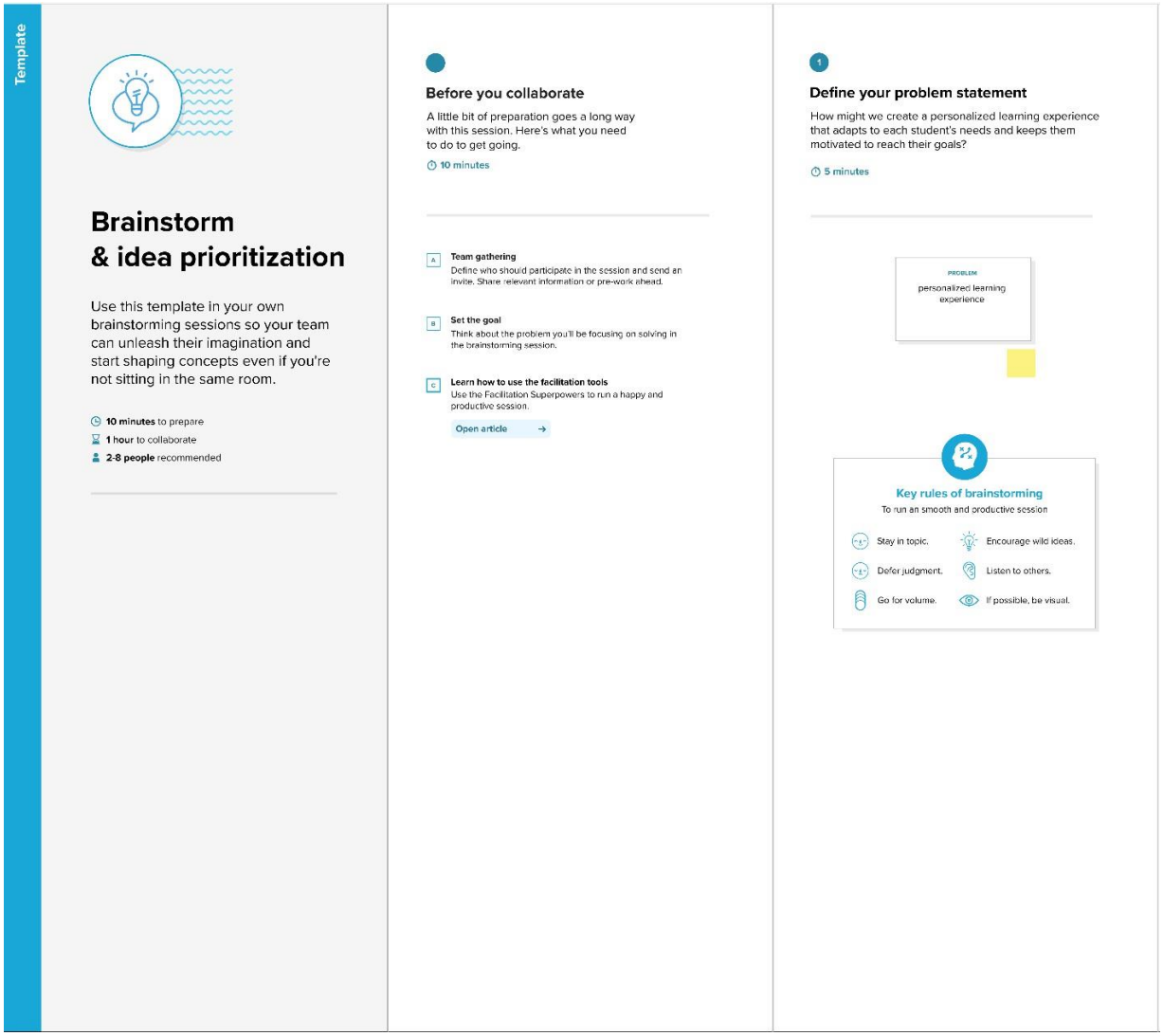
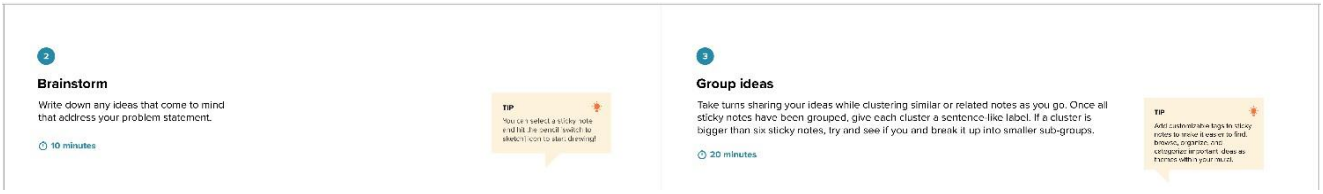
Ideation Phase

❖ Brainstorm & Idea Prioritization Template

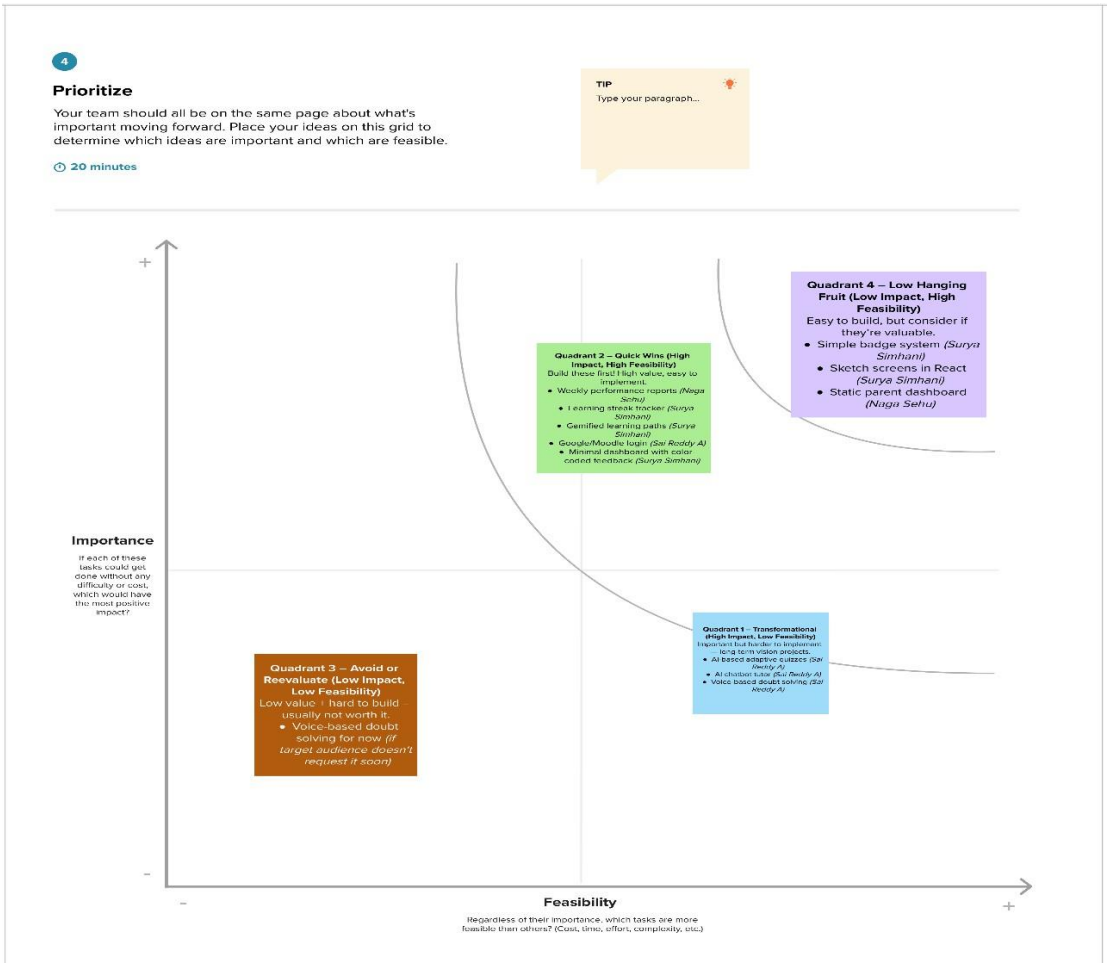
Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization



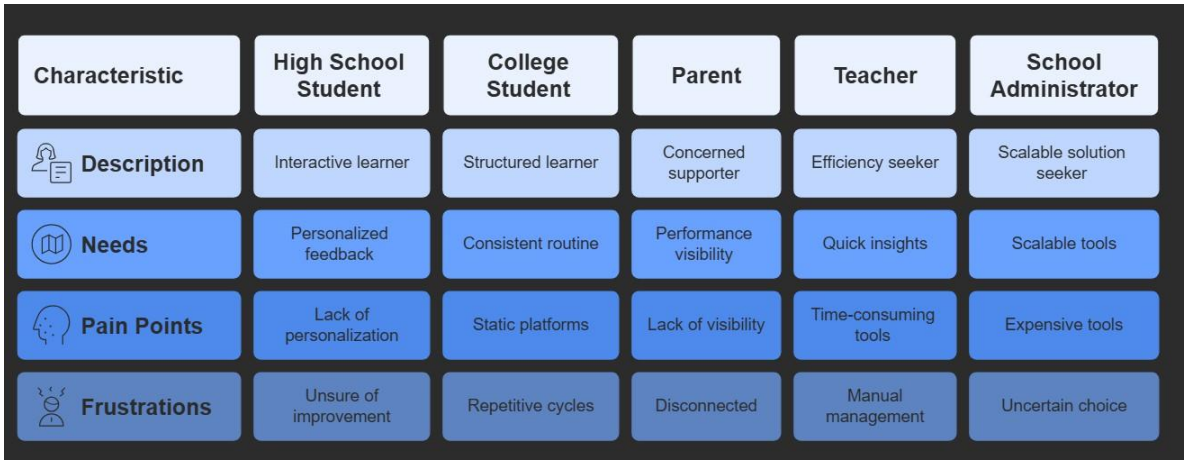
❖ Empathize & Discover

Empathy Map :

❖ Define the Problem Statements

Problem Statement :

PERSONA	I AM	I'M TRYING TO	BUT	BECAUSE	WHICH MAKES ME FEEL
1. Student (High School/College Aspirant)	A high school student preparing for NEET/JEE; curious and prefers interactive learning.	Prepare for exams using personalized mock tests and improvement feedback.	Most platforms are outdated, generic, or lack personalization.	They don't adapt to my performance or learning style.	Anxious and unsure if I'm improving or exam-ready.
2. College Student (UPSC/IAS/GATE Aspirant)	A college student juggling studies and exam prep; needs structured learning.	Build a consistent routine and get feedback from each quiz.	Most tools are not adaptive or designed for exam-specific improvement.	Static platforms don't analyze past performance or suggest progress paths.	Overwhelmed and stuck in repetitive preparation cycles.
3. Parent	A parent concerned about their child's progress and exam readiness.	Help their child stay motivated and understand their performance.	Lack of visibility and actionable insights on student progress.	No dedicated parent dashboard or easy-to-read reports.	Helpless and disconnected from my child's academic journey.
4. Teacher	A teacher aiming to use AI tools to enhance teaching and track student growth.	Assign quizzes and get quick insights into student weaknesses.	Existing tools are time-consuming and not built for teachers' workflows.	Lack of automation, analytics, and integration features.	Frustrated and burned out from managing tools manually.
5. School Administrator	A school admin responsible for ed-tech adoption and improving student outcomes.	Find scalable tools that personalize learning and track school-wide performance.	Tools are expensive, hard to scale, and don't cater to multi-user environments.	Built more for individuals than institutions with analytical needs.	Uncertain about choosing tools that benefit both staff and students.



Performance Testing

❖ Model Performance Test

Test Scenarios & Results

Test Case ID	Scenario (What to test)	Test Steps (How to test)	Expected Result	Actual Result	Pass/Fail
FT-01	Text Input Validation (e.g., topic, job title)	Enter valid and invalid text in input fields	Valid inputs accepted, errors for invalid inputs	Valid inputs accepted, error messages shown for invalid	Pass
FT-02	Number Input Validation (e.g., word count, size)	Enter numbers within and outside the valid range	Accepts valid values, shows error for out-of-range	Valid range accepted, error shown for invalid numbers	Pass
FT-03	Content Generation (e.g., blog, resume)	Provide complete inputs and click "Generate"	Correct content is generated based on input	Relevant and accurate content was generated	Pass
FT-04	API Connection Check	Check if API key is correct and model responds	API responds successfully	API response received successfully	Pass
PT-01	Response Time Test	Use a timer to check content generation time	Should be under 3 seconds	Content generated in 2.3 seconds	Pass
PT-02	API Speed Test	Send multiple API calls at the same time	API should not slow down	All requests responded within normal time	Pass

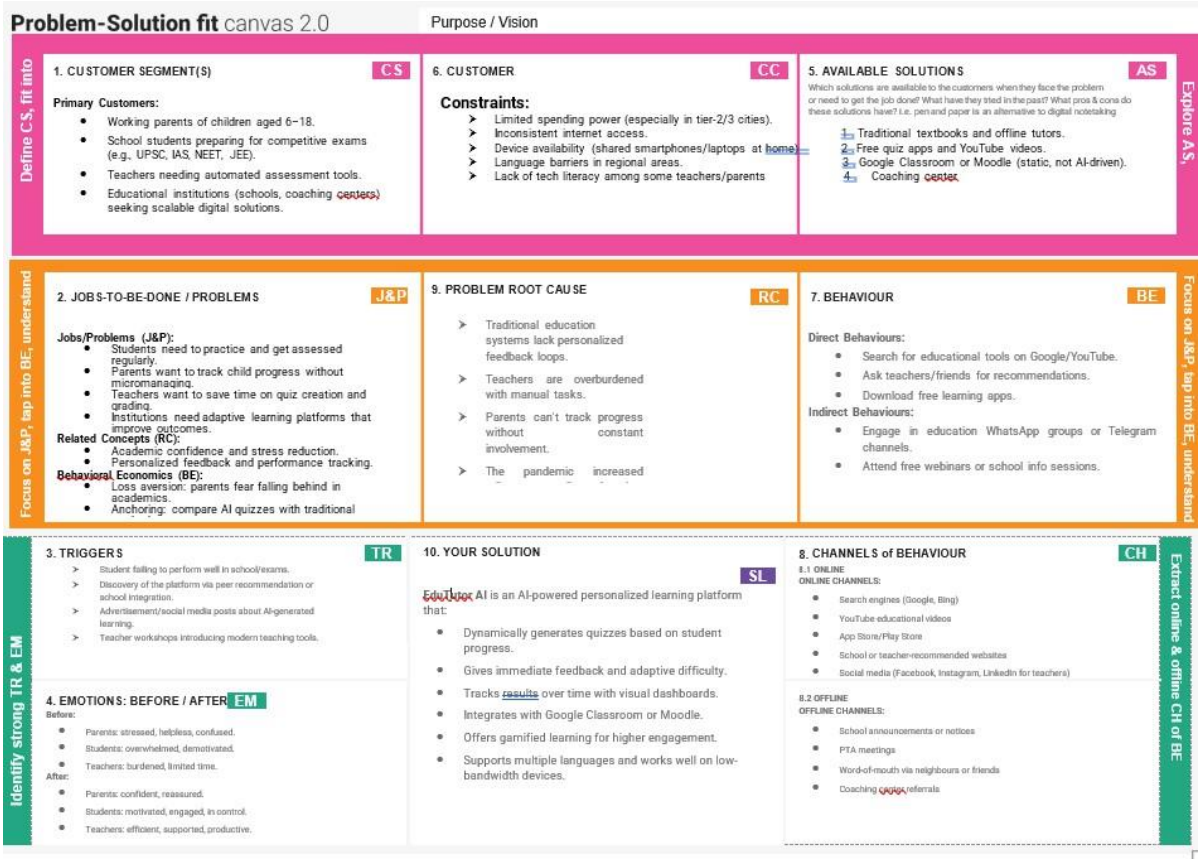
PT-03	File Upload Load Test (e.g., PDFs)	Upload multiple PDFs and check processing	Should work smoothly without crashing		

Project Design Phase

❖ Problem – Solution Fit

Date	27 June 2025
Team ID	LTVIP2025TMID29535
Project Name	EduTutor-AI_personalized-learning-with-generative-ai-and-lms-integration
Maximum Marks	2 Marks

Problem – Solution Fit :



1.

Project Planning Phase

❖ Product Backlog, Sprint Schedule, and Estimation

Use the below template to create product backlog and sprint schedule

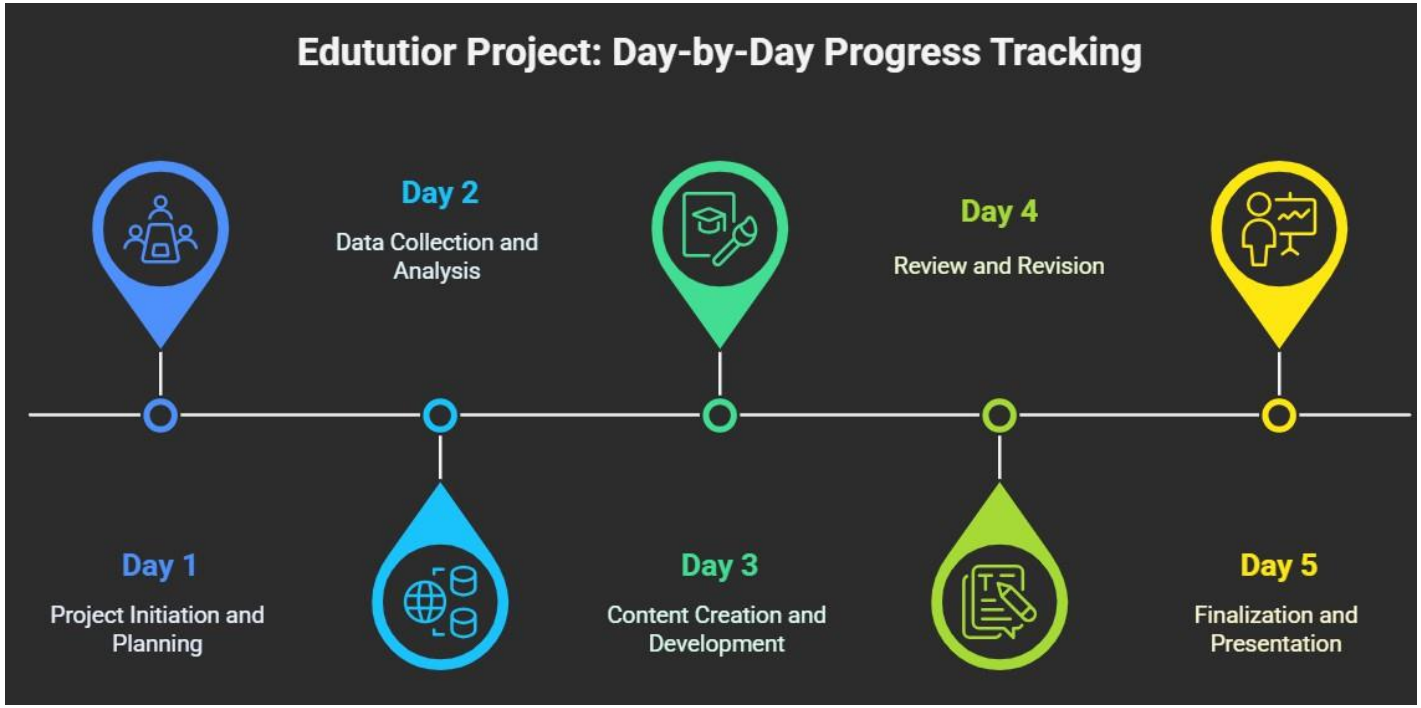
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	User Registration & Login	USN-1	As a user, I can register using email, password, and confirm password.	2	High	**Sai Reddy**
Sprint-1	User Registration & Login	USN-2	As a user, I receive a confirmation email after registering.	1	High	**Sai Reddy ; Srikanth Naga Sehu**
Sprint-1	User Registration & Login	USN-3	As a user, I can log in using email and password.	1	High	**Sai Reddy**
Sprint-1	Moodle Integration (LMS Connect)	USN-4	As a teacher, I can sync topics from Moodle to generate quizzes.	3	Medium	**Srikanth Naga Sehu**
Sprint-1	Quiz Generator (AI Backend)	USN-5	As a student, I get quizzes auto-generated based on my selected topic.	3	High	**Sai Reddy ; Srikanth Naga Sehu**
Sprint-2	Quiz Engine	USN-6	As a student, I can attempt multiple-choice quizzes with a timer.	3	High	**Sai Reddy,**
Sprint-2	Results Dashboard	USN-7	As a student, I can view my quiz scores, wrong answers, and explanations.	3	High	**Sai Reddy**,
Sprint-2	Adaptive Learning	**USN-8**	As a system, I can adjust quiz difficulty based on student performance.	5	Medium	**Sai Reddy**, **Srikanth Naga Sehu**

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2	Deployment	**USN-9**	As a developer, I want to deploy the app using Flask and ensure the frontend connects to the backend.	5	High	**Sai Reddy**, **Srikanth Naga Sehu**
Sprint-2	UI Design & Communication	**USN-10**	As a student, I see a clean, responsive dashboard with subjects, progress bar, and quiz history.	3	Medium	**Surya Simhani** *(Communication & UI Presentation)*

Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed	Sprint Release Date
Sprint-1	10	6 Days	**10 June 2025**	**15 June 2025**	10	15 June 2025
Sprint-2	19	6 Days	**18 June 2025**	**23 June 2025**	19	23 June 2025

Burndown Chart:



Project Design Phase

❖ **Solution Architecture**

Date	15 February 2025
Team ID	LTVIP2025TMID29535
Project Name	EduTutor-Al_personalized-learning-with-generative-ai-and-lms-integration
Maximum Marks	4 Marks

Solution Architecture:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

Solution Architecture Diagram:

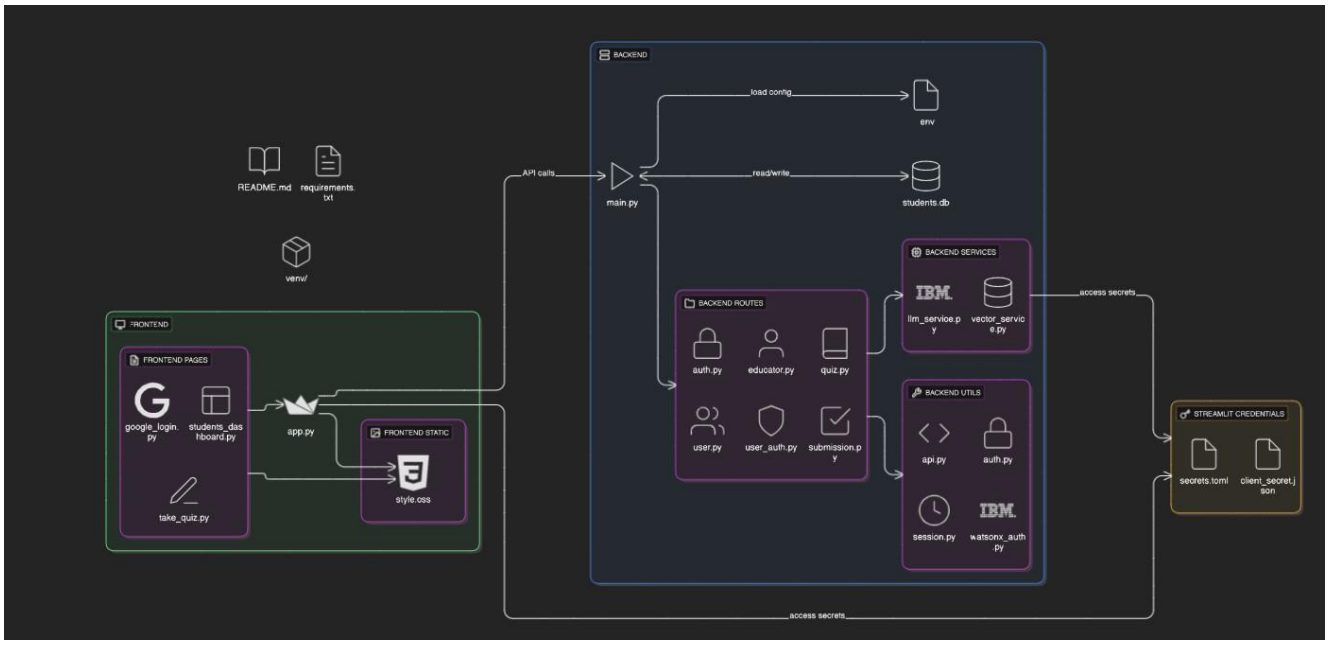
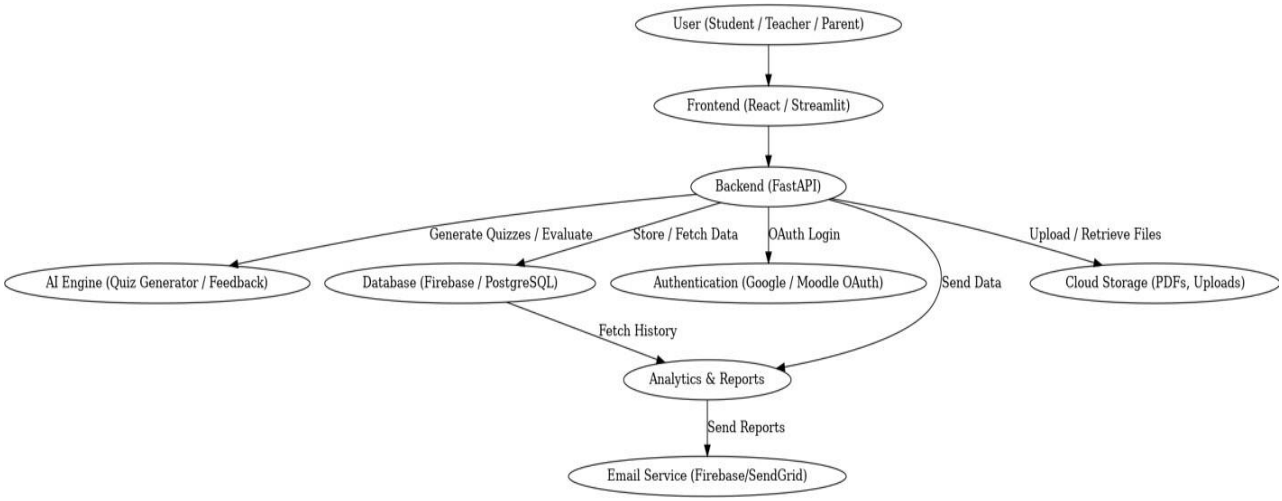


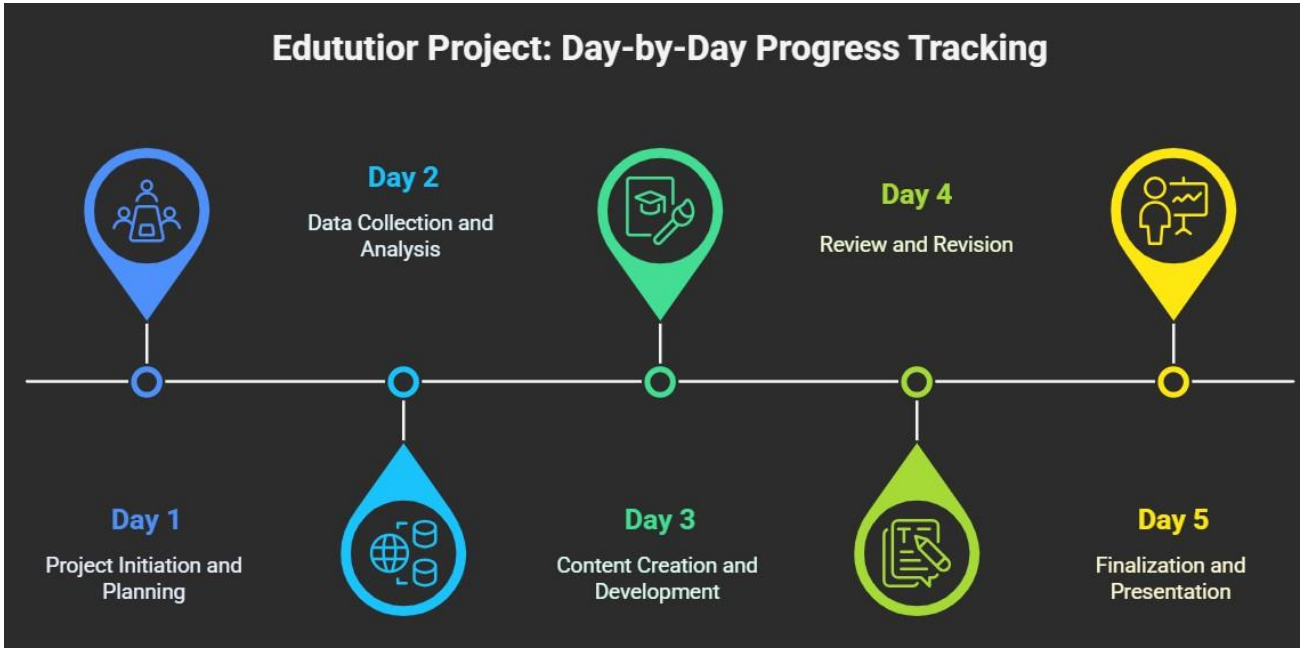
Figure 1: Architecture and data flow of the Edututor AI



Project Design Phase-II

❖ **Data Flow Diagrams:**

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

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance Criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive a confirmation email once I have registered for the application.	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook.	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail.	I can register & access dashboard using my Gmail account	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password.	I can securely log in and reach my dashboard	High	Sprint-1
	Dashboard	–	As a user, I can view my progress, past quiz results, and recommended quizzes.	I can see personalized content after login	High	Sprint-2

❖ Solution Requirements (Functional & Non-functional)

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	Quiz Generation	Generate quiz based on subject/topic
		Generate quiz based on difficulty level
		Use AI to create new questions dynamically
FR-4	Quiz Attempt & Evaluation	Start quiz with timer
		Submit quiz and auto-grade
		View score and correct answers with solutions

Non-functional Requirements:

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

NFR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The system should provide a user-friendly interface with intuitive navigation for students, parents, and educators.
NFR-2	**Security**	The application must ensure secure login, data encryption, and protection against unauthorized access, following OWASP standards.
NFR-3	**Reliability**	The system should be consistently available with minimal downtime and ensure data accuracy and backup recovery.
NFR-4	**Performance**	The system should handle multiple quiz submissions simultaneously with quick response times and low latency.
NFR-5	**Availability**	The solution must be available 24/7 with cloud infrastructure support, including failover mechanisms.

❖ Technology Stack (Architecture & Stack)

Technical Architecture:

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

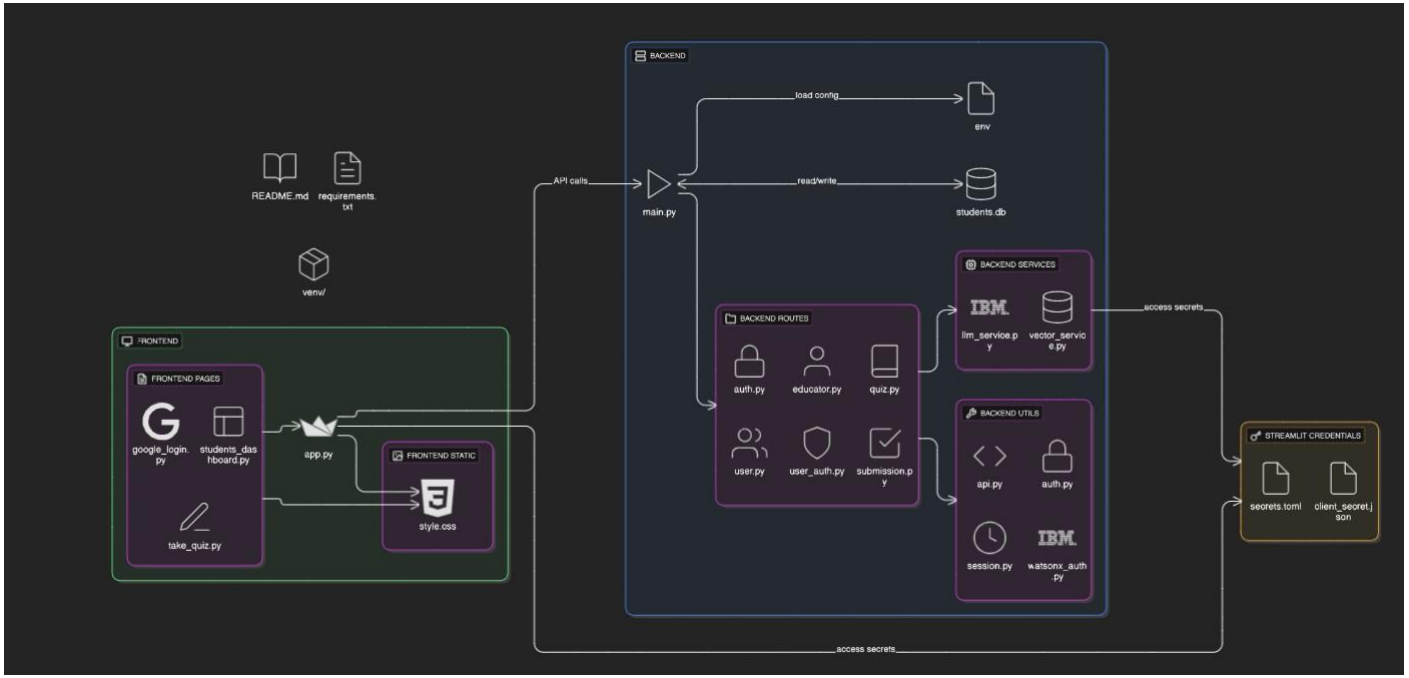


Table-1 : Components & Technologies:

S.No	Component	Description	Technology/Tools Used
1.	**User Interface**	How users interact with the application — for quiz access and results view	**Streamlit** (Student/Admin UI), **React JS**
2.	**Application Logic-1**	Backend logic for quiz generation, student evaluation	**Python** (Flask/FastAPI)
3.	**Application Logic-2**	Voice-to-text conversion for quiz questions (if used)	**IBM Watson Speech to Text (STT)**
4.	**Application Logic-3**	Chatbot for user support or quiz guidance	**IBM Watson Assistant**
5.	**Database**	Stores student records, quiz questions, scores	**MySQL**, **Firebase Realtime DB**, **NoSQL**
6.	**Cloud Database**	Cloud-hosted database for scalability	**IBM Cloudant**, **IBM DB2 on Cloud**
7.	**File Storage**	Stores result exports, user-uploaded files, etc.	**IBM Cloud Object Storage**, **Local Filesystem**
8.	**External API-1**	For quiz content enrichment (e.g., weather quiz, if applicable)	**IBM Weather API**, **Open Trivia DB**
9.	**External API-2**	For secure user verification or personalization	**Aadhar API**, **Google OAuth API**
10.	**Machine Learning Model**	AI-based model for adaptive quiz difficulty or score prediction	**Scikit-learn**, **Custom ML Model**
11.	**Infrastructure**	Where the app is hosted and deployed	**IBM Cloud Foundry**, **Kubernetes**, **Localhost**

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology Used
1.	**Open-Source Frameworks**	Frameworks used for frontend, backend, and ML	**Streamlit**, **React JS**, **Flask**, **Scikit-learn**, **Pandas**, **NumPy**
2.	**Security Implementations**	User data protection, secure login, and API access	**SHA-256 password hashing**, **OAuth2.0**, **JWT**, **IAM Controls**, **HTTPS**, **OWASP Guidelines**

S.No	Characteristics	Description	Technology Used
3.	**Scalable Architecture**	Designed using a 3-tier structure with potential for microservices (modularized quiz engine, auth)	**3-Tier Architecture**, **Docker**, **IBM Cloud Kubernetes**, **FastAPI/Flask APIs**
4.	**Availability**	Deployed with high availability through cloud services, backups, and horizontal scaling	**IBM Cloud Load Balancer**, **Redundancy on Kubernetes**, **Auto-Scaling**, **Cloud Foundry**
5.	**Performance**	Designed for fast quiz access and high concurrency; uses caching and stateless APIs	**Redis Cache**, **CDNs**, **Optimized DB Queries**, **Async APIs**, **Uvicorn (FastAPI)**