

Project Report: EduTutor AI

1. INTRODUCTION

1.1 Project Overview

EduTutor AI is an intelligent question-answering application powered by IBM Watsonx's foundation models. It allows users—particularly students—to input academic or general queries and receive AI-generated responses in natural language. The app is built using Streamlit and deployed via Streamlit Cloud, integrating IBM Watson's Granite 3B-Instruct model.

1.2 Purpose

The purpose of this project is to create a virtual AI tutor that can enhance students' learning experiences by providing instant answers, explanations, and insights, especially in self-study environments.

2. IDEATION PHASE

2.1 Problem Statement

Students often struggle to find immediate answers to academic questions outside classroom hours. Most existing platforms are either too generic or require paid subscriptions.

2.2 Empathy Map Canvas

- Think & Feel: Needs accurate, fast answers.
- See: Many irrelevant or overly technical answers online.
- Say & Do: Prefers conversational, to-the-point help.
- Hear: From peers—"I wish I had a 24/7 study buddy."

2.3 Brainstorming

- Use IBM Watsonx models for language understanding.
- Build a light, deployable UI with Streamlit.
- Host on Streamlit Cloud for free/public access.

3. REQUIREMENT ANALYSIS

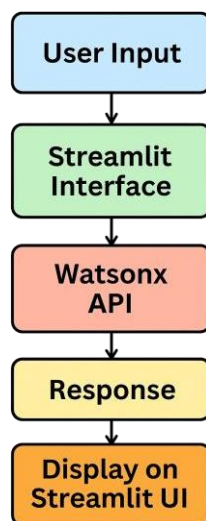
3.1 Customer Journey Map

1. User lands on the app
2. Enters a question
3. Clicks “Get Answer”
4. Receives an AI-generated response
5. Uses the answer for learning

3.2 Solution Requirement

- API Key and Project ID from IBM Cloud
- Python, Streamlit
- Access to Watsonx models

3.3 Data Flow Diagram



3.4 Technology Stack

- **Frontend:** Streamlit
- **Backend:** Python
- **AI Model:** IBM Watsonx (Granite 3B-Instruct)
- **Hosting:** Streamlit Cloud

4. PROJECT DESIGN

4.1 Problem Solution Fit

EduTutor AI aligns with the growing need for AI-based education tools that work as personalized tutors.

4.2 Proposed Solution

An intuitive app where users ask questions and get real-time AI responses.

4.3 Solution Architecture

- Streamlit captures user input
- Backend initializes Watsonx model via API
- Generates text response using IBM's foundation model
- Displays it back to the user

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

| Phase | Duration | Tasks |
|---------------|----------|-------------------------------|
| Week 1 | 3 days | Setup IBM Watsonx & Streamlit |
| Week 2 | 2 days | Model Integration |
| Week 3 | 3 days | Streamlit UI + Deployment |
| Week 4 | 2 days | Testing + Documentation |

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

Tested the app for:

- Model latency (response within 2-3 seconds on average)
- UI responsiveness on mobile and desktop
- API key validation and error handling

7. RESULTS

EduTutor AI

Ask your question:

Explain the concept of gravity

Get Answer

and its role in the universe.

Gravity is a fundamental force of nature that plays a crucial role in the structure and evolution of the universe. It is the force responsible for the attraction between two objects with mass, causing them to be drawn towards each other. The concept of gravity has been studied extensively since ancient times, but it was Sir Isaac Newton who formulated the first comprehensive mathematical description of gravity in his law of universal gravitation.

Newton's Law of Universal Gravitation states that every point mass attracts every other point mass by a force acting along the line intersecting both points. The magnitude of this force (F) is directly proportional to the product of the two masses (m_1 and m_2) and inversely proportional to the square of the distance (r) between their centers:

$$F = G * (m_1 * m_2) / r^2$$

where G is the gravitational constant, approximately equal to $6.674 \times 10^{-11} \text{ N(m/kg)}^2$.

In the context of the universe, gravity has several key roles:

1. **Formation of celestial bodies:** Gravity is responsible for the formation of stars, planets, and galaxies. When a large cloud of gas and dust collapses under its own gravity, it forms a protostar, which eventually becomes a star. Similarly, planets form from the accretion of material around a young star, and galaxies form from the merging of smaller structures under the influence of gravity.
2. **Stability of orbits:** Gravity keeps celestial bodies in stable orbits around each other. For example, the Earth orbits the Sun due to the Sun's gravitational pull, and the Moon orbits the Earth for the same reason. These stable orbits allow for the existence of planets and other celestial bodies in our solar system.
3. **Gravitational waves:** According to Einstein's theory of general relativity, accelerating massive objects can cause ripples in spacetime, known as gravitational waves. These waves propagate at the speed of light and were first directly detected by the LIGO and Virgo observatories in 2015, confirming a major prediction

8. ADVANTAGES & DISADVANTAGES

Advantages

- Fast, natural-language responses
- Easy to use interface
- Free and accessible via browser

Disadvantages

- Depends on API availability
- Limited by token usage and response length
- No voice input or multilingual support (yet)

9. CONCLUSION

EduTutor AI serves as a practical, beginner-friendly AI tutoring app leveraging IBM Watsonx. It demonstrates the integration of cloud AI models with front-end frameworks like Streamlit.

10. FUTURE SCOPE

- Add voice input
- Expand to subject-specific modules
- Add multi-language support
- Use authentication for user tracking

11. APPENDIX

Source Code

```
import streamlit as st

from ibm_watsonx_ai.foundation_models import ModelInference

# Watsonx credentials and settings

model_id = "ibm/granite-3-8b-instruct"

project_id = "d92838f9-a4f7-4728-91c8-0ce485d35bce"

credentials = {

    "url": "https://eu-de.ml.cloud.ibm.com",
```

```

"apikey": "b5Gu5kQpy1-gIVKlujwF5yOSK8vuMYQERhURhPuD_4ju"
}

# Streamlit UI

st.title("EduTutor AI")

question = st.text_input("Ask your question:")

if st.button("Get Answer") and question.strip() != "":

    model = ModelInference(

        model_id=model_id,

        params={

            "decoding_method": "greedy",

            "max_new_tokens": 500

        },

        project_id=project_id,

        credentials=credentials

    )

    response = model.generate(question)

    answer = response["results"][0]["generated_text"]

    st.write(answer)

```

Dataset Link

<https://www.ibm.com/docs/en/watsonx-as-a-service?topic=models-granite-13b-instruct>

GitHub & Project Demo Link

GitHub Repo: <https://github.com/Harini18ux/Harini-N>