**EduTutor**

Personalized Learning with Generative AI and LMS Integration

**Team ID :**

LTVIP2025TMID32215

**Team Member(s) :**

**Name :** Anusha Kommula

**Mail :** [Anushakommula645@gmail.com](mailto:Anushakommula645@gmail.com)

**Student ID:** 24B05A0510

**College :**

Shri Vishnu Engineering College for Women

**Department :**

Computer Science & Engineering

**INTRODUCTION**

**1.1 Project Overview**

In today’s fast-evolving digital education landscape, learners seek more interactive, accessible, and intelligent solutions to complement traditional learning. Artificial Intelligence (AI) has emerged as a powerful enabler in this transformation, offering personalized support and smart content delivery.

**EduTutor** is a GenAI-powered educational assistant designed to bridge the gap between static learning content and dynamic learner needs. It combines conversational AI, intelligent retrieval from learning materials, and automatic quiz generation to create a responsive and interactive learning experience.

The system is built using modern web technologies such as FastAPI and HTML/CSS for a smooth user experience, and it integrates with advanced foundation models like IBM Granite and Gemini flash for AI-driven responses.

**1.2 Purpose**

The purpose of this project is to:

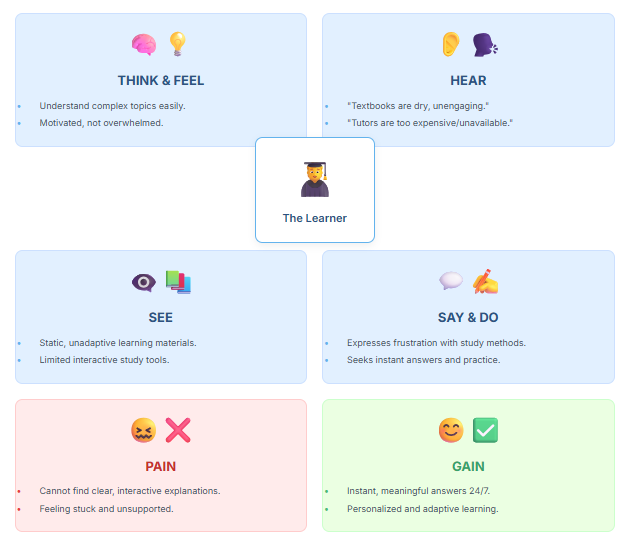
* Provide learners with a smart, AI-driven study companion.
* Enhance accessibility to educational content through natural language interaction.
* Enable self-assessment through on-demand quiz generation.
* Demonstrate the practical use of GenAI and RAG (Retrieval-Augmented Generation) in education.
* Serve as a model for deploying intelligent tutoring systems using open APIs and cloud deployment platforms.

**IDEATION PHASE**

**2.1 Problem Statement**

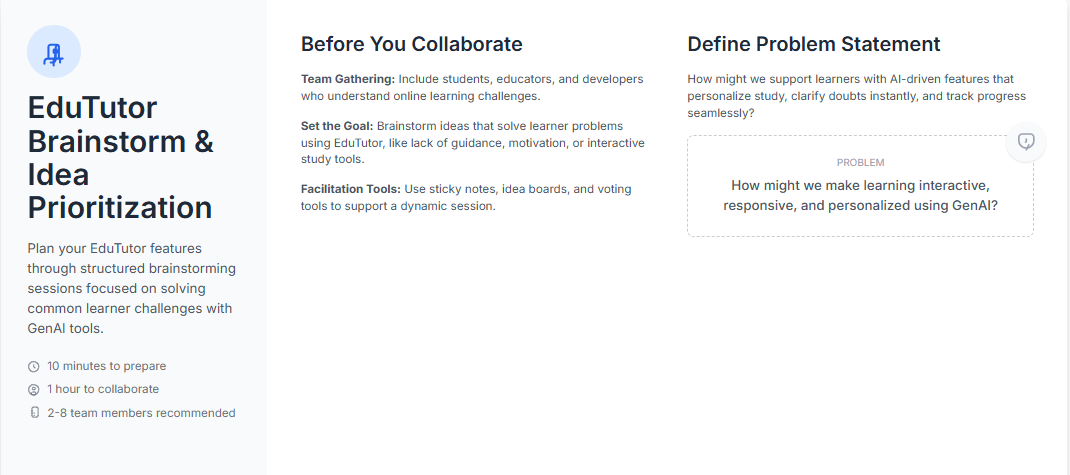
| **Aspect** | **PS-1** | **PS-2** |
| --- | --- | --- |
| **I am (Customer)** | a student preparing for exams | a self-learner exploring new subjects |
| **I’m trying to** | understand complex topics and revise efficiently | ask questions and get instant, meaningful answers |
| **But** | I can't find clear, interactive explanations and practice materials | I don't have a tutor available 24/7 |
| **Because** | traditional materials are too static and don’t adapt to my learning needs | hiring personal tutors is expensive and not scalable |
| **Which makes me feel** | overwhelmed and unmotivated | stuck and unsupported in my learning journey |

**2.2 Empathy Map**

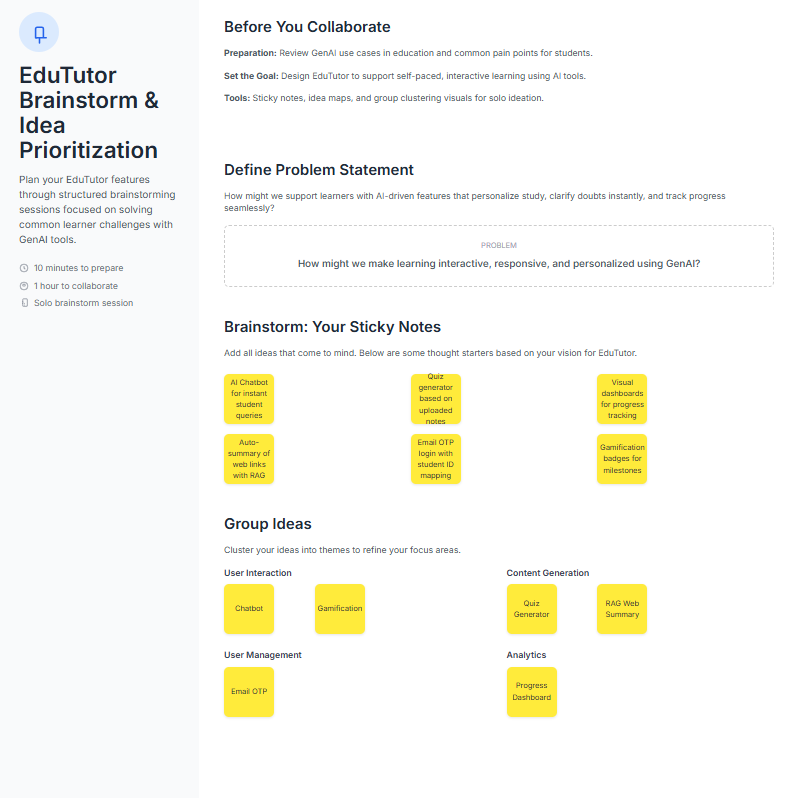


**2.3 Brainstroming**

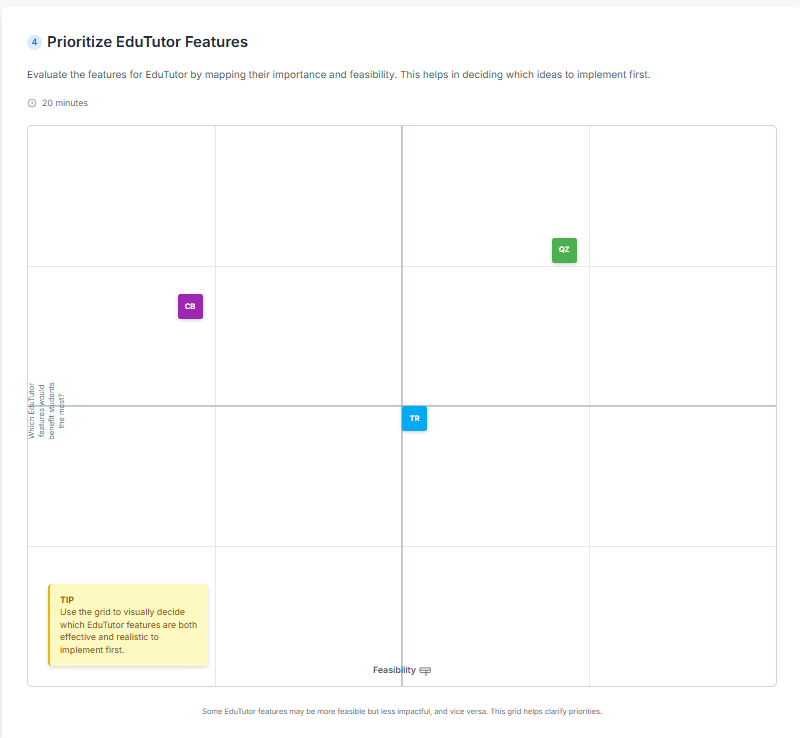
Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping



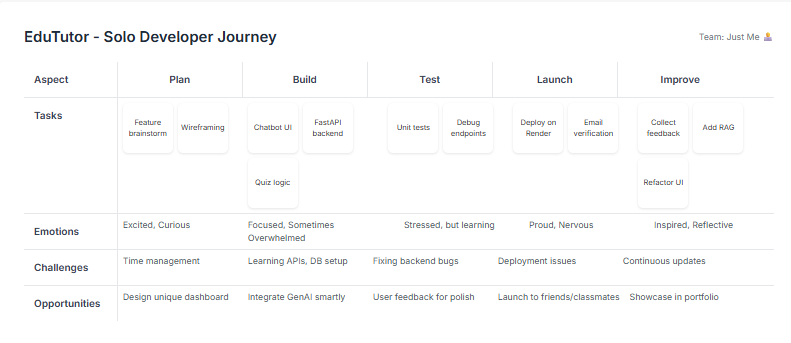
Step-3: Idea Prioritization



**REQUIREMENT ANALYSIS**

**3.1 Customer Journey Map**

The customer journey map for EduTutor illustrates how students interact with the platform from discovery to learning outcomes. It begins with curiosity and the need for academic support, leading them to sign up and explore features like the AI Chatbot and Quiz Generator. During onboarding, clear guidance and a clean dashboard help them navigate easily. As they start using the tools, personalized feedback and engaging quizzes enhance their learning. Students feel empowered through real-time help and relevant resources. Challenges may include adjusting to a new interface or trusting AI-based answers. Positive experiences come from fast responses, simplified notes, and measurable progress. The journey ends with students feeling confident, informed, and motivated to continue using EduTutor regularly.



**3.2 Solution Requirement**

**Functional Requirements:**

Following are the functional requirements of the proposed solution.

| **FR No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| --- | --- | --- |
| FR-1 | User Registration | Registration through Form |
|  |  | Registration through Gmail |
|  |  | Registration through LinkedIn |
| FR-2 | User Confirmation | Confirmation via Email |
|  |  | Confirmation via OTP |
| FR-3 | Dashboard Navigation | Access to Chatbot, RAG, Quiz |
|  |  | Sidebar navigation with dynamic content loading |
| FR-4 | AI Chatbot | Ask academic questions and get instant responses |
|  |  | Use IBM Granite model for intelligent answers |
| FR-5 | Quiz Generator | Generate topic-based quizzes using Gemini API |
|  |  | Display questions, capture responses, and show scores |
| FR-6 | RAG System | Accept URL as input and answer questions based on it |
|  |  | Display summarized knowledge from website |
| FR-7 | Note-taking / Reflection | Allow users to save notes during interactions |
|  |  | Store and retrieve notes in dashboard |
| FR-8 | Feedback System | Allow users to rate quizzes, chat responses, etc. |

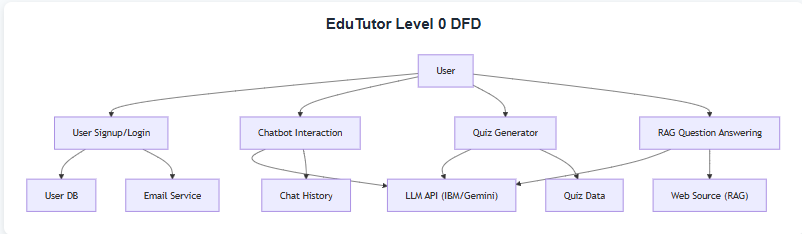
**Non-functional Requirements:**

Following are the non-functional requirements of the proposed solution.

| **FR No.** | **Non-Functional Requirement** | **Description** |
| --- | --- | --- |
| NFR-1 | Usability | User-friendly interface, responsive design for both desktop and mobile use |
| NFR-2 | Security | Secure login, OTP/email verification, HTTPS encryption |
| NFR-3 | Reliability | Accurate model responses, consistent backend behavior |
| NFR-4 | Performance | Fast response time for chatbot and quiz generation |
| NFR-5 | Availability | Platform should be accessible 24/7 with minimal downtime |
| NFR-6 | Scalability | Should support increasing users and API requests without lag |

**3.3 Data Flow Diagram**

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



**User Stories**

| **User Type** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Acceptance Criteria** | **Priority** | **Release** |
| --- | --- | --- | --- | --- | --- | --- |
| Mobile User | Registration | USN-1 | As a user, I can register with email, password, and confirm password | I can access my dashboard | High | Sprint-1 |
| Mobile User | Registration Confirmation | USN-2 | As a user, I receive a confirmation email after registering | I can confirm via email link or OTP | High | Sprint-1 |
| Mobile User | Social Login | USN-3 | As a user, I can register via LinkedIn | I can access my dashboard via LinkedIn login | Medium | Sprint-2 |
| Mobile User | Login | USN-4 | As a user, I can log in with email and password | I can access the dashboard | High | Sprint-1 |
| Mobile User | Dashboard | USN-5 | As a user, I can view the dashboard with Chatbot, RAG, Quiz features | Dashboard loads with all feature options visible | High | Sprint-1 |
| Mobile User | Quiz | USN-6 | As a user, I can take AI-generated quizzes | Quiz generates, accepts responses, and shows score | Medium | Sprint-2 |
| Mobile User | Chatbot | USN-7 | As a user, I can interact with an AI chatbot to ask academic questions | I receive accurate responses from chatbot | High | Sprint-1 |
| Mobile User | RAG System | USN-8 | As a user, I can enter a website URL and ask questions based on it | RAG returns relevant and sourced answers | High | Sprint-2 |

**3.4 Technology Stack**

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

Table 1: System Components

| **S.No** | **Component** | **Description** | **Technology** |
| --- | --- | --- | --- |
| 1 | User Interface | Web-based interface for registration, login, quiz, chatbot, and RAG | HTML, CSS, JavaScript |
| 2 | Application Logic-1 | Backend logic for handling user input, API routing, and UI integration | Python, FastAPI |
| 3 | Application Logic-2 | Handles AI responses via IBM Granite for chatbot and RAG features | IBM Granite (Foundation Model via API) |
| 4 | Application Logic-3 | Quiz generation logic powered by Gemini Flash | Gemini Flash API |
| 5 | Database | *Not applicable* (no database used) | *None used currently* |
| 6 | Cloud Database | *Not applicable* | *None used currently* |
| 7 | File Storage | Minimal local storage for static HTML/CSS/JS | Local Filesystem |
| 8 | External API-1 | AI-powered response generation | IBM Granite API |
| 9 | External API-2 | Quiz generation and cognitive responses | Gemini Flash API |
| 10 | Machine Learning Model | Used through API calls to IBM and Gemini for natural language tasks | IBM Granite, Gemini Flash (via API) |
| 11 | Infrastructure | Deployment-ready FastAPI project | Local or Render Cloud (deployment planned) |

Table 2: Application Characteristics

| **S.No** | **Characteristics** | **Description** | **Technology Used** |
| --- | --- | --- | --- |
| 1 | Open-Source Frameworks | FastAPI (backend), Tailwind CSS (UI), Vanilla JS | FastAPI, Tailwind CSS, JS |
| 2 | Security Implementations | Email OTP verification, HTTPS usage planned, rate-limiting (optional) | FastAPI middleware, SMTP, HTTPS |
| 3 | Scalable Architecture | Modular design; logic separated into quiz, chatbot, and RAG handlers | 3-tier architecture (UI/API/AI APIs) |
| 4 | Availability | Deployable to Render or other cloud platforms; fault tolerance via API fallback handling | Render Cloud, FastAPI error handling |
| 5 | Performance | Lightweight UI, async FastAPI backend, optimized static assets | FastAPI (async), Tailwind, JS |

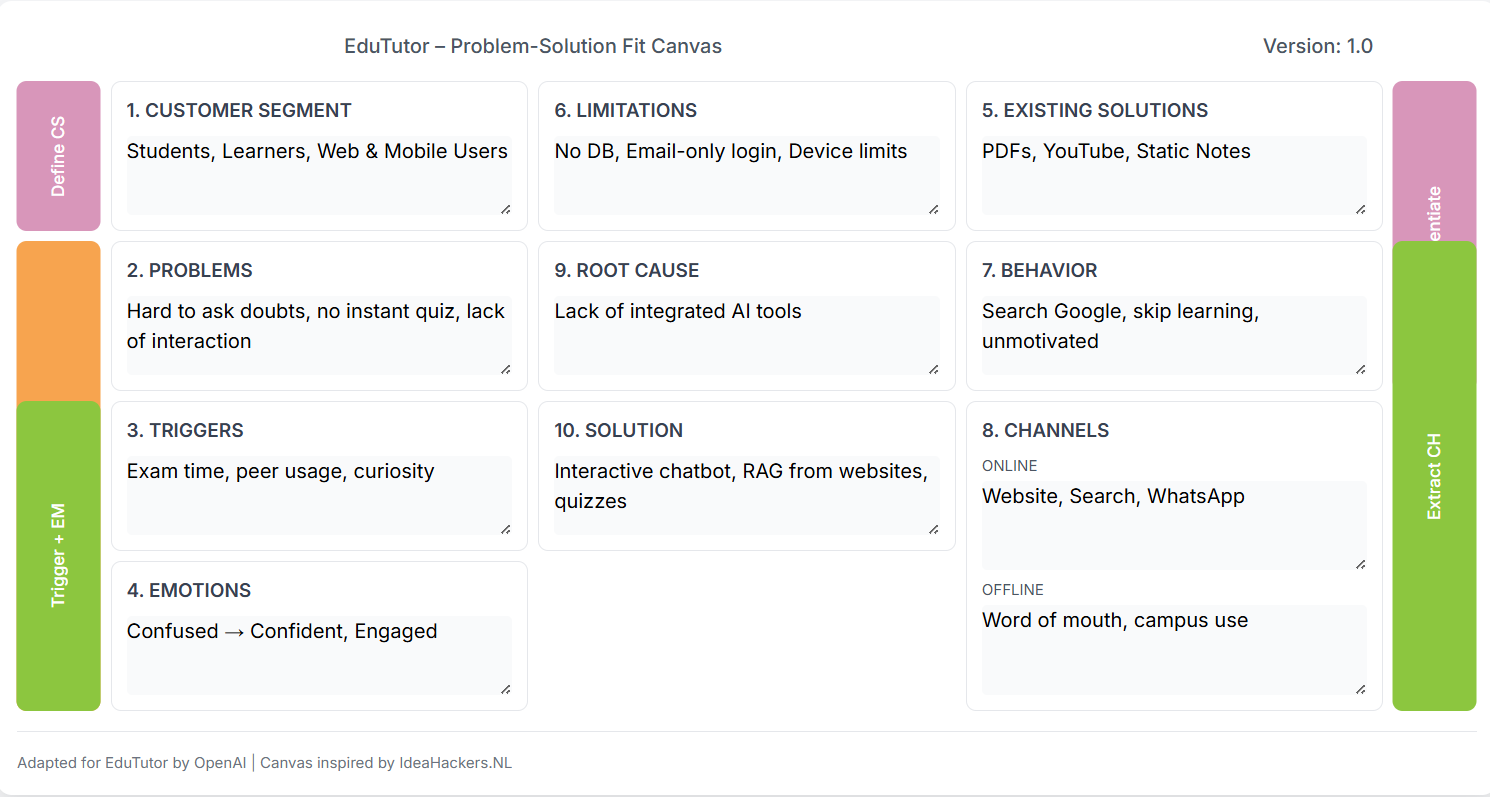
**PROJECT DESIGN**

**4.1 Problem Solution Fit**

The Problem-Solution Fit for EduTutor means identifying real learning challenges faced by students—such as difficulty accessing personalized study support, understanding complex concepts, or managing learning tasks efficiently—and offering a solution that directly addresses these needs. EduTutor integrates AI-powered tools like chat-based Q&A, quiz generation, and RAG (Retrieval-Augmented Generation) to simplify and enhance learning.

**Purpose:**

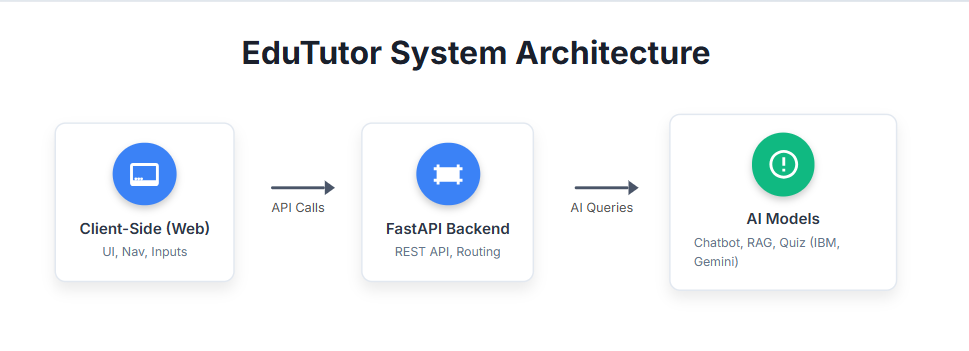
* Help students overcome academic challenges with an intuitive, AI-powered support platform.
* Accelerate learning outcomes and improve tool adoption by leveraging familiar web interfaces and fast-response AI.
* Craft effective communication and onboarding strategies that connect with students' study habits and digital behavior.
* Build credibility and engagement by solving students’ urgent issues like exam preparation and concept clarification.
* Understand the current student experience and improve it through targeted, intelligent educational features.



**4.2 Proposed Solution**

EduTutor is an AI-powered educational platform designed to simplify and personalize the learning experience for students. The solution integrates advanced GenAI tools like IBM Granite and Gemini APIs to support intelligent chat-based doubt solving, real-time website-based RAG (Retrieval-Augmented Generation) for conceptual understanding, and automated quiz generation for self-assessment. Users can access a centralized dashboard where they can chat with a tutor bot, explore topic-wise answers from websites, and take quizzes based on uploaded or referenced content. Built using FastAPI, Python, HTML, CSS, and JavaScript, EduTutor is lightweight, scalable, and tailored for both web and mobile users, making it an accessible solution for modern learners.

**4.3 Solution Architecture**



**PROJECT PLANNING & SCHEDULING**

**5.1 Project Planning**

EduTutor is developed using Agile methodology with 2 sprints (5 days each). Each sprint contains prioritized tasks like chatbot setup, RAG integration, and quiz generation using IBM Granite and Gemini Flash APIs. The team focused on AI interaction, user dashboard interface, and API response handling. Tasks were broken into user stories with estimated story points based on complexity.

**Product Backlog & Sprint Schedule**

| **Sprint** | **Functional Requirement (Epic)** | **User Story No.** | **User Story / Task Description** | **Story Points** | **Priority** | **Team Members** |
| --- | --- | --- | --- | --- | --- | --- |
| Sprint-1 | Chatbot Integration | USN-1 | As a user, I can ask questions to an AI chatbot | 3 | High | Dev A |
| Sprint-1 | RAG System | USN-2 | As a user, I can get answers based on a website link | 4 | High | Dev B |
| Sprint-1 | Dashboard UI | USN-3 | As a user, I see a dashboard with Chatbot, RAG, Quiz tabs | 3 | Medium | Dev A |
| Sprint-2 | Quiz Generator | USN-4 | As a user, I can generate a quiz from content | 5 | High | Dev B |
| Sprint-2 | API Integration | USN-5 | As a user, I get AI responses using IBM Granite and Gemini APIs | 4 | High | Dev A |
| Sprint-2 | Note-taking System | USN-6 | As a user, I can take and save notes on the dashboard | 3 | Medium | Dev A |
| Sprint-2 | Content Display Logic | USN-7 | As a user, dashboard loads tool content dynamically | 4 | Medium | Dev B |

**FUNCTIONAL & PERFORMANCE TESTING**

**6.1 Performance Testing**

**Environment Used:**

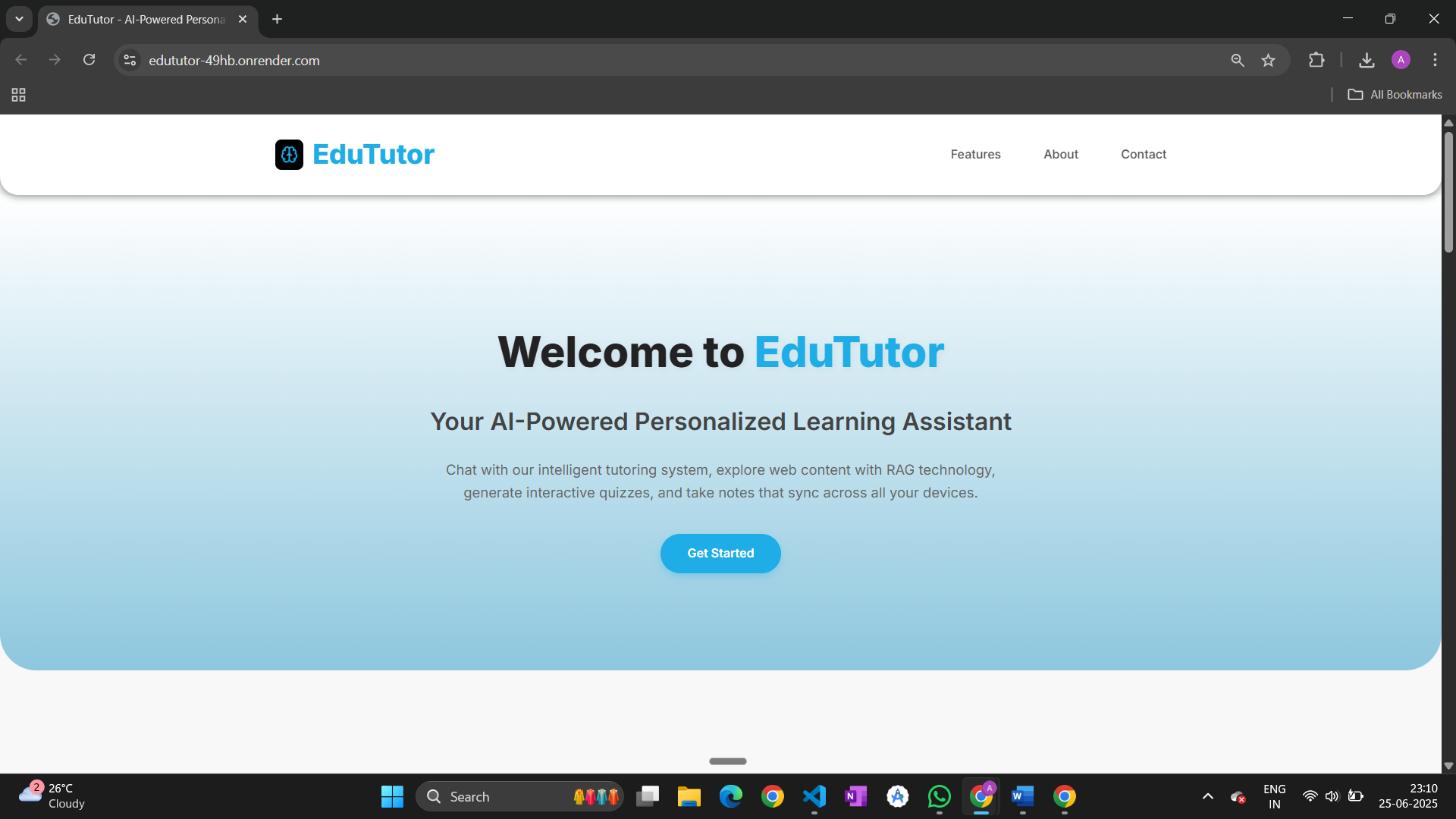
* Localhost (FastAPI backend)
* Frontend viewed via browser (Chrome/Edge)
* Visual Studio Code terminal for logging and debugging

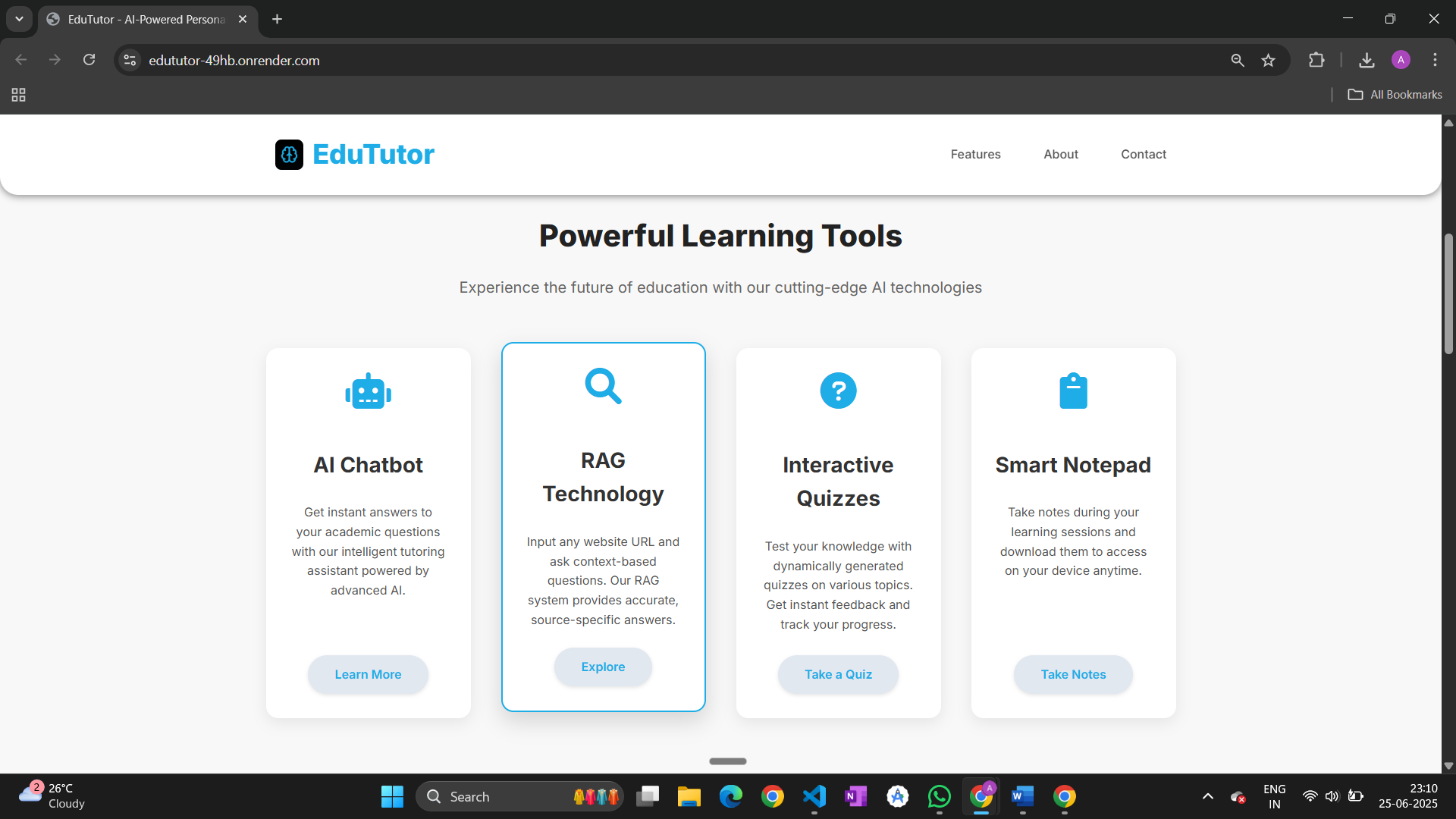
**Performance Metrics and Observations:**

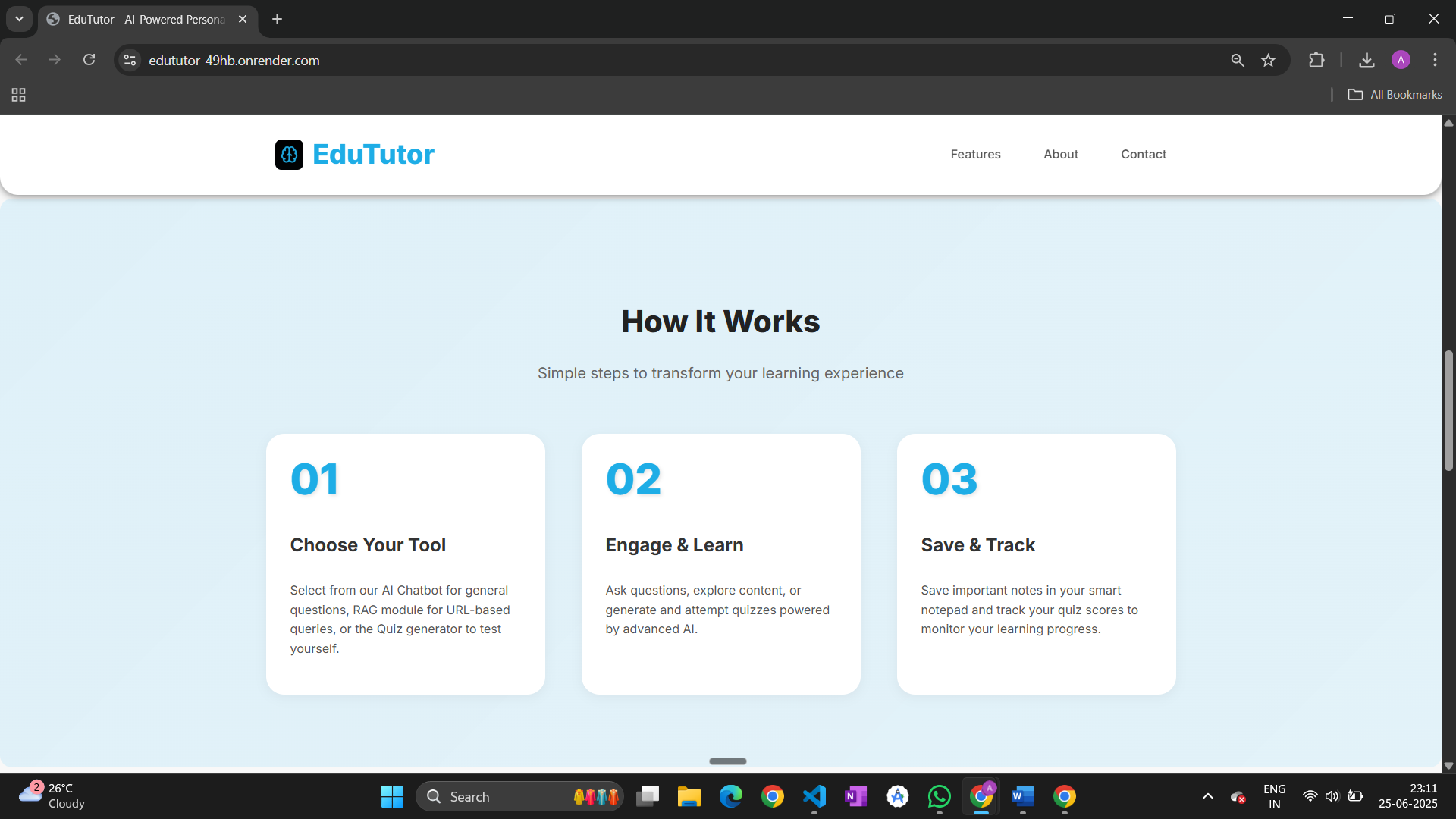
| **Test Area** | **Method Used** | **Expected Result** | **Actual Result** |
| --- | --- | --- | --- |
| Page Load Time (Dashboard) | Browser load timing (Dev Tools) | ≤ 2 seconds | ~1.3 seconds |
| API Call (Chatbot/RAG) | Browser + FastAPI log timing | ≤ 2 seconds | ~1.5 seconds |
| Quiz Generation (Gemini API) | Console log + Browser inspection | ≤ 3 seconds | ~2.1 seconds |
| Multi-Tab Use | Manual testing in 2–3 tabs | Smooth experience | Passed |
| Navigation Speed (Sidebar) | Manual observation | Instant or <1 second | Passed |

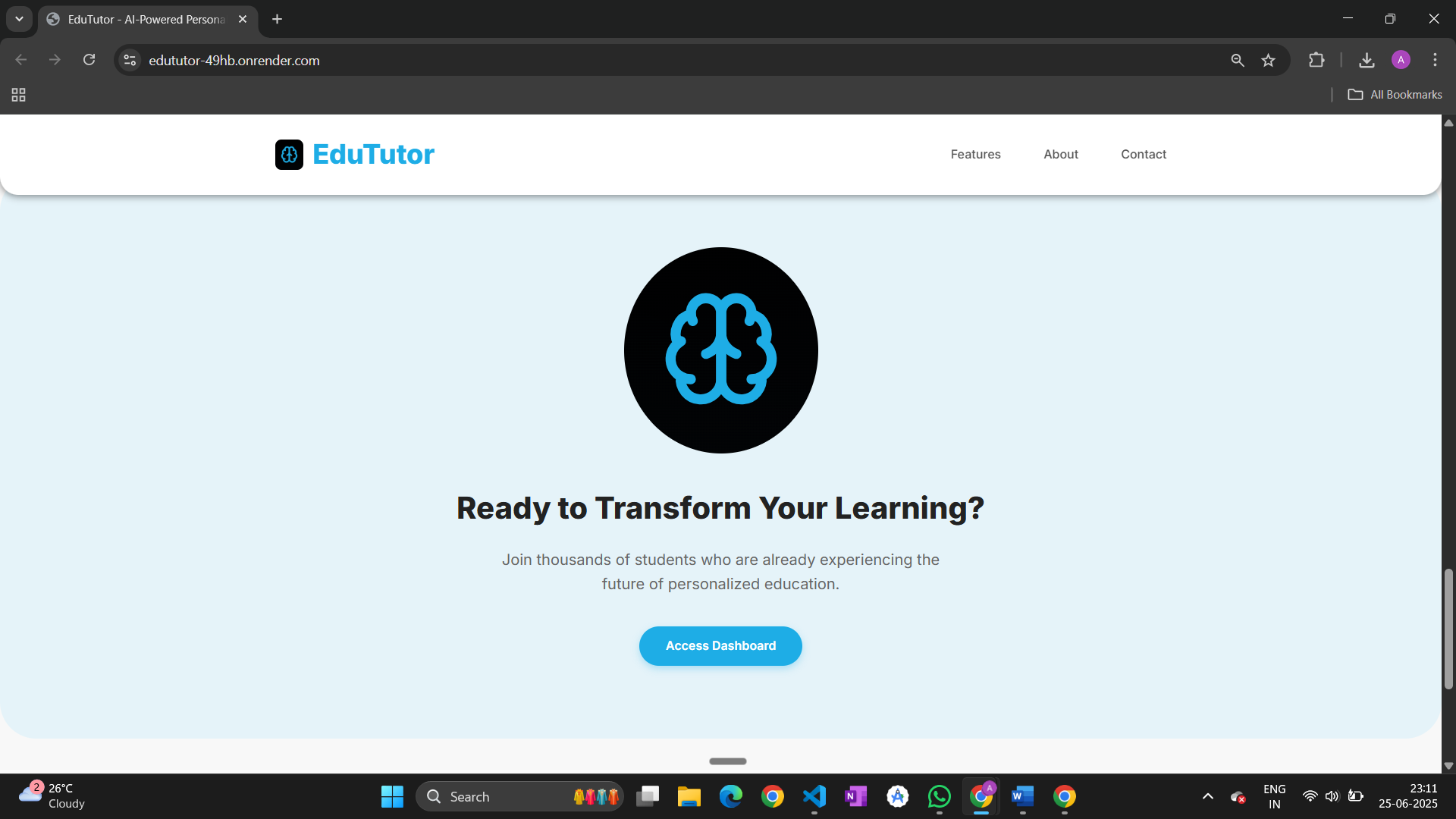
**RESULTS**

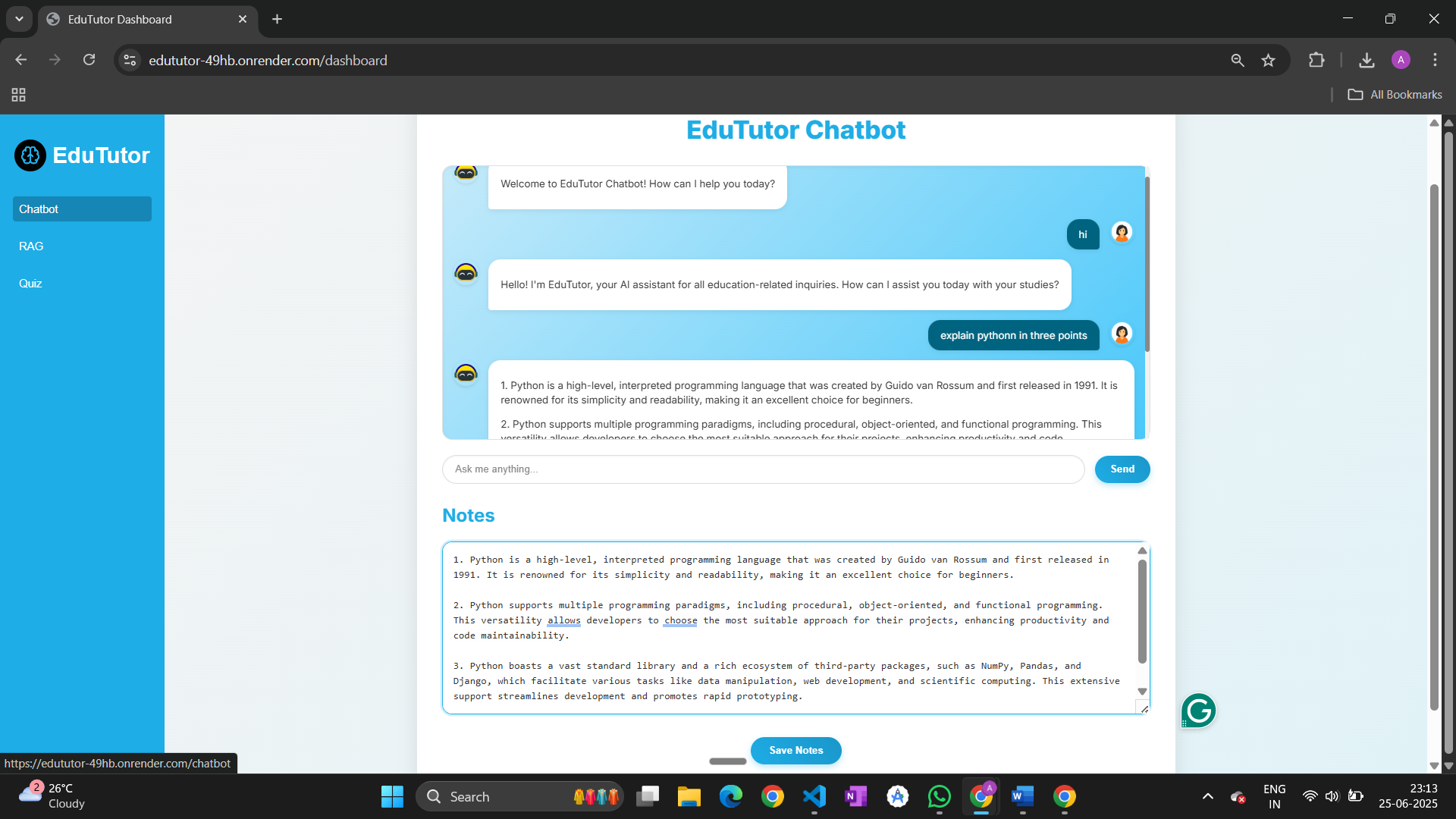
**7.1 Output Screenshots**

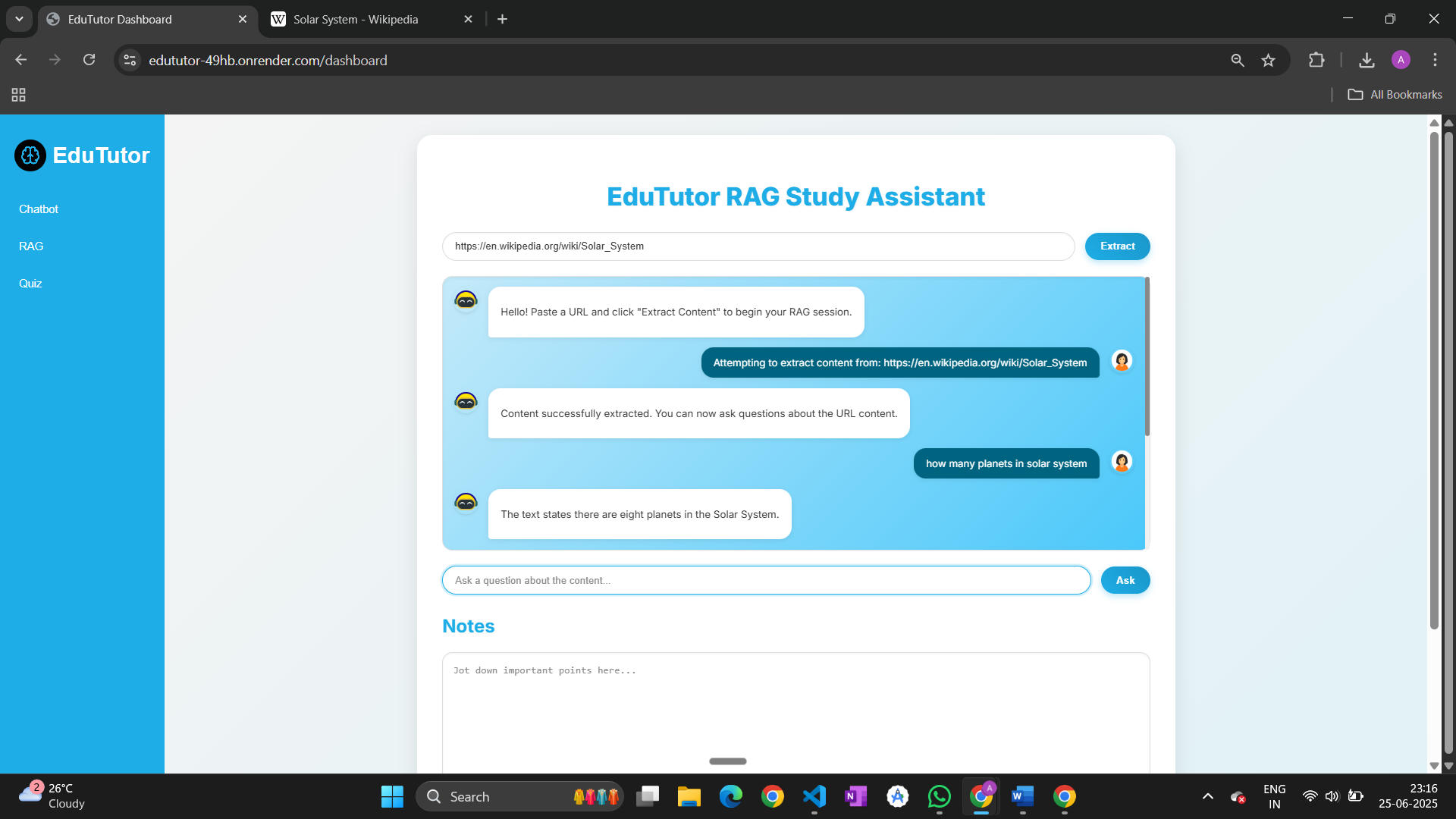
****

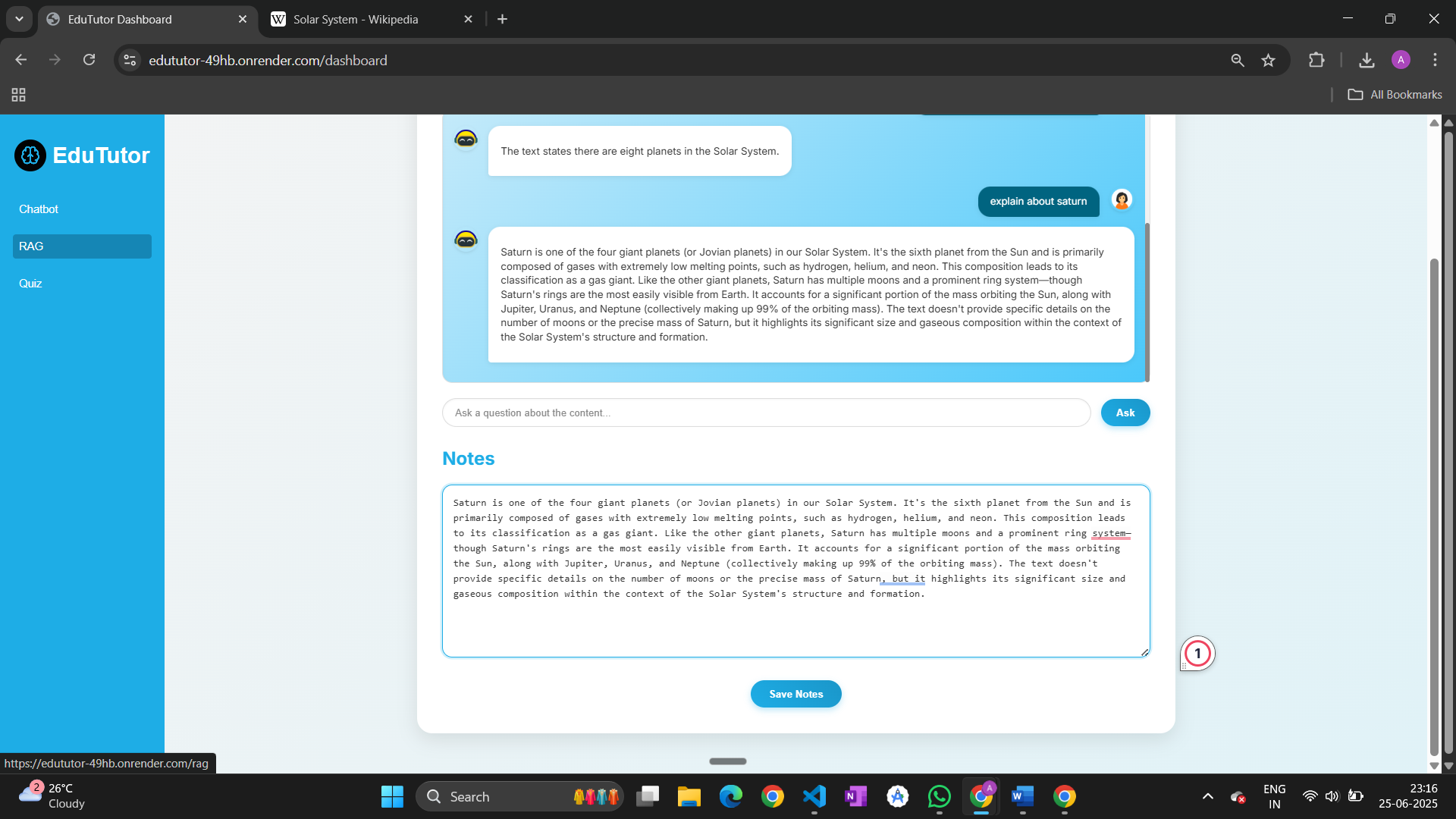
****

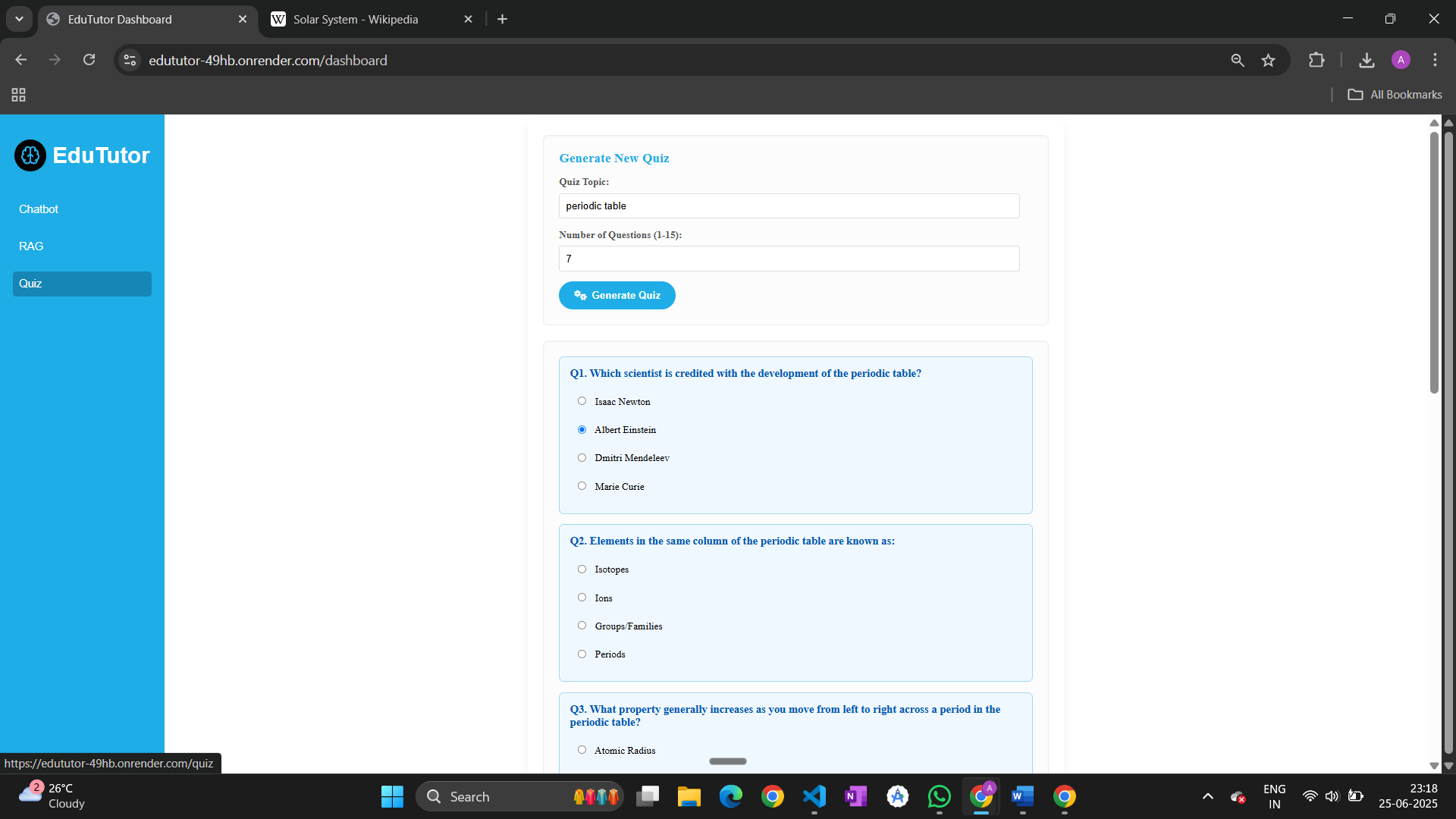
****

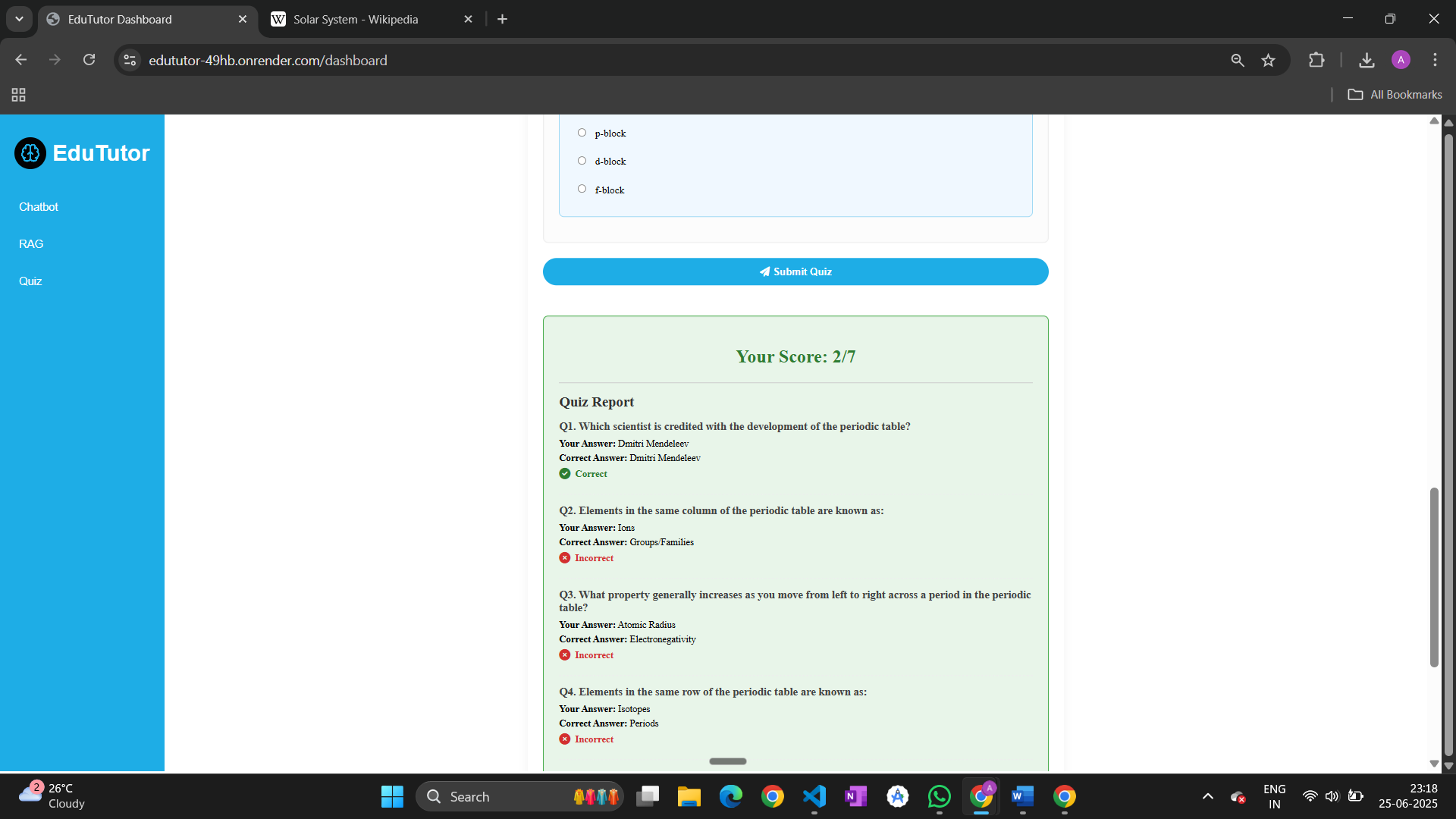
****

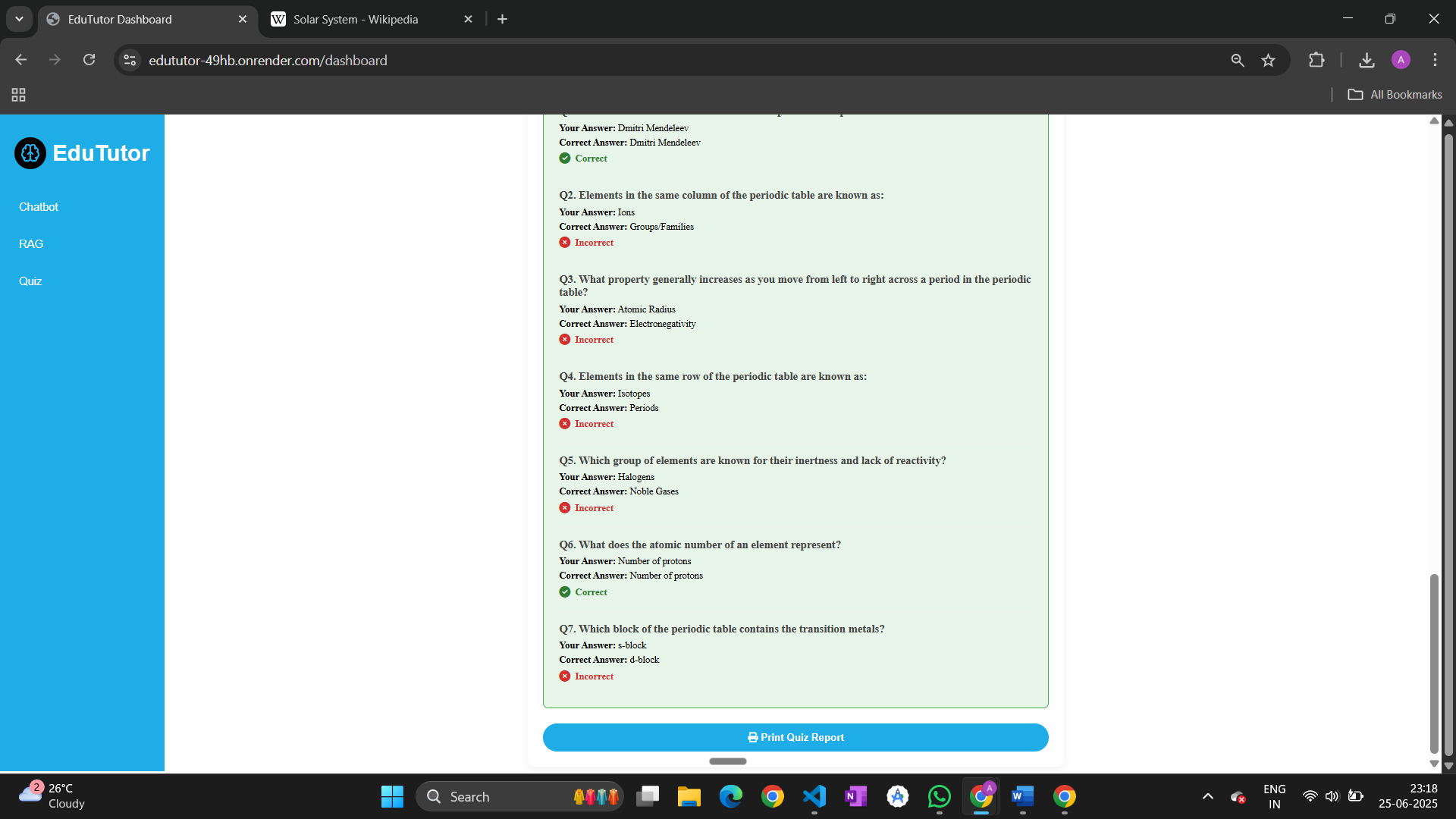
****





****





**ADVANTAGES & DISADVANTAGES**

**8.1 Advantages**

* **User-Friendly Interface**: Clean, responsive UI with easy navigation using HTML, CSS, and JavaScript.
* **AI Integration**: Utilizes IBM Granite and Gemini Flash APIs for smart responses and quiz generation.
* **No Database Overhead**: Lightweight app without the complexity of backend databases.
* **Fast Deployment**: Built with FastAPI and static files, making it easy to deploy and scale.
* **Educational Focus**: Supports self-learning through chatbot, document-based Q&A, and quizzes.

**8.2 Disadvantages**

* **No Data Persistence**: Lacks user authentication and database to store history, notes, or progress.
* **Limited Security**: No user login or OTP verification; not ideal for personalized learning systems.
* **Dependence on External APIs**: Requires stable internet and key-based access to IBM and Gemini APIs.
* **Performance on Large Scale**: Might need optimization or DB integration when scaling to multiple users or features.

**CONCLUSION**

EduTutor is a lightweight, AI-powered educational assistant designed to support learners with an interactive chatbot, RAG-based content retrieval, and a quiz module. Built using FastAPI, HTML, CSS, and Python, and integrated with IBM Granite and Gemini Flash APIs, the solution provides quick, real-time responses without the need for heavy backend infrastructure. While it lacks persistent chat history due to the absence of a database, its simplicity makes it easy to deploy and scale. Overall, EduTutor demonstrates how modern GenAI tools can be leveraged to enhance learning experiences in a minimalistic and efficient environment.

**FUTURE SCOPE**

EduTutor has the potential for significant enhancements to better serve students and educators.

* Add user login, signup, and profile management to personalize access.
* Store and retrieve chat history using a secure database.
* Integrate voice-based interaction for accessibility and ease of use.
* Support for regional languages to widen user reach.
* Implement adaptive quizzes based on student performance.
* Track learning progress with dashboard analytics.
* Introduce file upload and document-based RAG support.
* Expand AI support to include more models for richer responses.
* Optimize the system for mobile devices and low-bandwidth use.
* Enable collaboration tools like shared notes or discussion threads.

**APPENDIX**

**GitHub Repository:**  
 <https://github.com/AnushaKommula-645/EduTutor>

**Project Demo Link:**

Deployed link:  
<https://edututor-49hb.onrender.com>

Demo video:

<https://youtu.be/RXZ5cb0rWYM>