***AI Personal Tutor***

**1. Introduction**

**1.1 Problem Statement**

In-classroom or hard-coded e-learning models follow a "one-size-fits-all" approach and do not account for individual learner needs. They lack the flexibility required for different rates of learning, learning modalities, or knowledge gaps, resulting in learning wastage and disengagement. This issue can be addressed with an adaptive system capable of:

* Adapting learning content based on a student's past performance and needs.
* Dynamically adjusting content to challenge students without overwhelming them.
* Providing instant feedback on their progress for continuous adjustment.

**1.2 Objective**

The primary goal of the AI Personal Tutor project is to develop an adaptive machine learning-based tutoring system that enhances learning experiences. The main objectives include:

* Predicting whether a student is likely to answer a question correctly based on past performance and responses.
* Providing personalized learning content or practice questions to strengthen weaknesses and reinforce strengths.
* Continuously monitoring a learner's performance and refining recommendations to optimize learning outcomes.

**2. Literature Review**

The application of Artificial Intelligence (AI) in education has significantly transformed classroom dynamics, making adaptive learning systems a cornerstone of modern education. Techniques such as Knowledge Tracing, Collaborative Filtering, and Performance Prediction are crucial in tailoring learning experiences to individual student needs.

Knowledge Tracing effectively uses deep learning models like Long Short-Term Memory (LSTM) networks to predict future performance from past activities, adapting educational material dynamically to individual learning paths [3].

Collaborative Filtering leverages techniques like Singular Value Decomposition (SVD) to recommend content based on user behavior patterns, uncovering hidden learning activities and effectively personalizing learning trajectories [1]. Performance Prediction models, such as XGBoost, analyze response times and accuracy to forecast student performance, helping identify at-risk students and allowing timely intervention by educators [2, 3].

This project aims to develop a comprehensive system that not only recommends personalized learning activities but also predicts academic performance, thereby enhancing academic achievements among students.

**3. Technologies Employed**

**Backend:**

* FastAPI: A robust framework ideal for building backend APIs, supporting asynchronous calls and optimized model inference.
* TensorFlow: Powers the LSTM model for knowledge tracing, with flexibility for efficient training and deployment of deep learning models.
* XGBoost: Analyzes past behavioral data and various characteristics to predict student performance.
* Pickle: Serializes trained models for efficient storage and future inference without the need for retraining, facilitating easy deployment.

**Frontend**:

* Streamlit: A Python library used to quickly develop interactive dashboards, providing a live user interface for students and educators to interact with the AI tutor.

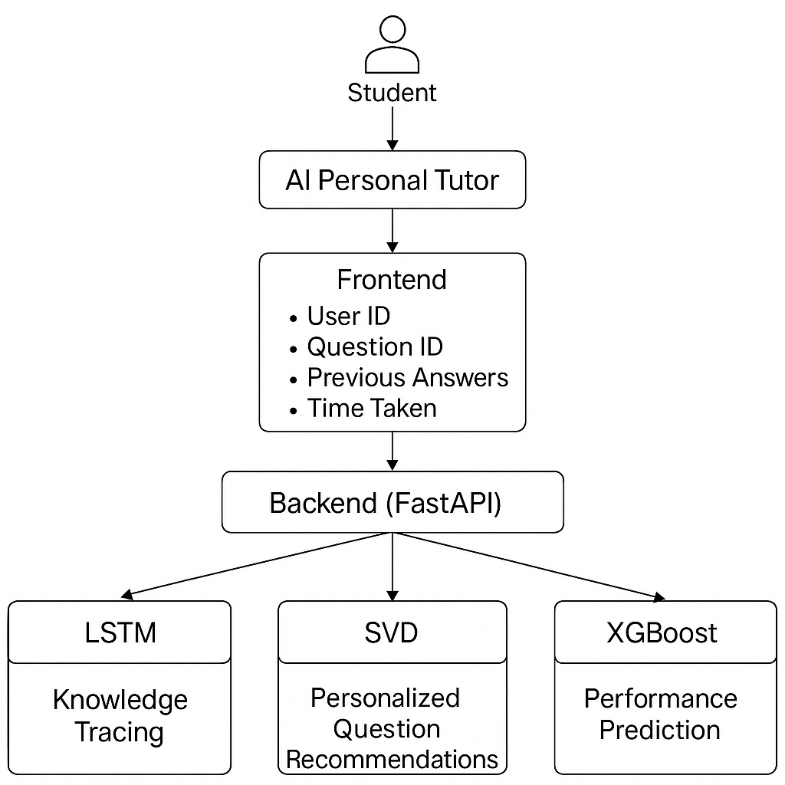
**Other Libraries**:

* NumPy, Pandas: Essential for data manipulation including cleaning, normalization, and processing.
* Matplotlib, Seaborn: Generate visualizations to analyze data distributions, model performance, and predictions.
* Scikit-learn: Used for preprocessing, splitting datasets, and evaluating model performance.

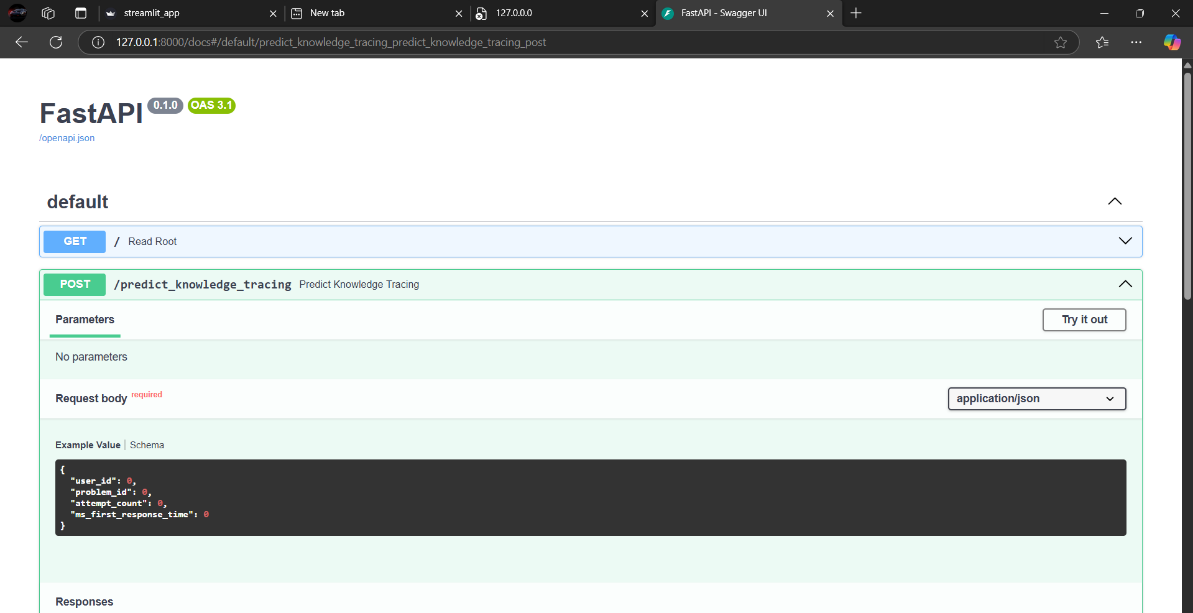
**4. System Architecture**

The AI Personal Tutor system architecture comprises several components:

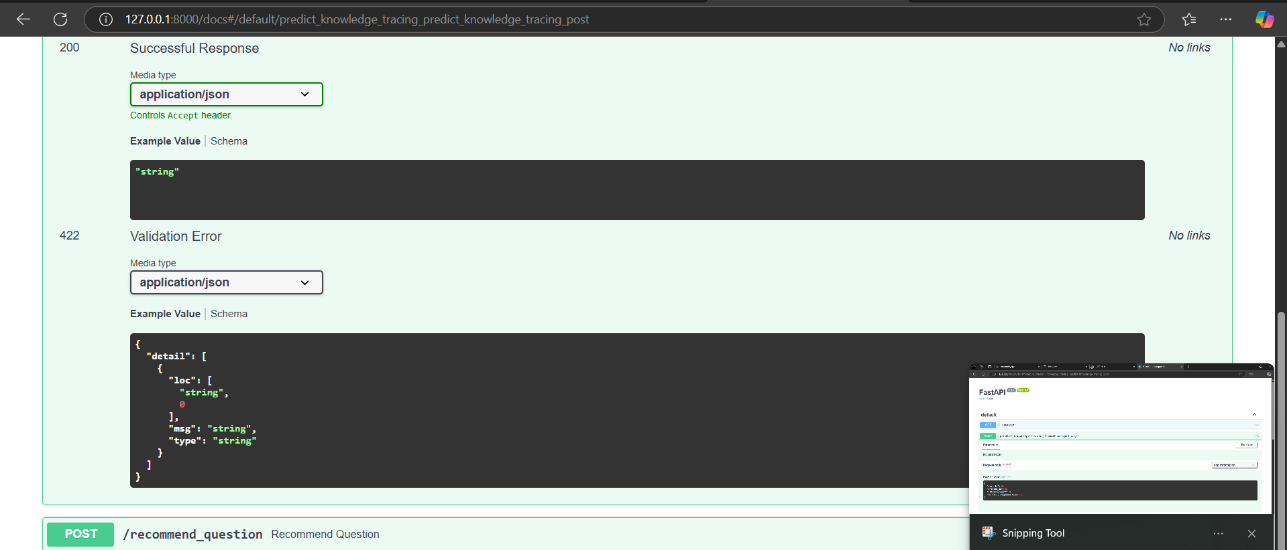
* Frontend (Streamlit): Collects user inputs such as user ID, question ID, previous answers, and time taken to answer questions, providing a user-friendly interface.
* Backend (FastAPI): Serves as a conduit between the frontend and machine learning models, handling data input and delivering real-time predictions and recommendations.
* Machine Learning Models:
  + LSTM: Manages knowledge tracing and predicts a student's ability to answer questions correctly.
  + SVD: A collaborative filtering model that suggests questions based on past performance.
  + XGBoost: Predicts future performance based on comprehensive student data



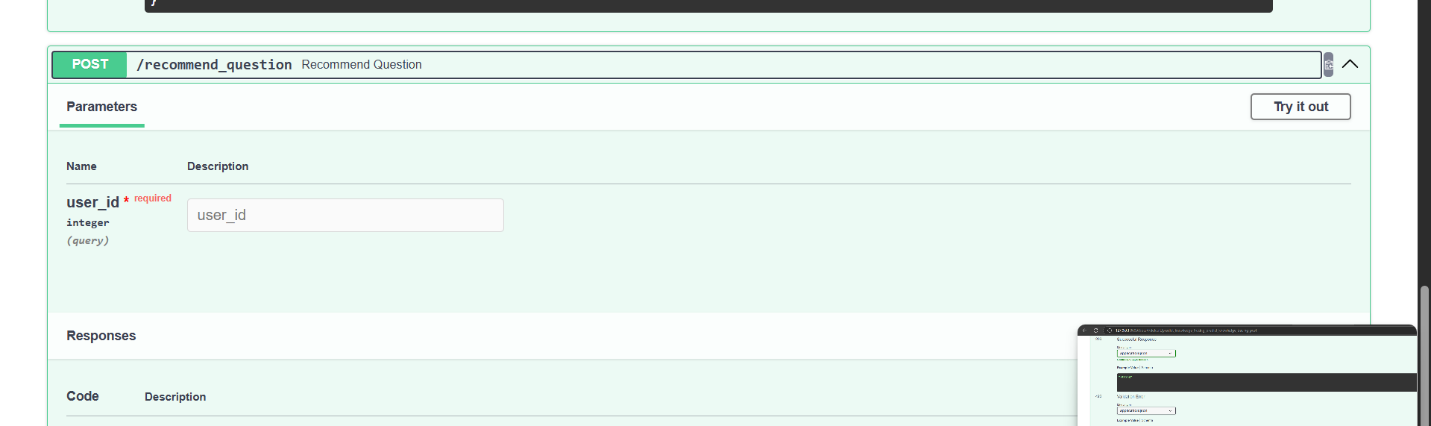
**Screenshot of FastAPI Documentation:**



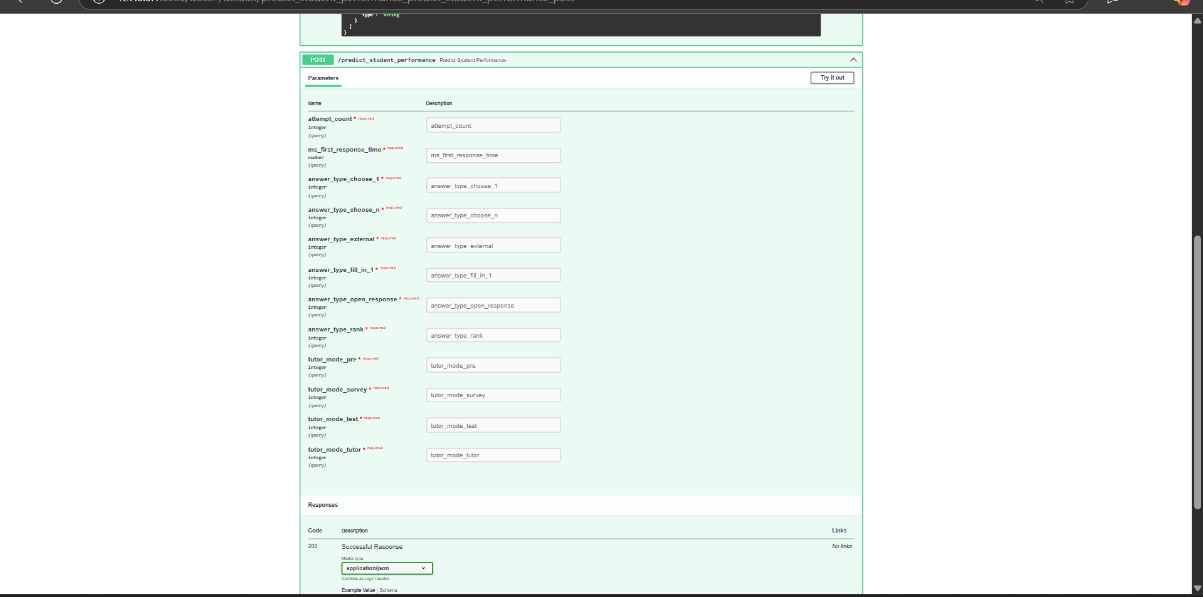
**Screenshot of FastAPI Response:**



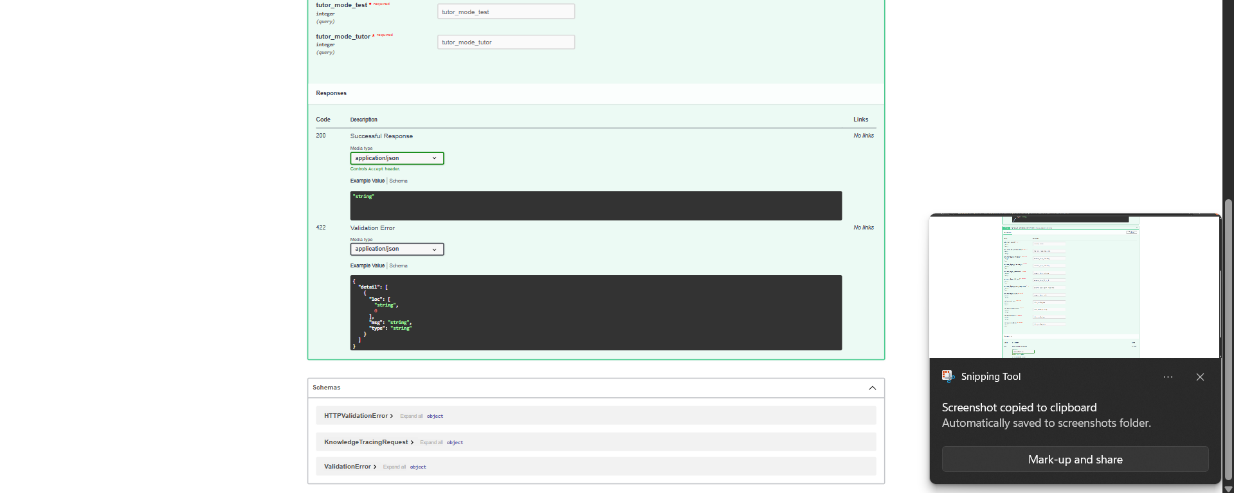
**Screenshot of FastAPI Parameters:**



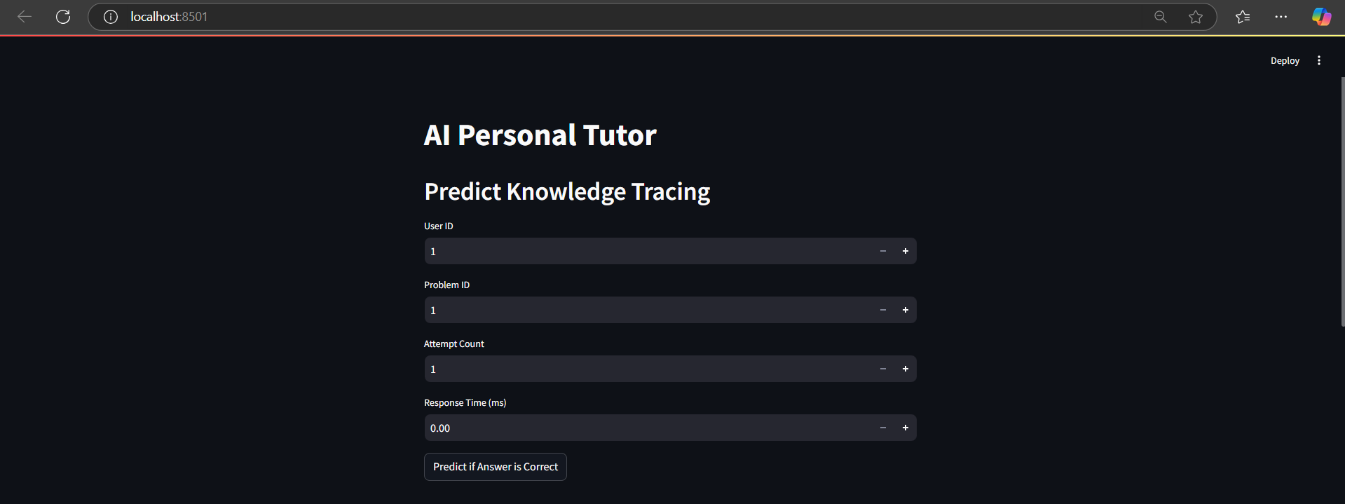
**Screenshot of FastAPI Request Body:**



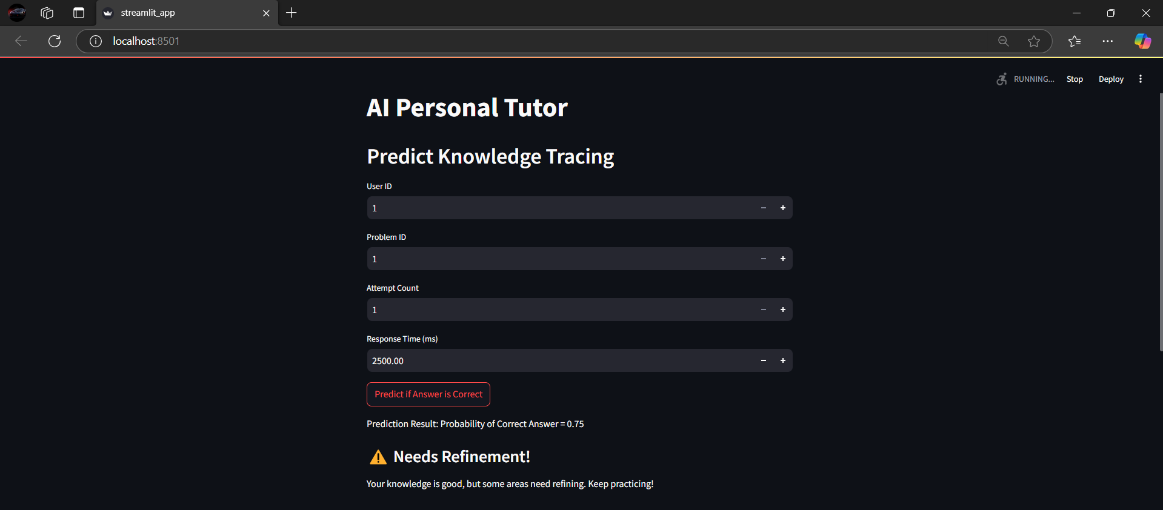
**Screenshot of FastAPI Error Response:**



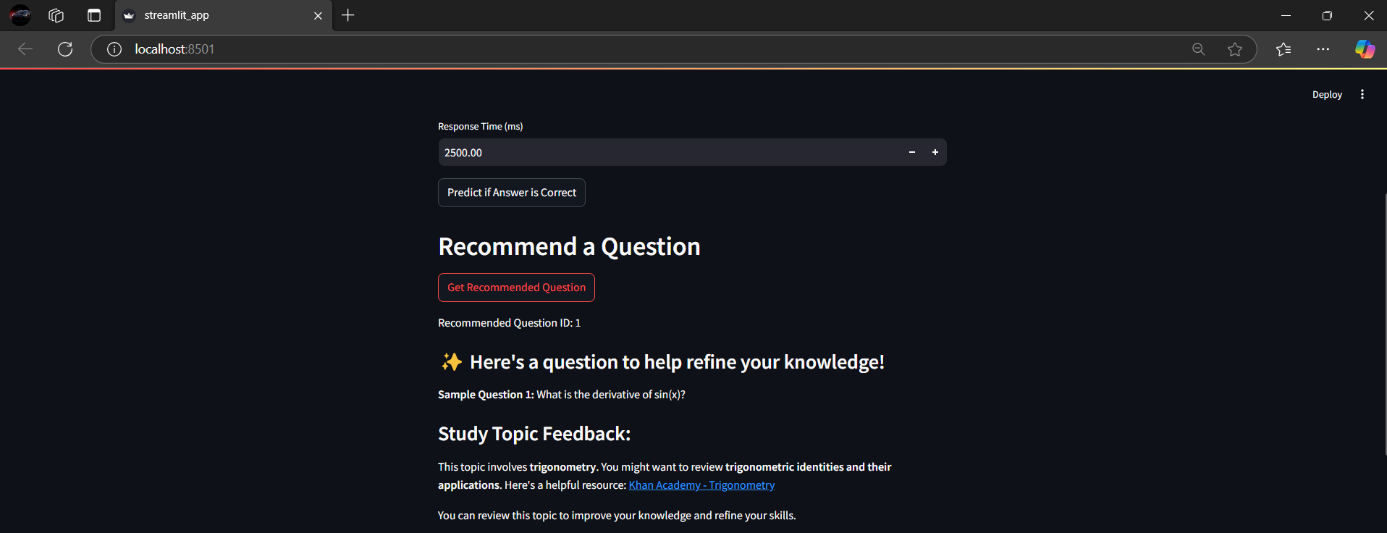
**Screenshot of Streamlit Interface:**



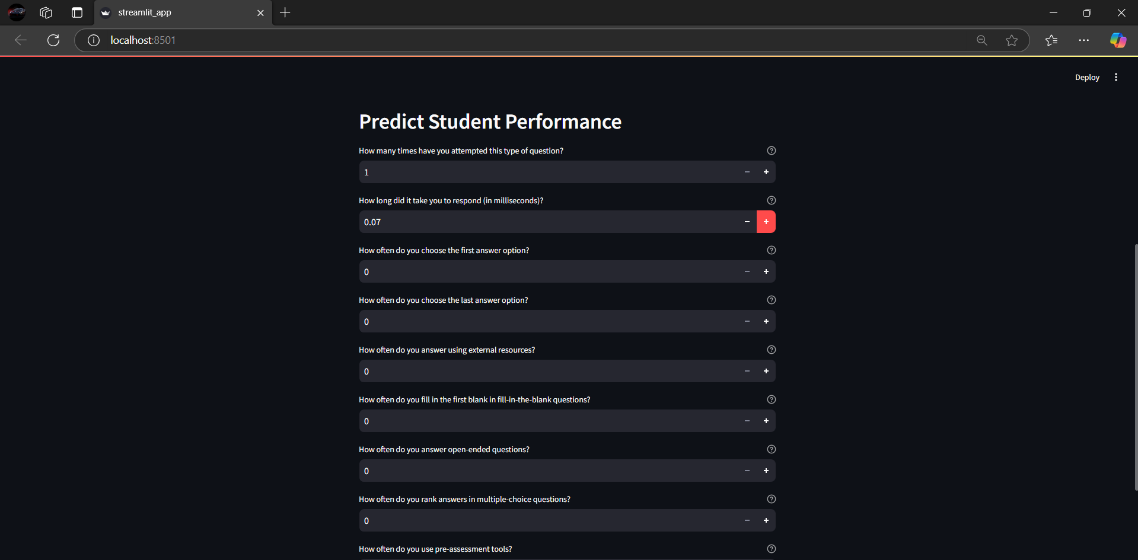
**Screenshot of Streamlit Interface 2:**



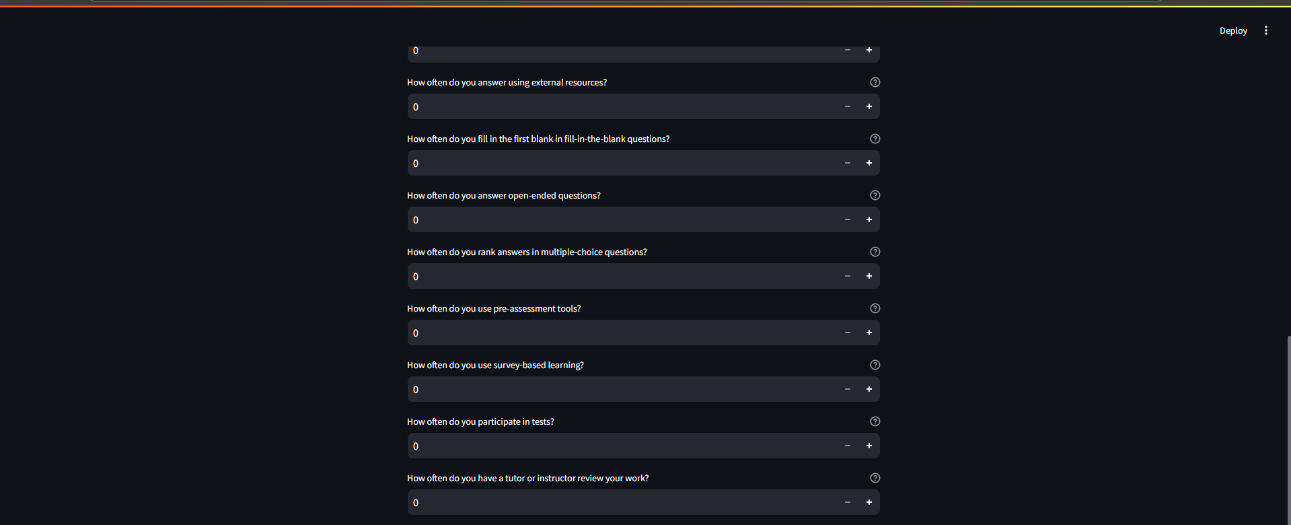
**Screenshot of Streamlit Interface 3:**



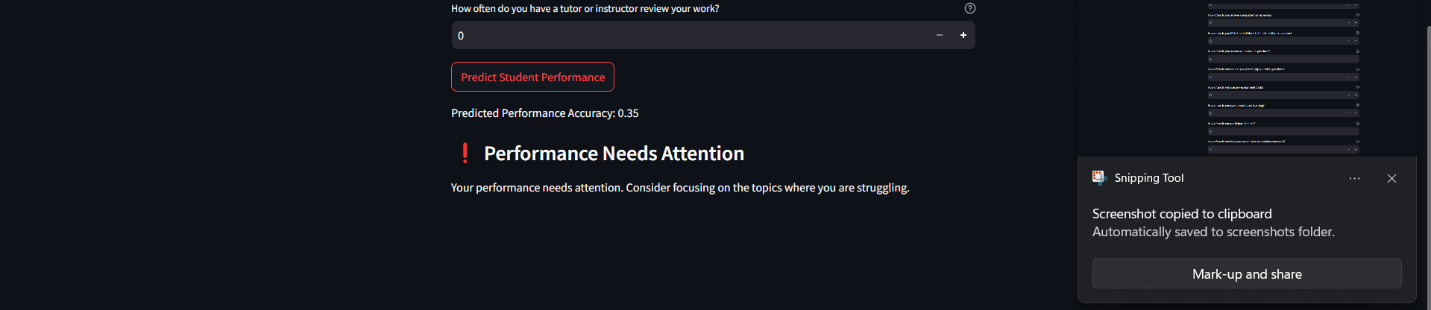
**Screenshot of Streamlit Interface 4:**



**Screenshot of Streamlit Interface 5:**



**Screenshot of Streamlit Interface 6:**



**5. Dataset**

Name: Assistments 2009–2010 Dataset.

Size: Approximately 1 million rows.

Key Features:

user\_id: Student ID.

problem\_id: Question or problem ID.

correct: Student's right answer (1 correct, 0 incorrect).

attempt\_count: Number of attempts made by the student.

ms\_first\_response\_time: First response time to the question.

tutor\_mode: Tutor mode utilized (e.g., feedback or hints).

answer\_type: Type of provided answer (e.g., written or multiple choice).

The target variable of the models is the correct field (1 or 0).

**Data Preprocessing**

Handling Missing Values: Missing values are replaced with the median or mode of the concerned column.

Normalization: Quantitative attributes like response time are normalized to remove variation from the dataset.

One-Hot Encoding: Feature categories like answer type are encoded as numeric values.

**2. Name: ETD4 Dataset (EdNet KT4)**

Data used in this stage is the ETD4 Dataset, processed from the EdNet KT4 ZIP file. The dataset is a large dataset that contains student interaction logs and student engagement data. The dataset features are applicable in predicting student behavior and performance:

COLUMNS:

timestamp: Student's action timestamp.

action\_type: Action type performed by the student (e.g., 'enter', 'submit', 'respond').

item\_id: ID of the item on which the student interacted.

cursor\_time: Action duration.

source: Source of the action (e.g., 'diagnosis', 'review').

user\_answer: Student answer.

platform: Platform on which student interacted (e.g., 'web').

Preprocessing Steps:

Handling Missing Values: Forward filling (fillna(method='ffill')) is applied to fill missing values, particularly in cursor\_time, source, and platform columns.

Feature Engineering: Time-based features are derived from the timestamp to learn more about the student interaction pattern. Values of columns such as action\_type, source, and platform are one-hot encoded to obtain numerical features.

Scaling: MinMaxScaler is used for scaling continuous features such as cursor\_time for optimal model performance.

Train-Test Split: The data is divided, with 80% used for training and 20% reserved for testing.

**6. Model 1: LSTM for Knowledge Tracing**

Purpose: LSTM (Long Short-Term Memory) is used to predict the likelihood of a student answering a question correctly based on previous performance.

Architecture:

Input: Sequence of previous answers, response time, and question difficulty.

Layer Structure: Dense layers followed by LSTM layers.

Output: Binary output (1 for correct, 0 for incorrect).

Training:

Epochs: 10

Batch Size: 64

Accuracy: ~80.6%

Loss: 0.42

Key Takeaways: LSTM excels in modeling sequential learning data as it captures long-term dependencies in time-series data.

**7. Model 2: SVD for Personalized Question Recommendations**

Purpose: SVD collaborative filtering is used to recommend personalized questions based on the student's past answers.

Technique:

Collaborative Filtering: SVD factorizes the user-item interaction matrix to generate recommendations.

Evaluation: RMSE of 0.409 on the test set indicates that the model provides effective recommendations based on past performance.

**8. Model 3: XGBoost for Predicting Student Performance**

Purpose: XGBoost forecasts future student performance based on historical behavior.

Features:

Response Time

Tutor Mode

Attempt Type

Number of Attempts

Results: RMSE of 0.358, reflecting high predictive accuracy.

**9. LSTM for Knowledge Tracing (Enhanced with ETD4 Data)**

Purpose: The LSTM model is trained to recognize patterns of knowledge and predict a learner's potential to answer questions correctly based on past interactions.

Architecture:

Input: Student interactions (timestamp, action type, platform, etc.), with normalized features like cursor\_time.

Output: Binary prediction (incorrect = 0, correct = 1).

The data from the ETD4 dataset is structured to fit the LSTM model setup (samples, time steps, features).

**10. Q-Learning for Difficulty Control**

Purpose: This layer uses Reinforcement Learning (RL) to adjust the difficulty of questions based on student responses.

States: Student-question pairs.

Rewards: +1 for a correct response, -1 for an incorrect one.

Actions: Selection of the next question to present.

Outcome: Personalizes learning by dynamically adjusting the difficulty level based on the developing capabilities of the learner.

**11. Results and Evaluation**

System: Highly accurate in sequence prediction tasks.

SVD: Enhances question recommendations and personalization.

XGBoost: Predicts student performance with exceptional accuracy based on prior interactions.

Advantages:

Scalable and modular.

Personalized learning paths.

Ability to provide immediate feedback and adapt based on student performance.

Model Performance:

Accuracy: Approximately 80.6% on the ETD4 dataset, demonstrating strong sequential prediction capabilities.

Loss: 0.42, indicating room for improvement.

**12. Conclusion**

The AI Personal Tutor leverages advanced machine learning models to provide adaptive, personalized learning experiences. Through knowledge tracing, performance prediction, and question recommendation models, it addresses the main challenges of traditional education with real-time, data-driven decision-making.

**13. Future Work**

NLP Support: Integration of Natural Language Processing for enhanced response capabilities.

Audio/Video Feedback: Could further engage users and enhance interaction.

Skill-based Curriculum Mapping: Directly mapping the curriculum by skill set or learning targets could further enhance personalization

**References**

1. T.-X. Cao Thi, "Singular value decomposition and applications in data processing and artificial intelligence," HPU2 Journal of Sciences: Natural Sciences and Technology, vol. 02, no. 03, pp. 34-41, 2023. [Online]. Available: https://doi.org/10.56764/hpu2.jos.2023.2.3.34-41&#8203;:contentReference[oaicite:0]{index=0}.
2. Y. Zhou et al., "Analysis on the Sustainability of a Controlled Ecological Life Support System (CELSS) Based on the Energy Flow Model," Applied Sciences, vol. 14, no. 07279, 2024. [Online].Available: https://doi.org/10.3390/applsci1407279&#8203;:contentReference[oaicite:1]{index=1}.
3. W. Su et al., "An XGBoost-Based Knowledge Tracing Model," International Journal of Computational Intelligence Systems, vol. 16, no. 13, 2023. [Online]. Available: <https://doi.org/10.1007/s44196-023-00192-y&#8203;:contentReference[oaicite:2]{index=2}>.
4. H. Chen et al., "Advances and perspectives on applications of 3D printing in the medical field," Science and Technology, 2023. [Online]. Available: https://doi.org/10.3390/scitech1020034&#8203;:contentReference[oaicite:3]{index=3}.