INTERNSHIP REPORT

REPORT ON GENERATIVE AI

# THIAGARAJAR COLLEGE OF ENGINEERING, MADURAI

**Department of Information Technology**

****

# SUBMITTED BY DHARSHINI G P AMIRTHA R S

# RASHMIKA K B

**Under the organization of**

**PIVOTAL RESEARCH PVT LTD**

**Number of days: 30.05.2025-30.06.2025**

### ****Disclaimer****

This report and its contents are the intellectual property of **Pivotal Research Pvt. Ltd.** and are intended solely for informational and internal purposes. The information, analysis, designs, and recommendations contained herein are proprietary and confidential, and may not be reproduced, distributed, or disclosed to any third party without the prior written consent of Pivotal Research Pvt. Ltd.

While every effort has been made to ensure the accuracy and completeness of the information presented, Pivotal Research Pvt. Ltd. makes no representations or warranties, express or implied, regarding the adequacy, reliability, or suitability of this report for any particular purpose.

All trademarks, service marks, and logos appearing in this report are the property of their respective owners.

**LIST OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Chapter No** | **TITLE** | **Page No.** |
| 1 | INTRODUCTION | 4-11 |
| 1.1 | GENERATIVE AI | 4 |
| 1.2 | LLM | 4 |
| 1.3 | COST ANALYSIS | 5 |
| 1.4 | FIREWORKS AI | 6 |
| 1.5 | PROMPTING | 9 |
| 1.6 | PYTHON PACKAGES | 11 |
| 2 | WORK FLOW | 12-17 |
| 2.1 | SYNTHETIC DATA GENERATION | 12 |
| 2.2 | MODEL SELECTION | 16 |
| 2.3 | FINE TUNING PROCESS | 17 |
| 3 | DEPLOYEMENT (FIREWORKS) | 20 |
| 4 | CONCLUSION | 21 |

1. **INTRODUCTION**

**1.1 GENERATIVE AI**

**Generative AI** refers to a category of artificial intelligence models that can **generate new content** such as text, images, audio, video, or even code. It learns patterns from existing data and creates outputs that resemble the data it was trained on.

**KEY FEATURES:**

* **Natural Language Understanding :** Understands grammar, meaning, and context in human language.
* **Text Generation** : Can generate explanation, notes, mnemonics, takeaways.
* **Contextual Awareness :** Remembers and responds based on the prompt or previous conversation.
* **Knowledge Retention :** Trained on vast datasets, it can answer questions across many topics.
* **Few-shot / Zero-shot Learning :** Performs new tasks with little or no training examples
* **Scalability :** Works across many industries and can be fine-tuned for specific domains (Education).

# LLM (Large Language Model) :

# 

# An LLM (Large Language Model) is a type of Generative AI designed to understand and generate human language using deep learning techniques.

A Large Language Model is an artificial intelligence model trained on massive text datasets to understand, generate, summarize, translate, and answer questions in natural language.

**1.3 COST ANALYSIS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Key Features | Speed | Pricing (API) | Typical Use Cases | Example Use Case |
| GPT-4o | Multimodal (text, vision, audio), faster, cheaper than GPT-4-turbo | Fast | $5/million input$15/million output | - Advanced virtual assistants  - Customer support bots  - Multimodal search  - Voice & image interpretation | - AI tutor that understands diagrams, solves math problems from handwritten notes, and explains verbally |
| GPT-4-turbo | Cheaper, faster GPT-4 variant with 128k token context | Fast | $10/million input$30/million output | - Large document summarization  - Code generation  - Data extraction from long reports | - Legal document summarizer or contract analyzer |
| GPT-4o-mini | Expected to be lighter, cheaper, faster (experimental phase) | Very Fast | Not publicly priced yet | - Mobile assistants  - Embedded devices  - Edge AI apps needing fast + cheap inference | - Smartwatch voice assistant that replies instantly and supports limited vision/audio tasks |
| GPT-3.5-turbo | Affordable, good for basic chat, QA, and automation | Good | $0.50/million input$1.50/million output | - FAQ bots  - Entry-level coding help  - Email and text generators- Customer service | - AI chatbot for a small business website to answer customer queries about services and pricing |
| Gemini 1.5 Pro | 1M token context, multimodal, highly capable | Good | $3.50/million input $10.50/million output | - Multimodal RAG - Deep memory AI - Enterprise assistants | AI analyst summarizing large meetings + presentation slides |
| Gemini 1.5 Flash | Lightweight, cheaper, very fast, 1M context | Very Fast | $0.35/million input $1.05/million output | - Realtime Q&A - Translation - Lightweight enterprise tools | Instant summarizer of user chat transcripts |
| Qwen 1.5 (7B/14B) | Good for Chinese-English tasks, open-source | Good | Free | - Bilingual chatbots - Education apps - Code generation | App that explains code in both English and Mandarin |
| Qwen2 | Good for Chinese-English tasks,open-source | Fast | Free | -Developer assistant  -Customer support  -Search apps | Coding assistant in integrated IDE |

**ED TECH STARTUP – AI LEARNING ASSISTANT FOR HIGH SCHOOL STUDENTS**

This is a LLM based Learning assistant for High School students it helps to prepare for their exams. For the selected topic by the students this app generates explanation.

Using the LLM model to classify the contents and give explanation, notes, mnemonics and takeaway taglines.

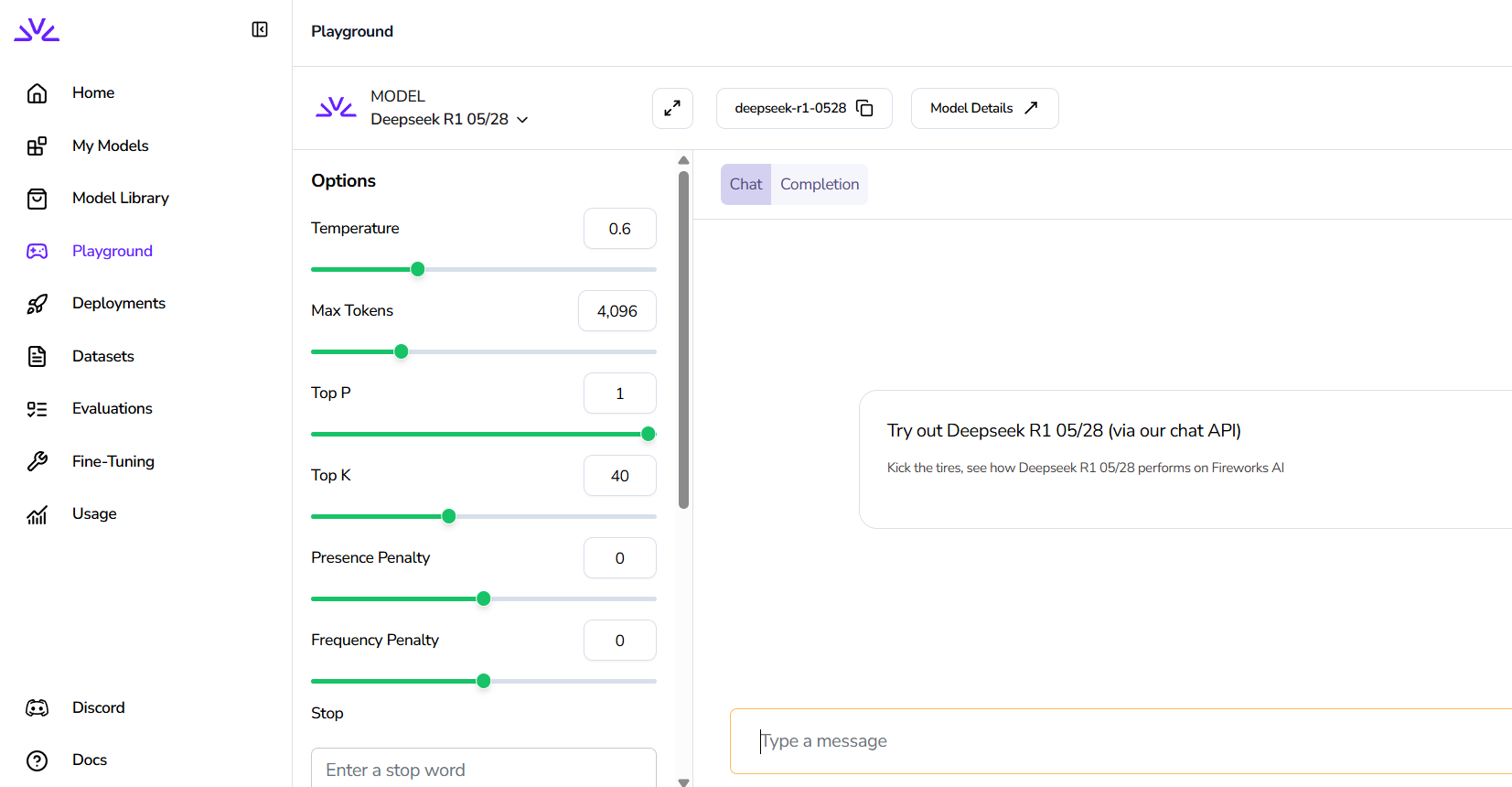
* 1. **FIREWORKS AI**

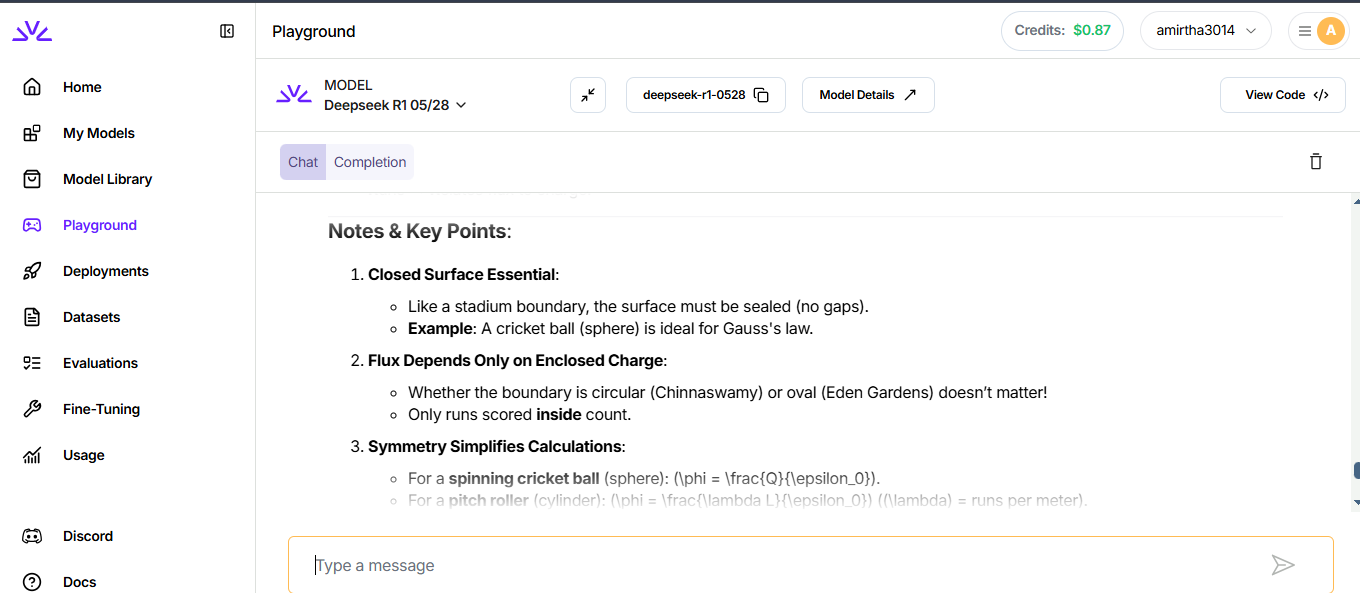
Fireworks AI is a platform that helps developers easily use powerful AI models (like ChatGPT-style models) in their own apps or tools.

# FEATURES:

* **LLM Hosting**: Run open-source models like Mistral, LLaMA, or Mixtral efficiently on their platform.
* **Fine-Tuning**: Customize models using your own data.
* **Low Latency**: Fast inference speeds for real-time use.
* **Enterprise Focus**: Tools tailored for startups and companies integrating generative AI into their products.
* **API Access**: Developers can access models through simple APIs, similar to OpenAI’s API.

**FIREWORKS AI PLATFORM:**



****

|  |  |
| --- | --- |
| Option | Meaning (Simple Explanation) |
| Temperature (0.6) | Controls **creativity**. Lower = more focused and predictable answers. Higher = more random and creative.  🔹 0.6 = balanced creativity. |
| Max Tokens (4096) | Controls **how long the model's output can be**.  1 token ≈ 1 word (not exact). Higher = longer answers. |
| Top P (1) | **Nucleus sampling**. The model looks at the smallest group of words that add up to P probability.  🔹 1 = considers **all** possible words (no filtering). |
| Top K (40) | Limits to the top K most likely next words.  🔹 40 = considers the **top 40 likely options** only. Lower = more strict. |
| Presence Penalty (0) | Discourages the model from repeating the **same topics**.  🔹 0 = no penalty, can repeat topics freely. |
| Frequency Penalty (0) | Penalizes repeating the **same words** too often.  🔹 0 = allows repetition. Higher = less repetition. |
| Stop | You can enter a **stop word** or phrase that tells the model when to stop generating output.  🔹 (Empty now) = model keeps generating until it hits max tokens. |
| Context Length Exceeded Behavior (None) | What to do if the input is too long.  🔹 None = do nothing; might cause errors if context exceeds token limit. |
| Echo (Off) | If ON, the model **repeats your input** before answering.  🔹 Off = only responds with the output (recommended for cleaner replies). |
| Function Calling | Lets the model **interact with functions** defined using JSON Schema.  🔹 Currently no functions are added. Useful for tools like weather, search, calculator, etc. |
| Input Transformations | Optional settings to **modify the input** before it goes to the model.  🔹 “Enable Document Inlining” lets large documents be injected directly into prompts if needed. |

**1.5 PROMPTING**

A prompt is a text instruction, question, or input that guides the AI to generate a specific type of response.

The four main areas to consider when writing an effective prompt are:

**• Persona - *Who should the AI act like?***

**• Task - *What do you want it to do?***

**• Context - *What information should the AI know?***

**• Format – *How should the response be structured?***

**Example:**

"You are a professional resume writer. Write a resume summary for a software engineer with 5 years of experience in Java and Python, aiming for a backend developer role. Return the summary in 3 bullet points."

**1.Prompt:**

explain gauss's law by considering them into anyone of the categories of concept driven or memory driven or foundational recall with mnemonics, mindmap, flash cards, if needed for the category that can make the student understand the topic quick and clear and make it crisp and also provide notes and breif explanation, and make sure they are related to Indian context and without foul language and sterotyping and make the content related to cricket which makes the student who are intrested in cricket.

**Output:**

We are going to explain Gauss's law in the context of cricket, which is very popular in India. We'll use the concept-driven approach with mnemonics and a mind map to make it quick and clear. We'll avoid foul language and stereotypes.

Gauss's Law: Concept-Driven Explanation with Cricket Analogy

Core Concept: Gauss's law relates the electric flux through a closed surface to the charge enclosed by that surface. It is one of the four Maxwell's equations that form the foundation of classical electrodynamics.

Cricket Analogy: Think of the electric field as cricket fielding positions and the ball as the electric charge.

* The Field (Electric Field): Imagine the cricket field. The fielders are placed at specific positions to stop runs (like the electric field lines representing the direction and strength of the field).
* The Ball (Charge): The cricket ball is like an electric charge. A batsman hitting the ball (charge) creates a disturbance (field) that the fielders react to.

Gauss's Law Statement: The net flux (Φ) through a closed surface is equal to the total charge enclosed (Q\_enc) divided by the permittivity of free space (ε₀).

Mathematically:  
Φ = ∮ E · dA = Q\_enc / ε₀

The prompt is

Analyze well and give. If it is concept driven in the output field there should be classification, notes and explanation alone. If it is memory driven the output field must be classification, mnemonic and explanation alone, If it is foundational recall the output field must be classification, take away and explanation. Note that explanation should be in simple words and not exceeding 3 lines and mnemonic should related to the content, no vulgar words, simple words.

The output is

{

"instruction": "Explain how life originated on Earth, including theories and experiments supporting it.",

"input": "6.1 ORIGIN OF LIFE",

"output": {

"classification": "Concept-Driven",

"notes": "- The universe is ~13.8 billion years old; Earth formed ~4.5 billion years ago with no oxygen.\n- Early atmosphere had gases like CH₄, NH₃, CO₂, and water vapor.\n- UV rays and volcanic activity led to chemical reactions forming organic molecules.\n- Panspermia theory suggests life came from space; Spontaneous generation was disproved by Pasteur.\n- Oparin and Haldane proposed life came from non-living molecules through chemical evolution.\n- Miller’s experiment recreated early Earth’s conditions and formed amino acids.\n- First life forms were likely non-cellular molecules like RNA; cellular life came much later in water.",

**1.6 PYTHON PACKAGES:**

**Transformers :** Use pre-trained LLMs (GPT, BERT, T5, etc.) from Hugging Face. Easy API for text generation, classification, etc.

**OpenAI :** Access OpenAI’s GPT-3, GPT-4, etc., via API. Good for chatbots and text generation.

**PEFT :** For parameter-efficient fine-tuning (e.g., LoRA) of large models.

**Pdfplumber :** pdfplumber is a Python package that allows you to extract text, tables, and metadata from PDF files — accurately, including layout and formatting.

**PyMuPDF :** Used to read, extract, and manipulate text and images from PDF files.

**pandas** : Used for data manipulation and analysis in tabular form (DataFrames).

**requests :** Used to send HTTP requests and interact with web APIs.

**json :** Used to parse and generate JSON data.

**os :** Used to interact with the operating system (e.g., file handling, environment).

**time :** Used for handling time-related operations like delays and timestamps.

**re :** Used for pattern matching and searching text using regular expressions.

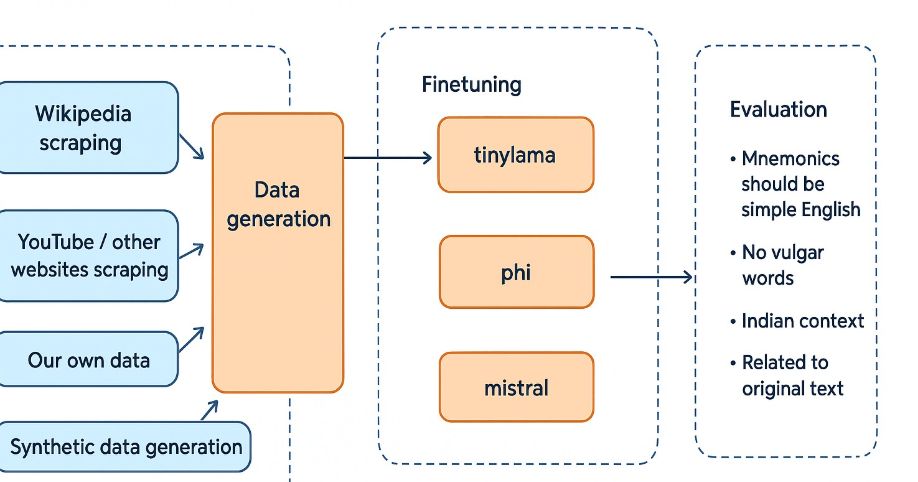
**datasets**: Access and manage datasets (especially for NLP) using Hugging Face’s library; integrates well with Transformers.

**sklearn.model\_selection** : Split data into training and testing sets for machine learning workflows.

**sklearn.preprocessing** : Convert text labels into numeric format for model compatibility.

**joblib**: Save and load Python objects (like models or encoders); optimized for large NumPy arrays.

1. **WORK FLOW:**

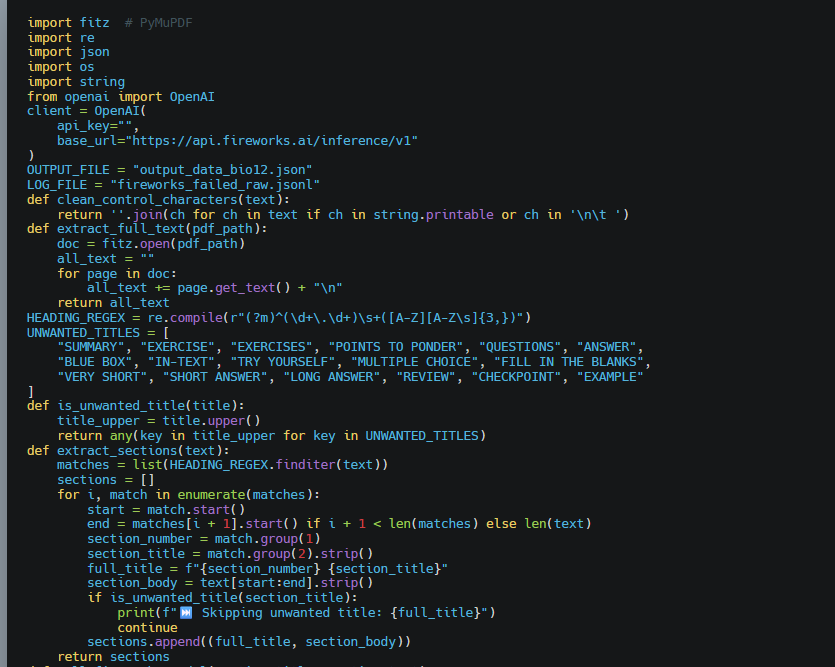


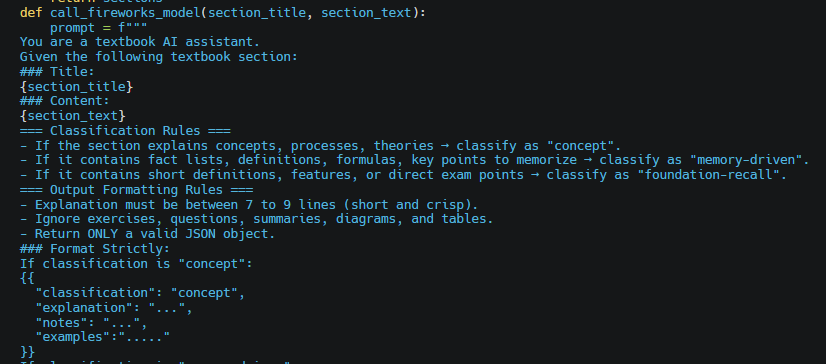
**2.1 SYNTHENTIC DATA GENERATION:**

To generate data for the extracted textbook content using a OpenAI API key in json format that includes instruction ,input and output it includes classification ,explanation, notes or mnemonics or takeaway taglines based on the classification.

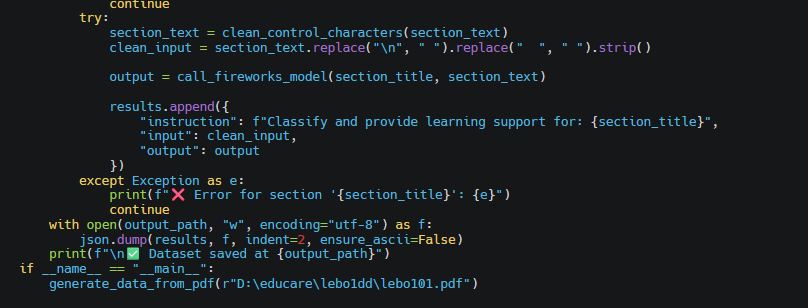
Code:

**data\_generation.py**









**Output:**

{"instruction": "Classify and provide learning support for: 1.1 Introduction",

"input": "1.1 Introduction\n\nMeasurement of any physical quantity involves comparison with a certain basic, arbitrarily chosen, internationally accepted reference standard called unit. The result of a measurement of a physical quantity is expressed by a number (or numerical measure) accompanied by a unit. Although the number of physical quantities appears to be very large, we need only a limited number of units for expressing all the physical quantities, since they are interrelated with one another. The units for the fundamental or base quantities are called fundamental or base units. The units of all other physical quantities can be expressed as combinations of the base units. Such units obtained for the derived quantities are called derived units. A complete set of these units, both the base units and derived units, is known as the system of units.",

"output": {

"classification": "concept",

"explanation": "Measurement in physics is fundamentally about comparing a physical quantity to a defined standard called a unit. Units serve as reference standards that are internationally accepted to ensure consistency and clarity in measurement. Although there are many physical quantities, they can be expressed using a limited set of fundamental units, known as base units. Derived units are formed by combining these base units to represent other physical quantities. Together, base and derived units constitute a system of units, providing a comprehensive framework for measurement.",

"notes": "- Measurement involves comparing a physical quantity to a standard reference called a unit.\n- Base units correspond to fundamental physical quantities and are internationally accepted.\n- Derived units are combinations of base units used to express other physical quantities.\n- The collection of base and derived units forms a system of units, ensuring uniformity in measurements."

}

}

**2.2 MODEL SELECTION:**

Evaluation Report: Model Selection for ED Tech Startup Application

**Overview**

The ED Tech Startup application, designed to assist students with concept-based learning through AI-generated explanations, required a language model that was not only lightweight and accurate, but also easy to fine-tune and deploy. Our selection process involved evaluating three compact models: Phi-2, Qwen1.5-1.8B-Chat, and TinyLLaMA (1.1B).

**Model 1: Phi-2**

We began our evaluation with Phi-2, given its small size (1.3B parameters) and reputation for strong reasoning abilities. However, practical testing revealed that the model frequently generated junk outputs—responses that were either unstructured, repeated, or unrelated to the input. The lack of formatting and technical clarity made it unsuitable for student-facing educational tasks. Additionally, Phi-2 is not instruction-tuned, which contributed to its inconsistent behavior.

**Model 2: Qwen1.5-1.8B-Chat**

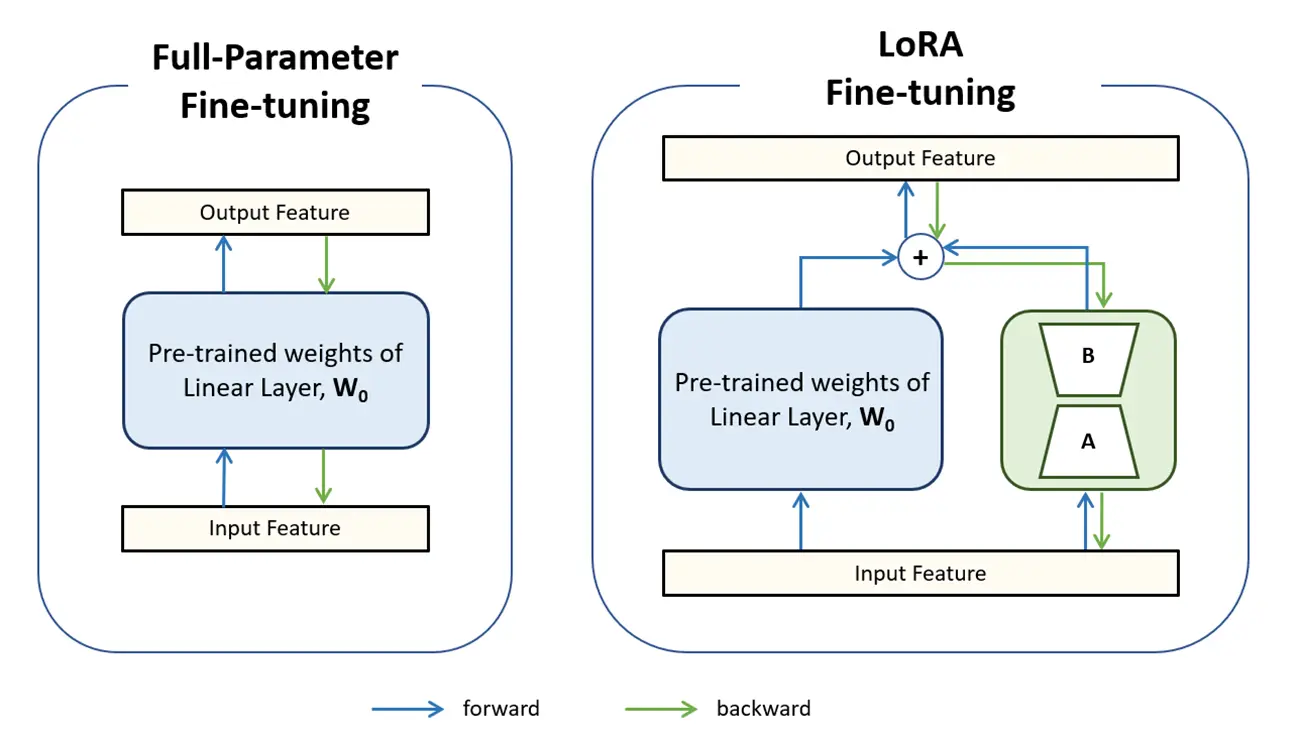
Next, we explored Qwen1.5-1.8B-Chat, which showed promising results in both general language understanding and domain-specific responses. We successfully applied fine-tuning to Qwen, and the model provided structured, technically accurate, and student-friendly answers. However, a significant bottleneck emerged in accessing and deploying the fine-tuned model. The lack of straightforward deployment tools or public endpoints made real-world integration difficult despite the model’s strong performance.

**Model 3: TinyLLaMA (1.1B)**

Finally, we tested TinyLLaMA, which was finetuned using the LoRA (Low-Rank Adaptation) technique on the Fireworks platform. This approach turned out to be the most seamless in terms of both training workflow and deployment. TinyLLaMA, although small, delivered coherent, properly formatted, and context-aware responses that met the needs of the ED Tech Startup application. The ease of integrating the model post-finetuning, along with reliable output quality, positioned TinyLLaMA as the most practical and efficient choice.

Among the three models, TinyLLaMA emerged as the best overall solution, balancing ease of use, quality of educational responses, and deployment readiness. While Qwen showed higher technical depth, its deployment complexity gave TinyLLaMA the edge for production use in ED Tech Startup.

**2.3 FINE TUNING PROCESS:**



|  |  |  |
| --- | --- | --- |
| Evaluation Aspect | Untrained Model Result | LoRA Fine-Tuned Model Result |
| Classification Accuracy | Correct: Concept Driven | Correct: Concept Driven |
| Original Explanation | Vague, unclear ball-car analogy, lacked technical depth | Clear: Talks about Newton's third law, internal energy loss, spring analogy |
| Clarity for Students | Slightly confusing, no structure | Clear cause-effect structure, momentum vs kinetic energy explained |
| Depth of Notes | Loosely worded, repetitive phrases | Precise, connects multiple concepts (elastic/inelastic, energy loss) |
| Accuracy of Notes | Generic wording, lacks correct physics terms | Matches HIGH SCHOOL terminology (e.g., “completely inelastic”) |
| Example Quality | Surface-level, mixed examples | Clear real-world examples (ball-wall for elastic, car crash for inelastic) |

**Full Fine-Tuning Issues You Faced:**

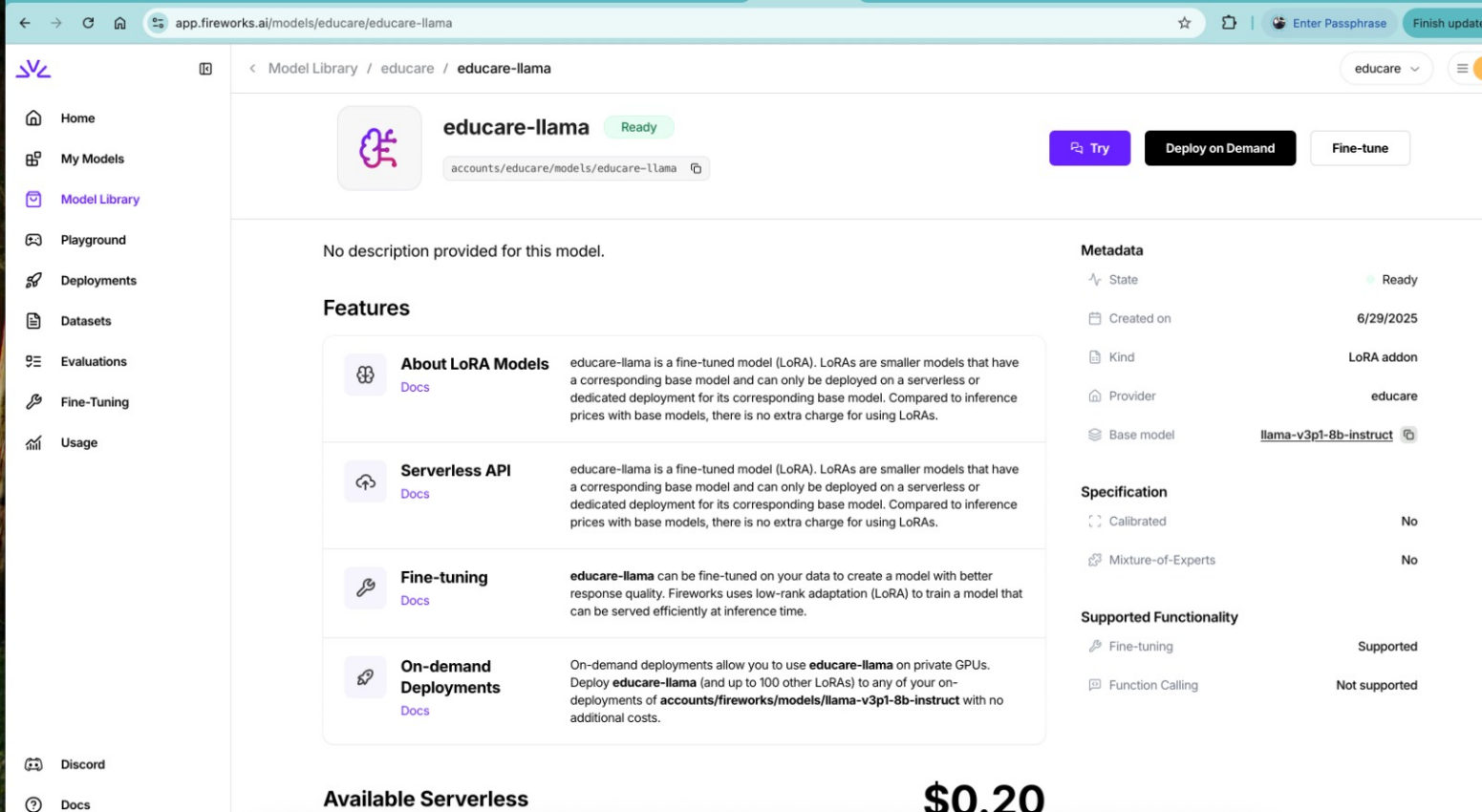
* Training took too long (required updating all model parameters)
* Risk of forgetting general language knowledge (catastrophic forgetting)
* Required a huge GPU with very high VRAM
* Difficult to run multiple experiments quickly

**How LoRA Solved These Problems:**

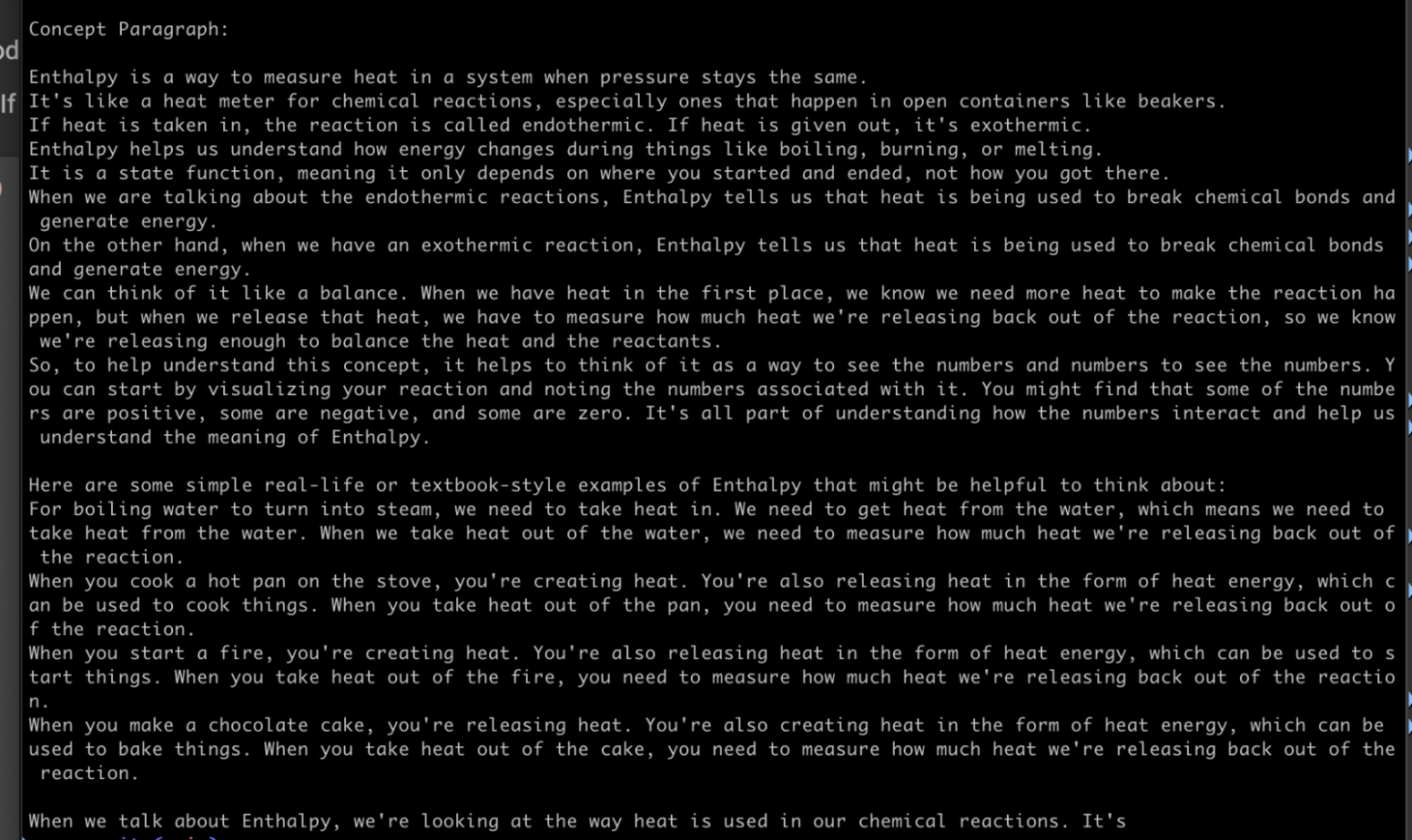
* Only small adapter weights were updated (much faster training)
* The base model remained frozen, preserving its original language skills
* Worked well even on smaller GPUs with limited memory
* Allowed faster, lightweight, and flexible iterations and experiments.

Automate the extraction of concept classification, simplified explanations, key learning points, and real-world applications of physics,chemistry,biology concepts.By switching from full fine-tuning to LoRA and using the Fireworks platform for deployment, we were able to efficiently customize and operationalize a large language model for structured educational outputs—despite significant hardware constraints. This pipeline now supports scalable educational content generation tailored to curriculum standards.

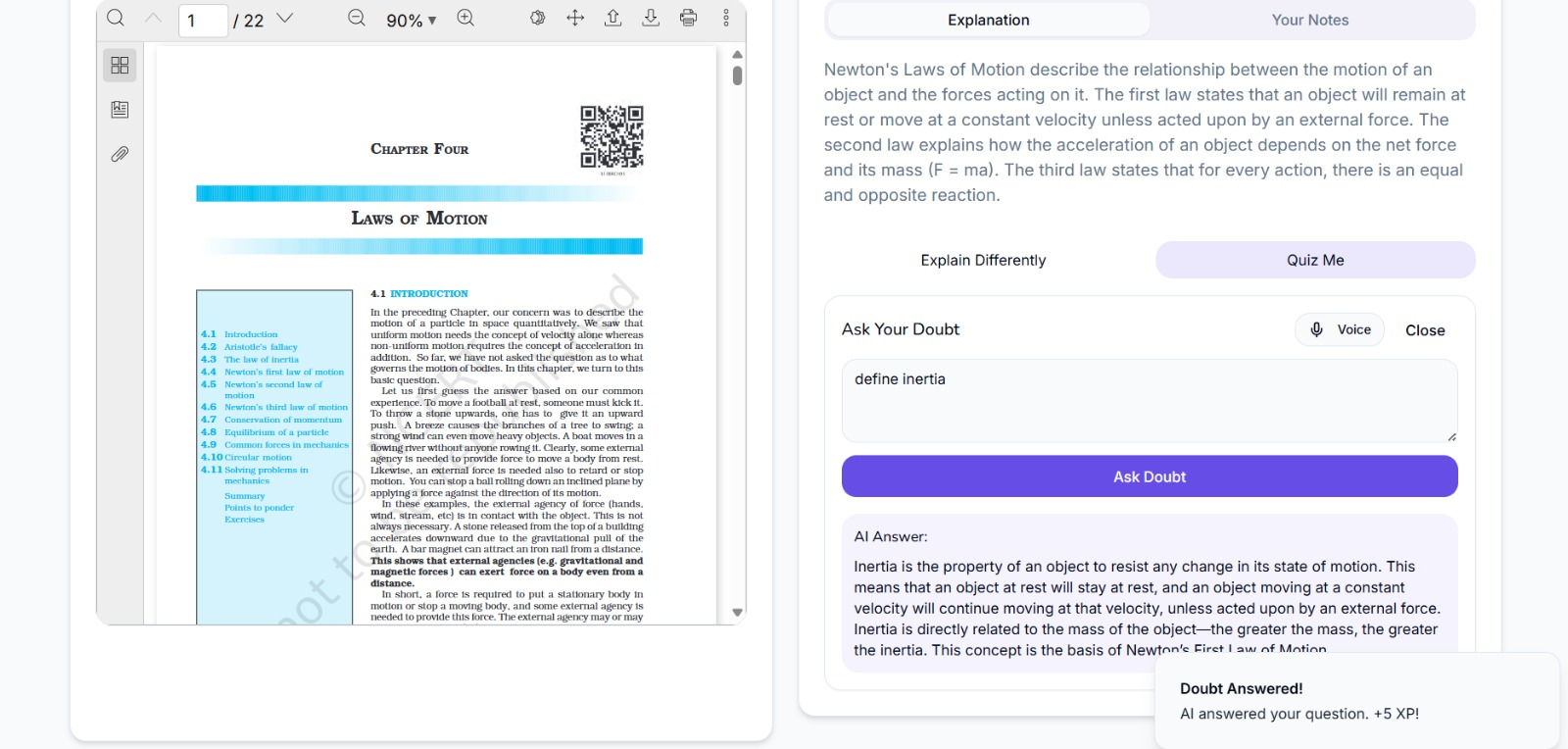
This is our trained model in fireworks using api key and also deploy the model in local and we test the model.

****

**After Training:**

****

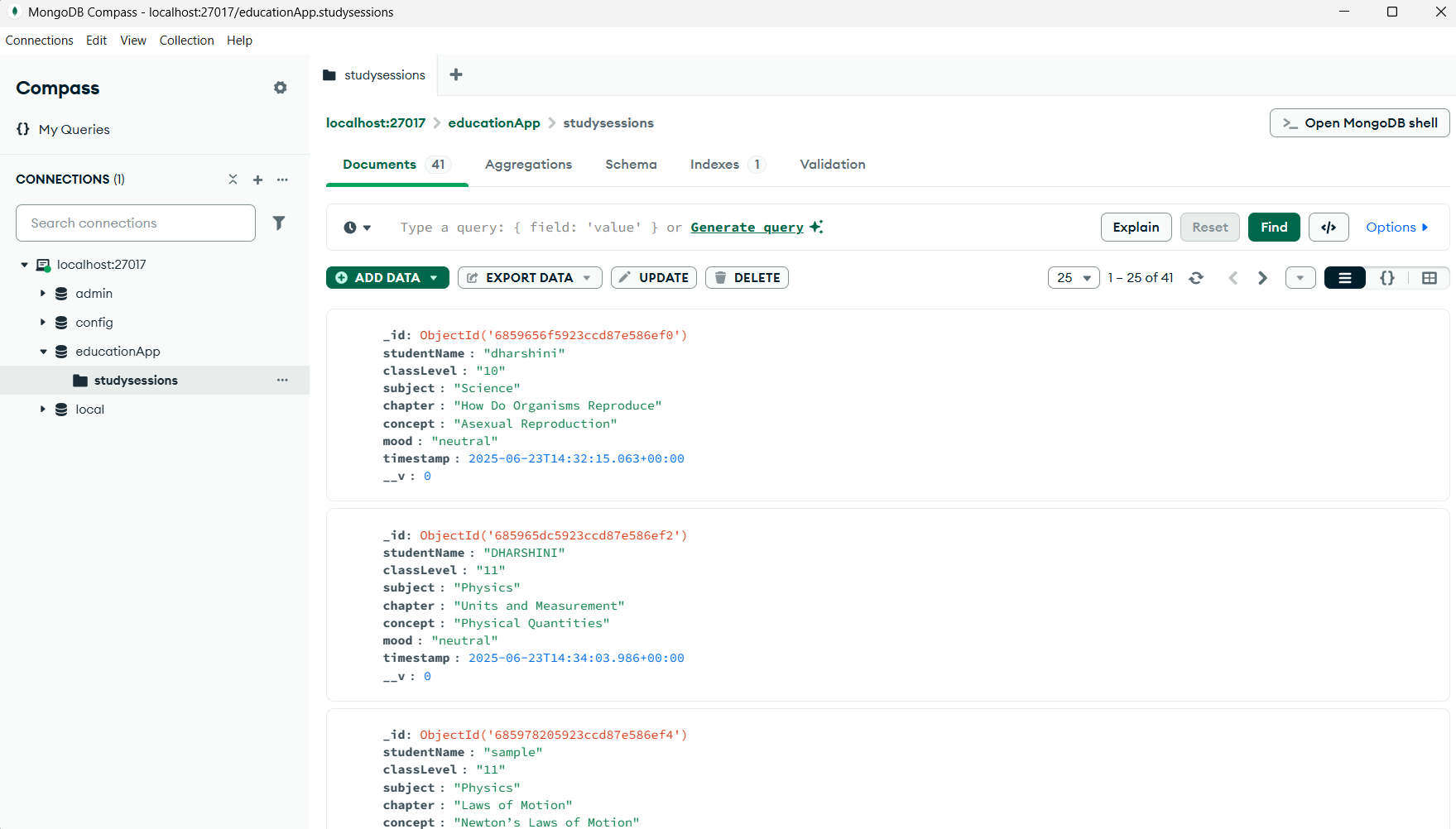
**3. DEPLOYMENT:**

****

**Database (MongoDB):**

**Features of MongoDB:**

* NoSQL:  
  Not a traditional table-based SQL database.
* Document-based:  
  Stores data as collections of documents, similar to JSON format.
* Schema-less:  
  No fixed structure for the data. You can store different types of data in the same collection.
* Scalable:  
  Easily handles large amounts of data, suitable for big applications.
* Flexible:  
  You can add new fields to data anytime without breaking existing data**.**

****

**4.CONCLUSION:**

We conclude that the ED Tech Startup Learning Assistant helps learners study in a distraction-free environment by providing explanations, mnemonics, takeaways, notes, and examples closely aligned with their textbook content. The topics are automatically classified into three categories: Concept-Driven, Memory-Driven, and Foundational Recall, ensuring that students receive learning support tailored to the nature of each topic. The LLM model used for this process was fine-tuned on the Fireworks platform and integrated into a web application for easy student access. Additionally, the data generation process is automated, using the original textbook content as input, which ensures that the responses generated are reliable, accurate, and curriculum-aligned. This makes ED Tech Startup a trusted and effective learning companion for exam preparation.