# **Technical Documentation**

## **Edu AI– Educational Content Generator**

**Sprint 2 – Week 3 Practical Project**  
 **Category:** Educational Materials

**API:** Gemini Pro (Google Generative AI)  
 **Frontend Technologies:** HTML, CSS

**Backend Technologies:** JavaScript, Server

## **1. Project Overview**

**Edu Ai** is a responsive, web-based AI content generator designed for students and educators across all grade levels. By using Google’s Gemini API, the tool creates high-quality, structured educational content such as quizzes, Step-by step Explanation, summaries, Real –world Examples and Common Misconceptions. It adapts the output based on grade level, content complexity, tone, and subject, making it a versatile tool for personalized learning.

**Primary Goals:**

* Enable students and teachers to generate custom educational materials instantly
* Deliver beginner, intermediate, and advanced-level content across disciplines
* Provide a user-friendly user interface with smart customization, saving, and export functionality

**Use Case:**

**Target Users:**

* Primary, High School and college students seeking study help
* Teachers designing lesson content or quizzes
* Curriculum developers creating targeted learning content

## **2. Features & Functionality**

### **🔧 Technologies Used**

|  |  |
| --- | --- |
| **Component** | **Technology** |
| User Interface | HTML, CSS, JavaScript |
| AI Engine | Gemini API (Google) |
| Device Support | Responsive (desktop/mobile) |

### **🎛️ Customization Parameters**

Users can control output with the following six parameters:

1. **Subject** – e.g., Math, History, Biology
2. **Grade Level** – Elementary to College
3. **Content Type** – Quiz, Lesson Plan, Study Guide, Summary, Activity
4. **Tone** – Formal or Conversational
5. **Difficulty Level** – Beginner, Intermediate, Advanced
6. **Length Preference** – Short/Brief or Detailed

### **⚙️ Functional Components**

* **Prompt Generator:** Assembles a complete prompt using selected parameters
* **Gemini Integration:** Sends request to Gemini API via fetch () and receives structured output
* **Output Display:** Scrollable, stylized display container for results
* **Save/Export:** Users can copy text or download it as a .txt file
* **Input Validation:** All fields required; invalid combinations are blocked
* **Performance Tracking:** Measures response generation time using JavaScript
* **Output Filtering:** Removes redundant or malformed outputs

## **3. Prompt Engineering Methodology**

### **🧠 Prompt Design Strategy**

Each prompt is built dynamically with a system role defined, and user inputs inserted into structured templates. The goal is to yield high-quality, relevant, and instructive results from the Gemini API.

#### **Sample Template:**

"You are a professional educator. Please create a **[difficulty level] [content type]** for **[subject]** designed for **[grade level]** students. Use a **[tone]** tone and provide content in a **[length]** format."

This structure enables consistent performance and clarity across diverse subjects and user needs.

### **🧩 Developed Prompt Templates**

|  |  |
| --- | --- |
| **Template #** | **Content Type** |
| 1 | Quiz Creator |
| 2 | Step-by step Explanation |
| 3 | Concept Summary Tool |
| 4 | Real –world Examples |
| 5 | Common Misconceptions |

Each template is modular, allowing the tool to adjust the difficulty level (Beginner/Intermediate/Advanced) and tone.

### **🧪 Output Quality Filtering**

* Gemini outputs are post-processed with JavaScript to:
  + Remove blank lines
  + Eliminate repeated headers or intro statements
  + Format content into bullet lists or sections

## **4. System Architecture & Performance**

### **⚙️ System Flow:**

1. User selects customization parameters
2. JavaScript constructs the prompt dynamically
3. Prompt is sent via fetch () to Gemini API
4. API returns structured educational content
5. Output is displayed with options to copy or export

### **⏱️ Performance Monitoring:**

|  |  |
| --- | --- |
| **Metric** | **Method** |
| Generation Time | Date.now() before/after API call |
| Token Estimation | Output length-based approximation |
| Error Handling | Try-catch blocks + UI alerts |

## **5. API Usage & Cost Analysis**

|  |  |
| --- | --- |
| **Feature** | **Detail** |
| API Used | Gemini Pro – Google Generative AI |
| Estimated Tokens/Request | 700–1000 tokens |
| Pricing Estimate | Free under dev quota; ~R0.032–R0.18/request |
| Quota Handling | Retry on failure, minimal calls, fallback text |

API integration was designed to minimize calls and handle API limits gracefully, ensuring a stable user experience without triggering usage thresholds.

## **7. Prompt Output Comparison Matrix (Sample)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Prompt Variation** | **Input Sample** | **Output Style** | **Level** | **Quality Notes** |
| Lesson Plan – History – Grade 6 | Conversational, Beginner | Outline + Objectives | Beginner | Clear, age-appropriate |
| Quiz – Biology – High School | Formal, Advanced | MCQs + short answers | Advanced | Well-structured, scientific terms |
| Summary – English Lit – College | Conversational, Intermediate | Thematic overview | Intermediate | Concise, good depth |
| Activity – Math – Grade 4 | Formal, Beginner | Step-by-step problem | Beginner | Interactive, simple steps |
| Study Guide – Physics – College | Formal, Advanced | Concept map summary | Advanced | Dense but organized |

## **Conclusion**

**EduAI** fulfills all the requirements of this Project by combining effective prompt engineering, user-centric design, and advanced Generative AI integration. It is scalable, adaptable, and enhances the learning process through responsive, AI-generated educational content.

TEAM MEMBERS

1. Mthunzi Malebadi
2. Ondela Citywayo
3. Segai Bryton Mampshika
4. Sinesipho Sibulo (Team Leader)