Project presentation

Presented by Team 3 - Group 5

01

Tackled problem.

- Misuse of AI tools for pedagogical purposes.
- Students struggling with organizing their studies.
- Lack of personalization for students using AI tools.

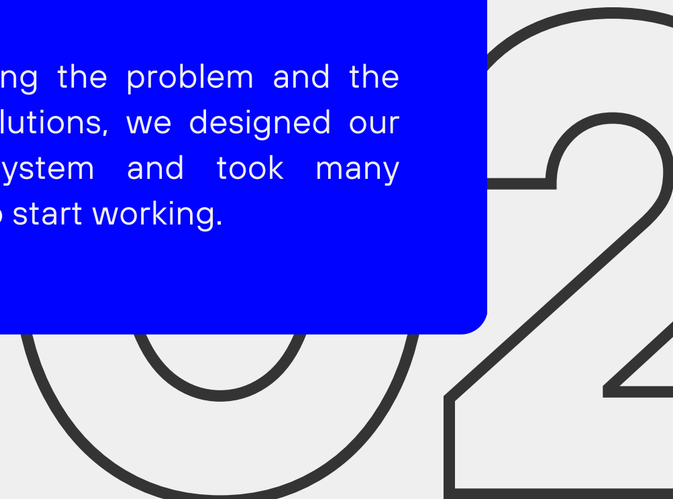
Design thinking

Running to implementation and technical side is a common mistake computer science students do when creating a project. A mistake that we made sure to avoid.

01. Understanding our potential customers: Algerian computer science students who are not gaining the necessary knowledge through LLMs.

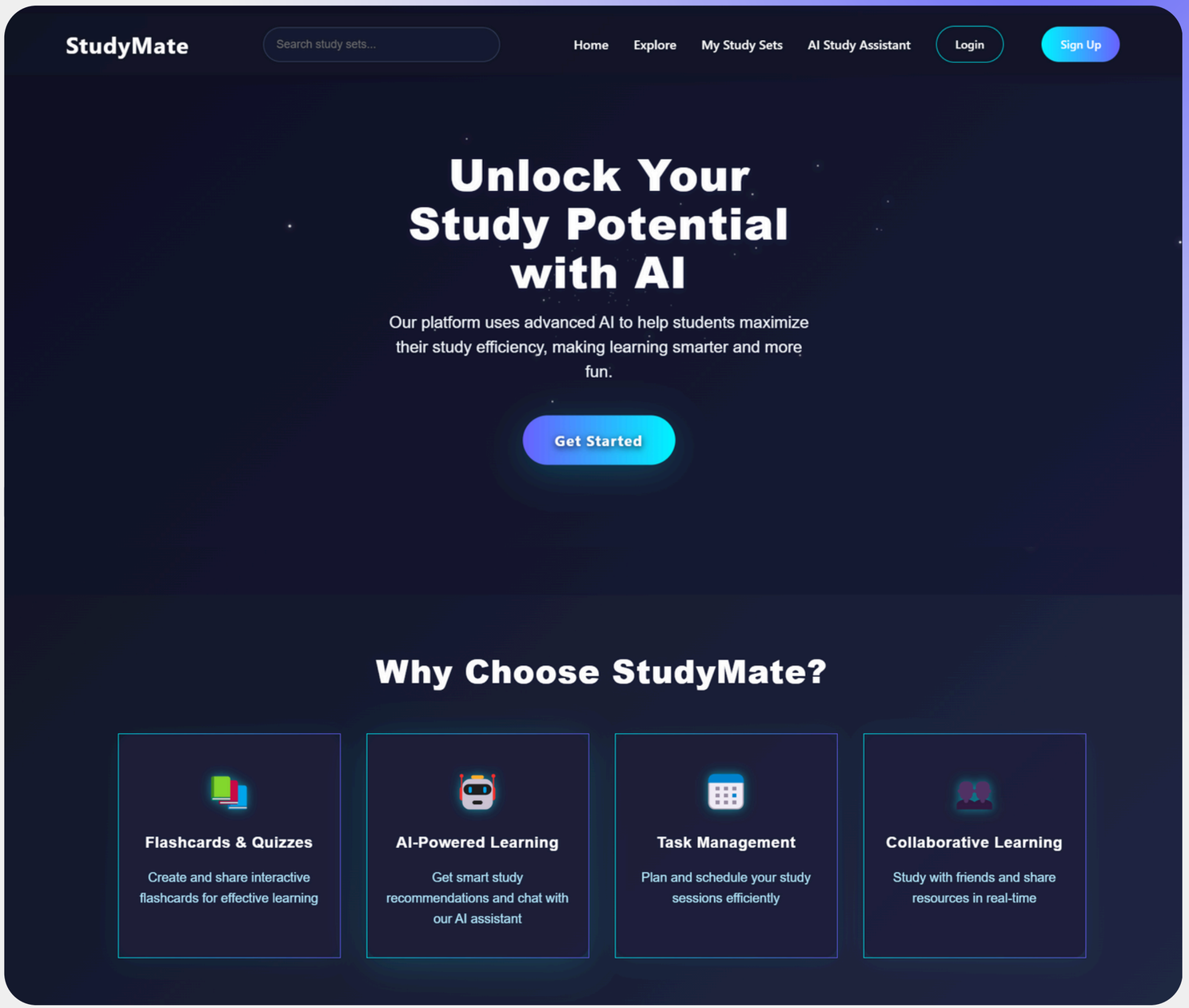
02. Understanding the problem and ideation process: We wanted to create a personalized solution for each student in each module.

03. After defining the problem and the possible solutions, we designed our software system and took many decisions to start working.

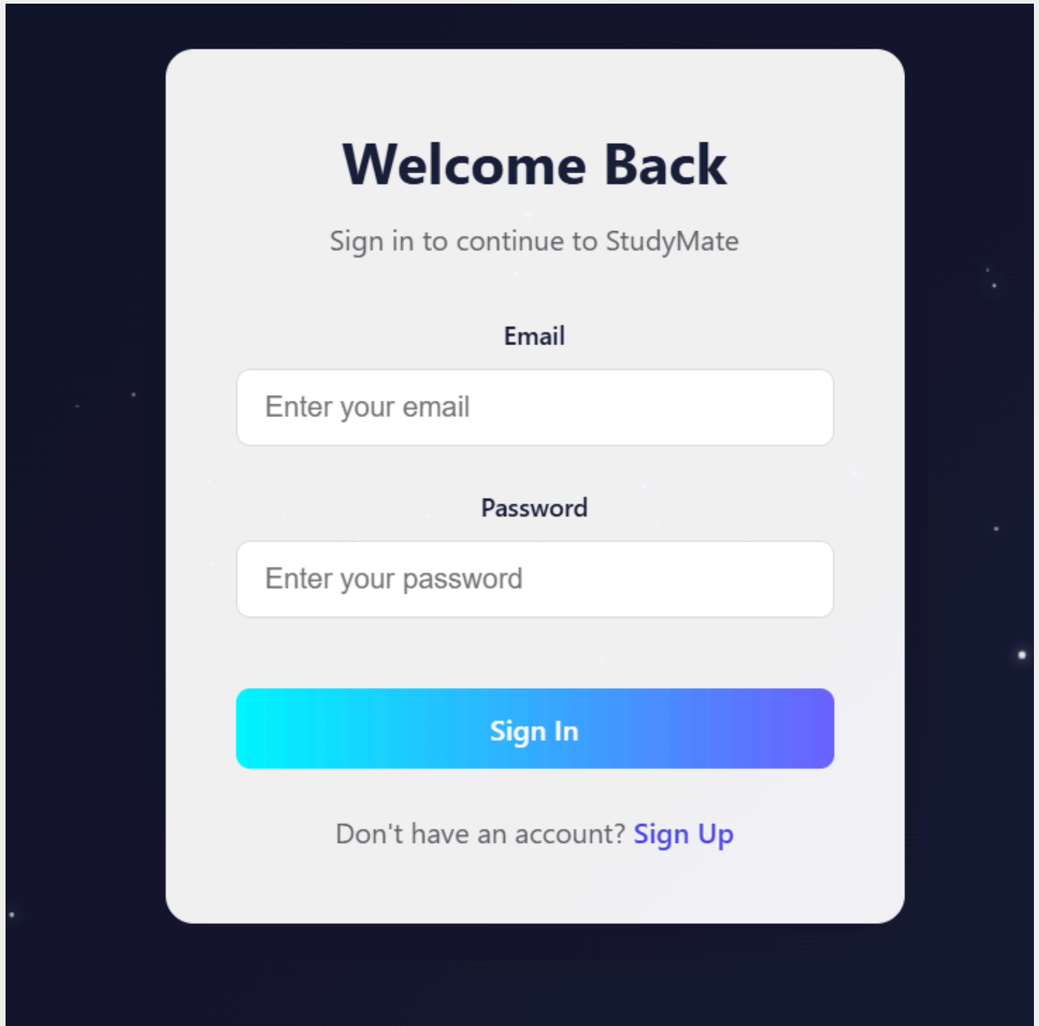


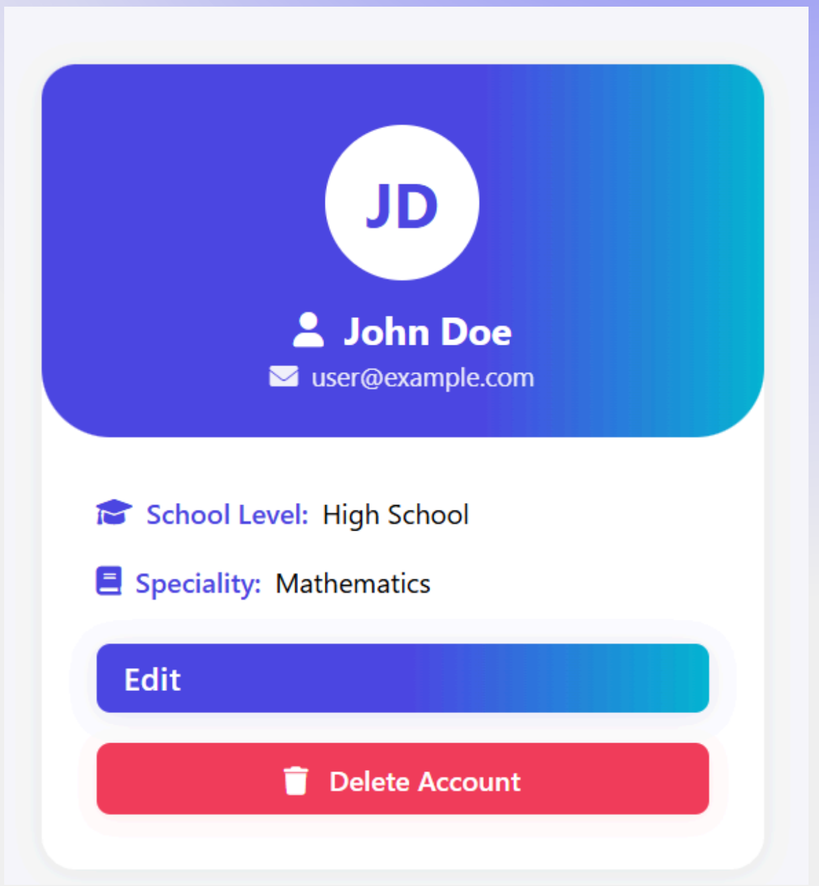
Study Mate

Our target is students juggling multiple courses, lectures, and deadlines. Their goal is : stay organized and study smarter .



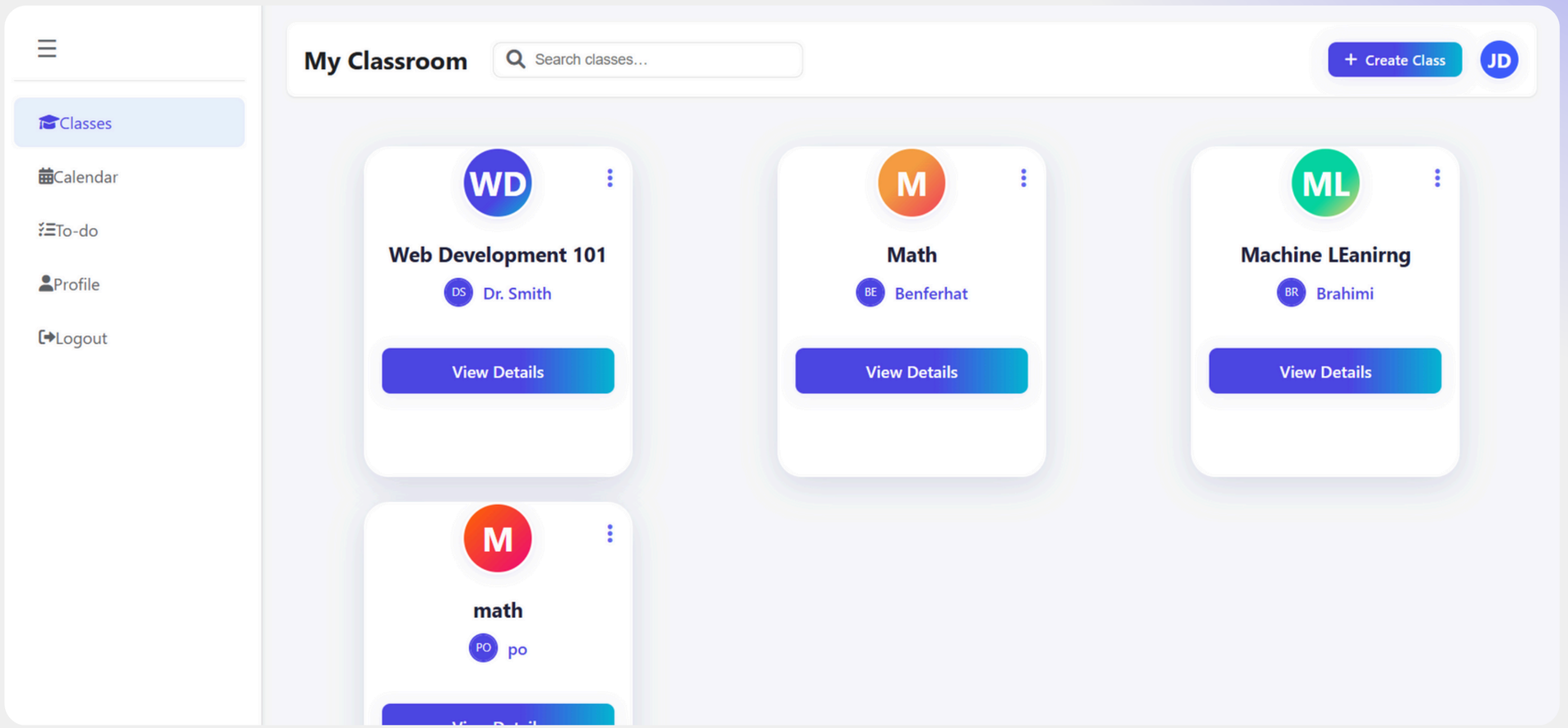
User Accounts:

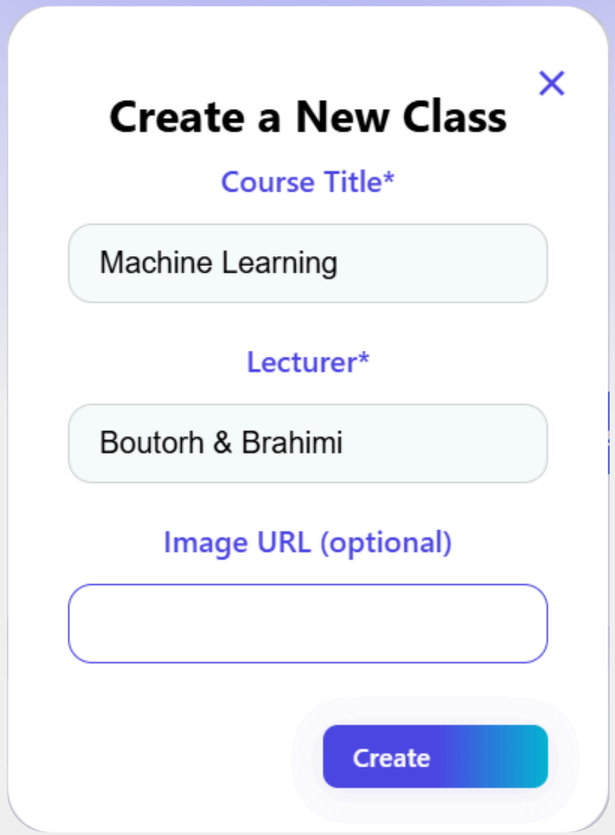




My Classroom

With a clear view of their upcoming deadlines, the next step is diving into the Classroom . Here, the students can join existing courses or add new ones. Each class acts like a personal workspace — they can upload multiple lectures and use AI-powered tools to get everything they need all in one place.



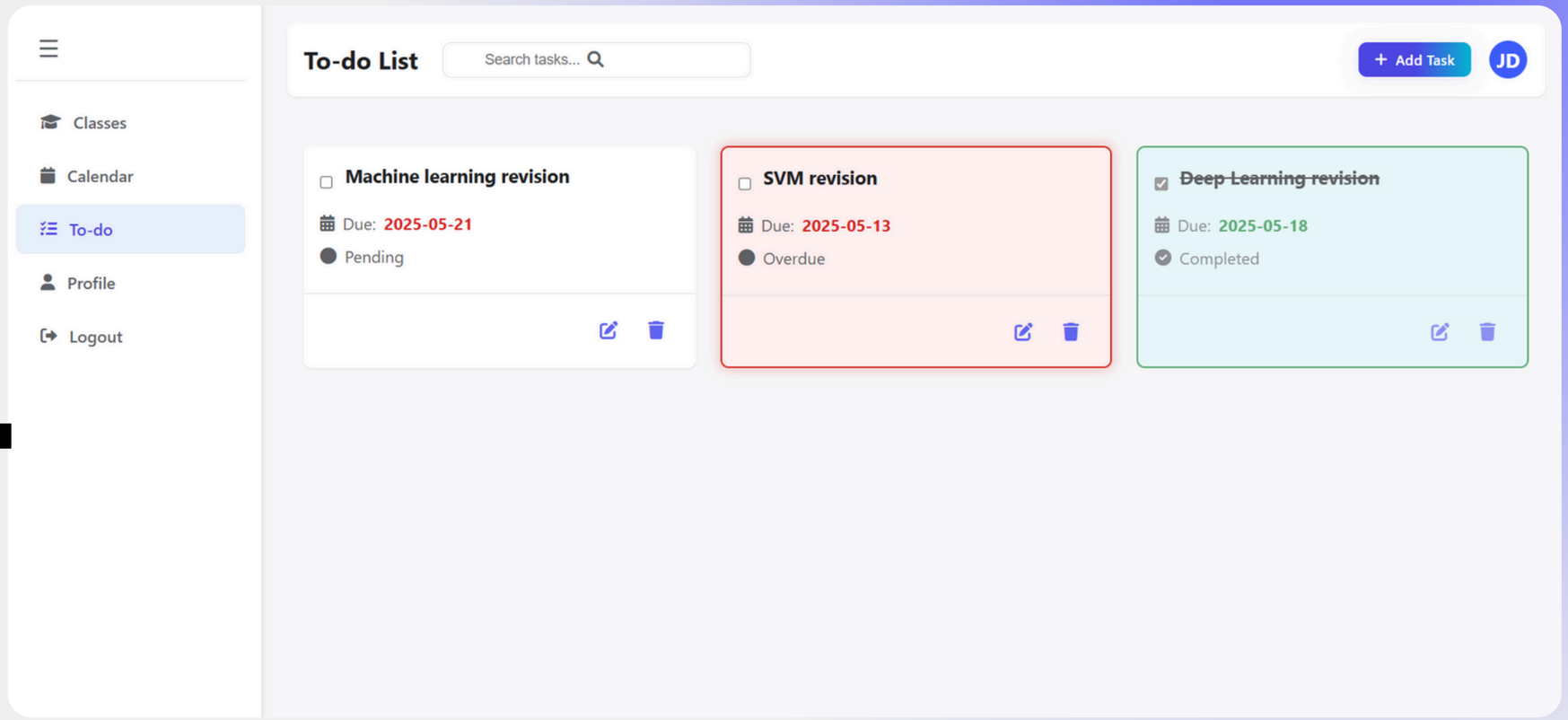


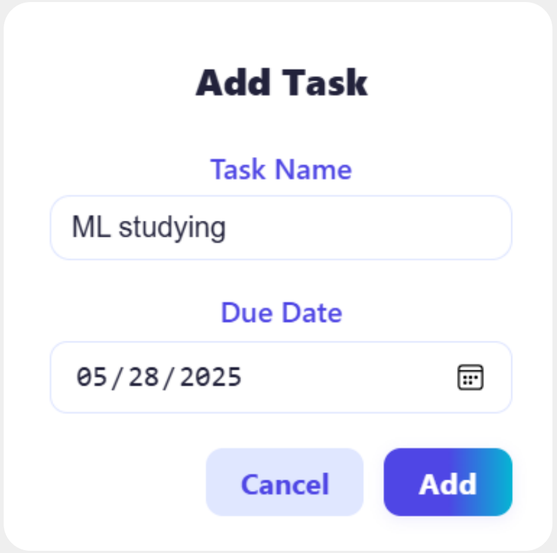
To-Do List & calendar

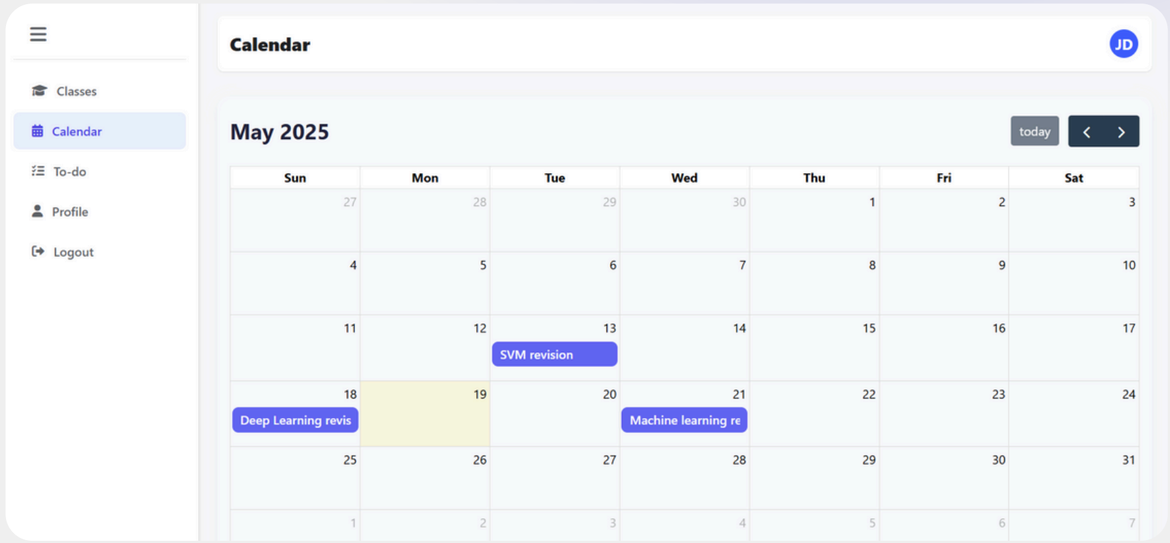
To support that, we designed a To-Do List where the students can easily create tasks.

Each task is automatically synced with the Calendar, giving them a clear overview of what needs to be done and when.

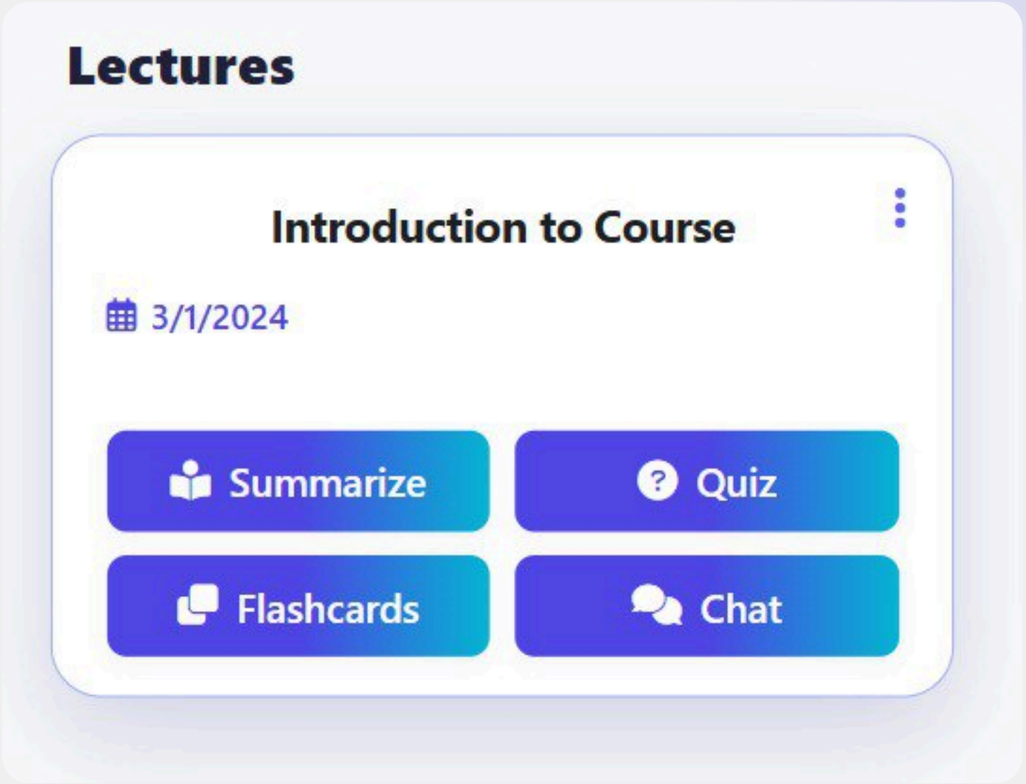
This integration helps them avoid overlapping deadlines and stay on top of their schedule."



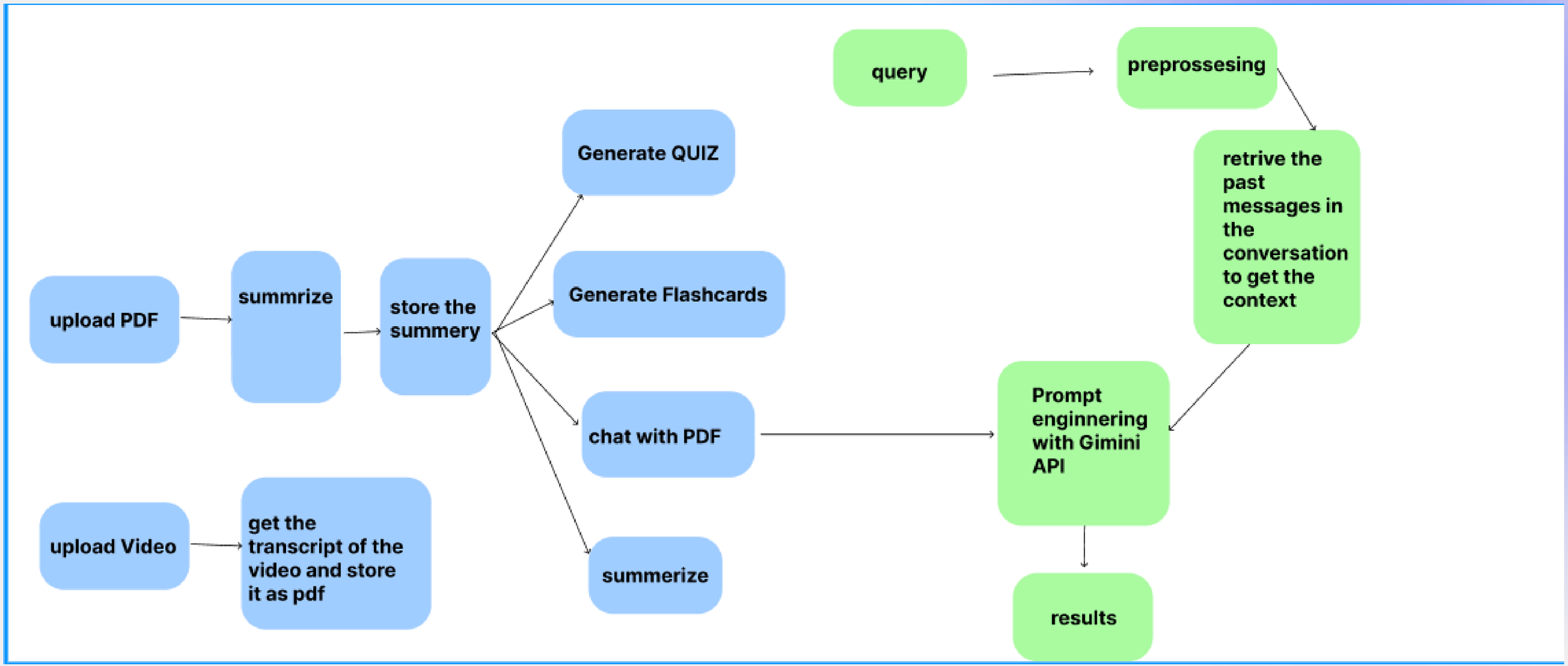




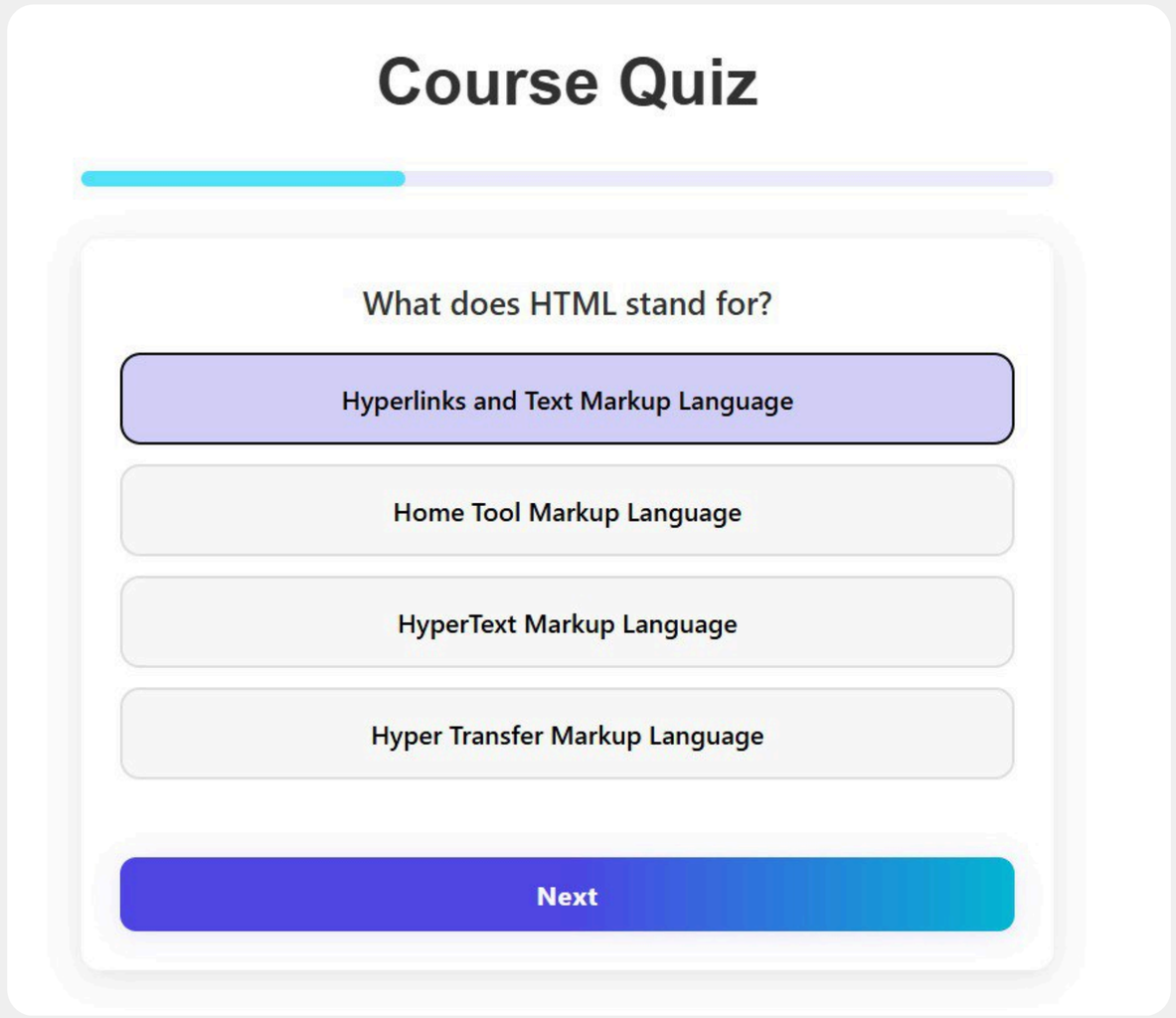
UPLOAD THE LECTURE PDF

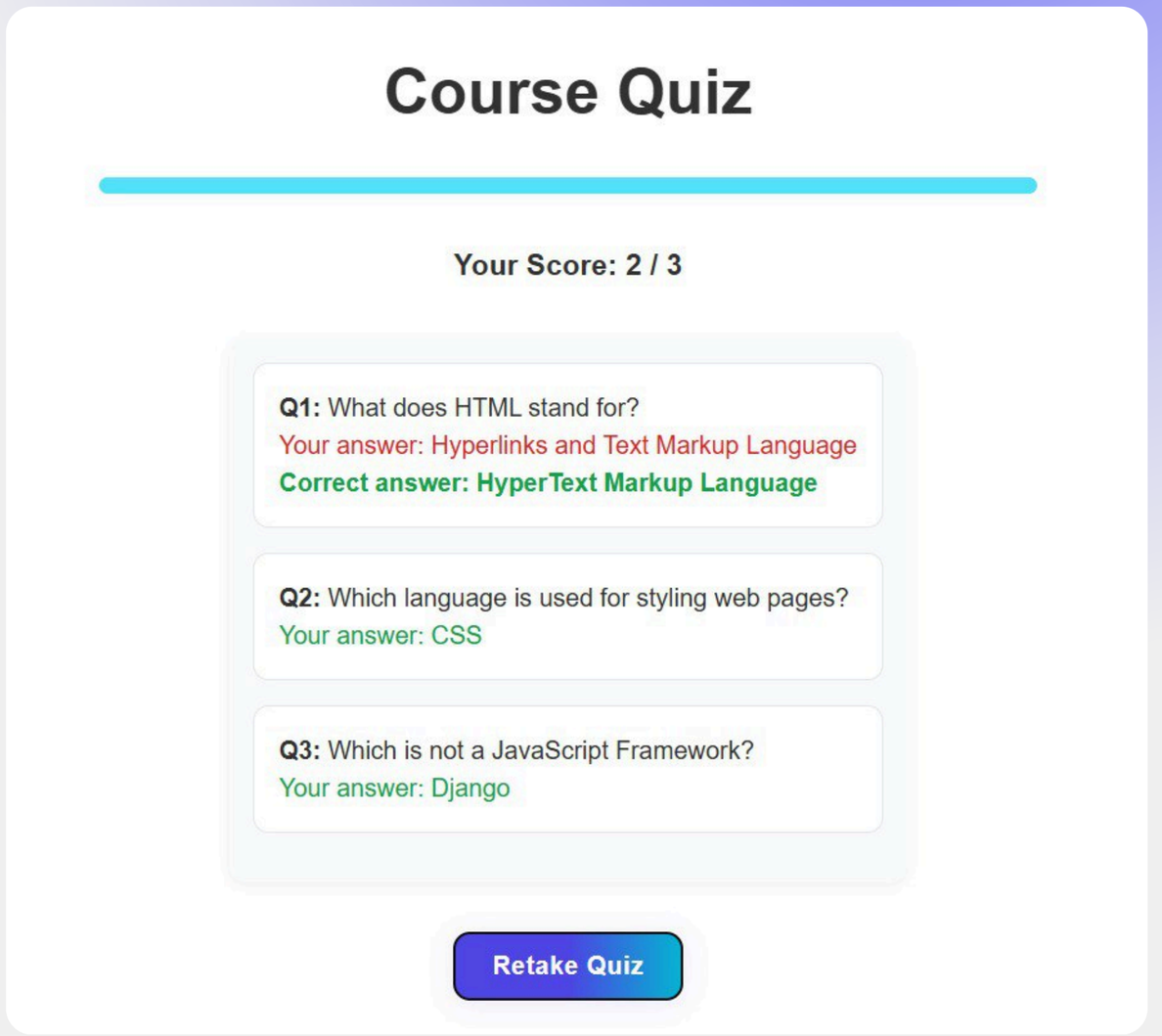


Flowchart diagram

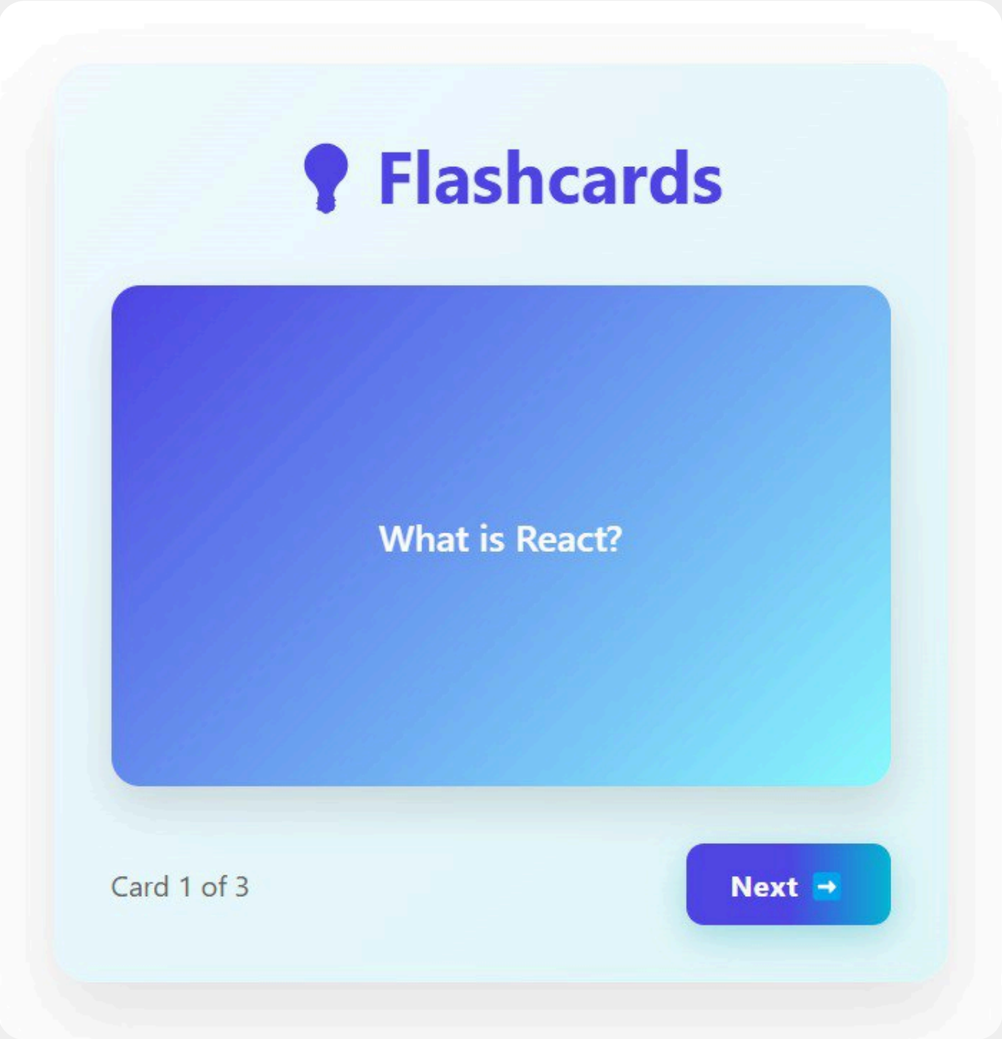

query
preprossesing
retrive the past messages in the conversation to get the context
Generate QUIZ
upload PDF
summrize
store the summery
Generate Flashcards
chat with PDF
Prompt enginnering with Gimini API
upload Video
get the transcript of the video and store it as pdf
summerize
results

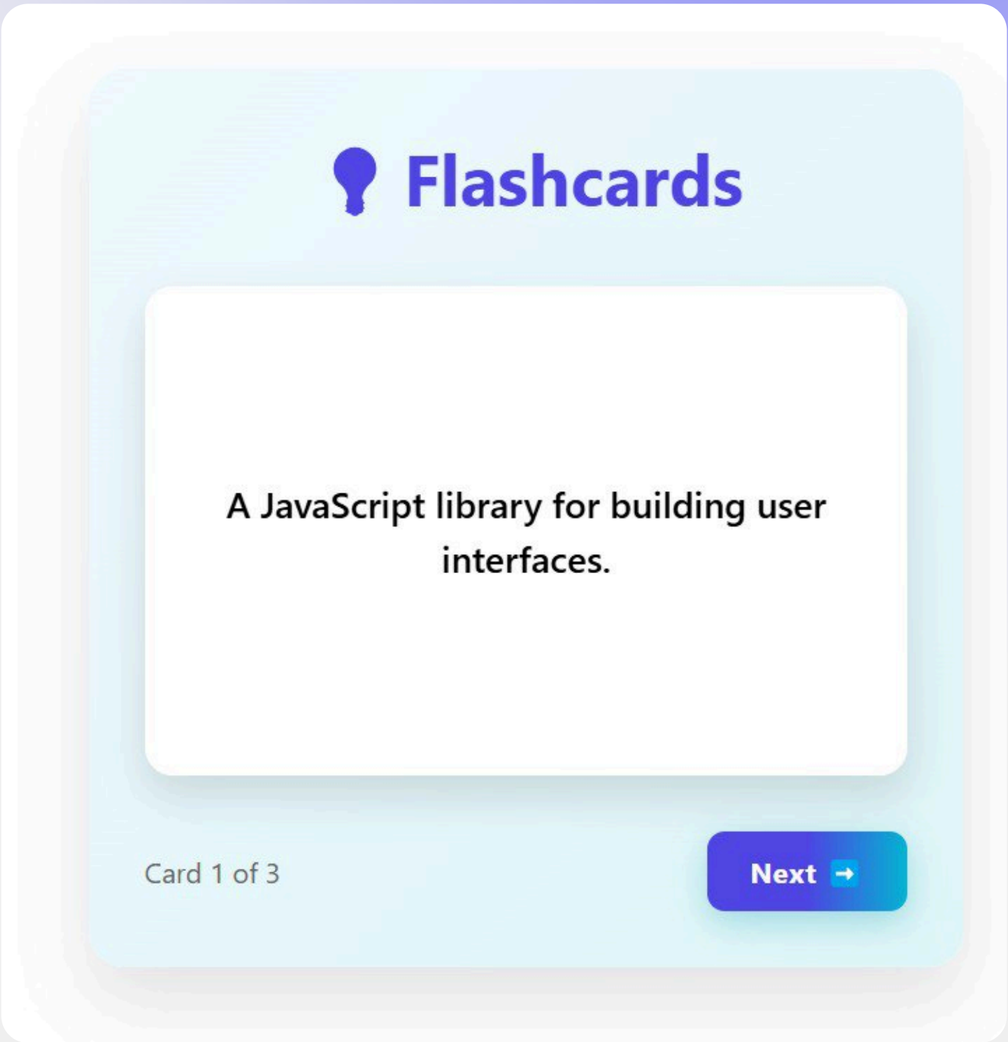
GENERATE QUIZ :



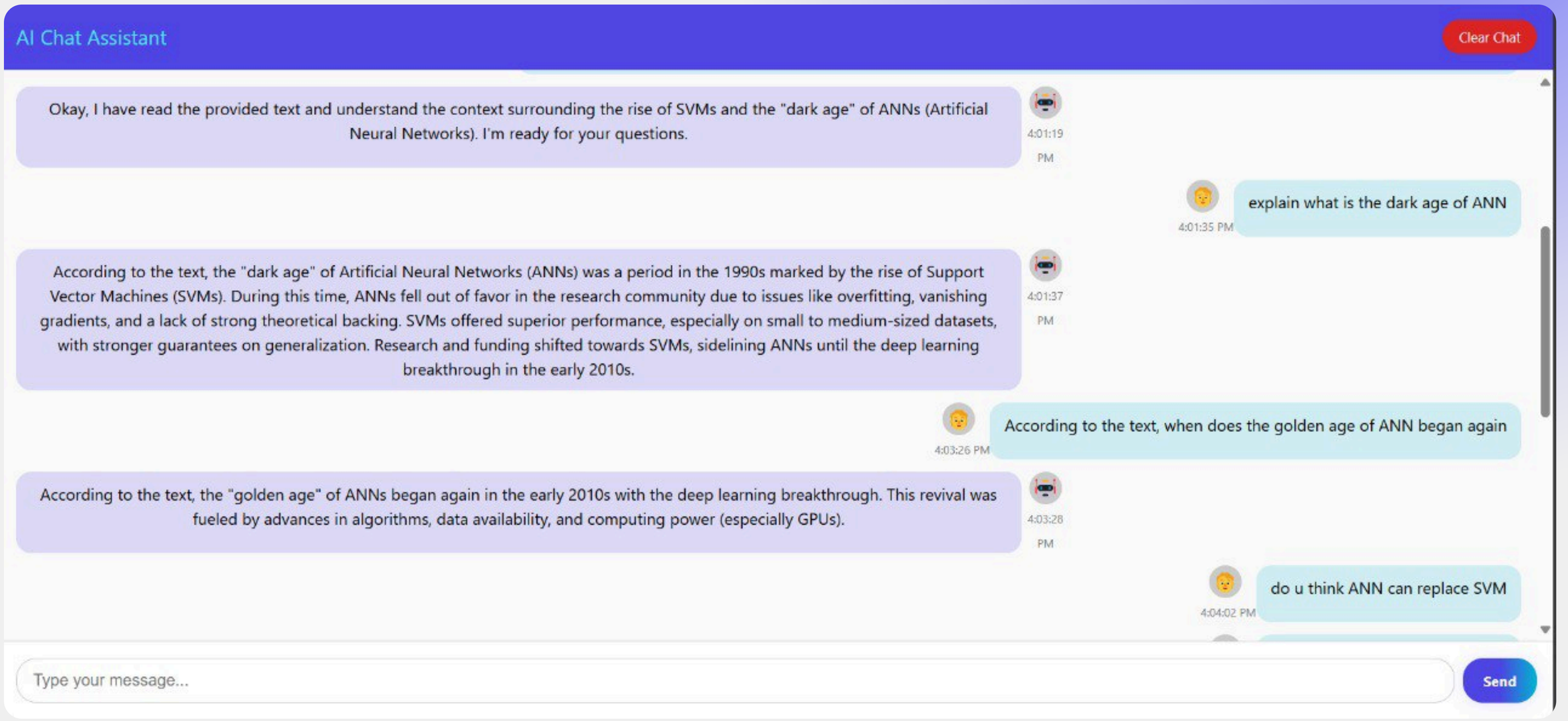


GENERATE FLASHCARDS :





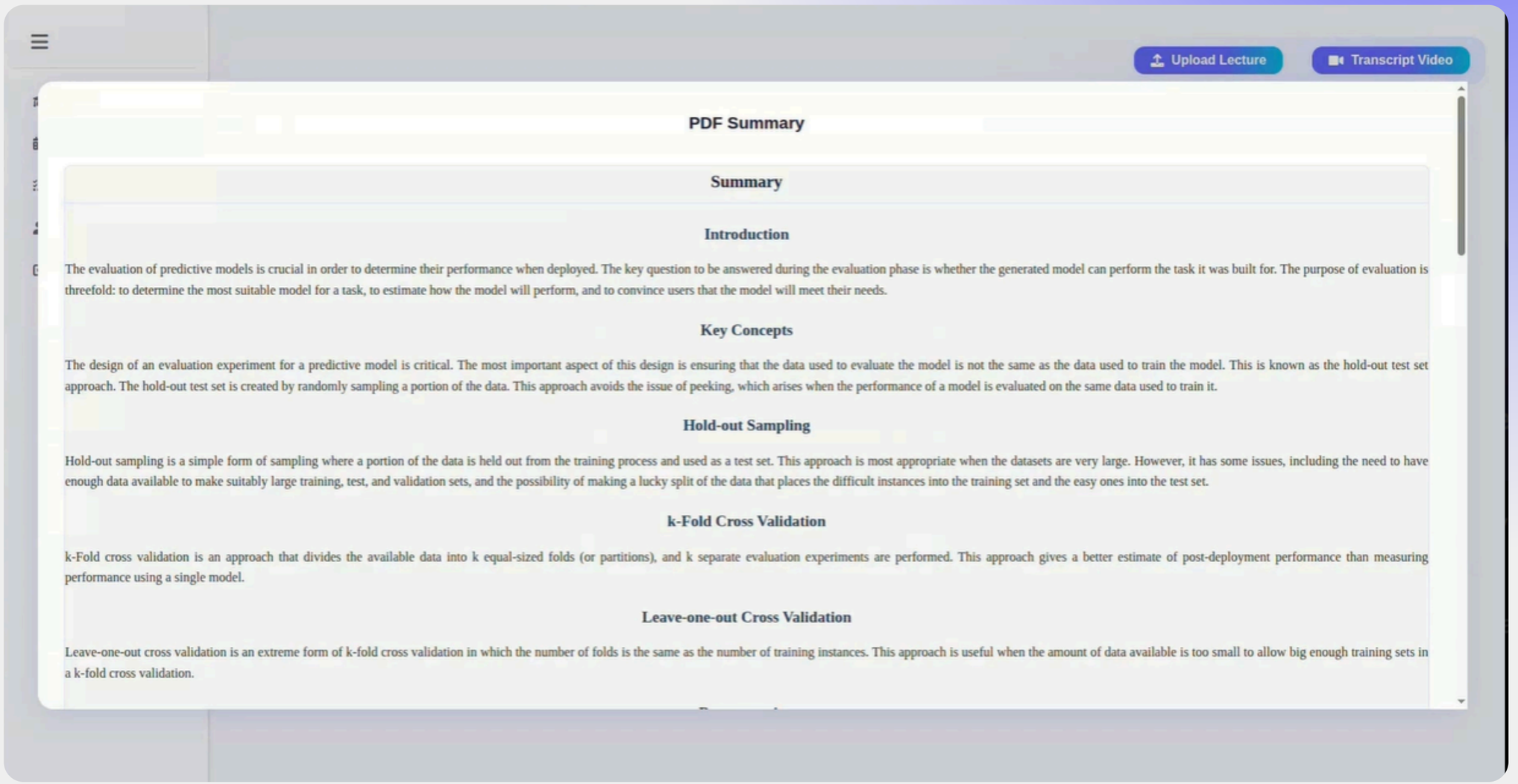
CHAT WITH PDF :



summarization feature

Overview & Purpose

Summary:


Upload Lecture
Transcript Video
PDF Summary
Summary
Introduction
The evaluation of predictive models is crucial in order to determine their performance when deployed. The key question to be answered during the evaluation phase is whether the generated model can perform the task it was built for. The purpose of evaluation is threefold: to determine the most suitable model for a task, to estimate how the model will perform, and to convince users that the model will meet their needs.
Key Concepts
The design of an evaluation experiment for a predictive model is critical. The most important aspect of this design is ensuring that the data used to evaluate the model is not the same as the data used to train the model. This is known as the hold-out test set approach. The hold-out test set is created by randomly sampling a portion of the data. This approach avoids the issue of peeking, which arises when the performance of a model is evaluated on the same data used to train it.
Hold-out Sampling
Hold-out sampling is a simple form of sampling where a portion of the data is held out from the training process and used as a test set. This approach is most appropriate when the datasets are very large. However, it has some issues, including the need to have enough data available to make suitably large training, test, and validation sets, and the possibility of making a lucky split of the data that places the difficult instances into the training set and the easy ones into the test set.
k-Fold Cross Validation
k-Fold cross validation is an approach that divides the available data into k equal-sized folds (or partitions), and k separate evaluation experiments are performed. This approach gives a better estimate of post-deployment performance than measuring performance using a single model.
Leave-one-out Cross Validation
Leave-one-out cross validation is an extreme form of k-fold cross validation in which the number of folds is the same as the number of training instances. This approach is useful when the amount of data available is too small to allow big enough training sets in a k-fold cross validation.

Why We Used APIs Instead of Local Models

Local Model Issues:

- Limited to 7B models (Mistral, LLaMA 2)
- Poor summary quality
- High memory/GPU requirements
- Poor scalability for real-time use

Model Evaluation – How We Chose the LLMs

- LLaMA
  - ROUGE-1: 0.64
  - ROUGE-2: 0.47 (slightly lower than DeepSeek)
  - BERTScore: 0.68
- DeepSeek
  - ROUGE-1: 0.66 (higher than LLaMA)
  - ROUGE-2: 0.50
  - BERTScore: 0.69

Why LLaMA 3.3 70B?

- Better ROUGE-1 (most relevant for extractive tasks)
- Supports 128K token input + 32K output
- Produced most coherent and detailed academic summaries

Video transcription

Video Transcript

Upload Summary PDF

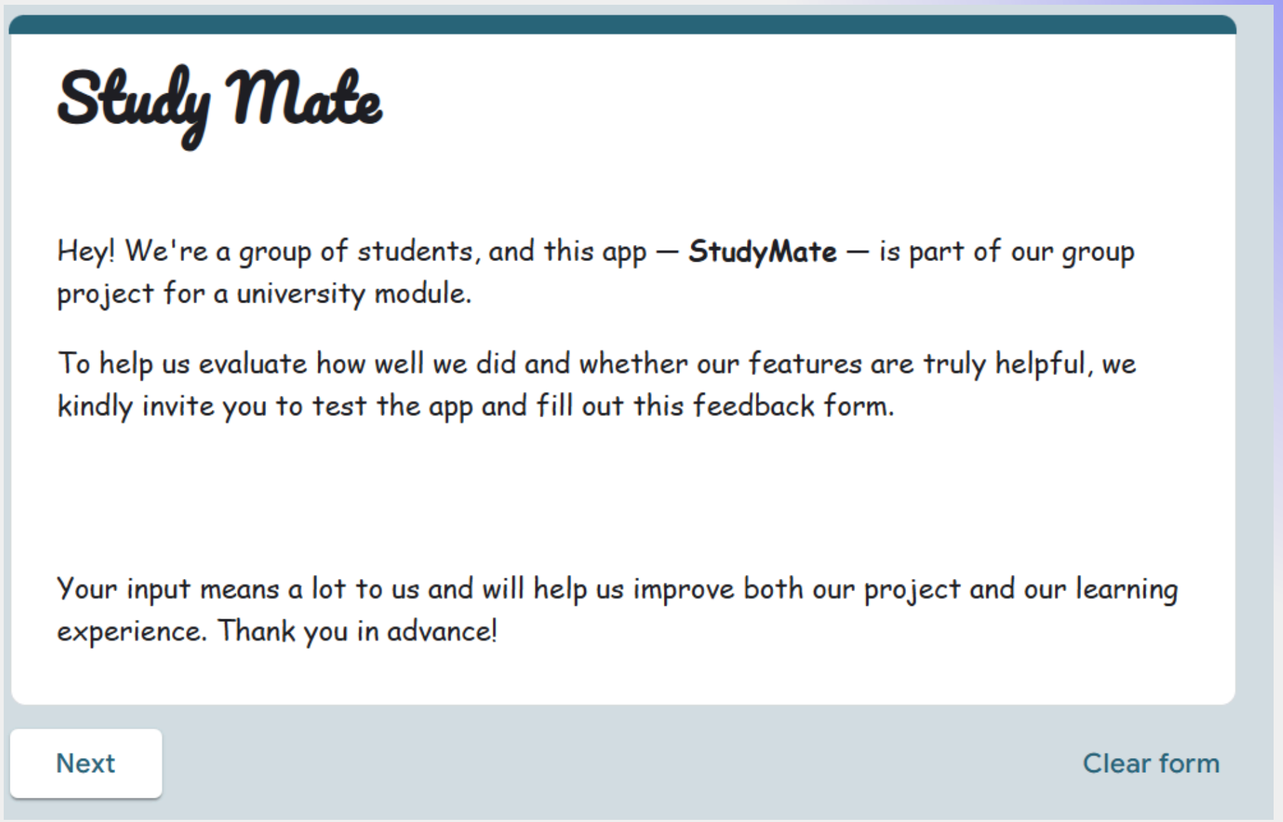
Summary – How to Evaluate Your ML Models Effectively? | Evaluation Metrics in Machine Learning!

How do you evaluate your ML models? Evaluation is a critical step in the model development process. It ensures that our model is good enough to perform well on unseen data. Before diving into evaluation metrics, remember that the data should be divided into train, test, and often validation sets. More on this here. Let's call one class positive and the other one negative. We can arrange the predictions in four possible ways. Either we predict the positive class correctly, or we predict the negative class incorrectly. And either we predict correctly the negative class or incorrectly the positive one. This is called the confusion matrix, a popular metric used in classification tasks. From the confusion matrix, we can extract other metrics like accuracy. Accuracy measures how often a model predicts correctly out of all the predictions it made. If we translate that to a formula, we will need to divide the total number of correct predictions by the total number of predictions. Let's say if a person has a certain flu, we predict positive, if not then negative. The flu is very rare and of 100,000 people, only 100 have it. Our model learned to classify all observations to the negative class, meaning none of them has the flu. When we calculate the accuracy score, we get a very high score. However, our model is useless. Imagine telling a person that he has no flu when he actually has. It is generally better in such cases to classify the person having the flu when he doesn't, than to miss a case where someone does have the flu, but the model says they don't. Two more formulas appear here, recall and precision. Their difference is in the denominator. In our problem terms, precision has false positives in the denominator, meaning a high precision value will mean the model does not predict flu when the person is healthy. Recall on the other hand, targets false negatives, and in the case of a very high recall, the model identifies everyone who has the flu and does not miss anyone who is sick. There is another measure that tries to maximize both recall and precision called F1 score. It takes the harmonic mean of precision and recall, meaning that you can get a high F1 score in the case you have high precision in recall. Thus, we use recall, precision, and F1 score for accuracy for balanced ones. For multi-class scenarios, the method is slightly different. We will refer to it later. There are also other important metrics, such as AUC and ROC curves. For regression and unsupervised tasks, the metrics are different. They are more complex ones, and we will talk about all those in the upcoming videos. So, stay with us. If you want to learn more about artificial intelligence, subscribe to our channel to be aware of the new videos. Press the like button and let's discuss AI in the comments section.

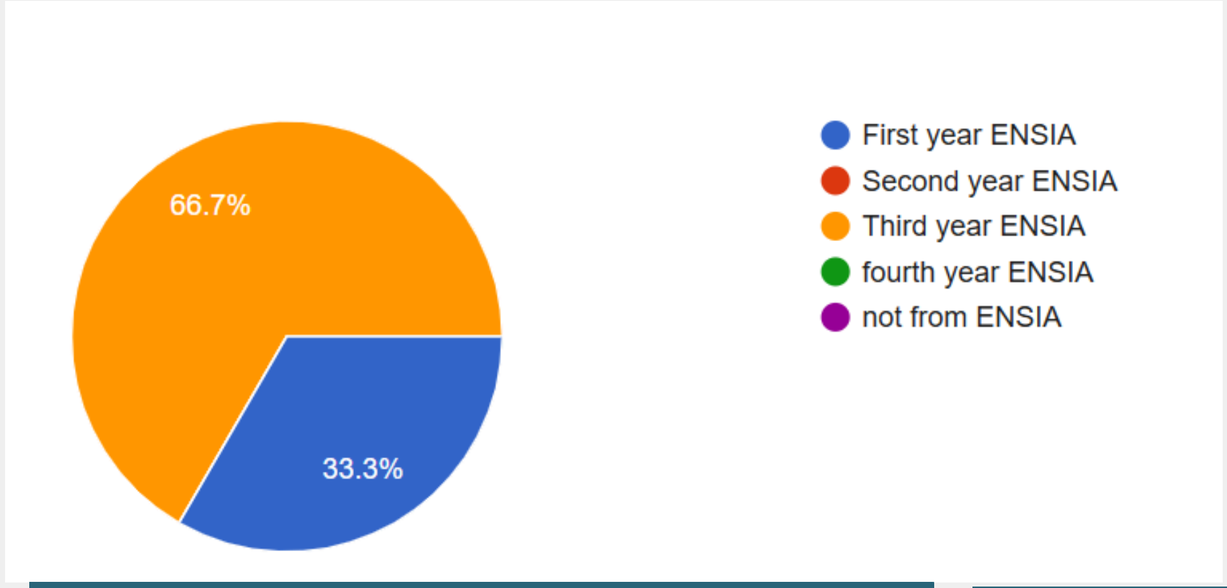
Duration: 3.0 min

Results and Discussion:

- User testing was conducted with real users (ENSIA students).
- App features were actively used during the testing.
- Feedback form was collected afterward.
- Overall feedback was positive — users enjoyed the experience.
- Users appreciated the all-in-one platform with multiple features.
- Constructive feedback was received for future improvements.



some statistics from test users



Video Transcript Feature

Did the transcript accurately reflect the content of the video?

Copy chart

3 responses

Average rating (4.00)

1 2 3 4 5

To-Do List & Calendar Feature

Was it easy to add, edit, and delete tasks or events?

Copy chart

3 responses

Average rating (4.33)

1 2 3 4 5

Quiz Quality

Did the quiz help reinforce the key concepts from the lecture?

Copy chart

3 responses

Average rating (4.33)

1 2 3 4 5

Summary Quality

Was the lecture summary comprehensive?

Copy chart

3 responses

Average rating (4.00)

1 2 3 4 5

some feadbacks :

Overall Suggestions

Do you have any suggestions to improve the app overall?

3 responses

- add notes
- Add customized colors for lectures
- it was great i think

Final reflections

- Proud of the progress
- Looking ahead with openness

11

Thank You

http://localhost:3000/