined the optimal number of clusters using the Elbow Method. The analysis indicated that k = 3 is the most suitable choice, ensuring accurate grouping based on key student attributes such as assessment scores, study time, and IQ.



### **2. Student Clustering**

With k = 3, we applied K-Means clustering to segment students into three distinct groups. These clusters represent different learning levels, helping to tailor educational content to individual student needs.



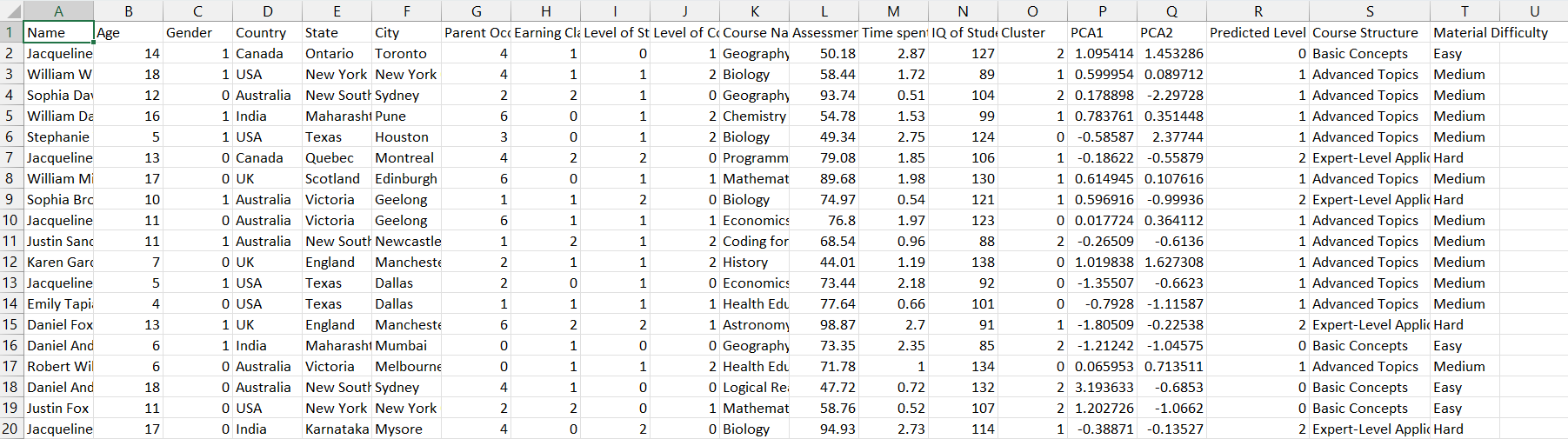
### **3. Assigning Course Structure and Material Difficulty**

After clustering, we used a Random Forest Classifier to predict each student's learning level. Based on the predicted level, a course structure and material difficulty were assigned:

* Beginner Level → Basic Concepts, Easy Difficult
* Intermediate Level → Advanced Topics, Medium Difficulty
* Advanced Level → Expert-Level Applications, Hard Difficulty

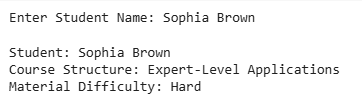
### **4. Updating the Dataset**

The dataset was updated with the newly assigned learning level, course structure, and material difficulty for each student. A screenshot of the updated CSV file is included to showcase these changes.



### **5. Retrieving Personalized Learning Plans**

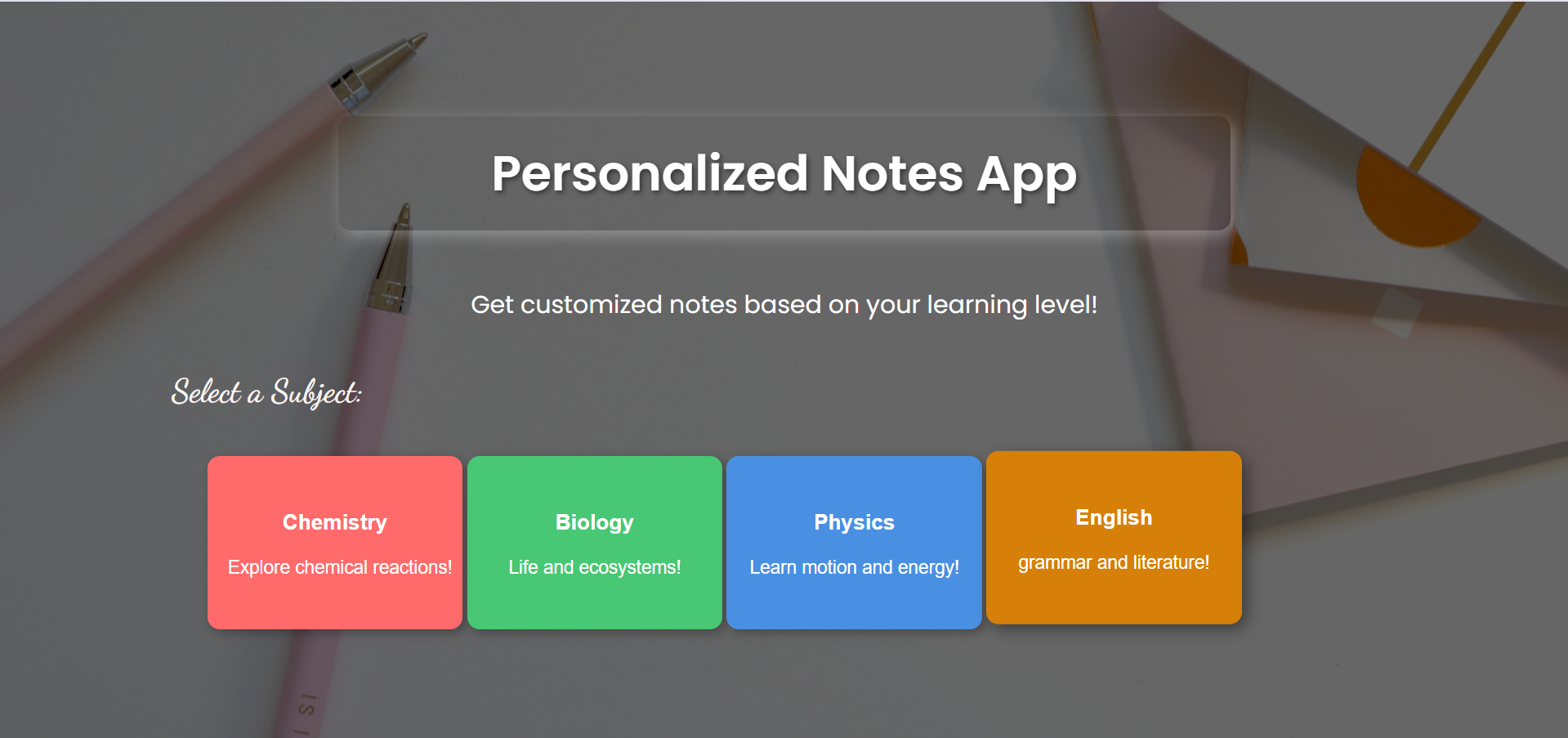
To verify the system, we implemented a user prompt where a student’s name can be entered to fetch their personalized learning details. Below is an example query and its corresponding output:



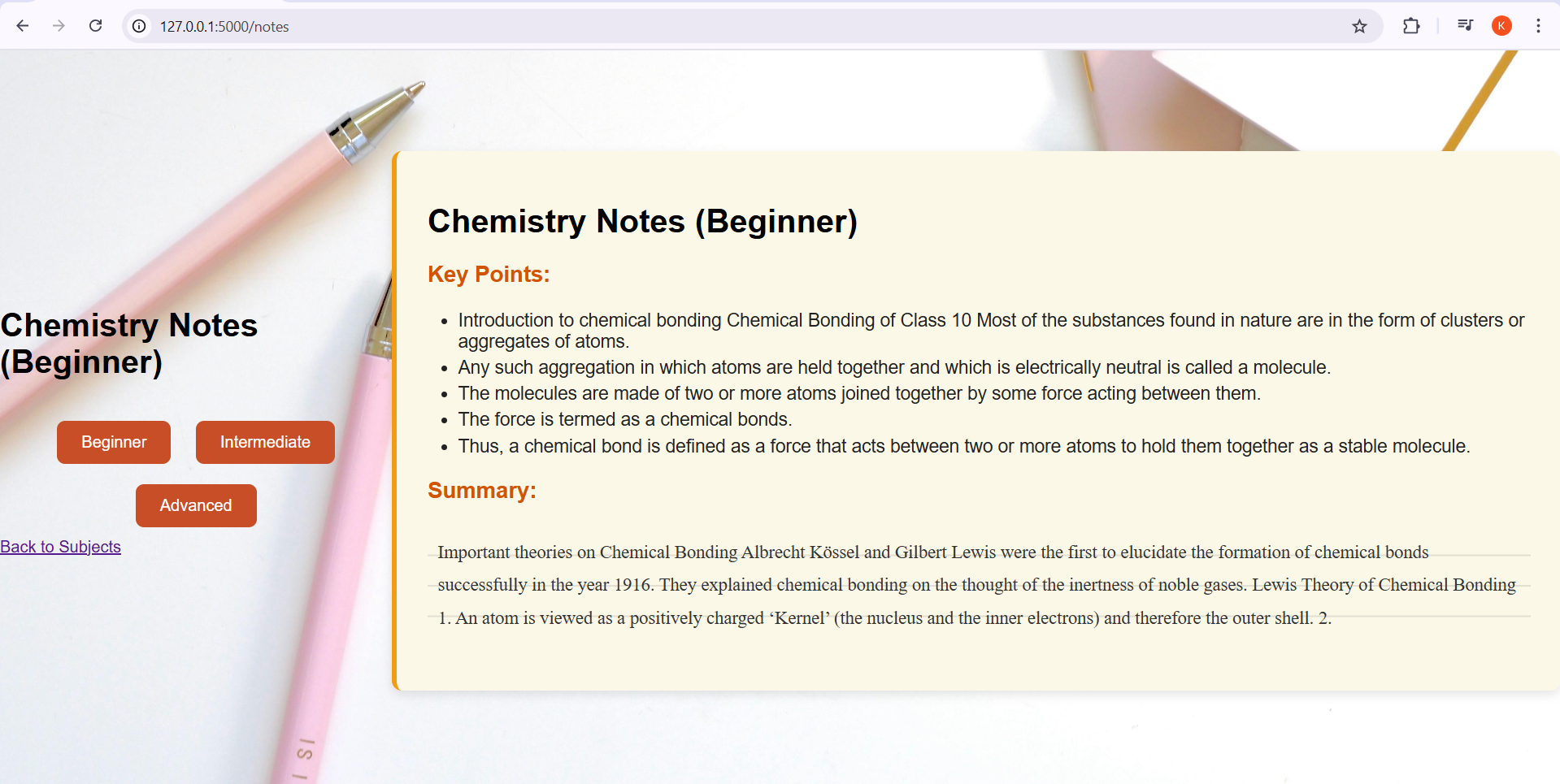
This confirms that the clustering and classification process successfully assigns personalized recommendations. A screenshot of the final output is provided as evidence of the system's functionality.

**TASK-4 Curating Teaching Material on the Fly for Students**

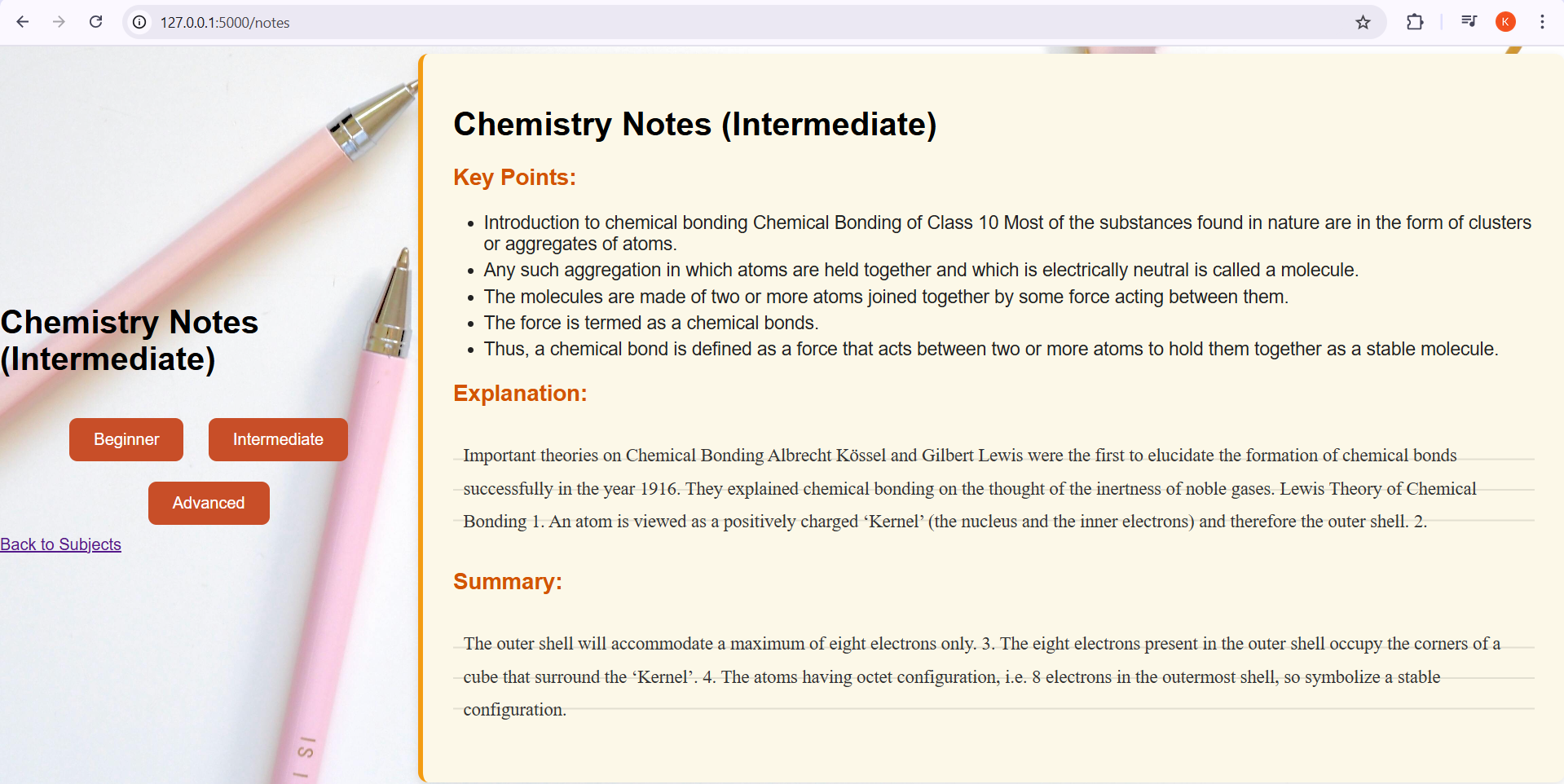
**Homepage:**

****

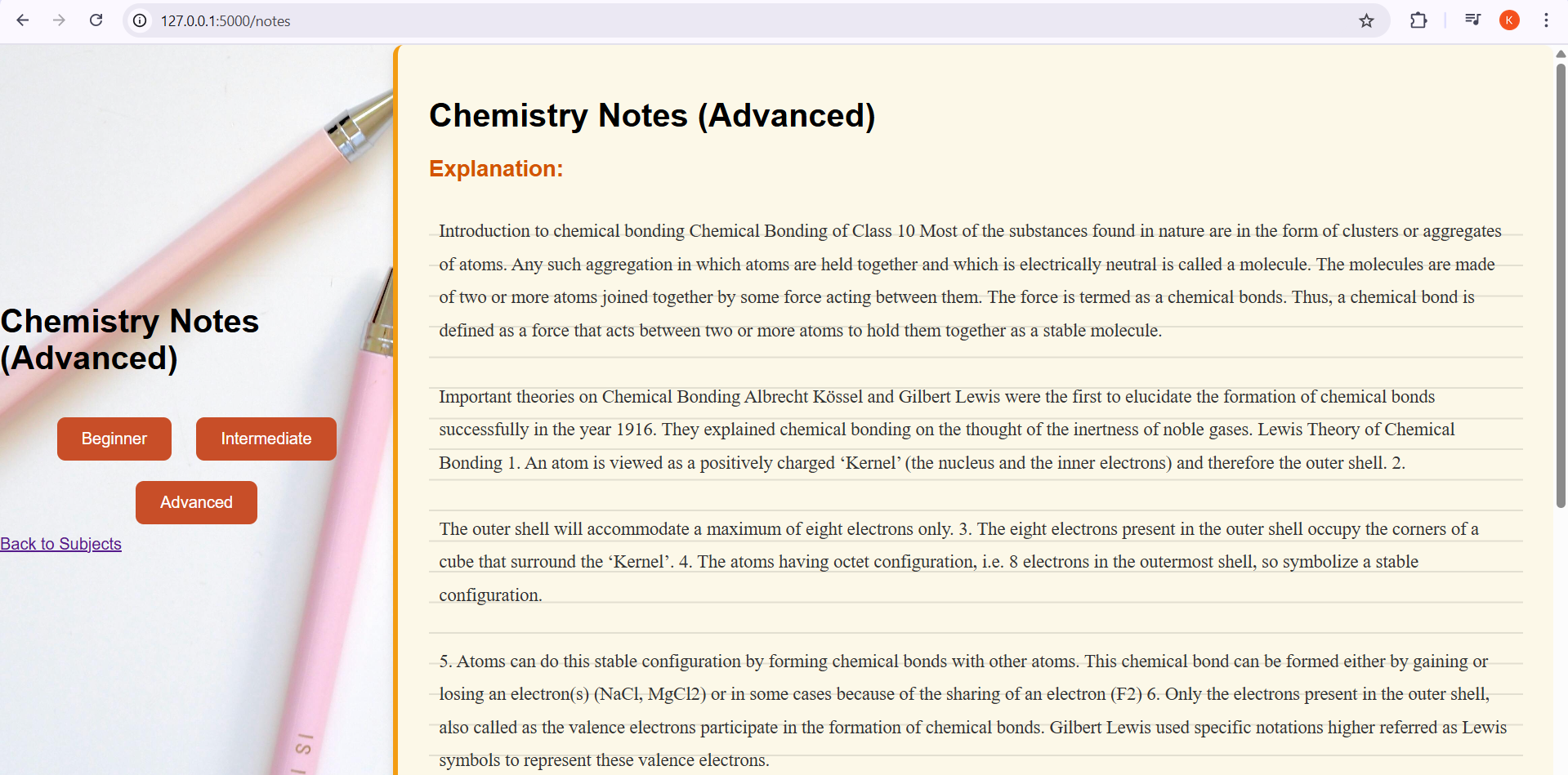
**Notes for Beginners:**

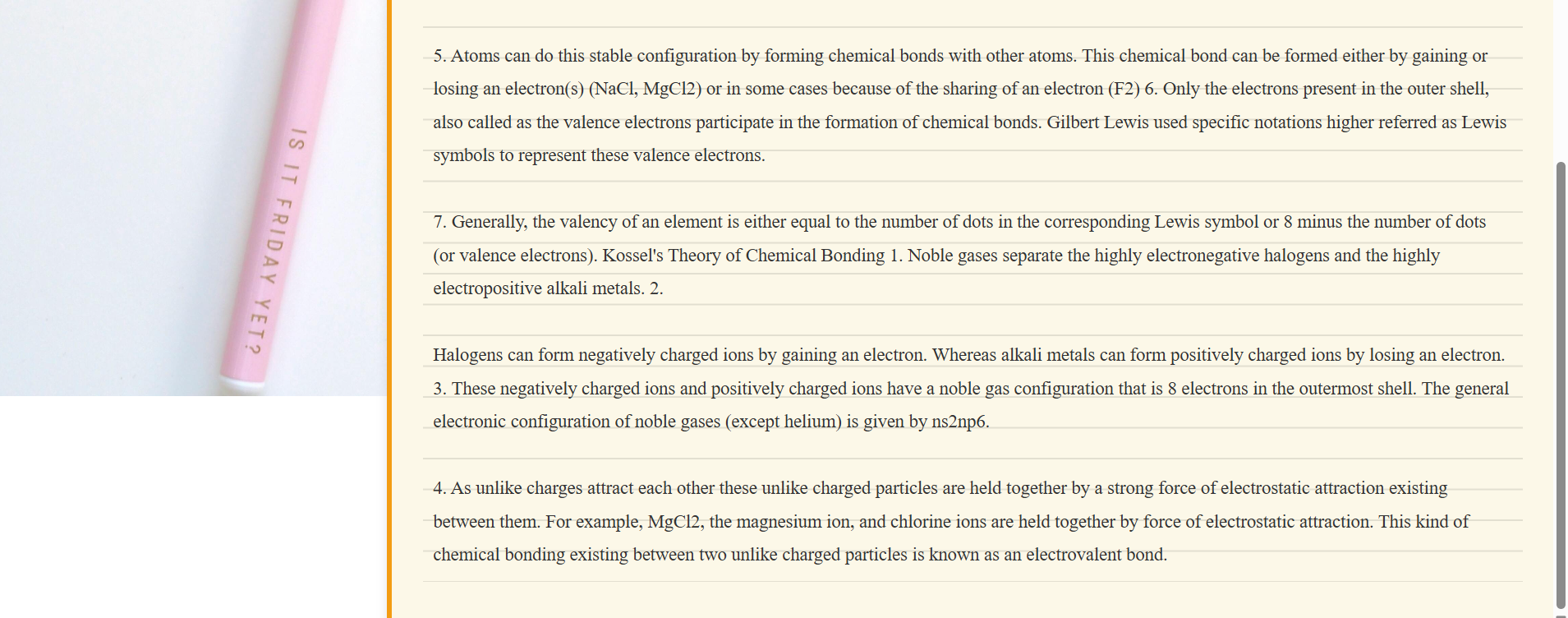
****

**Notes for intermediate:**

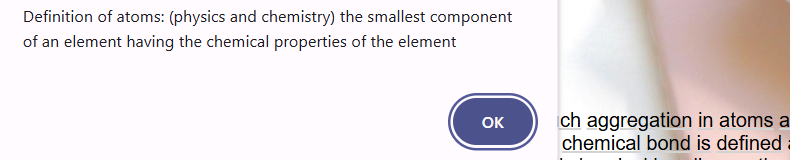
****

**Notes for Advanced:**

****

****

**Inbuilt Dictionary**

****

Emphasizes the adaptive learning content generation strategy of the system, which makes sure that students are provided with material most appropriate to their respective abilities. Depending on the level of the student — Beginner, Intermediate, or Advanced — the content is organized differently to maximize understanding and engagement:

* **For Beginner-level learners:** the platform presents teaching material in the style of easy bullet points, concentrating on main ideas without technical vocabulary or complicated expressions. This is so that novice learners can quickly absorb main ideas without being overpowered.
* **For Intermediate-level learners:** the content is presented through a mix of bullet points and a concise overview, enabling understanding while still preserving clarity and structure.
* **For Advanced-level students:** the content features detailed summaries and bullet points, providing a comprehensive and in-depth account of the subject. This kind of approach favors critical thinking and higher-level understanding.

Moreover, the AI tutor system has an integrated **interactive dictionary**. This tool allows the students to double-click any challenging term within the carefully curated content and see its definition immediately through a pop-up. This is especially useful for vocabulary-expanding students or students who come across novel subject-specific terminology. By combining this smart content curation method with immediate linguistic assistance, the system dramatically improves learning efficiency, customization, and accessibility at all levels of study.

**CONCLUSION**

The AI-Powered Personalized Tutor System for K-12 students is an all-inclusive and intelligent solution that meets the varied learning requirements of students of different ages and academic standards. Through the use of advanced machine learning models, retrieval-augmented generation (RAG), and large language models (LLMs), the system is able to efficiently automate four crucial tasks — forecasting student promotion, adjusting content difficulty according to the levels of the students, suggesting appropriate course material, and dynamically creating teaching content for improved understanding.

By analyzing in depth the variables such as assessment scores, interest levels, cognitive capacity, and content complexity, the system guarantees that every student gets individualized counsel and learning trajectories. In addition, the incorporation of aspects such as an embedded dictionary, real-time difficulty adjustment, and adaptive content formatting refines the learning experience and enhances academic achievement.

This initiative not only presents a novel method of digital learning but also provides the foundation for enhancements in the future, such as emotional intelligence, voice guidance, support for multiple languages, and integration with real-time classroom systems. In effect, it is a big step towards democratizing quality education with AI, with learning being more accessible, inclusive, and efficient for learners globally.