(Declared as Deemed to be University under Sec.3 of the UGC Act, 1956)

MoE, UGC & AICTE Approved

NAAC A++ Accredited

#### SCHOOL OF COMPUTER SCIENCE AND TECHNOLOGY

#### **INTEL® UNNATI INDUSTRIAL TRAINING 2025**

# AI – POWERED PERSONALIZED TUTOR

### PROJECT REPORT

#### **SUBMITTED BY**

TRINA JOAN LYNUS [URK23CS1308]

JEFFY EDWINA CHRISTA J[URK23CS1233]

STENY BERNICE L R [URK23CS5063]

TEAM NAME : COGNIFRIEND

**INSTITUTE MENTOR:** Dr. M MANICKA RAJA



**APRIL 2025** 



#### SCHOOL OF COMPUTER SCIENCE AND TECHNOLOGY

#### **Problem Statement**

AI-Powered Personal Tutor: An adaptive, scalable approach that enhances learning and student engagement.

"One size fits all" approach is followed by traditional education, in turn ignoring the strengths, learning speeds, and weaknesses of the students. This leads to:

- Disengagement that is caused by the lack of personalized content
- When the students are overwhelmed or under-challenged, the traditional methods lead to ineffective learning.
- Preventing educators from optimizing the learning pathways and limitations in progress tracking.

#### **Abstract**

In the emergence of AI and impersonal mass education, "Cognifreind" emerges as a transformative and a powerful solution that obliterates the archaic "one-size-fits-all" academic model. It is designed as an adaptive, scalable digital mentor, the intelligent system dynamically tailors content according to the user's preference. This is built upon a robust architecture featuring a frontend that is responsive using HTML, CSS and JavaScript along with a Node.js-Express backend, MongoDB for data persistence and storage and the AI engine that is built with Meta-llama-3.3-70B. Cognifriend crafts a uniquely personalised learning trajectory tailored for every single student. It transitions between the difficulty levels, ensuring that the students are neither overwhelmed nor under-whelmed, this is done by harnessing the new emerging cutting-edge AI through Together API. Cognifriend is built so that the learners are able to embrace the new-age learning with the emerging AI thus proving the traditional methods inefficient.



#### SCHOOL OF COMPUTER SCIENCE AND TECHNOLOGY

### 1. How AI Tutor solves this problem

Cogni Friend is designed in such a way that is intelligent and provides interactive learning.

- 1. Adaptation to difficulty: It dynamically adjusts the question's complexity based on student's performance.
- 2. Generation and evaluation of quizzes: It uses AI to create questions that are context-aware and also assess student's answers.
- 3. Progress Tracking: Records performance of the student as data to personalise future learning paths.

#### Example use case:

- A student who is struggling with algebra is given with questions that are simple in the initial period.
- As there is an improvement in their accuracy, the system progressively introduces problems that are more challenging.
- If a student scores above 85% consistently, then they are moved to advanced level.
- Similarly, scoring below 20% triggers the system so that it generates content that is beginner-friendly.



#### SCHOOL OF COMPUTER SCIENCE AND TECHNOLOGY

### 2. System Architecture

Cogni Friend is built on a modular architecture by integrating multiple technologies for learning experience that is seamless.

Component Technology		Purpose		
FRONTEND	HTML, CSS, JavaScript	Provides an interactive UI for quizzes.		
BACKEND	Node.js, Express.js	Handles quiz generation, answer evaluation.		
DATABASE	MongoDB	Stores user progress, questions, and scores.		
AI ENGINE	Transformer based LLM and LSTM	Generates questions and evaluates responses.		

Table 1 - System Architecture in Brief

#### 2.1 Frontend (User Interface)

Technologies used: HTML, CSS, JavaScript

Role: Provides responsive and engaging quiz interface

#### Frontend Breakdown

- HTML: Defines elements for the quiz(questions, buttons and options)
- CSS: Enhances the user experience by the usage if mobile-friendly, desktop-friendly layout and also behaves intuitive.
- Javascript: Manages dynamic content updates and also handles the user interactions.



#### SCHOOL OF COMPUTER SCIENCE AND TECHNOLOGY

### **Example HTML Code:**

```
<div id="quiz-container">
  <h2 id="question"></h2>
  <div id="options"></div>
  <button onclick="submitAnswer()">Submit</button>
</div>
Example CSS Code:
#quiz-container {
  padding: 20px;
  background: #f0f0f0;
  border-radius: 5px;
  text-align: center;
}
button {
  background-color: #007BFF;
  color: white;
  padding: 10px;
  border: none;
  cursor: pointer;
}
```

**Example JavaScript Code** (This essentially connects the front end to the backend)

```
async function fetchQuestion() {
  const response = await fetch('/api/generate-quiz');
  const question = await response.json();
  displayQuestion(question);
}
```

#### 2.2 Backend (Server logic)

Technologies Used: Node.js, Express.js

Role: Manages API requests, processes AI-generated responses, and handles quiz logic.

#### **Key Responsibilities:**

- 1. Handles user requests(e.g:/generate-quiz to get a new question)
- 2. Integrates with Together AI for evaluation and generation of quiz.
- **3.** Adjusts learning path based on the performance of the student.

#### **Example Express.js Route**

```
app.post('/api/evaluate', async (req, res) => {
  const { answer } = req.body;
  const evaluation = await queryAI(`Evaluate: ${answer}`);
  res.json({ result: evaluation });
});
```

#### 2.3 AI Integration (LLM and LSTM)

Role: The transformer-based LLM and LSTM generate context-aware questions based on difficulty level and various subjects. Evaluate student answers, providing detailed feedback and scoring.



#### SCHOOL OF COMPUTER SCIENCE AND TECHNOLOGY

#### **Example AI workflow**

- 1. Question generation
  - INPUT: "Generate a beginner-level question on Calculus"
  - OUTPUT: "Find the derivative of  $f(x) = 8x^2$
- 2. Answer Evaluation
- INPUT: "Student answer: 16x. Is this correct?"
- OUTPUT: { correct: true, explanation: "The derivative of  $8x^2$  is 16x." }

#### **Example AI API Calls:**

```
async function queryAI(prompt) {
  const response = await fetch('https://api.together.xyz/v1/completions', {
    method: 'POST',
    headers: { Authorization: 'Bearer API_KEY' },
    body: JSON.stringify({ model: 'meta-llama-70b', prompt })
  });
  return response.json();
}
```

#### 2.5 Database (MongoDB)

Role: Storage of quiz questions, user progress, and scores.

Why use MongoDB?

MongoDB is unstructured and easy to use and integrate. The data is often available in JSON format with keys and values and at most times is easy for the system and the model to read and access it.



### SCHOOL OF COMPUTER SCIENCE AND TECHNOLOGY

#### **Collections & Schema**

Collection	Fields	
users	userID, name, currentLevel, progressHistory	
quizzes	questionID, questionText, options, correctAnswer	
scores userID, quizID, score, timestamp		

Table 2 – Schema and data titles used in Mongo DB

#### **Updating a User's Progress After a Quiz**

```
db.users.updateOne(
    { userID: '123' },
    {
        $push: { scores: 85 },
        $set: { level: 'Intermediate' }
});
```

# 3.Step by Step Implementation

### 3.1 Frontend Development

- 1. HTML structure for displaying the questions.
- 2. The responsiveness is provided by CSS styling.
- 3. Implementation of JavaScript logic:
  - Questions are fetched from the backend.
  - Handles the submission of the answer.
  - Displays the result dynamically.



### SCHOOL OF COMPUTER SCIENCE AND TECHNOLOGY

#### 3.2 Backend Development

- 1. Setup Express.js so that it can handle API requests from the client side.
- 2. Define the routes:
  - /generate-quiz: Fetches questions from the AI model.
  - /evaluate: Submits and scores the user responses.
- 3. Integration with together API for generation of quiz.

#### 3.3 AI Model Workflow

Multi-Task BERT-LSTM Model for Educational Content Personalization is the culmination of two components.

#### BERT (bert-base-uncased)

From Hugging Face, we attain the module of Transformers which has the pretrained model for natural language understanding. This model takes input text (like course descriptions or queries) and creates contextual embeddings (numerical representations of meaning).

#### LSTM (Long Short-Term Memory)

LSTM is a type of Recurrent Neural Network (RNN) which processes the sequence of embeddings from BERT to capture sequential/temporal patterns. It helps learn dependencies in the text sequence (especially useful in tutoring/education scenarios)

#### Working of the model

It's a multi-task learning model that predicts three outputs from a student's learning data and textual input:

- Final Score: Regression Task
- Predicts how well a student might score (0 to 1 range).
- Learning Preference: Classification Task

Predicts the type of learner (e.g., Visual, Auditory, Kinesthetic) — using 3 output nodes is an add-on attribute to the project. The project is aimed to be rooted towards the functionality of K-12 learning, but we are using 'Coursera' data only to train and this can be implemented for university students.



#### SCHOOL OF COMPUTER SCIENCE AND TECHNOLOGY

In addition to K-12 learning, we have also included the functionality of identifying the disability type while classifying tasks. The model would predicts if a student has any learning disability (like ADHD, dyslexia, or none) — also with 3 output classes. All three are predicted simultaneously, which improves model generalization.

#### **Libraries and Tools Used**

Tool	Purpose			
PyTorch	Model building, training			
Transformers (HuggingFace)	Accessing BERT			
Pandas	Data handling			
Scikit-learn	Preprocessing (LabelEncoder, MinMaxScaler, train/test split)			
Zipfile & Files (Colab)	Saving and downloading model			
Requests	To call Coursera API			
TfidfVectorizer + Cosine Similarity	For course recommendation system using course titles			

Table 3 – Tools and modules imported for the AI to be functional

You're also combining it with a TF-IDF-based course recommendation system making the system able to

- Analyze and preprocess student data
- Understand text using BERT
- Predict preferences, scores, disabilities
- Recommend relevant Coursera courses

#### 1.3 Database Management

- 1.Storage of quizzes for consistency
- 2. Log student's responses to track improvement.
- 3. Adjusts the difficulty level in a dynamic manner.

NAAC A++ Accredited

#### SCHOOL OF COMPUTER SCIENCE AND TECHNOLOGY

### **4.System Integration Flow**

- 1. User clicks "Start Quiz", this action sends a request to the backend.
- 2. The backend fetches a question from the AI model.
- **3.** The frontend displays the questions along with the choices for the user to choose.
- 4. The student submits the answer and this is sent to the backend for evaluation.
- **5.** AI evaluates the response and sends the feedback to the frontend.
- **6.** System updates the user progress in MongoDB.
- 7. The next question is generated based on the updated learning level.

### **5.Challenges & Solutions**

Challenges	Solution		
CORS Issues	Added middleware in Express - (app.use(cors())).		
AI Latency	Optimized prompts & cached frequent questions.		
Database Efficiency	Used NoSQL structure for fast lookups.		

Table 4 – Prominent challenges faced and the most feasible solution implemented during the course of work.

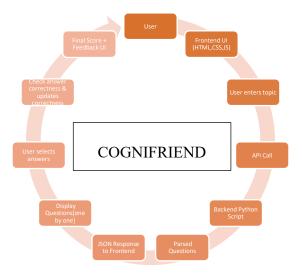


Fig 1- Functionalities of CogniFriend



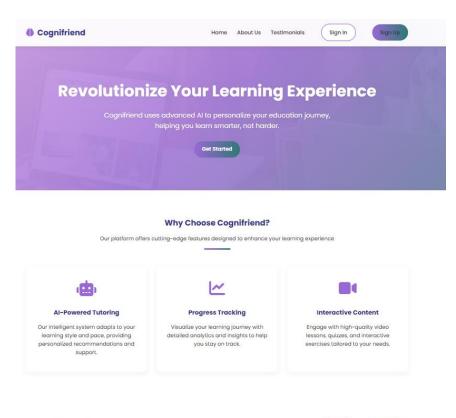
(Declared as Deemed to be University under Sec.3 of the UGC Act, 1956)

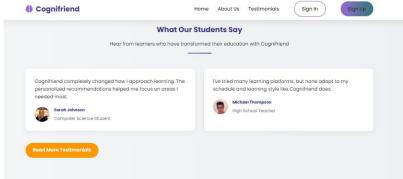
MoE, UGC & AICTE Approved

NAAC A++ Accredited

#### SCHOOL OF COMPUTER SCIENCE AND TECHNOLOGY

### 6. Screenshots of Implementation







#### **About Cognifriend**

Cognifriend was founded in 2022 with a mission to make personalized education accessible to everyone. Our team of educators, data scientists, and engineers have created a platform that leverages artificial intelligence to deliver tailored learning experiences.

- Personalized learning paths based on your strengths and
  weaknesses.
- Adaptive content that evolves with your progress
- Comprehensive progress tracking and analytics

Learn More About Us



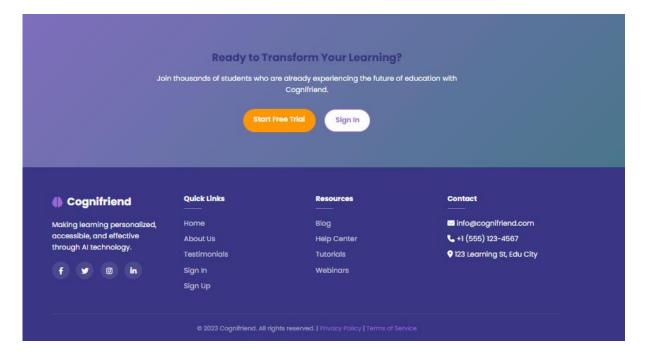


Fig 2,3,4 – Elements of the home page (index.html)



(Declared as Deemed to be University under Sec.3 of the UGC Act, 1956)

MoE, UGC & AICTE Approved

#### NAAC A++ Accredited

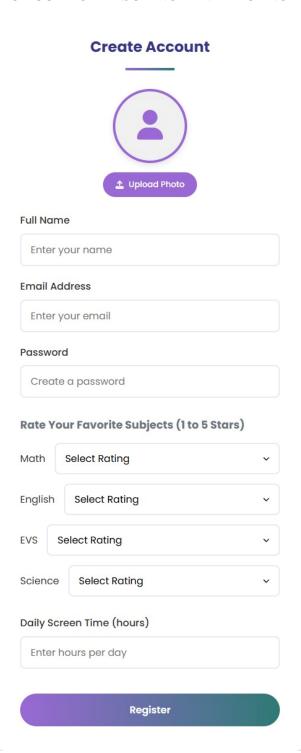


Fig 5 – Sign up page (sign\_up.html)



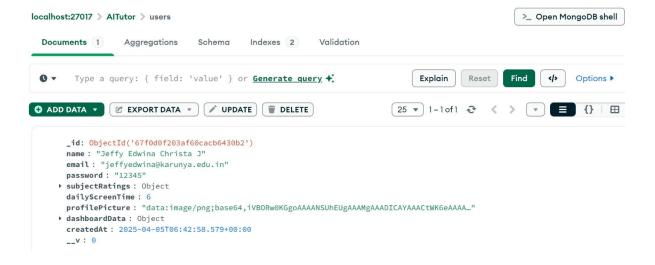


Fig 6 – Database using MongoDB (database name: AITutor, collections name: users)

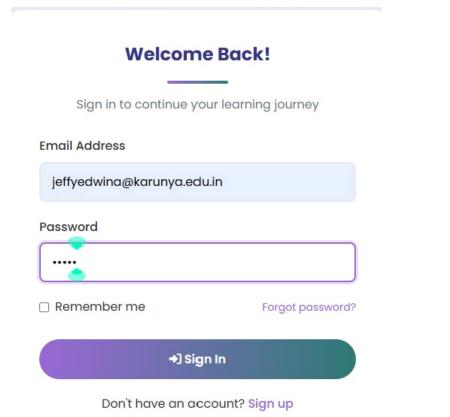
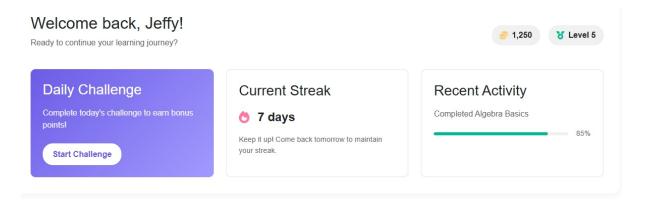
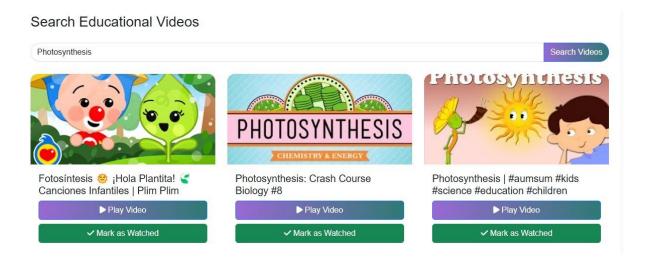


Fig 7 – Sign in page that access the data from the database to login to the page (sign in.html)







# Daily Goals

Complete 1 lesson	+50 pts
Watch 1 educational video	+30 pts
Take a practice quiz	+40 pts
Maintain your streak	+20 pts

Fig 8,9,10 – Dashboard of the student personalized and interactive with a game atmosphere (d.html)



(Declared as Deemed to be University under Sec.3 of the UGC Act, 1956)

MoE, UGC & AICTE Approved

NAAC A++ Accredited

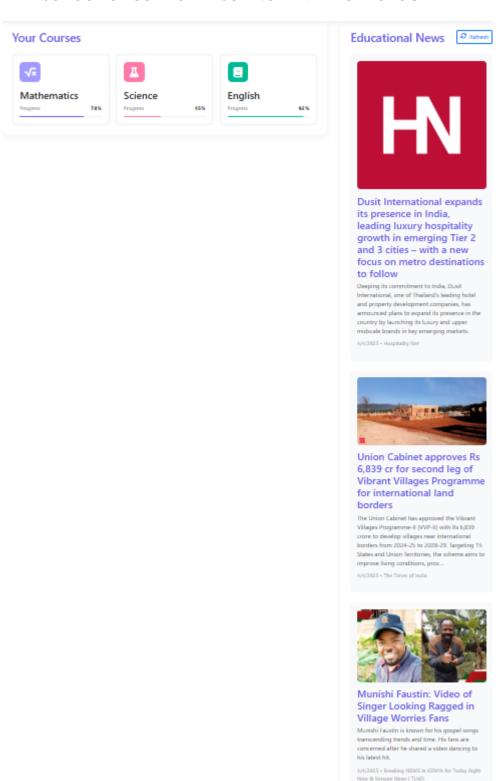


Fig 11 – Dashboard continuation with the progress tracked and educational news (d.html)



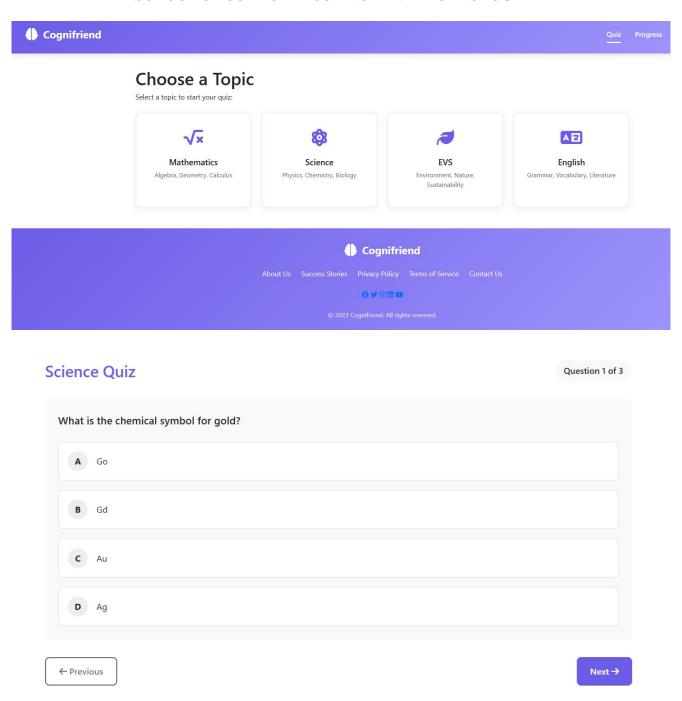


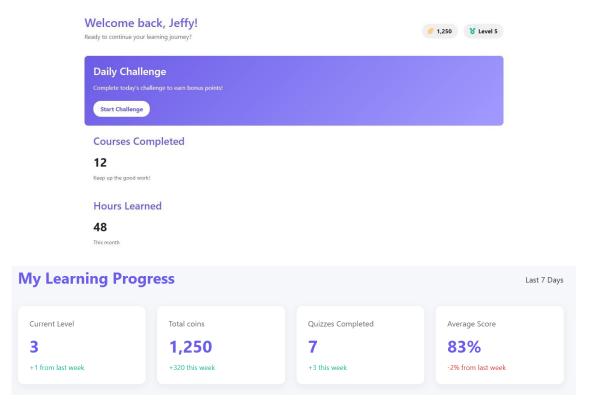
Fig 12,13 – The personalized quiz page with AI (q.html)



(Declared as Deemed to be University under Sec.3 of the UGC Act, 1956)

MoE, UGC & AICTE Approved

#### NAAC A++ Accredited



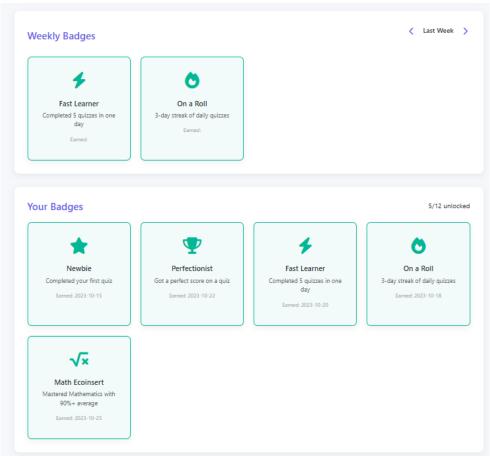
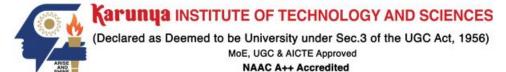


Fig 14,15,16 – Progress page gamified (progress.html)



Pr	Processed Dataset Preview:						
	course_title \						
0	(ISC) <sup>2</sup> Systems Security Certified Practitioner						
1	1 A Crash Course in Causality: Inferring Causal						
2							
3							
4							
	A LITE OF Happiness and Partitione						
	course_organization	Rating	Difficulty	Enrollment	Popularity		
0	(ISC) <sup>2</sup>	4.7	0	5300.0	Low		
1	University of Pennsylvania	4.7	1	17000.0	Medium		
2	Johns Hopkins University	4.5	1	130000.0	High		
3	Yale University	4.7	1	91000.0	High		
4	Indian School of Business	4.8	1	320000.0	High		

Fig 17- The result of the dataset being processed by the AI

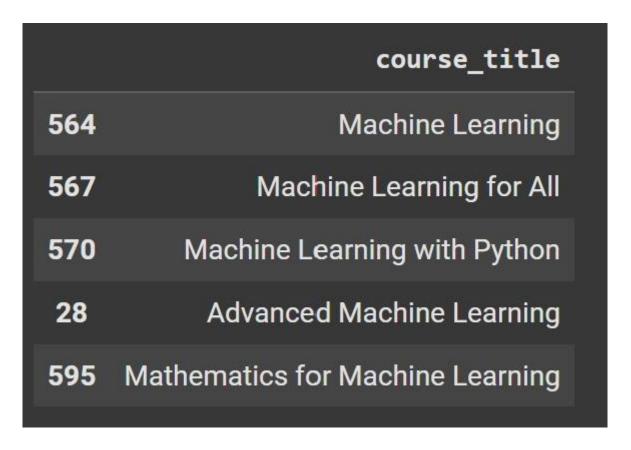


Fig 18 – The courses recommended for college students from Coursera using the AI built.





#### SCHOOL OF COMPUTER SCIENCE AND TECHNOLOGY

#### 7. Conclusion

The AI tutor is revolutionizing education by providing a personalized learning experience, adjusting dynamically the difficulty of the quiz based on the performance of the student It has interactive interface that ensures student engagement through real-time feedback, making traditional learning more enjoyable and effective. This is built on scalable stack of technology that includes AI, Node.js, MongoDB, etc. the platform handles the user requests effectively and manages the data. Additionally, future enhancements will also further enrich the user experience by integrating the multimedia support such as videos explanations and images to make the concepts more concise and intuitive. Gamification elements, weekly leaderboards and daily rewards will aid in boosting the motivation, while the voice interaction will enable students to answer the questions verbally making the AI tutor more engaging.